

Professional Expertise Distilled

# Learning Qlik Sense®

## The Official Guide

### *Second Edition*

Get the most out of your Qlik Sense® investment with the latest insight and guidance direct from the Qlik Sense® team

**Qlik**  | **Sense™**

Christopher Ilacqua  
James Richardson

Henric Cronström

**[PACKT]** enterprise   
PUBLISHING professional expertise distilled



# **Learning Qlik Sense® The Official Guide**

## **Second Edition**

---

# Table of Contents

[Learning Qlik Sense® The Official Guide Second Edition](#)

[Credits](#)

[About the Authors](#)

[Acknowledgments](#)

[About the Reviewer](#)

[www.packtpub.com](#)

[Support files, eBooks, discount offers, and more](#)

[Why subscribe?](#)

[Free access for Packt account holders](#)

[Instant updates on new Packt books](#)

[Preface](#)

[What this book covers](#)

[What you need for this book](#)

[Who this book is for](#)

[Conventions](#)

[Reader feedback](#)

[Customer support](#)

[Reviewing the example applications](#)

[Errata](#)

[Piracy](#)

[Questions](#)

[1. Qlik Sense® and Data Discovery](#)

[Continuing disruption](#)

[Qlik Sense® and the QlikView.Next project](#)

[Making sense of modern business](#)

[What is data discovery?](#)

[The empowered user](#)

[Interaction with data](#)

[Traditional business intelligence architecture](#)

[The Qlik® way](#)

[Color coding](#)

[Freedom of data navigation](#)

[Calculation on demand](#)

[Data discovery—the evolution of BI](#)

[Summary](#)

## [2. Overview of a Qlik Sense® Application's Life Cycle](#)

[Overview of an application's life cycle](#)

[Starting application authoring](#)

[What makes up a Qlik Sense® application?](#)

[Sharing an application](#)

[Continuing the application's life cycle](#)

[Summary](#)

## [3. Empowering Next Generation Data Discovery Consumers](#)

[Data discovery consumption requirements](#)

[Introducing the hub](#)

[Introducing streams](#)

[Exploring the components of the application](#)

[Sheets](#)

[Bookmarks](#)

[Data storytelling](#)

[Navigating and leveraging the associative experience](#)

[Navigation](#)

[Smart visualizations](#)

[Global search](#)

[Global filtering](#)

[Extending with Library](#)

[Summary](#)

## [4. Contributing to Data Discovery](#)

[Realities of the data discovery contributor](#)

[Creating private bookmarks](#)

## Creating and sharing private sheets

Creating a private sheet

Publishing a private sheet

Creating a new sheet

Adding a predefined visualization to a new sheet

Creating a Combo chart object

Publishing a private sheet

## Creating and sharing stories

Defining a story

Creating snapshots

Adding text

Adding shapes

Media library

Publishing your story

## Summary

## 5. Authoring Engaging Applications

### Preparations and requirements

The requirement specifications

The communication problem

A step-wise implementation

The process

### Getting started with the app creation

Creating a new app

### Loading your data

Loading additional tables

Using the Data load editor

Creating a database connection

Data connectors

### The analysis interface—sheets and visualizations

Creating a sheet

Adding visualizations

[Adding dimensions and measures](#)

[Defining bar charts](#)

[Storytelling](#)

[The application library](#)

[Which fields should be exposed?](#)

[Defining KPIs](#)

[Creating library entries](#)

[Best practices in data visualization](#)

[Dashboard](#)

[Analysis](#)

[Reporting](#)

[Structuring a sheet](#)

[Graphs and other visualizations](#)

[Dimensions and measures](#)

[The bar chart](#)

[The pie chart](#)

[The line chart](#)

[The KPI object](#)

[The gauge](#)

[The scatter chart](#)

[The tree map](#)

[The geographical map](#)

[Tables](#)

[Sorting and colors](#)

[Migrating applications from QlikView® to Qlik Sense®](#)

[Changes to the script](#)

[Changes to the user interface](#)

[Publishing your apps](#)

[Summary](#)

[6. Building Qlik Sense® Data Models](#)

[The QIX engine](#)

## [The Qlik Sense® data model](#)

[Creating a multitable data model](#)

[Linking tables](#)

## [Structuring your data](#)

[Normalization](#)

[Star schema and snowflake schema](#)

[Pitfalls in the data model](#)

## [The data model viewer](#)

[Using preview mode](#)

## [Summary](#)

## [7. Qlik Sense® Apps in the Cloud](#)

### [Why use the cloud?](#)

[Cloud sharing](#)

[Cloud content](#)

### [Using Qlik Sense® apps in the cloud](#)

[Uploading an app from the desktop](#)

[Creating an app in Qlik Sense® Cloud](#)

[Sharing an app in Qlik Sense® Cloud](#)

[Maintaining Qlik Sense® Cloud apps](#)

### [Using the Qlik DataMarket® content](#)

[Adding the QlikMarket® data](#)

## [Summary](#)

## [8. Extending the Qlik® Analytic Platform](#)

### [Qlik® Dev Hub](#)

[Web mashups](#)

[Extending the Qlik Sense® client](#)

[Engine API Explorer](#)

[Developer community – Qlik Branch](#)

## [Summary](#)

## [9. Administering Qlik Sense®](#)

[The Qlik Sense® architecture](#)



[Services](#)

[Clients](#)

[Applications](#)

[Nodes](#)

[Streams](#)

[Deployment and licensing](#)

[Single node or multinode](#)

[License and access passes](#)

[Tokens](#)

[Access rules](#)

[Management and monitoring](#)

[Importing and managing apps](#)

[Importing extensions](#)

[Users and user directories](#)

[Defining streams](#)

[Connectivity management](#)

[Tasks](#)

[System management](#)

[Security rules](#)

[Monitoring](#)

[Security](#)

[Authentication and authorization](#)

[Content security](#)

[Summary](#)

[10. Sales Discovery](#)

[The business problem](#)

[Application features](#)

[Who are our top customers?](#)

[The 360-degree customer view](#)

[Filtering customers](#)

[Reviewing shipments for top customers](#)

[Reviewing the bottom five customers](#)

[Who are our most productive sales representatives?](#)

[Analyzing products](#)

[Analyzing customer sales](#)

[Building the application](#)

[The SalesDetails table](#)

[The Customers table](#)

[The AggSales table](#)

[US States ISO CODE 2 polygons](#)

[Analyzing the Sales Discovery Library](#)

[Dimensions](#)

[Measures](#)

[Visualizations](#)

[Summary](#)

## [11. Human Resource Discovery](#)

[The business problem](#)

[Application features](#)

[Sheets](#)

[Training costs](#)

[Using the global selector](#)

[How the application was developed](#)

[Dimensions](#)

[Summary](#)

## [12. Travel Expense Discovery](#)

[The business problem](#)

[Application features](#)

[Tracking expenses](#)

[Analyzing expenses overspent](#)

[Digging deeper into the data](#)

[Creating an analysis story for travel expenses](#)

[Creating an overview](#)

[Sharing our analysis](#)

[Finishing the story](#)

[Developing the application](#)

[Examining the key tables](#)

[Expenses](#)

[PerDiemRates](#)

[Airfare](#)

[Department](#)

[Budget](#)

[LinkTable](#)

[Dimensions](#)

[Measures](#)

[Visualizations](#)

[Summary](#)

[13. Demographic Data Discovery](#)

[Problem analysis](#)

[Application features](#)

[Analysis](#)

[Using the lasso selector to make selections](#)

[Using the global selector to make selections](#)

[How the application was developed](#)

[Dimensions](#)

[Measures](#)

[Summary](#)

[Index](#)



# **Learning Qlik Sense® The Official Guide**

## **Second Edition**

---



# Learning Qlik Sense® The Official Guide

## Second Edition

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: February 2015

Second edition: December 2015

Production reference: 1211215

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham B3 2PB, UK.

ISBN 978-1-78588-716-1

[www.packtpub.com](http://www.packtpub.com)





# Credits

## **Authors**

Christopher Ilacqua

Henric Cronström

James Richardson

## **Reviewer**

Arthur Lee

## **Acquisition Editors**

Subho Gupta

Sam Wood

## **Technical Editor**

Pankaj Kadam

## **Copy Editor**

Laxmi Subramanian

## **Proofreader**

Safis Editing

## **Indexer**

Hemangini Bari

## **Production Coordinator**

Nilesh Mohite

## **Cover Work**

Nilesh Mohite



# About the Authors

**Dr. Christopher Ilacqua** is the research director of product validation at Qlik®. He has been working with leading customers and partners in the US for more than 3 years garnering feedback on Qlik Sense. Chris has over 25 years of experience in the field of planning and business intelligence, and he has established himself as a leading expert by advising, designing, and implementing hundreds of planning and business intelligence applications. He has a doctorate in business administration, a master's degree in business administration in accounting, and a bachelor's degree in marketing. His research interests focus on big data, data governance, mobile BI, SaaS, business collaboration, and cloud-based solutions. Additionally, Chris serves as an adjunct professor at New England College of Business, where he teaches graduate students strategic leadership, marketing, operations management, and MIS.

**Dr. Henric Cronström** is vice president product and technical product advocate at Qlik®, where he has worked for most of the time since the company was founded. At Qlik, Henric has had several different roles. For the first few years in Qlik's history, he was the product manager for QlikView, and then he moved into solution implementation and training. After many years in the field, including a role as a manager for the technical staff in QlikTech, Germany, he returned to Sweden as deputy manager for the development organization. In his current role, his main task is the communication of technical products on blogs, in the press, and directly with large accounts.

Henric has a doctorate in elementary particle physics from Lund University.

**James Richardson** is business analytics strategist at Qlik®. Prior to joining Qlik, James spent 6 years as a Gartner analyst covering business intelligence and analytics. During his tenure, along with advising hundreds of organizations on BI topics, James was the lead author of the Magic Quadrant for BI Platforms report and was the chairperson and keynote speaker for Gartner's European BI summit. Before Gartner, James spent 13 years at BI and performance management software vendor IMRS/Arbor/Hyperion in various roles. Prior to that, James worked for Sema Group developing software for industrial control systems (SCADA), and as an advertising copywriter. In 2015, he was named one of the UK's top 50 data leaders by Information Age magazine.



# Acknowledgments

As with any project of this size, there are a number of people who helped support this book. First of all, we would like to thank Anthony Deighton and Donald Farmer for their vision of Qlik Sense® and their executive support for this project. Additionally, we would like to thank Arthur Lee and the product management team at Qlik who supported the technical review of this book.

We would also like to thank all the Qlik employees that joined us over the last 3 years at customer sites to help solicit feedback and review parts of the text on Qlik Sense. These people include Jodi Antoni, Pierce Barber, Adrian Bereanu, Chris Brennan, Sean Donovan, Jeff DeAngelis, Dan English, Vinay Kapoor, Terry Kavouras, Mark Latessa, Arthur Lee, Erik Lövquist, Kevin Schulz, Joe Parker, and Elif Tutuk. Thank you for sharing your time, knowledge of Qlik products, and customers. This book could not have been written without the hundreds of hours of conversations with customers and partners.

The last four chapters of this book are based on the hard work of the Qlik demo team. These applications and new approaches to solve these business problems were created by Michael Anthony, Chuck Bannon, Alexander Karlsson, Jennell McIntire, Apeksha Patak, Francis Kabinoff, and Arturo Muñoz. Thank you for allowing us to bring life to your applications in this book.

Additionally, we wish to thank Brian Munz of the developer relations team and the Qlik legal team of Deborah Lofton, Alice Hume, and Heather Rossman for creating the legal framework that allowed us to explore Qlik Sense concepts with customers and partners. Also, we wish to thank the social influence and reputation management team of Maria Scurry and Toni Iafrate for their support and helping get the word out, both internally and externally, on the value of this project.

Finally and most importantly, we would also like to thank the companies who have contributed their time, resources, feedback, and passion for Qlik products for more than 3 years, which helped uncover the value of Qlik Sense in their organizations.



# About the Reviewer

**Arthur Lee** has been vice president of product management at Qlik<sup>®</sup> since June 2014. Arthur previously served various Qlik product management roles since joining Qlik in 2010. He is responsible for the product management function at Qlik. He has served as a business unit executive at IBM's business analytics business unit as a product manager among a variety of other product marketing roles at IBM/Cognos/TM1 from October 2004 to January 2010. Prior to IBM, he has held several financial and consulting roles within banks, retail, and software companies.

Mr. Lee holds a BA in finance and marketing from Boston College.





[www.packtpub.com](http://www.packtpub.com)

# Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.com).

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.packtpub.com](http://www.packtpub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at <[service@packtpub.com](mailto:service@packtpub.com)> for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

## **Why subscribe?**

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

## Free access for Packt account holders

If you have an account with Packt at [www.packtpub.com](http://www.packtpub.com), you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

# Instant updates on new Packt books

Get notified! Find out when new books are published by following @PacktEnterprise on Twitter or the *Packt Enterprise* Facebook page.



# Preface

Welcome! The purpose of this book is to help you learn about Qlik Sense®, Qlik's self-service visualization platform. Our aim is to help you get more from your data by applying Qlik Sense and its unique capabilities to your analytic needs. At the beginning of the book, we'll cover why Qlik chose to develop Qlik Sense, what data discovery is and can do, and the strategy and vision behind the product. In the subsequent chapters, we'll address practical considerations, including the Qlik Sense application's life cycle, how to meet the needs of different types of users, how to develop and administer engaging Qlik Sense applications, data modeling and getting the most out of the QIX engine. The book concludes by outlining a series of example applications built using Qlik Sense, to address analysis needs in sales management, HR, T&E management, and demographics.

# What this book covers

[Chapter 1](#), *Qlik Sense® and Data Discovery*, covers why Qlik chose to develop Qlik Sense, what data discovery is and can do, and the strategy and vision behind the product. This will include:

- Qlik's history in business intelligence and the evolution of data discovery
- The Qlik philosophy and approach to data discovery
- The difference between traditional BI and data discovery

[Chapter 2](#), *Overview of a Qlik Sense® Application's Life Cycle*, covers the Qlik Sense application's life cycle overview as a backdrop to highlighting key features and benefits of Qlik Sense in meeting these requirements. There are thousands of features in this initial release and this chapter will serve as a guide to the major components, features, and benefits to start your exploration of Qlik Sense. This will include:

- Overview of the hub
- Starting application authoring
- Components of a Qlik Sense application
- Sharing an application

[Chapter 3](#), *Empowering Next Generation Data Discovery Consumers*, highlights the key features in the context of specific user requirements that Qlik has identified as a consumer. This will include:

- Data discovery consumption requirements
- The hub
- Navigating and leveraging the associative experience

[Chapter 4](#), *Contributing to Data Discovery*, highlights the key features in the context of specific user requirements that Qlik has identified as a contributor or someone who seeks to share key findings from their analysis. This will include:

- Data discovery contributor requirements
- Bookmarks
- Private sheets
- Private stories
- Publishing to an existing application

[Chapter 5](#), *Authoring Engaging Applications*, highlights an approach to building engaging applications to meet your business problems. This will include:

- Data
- KPIs and measures
- Dimensions
- Security

[Chapter 6](#), *Building Qlik Sense® Data Models*, highlights the techniques and principles behind building a data model. This will include:



- Tables, fields, and records
- Joins and concatenations
- How to link tables and how not to link tables
- Normalization and de-normalization
- Synthetic keys and circular references

[Chapter 7](#), *Qlik Sense® Apps in the Cloud*, briefs us on using Qlik Sense Cloud. This will include:

- Sharing Qlik Sense Desktop apps in the cloud
- Creating Qlik Sense apps directly in the cloud
- Maintaining Qlik Sense Cloud apps
- Adding external data to apps using Qlik DataMarket

[Chapter 8](#), *Extending the Qlik® Analytic Platform*, gives an overview of the key APIs and how to use them to extend Qlik Sense solutions. This will include:

- API overview
- Dev Hub
- Single configurator
- Web mashup editor
- Extension editor
- Engine API explorer
- Qlik Branch overview

[Chapter 9](#), *Administering Qlik Sense®*, highlights the administration of Qlik Sense applications within an organization. This will include:

- Architecture
- Clustering and nodes
- Licenses and tokens
- Streams and security concepts

[Chapter 10](#), *Sales Discovery*, applies Qlik Sense to the challenges of analyzing sales performance within your organization. This will include:

- Common sales analysis problems
- The unique way Qlik Sense addresses these problems
- How the application was built

[Chapter 11](#), *Human Resource Discovery*, applies Qlik Sense to the challenges of analyzing human resource data. This will include:

- General information about common KPIs
- What a typical data model could look like
- An example of how to use the global selector
- Examples of dimensions and measures

[Chapter 12](#), *Travel Expense Discovery*, applies Qlik Sense to the challenges of analyzing travel expense management within an organization. This will include:

- Common travel expense management problems
- The unique way Qlik Sense addresses these problems
- How the application was built

[Chapter 13](#), *Demographic Data Discovery*, applies Qlik Sense to the challenges of analyzing demographic data. This will include:

- General information about common KPIs
- Examples of how to use the lasso selection with maps and scatter charts
- Examples of dimensions and measures



# What you need for this book

You will need a copy of Qlik Sense Desktop, which is available for free at <http://www.qlik.com/us/explore/products/sense/desktop>. After that, you just need some time and a good comfortable chair. Additionally, the sample application's examples and many others are available for you to explore live on <http://sense-demo.qlik.com/>. Please bookmark this link as additional demonstrations and examples are constantly being added and updated.



# Who this book is for

This book is for anyone who has struggled with decision-making and is looking for an intuitive way to analyze data through Qlik Sense. This can include casual users who are looking for an easy product that helps them understand their data and helps transform their data into actionable information. Additionally, this book provides information to analysts, developers, and administrators who seek to help organizations improve the performance of decision-making across their organization through the Qlik Sense enterprise server.



# Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: “Note that Pat has renamed the sheet to Pat-Product Analysis.”

Any command-line input or output is written as follows:

```
Load CustomerNo as CustomerID, ... From Table1 ... ;  
Load CustomerID, ... From Table2 ... ;
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: “There is no need to save the sheet but rather just click on the **Done** button.”

## Note

Warnings or important notes appear in a box like this.

## Tip

Tips and tricks appear like this.





# Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to <[feedback@packtpub.com](mailto:feedback@packtpub.com)>, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on [www.packtpub.com/authors](http://www.packtpub.com/authors).



# Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

# Reviewing the example applications

The examples and many others are available for you to explore live on <http://sense-demo.qlik.com/>. Please bookmark this link as additional demonstrations and examples are constantly being added and updated.

# Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the **Errata** section.

# Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at <[copyright@packtpub.com](mailto:copyright@packtpub.com)> with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

# Questions

You can contact us at <[questions@packtpub.com](mailto:questions@packtpub.com)> if you are having a problem with any aspect of the book, and we will do our best to address it.





# Chapter 1. Qlik Sense® and Data Discovery

In this chapter, we'll start getting to grips with what Qlik Sense offers by getting a better understanding of Qlik's background and how Qlik Sense was developed. In addition, we will examine the discovery-based approach to business intelligence that Qlik invented.

We will cover the following topics:

- Qlik's history in business intelligence and the evolution of data discovery
- The QlikView.Next project
- The Qlik philosophy and approach to data discovery
- The importance of the empowered user
- How a user really interacts with data
- The difference between traditional BI and data discovery

# Continuing disruption

In the world of technology, there's a lot of talk about creating new products that disrupt existing markets, but very few organizations can say they've done it for real. Qlik is one of them.

In 2007, the **business intelligence (BI)** software market changed forever. Oracle bought Hyperion, SAP bought Business Objects, and IBM bought Cognos. The conventional wisdom was that BI would effectively cease to exist as a standalone market, subsumed into larger stacks of technology.

However, this wasn't the case. In fact, by 2007, a revolution was already well underway. The BI world was being fundamentally disrupted, challenged by the new approach pioneered by Qlik (then called QlikTech). The disruptive technology Qlik developed was called QlikView. To differentiate QlikView from the established BI products, Qlik began to call the new disruptive approach **Business Discovery**, later adopting **data discovery** as this term gained industry-wide adoption.

Surprisingly though, when it was launched in 1994, what became QlikView was not consciously targeted at the BI software market. Rather, its initial task was to help its customer understand which of a number of individual parts and manufacturing materials were used across the range of the complex machines it manufactured, and which parts were *not* associated with particular items (a critical point we'll explore later in this chapter and revisit throughout this book). The goal was to visualize the logical relations between the parts, materials, machines, and products. This origin led to an approach completely different from BI at the time, one in which all the associated data points are linked automatically, enabling discoveries to be made through free exploration of data.

As it became more widely used and deployed, it was evident that what QlikView was being used for was a new type of BI. QlikView's speed, usability, and relevance challenged the standard approach that was dominated by IT-deployed data reporting products, which are slow performing, hard to use, and built around models that struggle to keep up with the pace of modern business needs.

QlikView's intuitive visual user interface, patented associative data handling—running entirely in memory—and its capability to draw data together from disparate sources changed the landscape. Discovery-led BI is about giving people the power to interact with and explore data in a much more valuable way than the older, reporting-led BI incumbency could. This is massively compelling to people who need to quickly ask and answer questions based on data in order to learn and make decisions, and proved very compelling to people jaded with the way things had been done before. QlikView became very successful, dominating the market it pioneered.

So what did Qlik do then? Sit back and relax, proud of its disruptive chops, safe in the knowledge that it had recast an established market in its image? No. Far from it. Instead, Qlik took the decision to transform the BI market again with a new product.



# Qlik Sense® and the QlikView.Next project

Qlik decided to design and develop a next generation data discovery platform. Developed under the project name QlikView.Next and launched as Qlik Sense, the product was anchored to five themes:

- **Gorgeous and genius:** Within this theme, Qlik focused on three product scenarios, with an overall emphasis on ad hoc analysis. The scenarios were that the product should be visually beautiful, support associative, comparative, and anticipatory analysis, and a seamless experience across all devices.
- **Mobility with agility:** This theme was about all users having access and the ability to answer new analytical questions as they arise in new situations and contexts when using a mobile device, with no difference between static and mobile experiences.
- **Compulsive collaboration:** Business intelligence and collaboration are inseparable; decision-making is, by nature, a collaborative activity. The intent was to build a product that could reside at the forefront of users' shared decision-making and give them the chance to communicate their insights through collaboration and storytelling.
- **The premier platform:** This theme was about enabling Qlik customers and partners to quickly and easily deliver apps and solutions that are perfectly relevant to their constituents. Within this theme, Qlik focused on four scenarios: data access, the development experience, expanding its ecosystem through broadened APIs, and offering a unified platform interface.
- **Enabling new enterprise:** With this theme, Qlik was focused on making capabilities such as security, reliability, and scalability available to all customers, not just the largest ones, and giving administrators and authors the same kind of gorgeous and genius experience other users get.

# Making sense of modern business

You may say, “Well, that’s all good, but it doesn’t really tell me why this matters or why Qlik Sense is important.”

To answer this, we have to think about what the focus of technology in our organizations has been in the recent past. For 25 years, most of our investment in IT has been on effectively improving reliability, using ERP or transactional applications to streamline processes, drive out inefficiencies, and deliver our products or services effectively. However, if most organizations, and particularly groups of competitors, are now operating at similar levels of procedural effectiveness, a key question arises, “What do we do differently to win?”

The answer lies in out-thinking our competitors through the use of data and analysis. This requires a shift of focus in both how we run our businesses and the IT world needed to do so. So far, analytics has too often been a poor cousin, something that happens afterwards on the edges, a tactical rather than strategic activity. That’s no longer good enough. Businesses using data-driven decision-making perform measurably better than those that don’t. When we can see (and measure) new things, we are driven to seek answers and thus, new ways of thinking and operating. Organizations that do not have analytics as a central part of their business activities will not thrive, or even survive, in the new reality.

Qlik Sense is about doing exactly that—freeing up the analytical skills of individuals in organizations, whatever their role. This book shows you how to make the most of that and alter how your organization uses information.



# What is data discovery?

So we've already mentioned the new style of BI that Qlik pioneered, data discovery, a few times. In this section, we'll look at that in more detail.

Over the years, there have been many names of the different business intelligence methods and tools, such as:

- Executive Information Systems (EIS)
- Management Information Systems (MIS)
- Online Analytical Processing (OLAP)
- Decision Support Systems (DSS)
- Management reporting
- Ad hoc query and reporting

Do we really need an additional label for something that in principle is the same thing? The answer is yes.

There is a fundamental difference between older technologies and data discovery, and it is in the approach. Most of the preceding tools are oriented towards technology, but data discovery is not. Instead, data discovery is oriented towards people—towards the users who need the information in their daily work.

Most of the preceding tools were developed for a small, select number of decision-makers, but again, data discovery is not. Data discovery is for everyone.

Decisions are made at all levels in a company. Obviously, managers are decision-makers, but we sometimes forget that machine operators and receptionists are also decision-makers, albeit at a more local level. They also need information to make better decisions.

We, at Qlik, believe that information can change the world and that every user can contribute to this transformation. Everyone should easily be able to view data, navigate in data, and analyze data. Everyone should be able to experience that “a-ha” moment of discovery.

Data discovery is not just business intelligence. It is user-centric, dynamic, and empowering. And it is fun!



# The empowered user

Since its founding in 1994, Qlik has believed that a business could improve its processes and product quality by empowering employees and encouraging them to engage in lifelong learning. And Qlik meant *all* employees—we saw everyone as a decision-maker, not just managers.

Although some things have changed since then, much remains the same. Users are still often in a situation where they are unable to analyze their data—data that they have the right to see, or should have the right to see, in order to do a good job. Rigid systems, technical limitations, and poor user interfaces are usually the culprits.

However, people's expectations of software have changed dramatically during the last decade. Applications from Google and Apple invite users to interact with simple, friendly interfaces. Search bars, Like buttons, and touchscreens have transformed the way people explore, consume, and share information. Today, people want the same ease of use from their business tools as they get from their consumer tools at home.

The current trends such as the consumerization of software, performance improvements of hardware, usability improvements of software, mobile devices, social networks, and so on just accelerate this change. All these trends are reshaping user behavior. Yesterday, a user was a passive end user, but the user of tomorrow will be both able and demanding. They will demand tools that are fast, flexible, and dynamic. They will demand tools that they can use themselves. The empowered user is here to stay.

# Interaction with data

The classic picture of business intelligence is that the user has one or several questions, and that the data holds the answers. So the problem boils down to creating a tool where the user can enter their questions, and the tool can return the answers.

However, this picture is incorrect. The truth is that the user does not always know the question initially. Or rather, if the user knows the question, they often already know the answer. So, the first thing the tool should do is help the user find the questions.

Finding the questions is a process that involves exploring the data. It involves testing what you suspect but don't know for sure. It also involves discovering new facts. Further, it involves playing with data, turning it around, scrutinizing the facts, and formulating a possible question. You use your gut feeling as a source of ideas and use the data to refine the ideas into knowledge; or to discard the ideas, if facts show that the ideas are wrong. You need to be able to play with the data, to turn facts around and look at them from different angles before you can say that you understand the data, and you need to understand the data before you can talk smartly about it.

When you have found a relevant question, you also need to be able to conduct an analysis to get a well-founded answer to the question.

Finally, the process involves presenting the answer to the question to other people as a basis for a decision or an action. The tool must support the entire process of going from ignorance to insight.

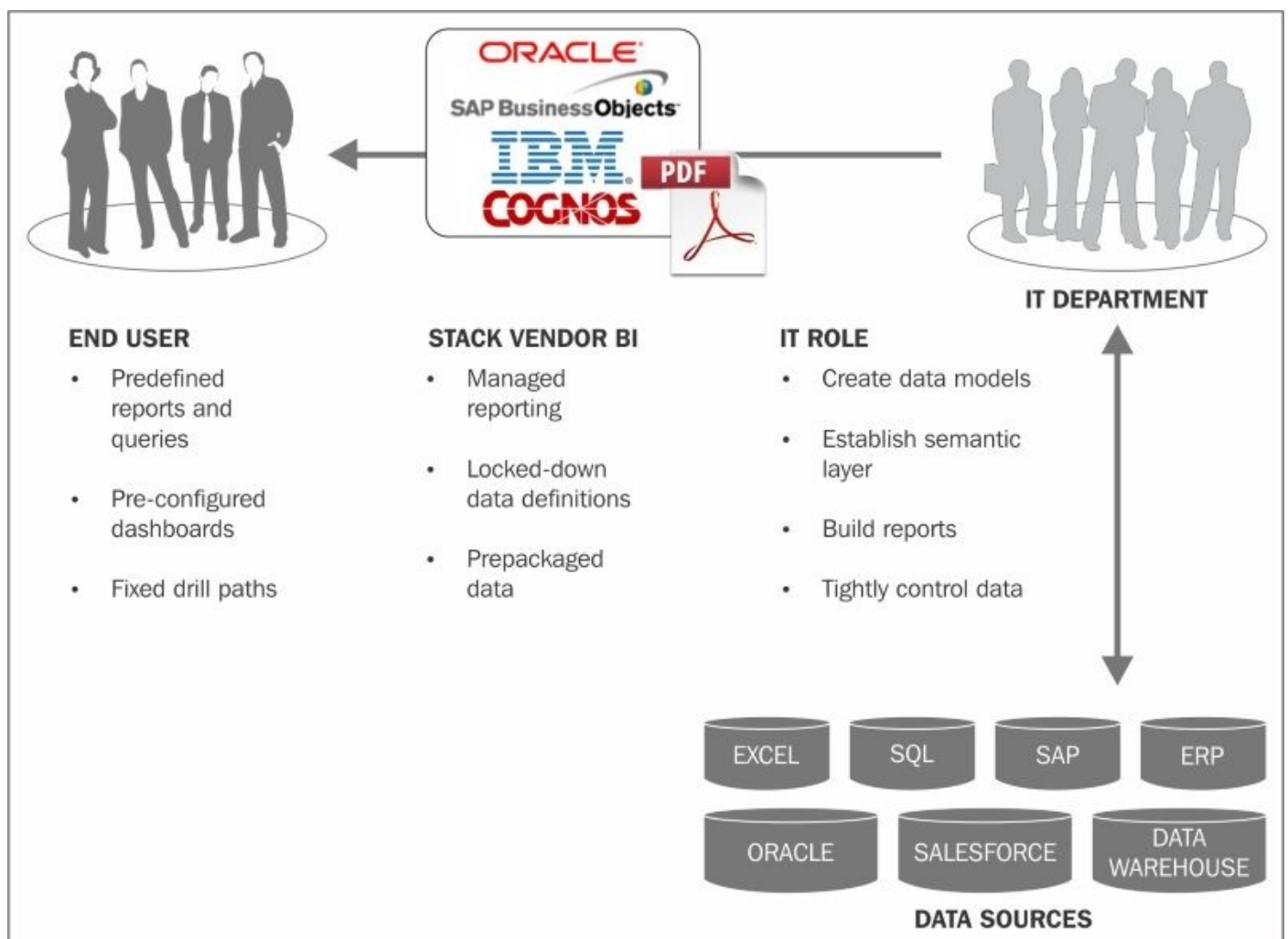
Hence, one major difference between data discovery and the more old-fashioned tools is that data discovery software supports the entire process—the process of coming from a blank mind, not knowing what you are looking for, all the way to attaining knowledge and taking action.

This is what data discovery is all about—helping you to prepare before you speak, act, or make a decision. It is the process of going from the darkness to the light, from the unknown to the known, from ignorance to insight. It is the process of going all the way from a blank mind to a substantiated argument.

# Traditional business intelligence architecture

It is quite clear that users representing the business want the ability to ask and answer questions on their own so that they can make better decisions, but traditional business intelligence solutions aren't well suited for user demands. Instead, it is common that the systems are created in a report-centric manner, where governance and system demands set the goals, rather than user demands. The solutions often have preconfigured dashboards, fixed drill-down paths, predefined queries, predefined views, and very little flexibility.

With traditional BI, the creation of the business intelligence solution often belongs to the IT organization, which has to do the following: create data models, establish a semantic layer, build reports and dashboards, and protect and control the data. Often, the creation of business intelligence solutions is not driven by user demands. The following figure depicts the traditional BI architecture:



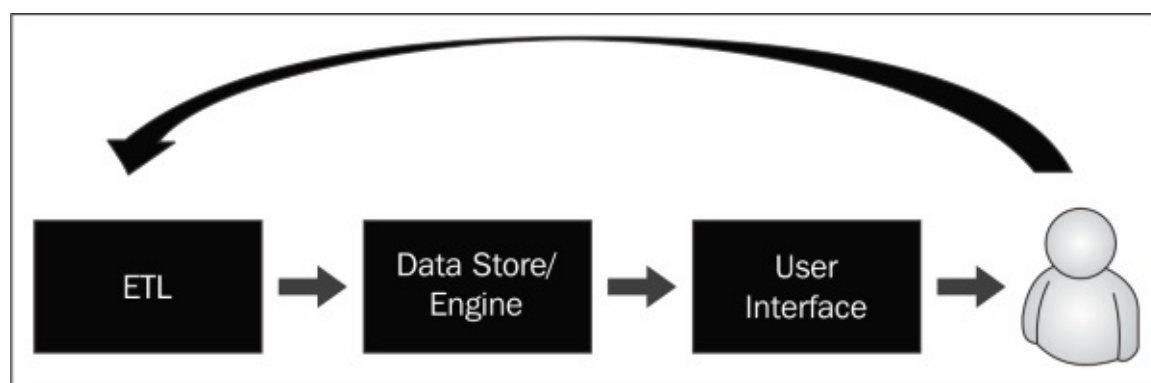
When analyzing data, you might want to set filters so that you can make selections, but with traditional tools, you often need to start at the top of predefined hierarchies. So, instead of selecting a customer directly, you may need to answer this question, “Which market does this customer belong to?”, then the country the customer belongs to, and only then can you specify the customer.

Further, in the drill-down hierarchy, you are often limited to the choice of one or all. For example, you can look at either a single customer or all of them. The possibility of choosing two or three specific customers doesn't exist, unless this has been specifically predefined by the data model developer.

Numbers are often precalculated to ensure short response times, but this has a drawback that if the developer hasn't anticipated a specific calculation, the tool will not be able to show it.

Further, the architecture of the tool is often made in three layers—referred as the **stack**. The first layer is the **Extract, Transform, Load (ETL)** layer, or the data load layer. The second is the **Data Store / Engine** layer, and the third is the **User Interface (UI)** layer. The three layers are different pieces of software, sometimes delivered by different software vendors.

These three layers also demand different skill sets. Often, the ETL expert knows little or nothing about the UI software, and the UI expert knows little or nothing about the ETL.



*The product stack in traditional BI*

This architecture also leads to problems. When an application is built, the feedback comes from users trying to use the application. It could be that KPIs are incorrectly calculated or that dimensions or measures are missing. It could also mean that the user realizes that the initial requirements were incorrect or insufficient. The feedback could imply changes in the UI, or in the data model, or even in the ETL component.

This type of feedback is normal—it happens with all business intelligence tools. It only means that the development of applications is a process where you need to be agile and prepared. The expectation that you should be able to define an application completely and correctly prior to a prototype or an intermediate version is just unrealistic.

This is where the architecture leads to problems. In order for a project to be successful, you need to be able to implement change requests and new user demands with short notice, and this is extremely difficult since three different pieces of software and three different groups of people are involved. The distance between the user and the ETL component is just too great for efficient communication. Hence, the traditional architecture leads to a broken process.



# The Qlik<sup>®</sup> way

Qlik has tried to solve all the drawbacks discussed in the preceding section by doing things differently.

First of all, you click and view. You don't need to formulate your question or tell the system more specifically what you want to look at. You just click, and by that, you say, "Tell me more about that...". Then you look at the calculation, KPI, or the field that might be interesting.

# Color coding

The color coding defines the answer. Some things are associated with what you clicked on, and they remain white. Others that are not associated become gray. The color coding is for simplicity. The user quickly gets an overview and understands how things work.

Showing the excluded reveals the unexpected, creates insight, and creates new questions. Hence, the gray color is an important part of making the Qlik experience an associative one—a data dialog and an information interaction—rather than just a database query. Showing you that something is excluded when you didn't expect it means answering questions you didn't ask. This surprise creates new knowledge in a way that only a true data discovery platform can.

# Freedom of data navigation

Via the associative experience, a user has total freedom to navigate through data and make any combination of selections. Any number of values can be selected. No drill-down paths need to be predefined. This allows the user to follow their own train of thought instead of someone else's. Start anywhere and just follow your intuition.

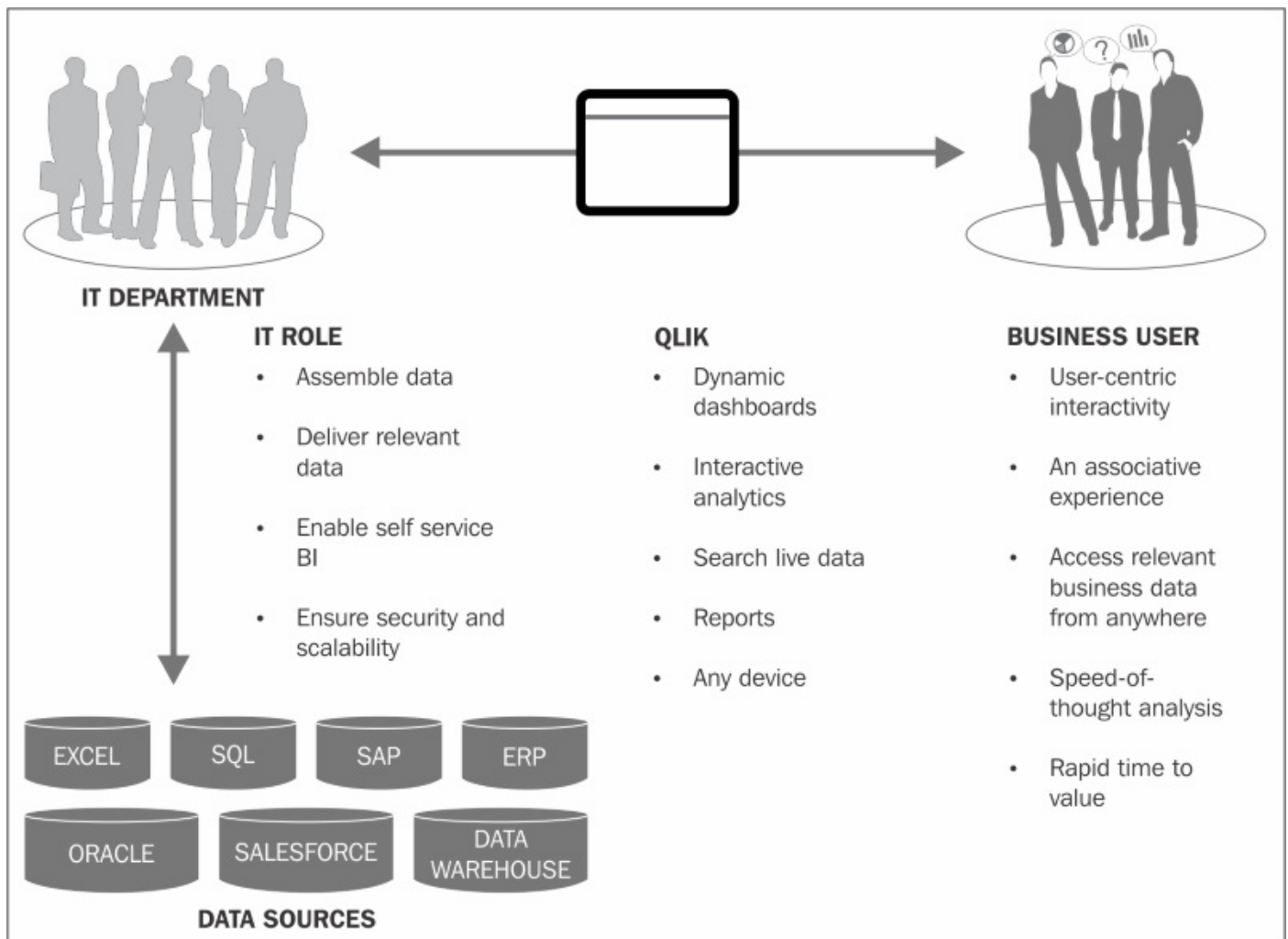
This total freedom when exploring data is really the core attribute of data discovery.



# Calculation on demand

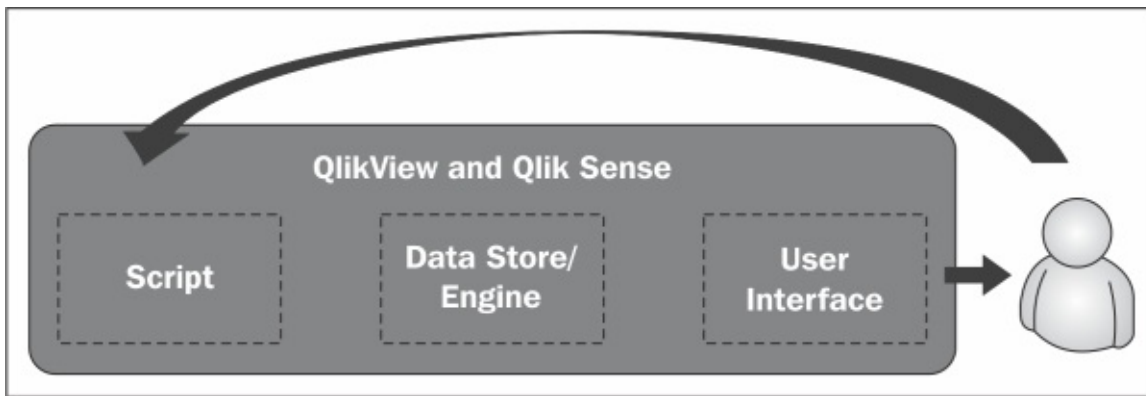
Further, no numbers need to be precalculated. Via the QIX engine, QlikView and Qlik Sense calculate everything on demand, usually in a fraction of a second. The short response time allows the user to *have a conversation* with the data, where one answer leads to the next question, which in turn leads to next, and so on. Only this way can you interact with data so that you learn from it.

The developer does not need to anticipate all questions that the user will pose. All they need to do is to create a logical, coherent data model, and Qlik Sense will be able to answer the question correctly:



The stack (ETL-Data Store / Engine-UI) is replaced by a single integrated environment. This makes it possible to develop applications in close cooperation with the users, and it can often be done by the users themselves. Feedback is implemented instantaneously and the changes can be evaluated just seconds later. This shortens the development cycle and ensures that the application meets the user demands much sooner than it would otherwise.

This stepwise implementation is crucial for the success of a business intelligence project. It is also the core of modern agile methodologies that are used in all types of software development.



*With Qlik Sense, all BI stack functions are integrated into one tool*

The development of business intelligence applications must be done as close to the user as possible to enable user feedback and short development cycles. It does not necessarily imply self-service capability, although it is good if this capability exists.

With the introduction of Qlik Sense, the groundbreaking work continues by enabling a new class of users who are highly mobile and require greater self-service capabilities. In Qlik Sense, the self-service capability has become a core feature. Users can define new graphs and visualizations that the app developer didn't think of. This functionality empowers the users even further.

With Qlik Sense, it has also become easier to share your findings and communicate them. This is something that is necessary in all environments where human interaction is important, which is pretty much everywhere.



# Data discovery—the evolution of BI

Data discovery is the future of business intelligence. With data discovery, users pursue their own path to insight, make discoveries collaboratively, and can arrive at a whole new level of decision-making. Users are not limited to predefined paths or precalculated numbers. They do not need to formulate questions ahead of time. They can interact with data, find the questions, ask what they need to ask, and explore up, down, and sideways, rather than only drilling down in a predefined hierarchy.

Organizations might still need standardized reporting for many cases, but data discovery is the approach that ultimately fulfills the promise of business intelligence for everyone.

Data discovery is the inevitable consequence of demands from active users who want information from the ever-increasing amount of data. From the very beginning, the core of the Qlik philosophy was the empowered user. It affects both the view of how BI solutions should be developed and how the user interface of the tool should be designed.

In summary, data discovery is user-centric; it is BI for the empowered user. It means total freedom in how data is explored. It should be simple and have as few limitations as possible. Data discovery means a user-centric development process so that user feedback can be implemented instantaneously.



# Summary

In this chapter, we looked at why Qlik developed Qlik Sense and at the ethos and value of data discovery in contrast to older forms of BI.

In the next chapter, we will look in detail at Qlik Sense itself and how its features help in meeting these requirements, beginning with the application life cycle.



# Chapter 2. Overview of a Qlik Sense® Application's Life Cycle

In the previous chapter, we outlined the evolving requirements driven by the market, and more importantly by business users seeking to help make better decisions within their organization. This chapter's goal is to highlight key features and benefits of Qlik Sense in meeting these requirements. There are thousands of features in the initial release of the software, and this chapter will serve as a guide to the major components, features, and benefits of Qlik Sense as you start exploring it.

In this chapter, we'll cover the following topics:

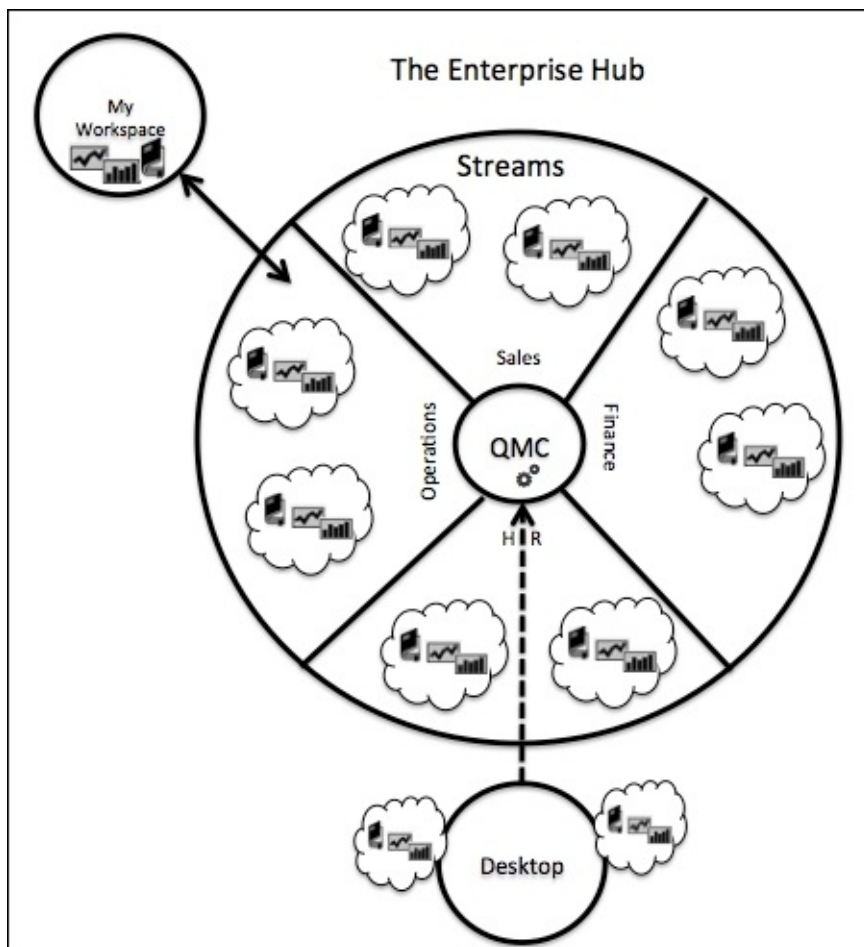
- Overview of the hub
- Starting application authoring
- Components of a Qlik Sense application
- Sharing an application



# Overview of an application's life cycle

As we begin our overview of a Qlik Sense application life cycle, it is best to start at the center of a Qlik Sense community collaboration, which is called the hub. The hub is made up of a number of streams that contain applications that are published by authors as well as users who can extend these applications by adding personal sheets and data stories. The **Qlik Sense Management Console (QMC)** governs this publishing through streams that have security rules. This approach provides the highly governed system that IT needs, while granting users the ability to explore information and share and collaborate on their findings.

Let's dig a bit deeper in each of these areas:



*Overview of the Qlik Sense hub*

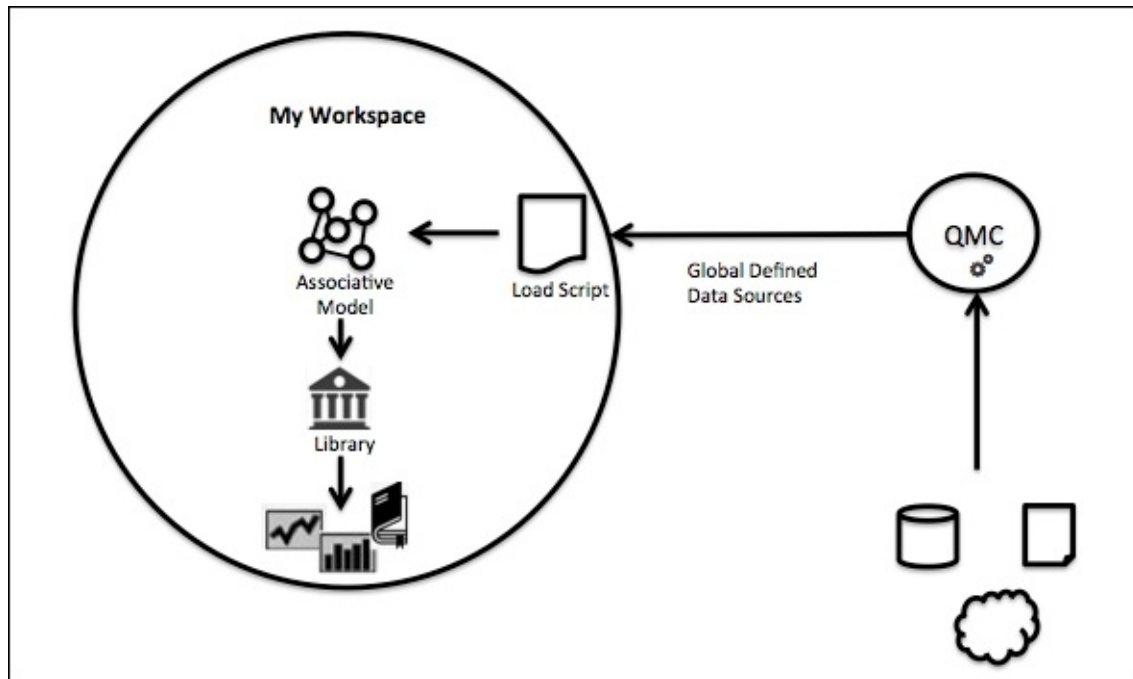
# Starting application authoring

The need for a Qlik Sense application often starts with simple questions, such as these: Why are sales down in my region? What products are not selling well? Are there opportunities to sell additional products to existing customers? When a customer purchases a product, do they also purchase a companion product? These types of questions lead to the identification of the place to find this data. Qlik Sense provides two starting points that can be from either the Personal (Desktop) or Enterprise Edition. This chapter will focus primarily on the Qlik Sense Enterprise Edition and mention the differences in the Desktop Edition.

The hub is made up of two main parts. The first is **My Workspace**, which enables users to create new Qlik Sense applications. The second part comprises defined **Streams**, which contain published applications to be used and extended by users. Streams are defined in the QMC, which provides a broad range of security rules to meet organizational requirements. Once an application is completed, it can be published to an authorized stream by the author. When published, the application cannot be altered without republishing by the author. The Desktop Edition contains only a hub for the creation of Qlik Sense applications and the application author must send all artifacts of the application, which must include at least the Qlik Sense document (QVF) and extensions used in the development of the application. Once received by the administrator, these artifacts are imported through the QMC and then published.

# What makes up a Qlik Sense® application?

Now, let's turn our attention to what components make up a Qlik Sense application. They are shown in the following diagram:



*A Qlik Sense application component*

Qlik Sense applications are made up of a number of components. Starting from the data source, these components include the following:

- **Global Defined Data Sources** are defined outside of Qlik Sense and managed by QMC.
- Based on these governed data sources, a **Load Script** is generated through or written, which transforms this data into Qlik's in-memory data model.
- Once the Qlik Sense data model is defined, the author can determine which fields will have the most value for users in the creation of private sheets for personal analysis. These fields will be used to create dimensions and real-time calculation expressions for measures.
- Additionally, fully defined charts for the most common views of information can be stored in the **Library**.
- Once the Library is defined, sheets (collections of objects), data stories, and bookmarks can be created.

All these components combine to create a dynamic baseline application to be explored by users.

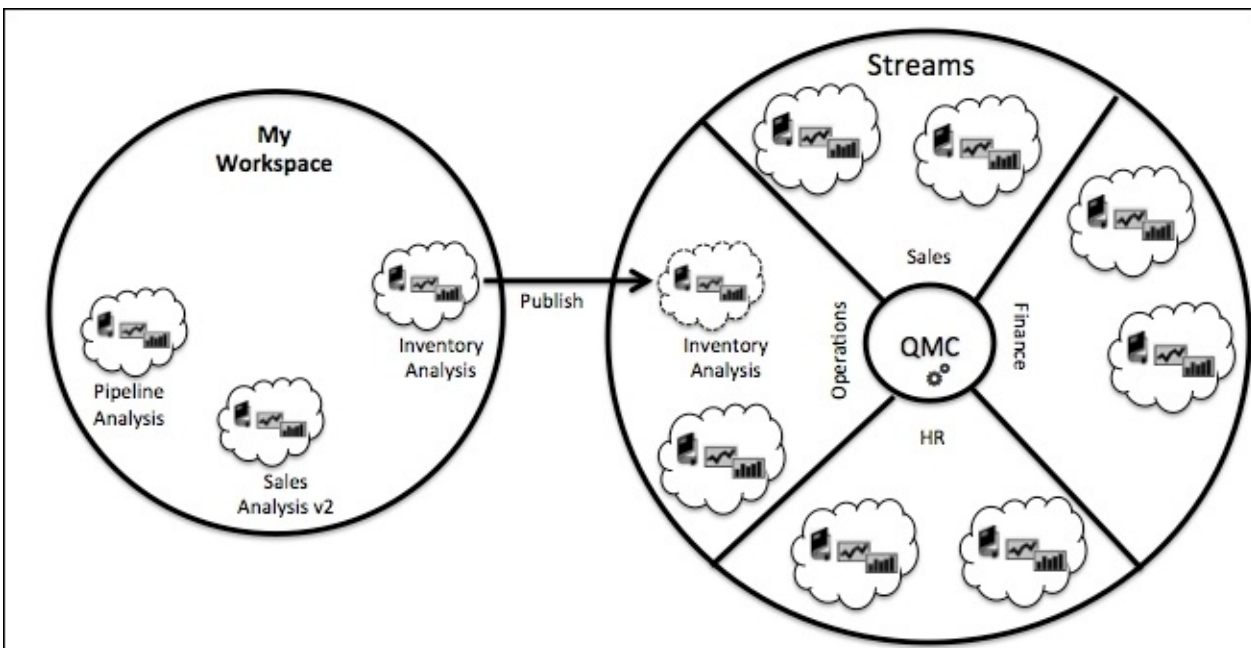
# Sharing an application

Let's turn our attention to how an application is shared with the Qlik Sense community. There are two methods:

- Qlik Sense Enterprise application publication
- Qlik Cloud



Once a Qlik Sense application is complete, the author can share it by publishing it to a stream in the Qlik Sense hub. The publishing process can be accomplished by an administrator who is responsible for a stream and has publishing rights in the QMC. A Qlik Sense author notifies a stream administrator that a Qlik Sense application is ready for publishing. The stream administrator logs into the QMC, identifies the Qlik Sense application by name and author, and publishes the application in a stream.

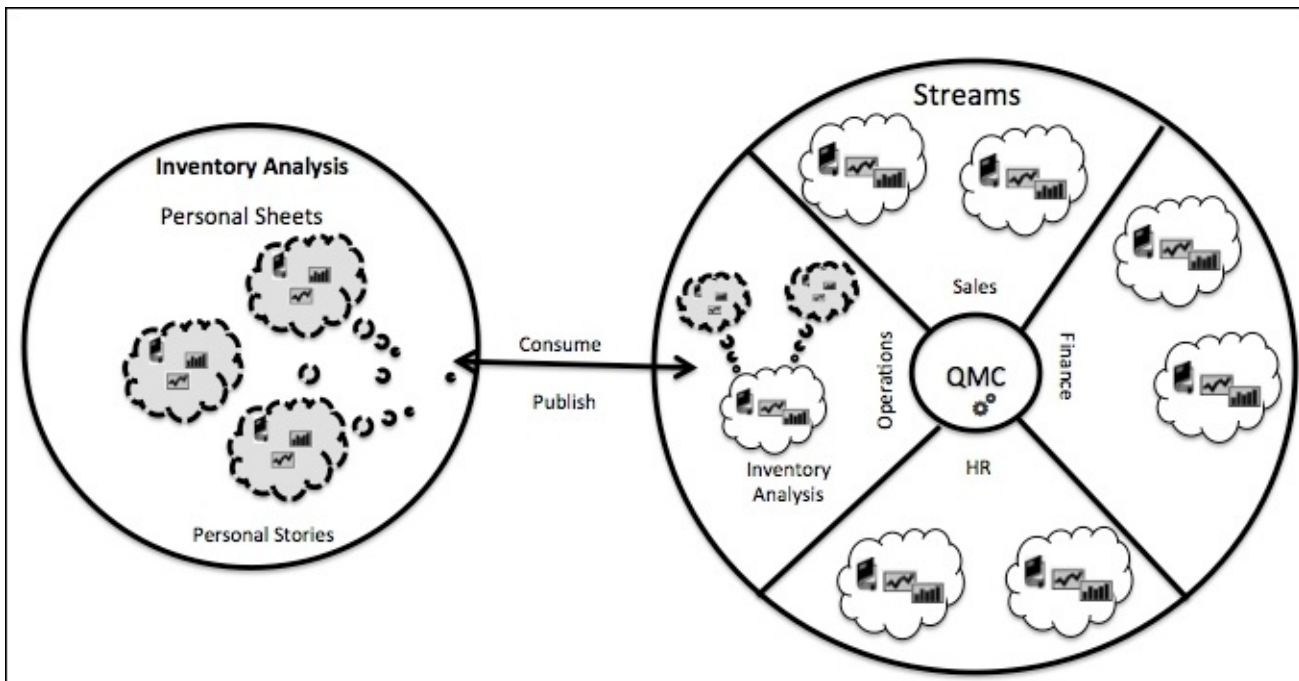


*Qlik Sense Enterprise application publication*

## Note

This method moves the application from the personal workspace, so a copy of the application should be made prior to publishing. Publishing and best practices for delegating publishing rights to an author in the QMC will be discussed in more detail in [Chapter 9, Administering Qlik Sense®](#).

Once an application is published to a stream, it is ready to be explored by users:



*Qlik Sense Enterprise application consumption*



### *Qlik Sense Cloud*

Qlik provides a free and easy way for up to five people to create and share Qlik Sense visualizations in the cloud. There are two ways to take advantage of Qlik Cloud:

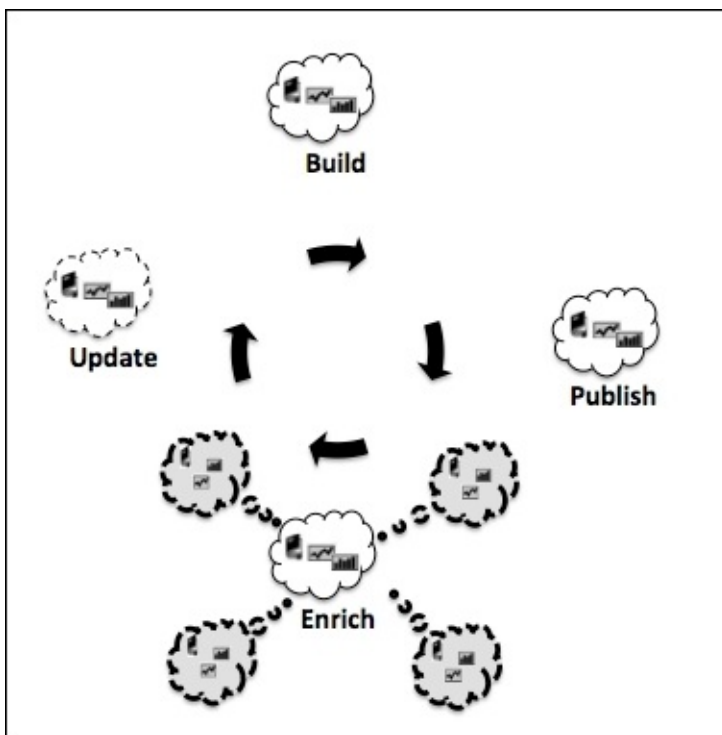
- Download Qlik Sense Desktop and create an application. Once created, register via the Qlik Sense client and upload your application.
- With the release of Qlik Sense Cloud, anyone can start their data exploration immediately by registering at <http://www.qlik.com/us/explore/products/qliksensecloud> and create your application and share it directly on the cloud.

Qlik Sense Cloud will be explored in more detail in [Chapter 7, Qlik Sense® Apps in the Cloud](#).

# Continuing the application's life cycle

One of the key features of a Qlik Sense application is its dynamic nature, which helps meet the broad requirements of data discovery. Users can explore the published sheets and data stories as well as create and share private sheets and stories based on the application library. The library allows for the creation of personal sheets and data stories in a controlled manner. As mentioned earlier, the library is a collection of dimensions, measures, and charts that are defined by the application author and cannot be modified once they are published to a stream but can be republished from the author's workspace. It allows a user to extend an application and share findings through personal sheets and data stories, while keeping consistent definitions across an organization.

Taking a step back, let's look at this new application model. A published Qlik Sense application is just at the beginning of its life cycle. Once published, the application can be expanded by the contributor within the stream using additional published sheets and stories based on the original application.



*Qlik Sense application life cycle*



# Summary

Enterprise Qlik Sense applications are built based on governed data sources defined in the QMC. These data sources are transformed into a QIX Engine. Once the model is defined, key dimensions and measures are created within the library. This library will be used to create sheets. Next, the application is published to a stream within the hub for consumption. The application is then explored by users, and key findings can then be shared through bookmarks, private sheets, and data stories. These artifacts enrich the application and can be published back in the stream for collaboration between other members of the stream.

The next chapter will explore each of these capabilities in more detail and how they meet the needs of key stakeholders within your organization.





# Chapter 3. Empowering Next Generation Data Discovery Consumers

In the previous chapter, we outlined the Qlik Sense application life cycle, which provided an overview of the key Qlik Sense application components. This chapter's goal is to highlight key features in the context of the specific user requirements that Qlik has identified as defining a data discovery consumer.

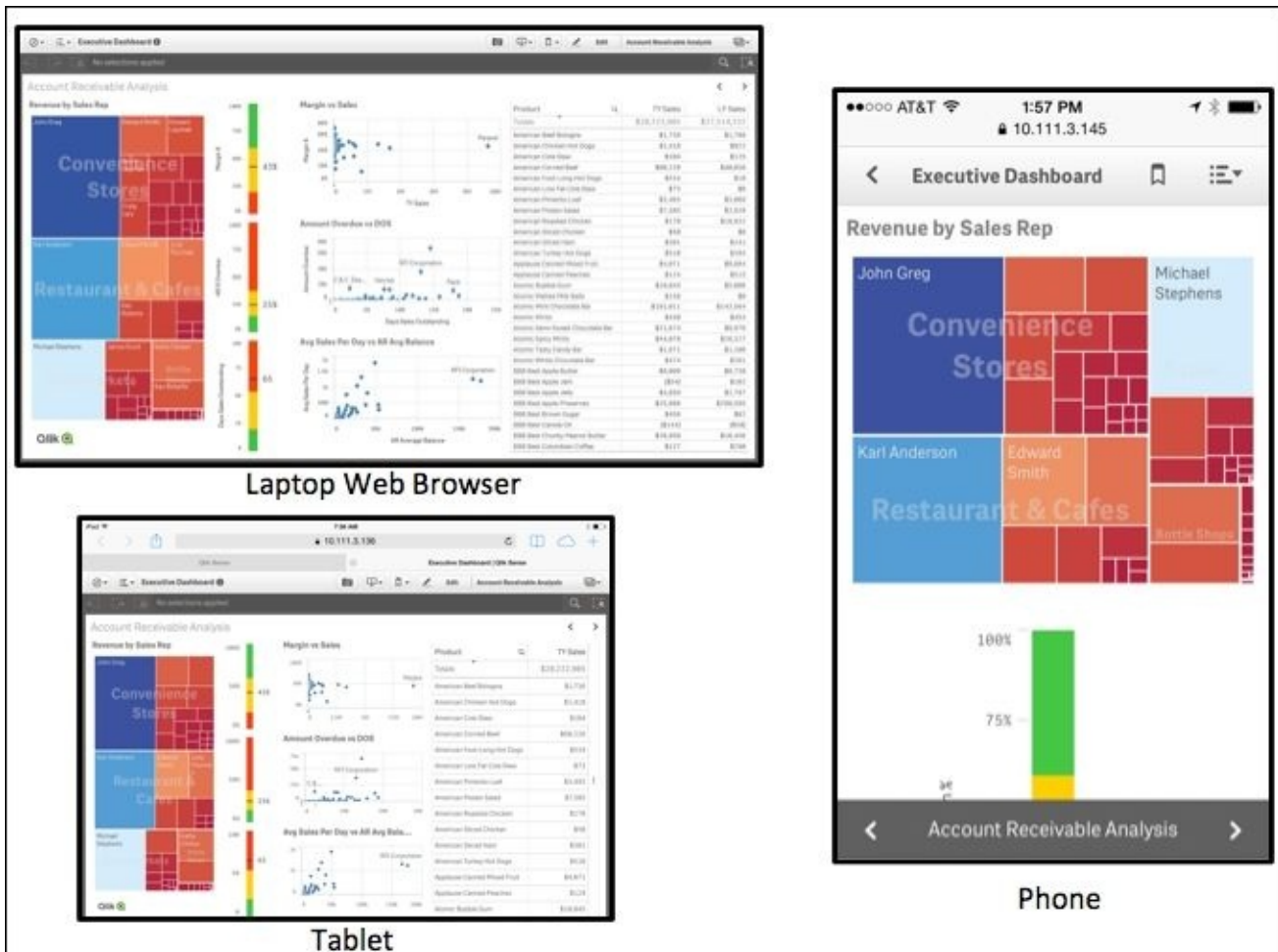
In this chapter, we'll cover the following topics:

- Data discovery consumption requirements
- The hub
- Navigating and leveraging the associative experience

# Data discovery consumption requirements

People's expectations of what technology should be and how it should work have been set high with the rise of mobile and touch devices. The notion of a fixed, predictable desktop has changed to a dynamic, unpredictable virtual desktop that exists on whatever device you have access to at the moment. This can include traditional desktop PCs running Windows, laptops, ultrabooks powered by Microsoft Windows, Apple Mac OS, hybrid devices running Windows 8.x, tablets, Chromebooks, smartphones... the list goes on. This new environment requires new approaches in both architecture and application design that create smarter applications to meet the demands of a broader access from varying devices. Qlik Sense was designed from the ground up to meet the diversity of requirements that now exist in your enterprise when it comes to delivering data to support decision-making.

Qlik Sense adapts to very different devices, including a laptop via Microsoft Windows, Apple iPad Air, and finally, an iPhone 5s, to name a few. The following screenshot shows the diversity of consumption by users today:



*Diversity of consumption*

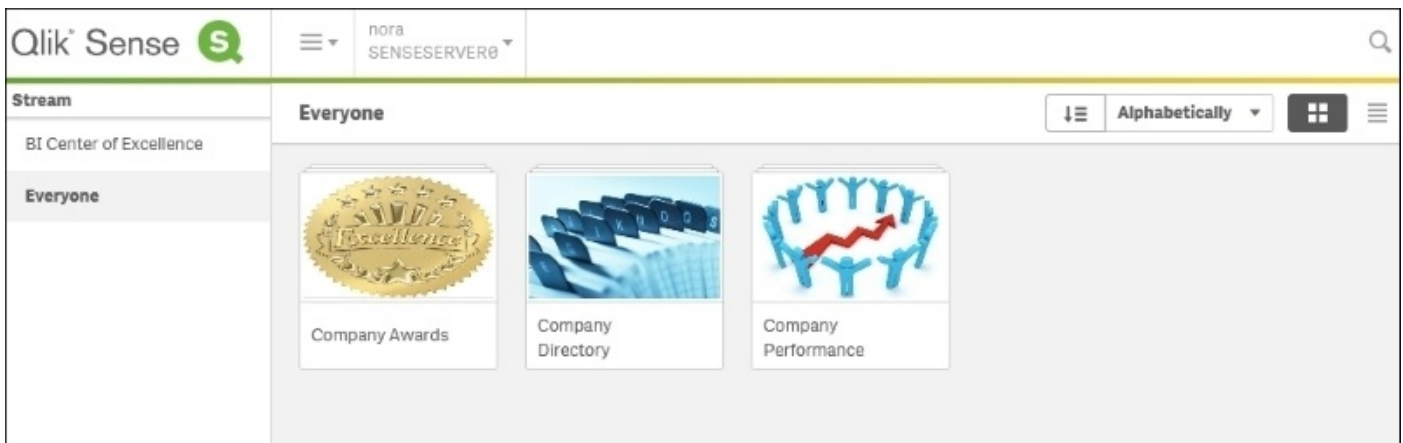
The key thing is that these Qlik Sense screenshots could have been taken using any device on the market. Critically, and uniquely, Qlik Sense uses **Responsive Web Design (RWD)**, along with progressive disclosure to provide an optimal data discovery experience for

users, whatever the form factor of the device. This is at the heart of the Qlik Sense architecture, the aim being to develop an app once and for it to be consumed/extended across any HTML5-compatible browser. For consistency and ease of illustration, the following key components of a Qlik Sense application will be illustrated from a laptop browser, but all these capabilities are available across tablets and smartphones as well. The following key Qlik Sense application components will be reviewed from a consumer perspective where the user has read-only access.

# Introducing the hub

As noted in the application life cycle in the previous chapter, Qlik Sense provides a rich collaborative environment that is governed by the QMC through streams. Let's begin our review with the hub, which is the center of a data discovery community. The hub is a collection of streams, which contain Qlik Sense applications. Through the QMC, an administrator defines the streams, and Qlik Sense inherits security access to these streams and applications through security rules. Security rules are covered later in [Chapter 9, Administering Qlik Sense®](#), and additional detailed examples are available in the Qlik Sense server user guide.

In this case, the consumer, let's call her Nora, has access to a default stream called **Everyone** as well as an administer-defined stream called **BI Center of Excellence**. The hub is designed for touch-friendly navigation (that is, it's designed to support selection and navigation using fingers!) between streams on the left-hand side of the display, searching and organizing the view in a number of sorted ways. Let's take a look at the hub:

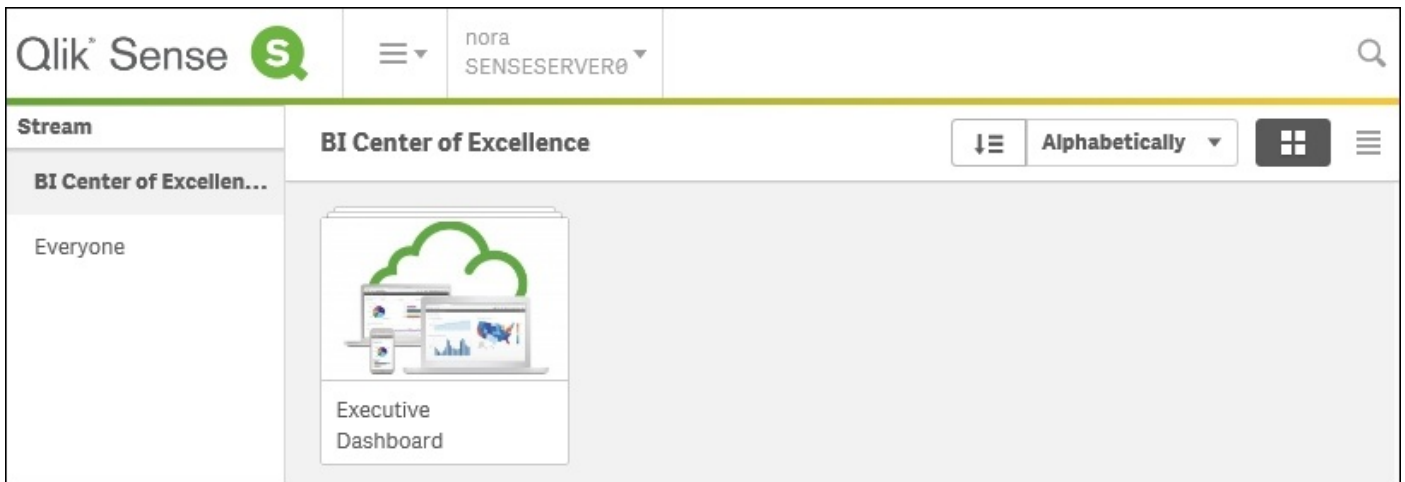


*The hub*

Now, let's turn our attention to streams.

# Introducing streams

Streams are an organizing principle for applications as well as security. Qlik thinks of streams as work streams for information that can be categorized based on maturity with gradual expansion of access by audience, subject matter, or any other organizing principle. Nora has access to two streams, the **Everyone** stream, which is a public stream created during the server installation, and the **BI Center of Excellence** stream. The **BI Center of Excellence** stream contains a single application called **Executive Dashboard**. **Executive Dashboard** will be used to illustrate how Qlik Sense provides insights to business decision-makers.



*The BI Center of Excellence stream*

Let's start with the components of a Qlik Sense application.

# Exploring the components of the application

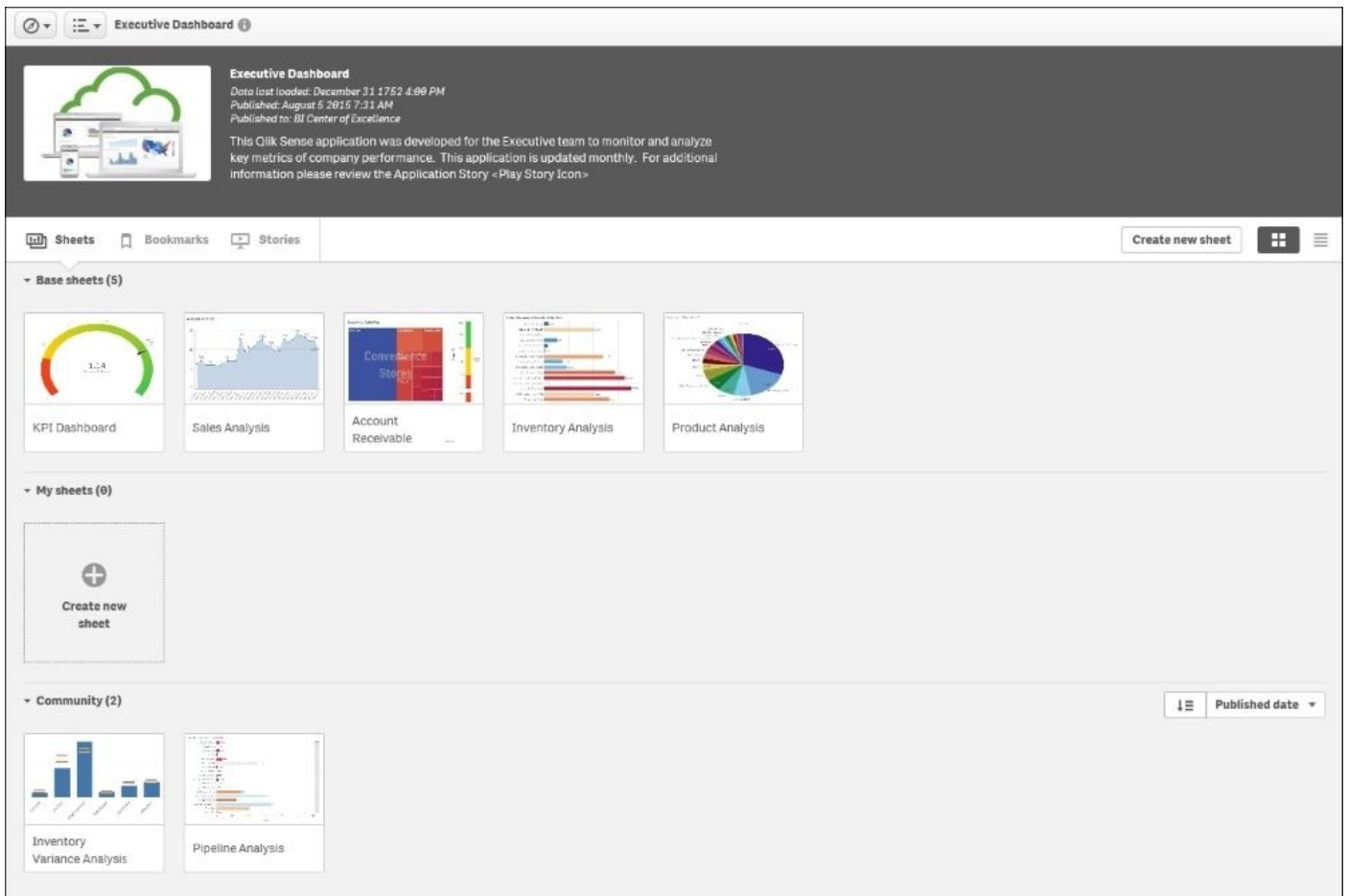
Qlik Sense applications are made up of three main components, which include **sheets**, **bookmarks**, and **stories**. In the case of Nora, who has consumer access, each of these components have been defined by the application author and are identified by the label **Approved**. This label identifies these items as part of the core components created by the author and cannot be modified once published.

## Sheets

Sheets are a core building block of Qlik Sense. Each sheet contains a collection of objects that are arranged to provide context for analysis on a particular subject. In this case, the sheets are contained in the application called **Executive Dashboard**. Note that sheets fall into the following three categories:

- **Base sheets:** These sheets are defined by the author of the application and become read only after publishing. They cannot be modified but can be duplicated as a private sheet for modification.
- **My sheets:** These sheets are similar to community sheets but are unpublished, so they can only be viewed by the author.
- **Community sheets:** These are private sheets that have been defined by a user and published to the hub. These can be defined based on duplicated approved sheets and/or new sheets that are assembled through the use of the application library. This will be discussed in detail in the next section, *Realties of data discovery power user*.

In the case of the **Executive Dashboard** application, there are five approved sheets that cover key application areas: **KPI Dashboard**, **Sales Analysis**, **Account Receivables Analysis**, **Inventory Analysis**, and **Product Analysis**. Each of these sheets provide a baseline for the consumer's analysis and exploration.



*The application overview*

Additionally, there are two community sheets, **Pipeline Analysis** and **Inventory Variance Analysis**, which were created by users who have *contributor* access rights. This is a power capability of Qlik Sense that allows users to share key findings across applications. Like base sheets and my sheets, approved sheets are stored with the Qlik Sense application.

## Bookmarks

Qlik Sense continues this popular feature, which was established in QlikView. Bookmarks allow a user to save the state of a sheet (their selections) so that they can be revisited at a future time, shared, and can be used to create data stories that allow users to combine key discoveries across many Qlik Sense sheets and add additional context through annotations. This example application contains four bookmarks as part of the published application.



The screenshot shows the 'Executive Dashboard' in Qlik Sense. At the top, there is a header with the title 'Executive Dashboard' and a refresh icon. Below the header, there is a large image of a laptop displaying a dashboard, with a green cloud icon above it. To the right of the image, the text reads: 'Executive Dashboard', 'Data last loaded: December 31 1752 4:00 PM', 'Published: August 5 2015 7:31 AM', and 'Published to: BI Center of Excellence'. Below this, a paragraph states: 'This Qlik Sense application was developed for the Executive team to monitor and analyze key metrics of company performance. This application is updated monthly. For additional information please review the Application Story <Play Story Icon>'. Below the main content area, there is a navigation bar with 'Sheets', 'Bookmarks', and 'Stories' tabs. The 'Bookmarks' tab is active, showing a list of 'Base bookmarks (4)'. The list contains four entries, each with a date and an information icon:

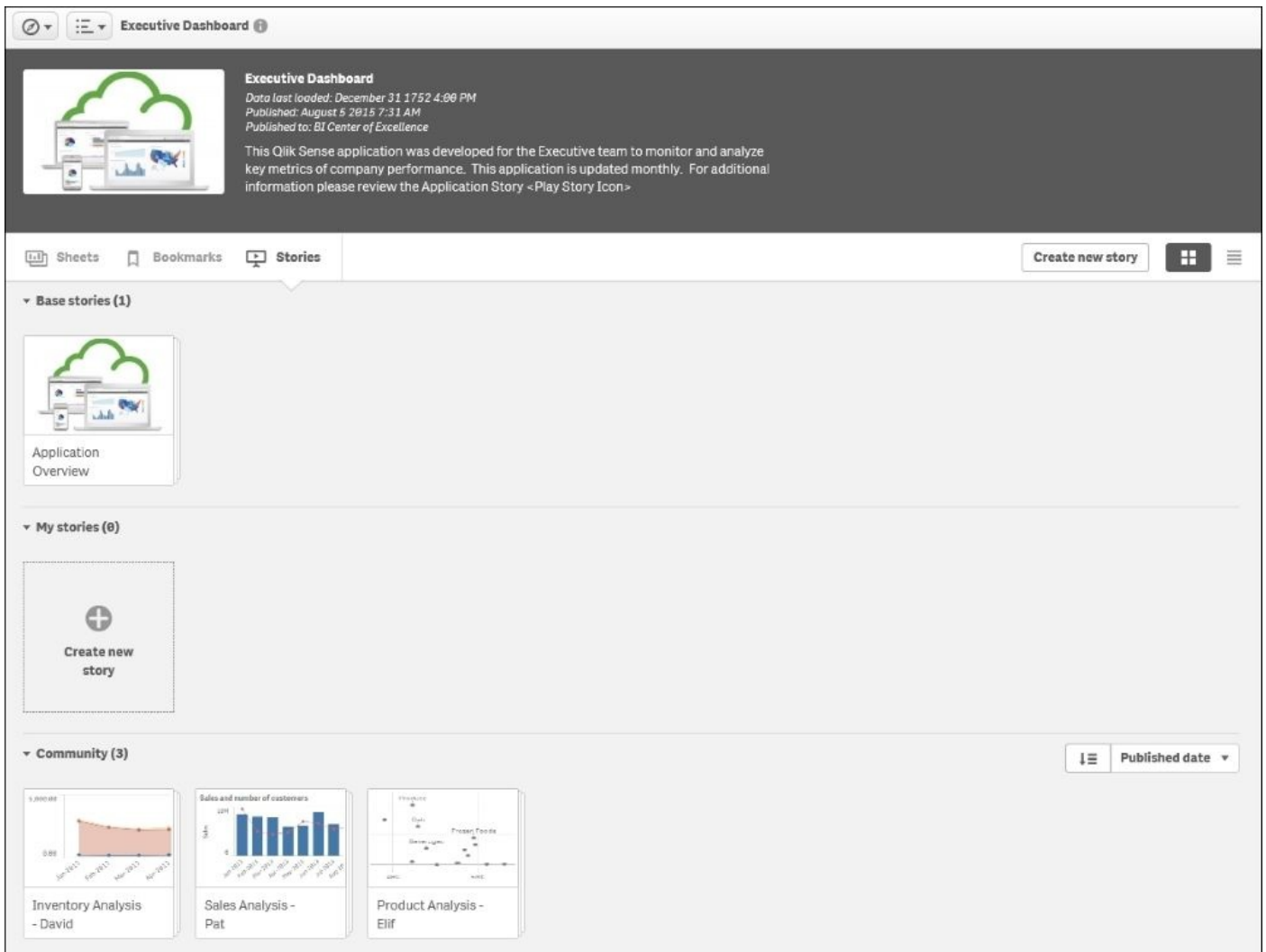
Bookmark Name	Date	Info Icon
KPI Dashboard - Product Group Desc ("Alcoholic Beverages")	09/12/2014	Info
Sales Analysis - Country ("Australia")	09/12/2014	Info
Account Receivable Analysis - SegmentDesc ("Convenience Stores")	09/12/2014	Info
Inventory Analysis - SegmentDesc (1), Product Group Desc (2)	09/12/2014	Info

### *Application bookmarks*

The **Approved bookmarks** section includes the KPI dashboard for alcoholic beverages, Australia's sales analysis, convenience store account receivables, and convenience stores' inventory analysis for deli and alcoholic beverages. Qlik Sense consumers can create bookmarks to save key discoveries to view at a later date. Once interesting information is found, a user may wish to combine visualizations and add annotations that highlight any key discoveries. This leads us to our next topic, *Data storytelling*.

### **Data storytelling**

Qlik Sense Stories are a collection of snapshots of key findings (visualization objects) that are assembled to share insights with others in an organization. Snapshots are a graphical representation of the state of visualizations at a certain point in time and are stored in the story media library. Although snapshots are static, they contain embedded bookmarks back in the source sheet, which enables users (or people who want to debate the detail of a narrative) to continue the exploration with live data from the point at which the snapshot was taken. Like sheets and bookmarks, base stories (published with the application by the author) and community stories (published by users who have contributor rights) can be seen.

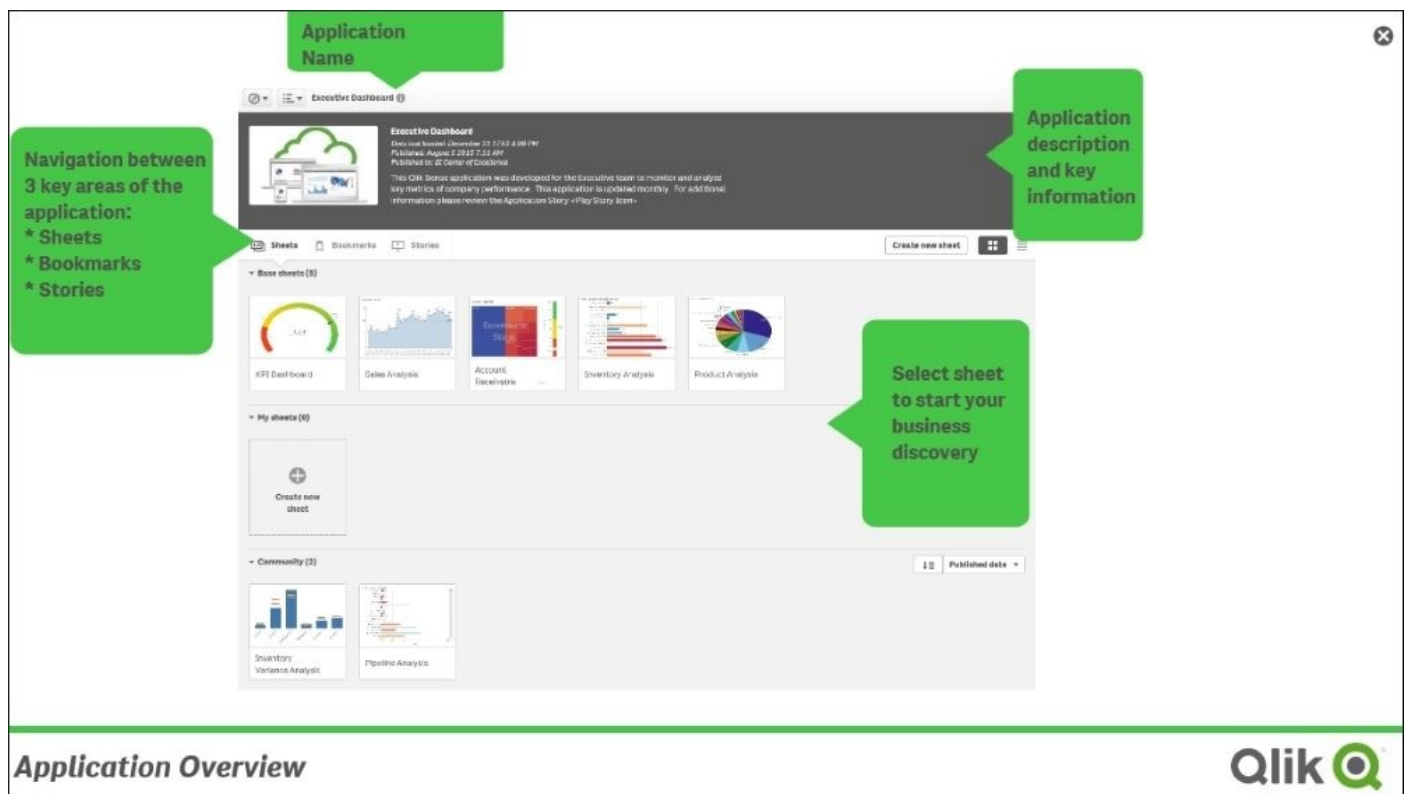


### *Application stories*

The **Executive Dashboard** application, shown in the preceding screenshot, contains four stories available for Nora to review. Community stories were published by Elif, David, and Pat highlighting product analysis, inventory analysis, and sales analysis, respectively. Additionally, there is an approved story named **Application Overview**, which was published as part of the application by the author to outline the goals and use of the application. It is a recommended best practice for application authors to include a story to spur the adoption of an application within the user community. This topic leads us to our next topic, *Navigating and leveraging the associative experience*, in which we will use the **Application Overview** story to provide an overview of the application.

# Navigating and leveraging the associative experience

As mentioned earlier, Qlik's intent in building Qlik Sense was to create a user experience that provides a natural and intuitive way to explore data and share key findings. To facilitate our discussion, we will refer to the **Application Overview** story. When selecting an application from the hub, Nora is provided with an application overview. This displays the application name, a short description, and a published date and time that provides key context for the timelines of the information.

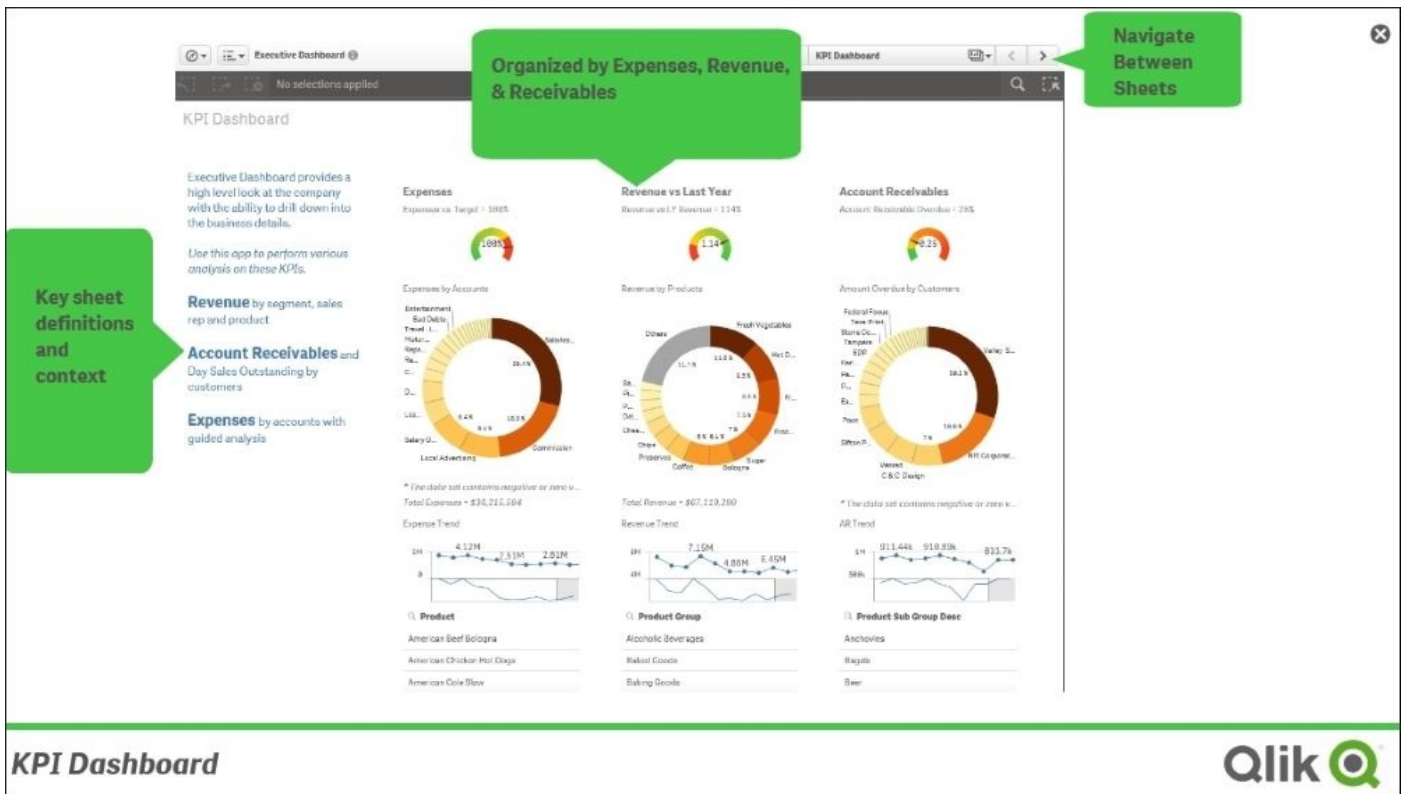


*The Executive Dashboard overview*

## Navigation

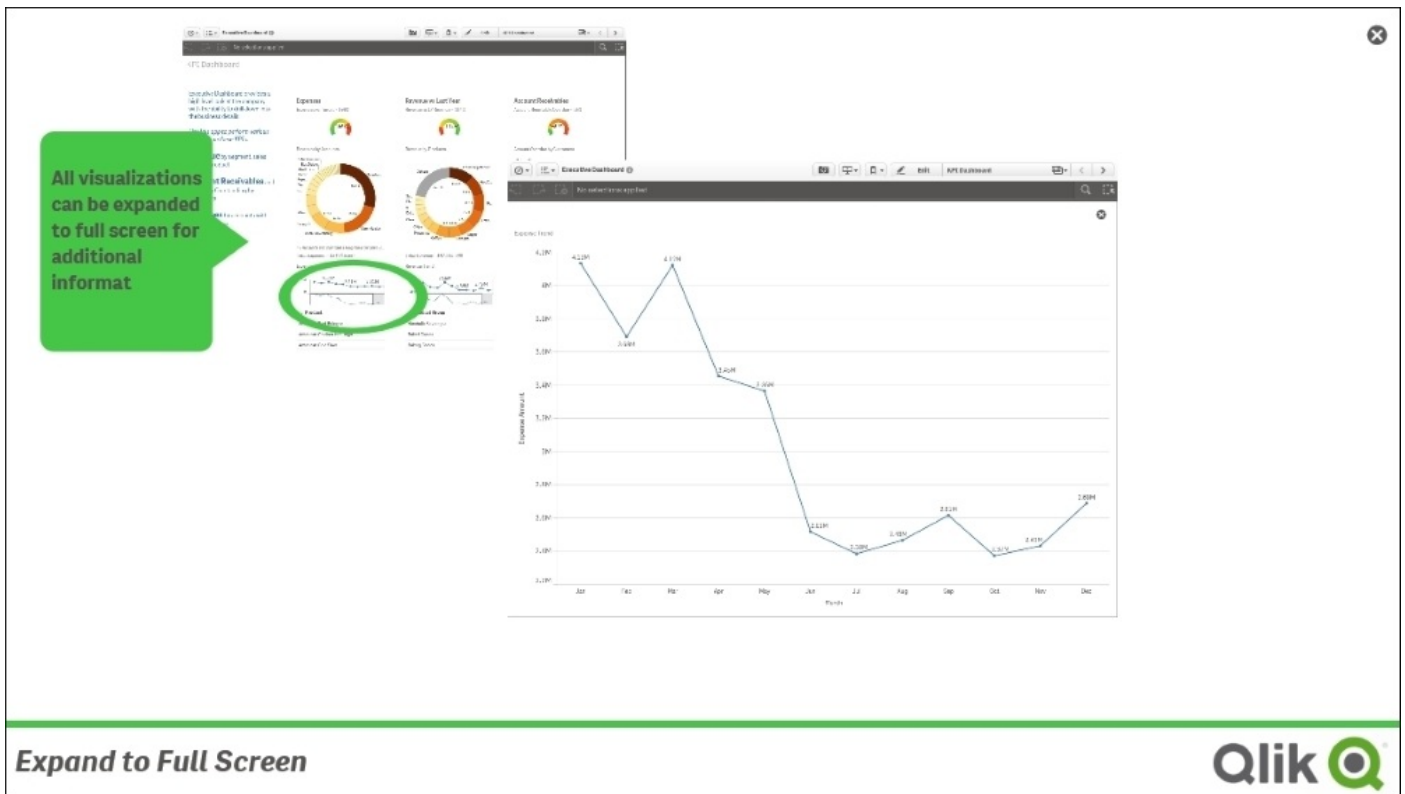
Additionally, there are three key areas to explore in a Qlik Sense application; they include sheets (highlighted), bookmarks, and finally, stories, which were discussed earlier. This application contains both approved sheets (developed by the application's author) and community sheets that are the results of contributors who have published private sheets they wish to share with the community. This process will be discussed in detail in the next section.

Now, let's open the first sheet named **KPI Dashboard**. As discussed earlier, sheets are an amalgamation of smart objects that display information based on the amount of space available. In **KPI Dashboard**, we can see that the sheet is divided into three key areas: **Expenses, Revenue vs Last Year, and Accounts Receivables:**



## KPI Dashboard

Each of these objects can be used as a filter to see data association and just as importantly, to see nonassociated data (informally known as “The Power of Gray” based on its default coloring that users of QlikView have enjoyed for years). Additionally, each of these objects can be expanded to fullscreen, as shown in the next screenshot. The expense sparkline chart can be expanded to fullscreen to reveal additional data points and trends. This also facilitates viewing and selections on mobile devices, where screen real estate is limited.



*Expand to Full Screen*

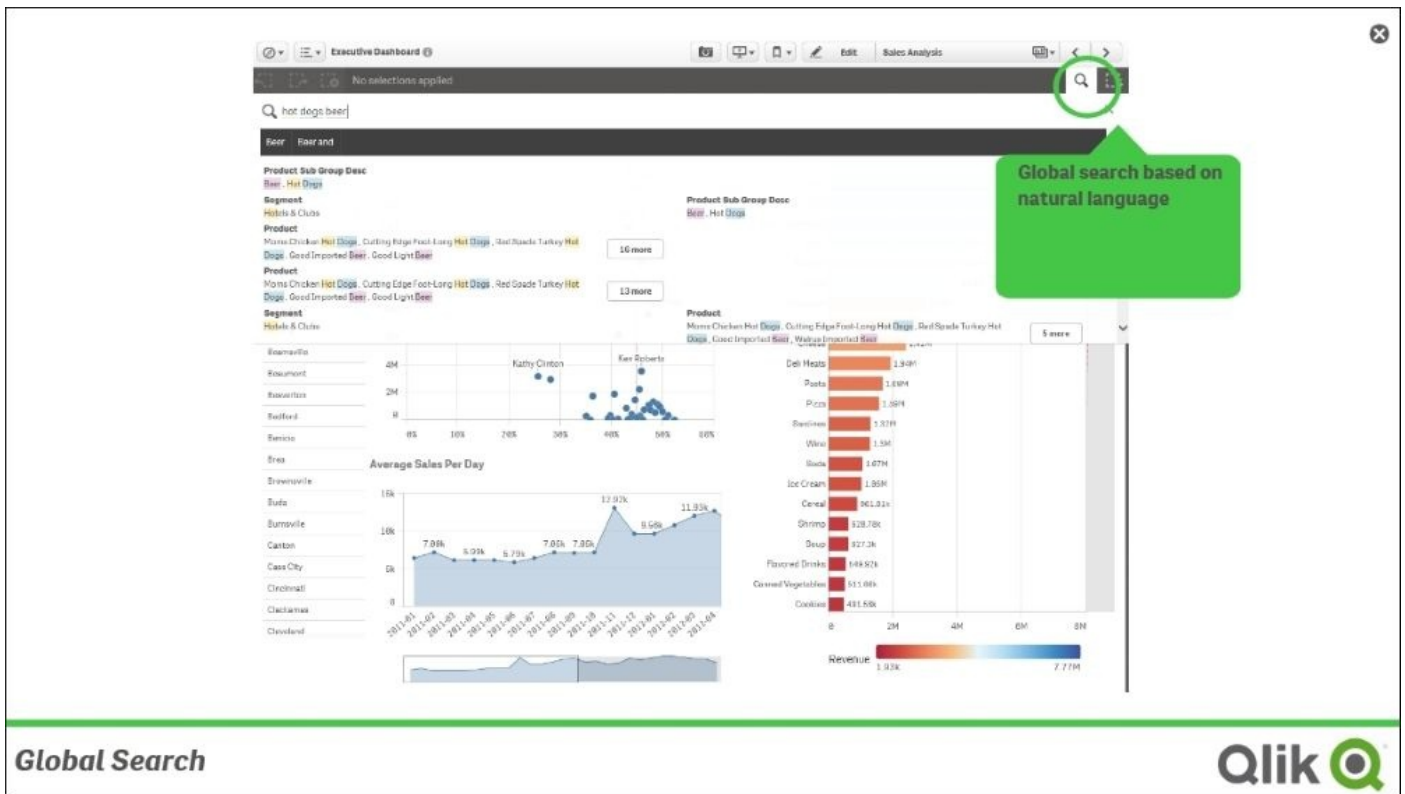
## Smart visualizations

As we review the **Sales Analysis** sheet, there are a number of innovative features that highlight the capabilities of Qlik Sense. First, let's review the sales margin versus sales revenue scatter chart. What makes this chart smart is how Nora interacts with it.

As mentioned earlier, Qlik Sense was developed for mobile devices, which implies touch interaction. In this case, the scatter chart supports multitouch selections on both the axes. In this example, Nora has selected to highlight the performance of sales representatives who have margins between 41 to 48 percent and sales between 3.69 million to 7.01 million. Additionally, these selections are in preview mode, which allows Nora to see the impact of these selections before confirming and moving on to the next phase of her discovery.

A second area to highlight is the use of smart scrolling noted in both the **Average Sales Per Day** area and the **Total Revenue by Product Group** horizontal chart. The scroll bars use thumbnails of the chart so that Nora can easily navigate to the key area for review. Additionally, scroll bars appear after the chart has reduced its size to a point where the entire dataset can no longer be shown in the allocated space within the sheet. This allows Nora to enter the numbers in a range selection within any chart, for example, the scatter—you can type in the exact number for the range filter. Also, you can move the filter range keeping the range as you scroll along the x or y axis.





Global Search



## Global Search

### Global filtering

To accompany global search, a fully structured approach to filtering is available on every sheet in the top right-hand corner called the global filter. In the global filter, we can see current selections in the top half of the sheet highlighted in green. The bottom half of the sheet is reserved for dimensions that have not been included in the filtering. Note the associated colors of green for selected elements, white for none selected, and gray for nonassociated elements. Light gray indicates excluded only by selection in the same field, whereas dark gray means excluded by selection in other fields. We can see that the current selections of **ARAge** as **31-60 Days**, **Customers** as **A&R Partners**, and **A2Z Partners** and **AccountDesc** as **Communications** are selected and highlighted in green. If we look at the **Customer** dimension, we see that all other customer names are dark gray because a customer can only have one name in this model. We also see that the other **ARAge** and **AccountDesc** dimension elements are light gray because they are excluded based on selections in other fields. This could change with a change in the selection criteria. Based on this example, global filtering provides a very powerful view of the relationships in the application's associative data model. It also centralizes filtering, leaving valuable screen real estate for visualization based on filtering and the exploration of information, and once selected, it appears in the **SELECTIONS** pane.

**Associated Color Key:**  
 \* Green: Associated  
 \* White: Not Associate for selection  
 \* Grey: No association

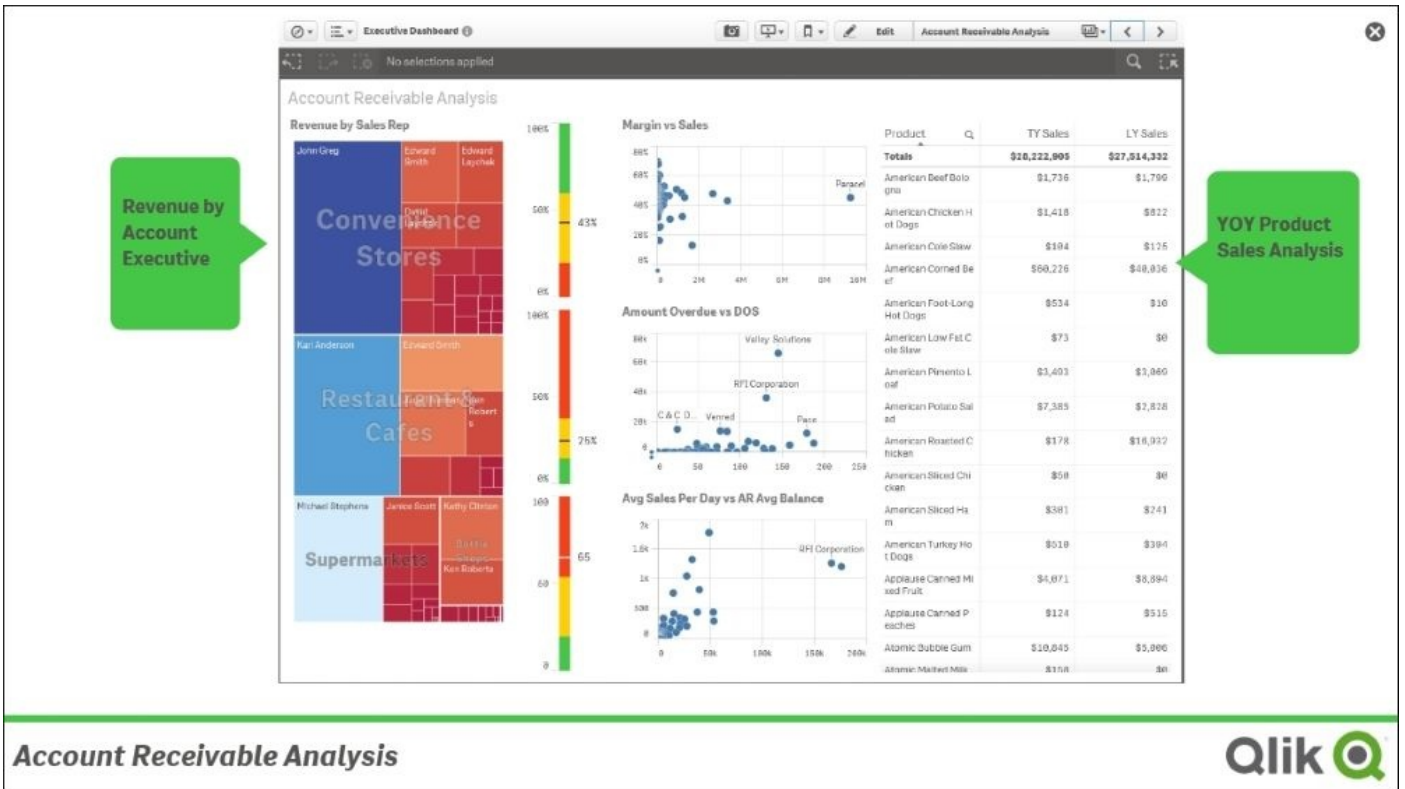
**Global filtering with associative relationships**

**Global Filtering** **Qlik**

## *Global Filtering*

Now, let's turn our attention to the **Account Receivables Analysis** sheet. This sheet is an interesting example of where there are no formal filter panes or listboxes (as there would commonly be in a QlikView app). Instead, each of the objects can be used to select areas to explore, and global filtering and global search can be used to augment or refine the selections at a finer detail level. In this case, revenue contribution for sales representatives by channel is displayed. Qlik Sense also supports a full range of objects, such as the table object to the right, which can be used to filter columns and supports exception formatting for variance reporting.



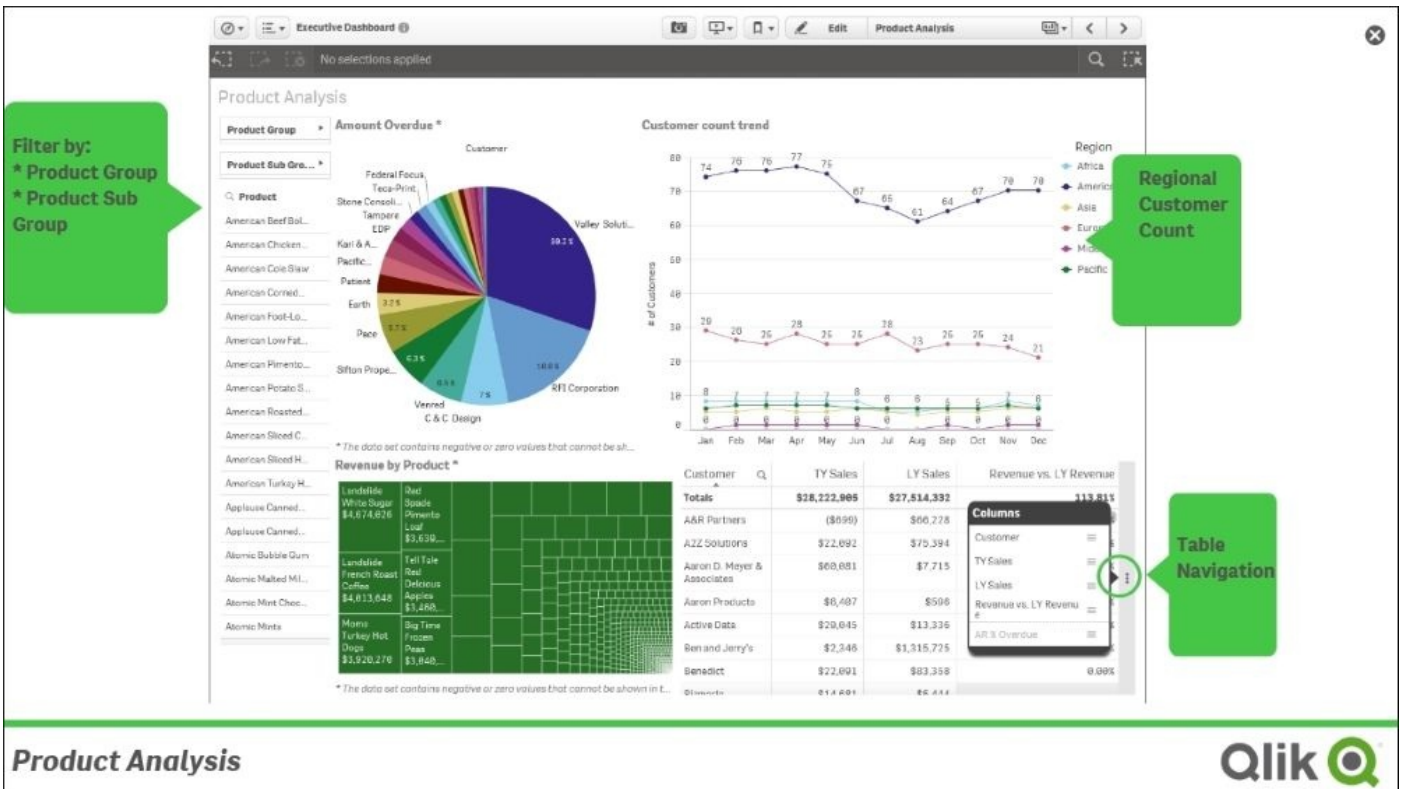


**Account Receivable Analysis**



*The Account Receivable Analysis sheet*

Finally, the table object has the ability to hide and show columns based on the allocated space for the table. The column selection menu within the table allows Nora to orient columns based on the viewable space available to the table.



**Product Analysis**

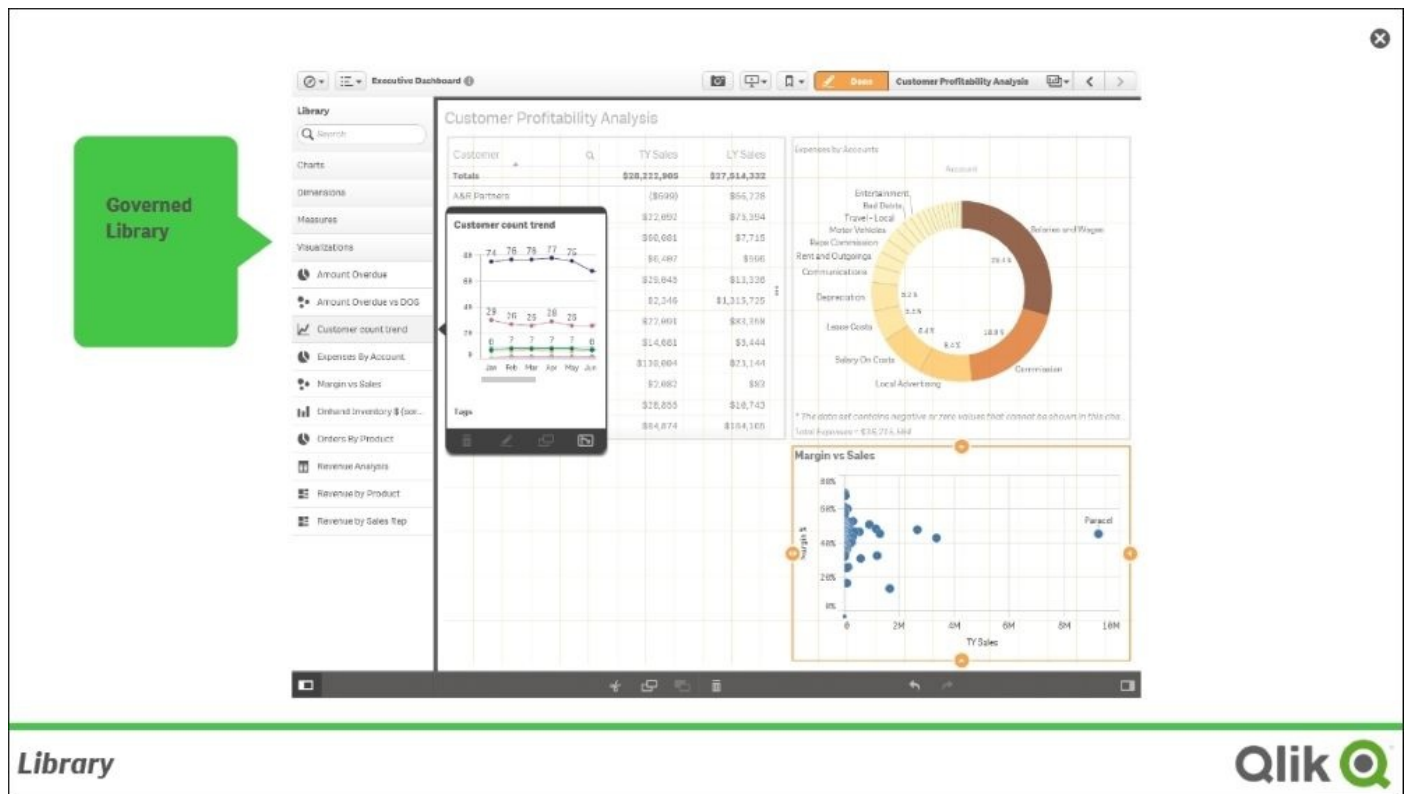


*Product Analysis*



# Extending with Library

As noted in [Chapter 1](#), *Qlik Sense® and Data Discovery*, the rise of BI consumerism and self-service is becoming an increasingly important attribute to meet the needs of the next generation consumers. Qlik Sense embraces this important requirement through **Library**. The Qlik Sense Library is a governed area where an application's author can store dimensions, measures, and preconfigured charts that can be used to create compelling analysis that can be shared across an organization. In this case, Nora is taking advantage of the **Customer count trend** line chart to extend an application.



## Qlik Sense Library

The Qlik Sense Library is at the center of a broad range of governed self-service capabilities that drives insight within an organization.



# Summary

This chapter covered how Qlik Sense meets the new requirements of consuming and extending discovery-based applications, meeting these requirements across a myriad of platforms spanning PC, Mac, and the never-ending flow of new mobile devices. This required Qlik Sense to be built with a new approach that is responsive to these new realities of self-service and mobile use.

Now, let's turn our attention to the contributor who seeks to not only consume but also extend and share their data discovery insights.



# Chapter 4. Contributing to Data Discovery

In the previous chapter, we outlined data discovery consumption requirements, which provided an overview of key Qlik Sense capabilities for users who wish to consume an application that is prebuilt. This chapter's goal is to highlight key features in the context of the specific user requirements that Qlik has identified as being needed by a data discovery contributor, or someone who seeks to share key findings from their analysis in a governed manner.

In this chapter, we will cover the following topics of Qlik Sense:

- Data discovery contributor requirements
- Bookmarks
- Private sheets
- Private stories
- Publishing to an existing application

# Realities of the data discovery contributor

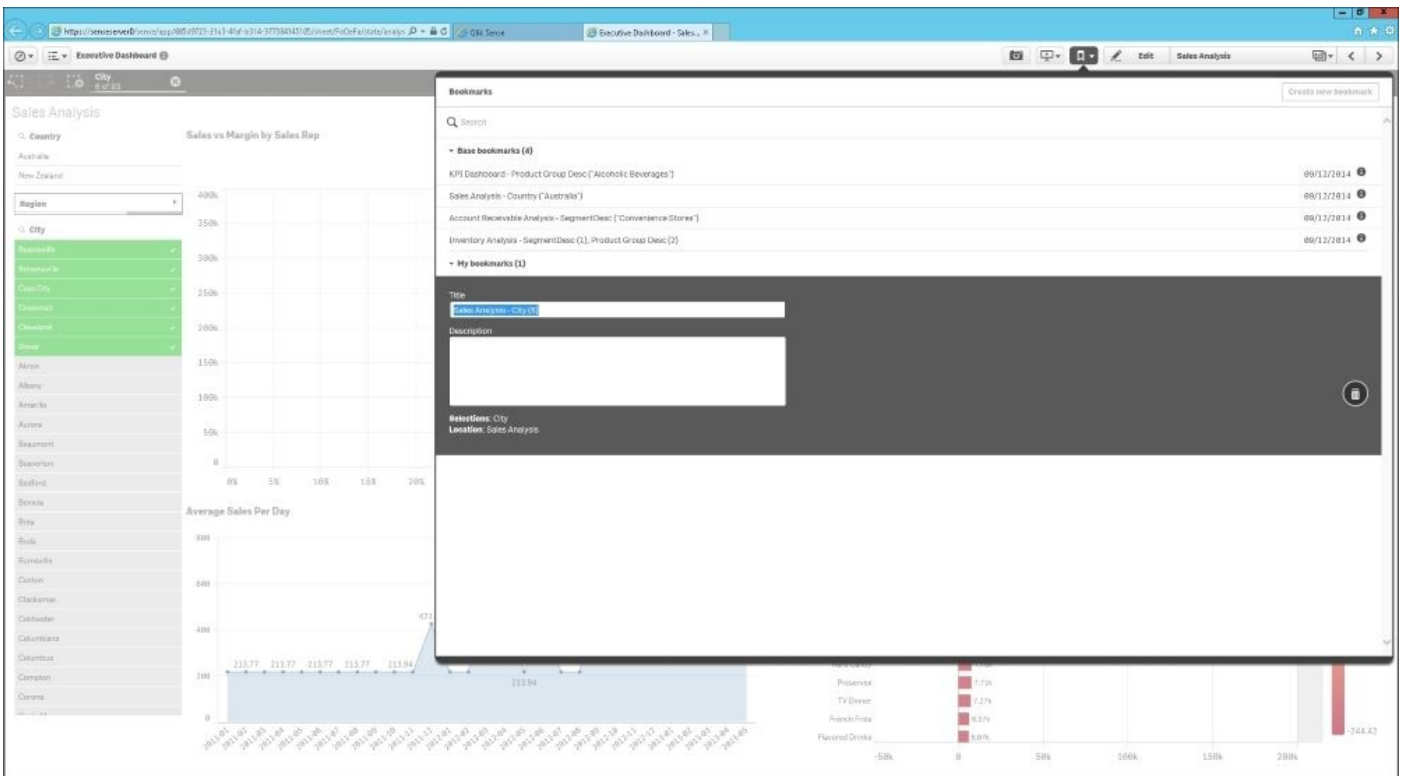
One of the strengths of Qlik Sense applications is the ability to share and extend the value of applications with other members of the stream. As noted in [Chapter 2, Overview of a Qlik Sense® Application's Life Cycle](#), there are a number of useful ways to share key business discoveries. These include the following:

- Bookmarks
- Private sheets
- Stories

Each of these capabilities helps analysts not only to consume Qlik Sense applications but also to share and spur additional conversation and insights. The stream administrator covered in [Chapter 9, Administering Qlik Sense®](#), enables these contributor capabilities. Let's take a closer look at each of these capabilities through the role of an analyst named Pat.

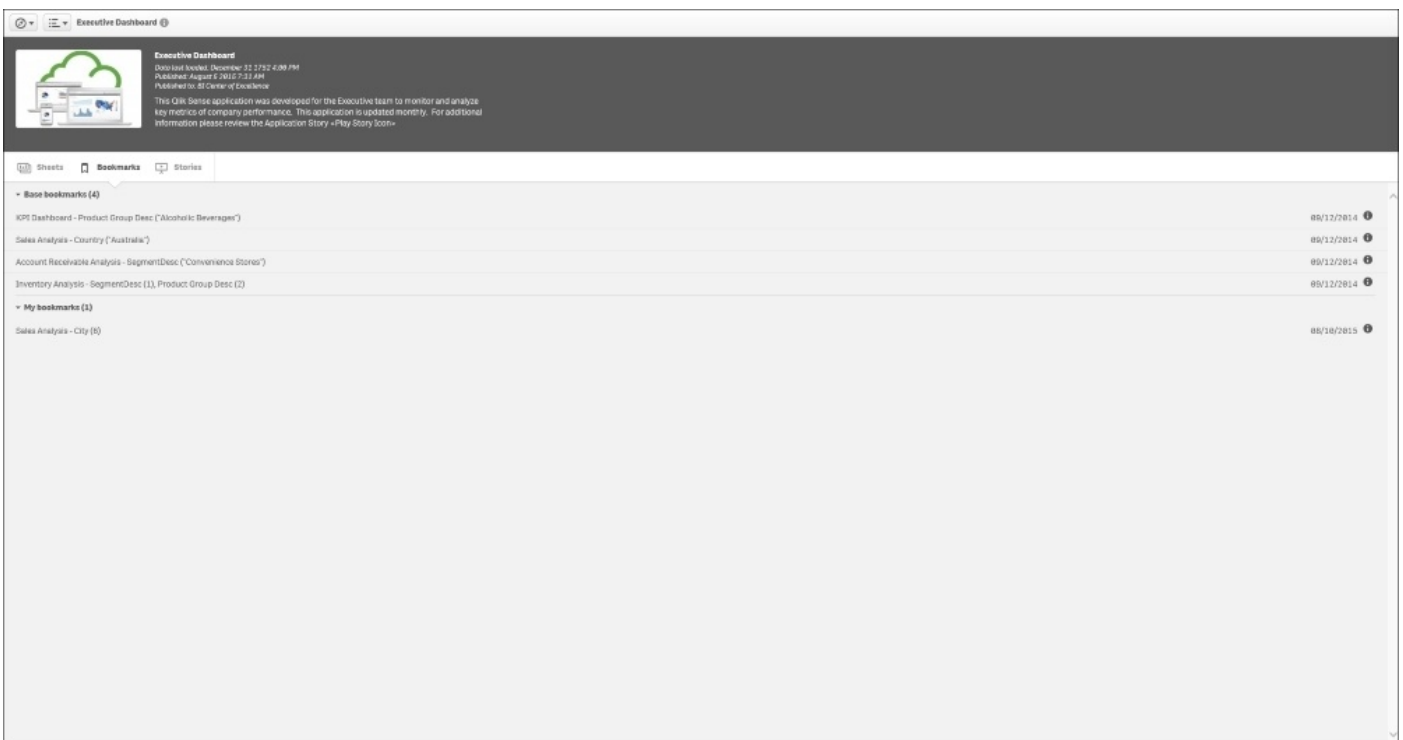






*City sales analysis bookmark*

Once saved, the bookmark becomes a part of the application under **My bookmarks** and can only be accessed by the creator, which in this case is Pat:



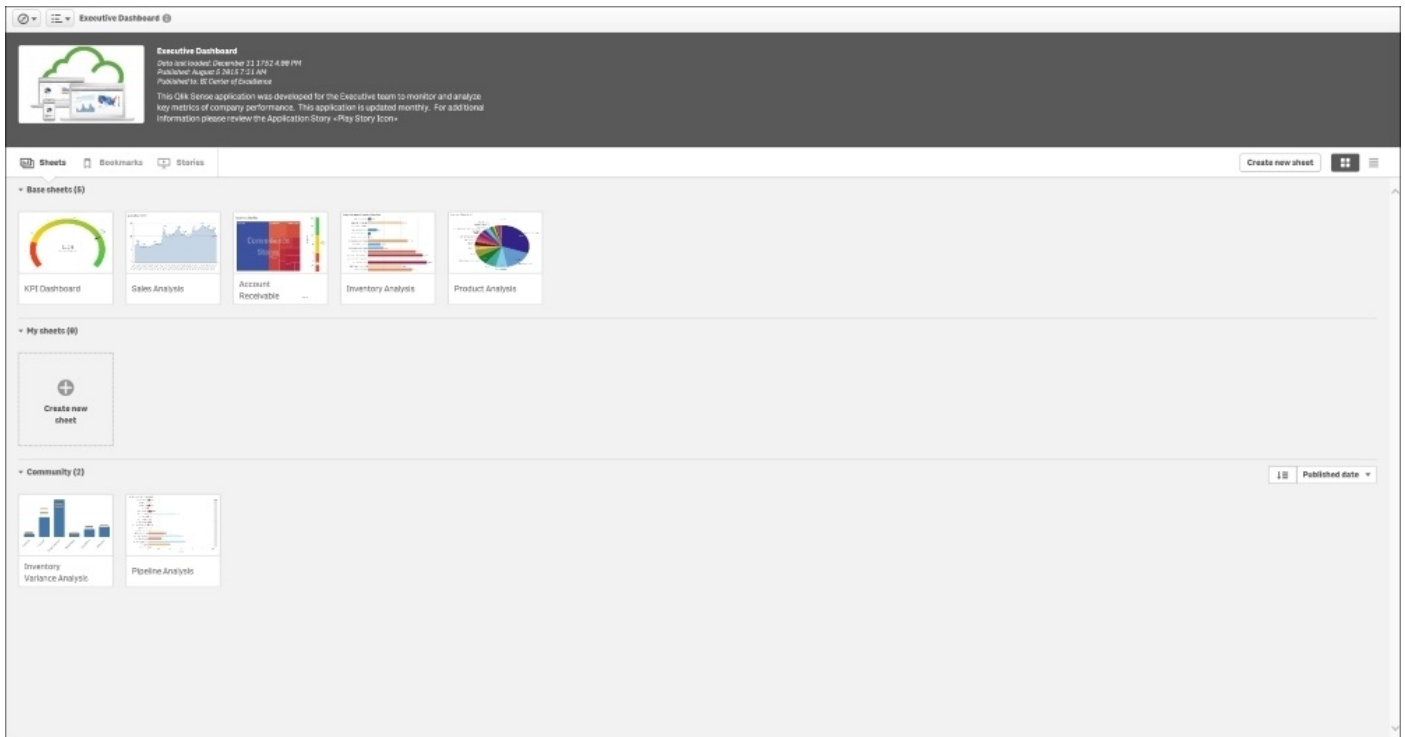
*The Executive Dashboard bookmark*

To summarize, simple Qlik Sense bookmarks can play an important part in bringing

context to the beginning of an analysis as well as saving key insights gained from an analysis. Although separate features, Qlik is seeing early adopters use bookmarks as the start of building critical mass with insight that can be shared through published sheets and stories, which are the topics of our next sections.

# Creating and sharing private sheets

As discussed in [Chapter 2, Overview of a Qlik Sense® Application's Life Cycle](#), the building block of a Qlik Sense application is a sheet. In the **Executive Dashboard** community, we can see the sheets associated with the **Executive Dashboard** application. These include **Approved sheets** (published by the application author), **My sheets**, which are private sheets defined by the contributor (Pat), and finally, **Community**, which are private sheets published by other contributors:

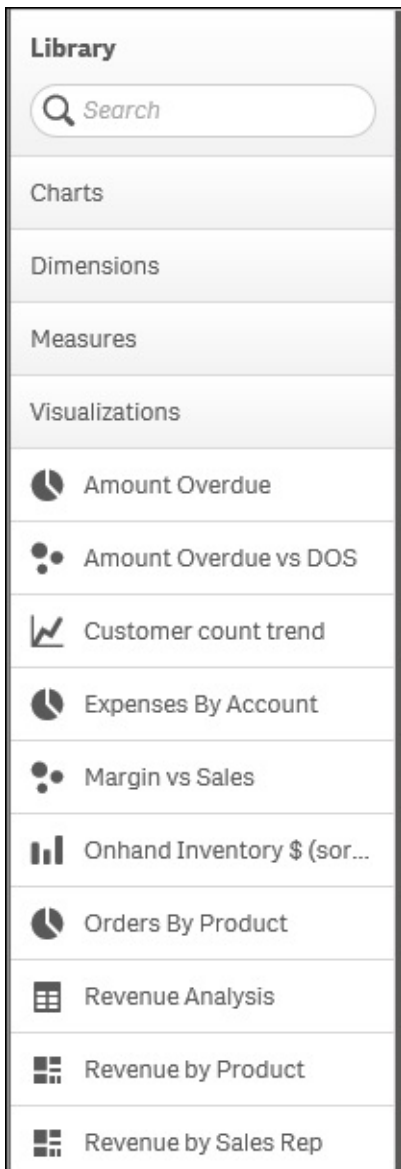


*The Executive Dashboard community*

Now, let's dig a bit deeper into how these sheets are built. There are two main ways in which private sheets are built, as follows:

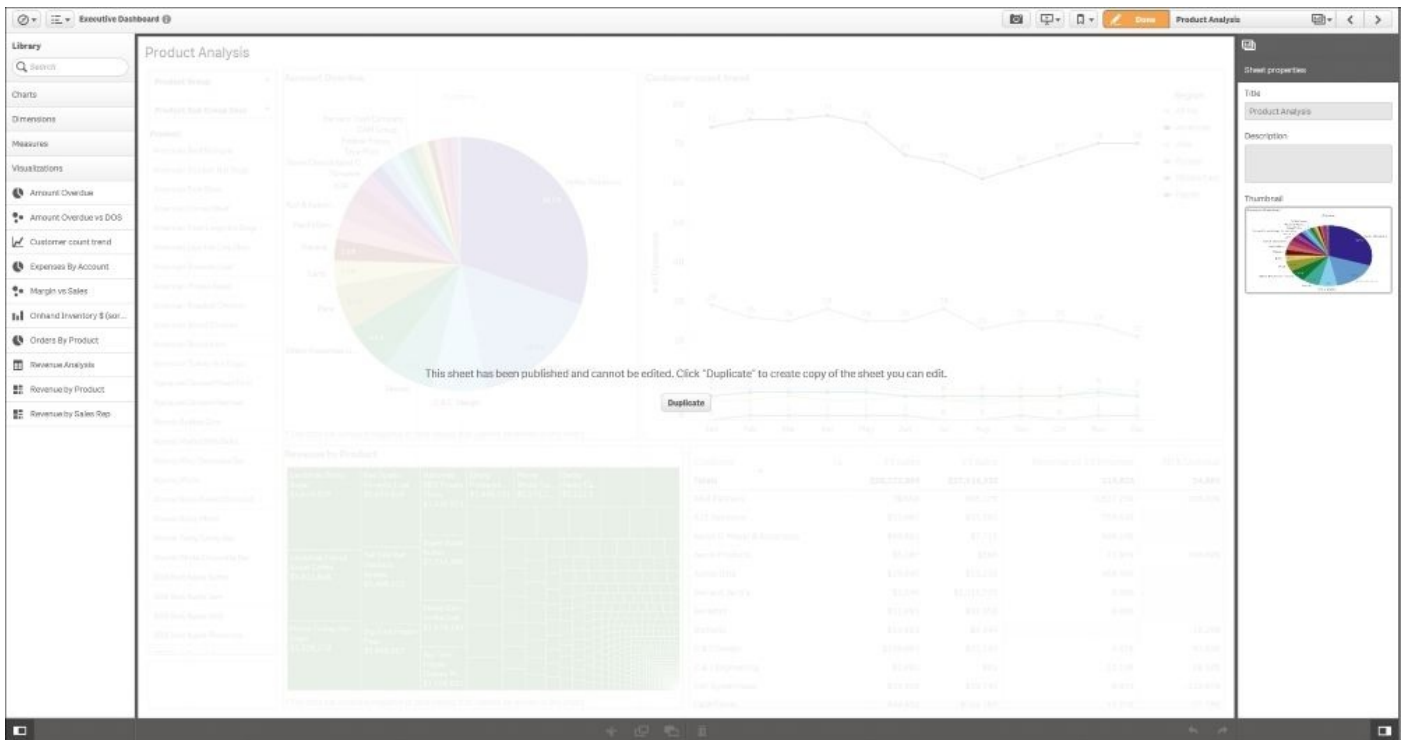
- Duplicate an approved sheet
- Create a new private sheet

In both cases, a key feature that allows a contributor to build strongly governed private sheets is the Qlik Sense Library. The Qlik Sense Library is a key component of an application that allows the author to expose key portions of the associative model in the form of **Dimensions**, **Measures**, **Charts**, and predefined **Visualizations**. How the Qlik Sense Library is created will be covered in more detail in the next chapter. The following screenshot shows **Library**, which can be searched for dimensions, measures, and prebuild visualizations:



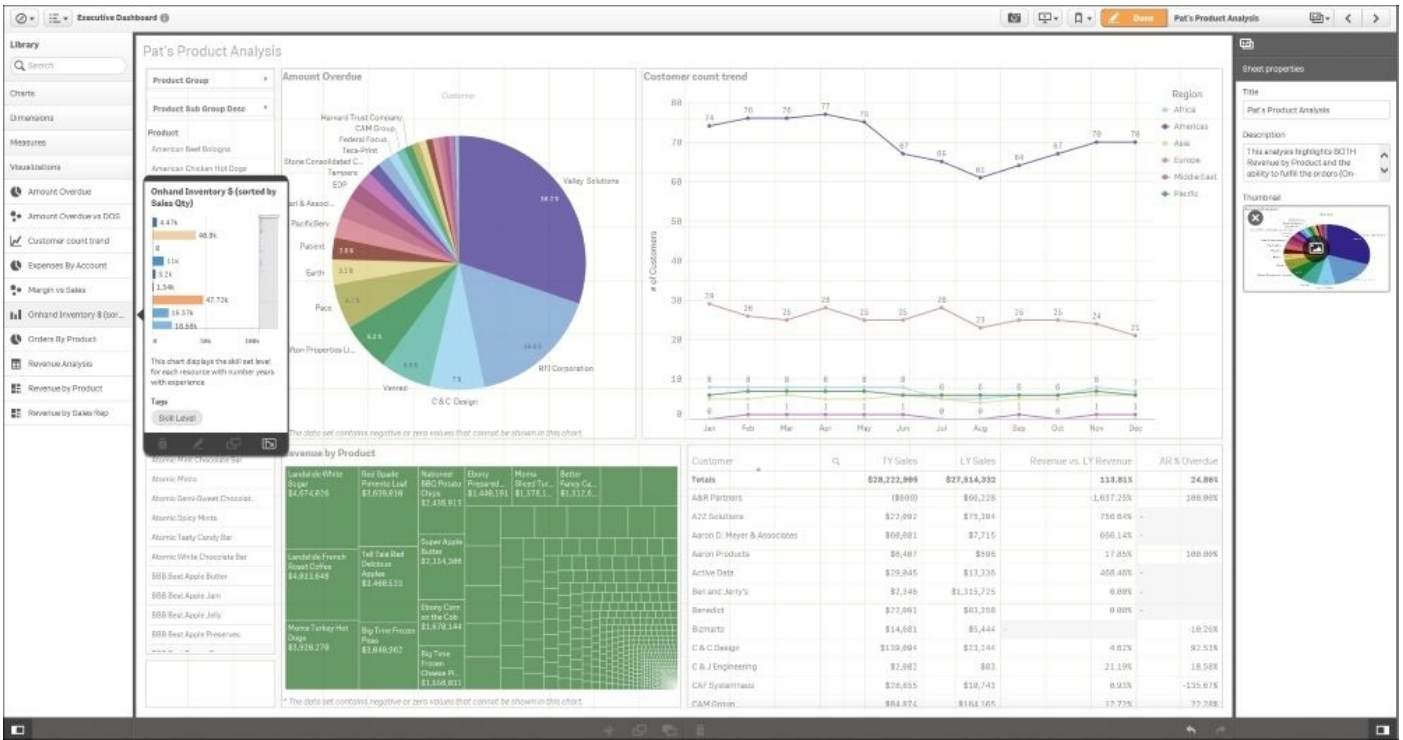
## Creating a private sheet

Now, let's turn our attention to creating a private sheet by the first method, duplicating an existing sheet, and then editing it to meet your requirements. The advantage of this method is that Pat can start the creation of her product analysis based on the approved **Product Analysis** sheet. The process begins with selecting the sheet that best aligns with the content you wish to analyze. In this case, Pat wishes to create a product analysis that integrates the inventory on hand with the approved **Product Analysis** sheet. As the **Product Analysis** sheet is an approved sheet, it cannot be edited and must first be duplicated before changes can be made.



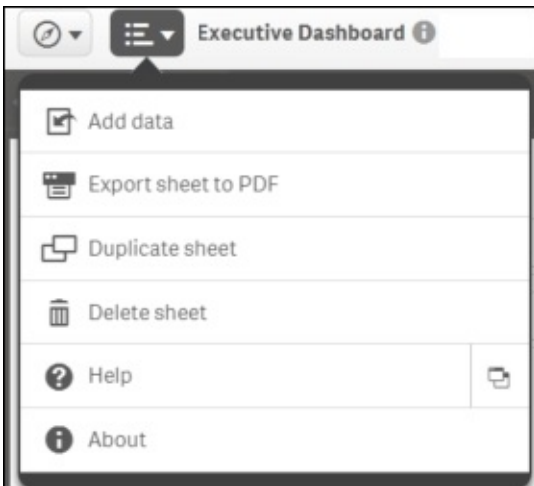
### Duplicating the Product Analysis sheet

Once the sheet is duplicated, it is automatically converted into a private sheet, where Pat can rename and alter the content and layout of the sheet through the use of **Library**. Note that Pat has renamed the sheet to Pat-Product Analysis as well as added a helpful description, which highlights the goals of this sheet—This analysis highlights both Revenue by Product and the ability to fulfill the orders (On hand Inventory\$) to recognize revenue. Additionally, there is a wide selection of preconfigured charts as well as dimensions and measures she can take advantage of in **Library**. In this example, Pat will replace the customer count line chart with the **Onhand Inventory \$ (sorted by Sales Qty)** horizontal bar chart from **Library**. This is one example of a variety of governed changes available to Pat in designing a new sheet. We will explore the breadth of changes to develop private sheets in the next section.

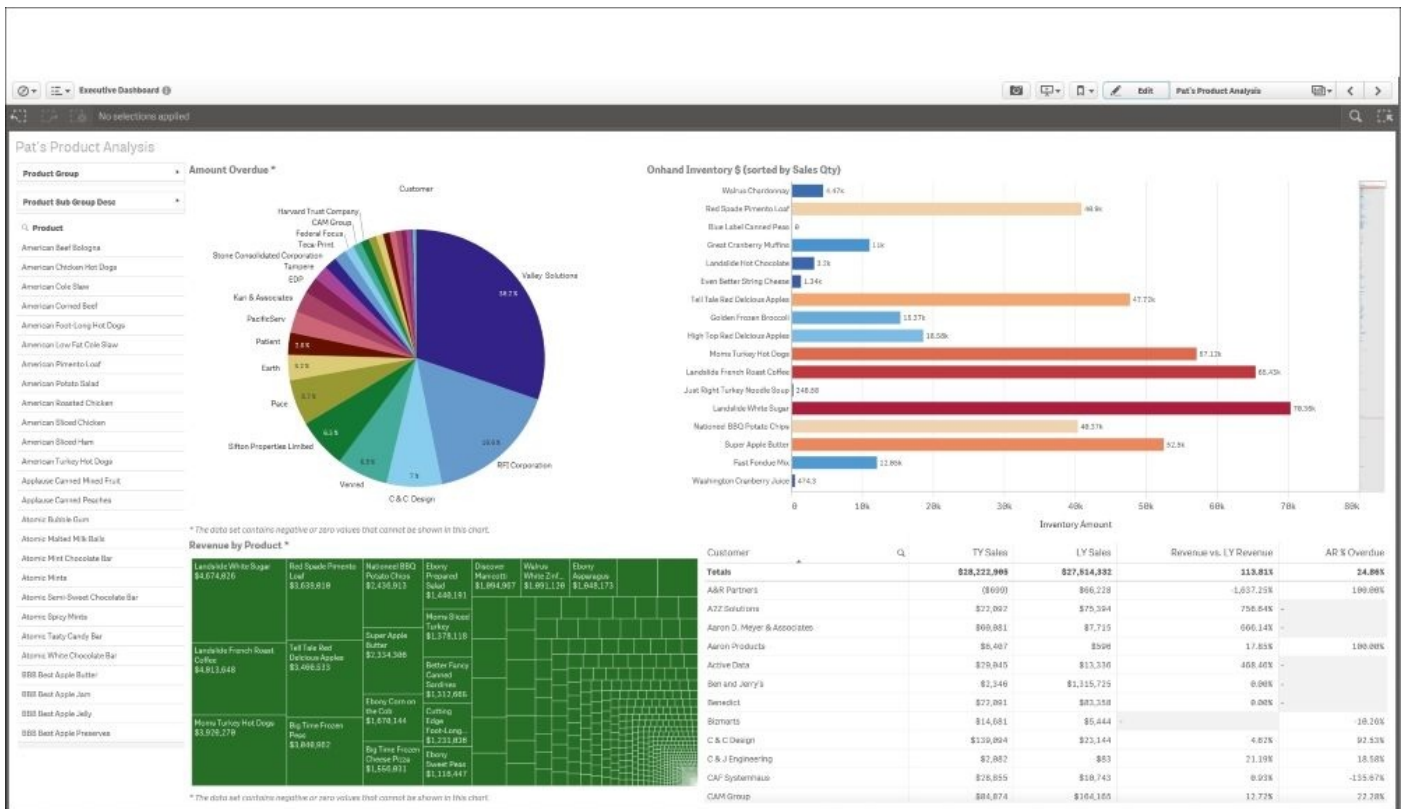


### Creating the Pat-Product Analysis sheet

With the **Onhand Inventory \$ (sorted by Sales Qty)** chart from **Library** dragged and dropped onto the sheet, Pat is ready to end the editing process. Since this process is all server-based, there is no need to save the sheet but rather just click on the **Done** button. Additionally, this sheet can be exported as a PDF and then either distributed via e-mail or printed:



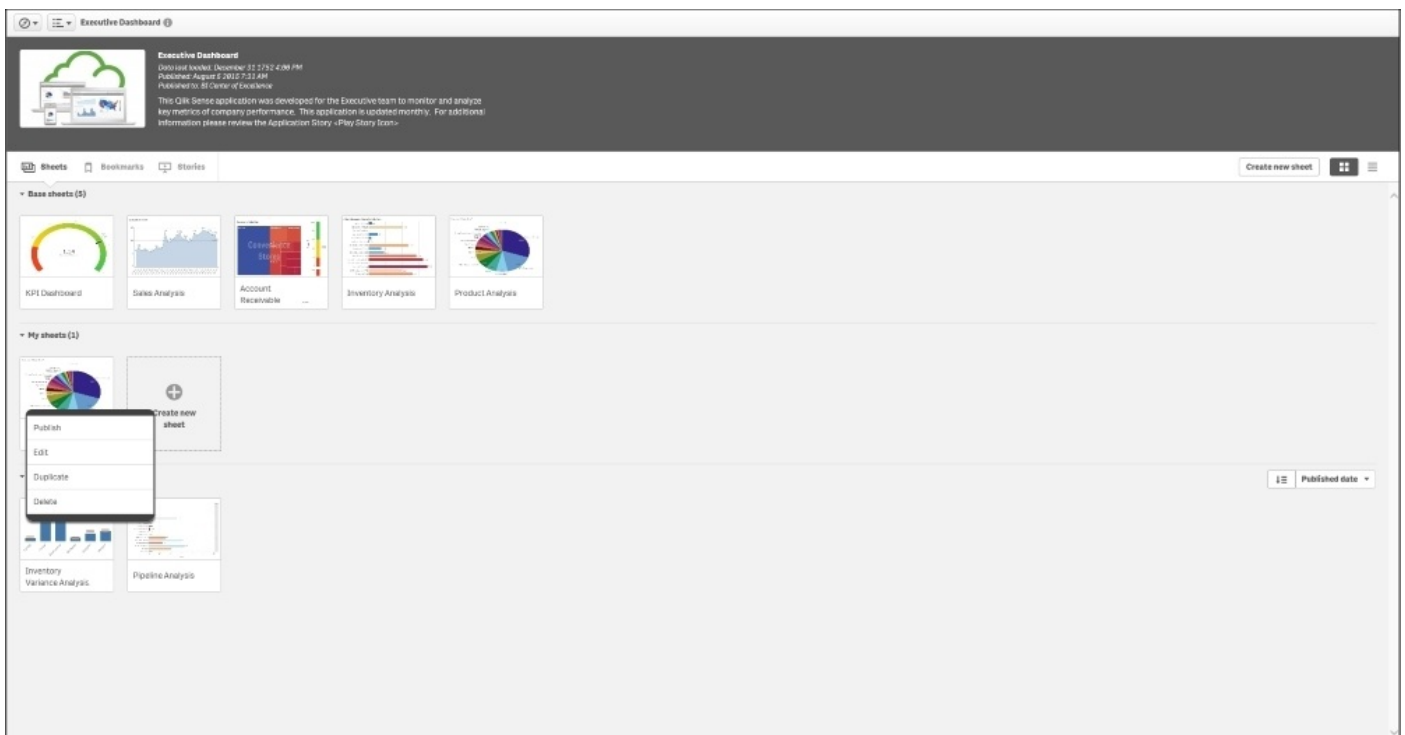
Export the sheet to PDF. This is how it will look:



The Pat-Product Analysis sheet

## Publishing a private sheet

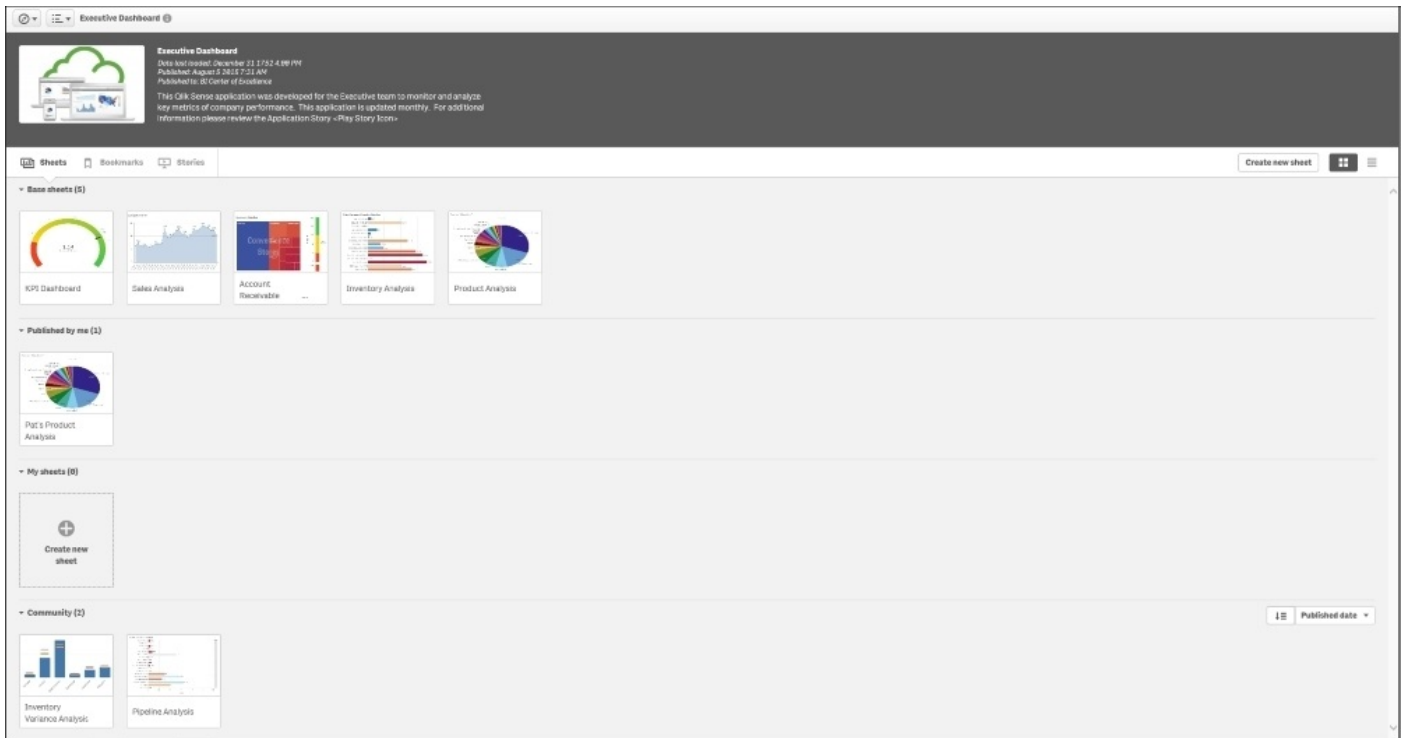
Now that the sheet is complete, let's return to the application overview. As you can see in the following screenshot, **My sheets** now contains a new sheet called **Pat-Product Analysis**, and with a right-click, it is ready to be published to the community:



Publishing the Pat-Product Analysis sheet



When the sheet is published, a new section will appear called **Published by me** that contains all published sheets by Pat. Also, note that Pat has a number of sheets that are in progress in the **My sheets** section. These published sheets can be duplicated by others who have access to this stream and are extended and shared as well.

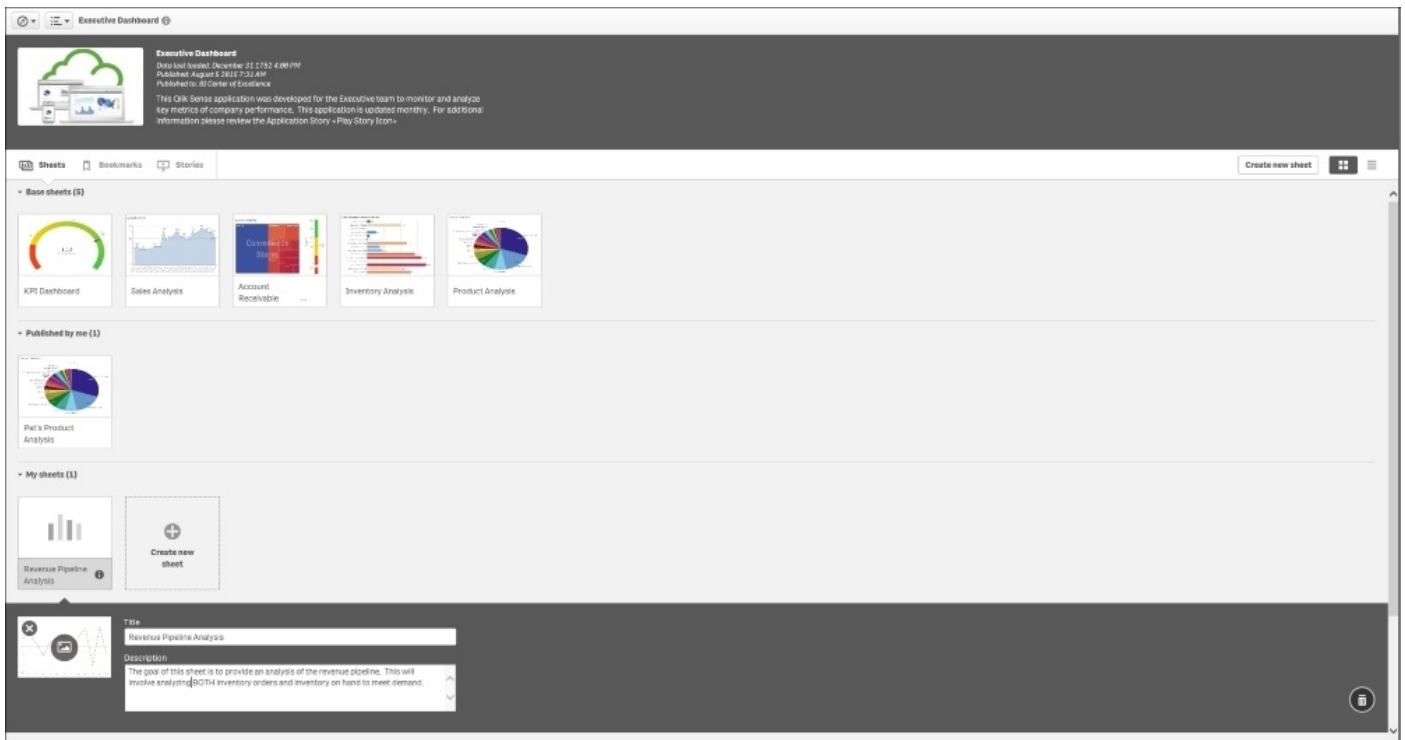


### *Pat's sheets*

In summary, a duplicated approved sheet is an excellent way to start the creation of a private sheet as it has the advantage of leveraging the existing defined sheets from the published application or the work of other analysts in the community. Now, let's turn our attention to creating a new sheet.

## **Creating a new sheet**

A second approach to sharing key business insights is to create a new sheet. As shown in **Creating a New Sheet** in the following screenshot, Pat creates a new sheet called Revenue Pipeline Analysis, which contains both order information and inventory on-hand information to meet customer demand. This allows Pat to create and share new information across the organization:

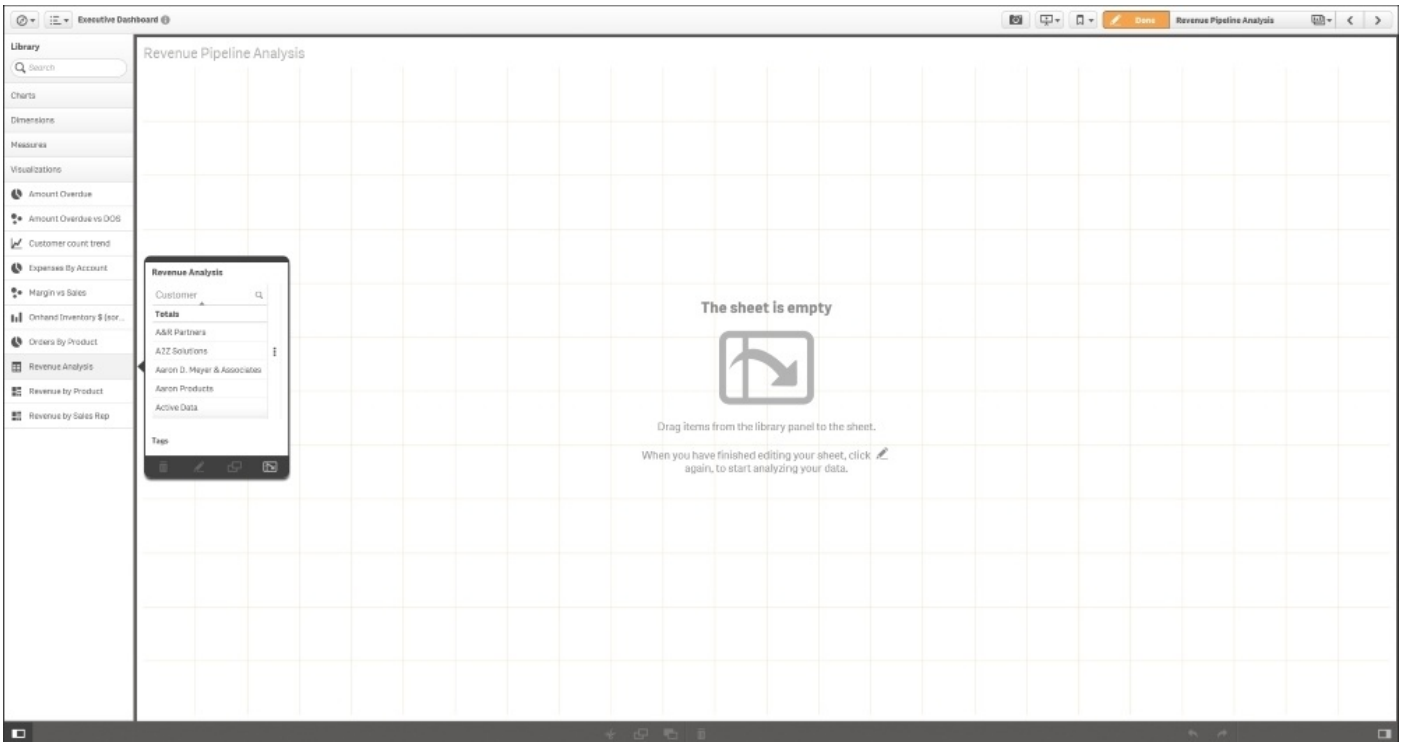


### *Creating a new sheet*

Once the sheet is created, **Library** and sheet properties are exposed and the sheet appears with a faint grid. This grid is a part of the responsive web design experience and facilitates the orientation and placement of objects from **Library**. This not only helps in the creation of the sheet but also plays a key role in how the objects will be viewed and consumed across multiple devices. Also, note that the creation and assembly of new objects is easily done by users due to the associative engine. Because of the associative model, every object is connected and no author prewiring is required. The associative engine permeates the use of Qlik Sense, not only its use, but also the creation of compelling solutions.

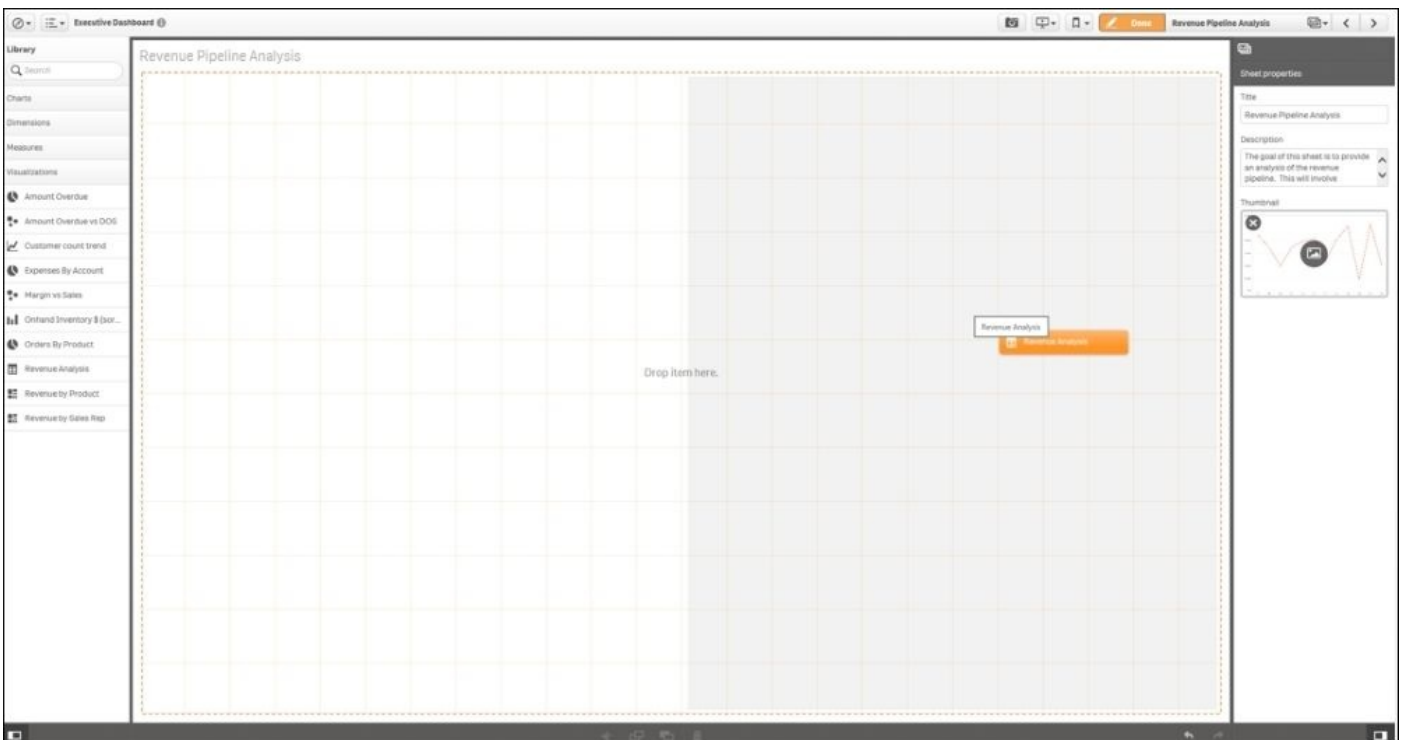
### **Adding a predefined visualization to a new sheet**

One of the key areas Pat is interested in is the customer revenue this year and in the previous year to help her better anticipate customer demand. Hopefully, the author of the application anticipated this common request and stored a table chart under **Visualizations** in **Library**. Specifically, the **Revenue Analysis** table chart is available in **Library** with a thumbnail shown to help Pat evaluate its applicability to the sheet content:



*Adding a predefined visualization to a new sheet*

Adding the **Revenue Analysis** object is a simple drag and drop movement. Note that the sheet grid will automatically make recommendations on the placement of the object:



*Drag and drop the object*

## Creating a Combo chart object

Once the object is placed, Pat notices that there are no visualizations available that allow

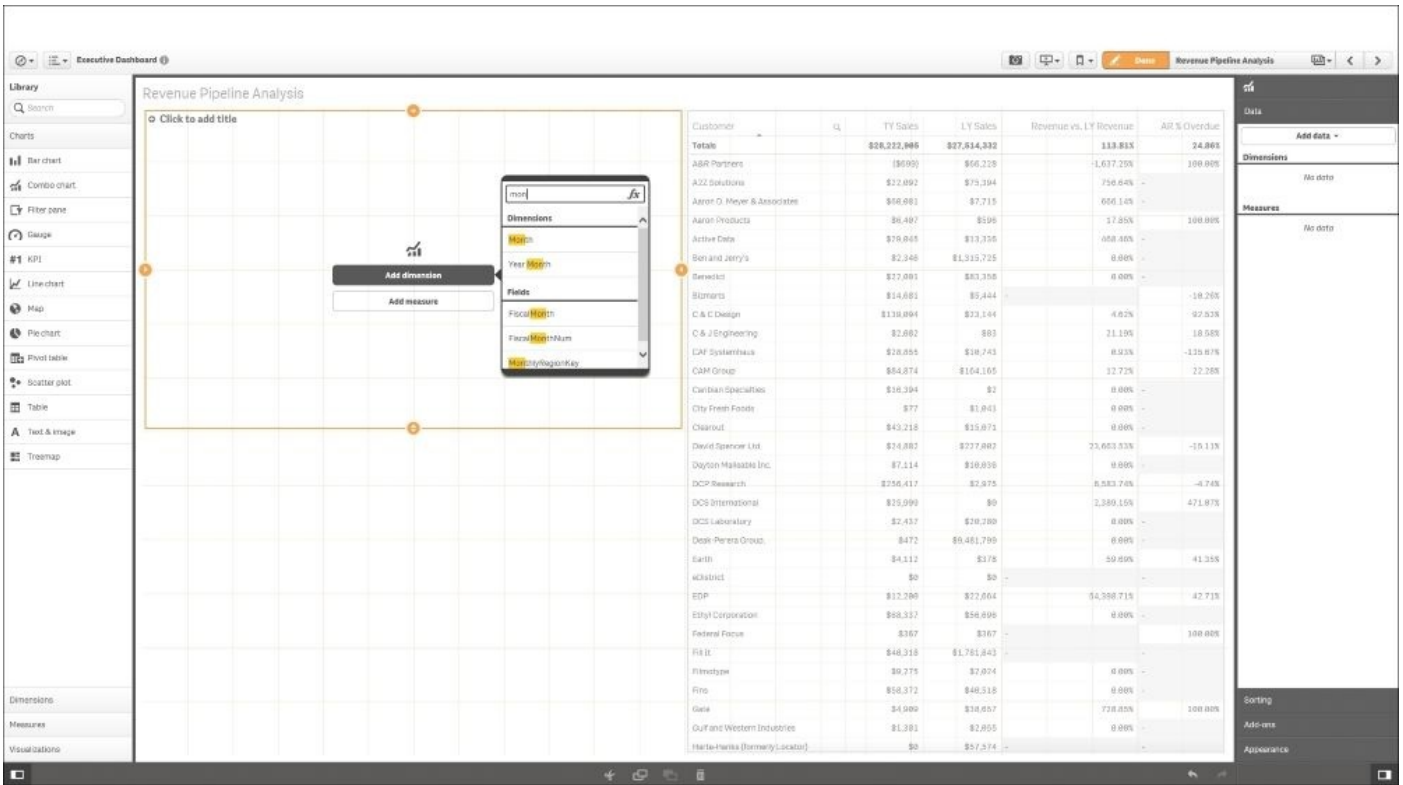
her to see the trend of inventory on hand and sales orders. This requires her to create a new chart based on dimensions and measures defined in **Library**. So, to begin this process, Pat selects and drags **Combo chart** to the sheet noted in the following screenshot:

The screenshot shows the Qlik Sense interface with a 'Revenue Pipeline Analysis' sheet. A 'Combo chart' object is being dragged from the Library. The sheet contains a table with the following data:

Customer	FY Sales	LY Sales	Revenue vs. LY Revenue	ARS Overdue
Totals	\$38,202,886	\$37,514,332	113.81%	\$4.86%
A&R Partners	\$599	\$66,278	-1,637.23%	100.00%
AZZ Solutions	\$22,892	\$75,304	756.64%	
Aaron D. Meyer & Associates	\$69,081	\$7,715	886.14%	
Aaron Products	\$6,487	\$596	17.85%	100.00%
Active Data	\$25,045	\$15,336	408.65%	
Ben and Jerry's	\$2,340	\$1,115,729	0.06%	
Berwick	\$72,091	\$83,338	8.06%	
Bitmarts	\$14,681	\$5,444		-19.20%
C & C Design	\$139,994	\$23,144	4.82%	92.52%
C. R. J. Engineering	\$7,887	\$83	21.19%	18.50%
CAF Systemhaus	\$28,855	\$18,743	8.03%	-155.67%
CAM Group	\$84,974	\$164,165	17.73%	27.10%
Carlson Bookstores	\$16,394	\$2	0.06%	
City Fresh Foods	\$77	\$3,941	0.00%	
Cleorcut	\$43,218	\$15,871	0.00%	
David Spence Ltd.	\$24,882	\$27,062	23,963.53%	-15.11%
Darton Malleable Inc.	\$7,114	\$19,036	0.00%	
DCP Research	\$258,417	\$2,975	6,582.74%	-4.74%
DCS International	\$25,969	\$8	7,388.13%	471.97%
DCS Laboratory	\$2,437	\$28,288	0.06%	
Deak-Powers Group	\$477	\$9,481,790	0.00%	
Earth	\$4,112	\$378	59.05%	-41.35%
eDistrict	\$8	\$8		
EDP	\$12,390	\$22,064	54,208.73%	-82.71%
Ethyl Corporation	\$66,337	\$56,696	0.06%	
Federal Focus	\$367	\$367		100.00%
FETH	\$48,318	\$1,781,843		
Firmotype	\$9,275	\$2,824	0.06%	
Flux	\$58,377	\$48,618	0.06%	
Gate	\$4,969	\$36,657	736.85%	100.00%
Gulf and Western Industries	\$1,381	\$7,855	0.00%	
Harta Hanks (Formerly Locator)	\$8	\$67,574		

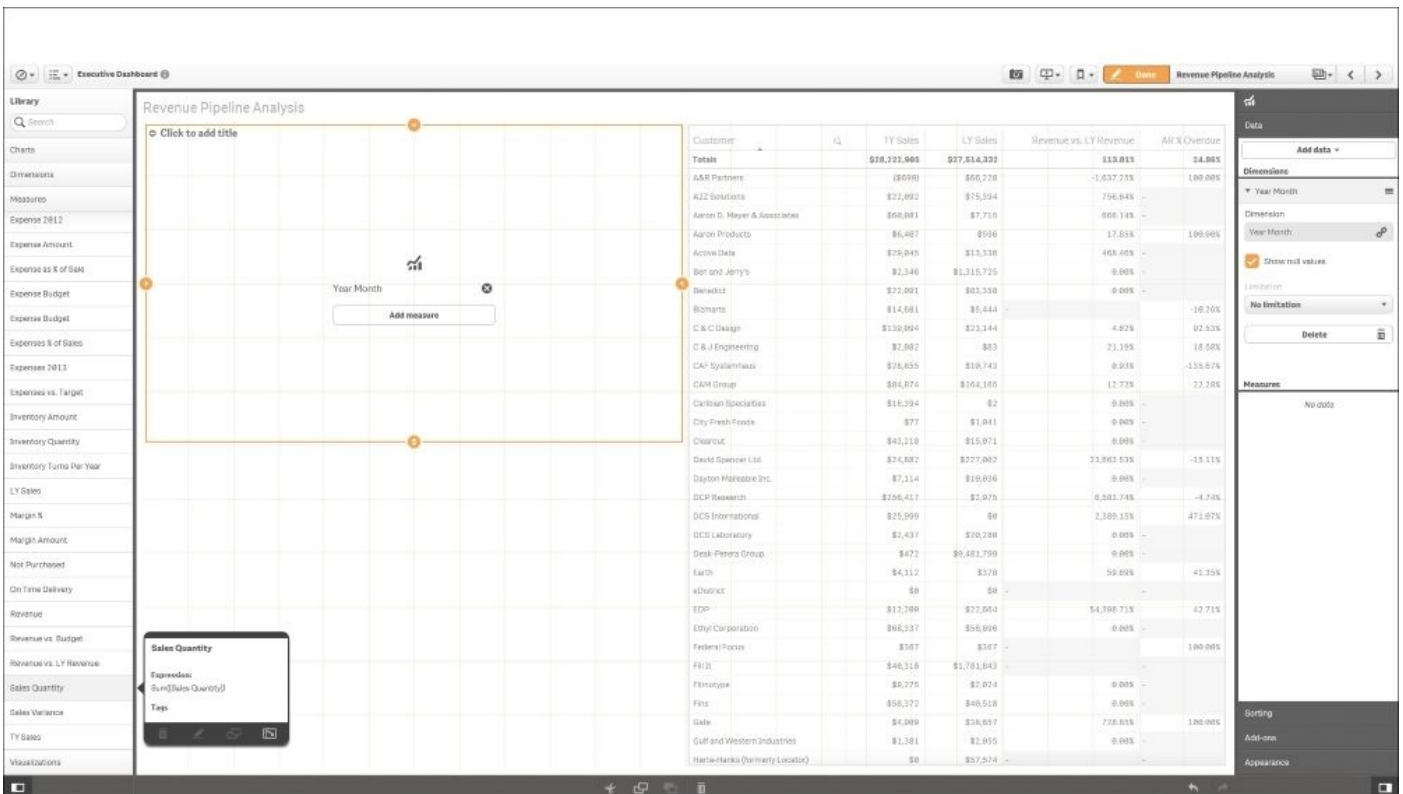
### Creating a Combo chart object

Once the Combo chart is in position, the object guides Pat on the requirements for visualization. The object highlights the requirement of at least one dimension and a measure. To speed up the task, the Qlik Sense search capability can be used to find the dimension; in this case, **Year Month**:



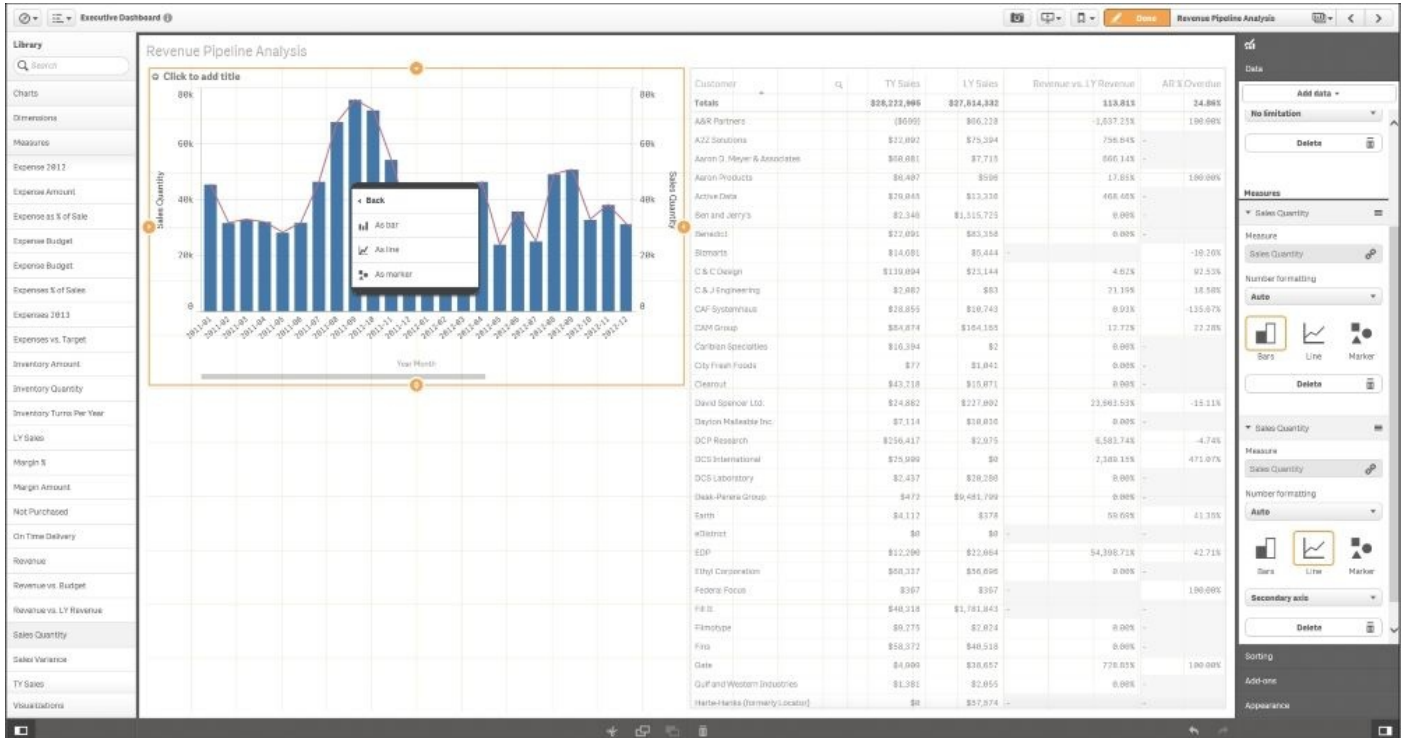
### Adding a dimension

The next step is to add the measures; as this is **Combo chart**, there will be two measures. The first measure added will be **Sales Quantity**. The **Sales Quantity** measure is available in **Measures**, and a tooltip reveals the expression that shows how it is defined:



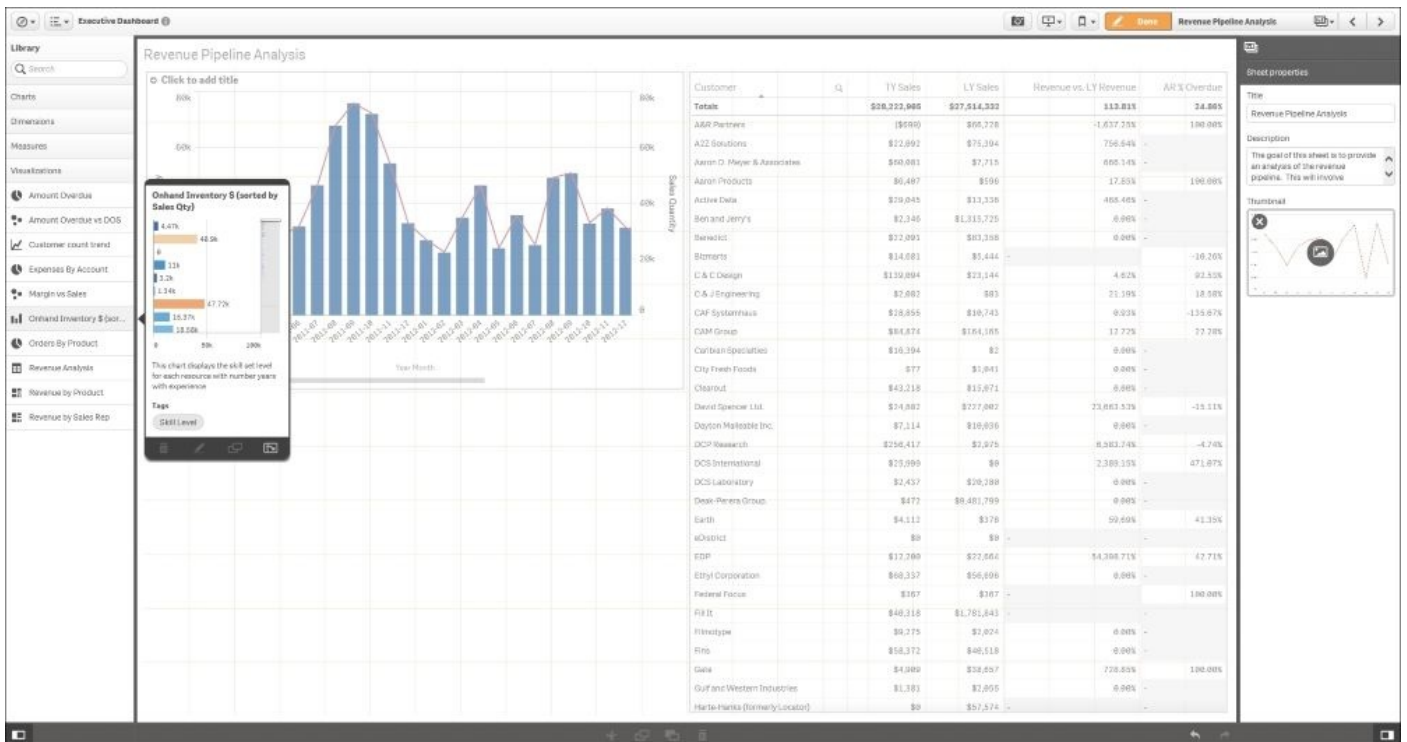
### Adding a Sales Quantity measure

Additionally, as the measures are dragged and dropped on **Combo chart**, the object continues to guide Pat on how to visualize the data. Qlik Sense provides guidance to add the inventory quantity and options for display. **Combo chart** supports bar, line, and marker chart types. In this case, Pat selects a line to compliment the **Sales Quantity** measure that is already displayed as a bar. The selection of charts can also be changed quickly while keeping the defined dimensionality of the previous chart:



### Adding the Inventory Quantity measure

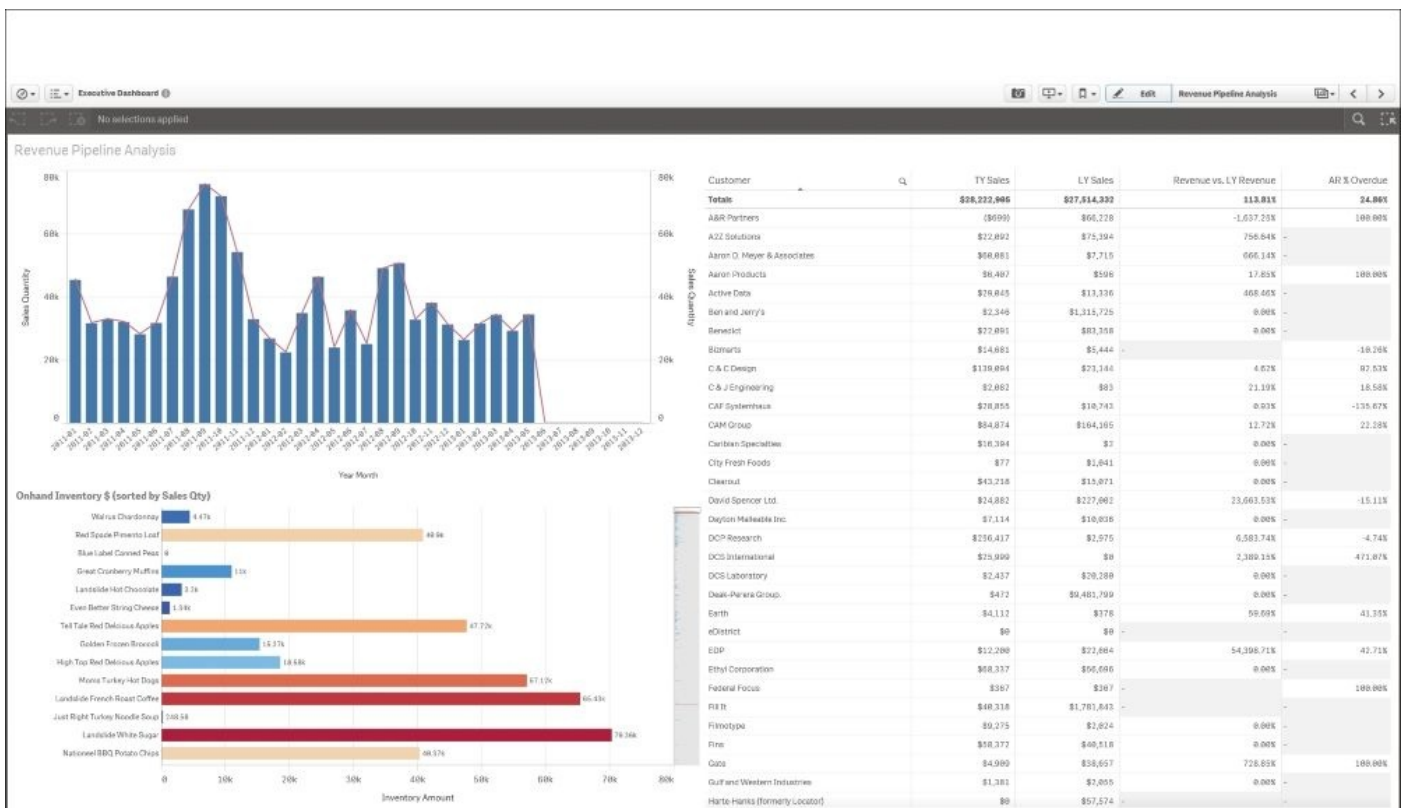
Finally, Pat completes the sheet layout by adding the **Onhand Inventory \$ (sorted by Sales Qty)** chart available in the **Visualizations** portion of **Library**, which is shown as follows:



Adding the Onhand Inventory \$ (sorted by Sales Qty) chart

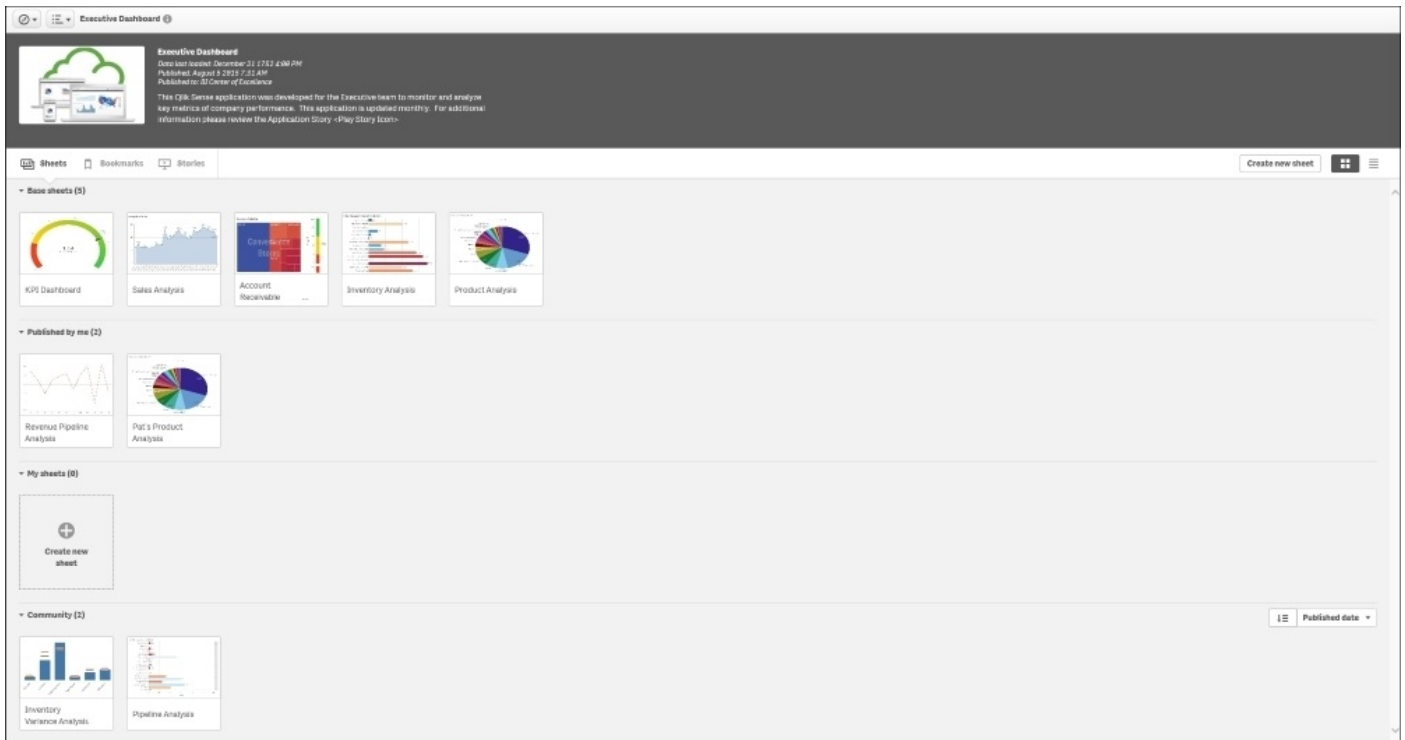
## Publishing a private sheet

With the new **Revenue Pipeline Analysis** sheet completed, Pat is ready to publish with a right-click, as described earlier. Also, it is worth mentioning that this assembled sheet is fully selectable during the process of assembly, and no wiring (connecting) of these objects is needed to allow them to communicate with each other across all sheets:



## Completed Revenue Pipeline Analysis sheet

Once the sheet is published, it is available to the **Executive Dashboard** application, where it can be consumed, duplicated, and expanded by other members of the community:



## Published Revenue Pipeline Analysis sheet

In summary, creating new sheets provides an alternative way to collaborate with members of the application's community. It allows contributors in a governed environment to start with a blank sheet to organize and share their thoughts and insights, and is managed centrally in the Qlik Management Console. Now, let's turn our attention to creating a Qlik Sense story, which adds additional capabilities for collaboration.

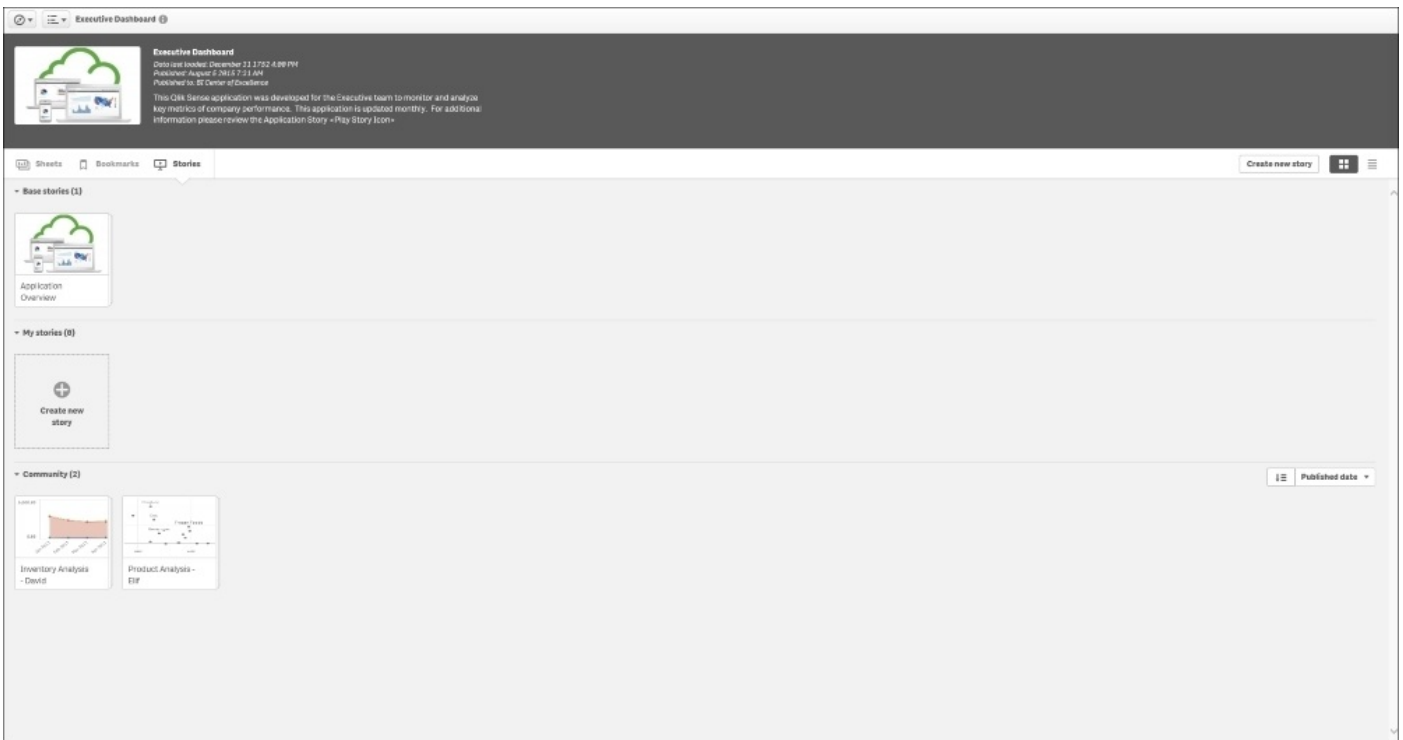




# Creating and sharing stories

Qlik Sense Stories provide an additional capability to collaborate and share business discoveries within the Executive Dashboard community. In the story overview, we can see that similar to bookmarks and sheets, stories have **Approved stories** (defined by the author of the application), **My stories** (private and only viewable by the author), and **Community** (published) sections. We covered the role of approved stories as a way for application authors to provide an overview about the application and intended use.

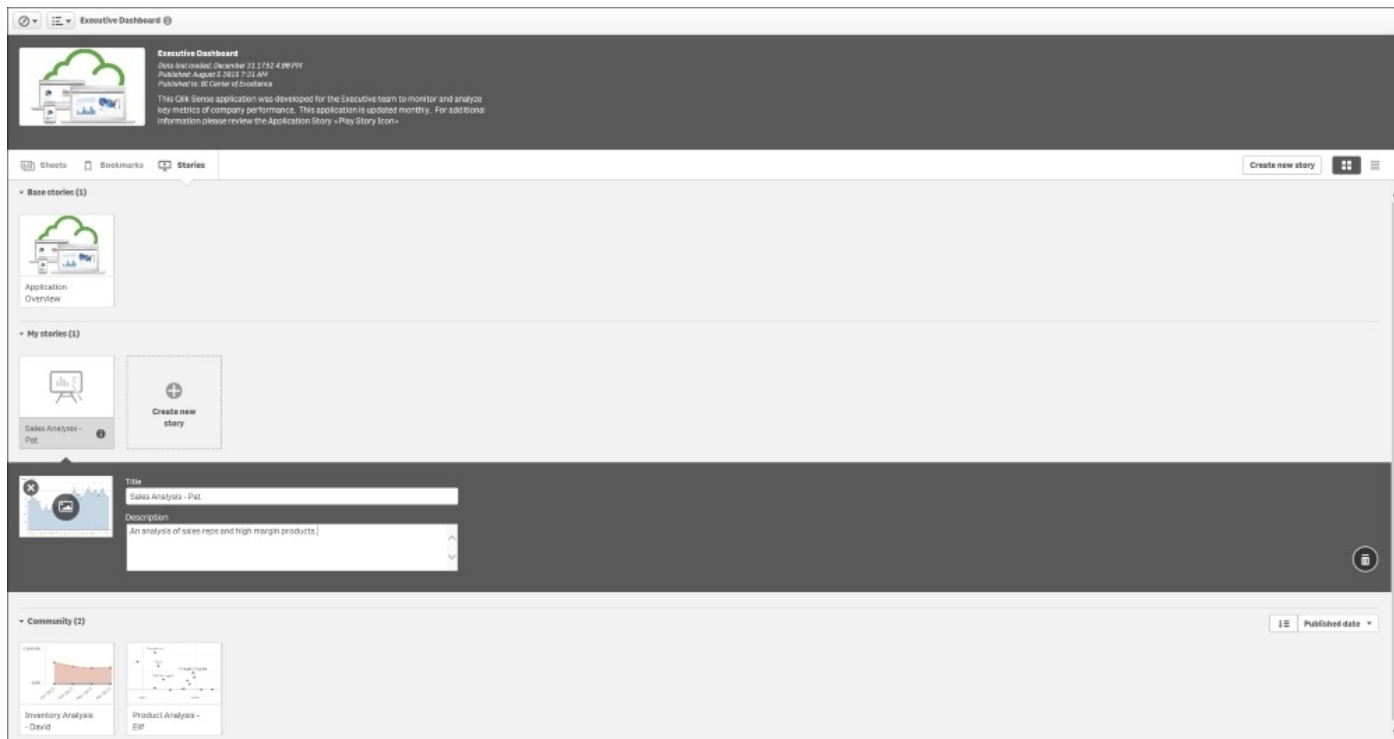
In this section, we will focus on the creation of a story by a contributor (Pat) who will use this capability to present a sales analysis to the community:



*Story overview*

# Defining a story

To begin with, Pat creates a new story by selecting the **Create new story** option under **My stories**. The default name is **My new story**, which Pat changes to **Sales Analysis** – Pat to reflect the goal of the story. Additionally, a description can be added to provide information on the goals of the story:

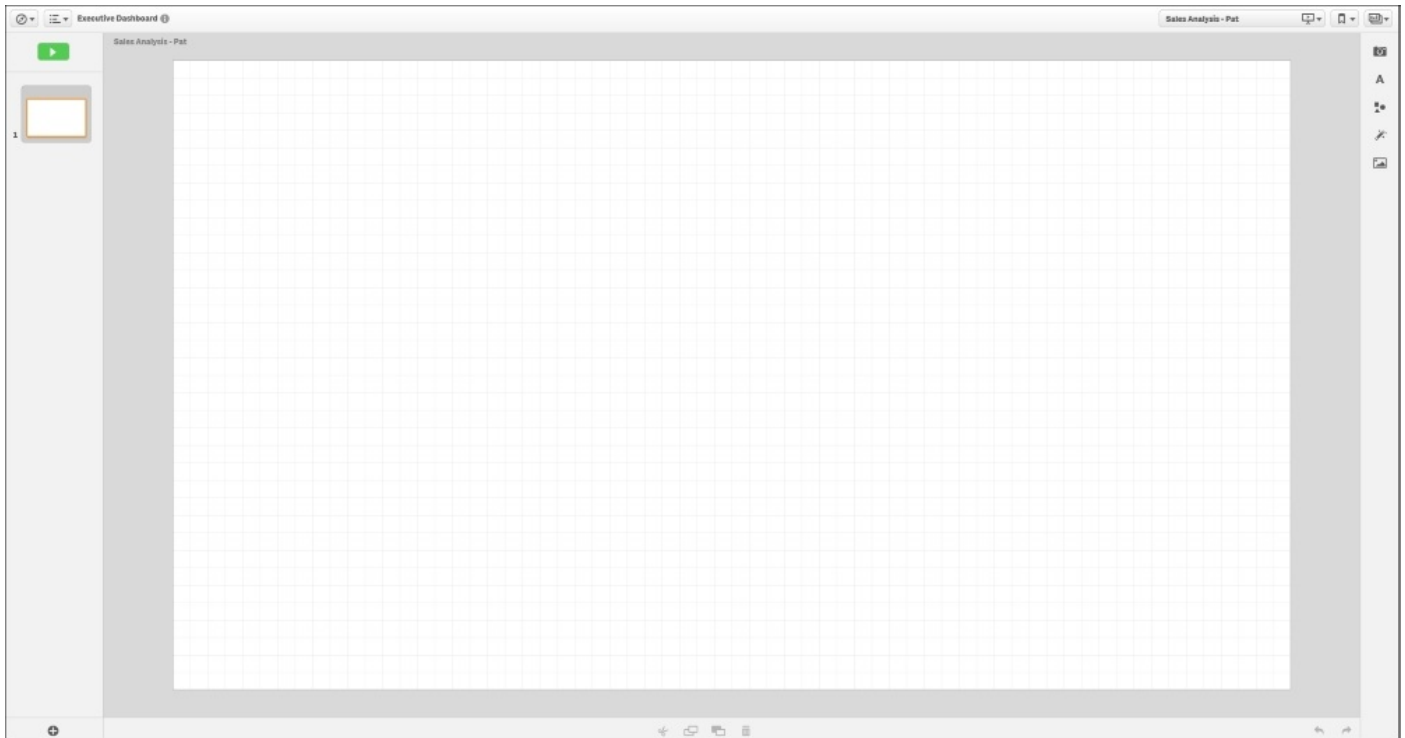


## *Defining a story*

Once the story is defined, Pat enters the story workspace, which provides a broad set of tools to create rich presentations that are dynamically linked to the Qlik Sense application. The story workspace contains the ability to create sheets as well as access the following libraries:

- Snapshot
- Text
- Shape
- Effect
- Media

We will explore each of these areas as Pat defines her presentation:



*Story workspace*

## **Creating snapshots**

Let's start with creating snapshots. The ability to create a snapshot is a general capability found on all sheets within an application. Snapshots provide analysts like Pat with the ability to capture insights across a Qlik Sense application and organize them with additional context through stories. By selecting the camera, all objects for which snapshots can be created are highlighted with an orange outline. Additionally, each object also contains an indicator that highlights the number of times snapshots have been created for the object. As you can see, Pat has been quite busy in selecting key objects for her story:



### Creating snapshots

Now that Pat has selected her snapshots, she prepares to organize them in a story. Note that all snapshots are stored in **Snapshot Library**:



### Story snapshots

## Note

These snapshots are organized by the date and time when they were taken. This is an important consideration because it means that snapshots are like photos, storing the visualization and data of the time the snapshot was taken. By design, snapshots are not updated when the application data is changed.

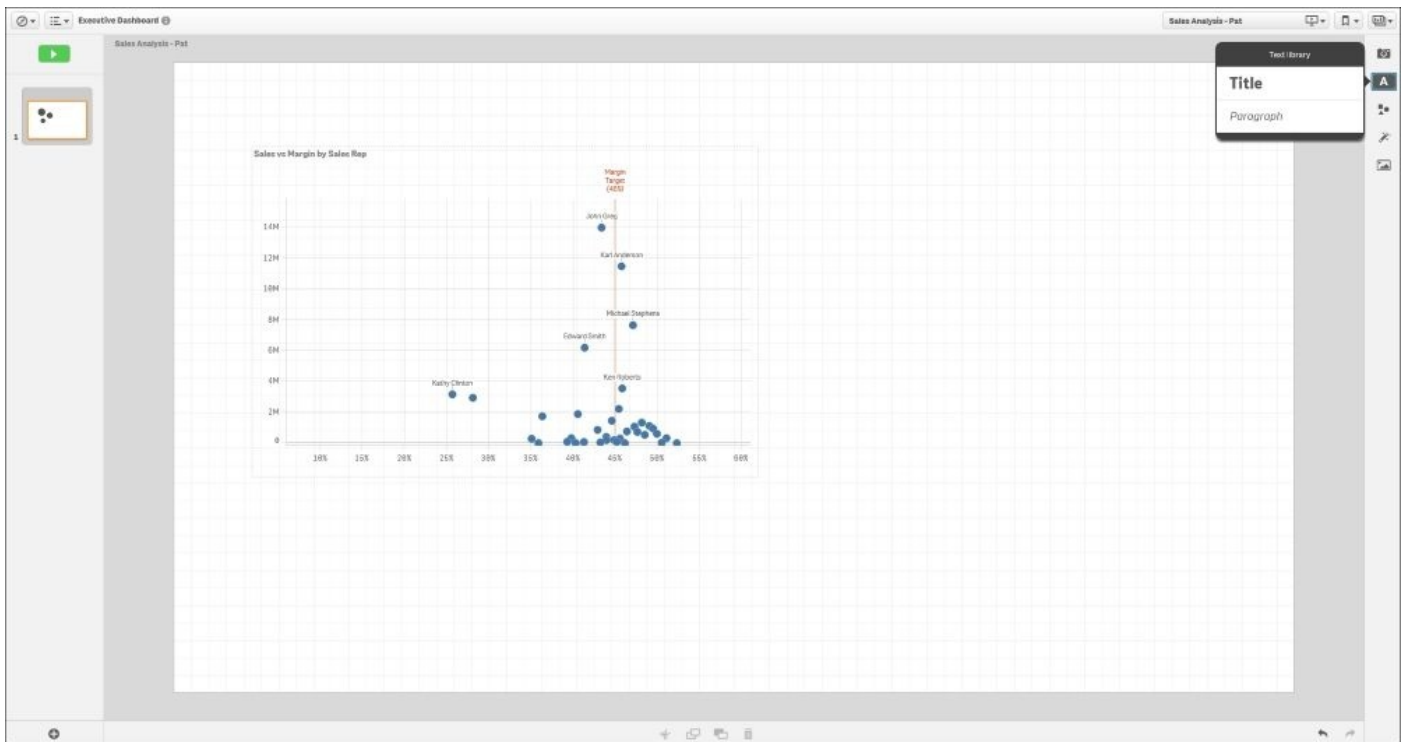
Once the snapshots are taken, Pat locates the shot and then drags and drops it onto the grid. Also, note that each snapshot can be (unlocked) edited with the ability to modify some of the properties, which can include turning on/off titles and labels:



*Adding a snapshot*

## Adding text

Now, let's add text to this sheet, which is accomplished through **Text library**. The **Text library** facilitates both the creation of titles as well as paragraphs that can be used to add comments to highlight key business discoveries:



*Adding text*

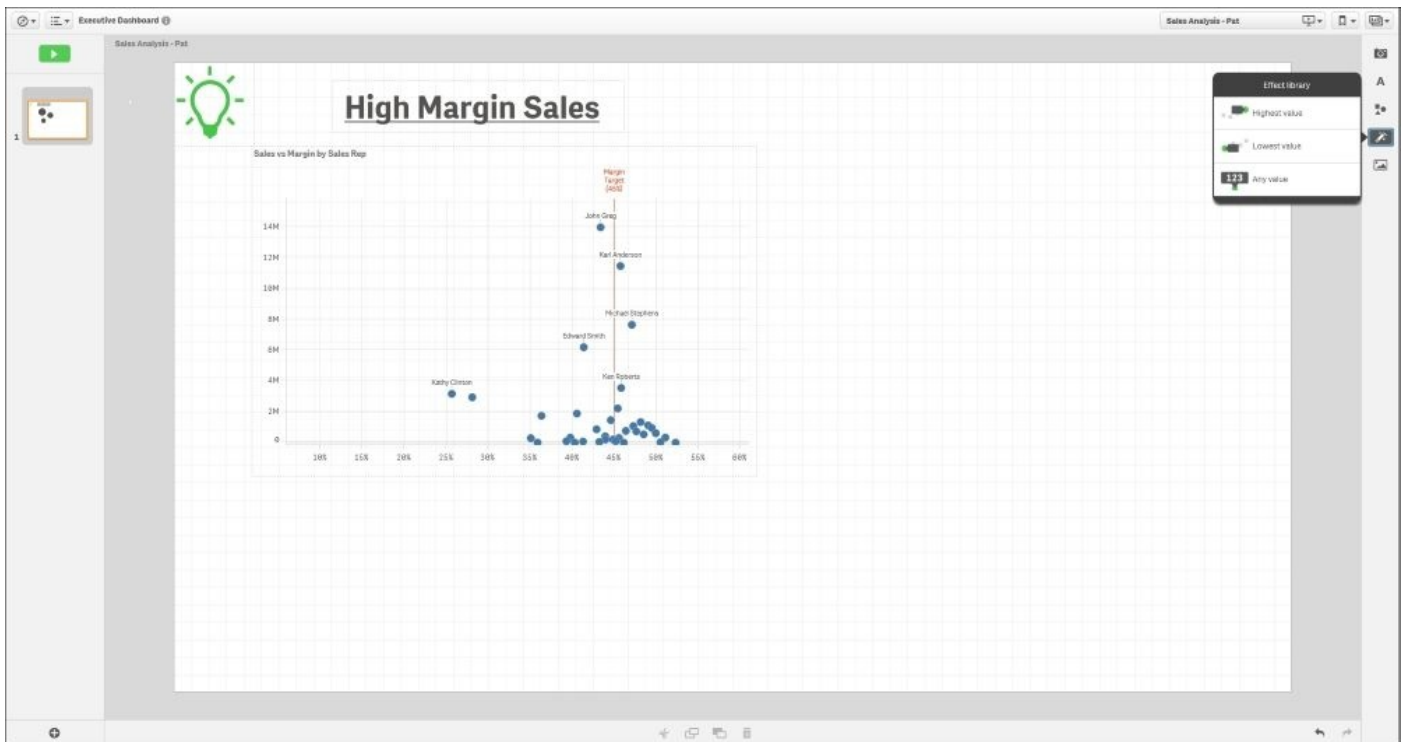
## Adding shapes

Pat has added the title **High Margin Sales** and emphasized it with bold and underline styles. Additionally, there is **Shape library**, which allows the integration of various shapes to highlight and emphasize the story:



*Adding a shape*

Pat chose the light bulb symbol to identify key ideas in this story. The symbol's default color is black, but can be changed. Additionally, there is **Effect library**, which can be used to highlight the lowest, highest, or a particular value within a chart:

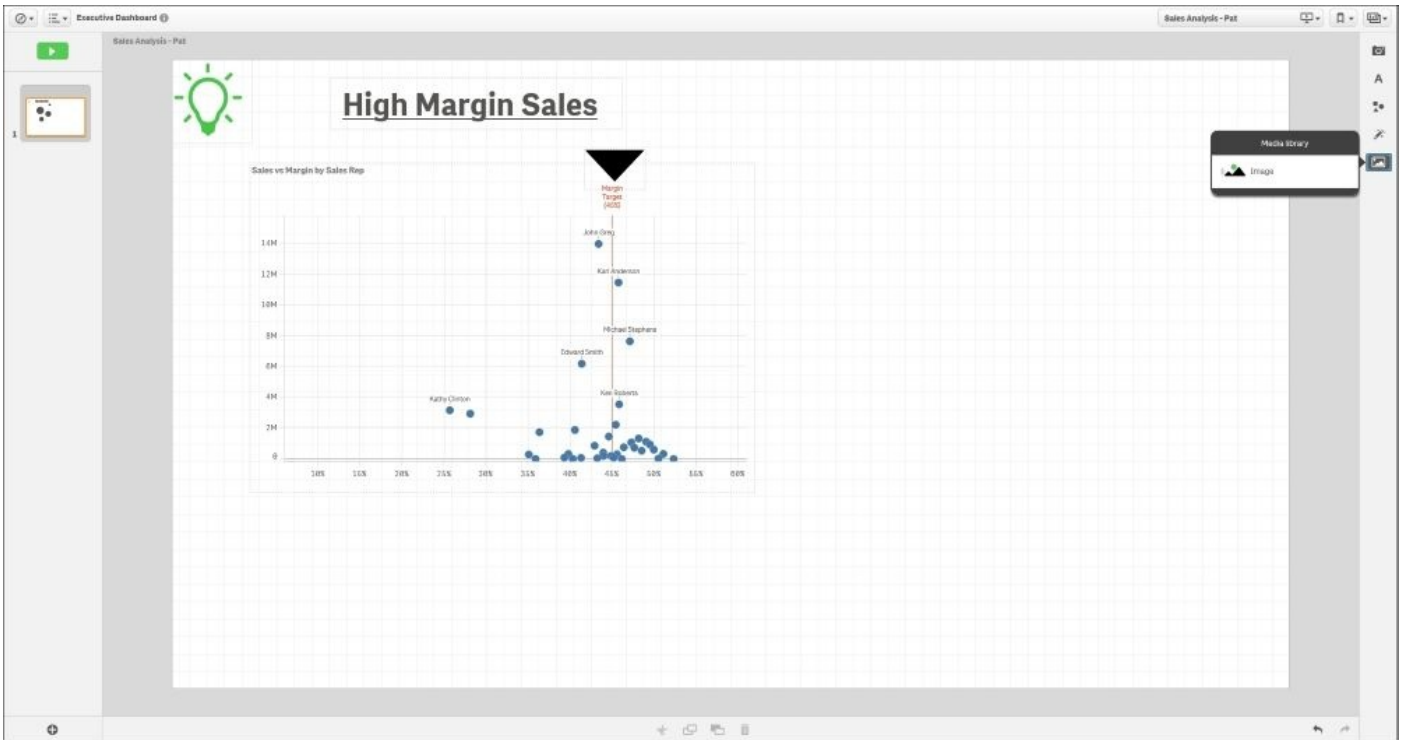


*Adding effects*

## Media library

The final major area is **Media library**, which offers contributors the ability to add images from outside Qlik Sense. Images are made available and managed by the Qlik Management Console through **Content library**. Additional information on this process is available in [Chapter 9, Administering Qlik Sense®](#).





*Adding media*

Additionally, Pat would like to add a portion of the **Sales Analysis** application directly in her story. Qlik Sense Stories also allow an approved sheet to be embedded within the **Sales Analysis** story. This enables Pat to share her analysis through a number of slides and also offers the viewer, the ability to continue their exploration through an active sheet.

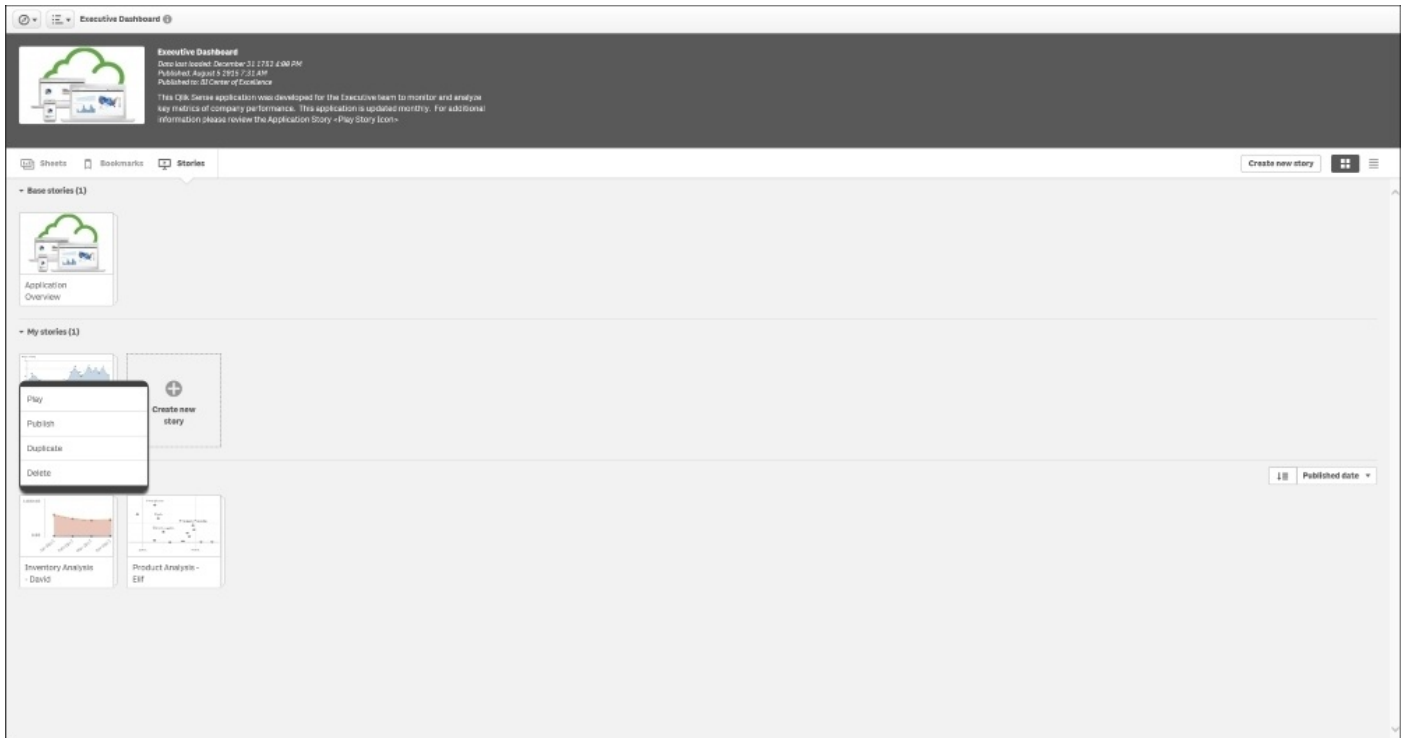


*Adding a dynamic slide*



# Publishing your story

Now that Pat is comfortable with the **Sales Analysis** story, the publishing process is similar to the publishing process of private sheets. To accomplish this, as illustrated, Pat right-clicks and selects **Publish** to move the story to the community and make it read only:



*Publishing the story*



# Summary

One of the strengths of Qlik Sense applications is the ability it offers contributors to actively share, collaborate, and extend the value of the application with members of the stream. Qlik Sense has a number of exciting ways to share key business discoveries. These include bookmarks, published sheets, and stories. Each of these approaches is highly governed and provides a wide range of capabilities to meet the needs of a contributor.

In the next chapter, we will explore using the skills we learned alongside some ideas of best practices in how to create author-engaging applications for Qlik Sense.



# Chapter 5. Authoring Engaging Applications

In the previous chapters, we looked at the application life cycle and the different roles of users: the consumer and the contributor. Having established the basic requirements, in this chapter we will dive into the details of app creation and discuss how it is done. We'll also look at best practices of visualization and how to employ them using Qlik Sense.

In this chapter, we will discuss the following topics:

- The process of building an app
- Data connectors
- The data model viewer
- Sheet objects—visualizations
- Best practices
- Migrating QlikView applications into Qlik Sense

# Preparations and requirements

Often the initial step in building an app is that you have some data that you want to analyze, but you *don't necessarily know* exactly what you want to look for in the data. As a business user, you can—and should—just load this data into Qlik Sense and start developing. Our experience is that the best way to develop the app is to start *without* first defining the requirements.

The reason is that when you load data and start to create visualizations, you *learn* from data. This knowledge is very important once you start defining what you want to analyze. Hence, you should first develop a basic app, then take a break and evaluate what you learned. *Now* is the right time to start formulating the requirements.

Another common case is the opposite situation: you know that you want to calculate a specific KPI, for example, supplier efficiency, but you don't necessarily know what data you need to be able to do this. In this case, you need to start with some research about where to find the relevant information, that is, in which database and in which tables.



# The requirement specifications

If you define a larger project, you will use what you know as a starting point for the requirement specifications for your app. The following questions might pop up:

- **Data:** Which data sources should be used? Which tables should be used? How should the tables be linked? Are there common keys? Is there more than one source for the transactions? Are there tables missing? How should the customer hierarchy be resolved?
- **KPIs:** Which calculations should be made? You could consider turnover, profit, cost, delivery accuracy, or product quality. Which definition of gross margin should be used? How should the given discount affect the calculation of a salesman's bonus? Which accumulations are needed: year-to-date or month-to-date?
- **Dimensions:** How should the KPIs be displayed? You could consider showing them per year, per customer, per salesman, per region, or per product. Which comparisons should be made: year-over-year or month-over-month? Are there drill-down hierarchies that need to be defined?
- **Security:** Is the data confidential? Who gets to see what? Can we allow offline usage? Is the authorization data driven or static? Do we need to include authorization information in the data model, or can we postpone the decision around security?

You will soon realize that creating the requirement specification is not an easy task.

# The communication problem

Discovering exactly what users, stakeholders, and sponsors want you to create is often the most difficult part of a business intelligence project. The communication between IT experts and nontechnical business users is often full of misunderstandings and misinterpretations. Business users often don't know what they want until they see it, and they frequently can't articulate their expectations in languages that IT experts use to design systems.

Few business users will know what a **data model** really means, so expecting them to be able to exactly define the requirements in technical terms is futile. Experienced authors can extract this information through discussions and clever questioning, but the number of people who are able to do this within an organization is limited.

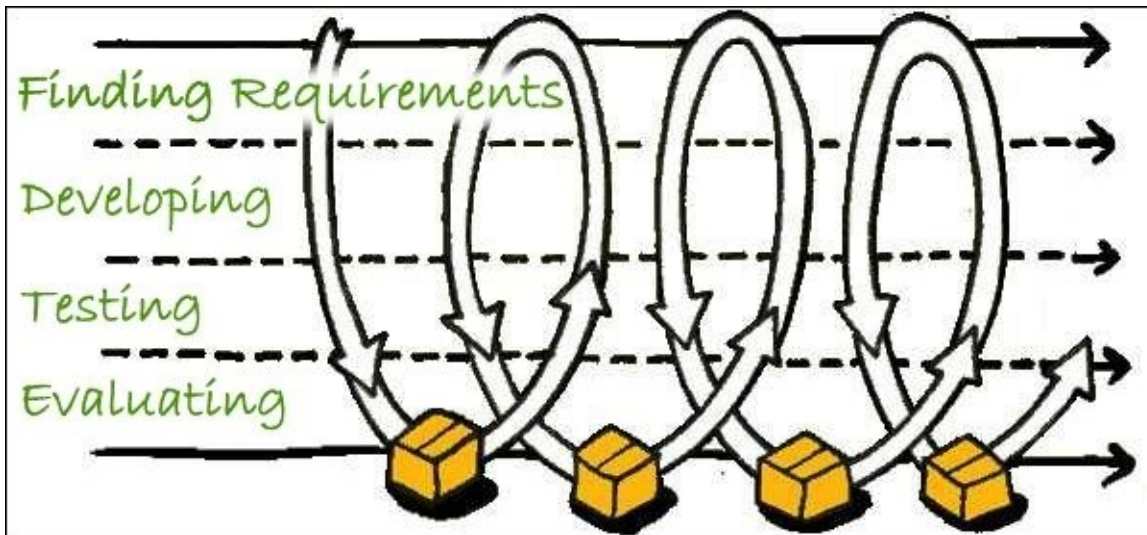
IT professionals often frame their requirement questions in technical language, for example, "Which table in the database should be used?" or "Which fields should be used to calculate the KPI?". However, business users may not have the technical knowledge to respond to these questions. Business users often explain their expectations in a technically vague language, which is not specific enough for designers to develop solutions.

On the other hand, *the business user is the customer*. The very reason why we develop an app in the first place is to supply the business user with a tool to analyze and learn from data. So, the requirement specifications *must* focus on the business user.

# A step-wise implementation

The solution to this communication problem is to use a step-wise implementation, where the app developer iteratively finds the requirements, develops the app further, tests what has been done, and finally evaluates the app together with the business user. The evaluation will lead to new requirements and to changes or refinements of the old requirements. The steps must be small and the typical cycle is hours or days.

In other words, you *discover* the requirements together with the business user. As the development proceeds, the app will converge to the needs of the business user.



*The iterative development process*

This means you *cannot* begin your app development with a detailed requirement specification. Rather, you should start with a very basic specification containing information about some of the needed data sources and ideas of some of the required visualizations.

Hence, irrespective of whether you are a business user or an app developer responsible for data modeling and difficult formulas, you should start by spending an hour or so to load the data and create some graphs with the goal to *learn* from data. Then, you are in a much better position to define or discuss requirements further.

# The process

The first step in building the app is to load the data. The data can be one single table or several tables linked logically by **key fields**. Key fields are fields that exist in more than one table and link rows in one table with rows in another. Together, they form a data model. The next chapter will discuss the data model, so, for the moment, we will not get into the details of this.

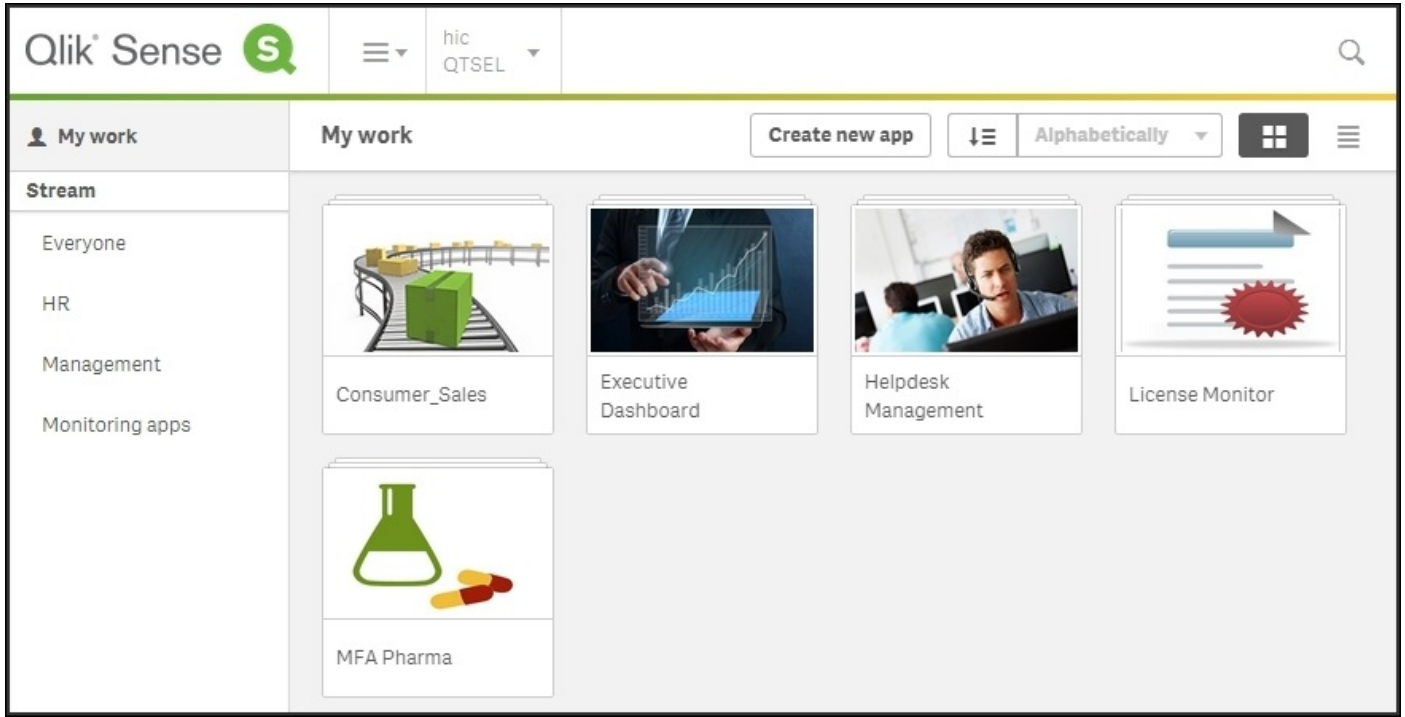
When you have a data model, you can start building the layout, which consists of different objects, for example, lists, graphs, tables, and filter panes, placed on different worksheets. The objects can contain formulas that define different calculations that will be calculated as the users make their selections.

The previously explained development model assumes that you have both a developer and a business user that participate in the development process. In real life, you will notice that the initial development efforts will usually be like this, but as the app takes shape, the business users will want to do more and more on their own—which is good. After all, the goal is to have business users who are self-sufficient and create apps as much as possible on their own.



# Getting started with the app creation

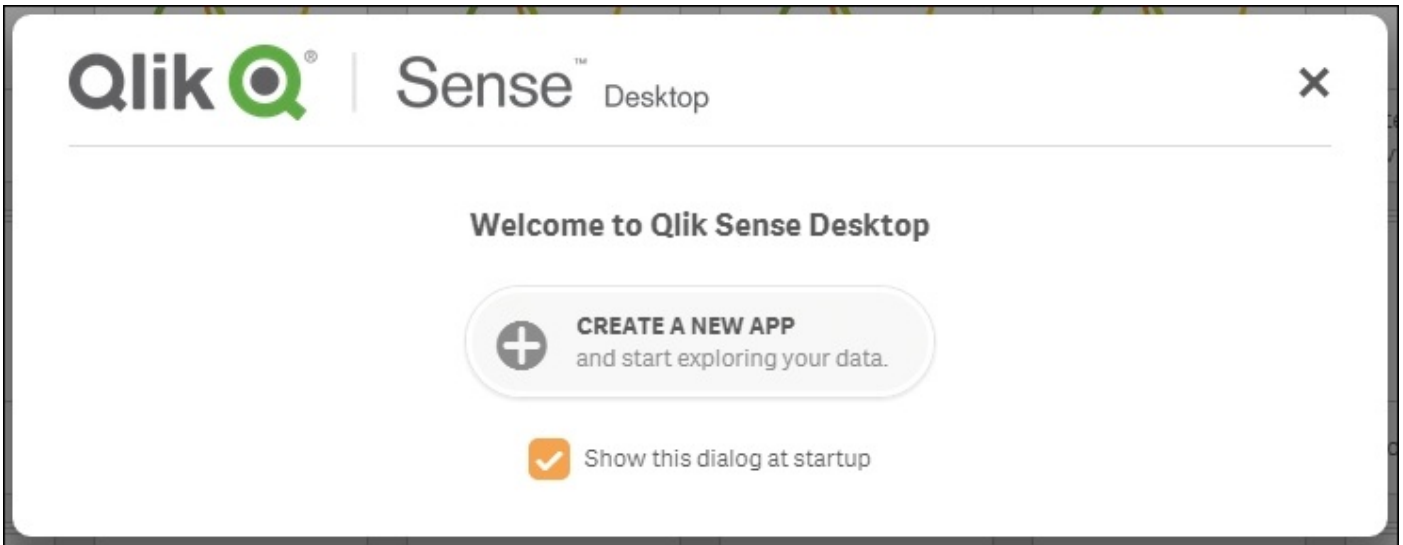
When you first open Qlik Sense, you come to the hub. This is the place where you have an overview of all your apps. The hubs look slightly different in the Desktop and Server versions, but they are essentially the same. The following screenshot shows what a hub looks like:



*The Qlik Sense hub*

# Creating a new app

In Qlik Sense Desktop, you are greeted with a dialog that asks you to create an app as shown in the following screenshot. In the Qlik Sense server, you will find the corresponding functionality on a button labeled **CREATE NEW APP** in the toolbar:



*The Qlik Sense Desktop welcome dialog*

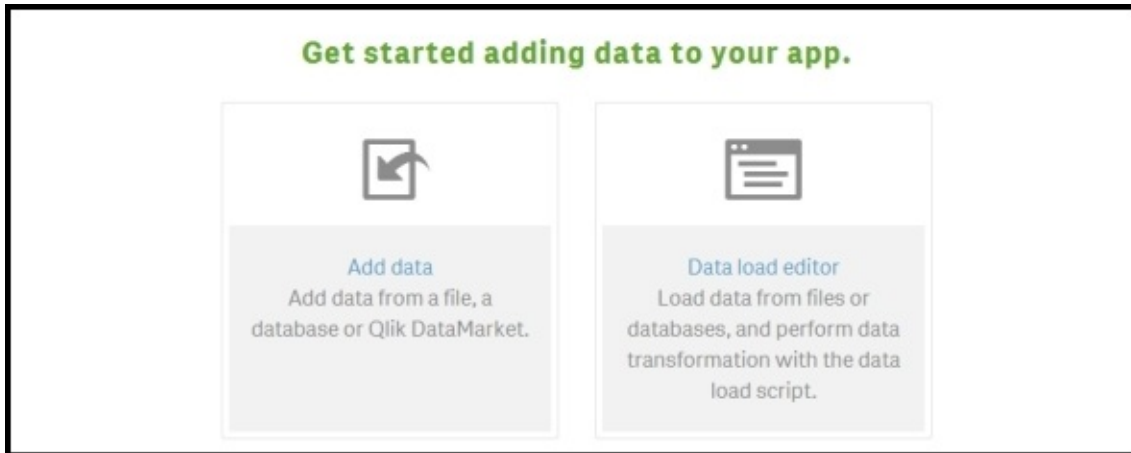
Creating an app means you will create an entity that will hold both the data and everything else needed to analyze it. In Qlik Sense Desktop, this is a file created in `C:\Users\User\Documents\Qlik\Sense\Apps`.





# Loading your data

Once you have named and opened your app, you will get a screen where Qlik Sense asks you to load your data:

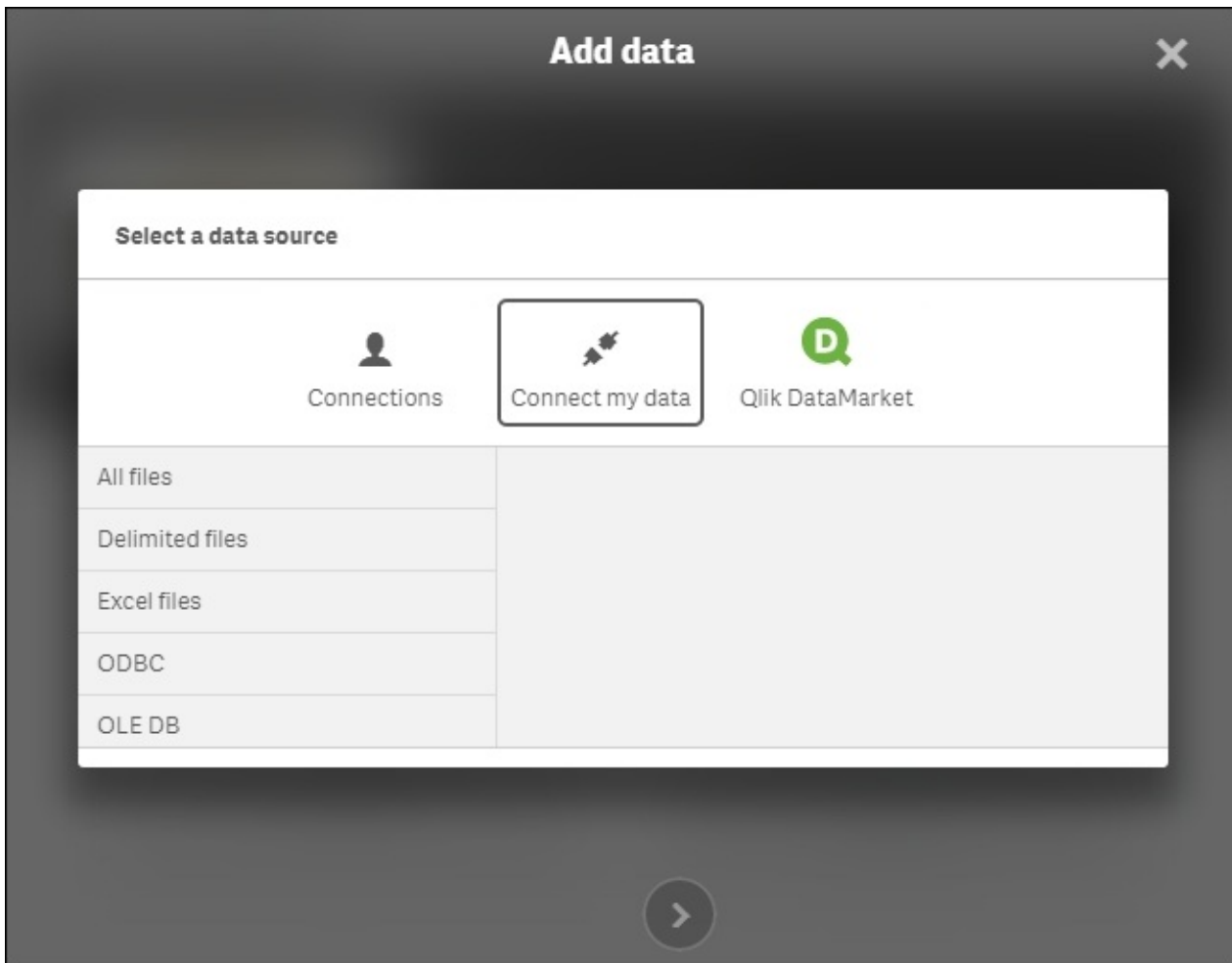


*The Get started screen*

When you have this option before you, you can load data in several different ways. The **Add data** command to the left will start with a wizard that helps you define what you want to load. In the background, it adds code to a script that defines the load sequence. It will, however, never show the script.

This is different from **Data load editor** that will take you to a script editor, where you can change the script directly.

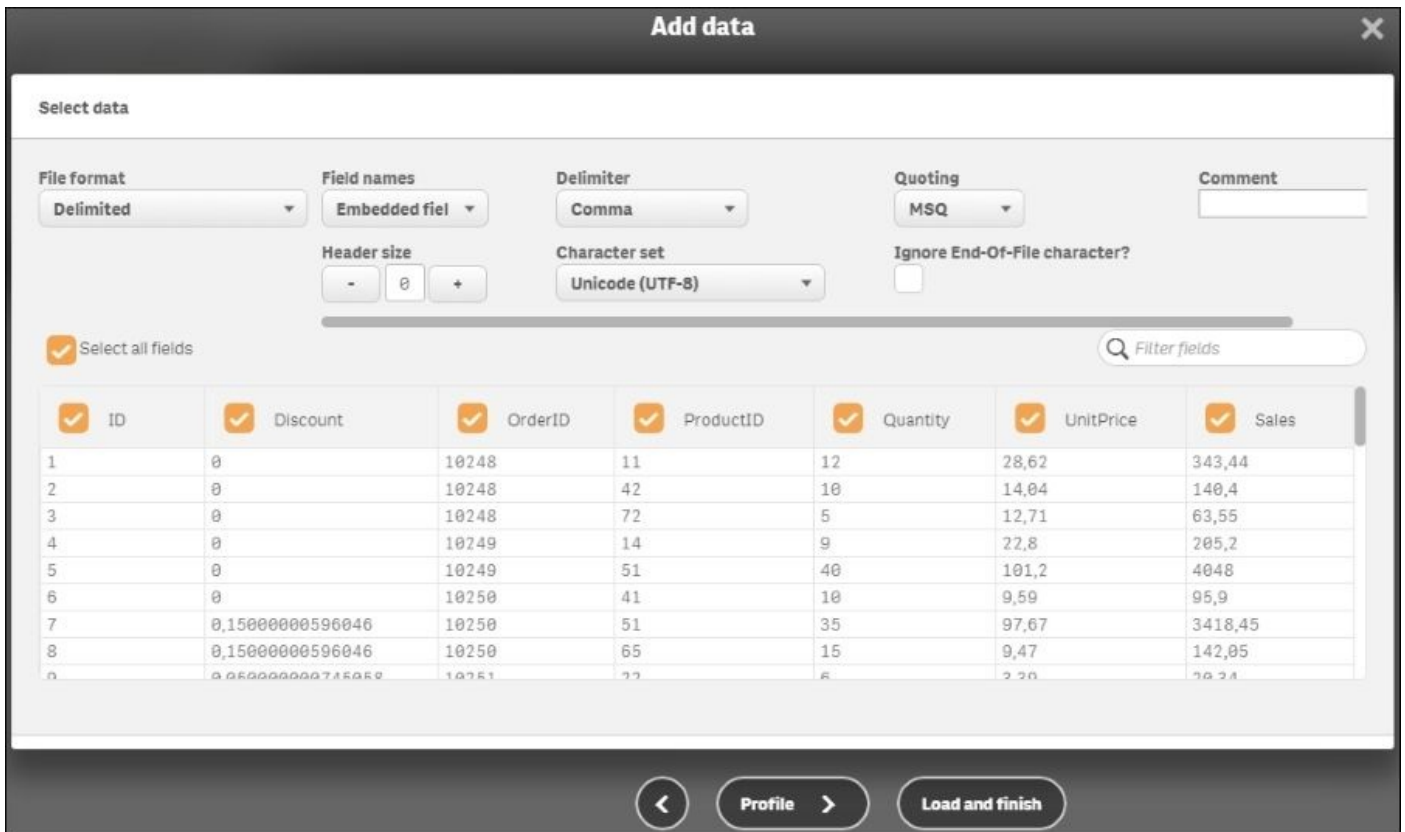
The easiest way is to use **Add data** dialog to the left. This will open the **Add data** dialog where you can define a data source; either a database table or a file, for instance, an Excel spreadsheet.



*The Add data dialog*

Use **Connections** for folders and connections that you have previously used, and use **Connect my data** if it is a new data source. In this dialog, you can select your database table or browse your way to a file containing a table.

Selecting a file will open a file wizard where you can tweak the details of how the file should be loaded so that you get the data you want, as shown in the following screenshot:



### *The file wizard*

You can specify the file type, whether the file contains empty lines at the top (before the data starts), whether the first line contains the field names or not, and so on. Make sure you get all the settings right before you click on **Load and finish**. If it is the first table you load, you don't need to use the **Profile** button. The next section explains where we load additional tables.

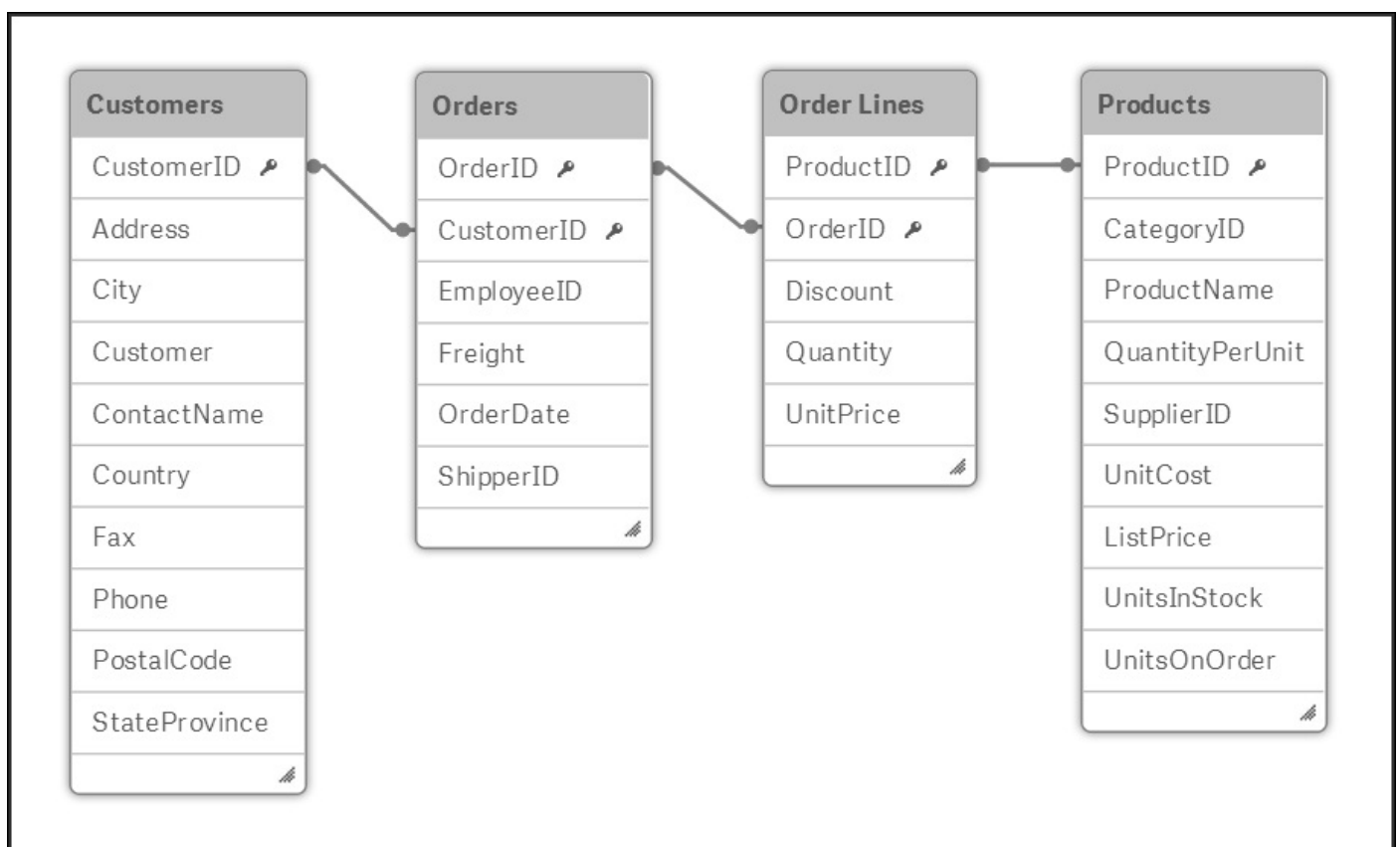
Clicking on **Load and finish** will store your settings and load the data; if everything goes well, you will get a message that the data was loaded successfully. At this stage, you can either start to edit the sheet or close the dialog and perhaps load additional tables.

# Loading additional tables

It is very common that you want to analyze data that is stored in several different tables. For example, you could have four tables: one table for the orders (one row per order), one table for the customers who placed the orders (one row per customer), one table for the order lines (one row per order line), and one table for the products (one row per product).

For such a case, the orders table will contain a field that specifies the customer that placed the order—a customer ID. In the same way, the Order Lines table will contain a field that points out which order the record belongs to, an order ID, and another field that tells which product the record refers to, a product ID. Such fields are called keys, and Qlik Sense uses these to link the tables and make sense of the data.

This way, all four tables are linked logically, as shown in the following picture:



*A simple data model made from four tables*

To obtain this data model in your app, you need to repeat the procedure for loading data that we discussed previously for each additional table. You can then open the **Add data** dialog from one of the menus to the left in the toolbar. This will take you to the familiar file wizard shown in the previous section, where you can define the file properties of the additional table.

Optionally, you can also go through the profiling step, which helps you define the keys. The profiling looks at the field values and compares these with those of the previously loaded fields. When this is done, it suggests which fields to use (or not to use) as links,

and renames these fields appropriately so that they become keys in your data model:

**Add data**

Table association recommendations ⓘ

⚠ Customers - Orders 1 warning 1 of 1

Current: No association  
Recommendations: ID CustomerID 99%

Rename both fields to:  
ID Use custom field name CustomerID

Customers					Orders					
CompanyName	ContactName	Country	ID	Fax	CustomerID	EmployeeID	Freight	OrderDate	OrderID	ShipperID
Boehuset Abel	Dennis Olsen	Denmark	94	86 22 33	7	5	26,24	2004-01-16	10360	1
Stefhanies	Staffan Blond	Sweden	127	(8) 11 27	63	7	55,52	2004-11-28	10361	2
Strach GS	Albert von Einstein	Germany	1	030-007	44	8	40,91	2001-11-20	10362	2
Tienda de la Esquina	Paco el Maco	Mexico	2	(5) 555-	17	8	36,54	2004-02-20	10363	2
Ropa Vieja	Sancho Panza	Mexico	3		19	2	33,65	2001-11-21	10364	2
Jims Trousers	Carl Montgomery	UK	4	(171) 55	3	2	35,04	2001-11-22	10365	2

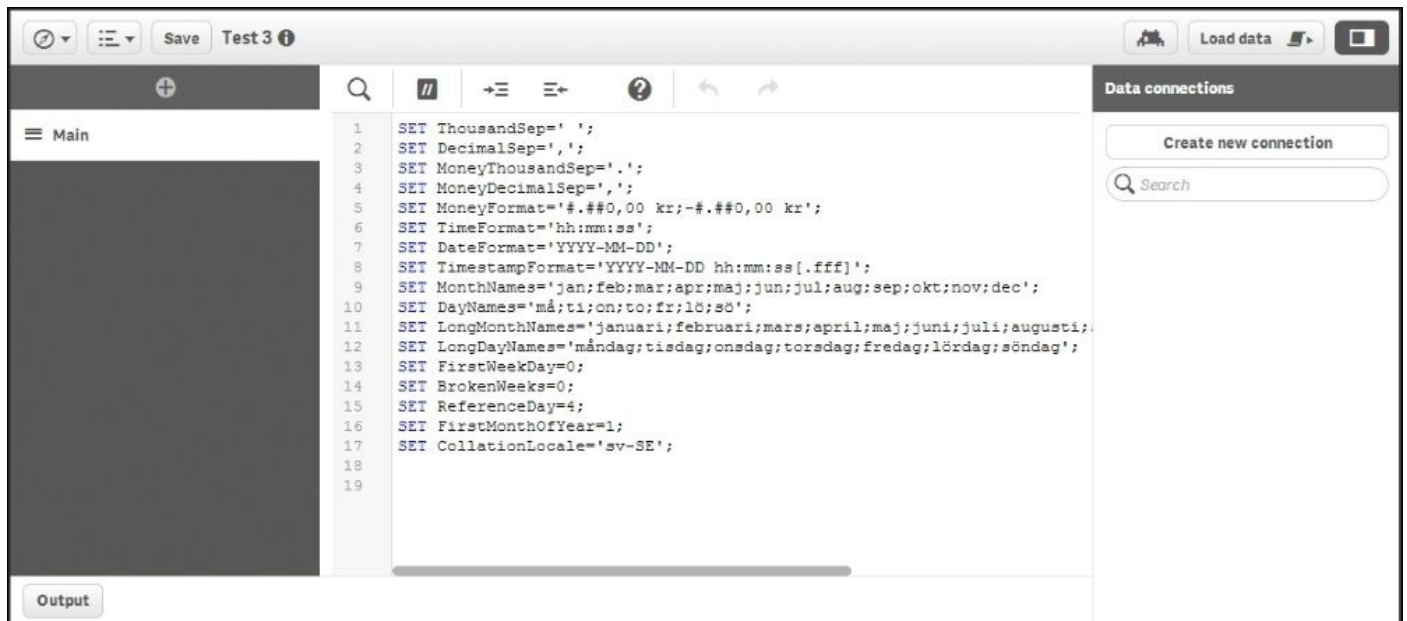
◀ Load and finish

### *The profiling wizard*

The next chapter will explain more in detail about data modeling and what you should think about when loading several tables.

# Using the Data load editor

On the screen where Qlik Sense asked you to load your data, there was a second option, **Data load editor**. Clicking on this option will open a new tab with a script editor, as shown in the following screenshot:



*The Qlik Sense Data load editor*

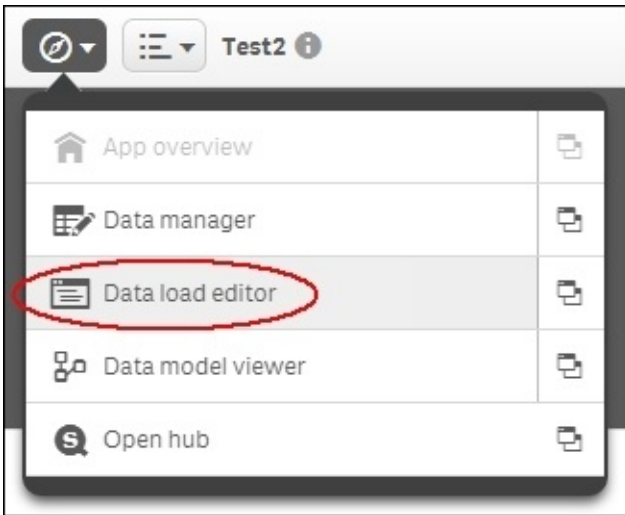
This editor is very similar to the QlikView script editor. It allows you to make very complex data transformations and basically load and transform any table. However, as with all powerful tools, it is also easy to make mistakes. Use it with caution.

When you define your data using the **Add data** command, the data load sequence will be stored in the load script as one or several Load statements. This means, these can be edited in **Data load editor** under the **Auto-generated** section, if you want to tweak them after they have been created. The section must, however, first be unlocked:



You can also create your script from scratch using **Data load editor**. If so, you must first create your data connections. These can be file folders, connections to regular databases, or connections to other data sources using other connectors.

This is how you do it: open the **Data load editor** from the initial dialog or the menu in the toolbar:

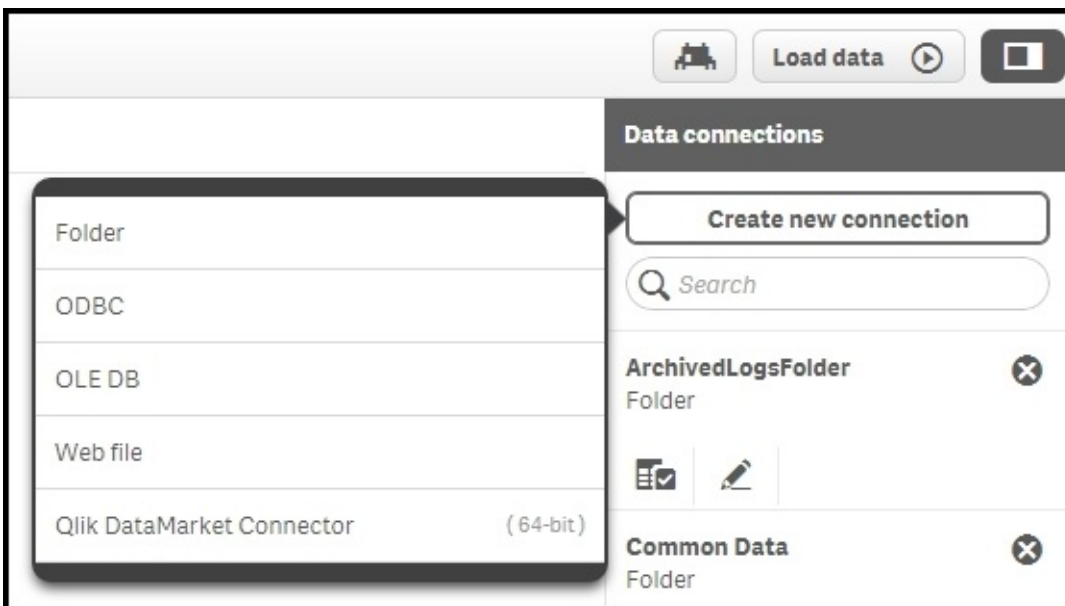


*The Data load editor command*

## Tip

Clicking on the icon to the right in the menu will open the dialog in a new tab.

Now, you will have **Data load editor** open. To the right, you have the **Data connections** panel. If you click on the **Create new connection** button, you will open a menu, where you can choose the connection type and then specify the properties of the connection in the subsequent dialog:



*Adding a data connection*

# Creating a database connection

If you want to create a database connection using **Open Database Connectivity (ODBC)**, you should choose **ODBC**. This opens the ODBC connection dialog, where you can choose the data source to be used:



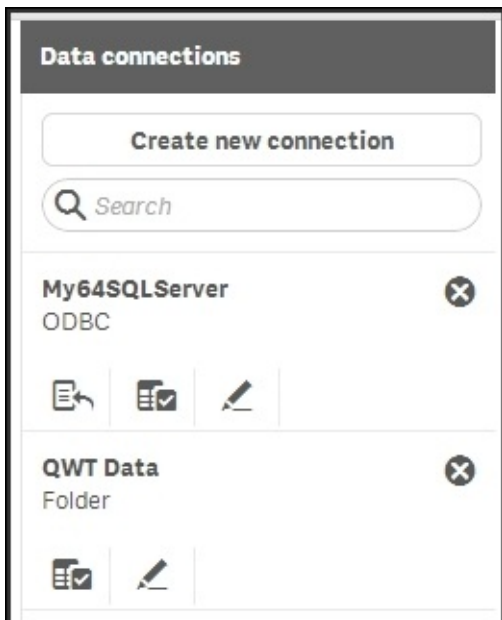
The screenshot shows a dialog box titled "Create new connection (ODBC)". It features two tabs: "User DSN" and "System DSN", with "System DSN" selected. Below the tabs are two radio buttons: "32-bit" (unselected) and "64-bit" (selected). A large text input field contains the text "My64SQLServer". Below this are three input fields: "Username" and "Password" (side-by-side) and "Name" (below them). At the bottom right, there are two buttons: "Cancel" and "Save".

*The ODBC connections dialog*

The data sources that you see are the ones defined in the Windows operating system. This means if you do the development on a server, the list is limited to those defined by the server administrator.

Once you have created these connections, you will have them displayed in a list of data connections, as shown in the following screenshot:





*The list of data connections*

Database connections, for example, the ODBC connection in the preceding screenshot, have three icons. The left one creates a Connect statement, the middle one creates a Select statement, and the right one edits the connection itself. Folder connections only have two icons. The left one creates a Load statement and the right one edits the connection itself.

Hence, to create a Load statement, you should click on the left icon for a folder connection and find the file that contains the table. This way, you can create a script in very much the same way as you would in QlikView, if you are familiar with it.

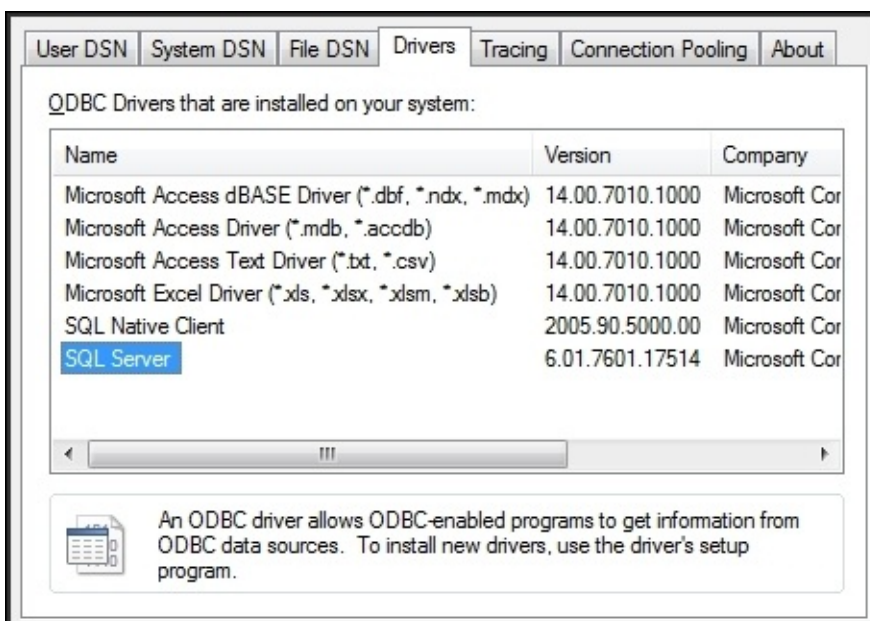
When you have created the script, you need to run it to load the data. This is done by clicking on the **Load data** button in the toolbar of **Data load editor**.

# Data connectors

As you have seen, Qlik Sense can, in addition to loading data from files, connect to databases using the ODBC and OLEDB interfaces. To see which databases you can connect to, you need to open **Data load editor** and click on **Create new connection**.

When you select **OLEDB** and then **Select provider**, you will see a list of the installed OLEDB providers. If your database isn't listed, you need to install the appropriate software from your database provider.

If you choose ODBC, you will see the defined data sources. However, you may still have drivers installed for which there are no data sources defined. To find out whether this is the case, you must open **ODBC Administrator** in Windows and look in the **Drivers** tab (as shown in the following picture). If your database isn't listed, you need to install the appropriate software from your database provider:



*The Windows ODBC Administrator*

## Tip

The default ODBC administrator is opened by navigating to **Control Panel | Administrative tools | Data Sources (ODBC)** in Windows. However, on a 64-bit OS, you may also want to use 32-bit drivers. To manage these, you need to open `C:\Windows\SysWOW64\odbcad32.exe`.

Once the ODBC driver is installed, you need to define a data source. We recommend that you do this on the **System DSN** tab in **ODBC Administrator**. When this is done, the data source will appear in the Qlik Sense ODBC dialog.

You can also use custom connectors with Qlik Sense, such as the Salesforce connector (as shown in the next picture) that you can download from the Qlik download page. These should be put in `C:\Program Files\Common Files\Qlik\Custom Data`. They will then

appear in your list of connectors:

Folder	
ODBC	
OLE DB	
Web file	
SalesForceDLL.dll	(64-bit)
Qlik DataMarket Connector	(64-bit)

*The list of connectors, including two custom connectors*



# The analysis interface—sheets and visualizations

Once you have loaded the data into Qlik Sense, it is time to create the visualizations in the analysis user interface. A basic set of sheets and visualizations should normally be supplied by the application developer, and additional ones can be created by the users themselves.

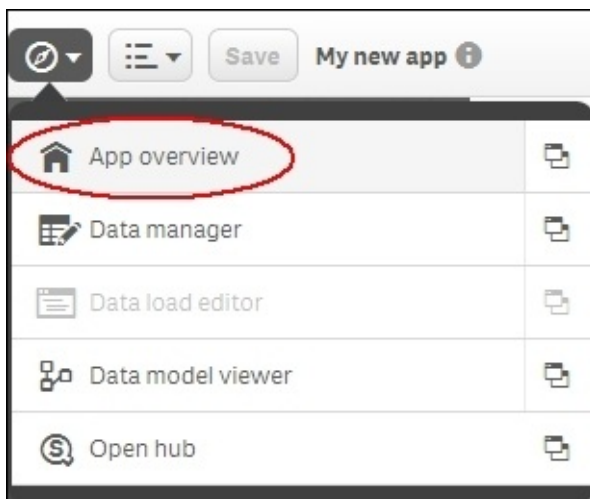
# Creating a sheet

When you have loaded the data, Qlik Sense will usually create the sheet for you, and take you there. So, if you see a big blank area with the text **The sheet is empty**, you can skip to the next section:



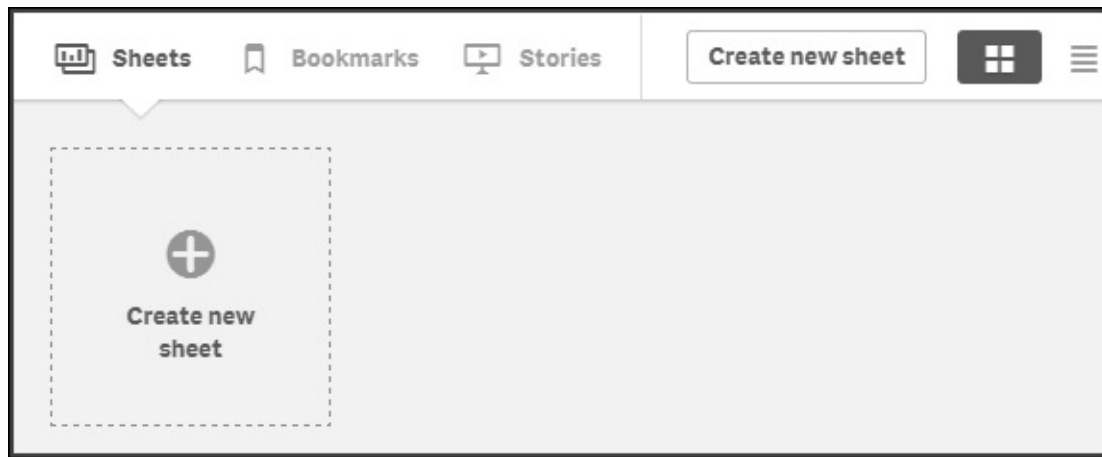
But if you are still in the Load editor, you may need to perform the following steps to create a sheet:

1. Go to **App overview** using the command in the top-left menu as shown in the next picture:



*The App overview command*

2. In **App overview**, you can create your first sheet by clicking on the sheet placeholder to the left, or on the button to the right:



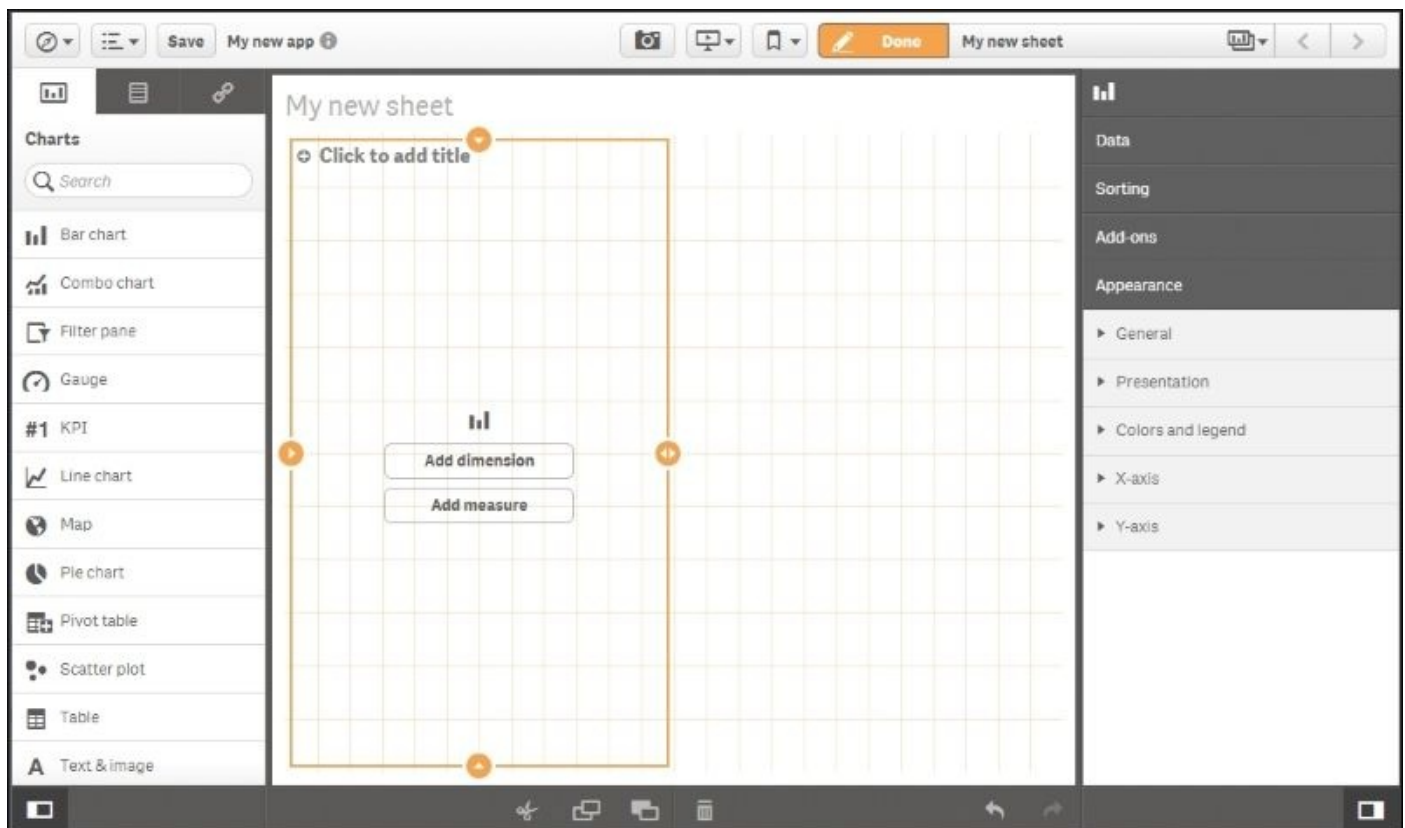
*The Create new sheet button*

3. Name it and hit *Enter*. You have now created an empty sheet and need to put some visualizations on it.
4. Click on the newly created sheet.

# Adding visualizations

At this stage, you are probably looking at an empty sheet with the text **The sheet is empty** located in the middle.

Click on the **Edit** button to the right in the toolbar to start adding things. Doing so will open the Assets panel to the left listing a number of object types: **Bar chart**, **Combo chart**, **Filter pane**, and so on. Now, you can drag and drop an object type onto your sheet, thereby creating such an object. If you, for instance, drag a bar chart onto the sheet, you will create an empty bar chart:

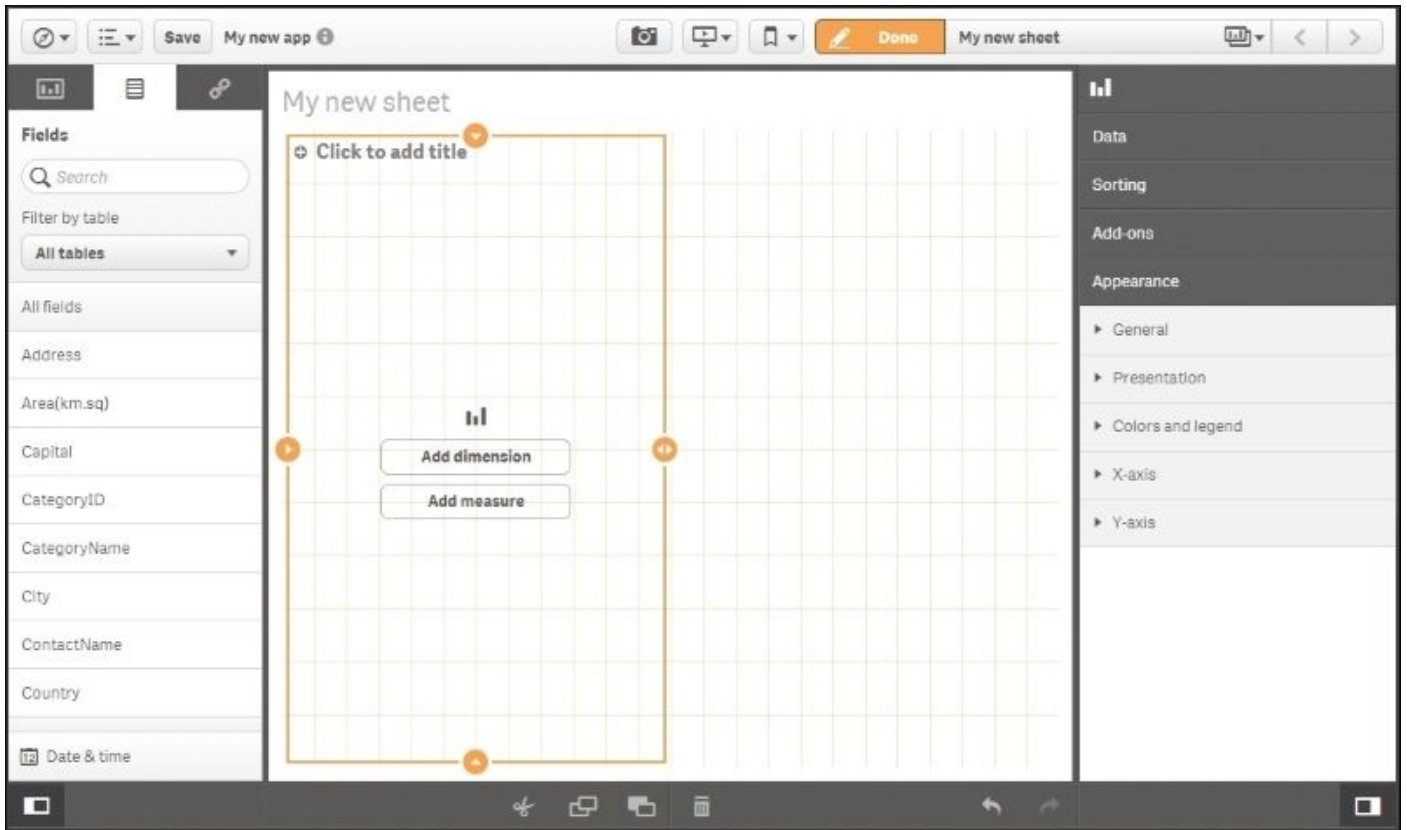


Depending on where you drop it, it will use all of the sheet or just half the sheet. Move the object around before you drop it, and you'll see. You can also adjust its size at a later stage.

Once you have dropped it, the bar chart will clearly show that it needs a dimension and a measure in order to display properly. You can click on the buttons on the bar chart to define these, but you can also use the Assets panel on the left.

The Assets panel shows object types, but if you look carefully, you will see that there are three tabs at its top—one for object types, one for fields, and one for the predefined library entities. So, if you click on the middle icon, you will see a list of fields that can be used as dimensions or as measures:





*The Assets panel now shows a list of fields*

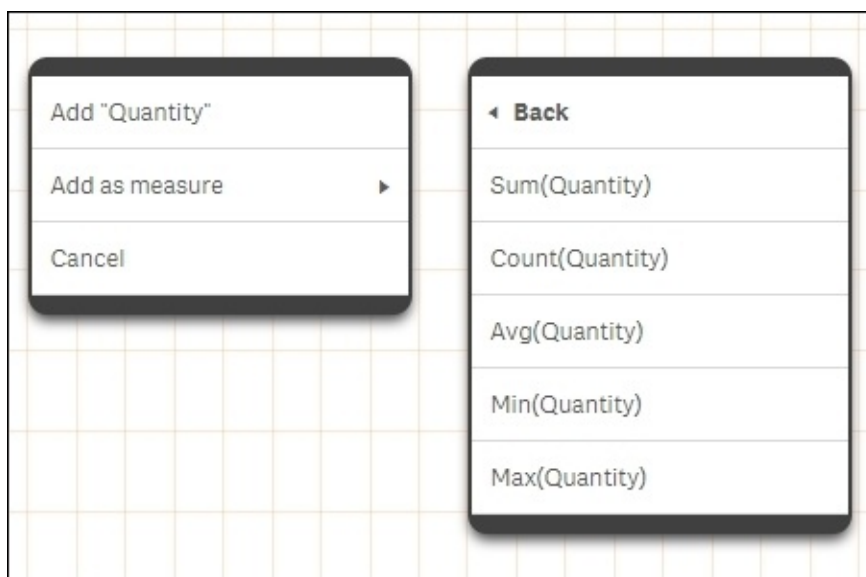
# Adding dimensions and measures

You can now drag and drop fields onto the bar chart, thereby creating the dimension and the measure.

A dimension is a field with discrete values, for example, customer, product, or month. A chart will create one number per dimensional value; hence, a bar chart will create one bar per distinct value in the chosen field.

A measure is usually a number, for example, sum of sales or number of orders, and this will constitute the height of the bars.

When you drag a field onto the empty bar chart, Qlik Sense will ask you what you want to do with this field. You can add it (as a dimension), or you can use it inside an aggregation function (Sum(), Count(), or Min()) to form a measure:



*Adding measures*

# Defining bar charts

When you have added both dimension and measure, the bar chart will appear as shown in the next screenshot. To the right, you will have the properties of the bar chart, where you can set its properties—the sort order, the colors, and so on. You can also define the dimension and the measure directly in the chart properties:



To see the final result, you need to click on **Done** in the toolbar, which takes you back to fullscreen.

Clicking on the **Save** button in Qlik Sense Desktop will save the application as a file with the extension .qvf in the application folder (C:\Users\  
<user>\Documents\Qlik\Sense\Apps). The file contains both data and script and it can be imported to other Qlik Sense installations. However, you may need to adjust the script so that it runs from the new location.

When you are done with the bar chart, you should click on **Save** and start creating your next visualization.

In the Server version of Qlik Sense, you don't have a **Save** button. The changes are saved automatically.

# Storytelling

An exciting feature in Qlik Sense is **storytelling**. Storytelling is basically a presentation mode, where you can first prepare a presentation—like a slide show—and then present it. Storytelling is also an excellent way to present an application and create an overview of its content.

When you create an application, you can—in addition to the normal application development of course—also create a story that can be used by anyone who uses the application. However, we believe that the more common use case is that stories will be created not by the application developer, but rather by contributors—power users who choose to add elements to the application. Hence, storytelling is described in [Chapter 4, \*Contributing to Data Discovery\*](#).



# The application library

As previously mentioned, the Assets panel can show object types and fields. However, it has a third tab for predefined library entities. If you click on this tab, you will see the application library:



The library contains entities that have been predefined and that can simplify the Qlik Sense usage for a business user. Dimensions, measures, or entire visualizations can be stored in the library.

You do not *need* to use the library—nothing has to be predefined for Qlik Sense to work. However, if you want to reuse formulas or you have a situation where your task is to deliver an app to a business user, it is a good idea to use the library.

## **Which fields should be exposed?**

Often, you have many fields in an app, of which maybe only a few should be exposed as dimensions. Then, you should use the library to define the fields that are appropriate as dimensions, and name them in a way that they can be easily understood.

A dimension can also be a group of fields that is exposed as a drill-down group. It can also be a formula using an `Aggr ( )` function that defines an array of values. In both these cases, it is a good idea to define the dimension in the library.

# Defining KPIs

Measures are formulas that define KPIs and other numbers relevant for the analysis. These are often used in multiple places in an app, so it is convenient to store the definition in one place only. Then, should you want to change the definition, you need to do it in the library only. Also, this is a way to ensure that there is only *one version of the truth*.



# Creating library entries

Library entries can be created in several different ways. The most obvious way is to enter the library and click on the **Create new** button.

An alternative way is to do it from **Data model viewer**. Here, you can mark a field and click on the **Preview** button (in the bottom-left corner of the screen). You have the option of adding the field as dimension or measure.

Whichever way you choose to use when you create your dimension, you will see the following dialog where you define your dimensions:

**Create new dimensions**

Single  Drill-down

Show fields from table:

All

Search

CategoryName

City

ContactName

Country

Customer

CustomerID

Day

Description

Field:

Country fx

Region fx

Customer fx

Name:

Description:

Tags:

Add dimension

Done

*The Create new dimension dialog*

When you create a measure, you will see the following dialog. Make sure that you have an aggregation function, for example, `Sum()`, `Count()`, or `Min()`, wrapping the field reference:

### Create new measure

Expression:

Name:

Description:

Tags:

Cancel Create

*The Create new measure dialog*

Visualizations can only be entered into the library by the use of drag and drop, so you need to have created the visualization on the sheet first.



# Best practices in data visualization

In the new world of ever increasing data volumes, the ability to visually communicate insights from data is an important skill set. Both the structure of an app and the chosen visualizations affect how data is perceived.

An app can contain many sheets, and the order of the sheets and what you put in them is the first consideration. The best practices can be summarized in three words: **dashboard**, **analysis**, and **report (DAR)**.

# Dashboard

The dashboard is where the high-level perception takes place. It is usually the first sheet or the landing page, and it should give just the most important information and have the least amount of interactivity. Its main purpose is to help users get an overview and scan for status changes. The users can see at a glance whether things are working or not. It's a starting point, such as a table of contents; the user gets an idea of what is available and then heads off to other parts of the app based on what they have seen.

Some advice on dashboards:

- Display data only on a high level
- Don't use too many KPIs
- Use no or very few filtering options
- The most important information should be large

# Analysis

The analysis pages should be more interactive: they should help users explore the data and look for answers to questions they may have formed on the dashboard page. Analysis pages are where the user spends time and interacts with the data. Typically, each sheet has a theme and covers a specific aspect of the business. The analysis pages are where the user learns from data.

Some advice on the analysis pages:

- Allow filter panes or listboxes to facilitate selections
- Make sure an entire page is about a particular topic
- Use graphs

# Reporting

The third type of sheet is the reporting sheet. It is not always necessary to have these, but often it is advantageous to gather the most granular information on dedicated sheets. This is the purpose of the reporting sheets: to give the most granular information with tabular data. This is where a user can spend a lot of time sorting and filtering through the details.

Further, users sometimes want to export or print data, and the starting point is often a reporting sheet. All Qlik Sense objects can be printed or exported, either as images, as data, or bundled in a PDF document.

Some advice on the reporting sheets:

- Display transactional data in tabular form
- Give the users the ability to view every detail, so they can assess what actions they need to take

# Structuring a sheet

The structure within a sheet is also important. When you create an app, it's your job to prioritize information and display it in such a way that users better understand the data and find their way in the app.

The human eye scans most content in an **F** pattern. The first time we see a page, we read the first line, then a bit of the second line, and then work our way down the left-hand side of a page looking for keywords. This means the content at the top of the page is the most important real-estate on a page, especially the top-left. The top of the sheet gives the users an idea of what content a page may contain and the scent of whether or not they are on the right track to finding what they are looking for. So, label the sheets appropriately.

It is also important that users easily find objects they are looking for. This applies to filter panes and listboxes, where the user makes selections. If used in several sheets, they should be placed in the same place in all sheets, if possible. Further, given how a human eye scans the page, these objects should preferably be placed to the left.



# Graphs and other visualizations

Visualization also includes choosing appropriate graphs. Getting the graph right is important; otherwise, the data can be misinterpreted. There are several highly regarded thought leaders who have written excellent reading material on this topic, for example, Edward Tufte and Stephen Few. If you have not read any book in this area and you intend to build business intelligence applications, we recommend that you do this. It will help you in your work.

# Dimensions and measures

Dimensions and measures are sometimes confused for one another, but it is really quite simple. You should start by asking yourself, “*What* do I want to show?”. The answer is usually sales, quantity, or some other number. This is your **measure**.

The second question you should ask yourself is, “*How many times* should this measure be calculated?” or “*Per what* do I want to show this measure?” The answer could be once per month, once per customer, once per supplier, or something similar. This is your **dimension**.

The dimension and the measure of a chart are the core of the visualization and often indicate what visualization to choose. It is important to understand which type of field is used as a dimension. For example, when showing trends over time, you should usually use a line chart or a bar chart. The same is also true for any dimension with an implicit, intrinsic order. By the same token, you should never use a line chart unless the dimension has an implicit order.

The fields used as dimensions can be classified into the following groups:

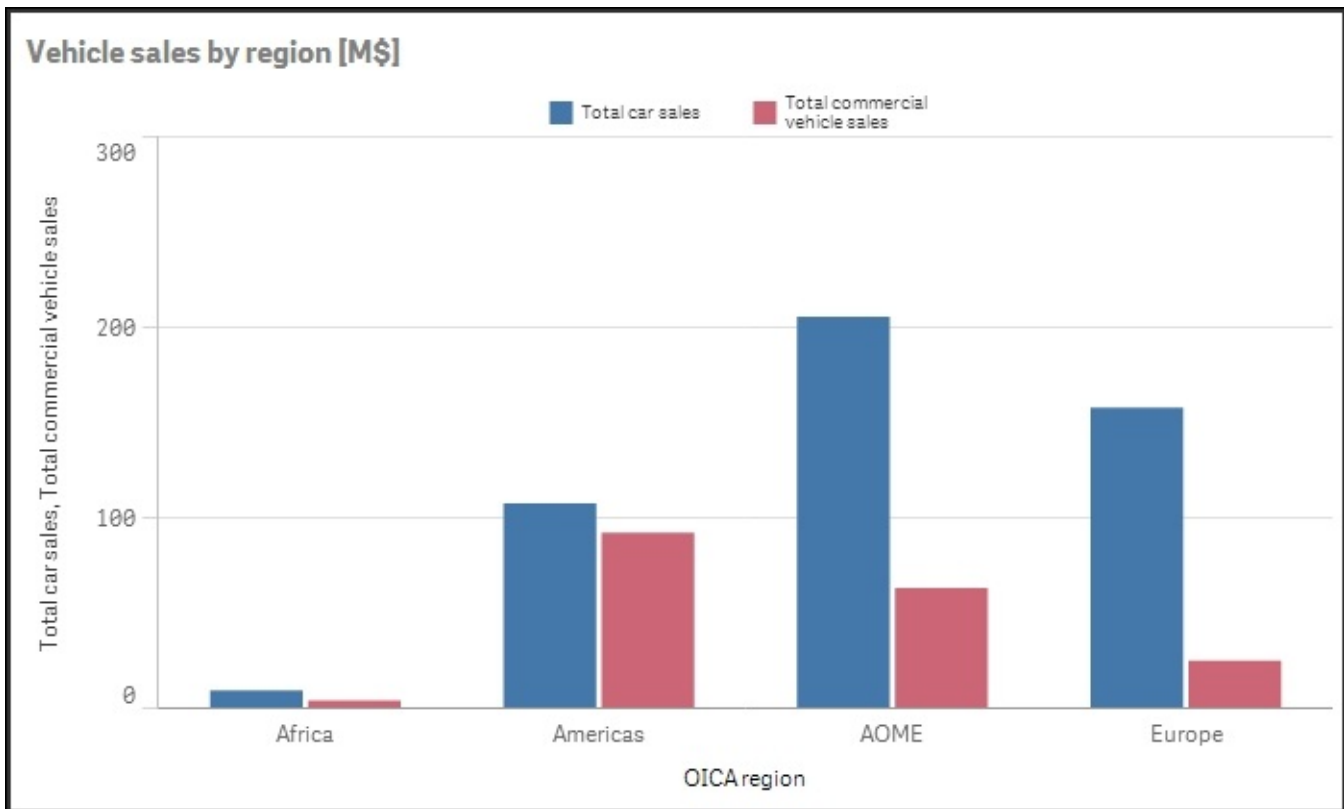
- **Nominals:** These are fields with qualitative values that *lack* intrinsic order, for example, product, customer, color, gender, and so on.
- **Ordinals:** These are fields with qualitative values that *have* intrinsic order, for example, ratings such as unsatisfied, neutral, or satisfied; that is, fields that have no numeric meaning.
- **Intervals:** These are fields with quantitative values that *lack* a natural zero. They are like coordinates, for example, date, time, longitude, temperature, and so on.
- **Ratios:** These are fields with quantitative values that *have* a natural zero. They are like amounts, for example, quantity, sales, profit, weight, and so on.

With this classification in mind, it is easier to describe what you can and cannot do with some graph types.

## The bar chart

The most common visualization is the bar chart. Bar charts can be used for almost any dimension, and it is easy to compare the sizes of two bars. Further, they are good for ordinal data, since the intrinsic order can be used. This also means that trends over time can easily be spotted.

If a second measure is added to the bar chart, you will get two bar series so that you can make comparisons both between the measures and along the dimension. For example, in the following chart, you can compare not only the sales regions, but also the commercial vehicle sales with total sales:



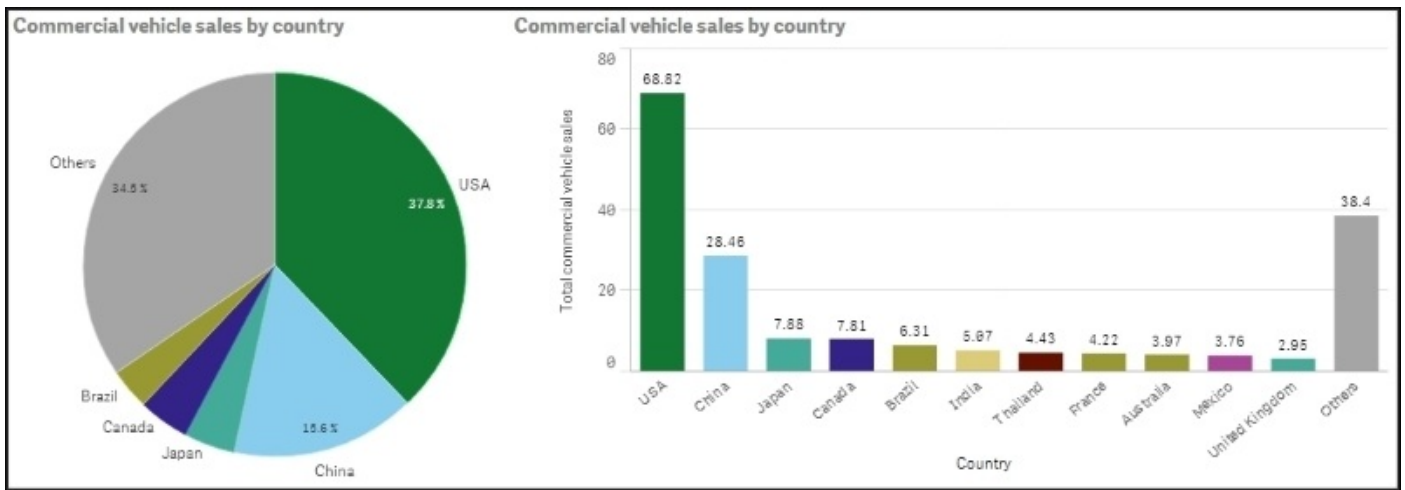
A bar chart is often the best visualization. By adding a second measure or a second dimension, you can get several series of bars.

## The pie chart

The next visualization is the pie chart. This should only be used if the dimension is of the nominal type and you want to display the relative proportions. Pie charts are not good for ordinal data, since the order of the dimensional values isn't obvious.

Pie charts are, by some experts, considered a poor visualization, and a bar chart is indeed often a better alternative since it conveys the information more efficiently.

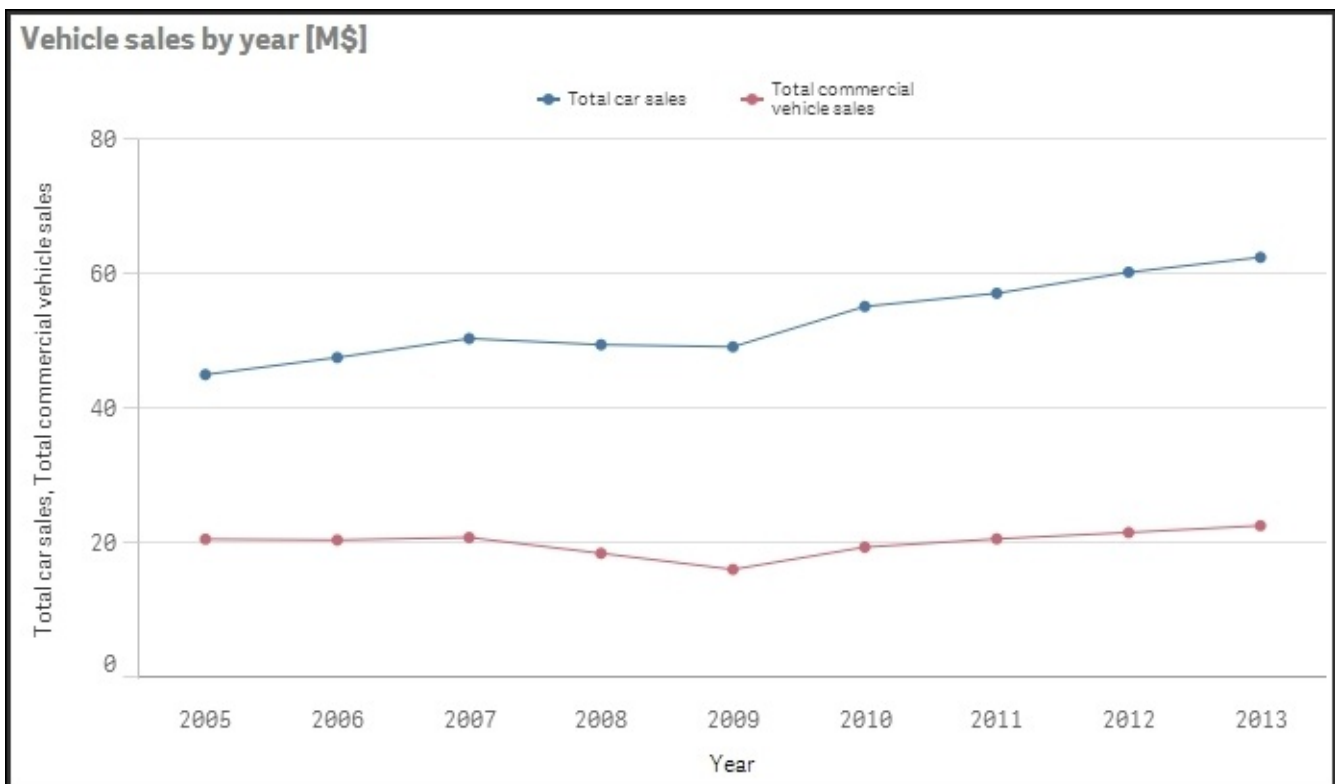
But pie charts are still useful to convey an overview of the relative sizes. For instance, in the following pie chart, you can clearly see that the combined sales in the USA and China constitute more than 50 percent—something that is not at all clear if you instead look at the corresponding bar chart:



Hence, a bar chart is often easier to read, but, in this case, the pie chart is better at showing the relative proportions of the largest countries. However, it can sometimes be hard to judge the relative sizes of the slices in a pie chart if there is only a small difference between them. Then, a bar chart is a better choice.

## The line chart

The next visualization is the line chart. This should only be used if the first dimension is of the ordinal or interval type. Line charts are particularly useful for showing a change over time. Several lines can be used, either using a second dimension or by adding measures:



Line charts are good when you want to analyze trends over time.

## The KPI object

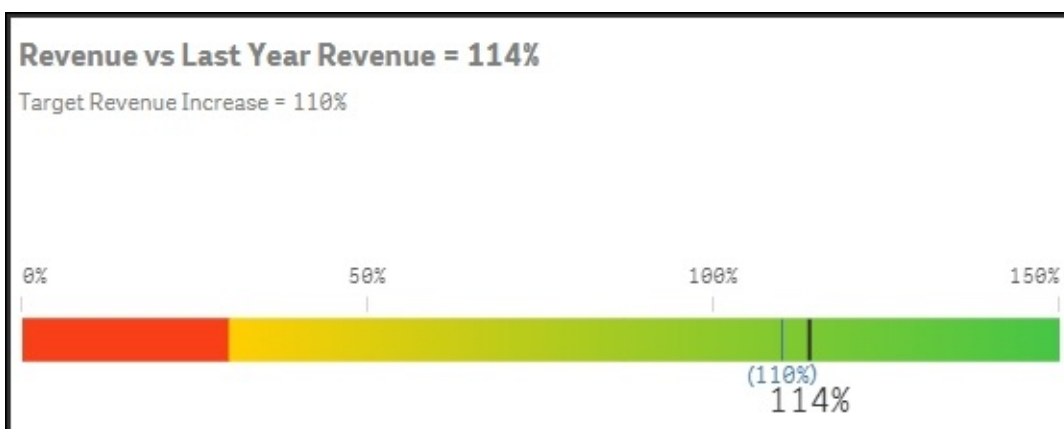
A new object in Qlik Sense is the KPI object. With this, you can display a measure or a KPI on a dashboard and label it so that it is clear what it is:



This allows you to create an overview so the user quickly and easily sees the main information. But you cannot show the number over a dimension, for example, per month or per customer, since there is no dimension in this object. It is just a number that is valid for the entire selected dataset.

## The gauge

The gauge object is similar to the KPI object, in that it shows a measure and has no dimension. It should be used the same way as the KPI object—to create an overview.



The difference with a KPI object is that a gauge can also show a basic graphical element—either a radial display like a speedometer or a linear display like a thermometer.

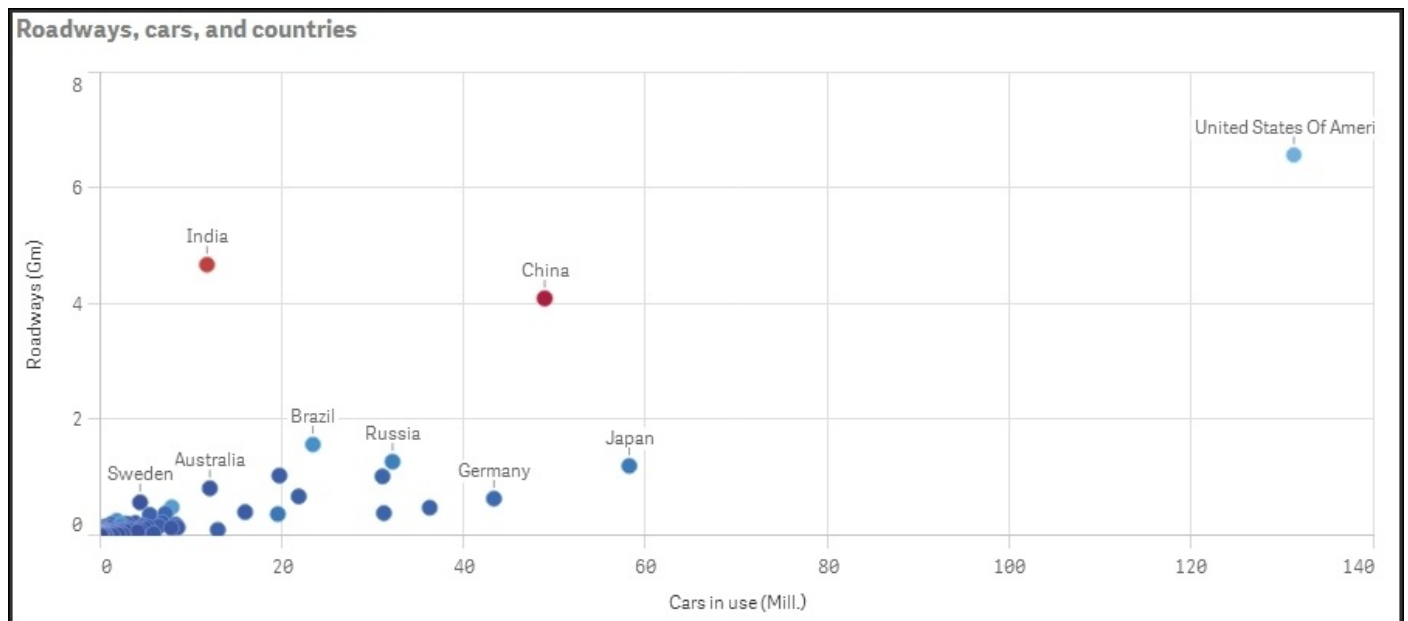
## The scatter chart

Scatter charts are good if you want to compare two different quantitative measures for the same dimension, that is, pairs of data per some dimension. Such plots are useful to find clusters of values, linear arrangements, gaps, and values that are much different from the norm. These are the kinds of patterns that are meaningful in correlation relationships.

The unique strength of this chart type is its ability to encode values along two quantitative scales used as two axes. Note that the logical dimension is *not* used as an axis. Instead,

two *measures* are used as axes.

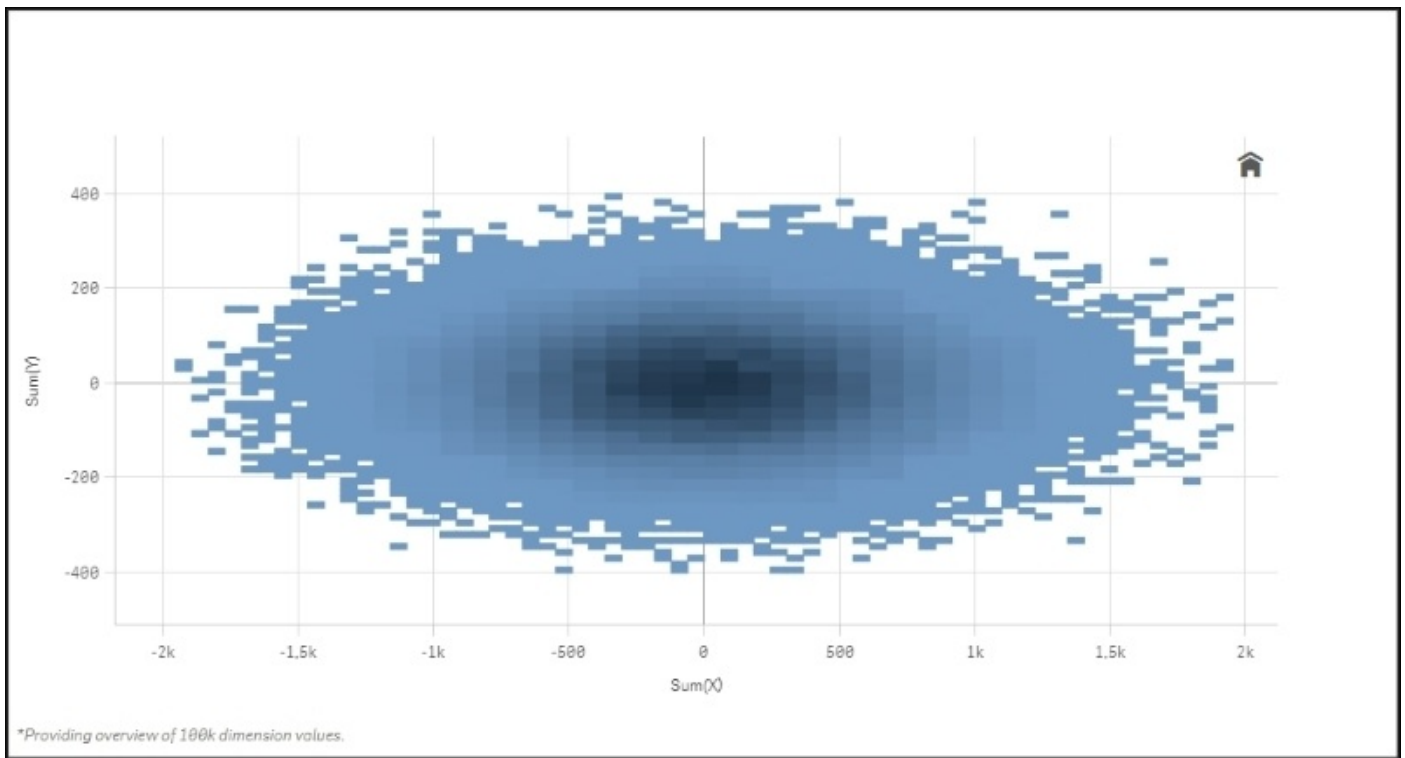
The dimension, which should normally be of the nominal type, defines the number of points in the scatter chart:



The preceding scatter chart shows the amount of roads per country (in million kilometers) versus the number of cars (in million units). The color indicates the country's population.

If you have a scatter chart with large amounts of data, Qlik Sense uses an algorithm to create an overview of the data, as shown in the following screenshot—it pixelates the data points and color codes the density of the data points.

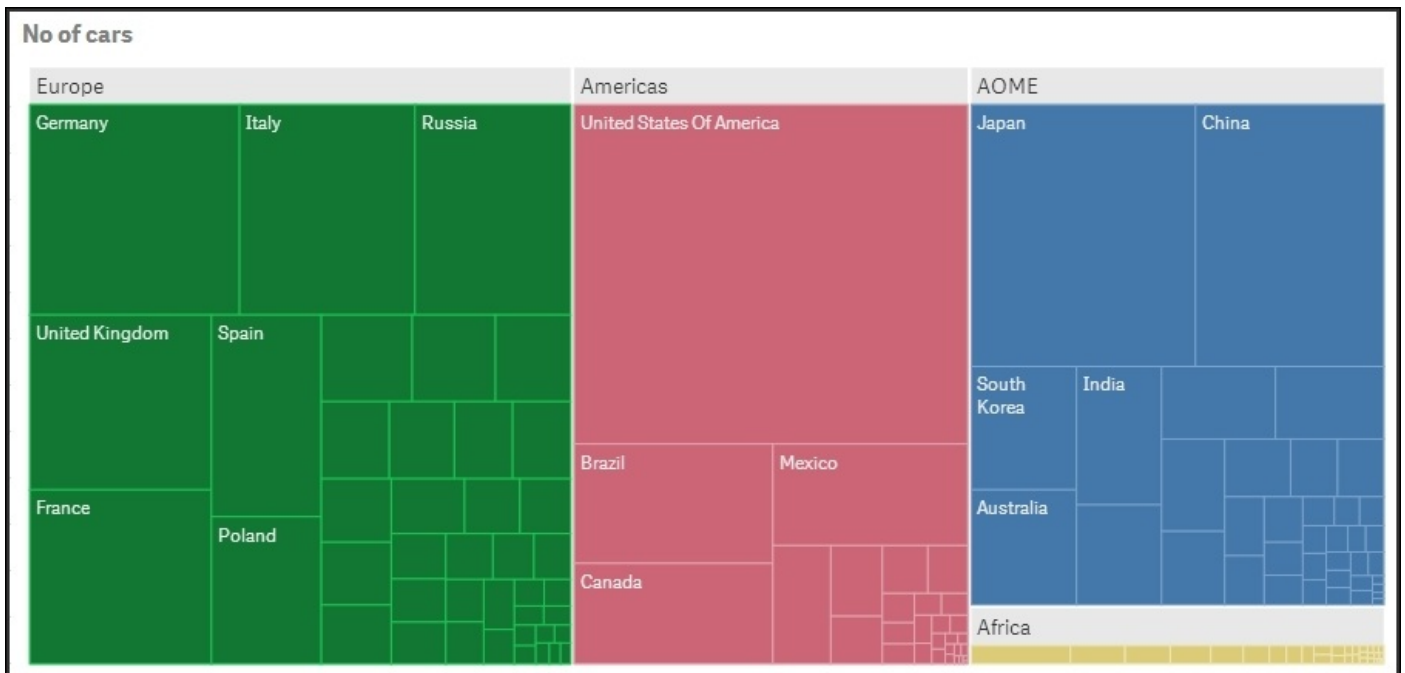
Not only does this result in a better overview of a large number of data points, it also minimizes the amount of information that needs to be transferred from the server to the client.



However, if you drill down in the chart by zooming or making selections so that the number of data points is reduced to less than 1,000, the data will be shown as individual data points.

### **The tree map**

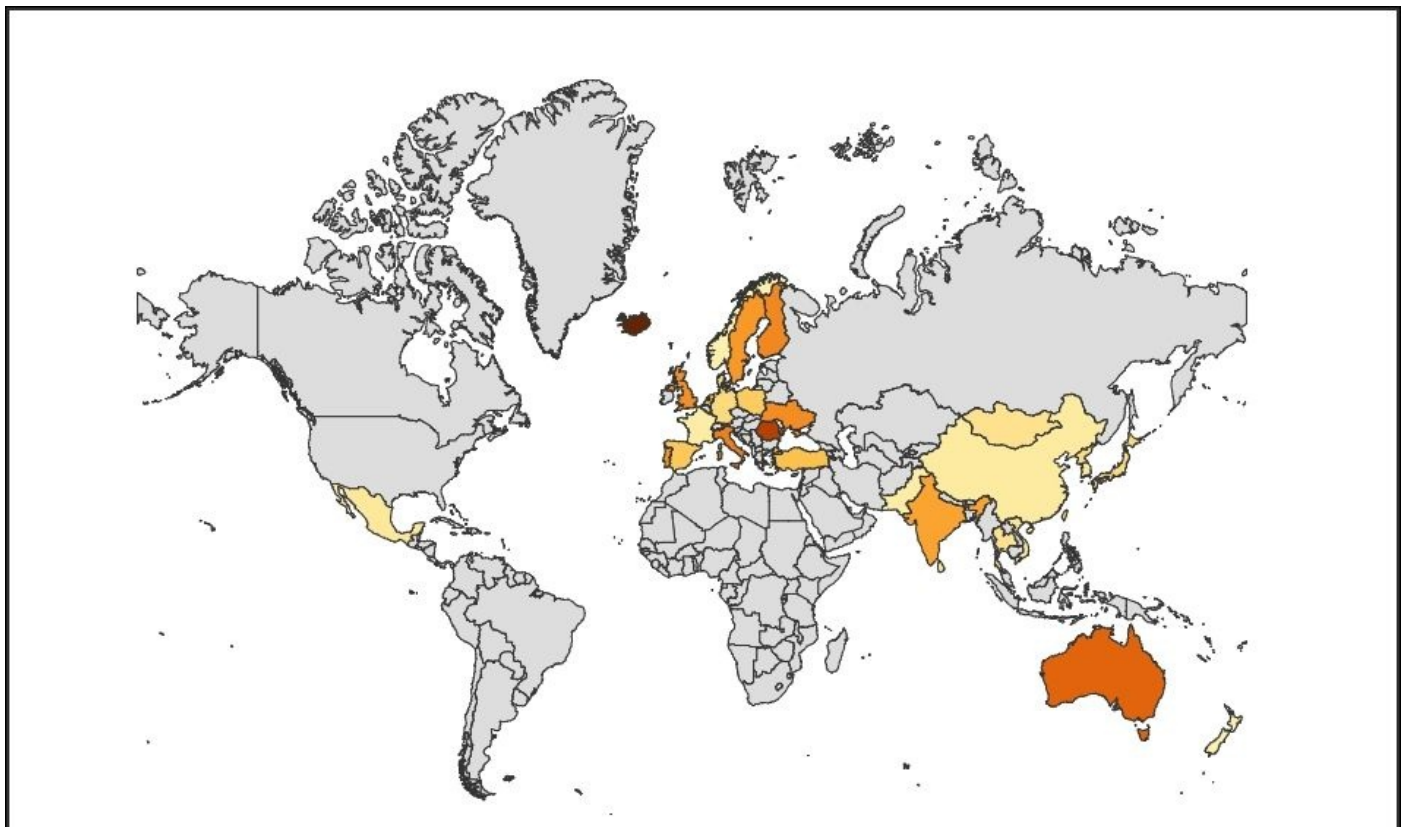
Another chart type to mention is the tree map (called block chart in QlikView). This is an excellent visualization if you have two or more dimensions and one single measure. The reason why it is called tree map is that it was originally designed to display hierarchical relationships that have a structure like a tree. In Qlik Sense, you can use it for nonhierarchical relationships that have no resemblance with trees whatsoever. Again, the dimensions should be of the nominal type:



*A tree map showing the number of cars in different countries*

## The geographical map

The last chart type to mention is the geographical map. This visualization allows you to display regions and areas on a map of a country or region. Also, here the dimension is of the nominal type:





In Qlik Sense, you can connect maps to the data. Just as all other charts, a geographical map can be used for both input and output. Selections in other objects will affect how the geographical areas are displayed, and selections can be made directly in the map.

## Tables

There are two table types in Qlik Sense—the standard table and the pivot table. They are similar to each other in that, just as charts, they can have both dimensions and measures. But they are slightly different in appearance and capabilities.

The pivot table is excellent for grouping data and showing it in a compact way. Dimensions can be used both as rows and columns, and rows can be collapsed into the above dimension:

Sales Rep ▾  Product Sub Group ▾	Year ▾		Measures	
	2006		2007	
	Budget \$	Actual Amount	Budget \$	Actual Amount
⊕ Edward Smith	\$41,793	\$114,923	\$47,363	\$56,069
⊖ Brenda Kegler	\$625,729	\$1,149,330	\$101,128	\$199,230
Oysters	\$105,807	\$116,781	\$17,709	\$38,121
Cheese	\$414,192	\$729,785	\$17,709	\$38,121
Bologna	\$264,974	\$425,666	\$83,419	\$161,109
Cereal	\$137,468	\$129,823	-	-
⊕ Max Blagburn	\$1,620,736	\$465,953	\$1,407,075	\$749,431
⊕ Teresa Lynch	\$476,841	\$166,425	\$139,496	\$108,195

The standard table is different. All data is displayed in one long table, and its advantage is that you have full freedom in how to sort the data any way you want. In addition, you can make searches directly in the table:

Sales Rep 🔍	Product Sub Group 🔍	Year 🔍	Budget \$	Actual Amount
<b>Totals</b>			<b>2925368.24</b>	<b>2669198.85</b>
Brenda Kegler	Bologna	2006	264974.49	425666.36
Brenda Kegler	Bologna	2007	83419.39	161109.44
Brenda Kegler	Cereal	2006	137467.76	129823.25
Brenda Kegler	Cheese	2006	414192.45	729785.03
Brenda Kegler	Cheese	2007	17708.5	38120.57
Brenda Kegler	Oysters	2006	105806.65	116781.42
Brenda Kegler	Oysters	2007	17708.5	38120.57
Edward Smith	Bologna	2006	41792.8	114922.91

Both table types are good for showing details of the data, but to give the user an overview, a graphical visualization is better.

## Sorting and colors

Once you have chosen the appropriate chart type, you should choose the appropriate chart settings, for example, scale, sorting, and appearance. Nominals should be sorted alphabetically or by some form of measure, for example, the size of the measure. The other types should be sorted according to the intrinsic sort order.

You also need to label the chart, for example, add a title, descriptions, y axis units, and so on.

Finally, you should also make sure to use the appropriate colors. But be careful here... bright colors are beautiful, but when it comes to data visualization, it is best practice not to use highly saturated colors. Instead, it is good to tone it down a bit. The main reason is that lighter colors are much easier on the eyes, so they show data better, for example, when displayed on large screens.

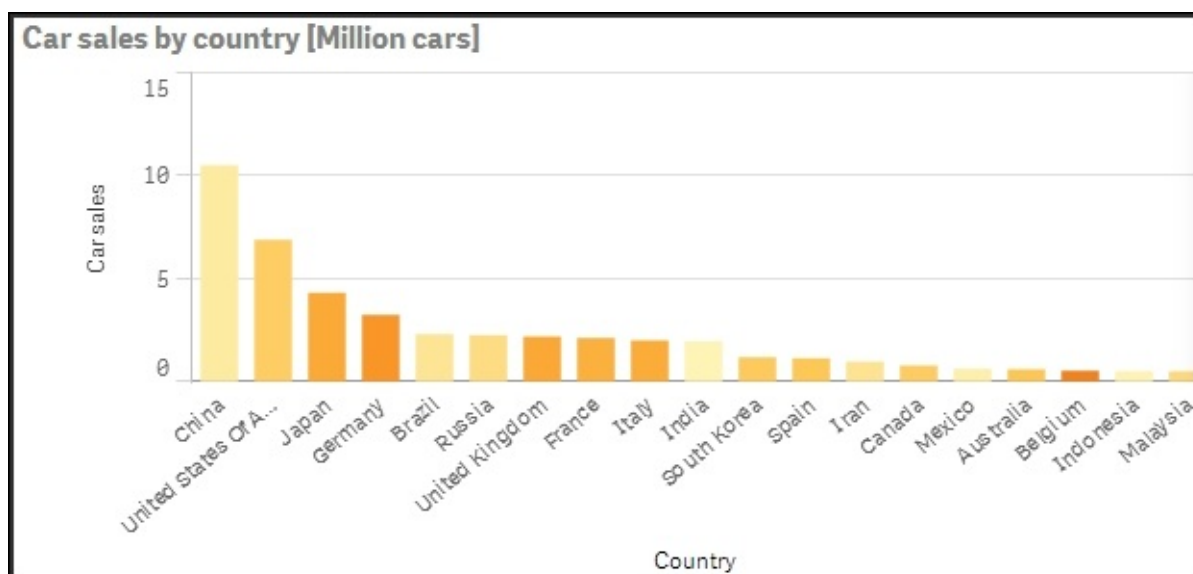
Further, bright colors draw attention, so they should only be used to highlight a specific field value or condition.

## Tip

The color should *never* be a decoration only. It should *always* carry information, one way or another.

One way is to use the same color everywhere for the same dimensional value. This way, the user will easily identify the corresponding bars or slices in multiple visualizations.

Another way is to use a color that has an intensity that is in proportion to a specific calculation. For instance, in the following bar chart, the bars indicate the total number of sold cars in different countries and the color intensity indicates the number of sold cars per capita. This is one good way of using colors.



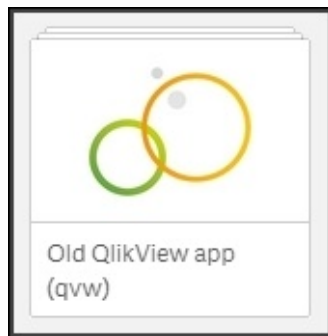




# Migrating applications from QlikView® to Qlik Sense®

All QlikView applications since QlikView Version 8 can easily be migrated. However, the conversion is only partial. The data and the script will be converted, but nothing from the layout is used. Perform the following steps to migrate applications:

1. Move the QlikView app—the .qvw file—to your Qlik Sense Desktop app folder at C:\Users\\Documents\Qlik\Sense\Apps. The file will then appear in your Desktop hub as a legacy app. Its name will have (qvw) after it:



2. You can now open the app and see the data model and the existing script.
3. Once you make changes, you will need to save these. This is when the conversion takes place. The old app with the new changes will be saved into a file with extension .qvf, and the old .qvw file will be renamed to \*.qvw.backup.

## **Changes to the script**

You might also need to make changes to the script. The structure of the script can remain the same, but all references to databases and files should be changed. In Qlik Sense, you need to use the data connections library. Hence, you must create the library entries that you need, and then replace connect strings and file references with references to the library.

# Changes to the user interface

The modern layout in Qlik Sense with responsive design has very little to do with the old static layout in a QlikView app. A consequence is that you will need to recreate all objects—all charts, lists, and tables. In fact, you will often have to rethink your entire layout. If you are deploying your apps in an environment where users have tablets or smartphones, you would probably have had to do this anyway, since the old style QlikView apps display poorly on small screens.

The best way to do this is to have both Qlik Sense and QlikView displaying the same app simultaneously. Then you can go through the app sheet by sheet and decide how to design your new Qlik Sense app.

The syntax for formulas has not changed, so it is advisable to use copy and paste when moving complex formulas from QlikView objects into Qlik Sense objects.

Currently, there is no support for migration in the Qlik Sense server. So, if you want your old app on a server, you need to first convert it using Qlik Sense Desktop and then import the new file to the server.

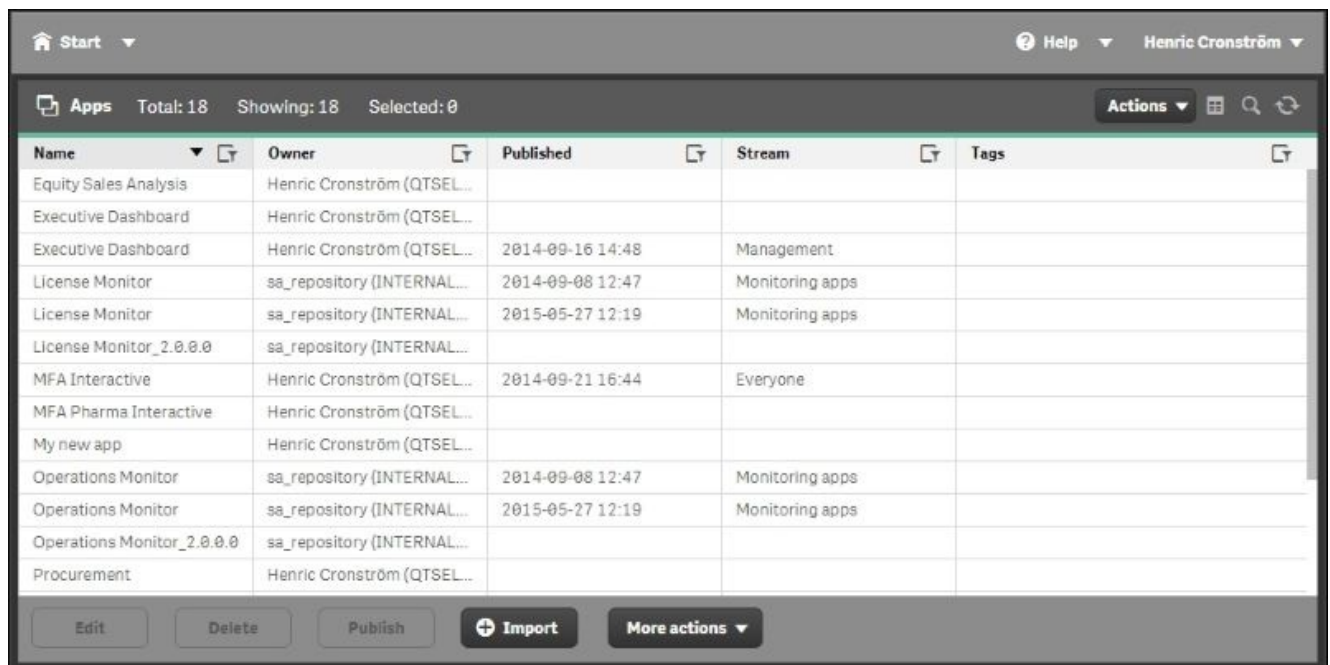




# Publishing your apps

When you have created an app, the next step is to make it available for other users. Perform the following steps if you have developed your app using Qlik Sense Desktop:

1. Import the app using the **Apps** sheet in the Qlik Sense Management Console. There, you will find an **Import** button at the bottom:

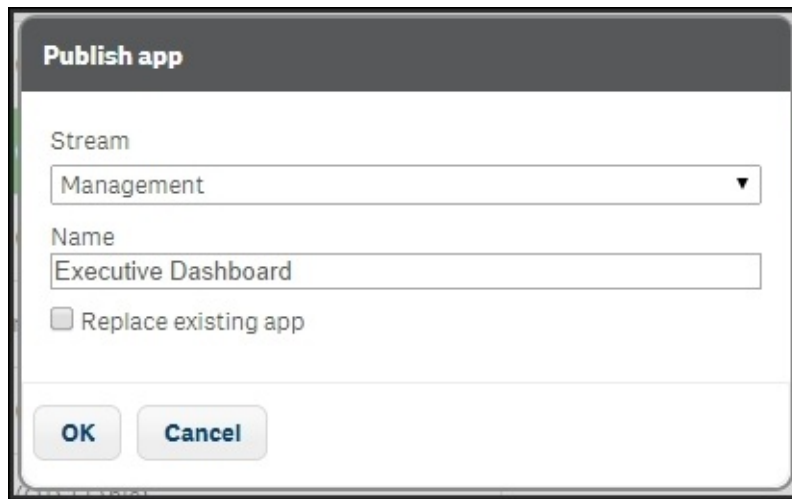


2. Once the file is imported, you may also need to assign the correct owner: mark the file, click on **Edit**, and change the owner, if necessary. The file will then appear under **My Work** in the Qlik Sense hub.

## Note

If you have developed the app using the Server version, it will already be under **My Work**.

3. Before you publish the app, it is recommended that you make a copy of the file. Mark the file on the **Apps** page in the Qlik Sense Management Console and select **Duplicate** in the **More actions** menu.
4. Now, you are ready to publish. You can publish the app by marking the file and clicking on the **Publish** button.
5. Publishing means that you move an app from your personal workspace to a stream, which means the file can be accessed by many people. The properties of the stream, including security rules, will then be applied. Note that the file will be removed from your personal workspace. This is the reason why it is a good idea to make a copy of it.
6. When publishing, you will be prompted to choose a stream for your application:



The image shows a dialog box titled "Publish app". It contains a "Stream" dropdown menu with "Management" selected, a "Name" text input field with "Executive Dashboard" entered, and a checkbox labeled "Replace existing app" which is currently unchecked. At the bottom of the dialog are two buttons: "OK" and "Cancel".

7. Currently, there is no way to “unpublish” an app. If you want to remove it from a stream, you need to delete the app.



# Summary

In this chapter, we looked at the functions and commands you need to know to build engaging applications—both from a process perspective and, more practically, what you need to do to load the data and create an inviting user interface.

In the next chapter, we'll move into the basics of data modeling, which is an area you need to master in order to make advanced applications with multiple source tables.



# Chapter 6. Building Qlik Sense® Data Models

In the previous chapters, we looked at how to create a definition of which data to load. But we did not look at which considerations you should make on how to load and link different tables.

So, in this chapter, we will discuss the following topics:

- Data modeling
- How a data model reflects business processes

# The QIX engine

The QIX engine is the core of the logic in Qlik Sense. All evaluations and calculations are made by this engine.

Every user selection implies a new logical situation. Other field values than the ones used earlier are possible; summations need to be made, so the charts and the KPIs get different values than what we had before. Everything needs to be recalculated, and the data model defines how this is done.

When the user makes a selection, Qlik Sense first evaluates which field values of other fields are possible and which rows are possible. In this evaluation, the key fields are used to propagate the selection to other tables. This is *the logical inference*. The second step is to calculate all formulas based on the possible rows. In this step, all *aggregations* are made.

The data model defines *how* the QIX engine's logical inference and calculations will be made.



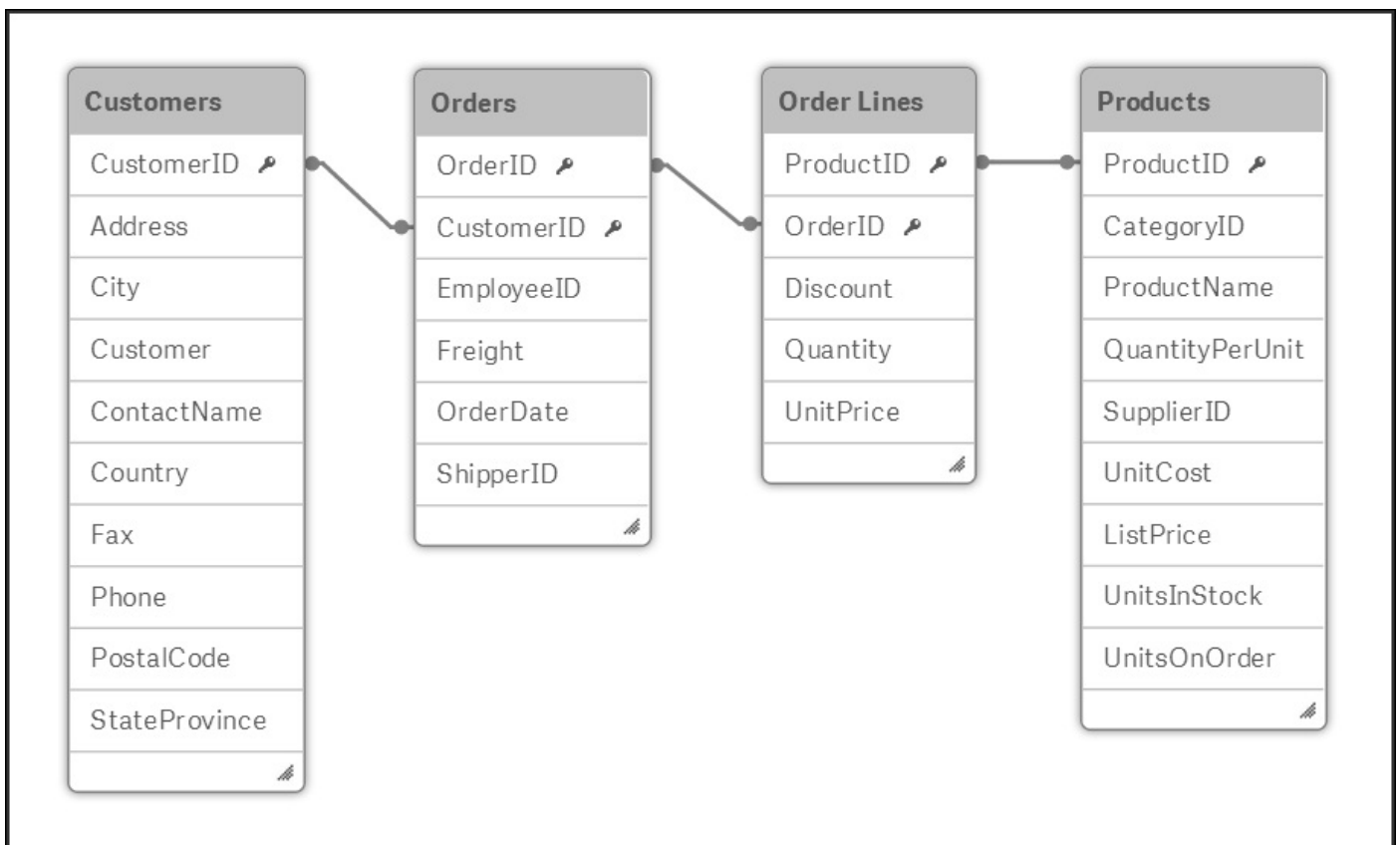


# The Qlik Sense® data model

Data used in Qlik Sense needs to be in a tabular form, very much like a table in Excel. A column in the table is referred to as a *field* and every row is a *record*.

When data is loaded into Qlik Sense, it is stored in the QIX internal database. In the simplest case, the data is just one single table. However, more commonly, the data model consists of several tables with links between them. These define how the different tables relate to each other. It is, hence, a *relational model*.

In the previous chapter, we saw an example where four tables were used: Customers, Orders, Order Lines, and Products:



*A simple data model made from four tables*

This is in fact the core of a very common business application—a sales analysis based on the registered orders.

The structure is not a coincidence. Rather, the reason why it looks the way it does is that it is a reflection of the real business processes. The relations these four entities have in reality dictate the data model:

- *A customer may over time place several orders.* Hence, customers and orders should be stored in different tables, and the customer ID should be stored in the Orders table.
- *An order may contain several order lines.* Hence, orders and order lines should be

stored in different tables, and the order ID should be stored in the `order_lines` table.

- *Several different order lines can refer to the same product.* Hence, products and order lines should be stored in different tables, and the product ID should be stored in the `order_lines` table.

You could add a number of additional tables, and for each table you will need to ask yourself what its relation is with the already existing tables. The new table could be a list of countries to which the customers belong, of product groups, of shippers, of suppliers, of invoices, and so on. The business processes define whether a customer can belong to more than one country, whether a product can belong to more than one product group, and whether a single invoice can refer to more than order.

Hence, you should look at the real-life entities and their relationships to understand how they should be modeled in the Qlik Sense data model.

# Creating a multitable data model

Loading several tables is technically just a matter of using several `Load` or `Select` statements in the script. Joins, in the way a database manager knows it from SQL, are usually not needed.

Normally, each `Load` or `Select` statement creates one table in the data model. Hence, if you want to load four tables, you should have four different `Load` or `Select` statements, each defining the appropriate table.

There are, however, some exceptions to this rule:

- If a loaded table contains exactly the same set of fields as a previously loaded table, the new table will *not* be created as a separate table. Instead, the loaded records will be appended to the existing table, which most likely is what you want. This way you can add more data to an existing table.
- If the `Load` or `Select` statement is preceded by the `Concatenate` or `Join` keywords, the loaded table will be merged with an existing data table. The `Concatenate` keyword is used if you want to add records to the table, just like in the previous bullet, but the two tables have slightly different sets of fields. The `Join` keyword is used to create the product between two tables, that is, the same as a `JOIN` in SQL.
- If the `Load` or `Select` statement is preceded by the `Generic` keyword, the loaded table will be transformed into several data tables. This is a keyword you need to use if your table is a generic database, that is, if the second to last column contains an attribute name and the last column contains the attribute value.
- If the `Load` or `Select` statements are preceded by the `Mapping` keyword, the loaded table will not be used as a normal data table. The table will be used for other purposes.
- A previously loaded table can be deleted using the `Drop` command. This is especially useful if you make many transformations and need temporary tables.

# Linking tables

Further, when loading multiple tables, the links between the tables are defined by the field names. At the end of the script run, the existing tables will be evaluated. If the same field name is found in more than one table, this field will be considered to be a field that links both the tables. This way, a data model is created. The logic in the script evaluation is hence identical in Qlik Sense and QlikView.

This means you need to make sure that the fields you want to use as links between the different tables, the key fields, are named the same in all tables you want to link. You can do this using the Profiling dialog (refer to the previous chapter) or you can rename them yourself by editing the script.

For instance, if the key is called `CustomerNo` in one table and `CustomerID` in the other, you could rename them like this:

```
Load CustomerNo as CustomerID, ... From Table1 ... ;  
Load CustomerID, ... From Table2 ... ;
```

Also, it is important that you make sure that you don't have any unwanted links. For example, you may have a field called `Description` in two different tables. This is of course not a key, but rather just a short name for something that in one table may be a product description and in another a customer description. Also, here you need to rename the fields:

```
Load Description as Description1, ... From Table1 ... ;  
Load Description as Description2, ... From Table2 ... ;
```

The goal is to create a script that defines a logical, coherent data model that corresponds to the business situation.



# Structuring your data

In a database, there are rules about where different entities are stored. For instance, everything about the customers should be stored in the `Customers` table. A customer identifier is stored in this table, which means that the necessary data can be retrieved by a simple lookup in the `Customers` table. So, if you need to refer to a customer from another table, you just store this identifier in the other table.

# Normalization

The identifier needs to be unique in the Customers table, so that every record can be addressed. Here the field is called a *primary key*. In other tables, duplicates may exist. For example, several records in the Orders table might refer to the same customer. If this is the case, the key is called a *foreign key*.

A consequence of this setup is that a database can have many tables, often thousands. One table for customers, one for orders, one for order lines, one for products, one for product groups, one for shippers, one for invoices, and so on. Each table contains one entity type, and each instance of this entity has exactly one row in the table—one record.

In the customer example, it means that each customer is stored in one row only, and all the attributes of this customer are stored in the different columns of this row. This is called *normalization*.

The main goal with normalization is to avoid redundancy. In a transactional database, it is important that a piece of information is stored in only one place. Not only does it take less space, it also speeds up the update process and minimizes errors. You can imagine the confusion if, for instance, a customer address is stored in several places and the different occurrences contain different addresses. Which version should you trust?

So, the source data is often highly normalized. But does this mean that the Qlik Sense data model should be normalized too?

Yes and No.

The Qlik Sense data model does not need to be as normalized as the source data. Redundancy is not a problem, and if duplicating data improves the user experience, you should definitely consider using duplicate data. The data model should be optimized for user experience and performance, not for minimal size.

But *some* normalization has great advantages—structuring the data in entities (tables) simplifies the data modeling work. It also makes maintenance simpler, since a data model and a script can be understood by a developer who has never seen it before. Finally, the QIX engine works better with a normalized model. It is easier to make it calculate numbers correctly and avoid double counting, which is a real problem when you start to de-normalize.

So the bottom line is that you should have *some* normalization, but it does not need to be as strict as in the source data.

The main case in which you need to de-normalize is if you use the same entity in different places in the data model.

For example, you may use a table listing external organizations in the context of supplier, shipper, as well as customer. This means the Organization table is used in different roles. In such a case, you should load the organization table three times: first as a supplier, then as a shipper, and finally as a customer, linking to the three different foreign keys.

Another common situation is that you have several dates in your data model: OrderDate,

RequiredDate, ShippingDate, and InvoiceDate. In other words, the date is used in different *roles*. Also, here you should load the dimension—the calendar—several times, once per date field.

Another reason to de-normalize is for optimization purposes. One of the cases would be if you have several very large tables linked to each other, for example, if you have an order headers table as well as an order lines table, and both are large (millions of records). From a performance perspective, this is not optimal. The QIX engine will need more CPU cycles than if the information of the two tables had been stored in one single table. So, you might want to join both the tables for performance reasons.

A small word of warning though, joining tables is not always safe. This operation may lead to a record being duplicated on multiple records, and if the record holds a number, the summation made by the QIX engine will lead to an erroneous result—the number will be counted twice. In the case of order headers and order lines, you know that an order line can belong to one order header only, so the numbers in the order lines table will not be duplicated. Hence, it is safe to join here.

However, if you have a number in the orders table, it will be duplicated. But luckily, this table rarely contains any numbers.



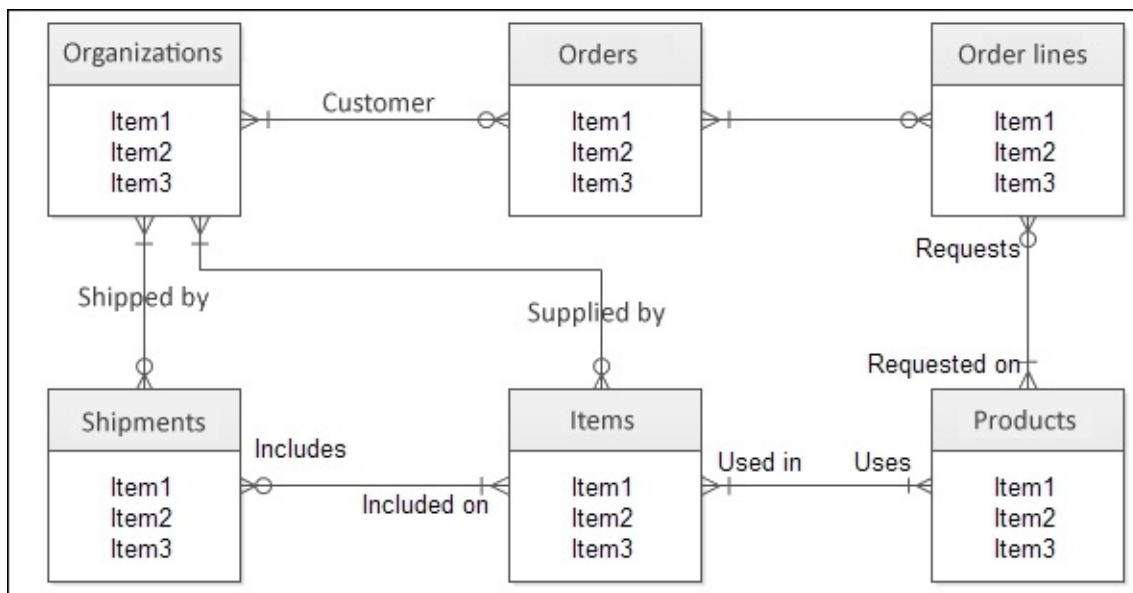
# Star schema and snowflake schema

The normalization is usually quite different in the source data model and in the analytical model. For the source data, one often talks about *transaction tables* and *master tables*. The transaction tables are the ones that store orders, invoices, and other transactions. In these, new records are typically added every hour, minute, or even second.

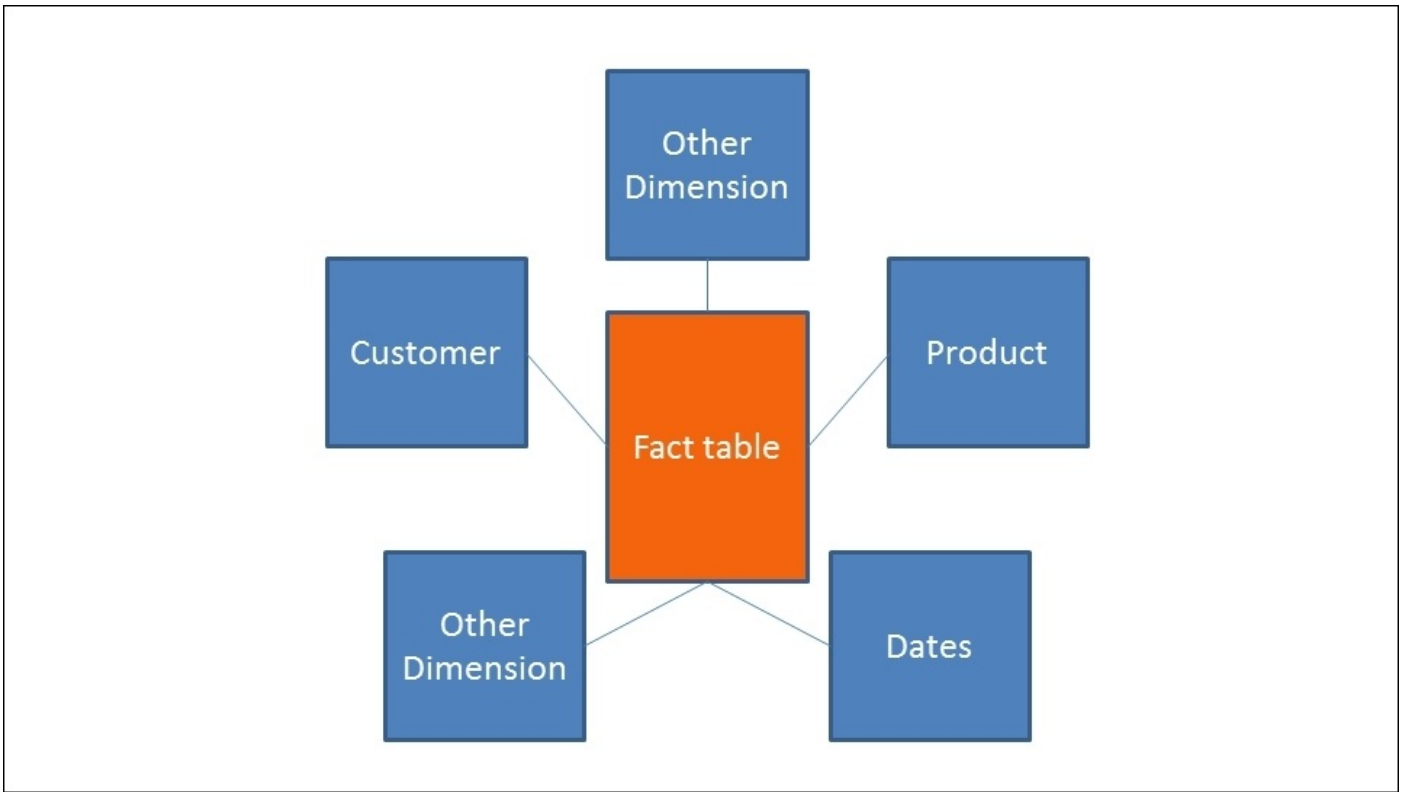
This is very different from the master tables, where new records are added much more rarely: Products, Customers, and the Calendar are typical master tables.

Master tables are often used for many purposes and are usually linked to several transaction tables, which makes the data model look as if it has circular references. This is, however, not a problem, since every link means a separate lookup in the master table.

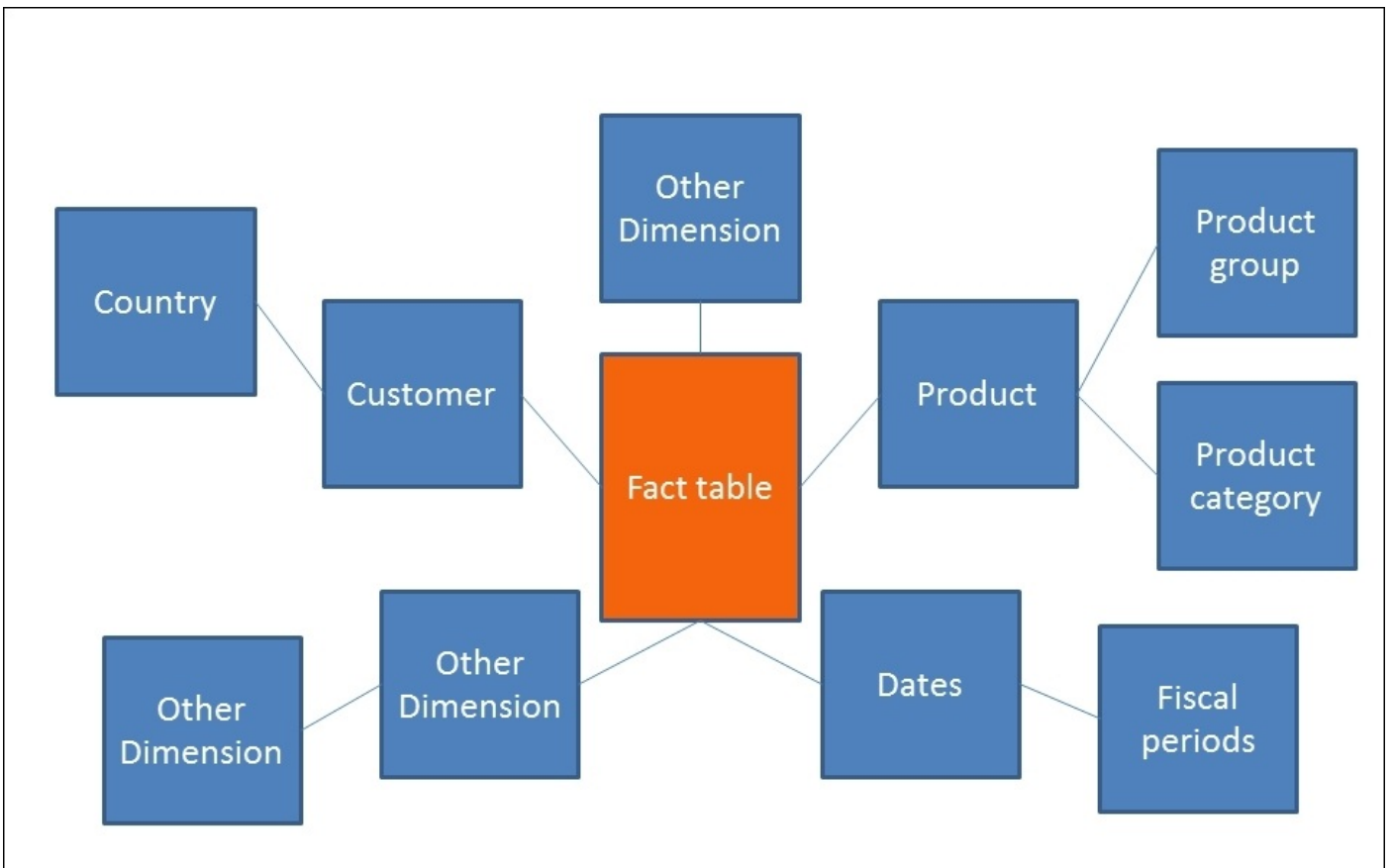
For example, in the following relational model, the Organizations table is linked to the transactional data through three keys: Customer, Shipped by, and Supplied by. This means that a specific shipment item can be linked to several organizations—one customer, one shipper, and one supplier:



In an analytical model, the tables are used in a different way. The transactions are joined into one single *fact table*, and the master tables are loaded as *dimensions*. The reason for this is partly historical. Older hypercube tools could not work unless all metrics were collected into a fact table. In addition, they could use hierarchical dimensions surrounding the fact table. The model will then look like a star; hence the name *star schema*:



This model has only one layer of dimensions—all the tables are directly linked to the fact table. But if the model instead uses dimensions in two or more levels, the model is called a *snowflake schema*:



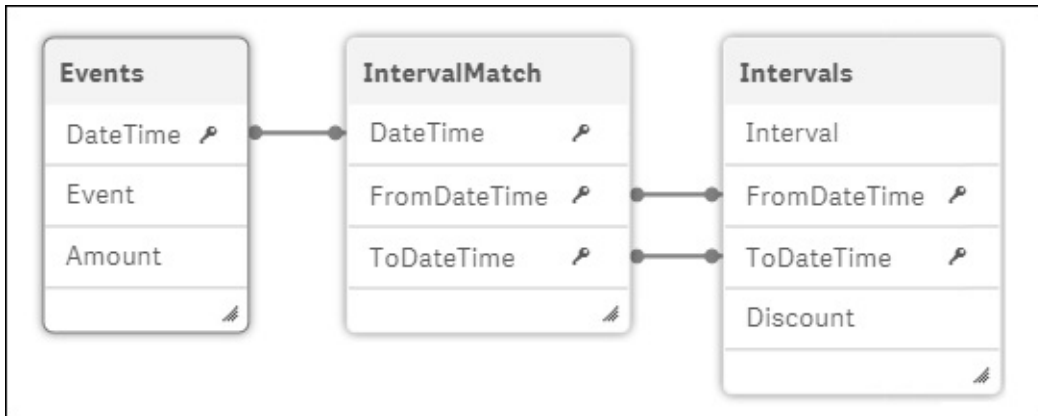
For Qlik Sense, the difference is minimal. All three data models can be used—provided that master tables used in several roles are also loaded several times. It is also possible to have metrics in any table, not just the fact table.

A star schema is, however, both simple and efficient, so we strongly recommend using this as an initial model for your data. It is then easy to build further and make it more complex.

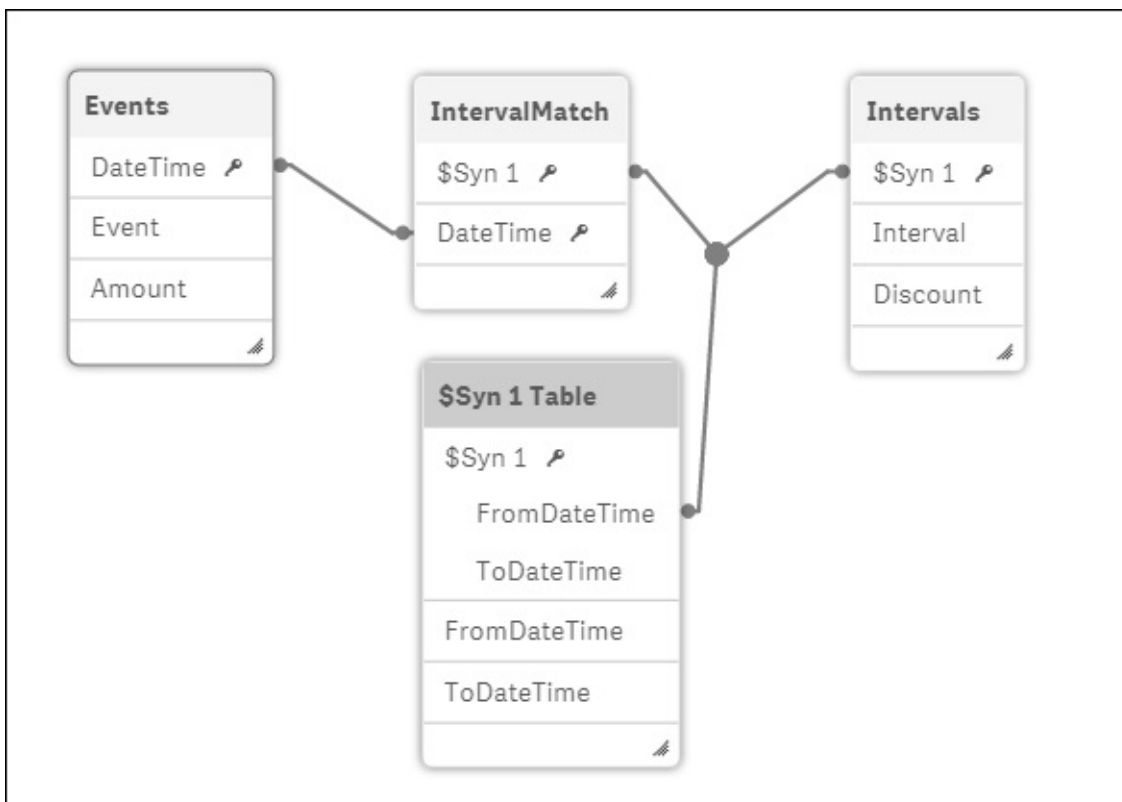
# Pitfalls in the data model

When you create your data model, you should look out for two potential problems: synthetic keys and circular references.

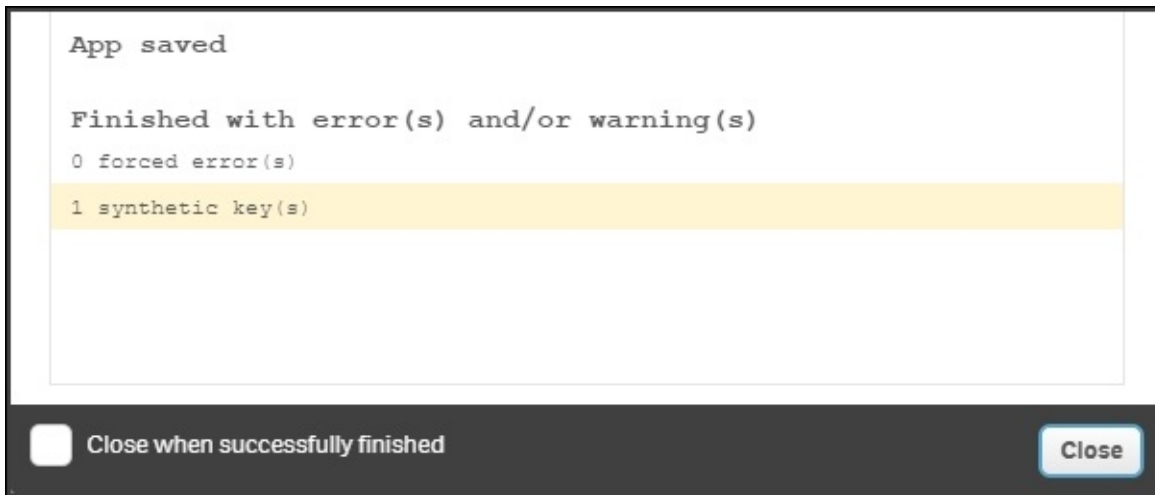
Synthetic keys are automatically created if you have multiple keys between two tables. They are not by themselves bad, but they could be a warning sign of a bad data model. If you have made a mistake when loading the data, the first sign is often one or several synthetic keys. Then, you need to go back and investigate why this has happened:



The preceding diagram shows a synthetic key modeled the way you loaded the data. It is a correct one that you don't need to change. Internally, it is stored differently; refer to the following diagram. In the data model viewer (as shown in the following diagram), you can toggle between these two views:

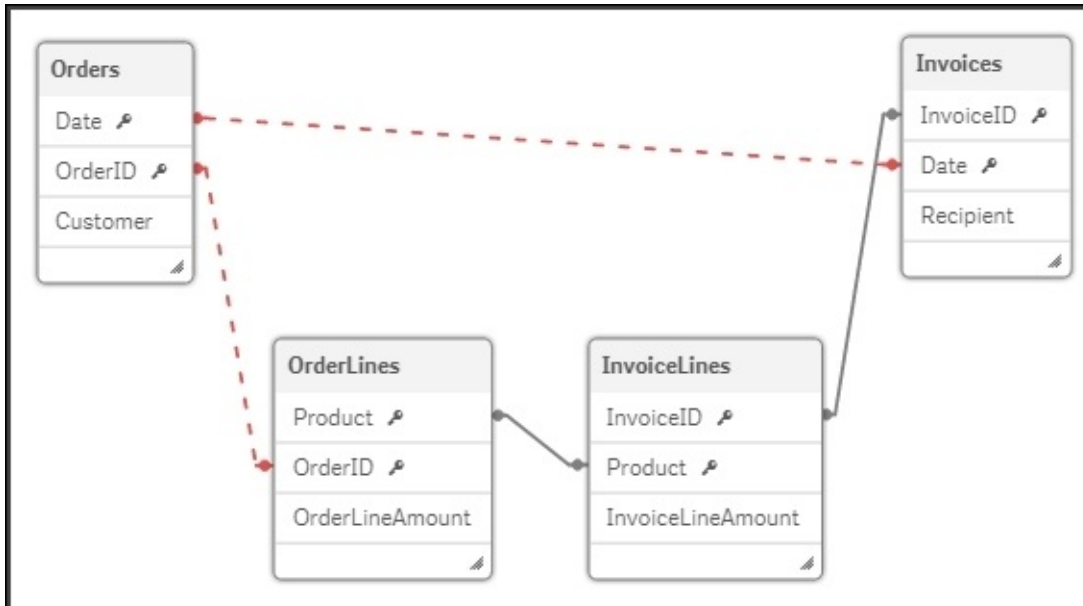


Qlik Sense will, at the end of the script run, warn you about these potential problems, as shown in the following screenshot:

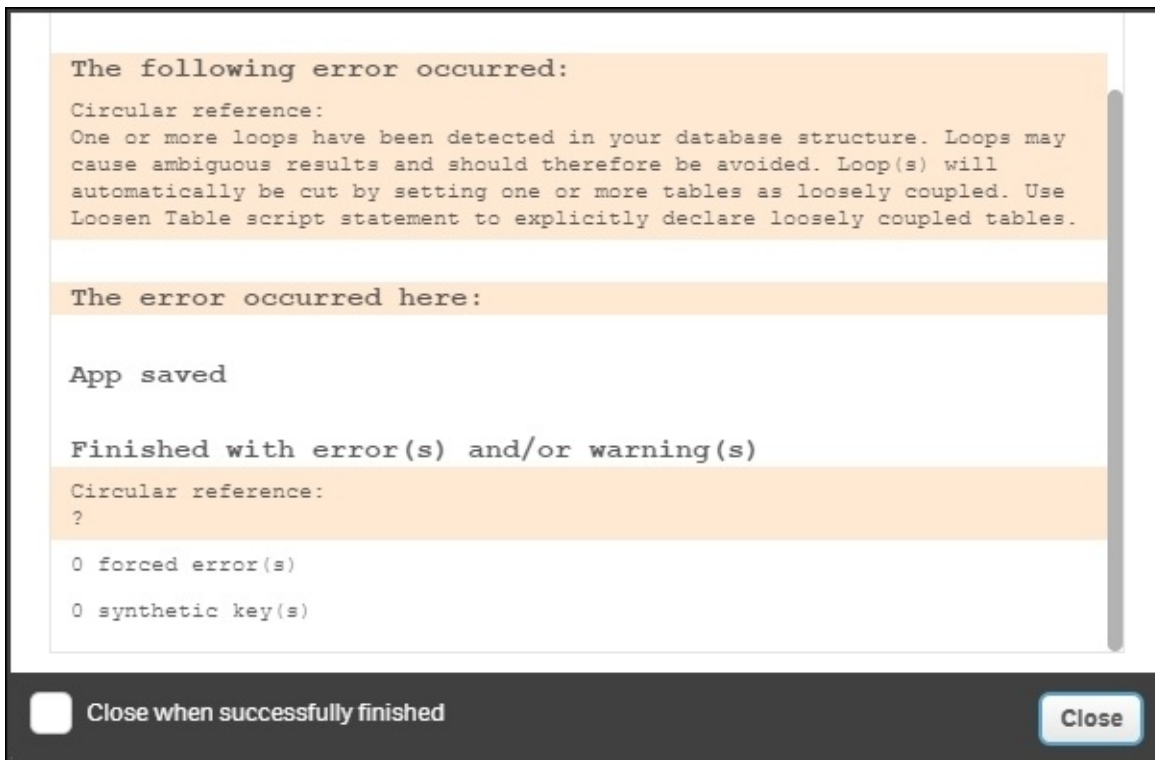


Usually, you do not want synthetic keys—you want one single key in each table link. However, if you know that you have multiple keys linking two tables, and that this is the way you want your data model, then there is no problem in having synthetic keys.

The second potential problem is circular references. This happens if you load data in such a way that the links between the tables form a loop. The following diagram is a typical example:



The circular reference from a data modeling perspective is an *error* and not just a warning, and you will get an error message at the end of the script run:



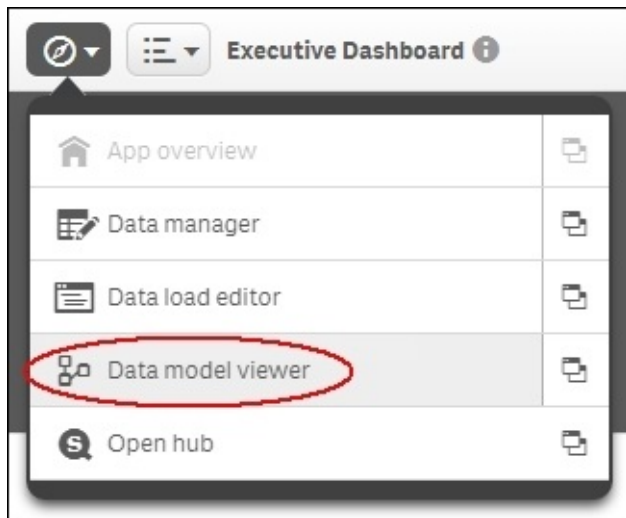
If you have a circular reference, you must rethink your data model with the goal of removing the loop.

It could sometimes be difficult to figure out how to remove a circular reference, but a good advice is to look at every link in your data model and ask, “Are these two fields *really* the same thing? Or do the fields have different roles?.” In the preceding screenshot, you have a circular reference where the Date field has two different roles: one is the date when the order arrived, and the other is the date when the invoice was sent. These two dates need not necessarily be the same. Hence, they should not be linked but instead loaded as two different fields.

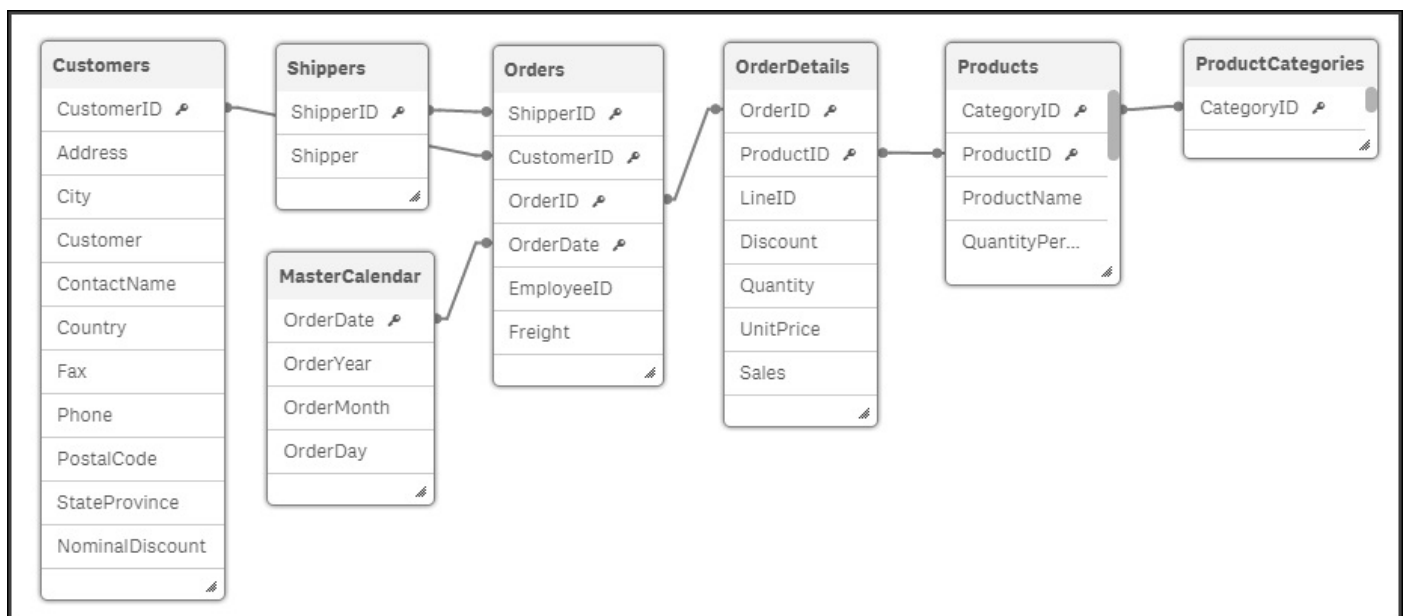


# The data model viewer

The script defines the data model, but if you want to view it graphically, you should use data model viewer. This is opened from the toolbar menu, as shown in the following screenshot:



Clicking on the icon to the right in the menu will open **Data model viewer** in a new tab. Once this is open, you can visually see what the data model looks like. If you have more than one table, they should be linked by the key fields and should look something similar to the following screenshot:

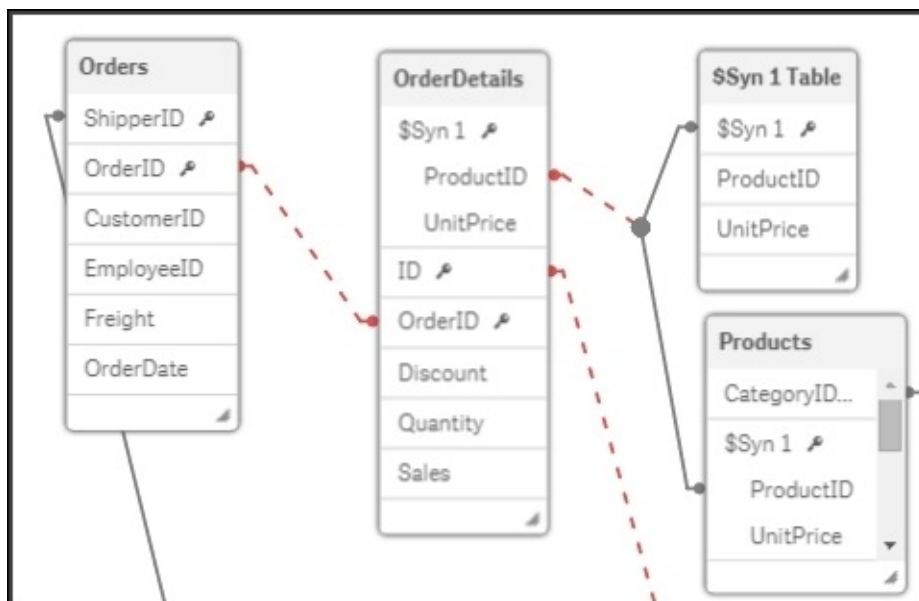


The **Data model viewer** is an excellent tool to get an overview of the data model. It is also a very good debugging tool in the application development process.

Check whether the data model looks the way you want it to and make sure that you have no circular references. Circular references will be marked with red links and synthetic



keys will be named \$Syn:



# Using preview mode

The **Data model viewer** option has a useful additional feature, preview mode. This mode allows you to preview both the data and metadata of a field.

Select a field in a table and click on the **Preview** button to the lower-left corner of the screen. This opens the **Preview** panel in which you can see data about the field and some sample data records from the table. In addition, you can define dimensions and measures based on the chosen field, as shown in the following screenshot:

The screenshot displays a data model diagram with the following tables and fields:

- AggSales**: AggKey
- FinalSalesHeader**: Order Number
- SalesDetails**: Order Number, Item-Branch Key, DateKey, Key, Address Number, AggKey, BudKey
- Customers**: Address Number
- FinalItemBranch**: Short Name, Item-Branch Key
- FinalItemMaster**: Short Name

The **Preview** panel for the **DateKey** field in the **SalesDetails** table is shown below:

DateKey		Preview of data				
		Order Number	Item-Branch Key	DateKey	Key	Address Number
Density	100%	200286	CR-10799	41476	7/21/2013200286CR-10799	10025919
Subset ratio	100%	205053	CR-10799	41452	6/27/2013205053CR-10799	10025919
Has duplicates	true	202518	CR-10799	41402	5/8/2013202518CR-10799	10025919
Total distinct values	533	202147	CR-10799	41402	5/8/2013202147CR-10799	10025919
Present distinct values	533	205054	CR-10731	41452	6/27/2013205054CR-10731	10025919
Non-null values	207976	200285	CR-10384	41476	7/21/2013200285CR-10384	10009985
Tags	\$key \$numeric \$integer \$timestamp \$date	211103	CR-10384	41542	9/25/2013211103CR-10384	10009985



# Summary

In this chapter, we looked at the functions and commands you need in order to create a logical and coherent data model that reflects your business processes.

In the next chapter, we'll move away from app creation and start examining how a Qlik Sense server can be deployed in the cloud.



# Chapter 7. Qlik Sense® Apps in the Cloud

In the previous chapters, we looked at how to go about creating Qlik Sense apps.

In this chapter, we look at sharing those apps with colleagues and friends in the cloud. We will cover the following topics:

- Why using the cloud makes sense
- Sharing Qlik Sense Desktop apps in the cloud
- Creating Qlik Sense apps directly in the cloud
- Maintaining Qlik Sense Cloud apps
- Adding context through external data from Qlik DataMarket

# Why use the cloud?

Deploying software in the cloud removes infrastructure challenges and provides access to data and computing power, software, and services from virtually anywhere. For Qlik Sense Enterprise customers, this might mean a large organization choosing to deploy Qlik Sense apps on a *private* (internal) cloud, using a platform such as AWS, for example. This could enable them to scale up their applications in a cost-effective and flexible manner. Deploying using a private cloud model is out of the scope of this chapter; however, it instead covers how Qlik Sense Desktop users can also get the benefits of the *public* cloud in two ways—through sharing apps and adding data content.

# Cloud sharing

The key value of cloud enablement for Qlik Sense is bringing together data and people by deploying apps in the cloud. This fosters collaboration through the sharing of insights and allows more people to create and discover through Qlik Sense.

Qlik Sense Cloud allows developers to share dashboards and storyboards for free via the Web for up to five users. Qlik Sense Cloud also offers the ability to create apps in the cloud, removing the need to use the desktop, and meaning that apps can be developed without the need to download a Windows executable, on any device running an HTML5-enabled web browser.



# Cloud content

Qlik Sense developers can use the cloud as a source of data to broaden the scope of their Qlik Sense apps via Qlik DataMarket.

Qlik DataMarket provides and integrates third-party data with the Qlik Sense visual analytics platform, much of it for free. With data-as-a-service, app developers can use a comprehensive library of external data directly within Qlik Sense, allowing them to augment and cross-reference their internal data to gain context and so drive new insights.

Qlik DataMarket comes preconfigured with many ready-to-use data sources via a subscription model. Data sources include business demographics, currencies, population, economic indicators, development indicators, and weather, all offered at various levels of granularity, some down to postal code.



# Using Qlik Sense® apps in the cloud

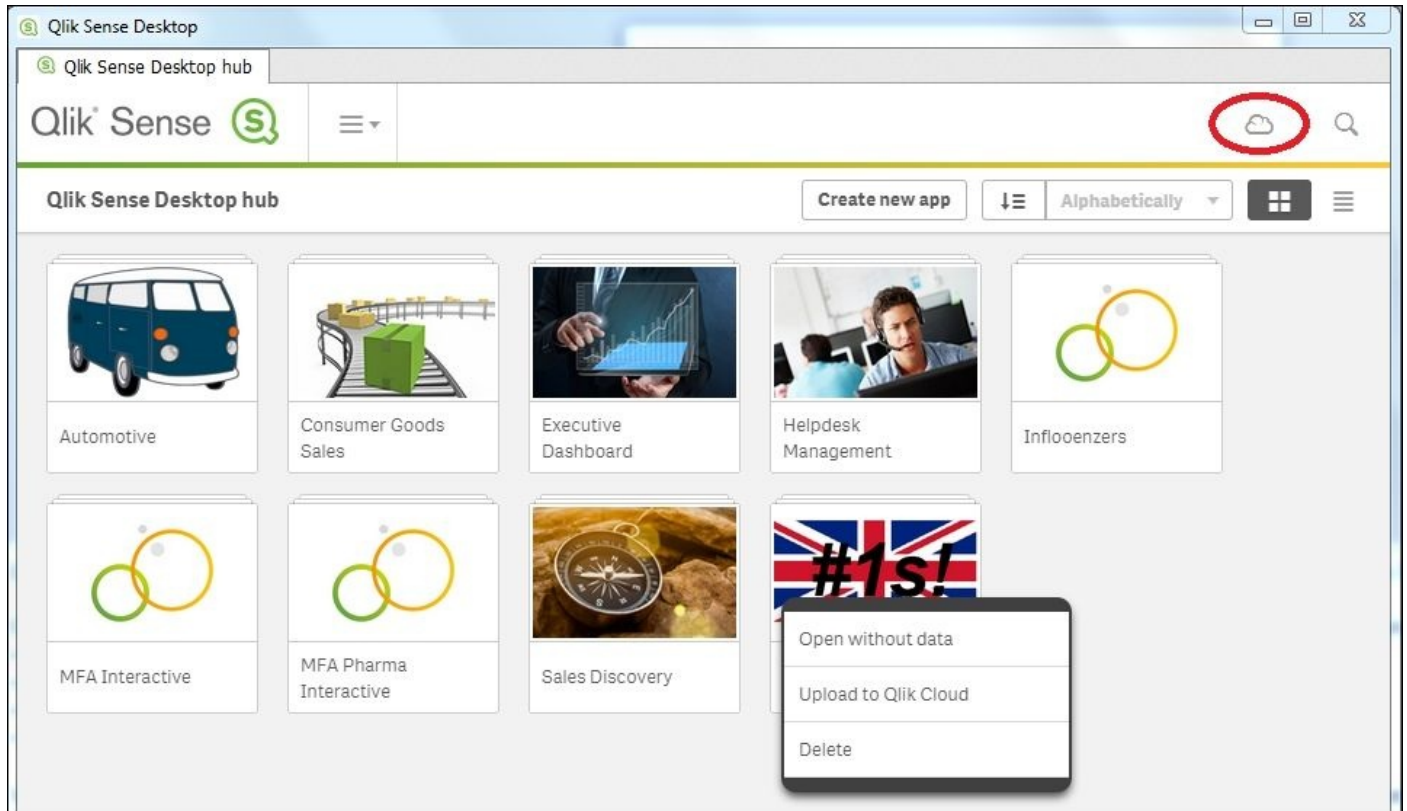
So, you've built a Qlik Sense Desktop app and you want to share it with others, either colleagues or friends, so you can find insights together. The cloud is the best way to do that, and doing so is built directly into Qlik Sense.

Qlik Sense isn't just for work, playing with data can be fun too. In this chapter, I will share a Qlik Sense app exploring UK music singles chart data going back to 1952.

# Uploading an app from the desktop

To upload an app, you first need to log on to Qlik Sense Cloud on the Web. From the Qlik Sense hub, there are two ways to open Qlik Sense Cloud: either by clicking on the cloud image in the top-right corner of the window or by right-clicking on the app itself. Both the actions will take you to the Qlik Sense Cloud Hub in a browser. At this point, if you haven't done so already, you'll have to register.

The following screenshot shows the option for in-built cloud uploading:

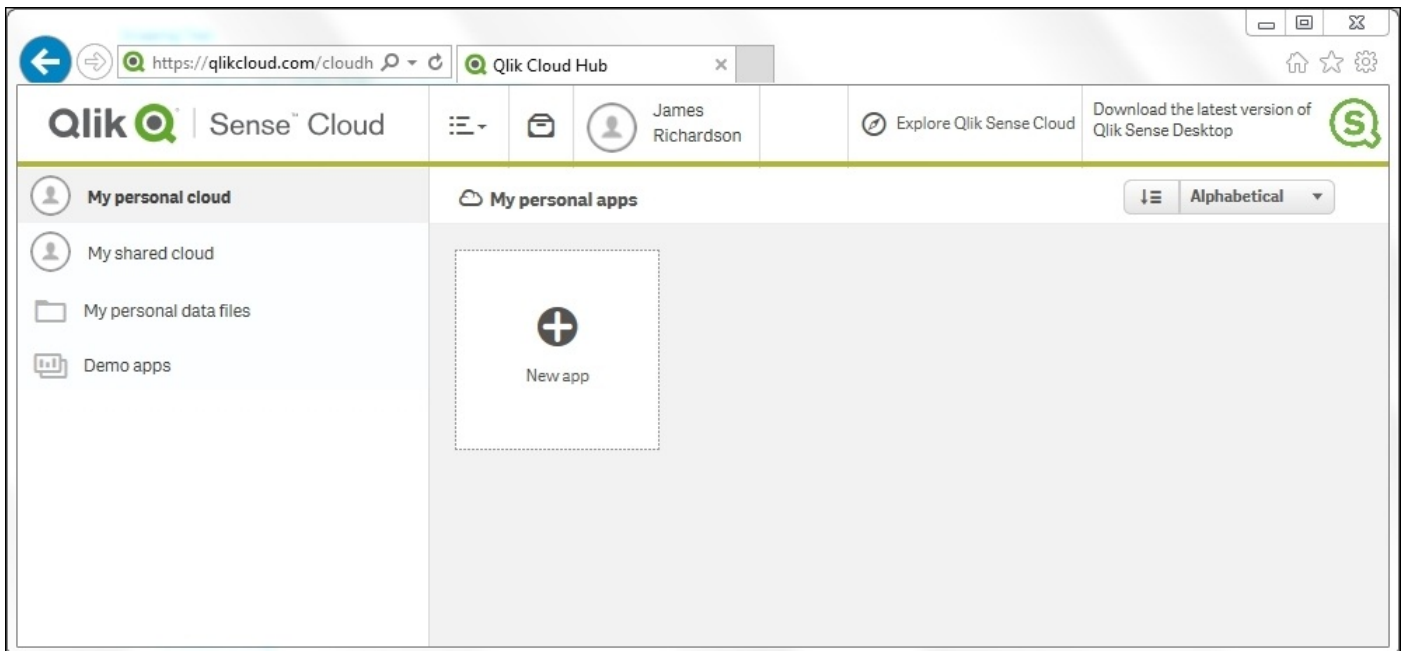


## *Options to upload to Qlik Sense Cloud from the desktop hub*

As you'll see when you log in, Qlik Sense Cloud is intentionally simple, presenting just a few options. The actions you can take are:

- Uploading Qlik Sense apps
- Uploading personal data files and creating apps directly in the cloud
- Sharing apps with up to five people
- Managing your Qlik Cloud profile

The following screenshot shows the Qlik Sense Cloud Hub:



### *Qlik Sense Cloud Hub options*

Uploading an app you have built on the desktop is simple. Click on **New app**, navigate to the file in the directory, and import. The uploaded QVF file brings the data model, sheets, and stories with it to the cloud for sharing. Note that the source data that was loaded into the app is not uploaded to Qlik Sense Cloud. This has implications for maintaining the app—more on this in a later section.

Create a new app in the cloud

Title

Create app

You must upload data files in the [My personal data files](#) section of the Qlik Sense Cloud hub to access your data within your apps

\_\_\_\_\_ Or \_\_\_\_\_

Upload an app

Your apps can be found at the following location:

C:\Users\<<your\_user\_name>\Documents\Qlik\Sense\Apps  
\<<file\_name>

Choose file Import

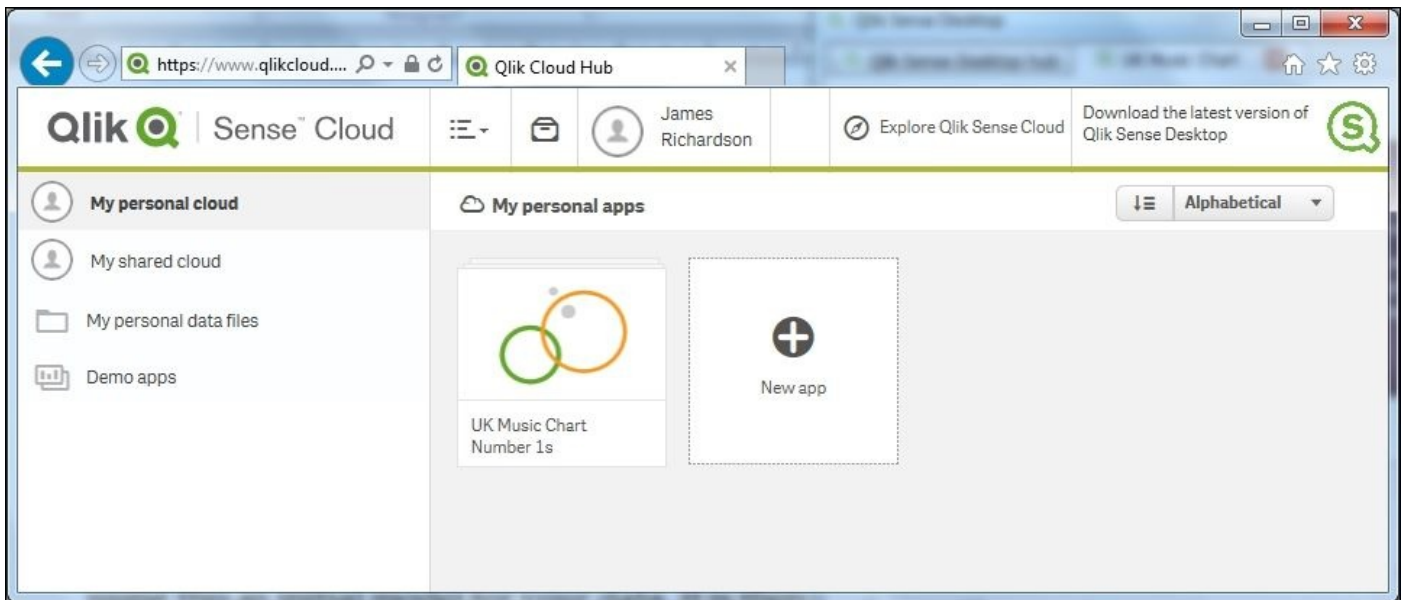
Upload requirements:

- You must close the app before importing it into Qlik Sense Cloud.
- An app may be up to -1.
- Your total storage space is -1.

Close

In the following screenshot, you can see the music chart app after upload. You'll notice that when uploaded, the app appears with the default thumbnail picture (rather than the custom image). Qlik Sense Cloud Hub does not currently support custom thumbnail images.

While not visible in the cloud hub, within the cloud app's information pane, you can show the custom image by selecting it from the media library. The media library is imported on upload, and includes thumbnails and any graphic files you've added to stories in the desktop app:



After the upload, you can use the app in exactly the same way as if it was running on a desktop. There's no difference; it's just Qlik Sense running on a server hosted by Qlik in the cloud. Modifications to the app can still be made using the **Edit** button, but only when the app is in the **My personal cloud** stream (more on this later).

Note that, unless you selected the publish option at upload, only you can see it—it's not been shared with anyone else yet.

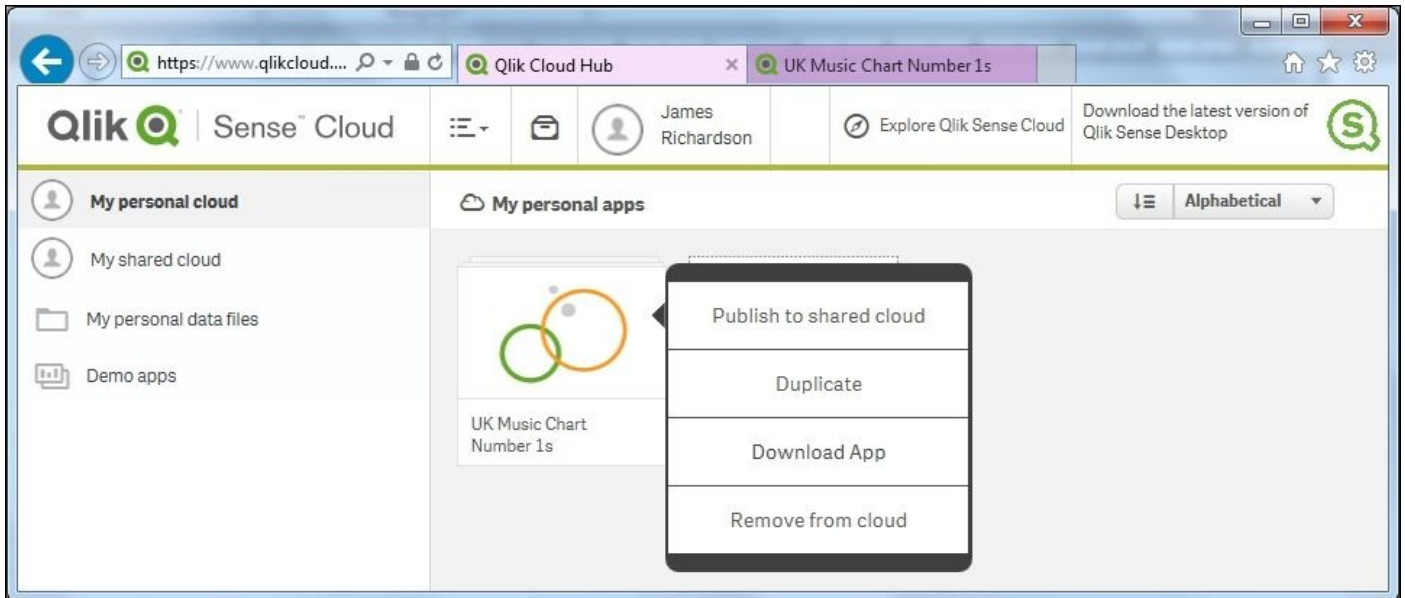
# Creating an app in Qlik Sense® Cloud

Qlik Sense apps can also be created directly in Qlik Sense Cloud. The only difference in doing so is that data must first be loaded into the `My personal data files` folder on the Qlik Sense Cloud Hub in order to be available for loading.

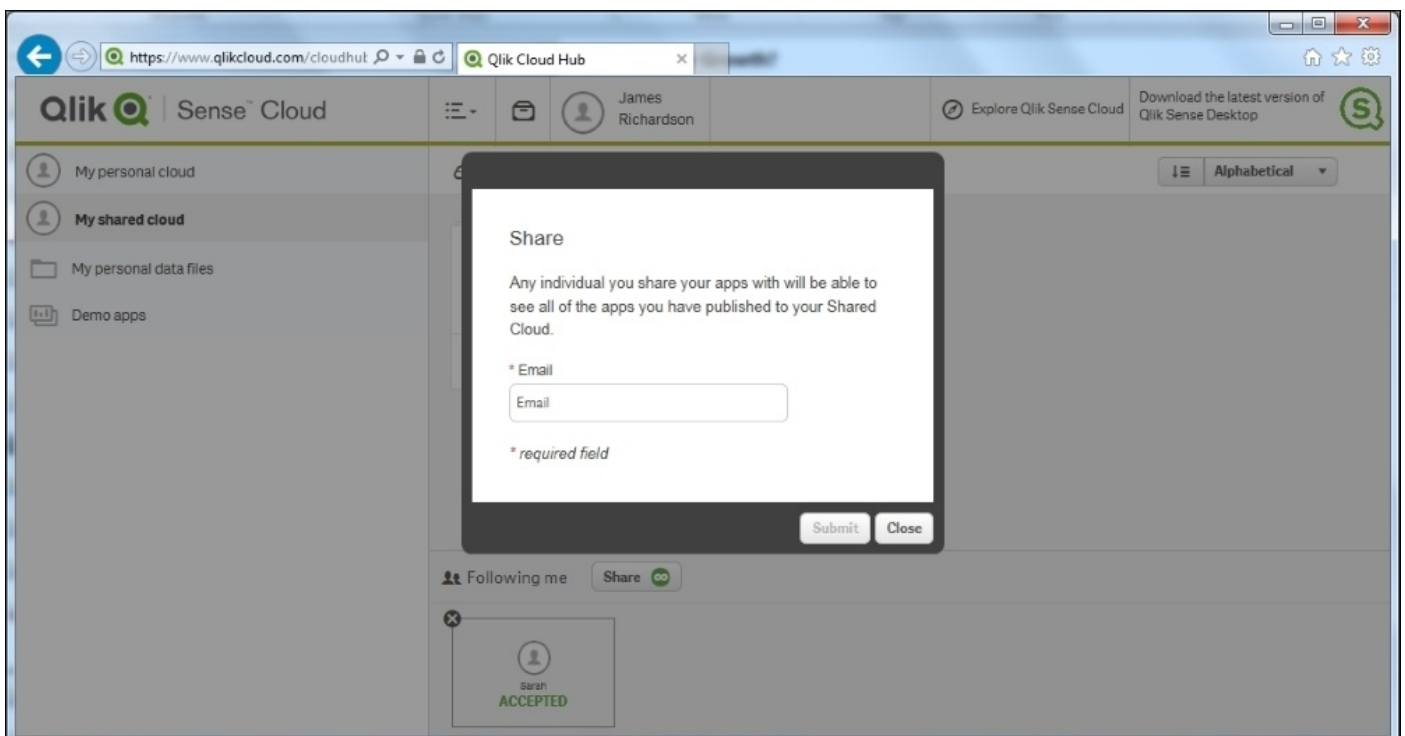


# Sharing an app in Qlik Sense® Cloud

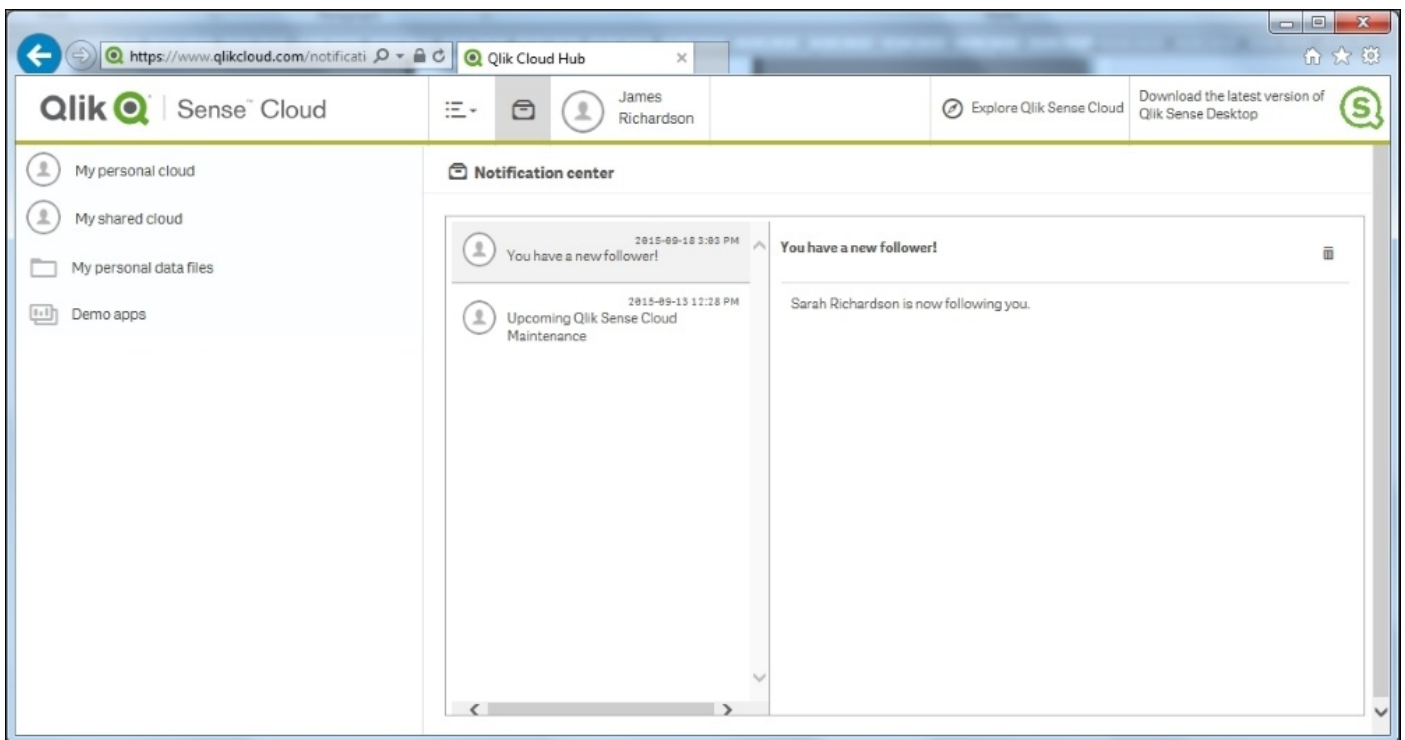
Qlik Sense Cloud is designed to enable people to share apps. To do this, a cloud app needs to be published, moving it from the **My personal cloud** stream to the **My shared cloud** stream. To do so, right-click on the app icon in the cloud hub and select **Publish to shared cloud**, as shown in the following screenshot:



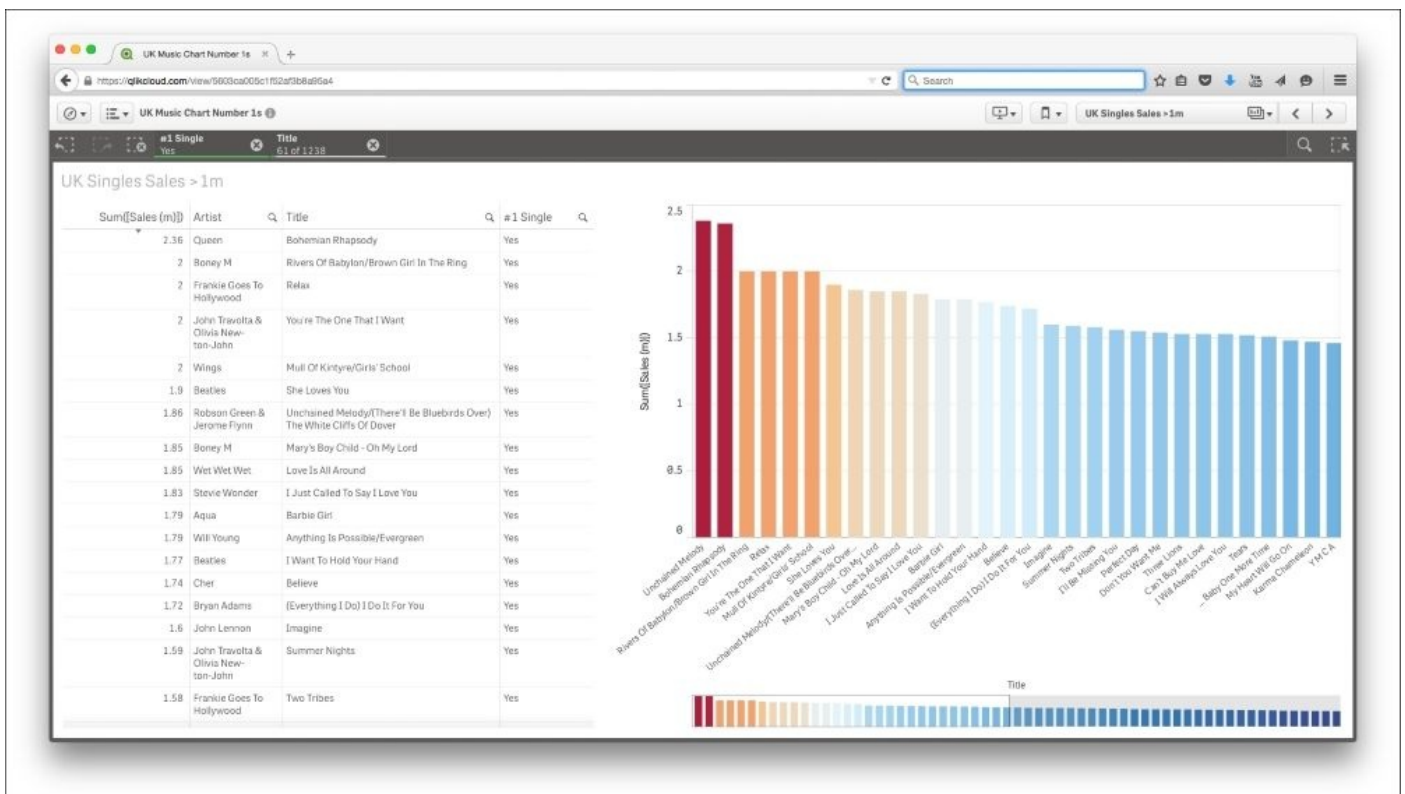
Once published, you can invite up to five people to access Qlik Sense apps in your shared cloud stream. Note that anyone you share it with will be able to see all the apps you have published. To do so, click on **share** and enter their e-mail address as shown in the following screenshot. When people accept the invitation to share and register on Qlik Sense Cloud, they appear as followers at the bottom of the **My shared cloud** stream:



By selecting the **Notification center** (in-tray icon), you can see who's viewed apps and other alerts about Qlik Sense Cloud, as shown in the following screenshot:



People that you share apps with can use those apps, making selections, and so on, but they cannot share them with others or modify them. You can see in the following screenshot (from my follower Sarah's Mac) that there is no edit pen icon on the shared app:





# Maintaining Qlik Sense® Cloud apps

Some thought needs to go into maintaining apps in Qlik Sense Cloud:

- Just as with Qlik Sense Server, in Qlik Sense Cloud, when an app has been published to a shared stream, it can no longer be modified. To make changes, it must be *unpublished*. When this happens, your followers will no longer be able to see the app, until you decide to publish it again.
- For uploaded apps, rather than those created in the cloud, it's important to note that any modifications made to the cloud version of an app (for example, adding new sheets, and so on) will not be reflected in the original Desktop version. The simplest way to deal with this is to download the Cloud version to the Desktop, by right-clicking on the app in the Qlik Sense Cloud Hub. Ideally, you should decide which version (Cloud or Desktop) is to be the master on uploading in order to avoid version management issues.

If there are changes to the source data that need reflecting in a Qlik Sense Cloud app, there are differences depending on how the cloud app was created.

- If the app was built directly in Qlik Sense Cloud using uploaded data sources, then the easiest approach is to take down the data, using the **Remove from cloud** right-click option, and then upload the new dataset before reloading. If the data file has the same name and structure, the reload will work with no problems.
- If the app was uploaded directly from the desktop, then it is a little more complex, as the source data will not be in the cloud. In this case, there are two options:
  - Remove the app from the cloud, reload the changed data on the desktop, and re-upload the app.
  - Upload the changed data to the **My personal data files** space on Qlik Sense Cloud and then modify the load script. The file path in the load script will need to be amended so that each section points to the uploaded data files, which reads `lib://<user>/<file>`.



# Using the Qlik DataMarket® content

Adding externally sourced data to any business intelligence app can yield new insights. For example, looking at how the weather affects the sales of certain products could be hugely beneficial to retailers. Much data is freely available on the Web, but it is rarely in an easily consumable form. This is exactly the problem Qlik DataMarket solves, by providing curated, normalized datasets that can be easily loaded into Qlik Sense available in the cloud.

Note that Qlik DataMarket works as a source for Qlik Sense irrespective of whether you are running in the cloud or not.

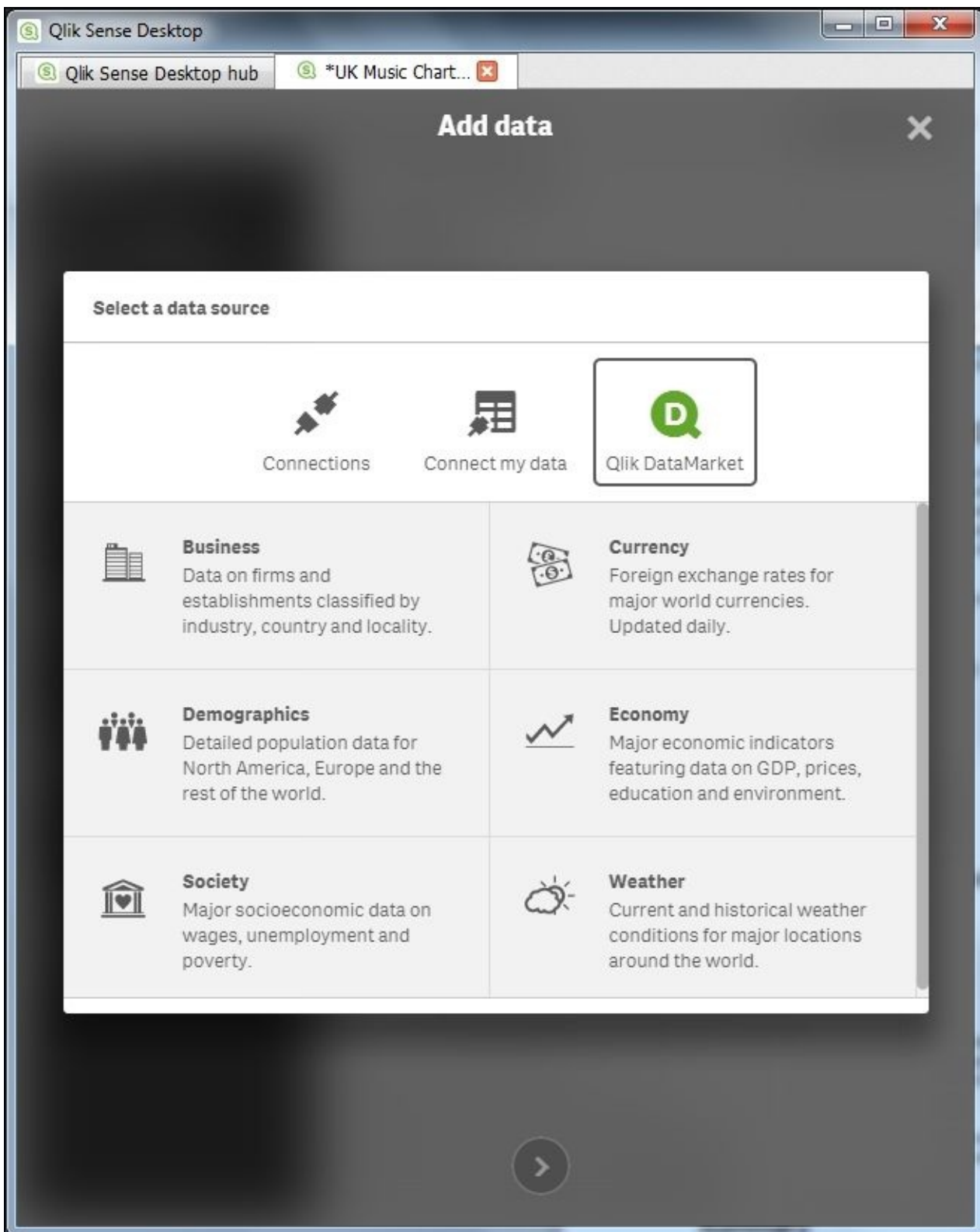


# Adding the QlikMarket® data

The UK music chart app contains a field showing the country, the musician, or the group it originated from. Perhaps it would be interesting to know which countries have had more hits per head of population. It is easy to do this in Qlik Sense, and can be done directly with cloud deployed apps too.

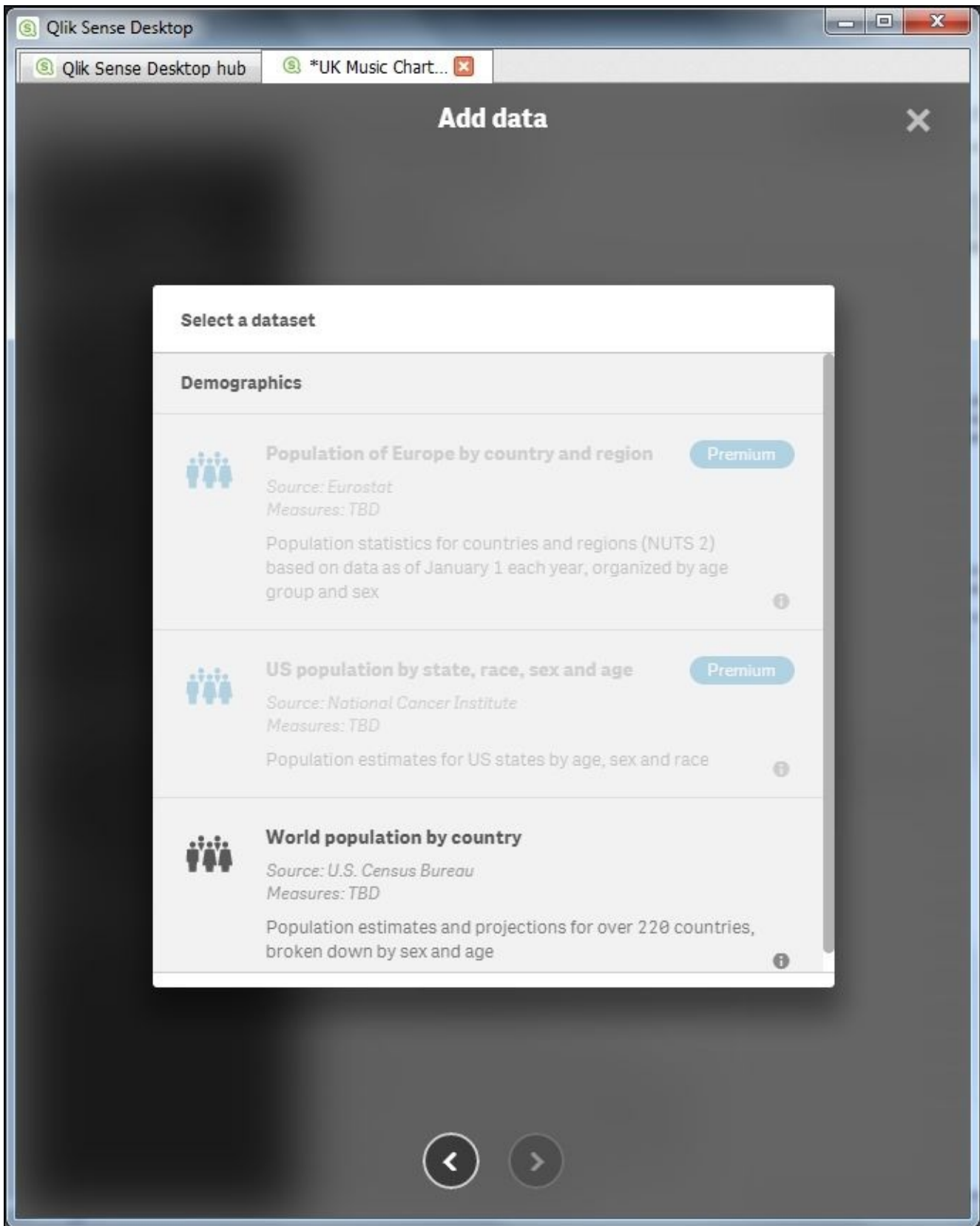
First, within the app, open the **Data Manager** and click on **Add data** and select **Qlik DataMarket**. This shows you a menu with the categories of data available. In this case, select **Demographics** as shown in the following screenshot:



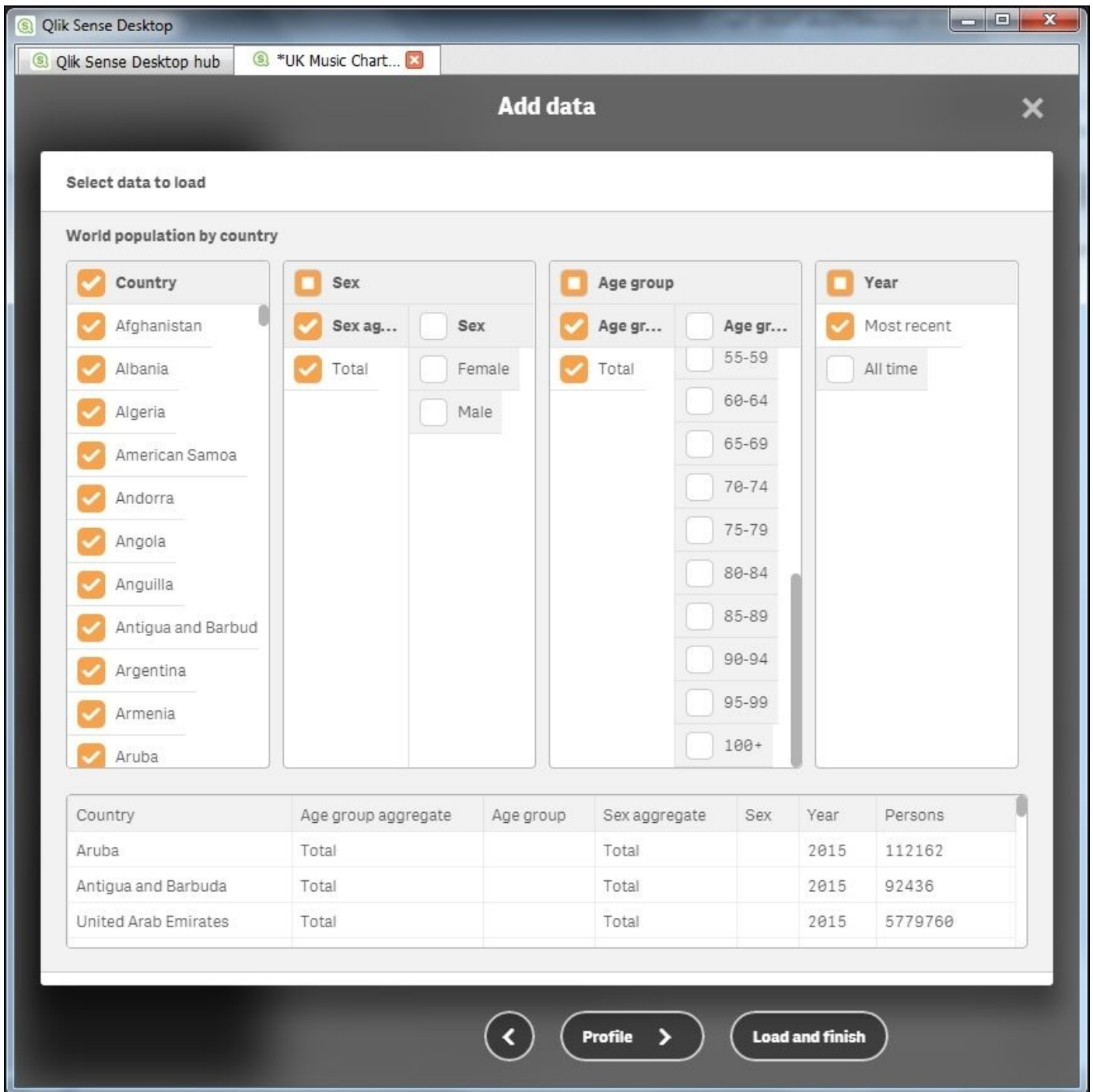


### *Qlik DataMarket data categories*

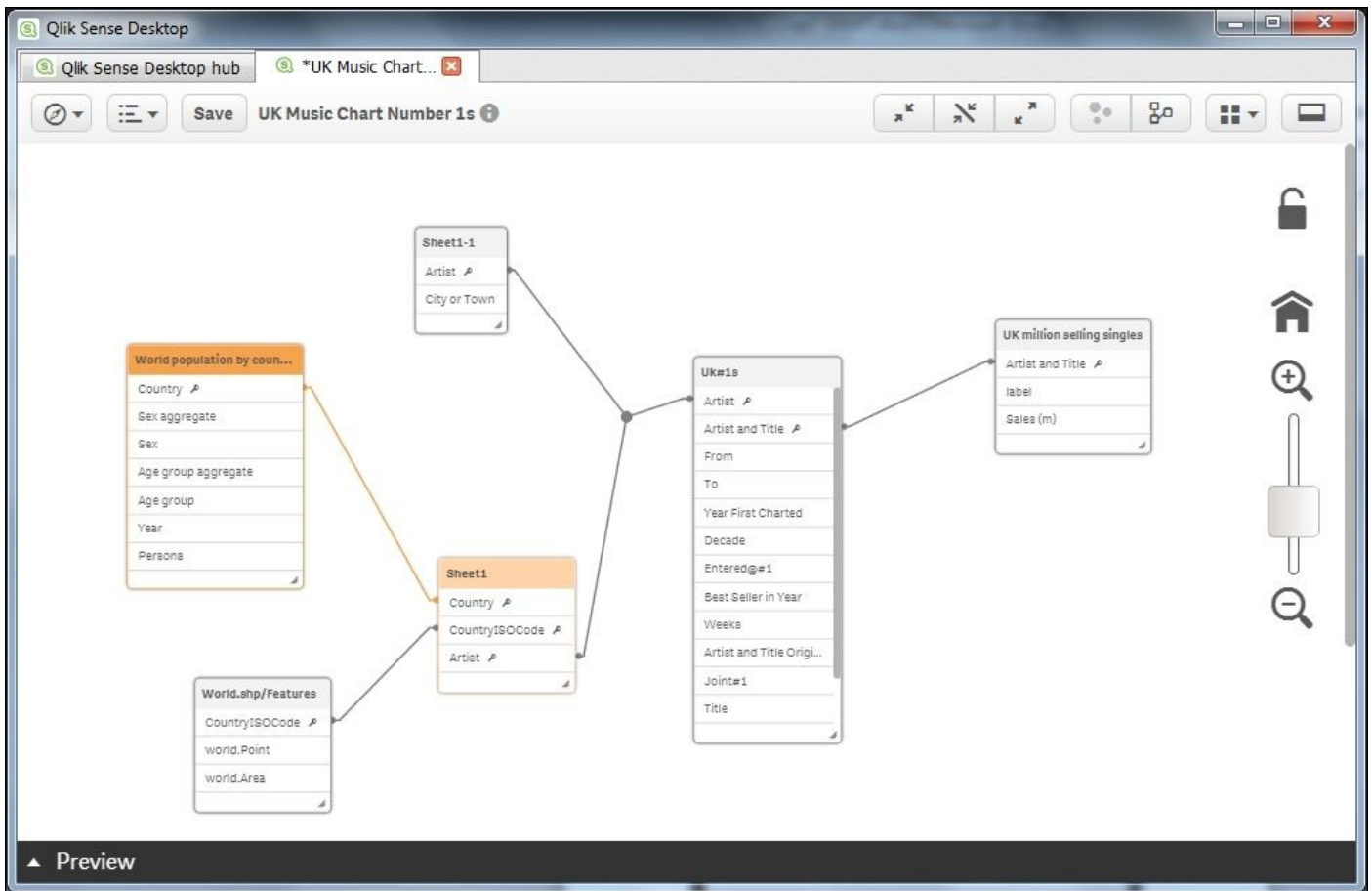
At this point, the available datasets containing demographic information are shown, including those that are premium (that is, paid for) and those that are free. In this case, the free dataset world population by country is what is needed:



At this point, the data from the set is selected before loading:



In the app, there is already a field called **Country**, which Qlik Sense uses as a key in order to allow you to work with the data, as the **Data Model Viewer** shows:



A few clicks and the app is enriched with content that provides context for analysis and insight, enabling users to ask more questions based on a broader scope and increased context.



# Summary

In this chapter, we looked at sharing Qlik Sense apps in the cloud, and at using Qlik's Data-as-a-Service offering to add context to apps.

In the next chapter, we'll look at extending the use of Qlik Sense through the Qlik Analytic Platform (QAP).



# Chapter 8. Extending the Qlik<sup>®</sup> Analytic Platform

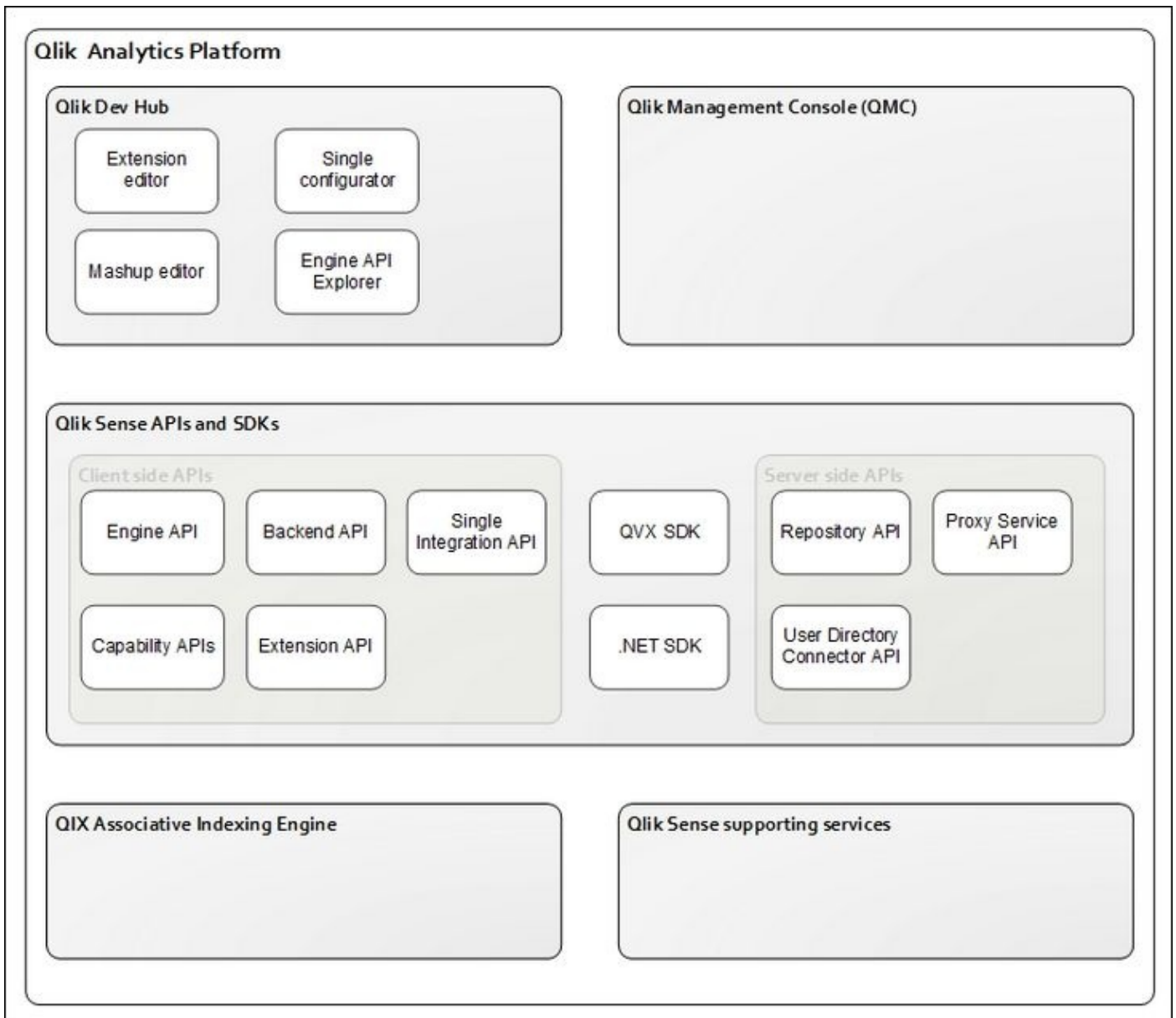
In the previous chapters, we outlined the various capabilities of Qlik Sense and their use. One of the advantages of Qlik Sense is that it is built on open API 's **Qlik Analytic Platform (QAP)**, that allows the customers and partners to extend their analytic solutions. This chapter will provide an overview and some interesting examples of how to enrich your solutions with QAP. It is not meant to replace the *Qlik Sense for Developers help* documentation, which can be found at <https://help.qlik.com/sense/2.1/en-us/developer/#Home-developer.htm>.

With this said, in this chapter, we will share some interesting examples and resources in the following key areas:

- Web mashups
- Extending Qlik Sense client
- Developer community – branch

QAP is made up of the following three layers, which include the client layer (Qlik Hub and Qlik Management Console), API, the SDK layer, and finally the Engine layer, which contains the QIX engine and all the supporting services:

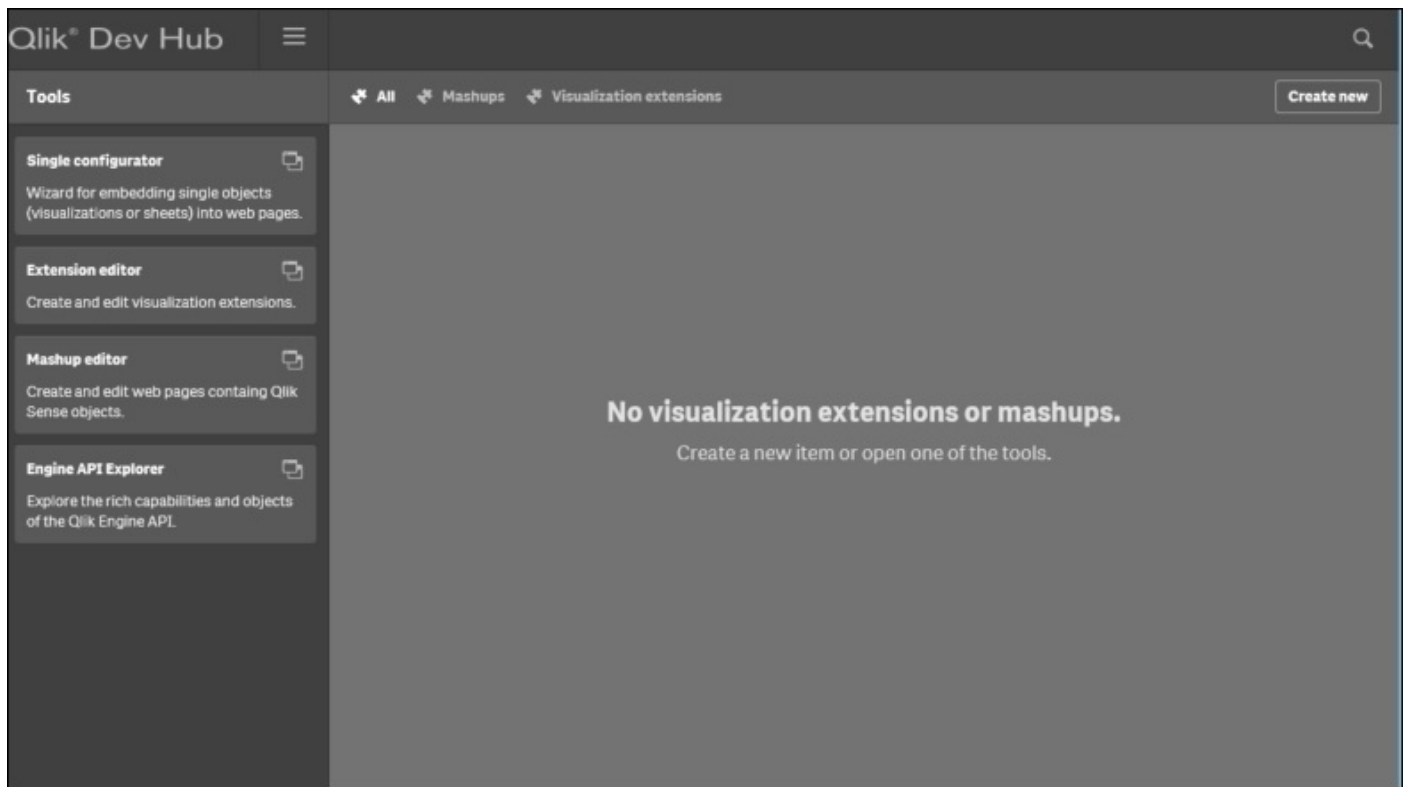




We will explore each of these layers through specific examples on how QAP is used to extend your data discovery solutions.

# Qlik® Dev Hub

Any discussion of QAP would be remiss without reviewing the new Qlik Developer Hub which was released in Qlik Sense 2.1:



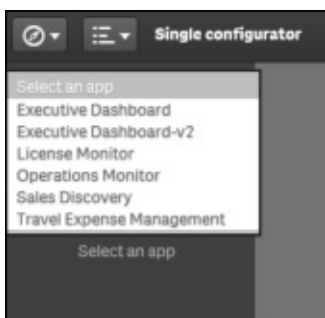
The **Qlik® Dev Hub**, shown in the preceding screenshot, was developed specifically to simplify access and development using Qlik's APIs, but it is not a replacement for the rudimentary concepts of JavaScript, HTML (Hyper Text Markup Language), and CSS (Cascading Style Sheets). The **Qlik Dev Hub** can be accessed via any Qlik Sense Server, or Qlik Sense Desktop via a browser. The link is <https://<servername>/dev-hub/>, and for the desktop version of Qlik Sense, it is <https://localhost:4848/dev-hub/>. This chapter will focus primarily on Qlik Sense Enterprise. Dev Hub provides four key tools for extending Qlik Sense solutions. They include the following:

- **Single configurator:** A Qlik Sense tool that provides an easy way of creating simple mashup pages by returning a URL that will resolve to a Qlik Sense object.
- **Extension editor:** An editor for JavaScript files and QEXT files. It assists you with creating new visualization extensions as well as editing existing ones.
- **Mashup editor:** An editor for JavaScript files and HTML files. It assists you with creating your own mashups displaying Qlik Sense data on your website. You can use the templates provided with Qlik Sense to get started with building your own mashups.
- **Engine API Explorer:** A tool that helps you explore the capabilities of Qlik Engine API.

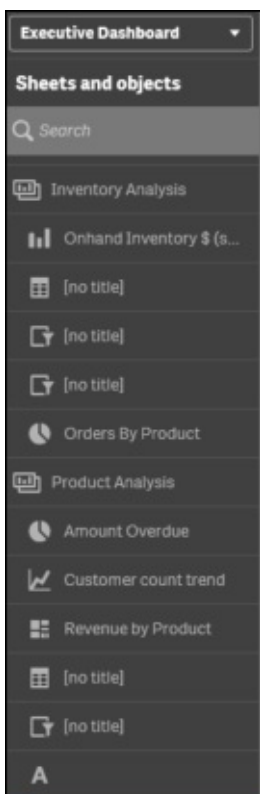
With this said, now let's take a closer look at the most common solution extensions that can be generated via Dev Hub.

# Web mashups

One of the most common requests is: can Qlik Sense create a web mashup with Qlik Sense objects? There are two options based on the level of control and interactivity required of the Qlik Sense objects. The first is the single configurator which provides an easy way to create simple mashup pages without having to create any code. It simply generates a URL that returns a complete HTML page with an embedded Qlik Sense visualization. To create a Qlik Sense object link, select the **Single configurator** as shown in the following screenshot. Once selected, you will need to select the application you wish to access for the mashup. In this example, the **Executive Dashboard** application was selected. It is important to note that a developer should access a published application that is in a stream that aligns with the requirements of the resulting mashup. There is additional information on administration in [Chapter 9, Administering Qlik Sense®](#).



Once selected, a listing will be generated of all the sheets and respective objects. A key point to highlight is that *both* sheets and individual objects are available based on your web page requirements.



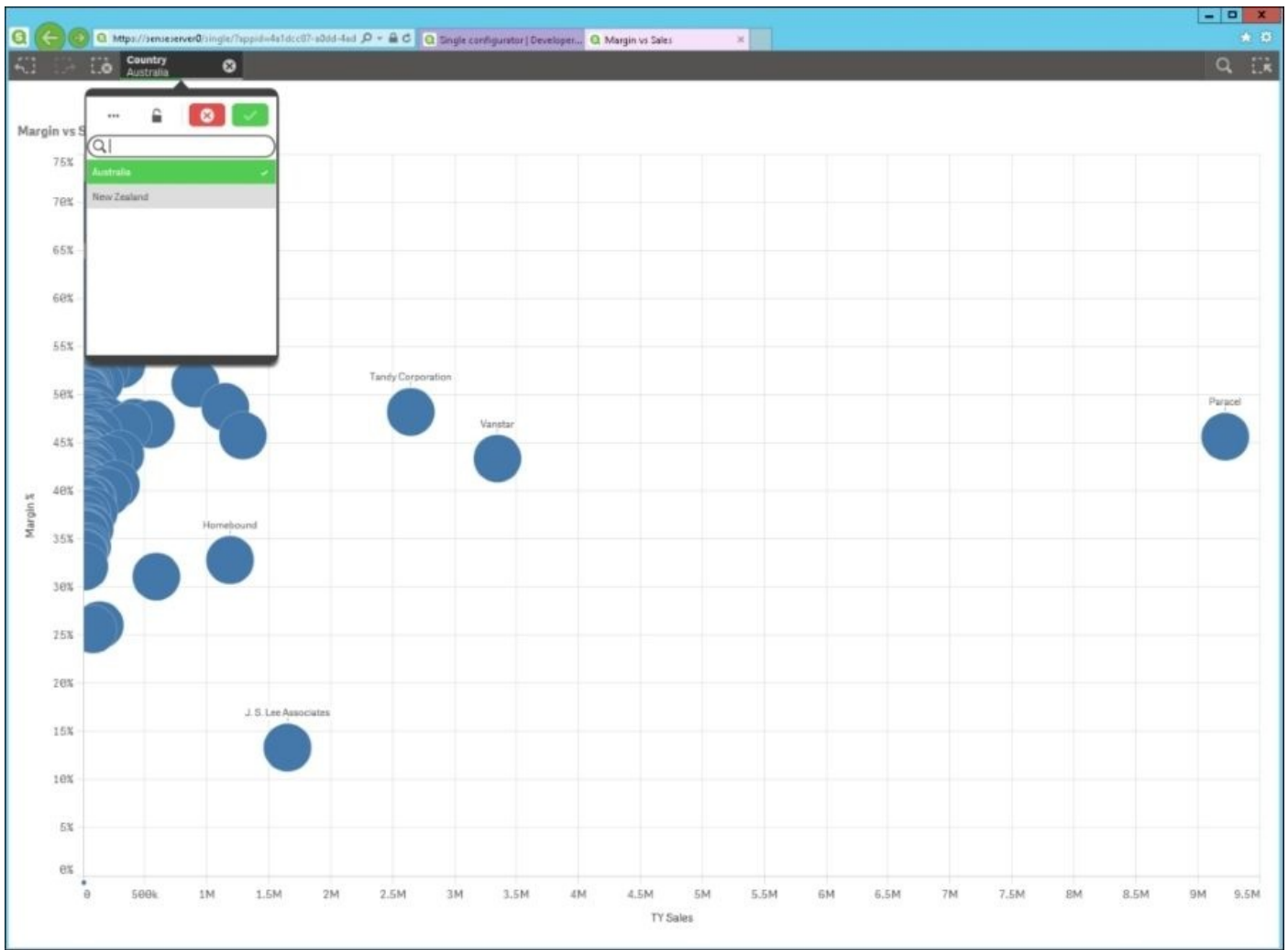
For this example, **Margin vs Sales** was chosen with selection bars showing so that a user can interact with this visualization.

In the following screenshot, you can see the use of the **Single configurator** generating the URL and the Iframe code for the **Margin vs Sales** by Sales Rep scatter chart that is contained in the **Executive Dashboard** Qlik Sense application:

The screenshot displays the Qlik Sense Single configurator interface. The central preview area shows a scatter plot titled "Margin vs Sales" with "Margin %" on the y-axis (0% to 75%) and "TY Sales" on the x-axis (0 to 9M+). Data points are labeled, including "Tandy Corporation", "J.S. Lee Associates", and "Parcel". The interface includes a sidebar with "Executive Dashboard" and "Sheets and objects", a central preview area with the chart, and a right-hand "Options" panel. Arrows point to the URL and IFrame code fields, the "Margin vs Sales" selection in the sidebar, and the "Show selections bar" option in the Options panel.

Company	TY Sales (Approx.)	Margin % (Approx.)
Parcel	9M	45%
Tandy Corporation	2.5M	50%
J.S. Lee Associates	1.5M	15%
Other Sales Reps	0.5M - 1.5M	15% - 70%

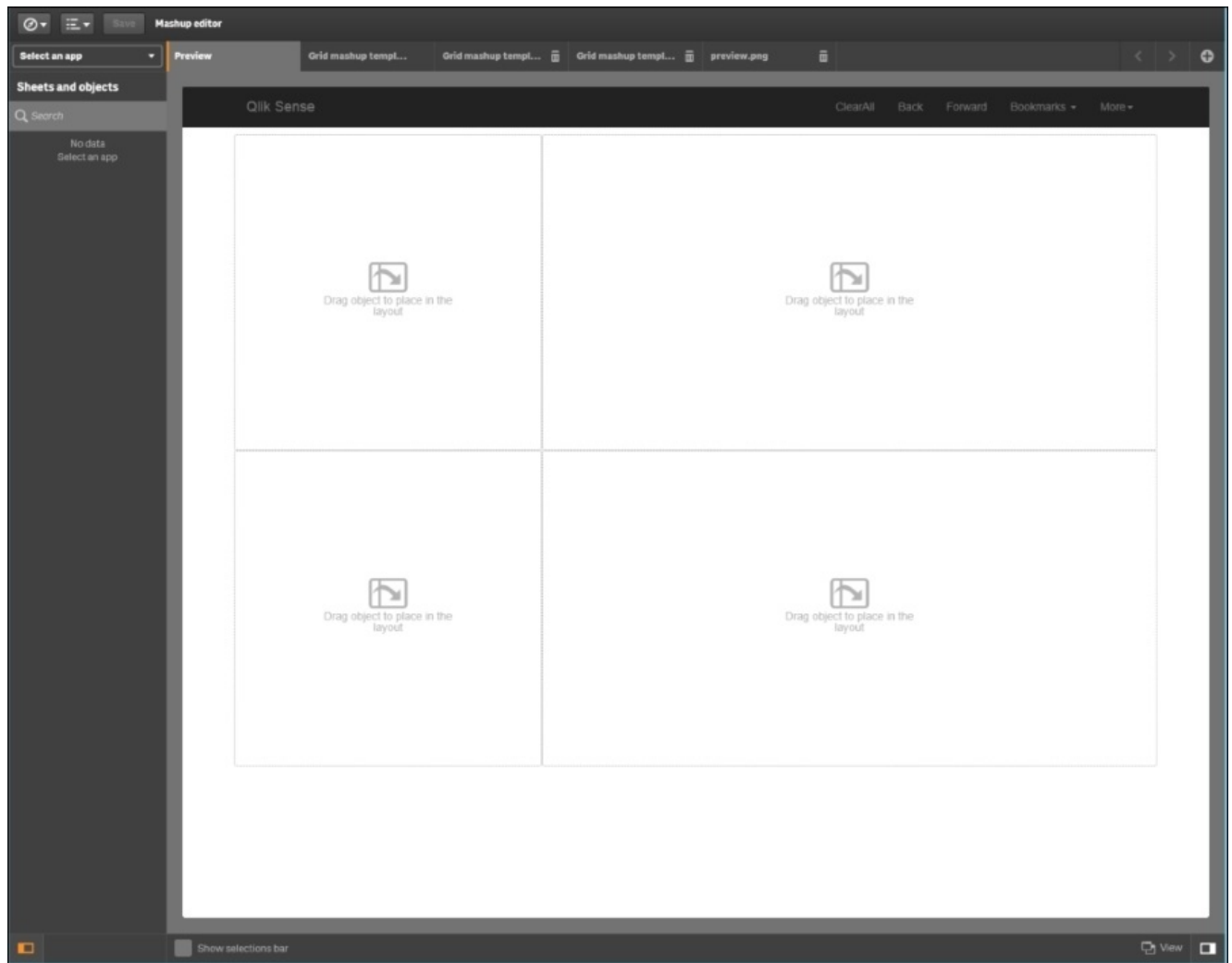
The resulting URL can then be embedded into a web page. Dev Hub offers a convenient **View** option so that a developer can see the results of the generated URL outside the **Single configurator** editor.



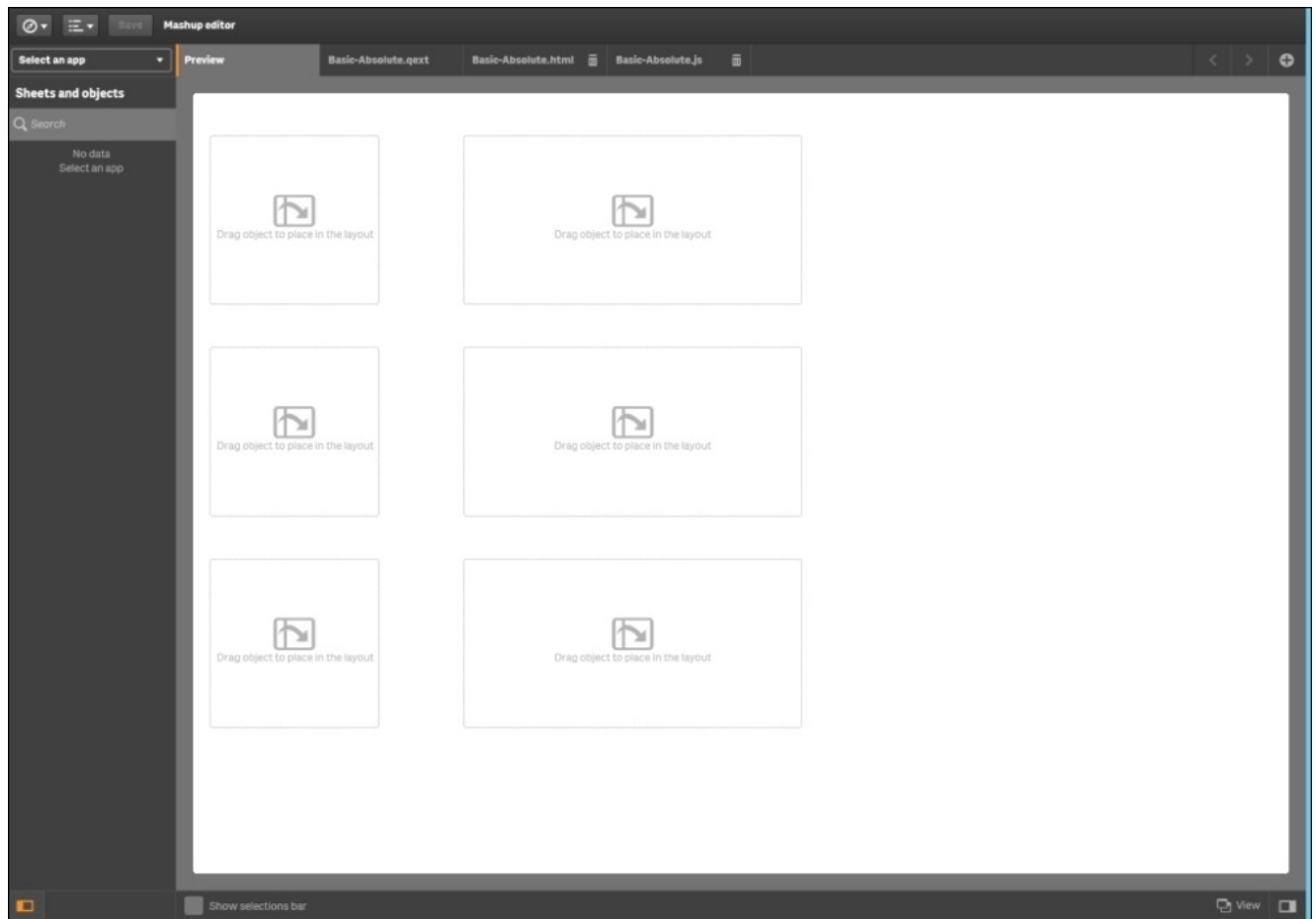
Additionally, multiple single Qlik Sense objects can be embedded and they will share common selections and interactions because of the QIX engine. Please note that not only can the visualizations be embedded, but the entire Qlik Sense sheets can be embedded as well. For more information, please refer to the *Qlik Sense for Developers help* site.

Now, let's move onto more complex mashup requirements with the **Mashup editor**. To start the creation of a new mashup, select **Create new** and immediately you will notice that the **Mashup editor** provides the following four pre-built templates to ease your development:

- Grid mashup template:

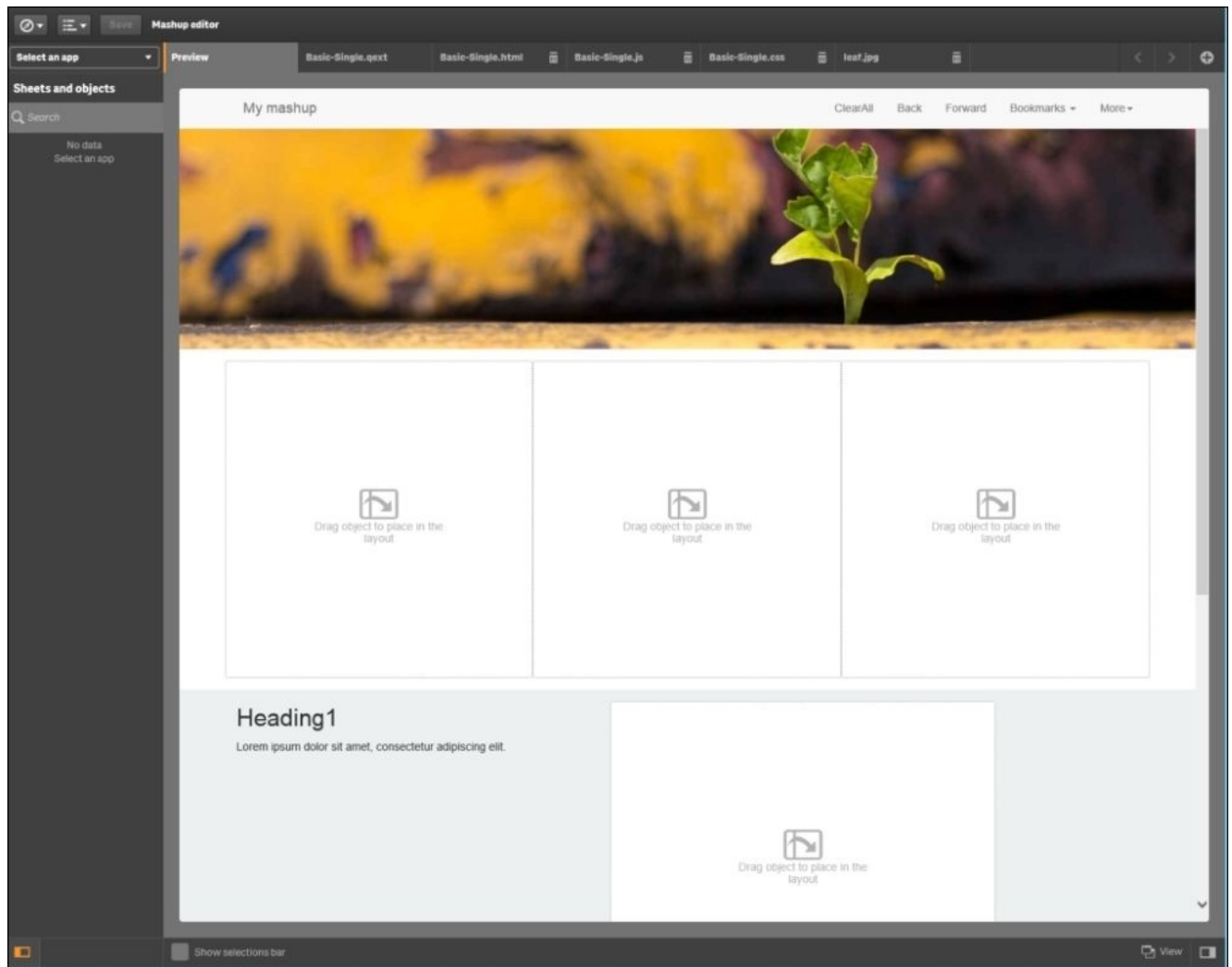


- Basic mashup template with absolute positioning:

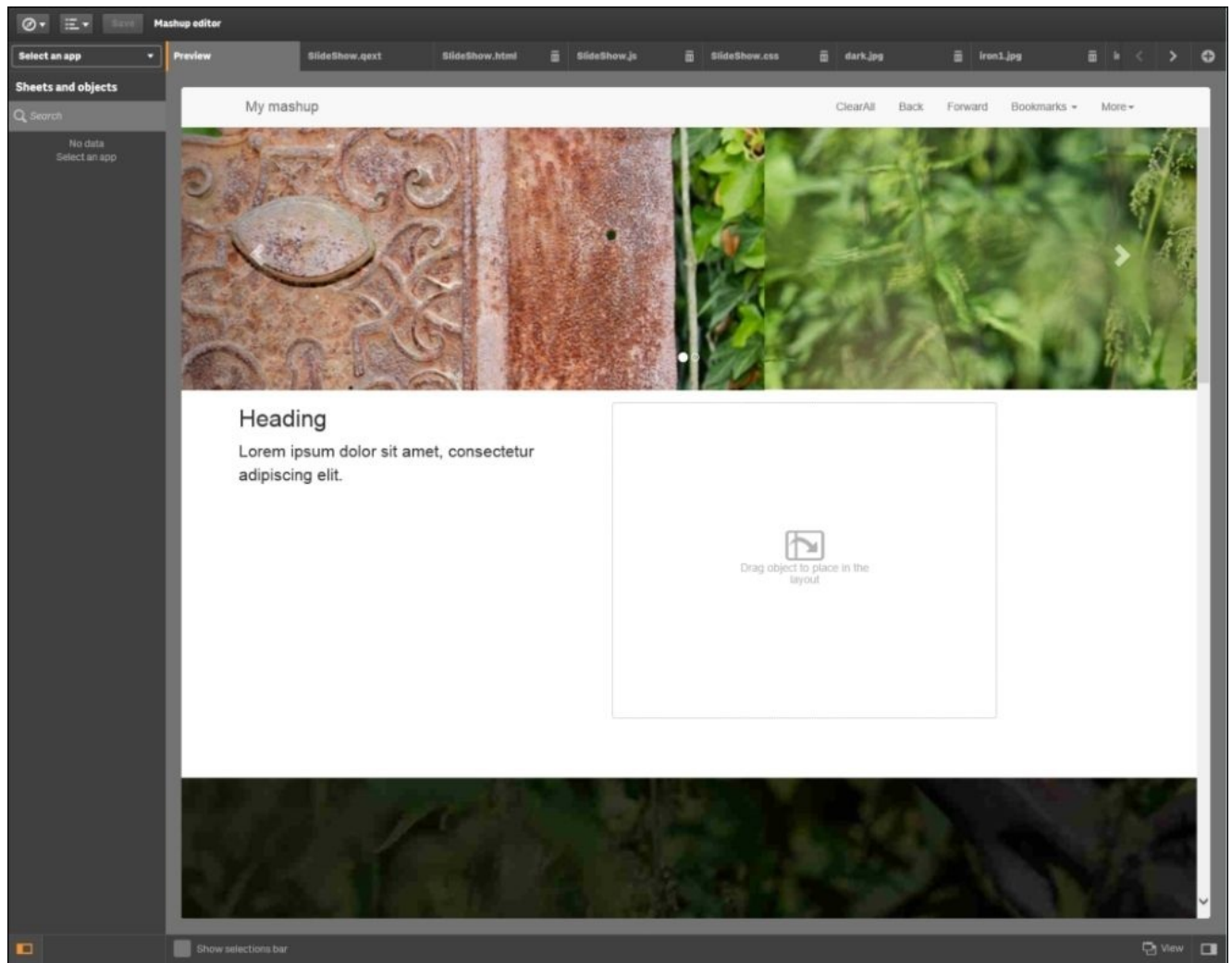


- Basic-single mashup:

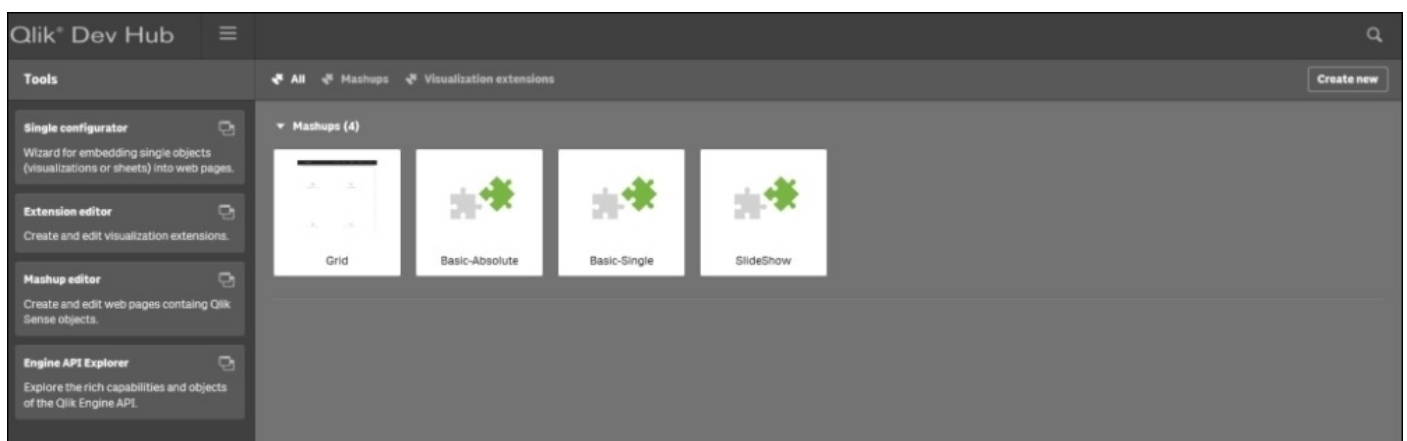




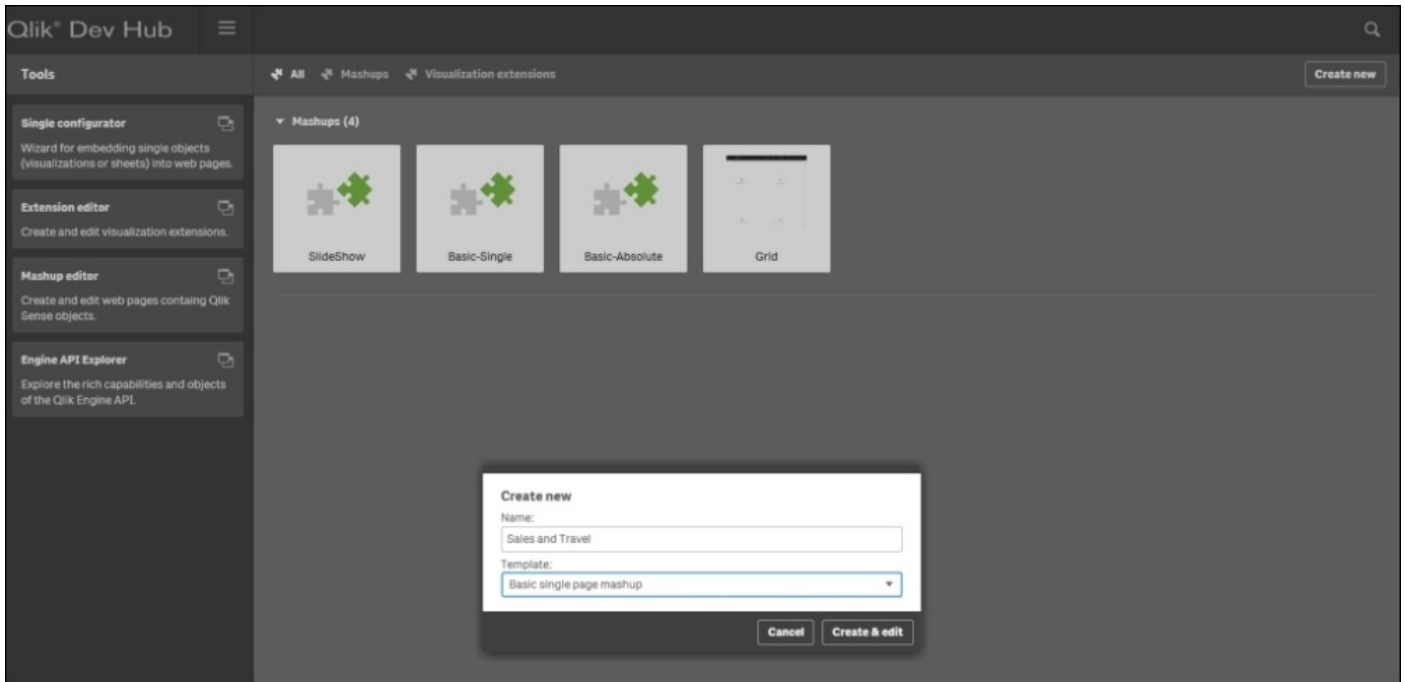
- SlideShow mashup



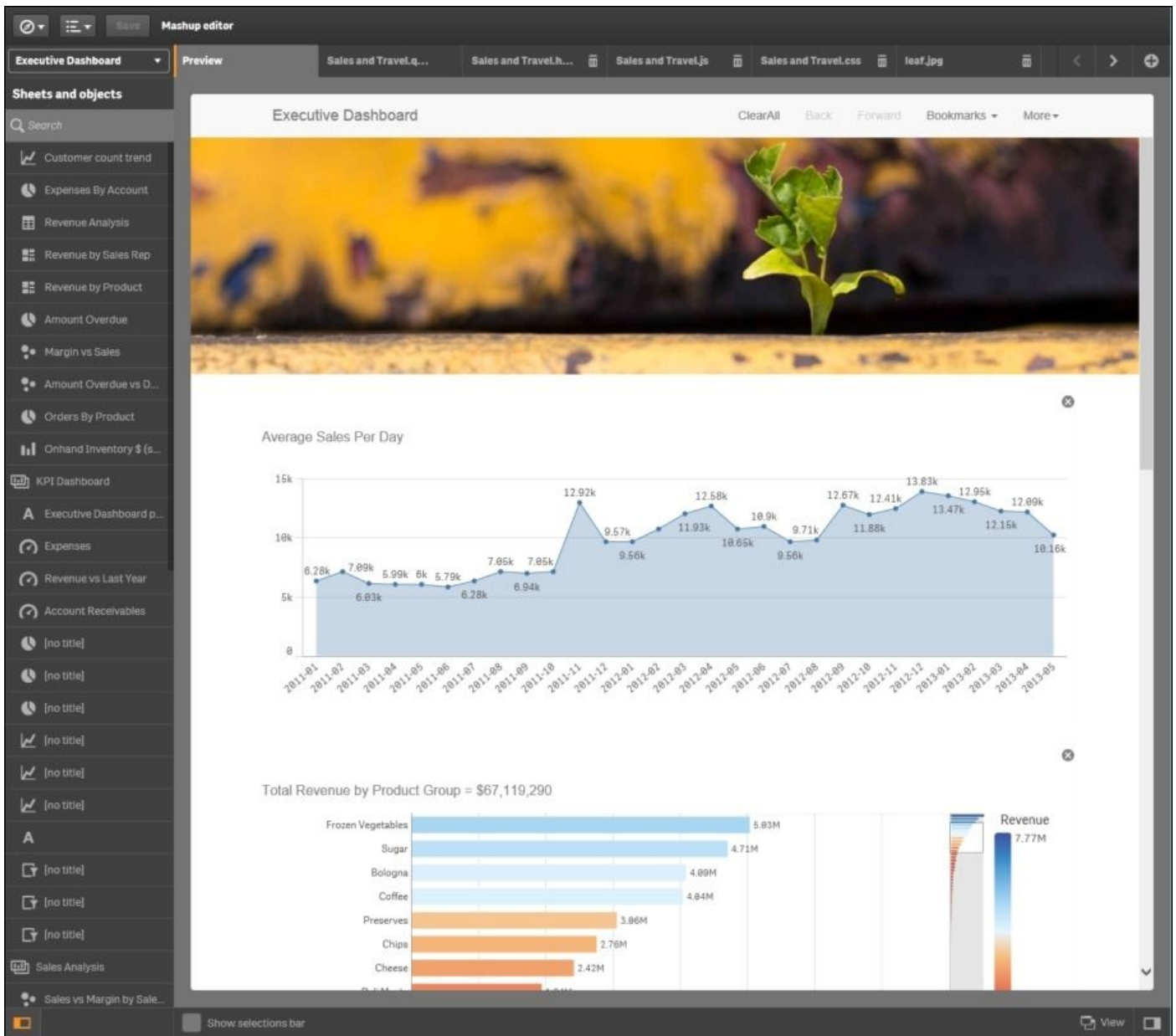
Once created, all these mashups are stored in the content store of the Qlik Sense server. Dev Hub provides an easy way to filter mashups versus visualization extensions. The following are the four templates that were generated:



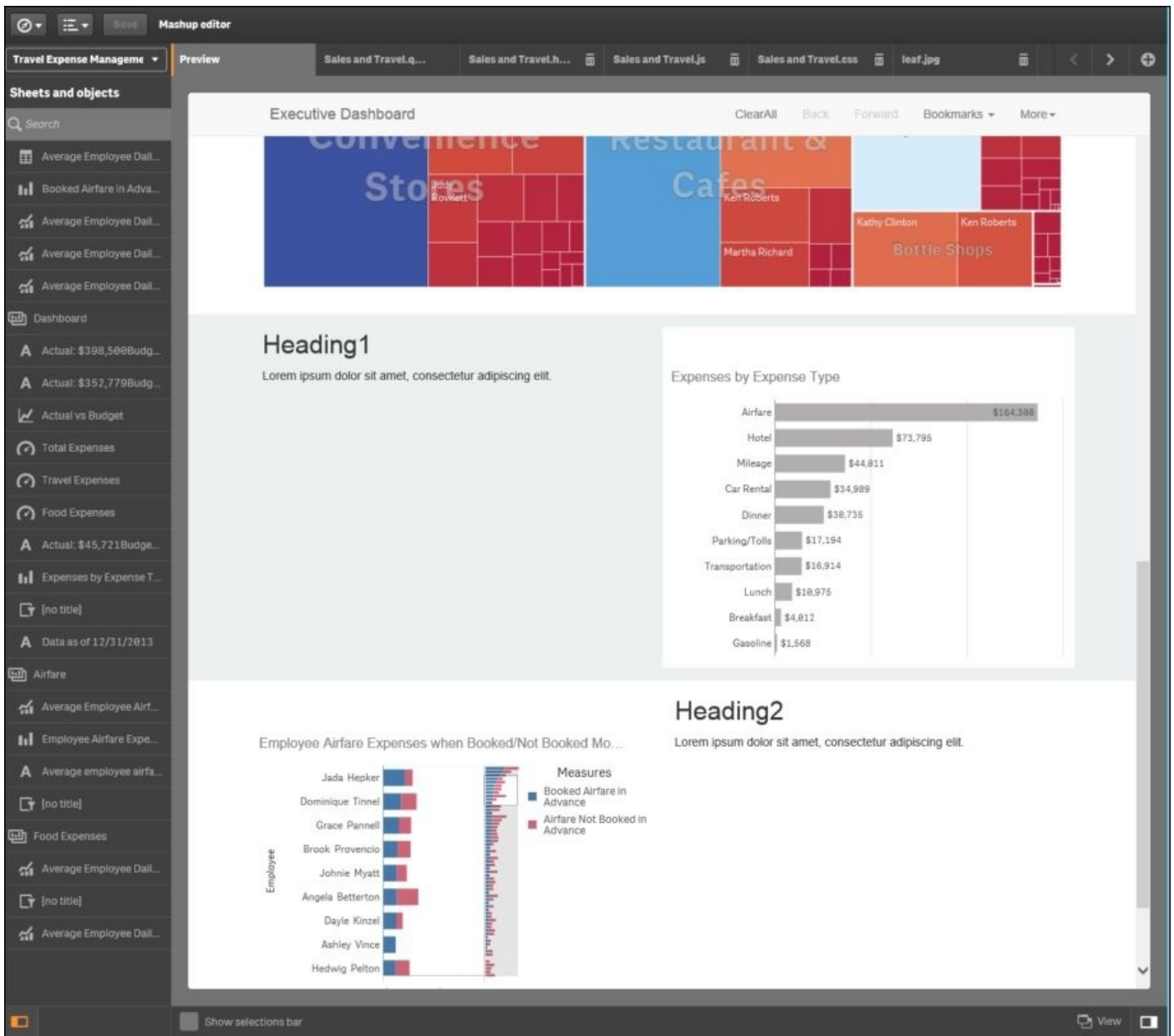
Now, let's dig a bit deeper as we build out the Basic-single mashup with the creation of a sales and travel mashup that spans the two Qlik Sense applications:



With the template created, we are ready to begin. One of the advantages of working in the Dev Hub is that you can create mashups that can span different Qlik Sense applications, in this case, the **Executive Dashboard** that contains sales analysis visualizations and **Travel Expense Management** which tracks travel and food expenses. From the **Executive Dashboard** app, we have dragged **Average Sales Per Day**, **Total Revenue by Product Group**, and finally, **Revenue by Sales Rep**:



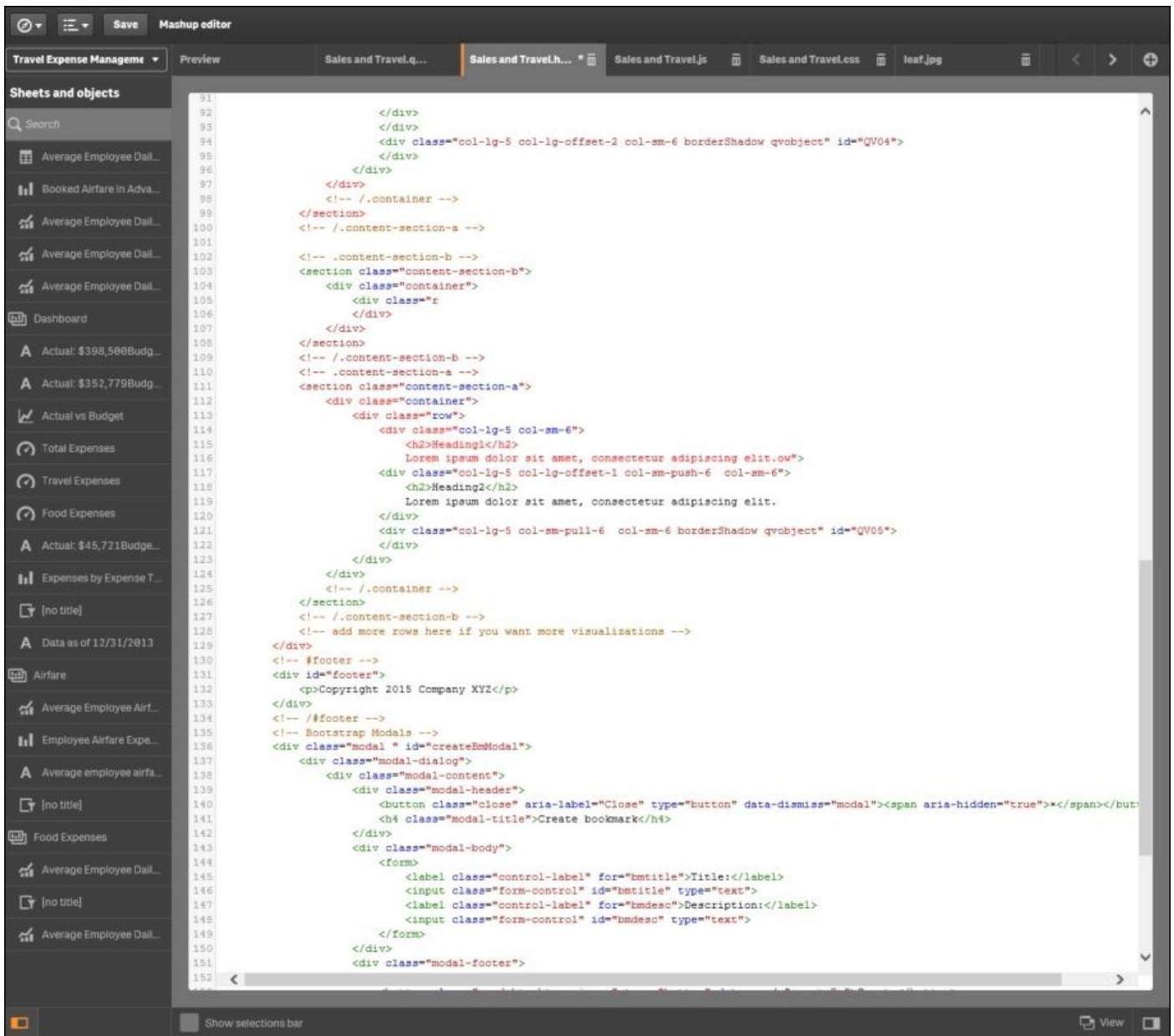
Now, let's combine these Qlik Sense sales objects with **Expense by Expense Type**, and **Employee Airfare Expense** from the **Travel Expense Management** app:



With the key Qlik Sense objects all in place, we can focus on changing the headings for the mashup page. Fortunately, the **Mashup editor** generates the following four files for this template. They include:

- Sales and Travel.qext
- Sales and Travel.html
- Sales and Travel.js
- Sales and Travel.css

To make changes to all the headers and other formatting, just edit Sales and Travel.html:



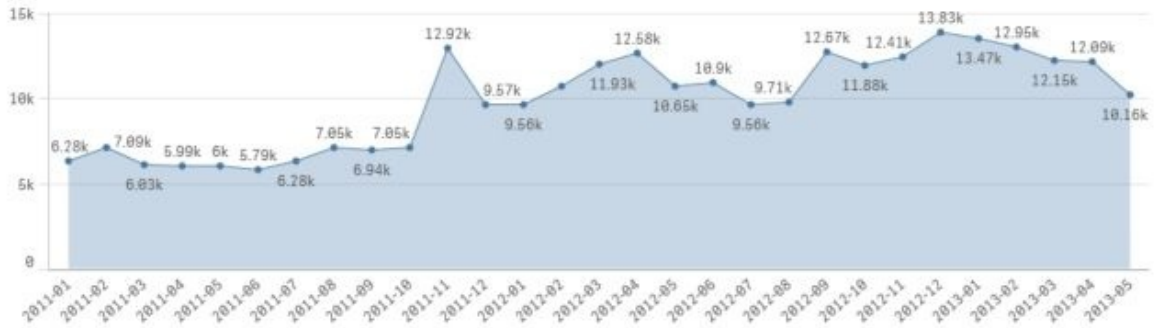
In the case of this example, the Header 1 and Header 2 titles were removed because the Qlik Sense app object titles were self-explanatory. Additionally, because this is a generated HTML file, it can be edited for additional formatting and content. The following is the resulting web page that is previewed from the **Mashup editor**:

Executive Dashboard

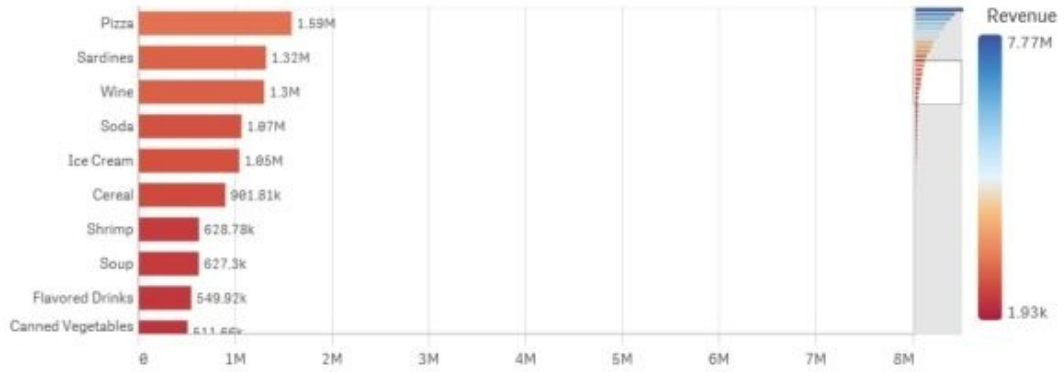
[ClearAll](#)
[Back](#)
[Forward](#)
[Bookmarks](#)
[More](#)



Average Sales Per Day



Total Revenue by Product Group = \$67,119,290



# Extending the Qlik Sense® client

Now, let's turn our attention to extending the visualization objects in Qlik Sense. One of the advantages of an open API is that it can be extended easily to include external visualizations. In our example, we will explore adding a People Chart to Qlik Sense. This example is available in the following locations:

- Qlik Sense Desktop: ... \Users\  
<UserName>\Documents\Qlik\Examples\Extensions
- Qlik Sense: ... \ProgramData\Qlik\Examples\Extensions

We start the process by reviewing the People Chart that is installed with Qlik Sense:

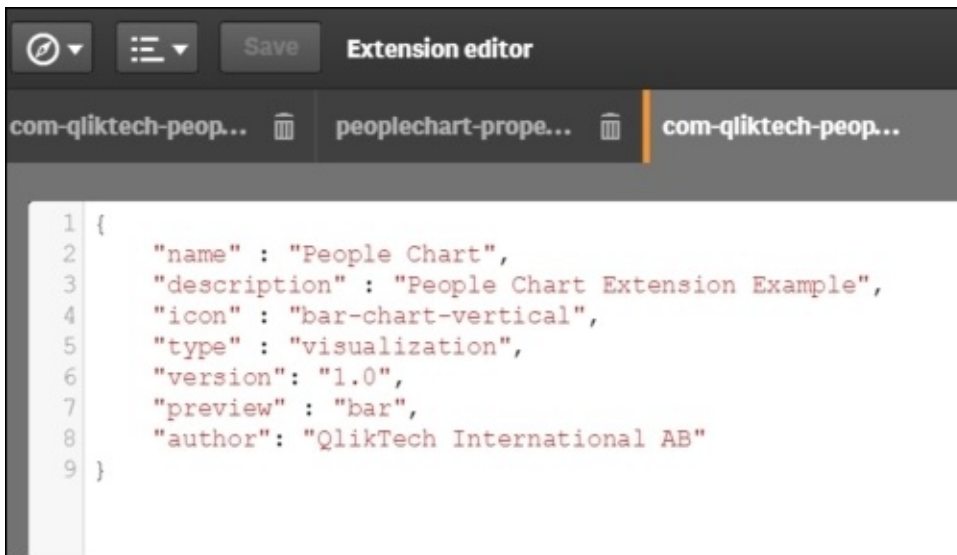


This visualization extension is made up of the following four files:

- `peoplechart-properties.js`: This JavaScript sets the properties in which the rendering and drawing scripts will operate
- `com-qliktech-peoplechart.js`: This JavaScript pulls together the properties, rendering, and drawing scripts for execution
- `com-qliktech-peoplechart.qext`: This file is primarily used to document the extension name, description, type, and so on
- `peoplechart.css`: Cascading Style Sheets (CSS) describes how HTML elements of the extensions will be displayed

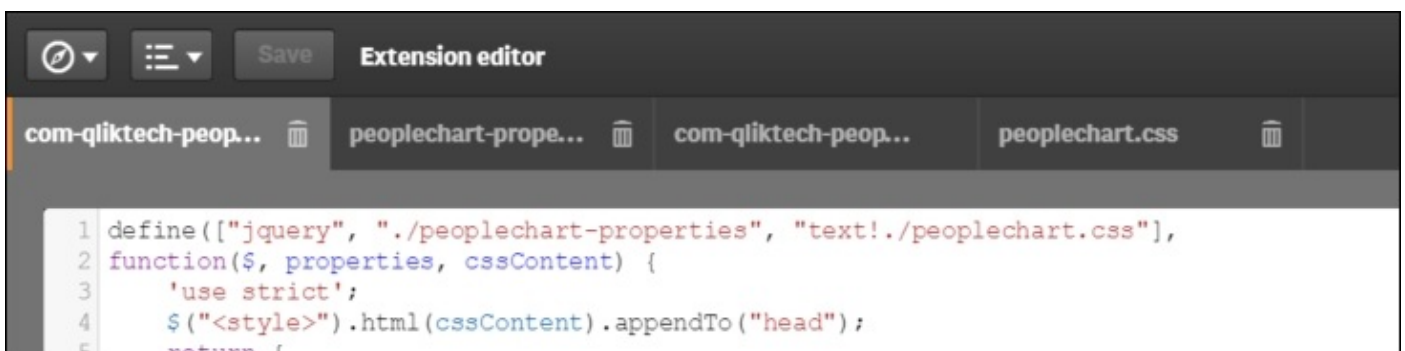
The first step in defining this extension is to edit the applicable fields of the `com-qliktech-peoplechart.qext` file to set a default title, description, icon, and type:





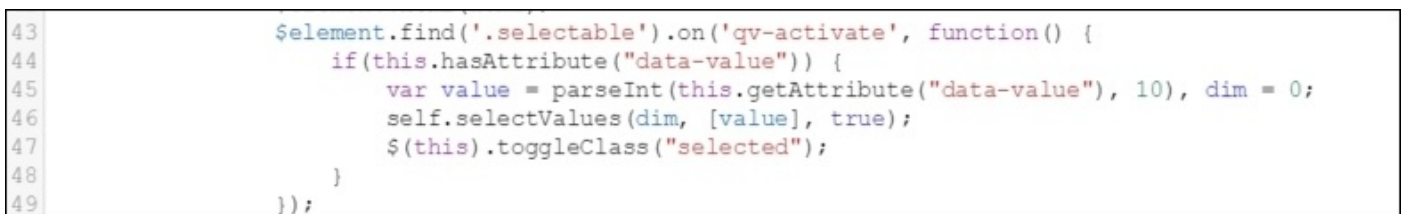
```
1 {
2   "name" : "People Chart",
3   "description" : "People Chart Extension Example",
4   "icon" : "bar-chart-vertical",
5   "type" : "visualization",
6   "version": "1.0",
7   "preview" : "bar",
8   "author": "QlikTech International AB"
9 }
```

Once this is completed, we need to define the properties for the extension in our Java code. In this example, we will do this in the `com-qliktech-peoplechart.js` file, which is loaded in the define statement:



```
1 define(["jquery", "./peoplechart-properties", "text!./peoplechart.css"],
2 function($, properties, cssContent) {
3   'use strict';
4   $("<style>").html(cssContent).appendTo("head");
5   return {
```

The next step is to enable selections. To accomplish this, we use the `selectValues` function to reuse Qlik Sense standard selection UI. Also, make sure to set the selected CSS class on the selected elements:



```
43     $element.find('.selectable').on('qv-activate', function() {
44         if(this.hasAttribute("data-value")) {
45             var value = parseInt(this.getAttribute("data-value"), 10), dim = 0;
46             self.selectValues(dim, [value], true);
47             $(this).toggleClass("selected");
48         }
49     });
```

Now that it is completed, we need to implement the paint method. In the paint method, we create the HTML for our extension based on the data in the layout parameter. Then, we set the content of the `$element` parameter to display the extension content. It is also important to tag your elements with `class= 'selectable' data-value= '0'`:

```

20 paint : function($element, layout) {
21
22     var self = this, html = "", measures = layout.qHyperCube.qMeasureInfo, w = $element.width() - 130, qData = layout.qHyperCube.qDataPages[0], vmax = (measures && measures[0]) ? measures[0].qMax * 1.5 : 1;
23     if(qData && qData.qMatrix) {
24         $.each(qData.qMatrix, function(key, row) {
25             if(row.length > 1) {
26                 //dimension is first, measure second
27                 var dim = row[0], meas = row[1];
28                 if(dim.qIsOtherCell) {
29                     dim.qText = layout.qHyperCube.qDimensionInfo[0].otherLabel;
30                 }
31                 html += '<div title="' + dim.qText + ' ' + meas.qText + '"';
32                 //negative element numbers are not selectable
33                 if(dim.qElemNumber > -1) {
34                     html += "class='selectable' data-value='" + dim.qElemNumber + '"';
35                 }
36                 html += '>';
37                 html += "<div class='label'" + dim.qText + "</div>";
38                 html += "<div class='bar' style='width:" + Math.round(w * (meas.qNum / vmax)) + "px;'><img alt='bar' style='width:" + Math.round(w * (meas.qNum / vmax)) + "px;'></div>";
39                 html += "</div>";
40             }
41         });
42         $element.html(html);

```

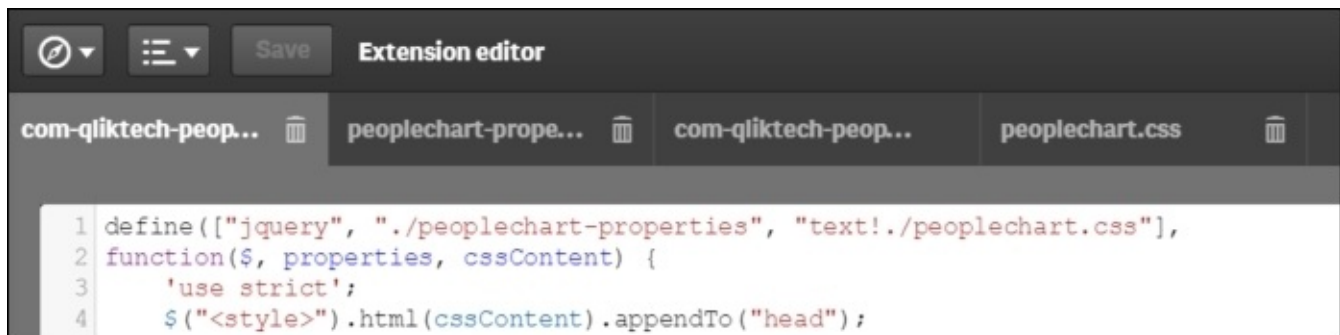
Additionally, let's make sure that this object is available for storytelling by setting `canTakeSnapshot` to `true`:

```

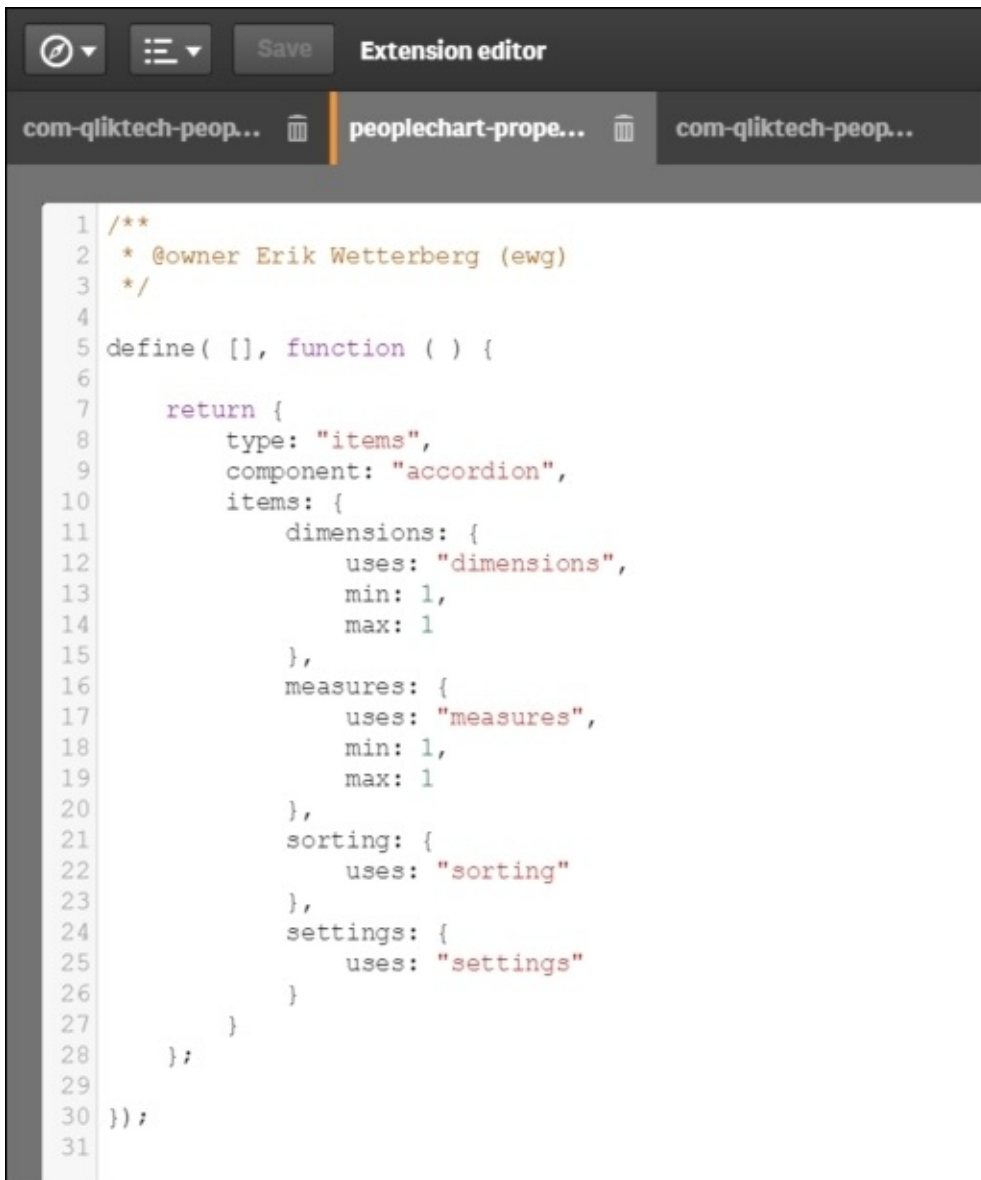
17     snapshot : {
18         canTakeSnapshot : true
19     },

```

It is a good programming practice to keep your styling in a separate CSS file. Qlik Sense sets the CSS class `qv-object-[extension name]` on your extensions. You should prefix your CSS rules with that. You then load your CSS file with RequireJS and add its content to the HTML page:

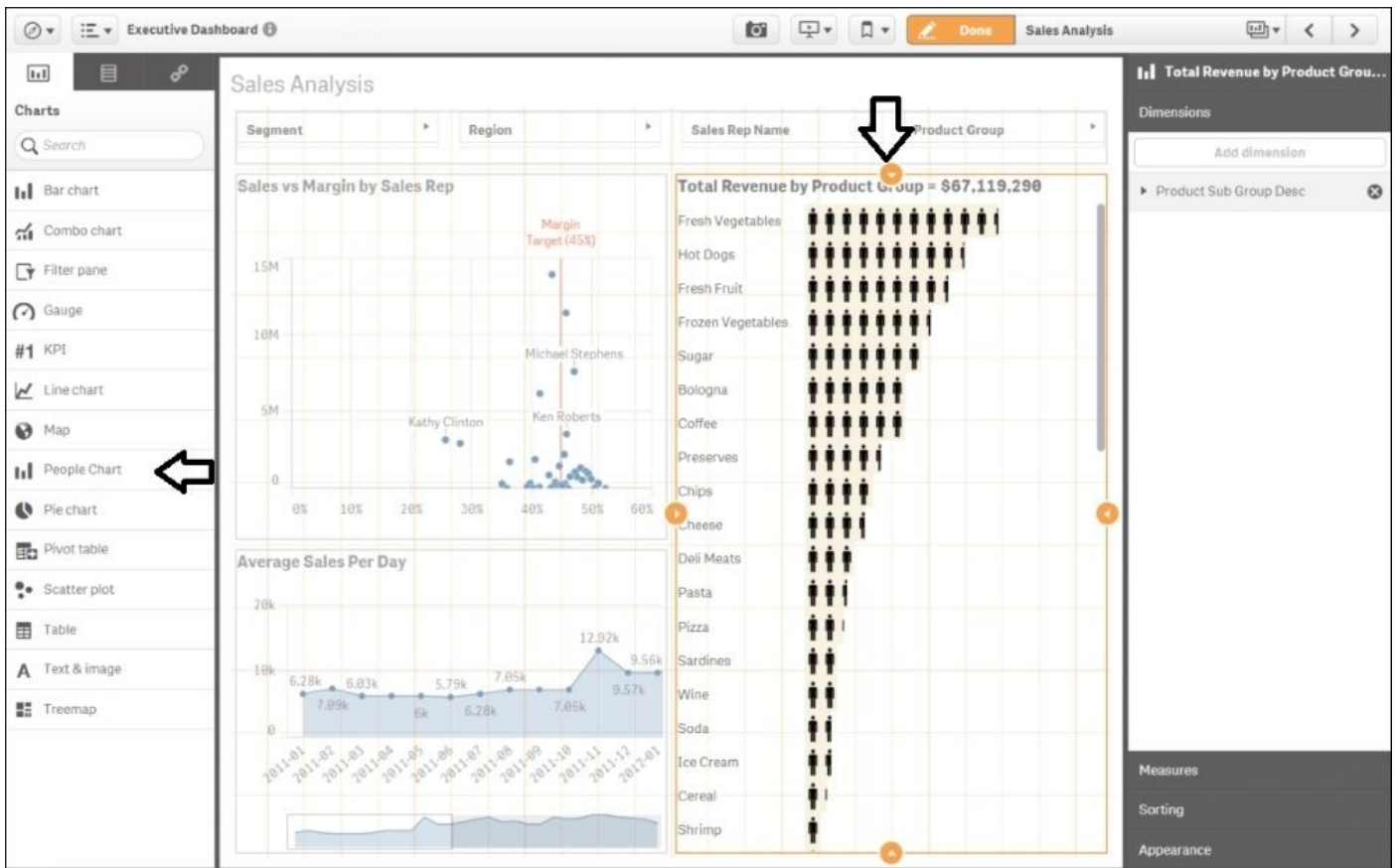


Finally, in the `peoplechart-properties.jss`, we need to define the accordions for reuse on the properties panel and the minimum requirements of one dimension and one measure to render this object:



```
1 /**
2  * @owner Erik Wetterberg (ewg)
3  */
4
5 define( [], function ( ) {
6
7     return {
8         type: "items",
9         component: "accordion",
10        items: {
11            dimensions: {
12                uses: "dimensions",
13                min: 1,
14                max: 1
15            },
16            measures: {
17                uses: "measures",
18                min: 1,
19                max: 1
20            },
21            sorting: {
22                uses: "sorting"
23            },
24            settings: {
25                uses: "settings"
26            }
27        }
28    };
29
30 });
31
```

Once the extension is saved, it is available for use by developers in the **Qlik Sense Chart Library**. In the following screenshot, you can see the Word Cloud chart type is available and was used to convert the horizontal bar chart of **Total Revenue by Product Group** into a Word Cloud:



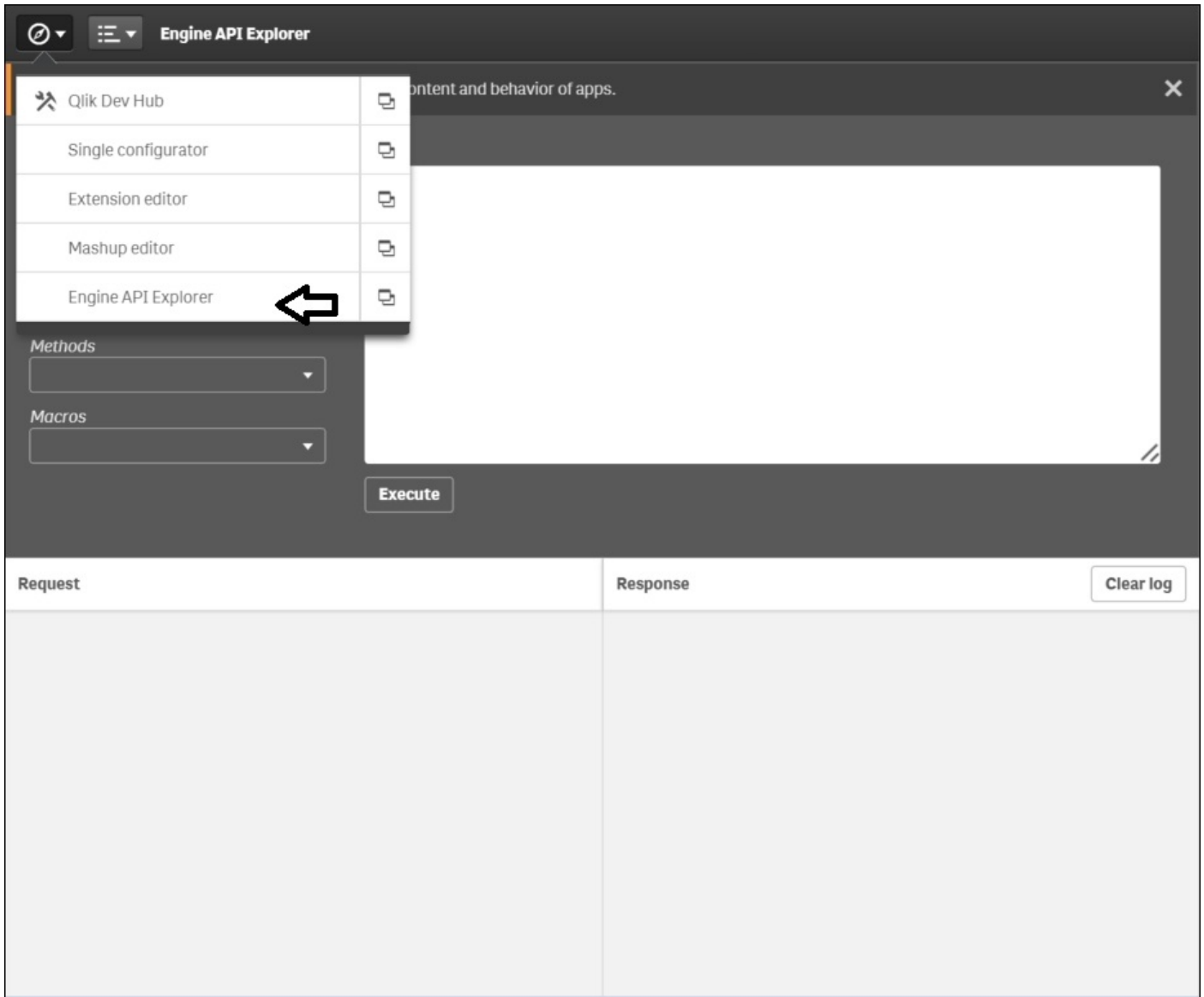
Additionally, please note that all extensions are stored in the Qlik Sense content store and is managed by the QMC. Additional information on administration is available in [Chapter 9, Administering Qlik Sense®](#).

Name	Owner	Tags
basic-template	sa_repository (INTERNAL\sa_repository)	
com-qliktech-peoplechart	administrator (SENSESERVER0\administrator)	
gridmashup-template	sa_repository (INTERNAL\sa_repository)	
mashup-template	sa_repository (INTERNAL\sa_repository)	
plant-template	sa_repository (INTERNAL\sa_repository)	
slideshow-template	sa_repository (INTERNAL\sa_repository)	

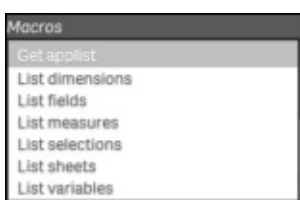
Now, let's turn our attention to the final area of Dev Hub, that is, **Engine API Explorer**.

# Engine API Explorer

Qlik **Engine API Explorer** is a tool that allows developers to send messages and receive answers from the QIX engine. This provides an easy way for developers to form handles, methods, and macros to send the QIX engine and test the results before coding an application. To access the **Engine API Explorer**, select the **Engine API Explorer** from the **Qlik Dev Hub** menu:



The first step is to connect to a Qlik Sense app. To do this, we first must know what applications are available on this server. Fortunately, there is a full list of macros available to help explore QIX Engine:



In our case, we will be using the Get applist to see what applications are available to explore on this server. Based on the response, the **Executive Dashboard** is available:

The screenshot shows the Engine API Explorer interface. On the left, the 'Send' tab is active, displaying a REST client request:

```
{
  "name": "GETAPPLIST",
  "handle": "-1",
  "method": "GetDocList",
  "params": []
}
```

The 'Methods' dropdown is set to 'Get applist'. Below the request, the 'Request' tab shows the raw JSON:

```
{
  "handle": "-1",
  "method": "GetDocList",
  "params": [],
  "id": 3,
  "jsonrpc": "2.0"
}
```

The 'Response' tab shows the JSON response from the server:

```
{
  "jsonrpc": "2.0",
  "id": 3,
  "result": {
    "qDocList": [
      {
        "qDocName": "Executive Dashboard",
        "qConnectedUsers": 0,
        "qFileTime": 0,
        "qFileSize": 0,
        "qDocId": "4a1dcc87-a0dd-4ad3-934c-ae25253e5679",
        "qMeta": {
          "modifiedDate": "2015-11-18T14:43:11.703Z",
          "published": true,
          "publishTime": "2015-11-18T14:43:07.537Z",
          "privileges": [
            "read"
          ]
        },
        "description": "This Qlik Sense application was developed for the Executive team to monitor and analyze key metrics of company performance. This application is updated monthly. For additional information please review the Application Story",
        "dynamicColor": "",
        "create": null,
        "stream": {
          "id": "38904d82-e4c3-4205-b82c-794a952bd670",
          "name": "BI Center of Excellence"
        },
        "canCreateDataConnections": false
      },
      {
        "qDocName": "Executive Dashboard-v2",
        "qConnectedUsers": 0,
        "qFileTime": 0,
        "qFileSize": 0,
        "qDocId": "88089685-7065-430b-bdde-7e585d424c50",
        "qMeta": {
          "modifiedDate": "2015-11-18T14:42:29.963Z",
          "published": false,
          "publishTime": "1753-01-01T00:00:00.000Z",
          "privileges": [
            "read",
            "update",
            "delete"
          ]
        },
        "description": "This Qlik Sense application was developed for the Executive"
      }
    ]
  }
}
```

A black arrow points to the first item in the 'qDocList' array, which is the 'Executive Dashboard' application.

Now we are ready to connect to the **Executive Dashboard** app. Note the send commands generated and the engine response. If satisfied with the response, the developer can then copy and paste into their development environment:

Engine API Explorer

**NOTE:** The Engine API explorer might change the content and behavior of apps. ✕

App  
 Executive Dashboard Connect

Handles

Methods

Macros

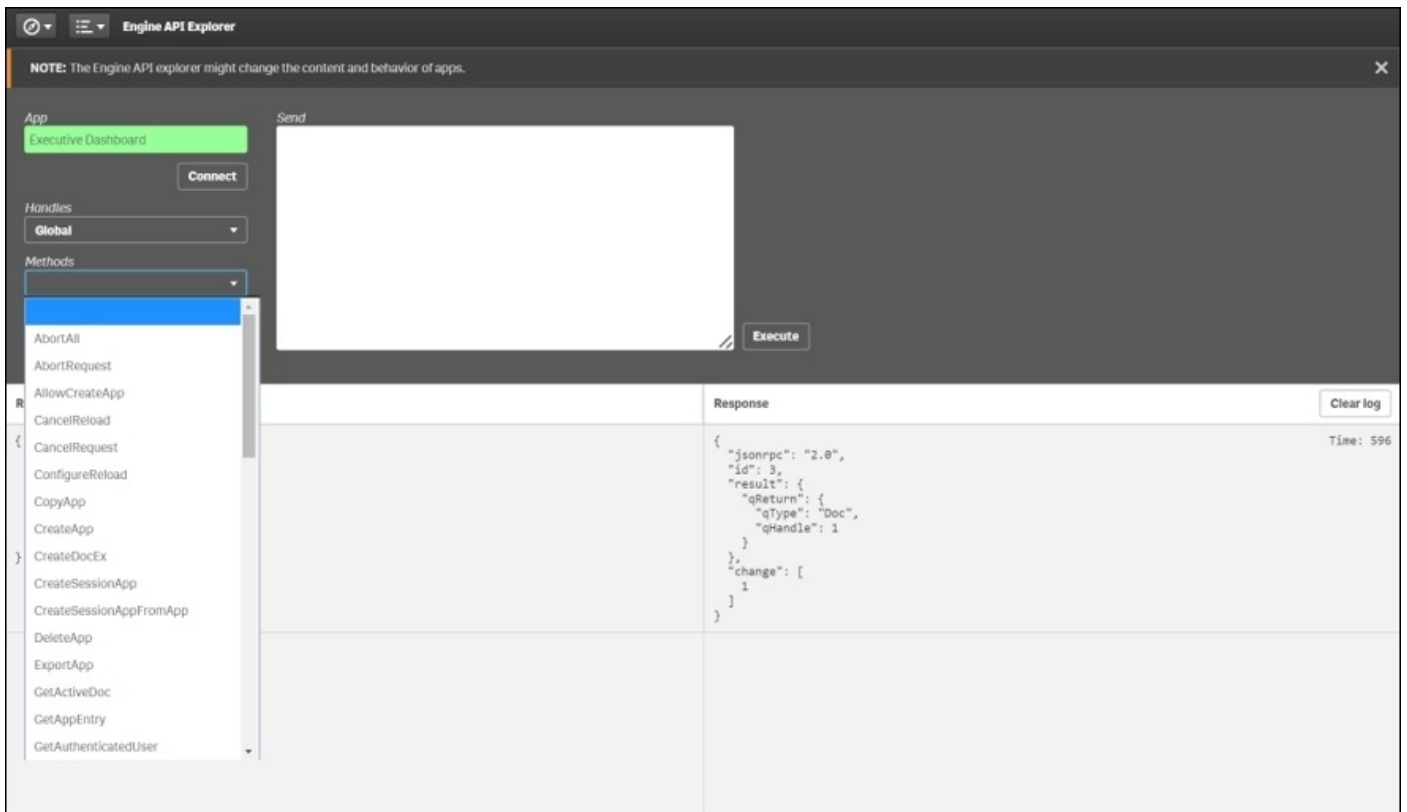
Send

Execute

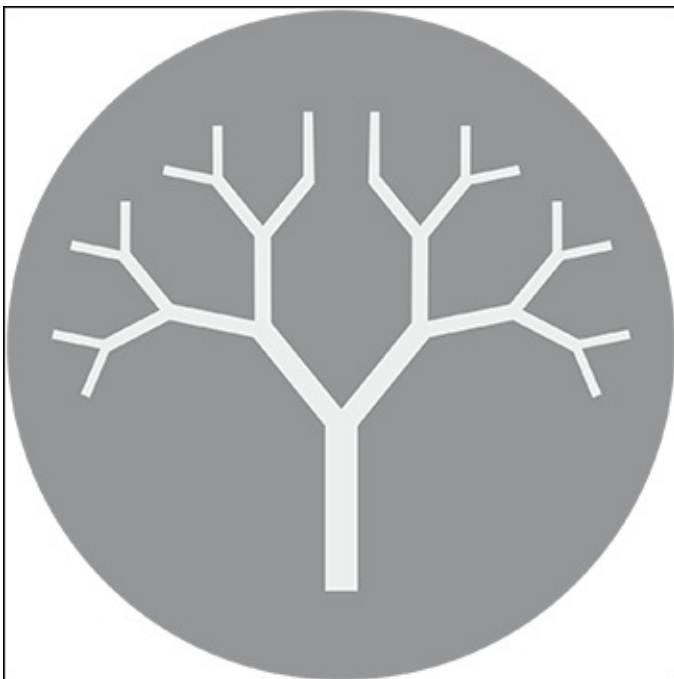
Request	Response <span>Clear log</span>
<pre>{   "method": "OpenDoc",   "handle": -1,   "params": [     "Executive Dashboard"   ],   "id": 7,   "jsonrpc": "2.0" }</pre>	<pre>{   "jsonrpc": "2.0",   "id": 7,   "result": {     "qReturn": {       "qType": "Doc",       "qHandle": 1     }   } }</pre> <p>Time: 35</p>

Now that we are connected, there is a global method to manage every aspect of a Qlik Sense app from outside the Qlik Sense client. This includes the full application life cycle from creation, maintenance, versioning, to deletion:





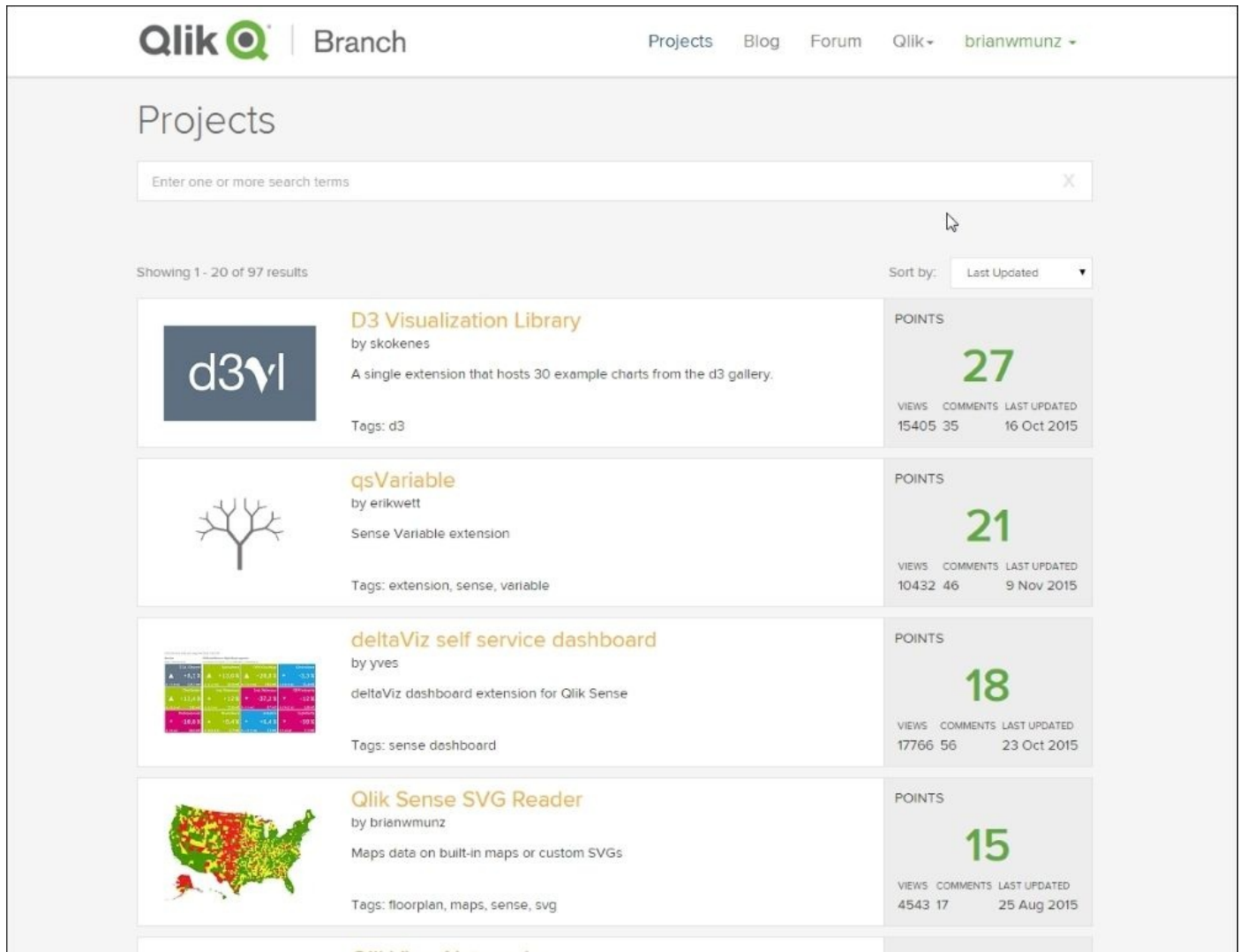
Now that we have explored the Qlik Analytic Platform through the Dev Hub, let's turn our attention to the value of having an open API through a developers community called Qlik Branch:





# Developer community – Qlik Branch

Qlik Branch (<http://branch.qlik.com>) is an open source community specifically designed for *developers*. It is a place to share and collaborate on projects and innovations created with Qlik products with an open source philosophy. All projects posted are required to have the code readily accessible, and they must be downloadable directly from the site. In short, everything on the site is free to use and free to modify in the spirit of open source:



The screenshot shows the Qlik Branch website interface. At the top, there is a navigation bar with the Qlik logo and the word "Branch". To the right, there are links for "Projects", "Blog", "Forum", and a user profile for "brianwmunz". Below the navigation bar, the main heading is "Projects". There is a search bar with the placeholder text "Enter one or more search terms". Below the search bar, it says "Showing 1 - 20 of 97 results". To the right of the results, there is a "Sort by:" dropdown menu set to "Last Updated". The main content area displays a list of projects, each with a thumbnail image, a title, author, description, tags, and a statistics box. The statistics box includes "POINTS", "VIEWS", "COMMENTS", and "LAST UPDATED".

Project Title	Author	Description	Tags	Points	Views	Comments	Last Updated
D3 Visualization Library	by skokenes	A single extension that hosts 30 example charts from the d3 gallery.	d3	27	15405	35	16 Oct 2015
qsVariable	by erikwett	Sense Variable extension	extension, sense, variable	21	10432	46	9 Nov 2015
deltaViz self service dashboard	by yves	deltaViz dashboard extension for Qlik Sense	sense dashboard	18	17766	56	23 Oct 2015
Qlik Sense SVG Reader	by brianwmunz	Maps data on built-in maps or custom SVGs	floorplan, maps, sense, svg	15	4543	17	25 Aug 2015

Additionally, Qlik Branch is a place to find and download solutions for your projects and/or network with the developers to help extend your Qlik Sense or QlikView solutions. The site contains a wide variety of projects leveraging the APIs of Qlik Sense, including visualizations, web mashups, server automation, and connectors, to name just a few.

As a developer, Qlik Branch is a great place to get started and involved in the growing community. Many of the projects on the site could serve as a great starting place for development efforts. Additionally, a developer can stay up to date with API-related news, ask questions in the forum, join the public Slack channel, as well as educate themselves with the resources currently available or planned for the future. Furthermore, a more enterprising developer could make a name for themselves by sharing their expertise and

creating valuable content, potentially driving business their way as the customers look to implement:

**Qlik** | Branch

Projects Blog Forum Qlik- brianwmunz

## Brian Munz

brianwmunz Joined July 16, 2014

Sr. Manager of Developer Relations, including Qlik Branch, Hackathons, and anything in between. I have a web dev background primarily in Pharma, e-commerce, and now B.I. & Analytics. My first Qlik extension was an ugly Gantt chart using Flash, and I can only hope that it has been lost and forgotten forever.

Projects (17) Blog (1)

Enter one or more search terms

Showing 1 - 10 of 11 results Sort by: Last Updated

<b>Grid Chart Extension</b> Tags:d3js, interactive, selections	Updated 16 Oct 2015	Summary A Qlik Sense extension that mimics the QlikView 11 grid chart.
<b>D3 Visualization Library</b> Tags:d3	Updated 16 Oct 2015	Summary A single extension that hosts 30 example charts from the d3 gallery.
<b>Filter Tree Panel</b> Tags:filters, listbox, qv11	Updated 13 Oct 2015	Summary A scrollable, expandable list of filters in QV11.
<b>senseUtils</b> Tags:library, web	Updated 24 Sep 2015	Summary A JavaScript library that helps build web solutions on top of the Qlik Sense APIs
<b>Hover Drill Bar Chart with D3</b> Tags:animation, d3js, drill, hover	Updated 27 Aug 2015	Summary A bar chart that can drill between two dimensions by hovering over the axis.
<b>Magnetic Field Visualization</b> Tags:dogs	Updated 24 Jun 2015	Summary An example visualization of magnetic fields built with QV11 out of the box charts.
<b>Utilization Planner</b> Tags:d3js, interactive, schedule	Updated 24 Jun 2015	Summary An interactive bar chart for scheduling at a weekly

Flag Unapprove  
Delete Edit

LOCATION & CONTACT  
Kimberton  
PA  
USA

@brianwmunz  
http://branch.qlik.com  
bmz@qlik.com

COMPANY  
Qlik

We look forward to you joining the thousands of developers who have joined together in building world class visual analytic solutions.



# Summary

In summary, one of the strengths of Qlik Sense is that it is built on open API 's Qlik Analytic Platform (QAP) that allows customers and partners to extend their analytic solutions. This chapter has provided an overview and some interesting examples of how to enrich your solutions with QAP. It is not meant to replace the *Qlik Sense for Developers help* documentation which can be found at <https://help.qlik.com/sense/2.1/en-us/developer/#Home-developer.htm>, but rather serve as an introduction for power users who seek to expand their skill sets, as well as developers who are new to Qlik Sense.

In the next chapter of this book, we will explore the key features of administrating your Qlik Sense environment.



# Chapter 9. Administering Qlik Sense®

Having established how to develop attractive and engaging applications with Qlik Sense, it's time to turn our attention from authors and business users. Instead, we will consider the requirements of administrators. In this chapter, we will move away from data and analysis to what's needed to run a Qlik Sense installation.

In this chapter, you will find information about the following topics:

- Architecture
- Clustering and nodes
- Licenses and tokens
- Streams and security concepts



# The Qlik Sense® architecture

Qlik Sense has an architecture that is different from the QlikView Server architecture. Some components are very similar; others are very different. Hence, even if you know the QlikView architecture, you need to look at the following sections. In them, you will find an overview of some of the concepts in Qlik Sense.

# Services

When you install the Qlik Sense server, you will install seven services. These are the cornerstones of the architecture. They can be deployed in different ways to suit different deployment purposes.

The Qlik Sense services are as follows:

- **Qlik Sense Engine Service (The QIX engine):** This is the application service, which handles all application calculations and logic. Everything that concerns the data analysis is handled by this service.
- **Qlik Sense Printing Service:** This manages the Qlik Sense exports, reporting, and printing. This is new for Qlik Sense Version 2.
- **Qlik Sense Proxy Service (QPS):** This manages the Qlik Sense authentication, session handling, and load balancing.
- **Qlik Sense Repository Service (QRS):** This manages persistence of apps and the synchronization of licensing, security, and service configuration data.
- **Qlik Sense Repository Database:** This service runs a PostgreSQL database used by the QRS.
- **Qlik Sense Scheduler Service (QSS):** This manages the scheduled reloads of Qlik Sense apps as well as other types of events, for example, task chaining.
- **Qlik Sense Service Dispatcher:** This is a service controller that is used to launch and manage additional Qlik Sense services.

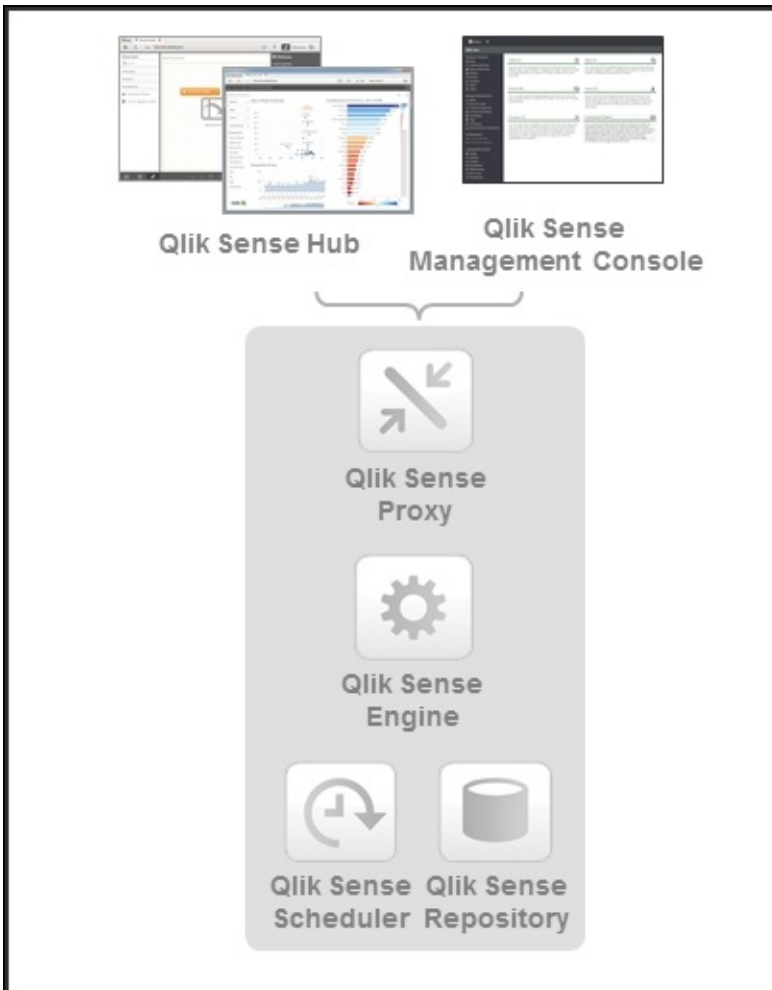
In a standard installation, all seven services run on the same computer, and this works fine as long as the load on the server doesn't become too heavy.

The services can run under any account, but should preferably run under an account dedicated to the Qlik Sense services.

# Clients

Qlik Sense has two different clients: the hub and the management console (QMC).

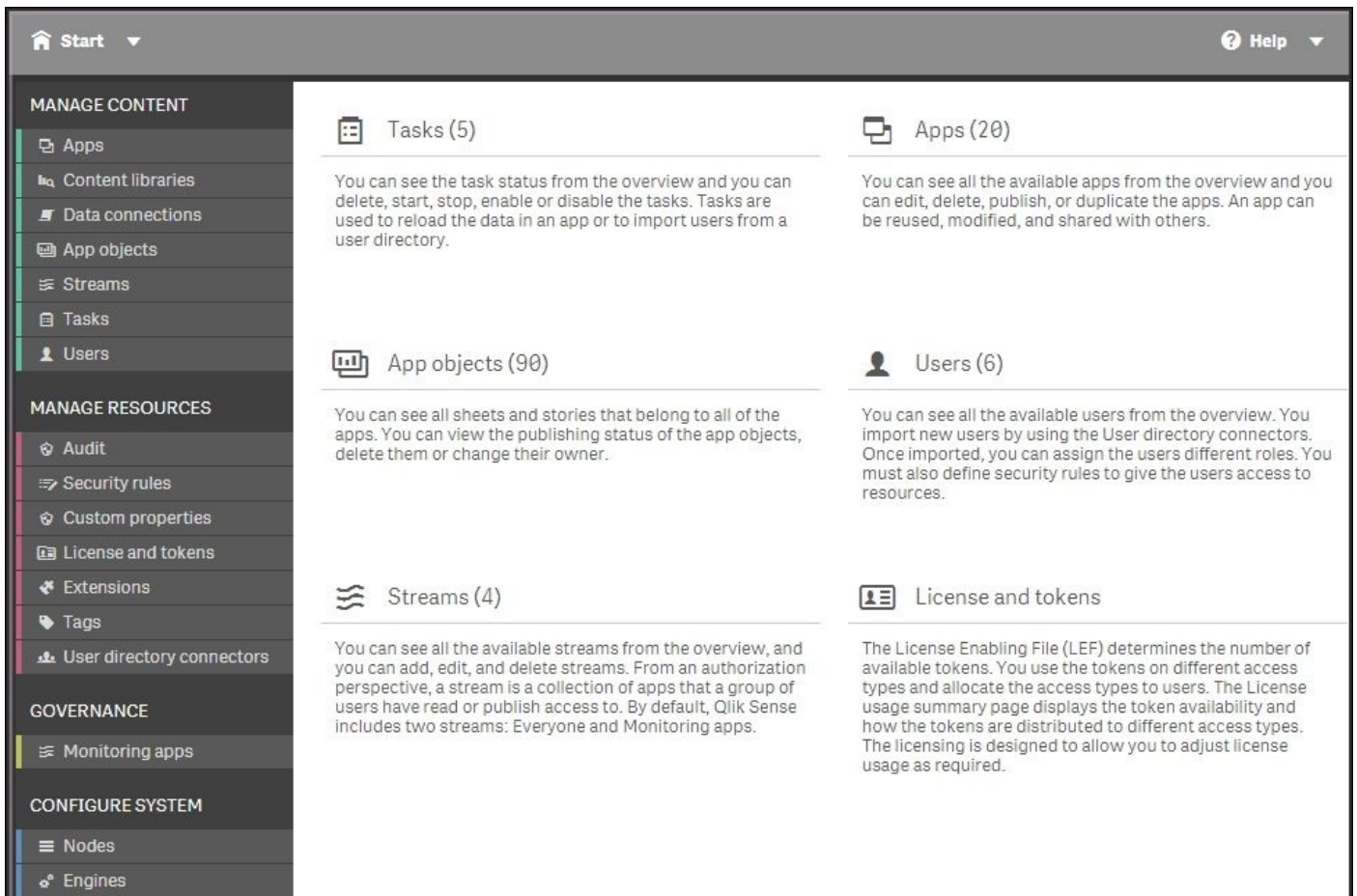
The hub is used to access, edit, and publish apps. It always runs in a web browser, regardless of whether you use a desktop computer, tablet, or smartphone to access it.



*The basic Qlik Sense architecture*

**Qlik Management Console (QMC)** is used for all types of administration. QMC is a web page found at [https://<computer\\_name>/qmc/](https://<computer_name>/qmc/).

A link to this is installed in your Start menu during the installation.



*The QMC start page*

In QMC, you can manage and monitor everything for your installation: apps, streams, security, users, and so on.

To the left, you have the four main groups: tools to manage the content, tools to manage resources, tools for governance, and tools to configure the system.

The QMC is a multiuser environment, designed for the delegation of administration of, for example, streams to authors, if this fits a company's work process.

# Applications

The apps are subject-specific; files that contain data, prepared visualizations, load script, and so on. This is where the analysis is done. From a user's perspective, an app is organized into sheets, sheet objects (visualizations), bookmarks, and stories. An app can be private or published to a stream.

If you want to access an app to do analysis, you can access it through the hub. However, if you want to perform any administrative task, such as importing or publishing an app, you can do it through QMC.

# Nodes

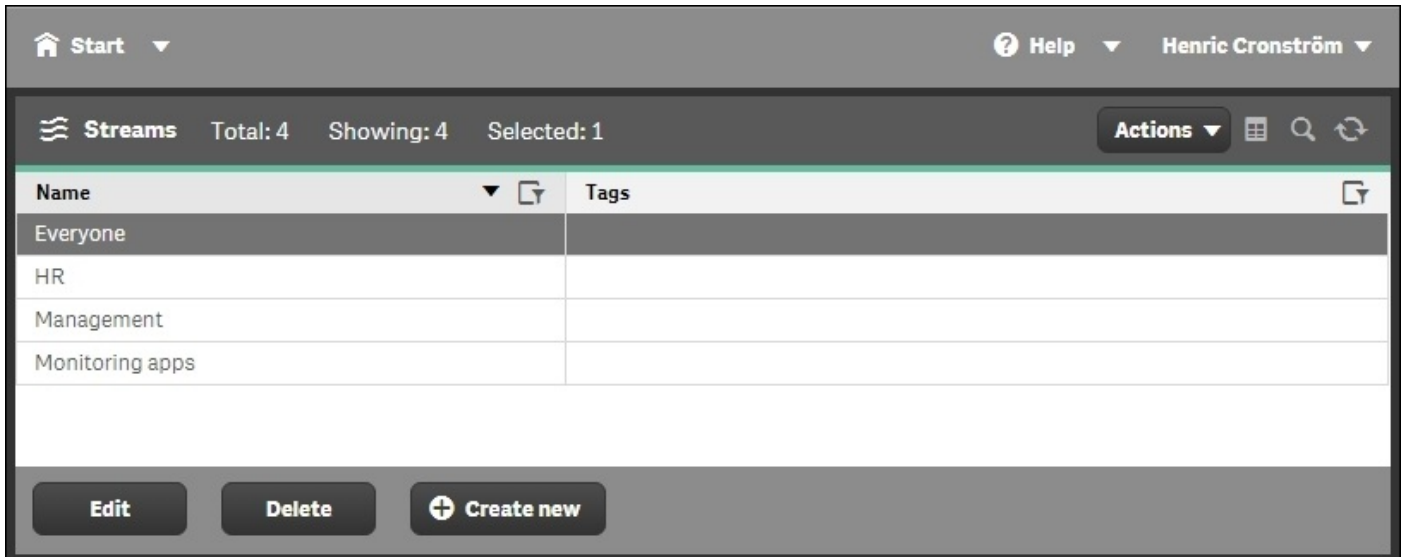
Qlik Sense's site has an architecture that allows a distributed deployment. In other words, you can have several computers, each with a Qlik Sense installation, that work together and are managed as one coherent server. In such a configuration, each computer is called a **node** and the entire installation is called a **cluster**.

The installation can be configured so that data is synchronized between the different nodes, and so that the appropriate server is used for the client request. The purpose is, of course, to increase the system resilience and deployment flexibility.

# Streams

The next important concept in Qlik Sense is **streams**. A stream is a dynamic, collaborative workgroup that is used when publishing applications. Hence, when you publish an app to a stream, you publish it to a group of people.

A stream has members, security rules, and tags. It enables the user to read or publish apps, sheets, and stories. The users who have publishing rights to a stream create the content for that specific stream, and the users who have read access are the consumers of the apps.



Name	Tags
Everyone	
HR	
Management	
Monitoring apps	

Buttons: Edit, Delete, Create new

*The Streams sheet*





# Deployment and licensing

Deploying a Qlik Sense server is usually straightforward, but there are still a couple of things to think of. The first question is about clustering.

# Single node or multinode

Normally, you should just install the Qlik Sense server, making sure that it is set as a central node during the installation. Then, you will get a single node installation.

However, sooner or later you need to ask yourself a question about clustering, “How many servers do you want in your cluster?” This book is not a comprehensive guide to clustering issues; it will only point out the basics and the questions you need to ask.

In a standard installation, all seven services run on the same computer, and this works fine as long as the load on the server doesn’t become too heavy. However, as soon as your installation starts to grow, you may need more computers to handle the load. If so, you can set up a cluster so that you have additional computers running only some of the services and still manage the entire cluster as if it was only one computer.

By far, the most common case is that Qlik Sense engine (the QIX engine) has a very large load, either due to many users or because some applications are large. Then, it might be a good idea to add one or several computers and use a separate Qlik Sense engine on each computer.

Another case is that you have several physical locations and want one node in each location, with the same content on each node. This way, the users always use the local node.

One of the computers must be set as the central node, that is, as the master. Here, you enter your license key and manage the entire cluster. Data will be synchronized from this node to other nodes.

It is possible to use the same entry point—the Qlik Sense proxy—for the entire cluster so that users don’t notice that there are, in fact, several computers.

On the central node, it is recommended that you have a dedicated QPS and QIX that are used specifically for the QMC and not for the hub.

In addition, the central node must have the QSS installed even if other nodes with schedulers are added. This is because the scheduler on the central node is considered to be the master scheduler, which coordinates all scheduler activities within the site.

Hence, when defining your deployment strategy, you should try to answer some questions:

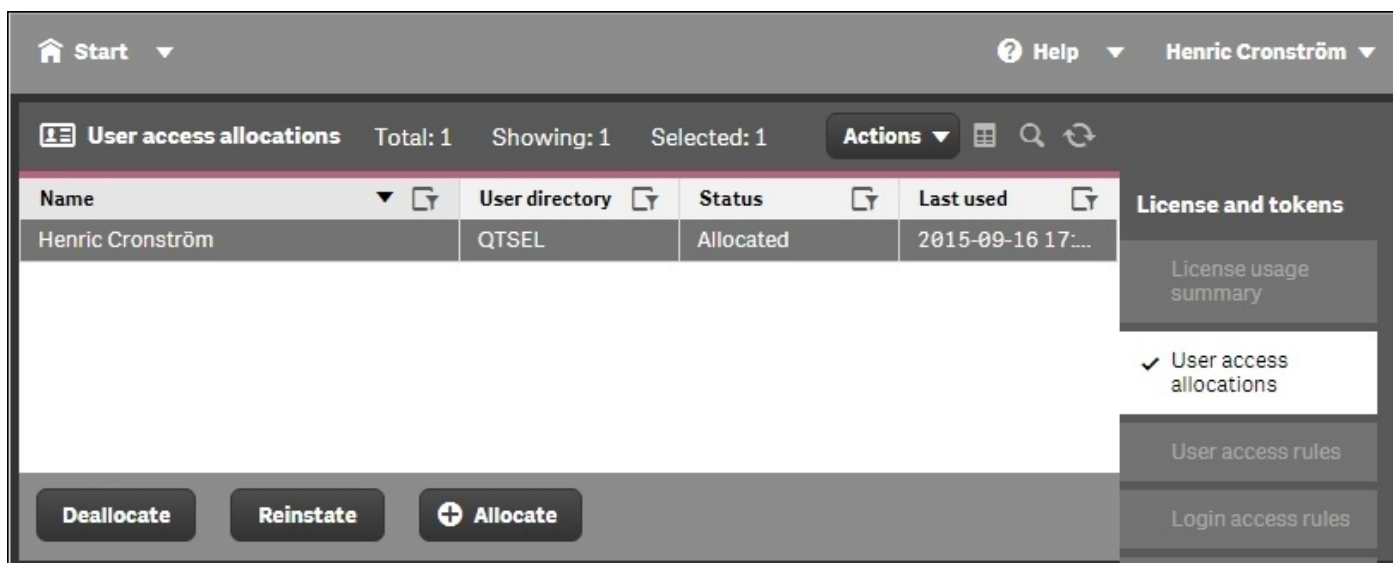
- What is the estimated number of computers needed to handle the number of apps and users you expect?
- Should the users use the same proxy so that you can set up rules for load balancing? Or should they use different entry points in the different locations?
- Do you want/need separate computers that are only used to run jobs, for example, to refresh the apps?

The answers will help you decide whether you need a cluster of Qlik Sense servers. If you don’t know, or if it is your first server in a cluster, you should just install the Qlik Sense server making sure that it is set as the central node during the installation.

# License and access passes

The first thing you need to do after installing Qlik Sense is to enter the license key and make sure that you get a valid **License Enabling File (LEF)** from the Qlik license activation server. However, this is not enough to get going. You also need to assign a license to yourself. Alternatively, to express this in the correct terminology, you need to *allocate a token as a user access pass for yourself*.

This is done by navigating to **License and tokens | User access**. Here, you can click on the **Allocate** button, select a user, and click on **Allocate**. This means you have given this user unlimited access to the Qlik Sense server. Unlimited means unlimited from a license perspective—you may still define restrictions on this user from a security perspective.



The screenshot displays the 'User access allocations' page in Qlik Sense. At the top, there is a navigation bar with 'Start', 'Help', and the user name 'Henric Cronström'. Below this, the page title is 'User access allocations' with statistics: 'Total: 1', 'Showing: 1', and 'Selected: 1'. There are icons for 'Actions', a grid, search, and refresh. A table lists the allocation details:

Name	User directory	Status	Last used
Henric Cronström	QTSEL	Allocated	2015-09-16 17:...

At the bottom of the table, there are three buttons: 'Deallocate', 'Reinstate', and 'Allocate'. On the right side, there is a sidebar titled 'License and tokens' with options: 'License usage summary', 'User access allocations' (checked), 'User access rules', and 'Login access rules'.

*User access has been granted to one of the authors*

# Tokens

The reason for this procedure is Qlik Sense's flexible licensing model. In the QlikView license model, you bought Named CALs or some other license from Qlik, and that was then the license you had. To convert from one type of CAL to another was not possible, unless you contacted Qlik.

In the Qlik Sense model, you have a greater degree of freedom. Here, *you* decide how you want to allocate the licenses you bought. Some are allocated to the equivalent of a Named CAL, while some are allocated to another license type. As a consequence, you don't buy licenses. Instead you buy **tokens**, which is a kind of currency that you can convert into licenses at a later stage. In the initial configuration, no tokens are assigned to be used and hence, the need to allocate a token to yourself.

Another consequence is that the terminology has changed. A Named CAL is no longer called Named CAL. Instead, it is called a **user access pass**. So in the preceding case, you have effectively given yourself a Named CAL.

The Qlik Sense user access pass—and the QlikView Named CALs for that matter—is a general unlimited license that should be given to frequent users, that is, users that analyze data regularly, many times every month or perhaps even daily.

In Qlik Sense, there is a second license type that is designed to cover the needs of infrequent users. It is similar to the Usage CAL that exists in QlikView. It is called **login access pass**. A login access pass is equivalent to one login per month, that is, the login counter is refreshed so that a new login is possible every 28th day.

You can create login access passes in batches of 10, and 10 login access passes cost 1 token. These 10 logins can be used anyway you want. They can, for example, be used by 10 different people that log in once per month or by one single user who logs in 10 times every month.

The screenshot shows the 'Login access rules' configuration page in Qlik Sense. At the top, there are navigation elements: 'Start', 'Help', and the user name 'Henric Cronström'. Below this, the page title is 'Login access rules' with summary statistics: 'Total: 3', 'Showing: 3', and 'Selected: 1'. An 'Actions' menu is visible. The main content is a table with the following data:

Name	Allocated tokens	Used login access ...	Remaining login ac...
Finance department	3	0	30
Guests	3	0	30
HR	1	0	10

At the bottom of the table, there are three buttons: 'Edit', 'Delete', and 'Create new'. To the right of the table is a sidebar titled 'License and tokens' with several menu items: 'License usage summary', 'User access allocations', 'User access rules', 'Login access rules' (which is checked), and 'Site license'.

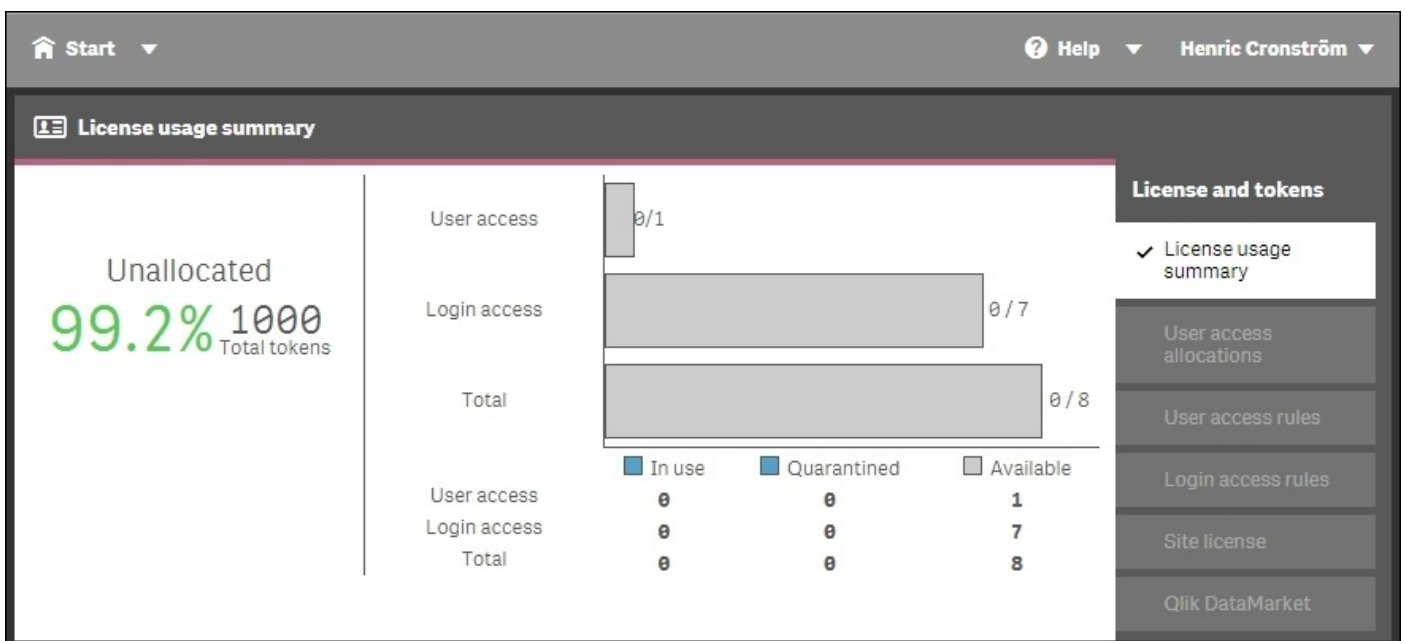
*Login access passes can be created in pools for different groups of people*

# Access rules

Typically, you would use several tokens to create a pool of logins that can be dedicated for a group of people within your domain. You can create access rules both for user access passes and login access passes, and you should do this for your own benefit. This means that you can, for instance, say that anyone in the finance department will get a user access pass when logging on, whereas the users of another department will share the login access passes of a specific pool.

Hence, when you create the user and login accesses you want, you can get an overview of the **License usage summary** page, where you can clearly see the number of used tokens and how many you have left to allocate.

You can clearly see how many tokens you allocated to licenses, and how these are distributed over the two access types in the following screenshot:



*The License usage summary in QMC*

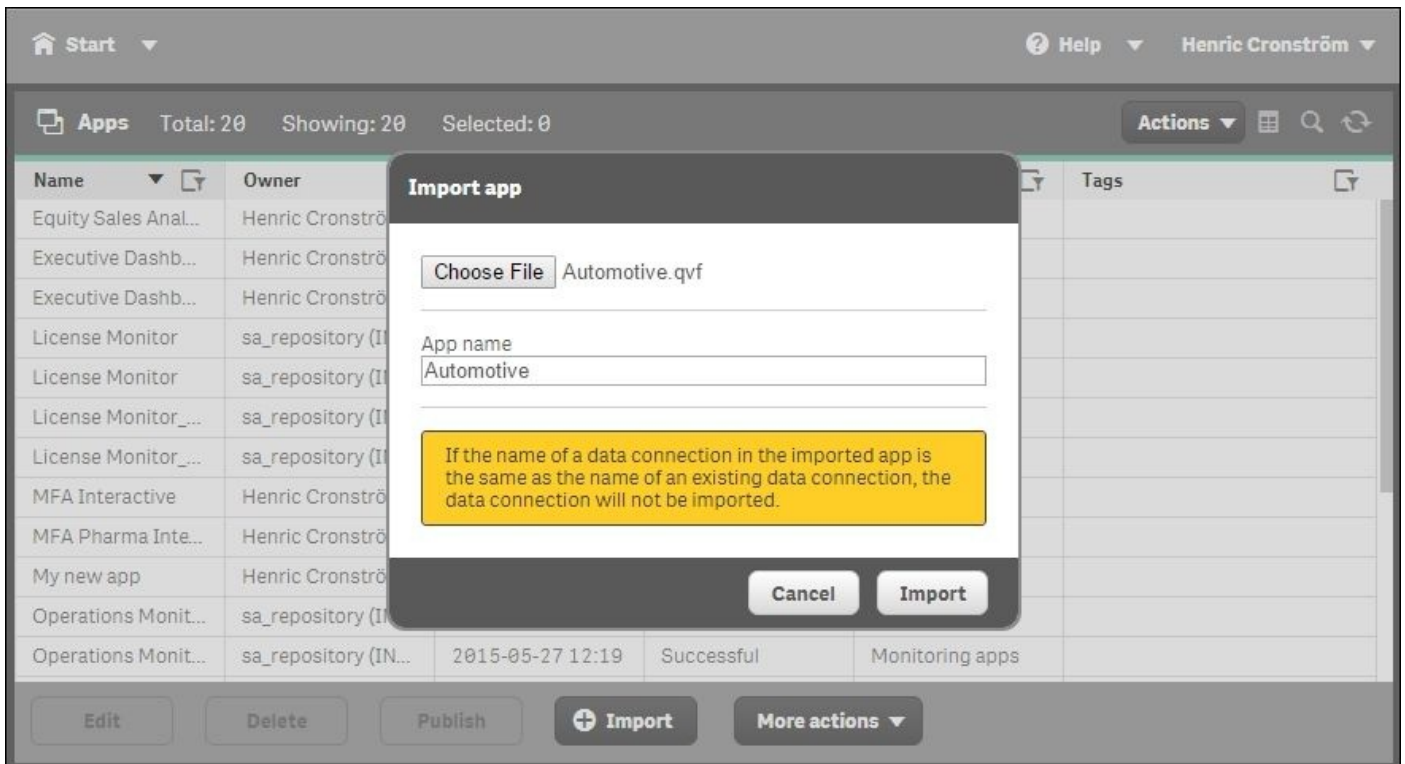


# Management and monitoring

So far, this chapter has dealt with managing the installation and the licenses, but very little has been mentioned about the real purpose of the Qlik Sense administration, which is, How to handle data and the analysis of data? How to handle the applications, the users, the data connections, the distribution, and so on? This section will cover these areas.

# Importing and managing apps

Once the Qlik Sense server is deployed, you might want to import an app that you have created in Qlik Sense Desktop. This is something that you can do in the **Apps** sheet. Look for the **Import** button at the bottom of the screen.

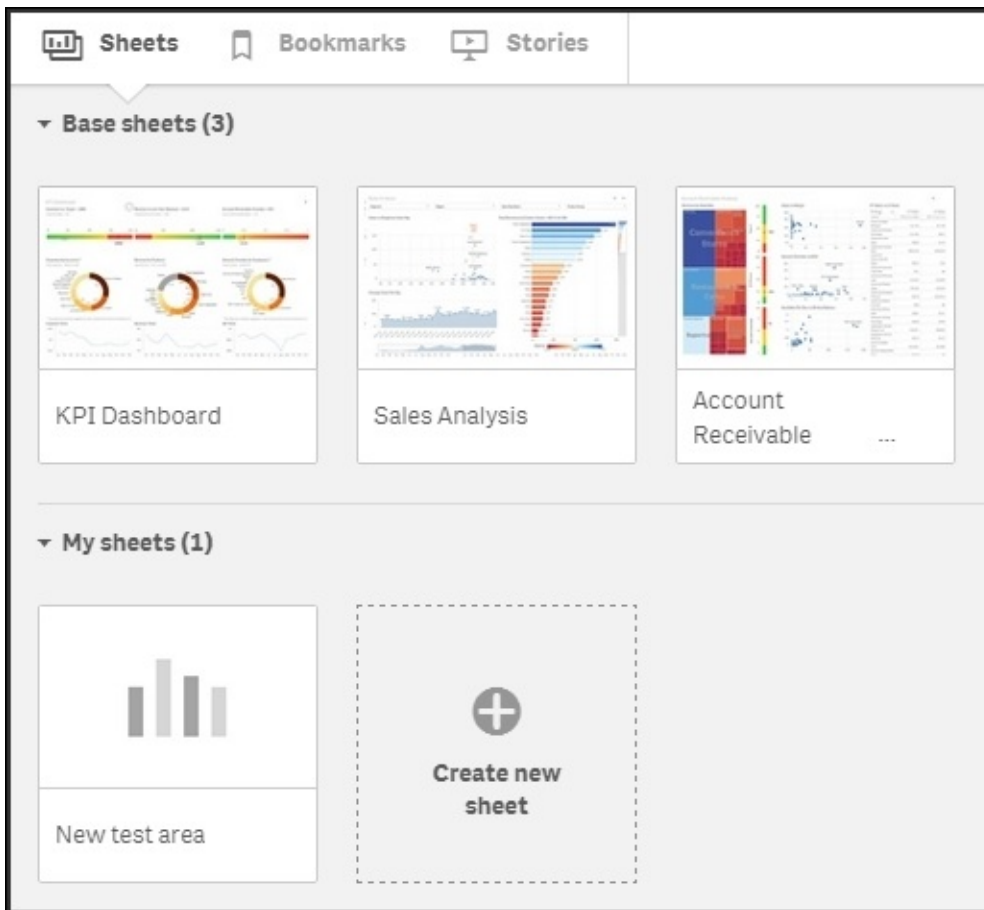


*Importing an app created in Qlik Sense Desktop in the Apps screen*

Once imported, you can set the owner of the app. Then, it will appear in the **My work** area in the hub of the owner. However, the app is still not published, which means other users cannot see it.

When you publish it, you move the app from **My work** to another stream, and once it is published, its layout is fixed and cannot be changed.





*Once an app is published, the app overview in the hub changes*

This is obvious if you look at the app overview in the hub. Here, you now have two rows of sheets: one with sheets that are fixed and public, and another with private sheets that aren't visible to other people.

You can also see this difference on the sheet listing the app objects. This sheet lists all sheets and stories, and QMC clearly indicates whether an object is public and who the owner is.

The user who creates an app is automatically designated as the owner of the app and its app objects. The app objects are published when the app they belong to is published. However, the users can add private app objects to the apps and share them by publishing the app objects from Qlik Sense. When this is done, the app overview in the hub gets three rows of sheets, as shown in the following screenshot:

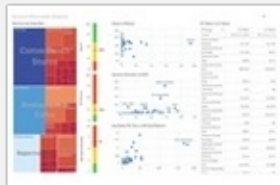
▼ Base sheets (3)



KPI Dashboard



Sales Analysis



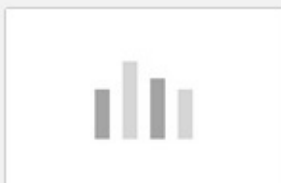
Account  
Receivable ...

▼ Published by me (1)



My public analysis

▼ My sheets (1)



New test area

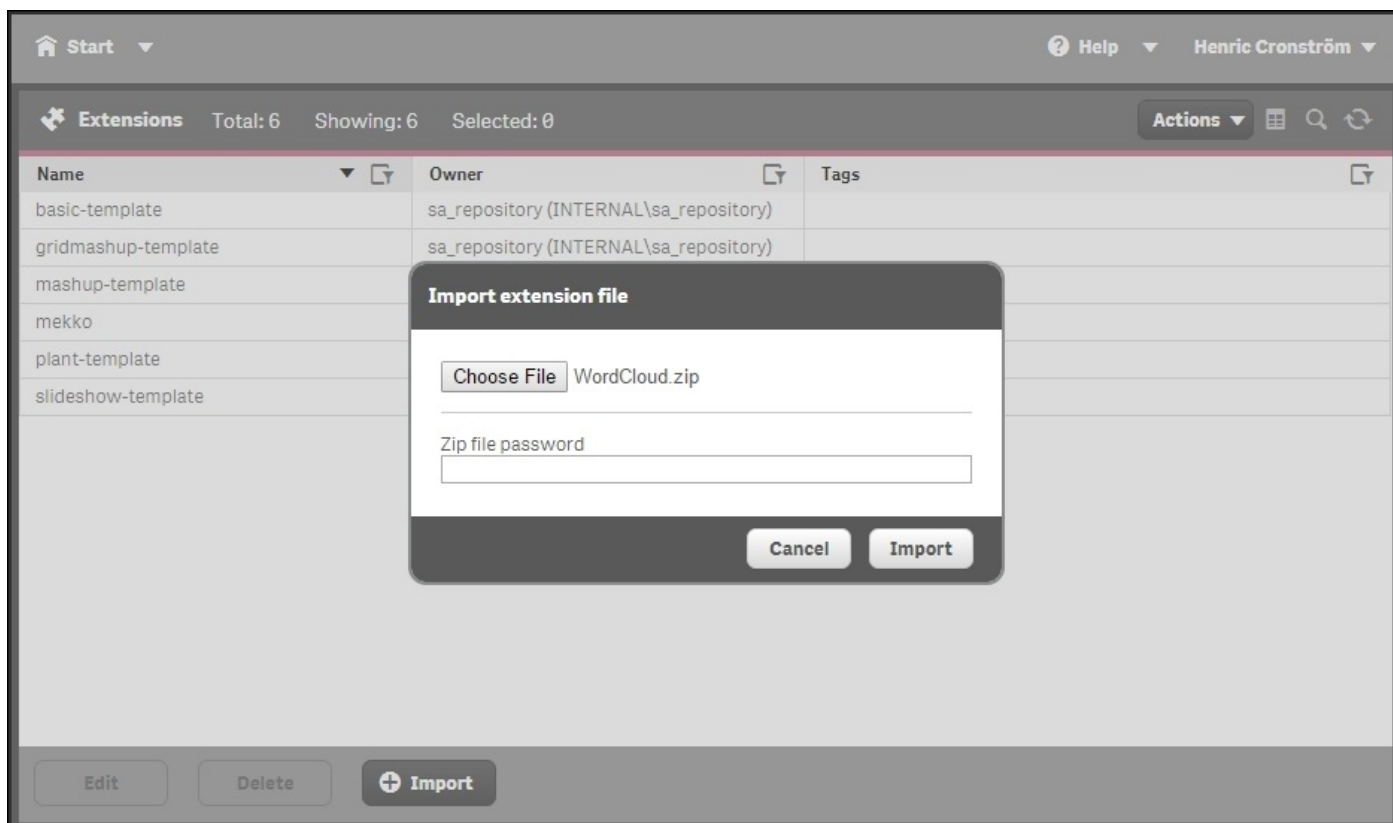


Create new  
sheet

# Importing extensions

As you saw in the previous chapter, it is very simple to create additional visualizations, extensions, in Qlik Sense. To use them, you need to import them to your Qlik Sense server.

All you need to do is to navigate to the **Extensions** sheet. Look for the **Import** button at the bottom of the screen. By clicking on this button, you can browse to the location of the ZIP file containing the extension and import it.



# Users and user directories

As soon as you want to manage the Qlik Sense server in terms of ownership and access rights, you need to have your users defined. Normally, these are already defined in a user directory, for example, in Windows Active Directory. Hence, you want to reuse these definitions.

On the **User directory connectors** page, you can define several sources for your users and user groups. You need to do this and sync at least one of them before you can start distributing licenses and access rights to your user groups.

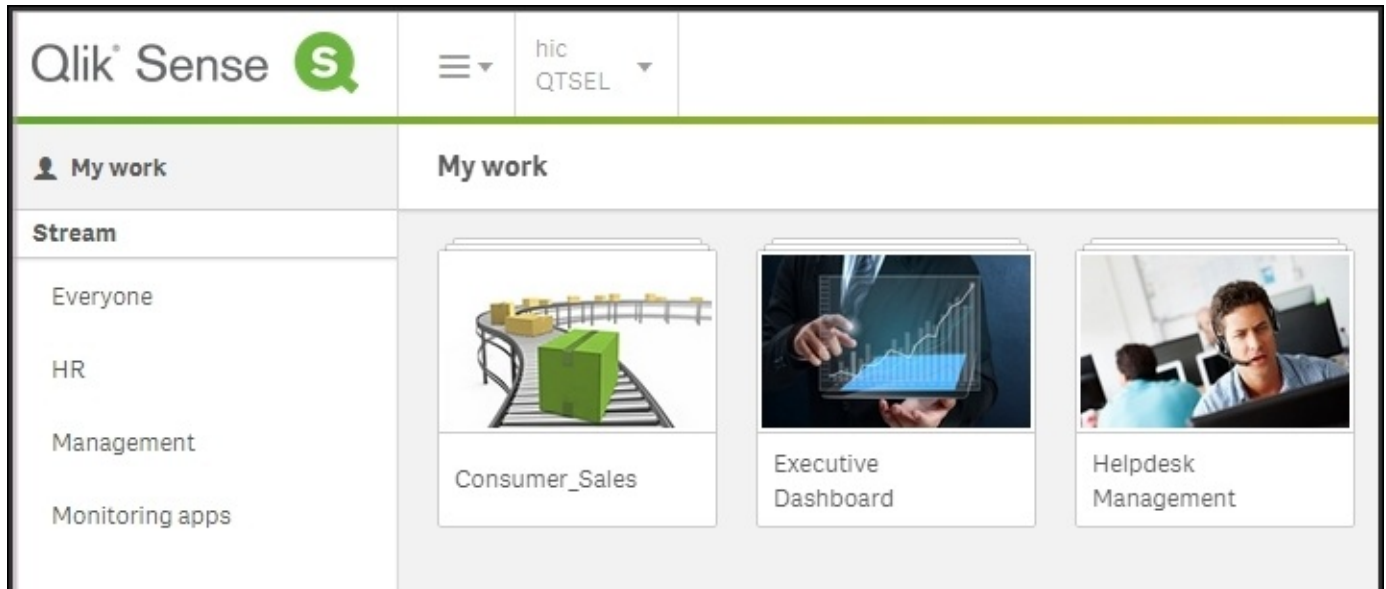
The users are managed on the **Users** sheet. However, when you first start Qlik Sense, the list of users is fairly short: just you and a couple of system users. To populate the list of users, you have two options:

- Define a user directory connector and sync the users in it
- Define rules for the access passes so that the users can assign licenses to themselves without you having to do it

# Defining streams

Once you have created or imported an app, you may want to publish it. Publishing an app means that you move it from your personal workspace to a stream of your choice.

You have already seen the streams in the hub, where they appear to the left as groups of apps. **My work** is your personal stream that no one else can see.



*Streams, as seen from the hub*

An app can be published to only one stream. By default, Qlik Sense includes a stream called **Everyone**, and you can create any number of additional streams from the **Streams** sheet. You should most likely create one stream for each distinct user group. Use the **Create new** button in the upper-right corner of the screen.

When creating your stream, you have the option to add a security rule, making the stream available only to some users. This is a very important security feature. One obvious example is, if you have a set of apps that should be seen only by the human resources department. Then, you should create a stream for this group and use the user information from the directory service to give access to this stream.

Another common case is if you want to delegate the administration of a stream to a group of users. The following screenshot shows a security rule that grants access to the **Human Resources** stream and to all users belonging to the **HR** user directory:

**BASIC**

Actions

- Create
- Read
- Update
- Delete
- Publish
- Change owner

user	userDirectory	=	+
value	HR		

**Properties**

Identification

Advanced

✓ Basic

Tags

**Associated items**

Preview

# Connectivity management

**Connectivity** means the connection to source data. Source data can be ERP systems of different kinds, file folders, web addresses containing tables, and so on. When running a Qlik Sense script, data is pulled from the different sources into the Qlik Sense app so that it can be analyzed at a later stage.

With Qlik Sense, it is easy to get an overview of all data connections used, something that used to be a challenge. By opening the **Data Connections** sheet, you get a list of the data connections used in different apps.

The data connections can be managed and security can be set separately for different connections. It is, for example, possible to prevent some users from using a specific data connector. This way, you can control the usage and ensure that data is used in the correct way.

# Tasks

On the **Tasks** page, you define the jobs that need to run in the background. These are of two kinds: **reload tasks** and **user synchronizations**.

The reload tasks are necessary to refresh data in the apps, which means you need to set up tasks so that they are refreshed with the frequency you want. Most apps should be refreshed once a day, but some others only need to be refreshed once a month. There are both advantages and disadvantages with a frequent refresh of the data. If it is refreshed rarely, for example, once per month, the users will not have the latest data.

On the other hand, if you refresh data too often, such as once per hour, you will have a heavy load on your server handling the reload tasks. You will also create a situation where two users in a meeting may have different opinions about what the correct number for a specific KPI is, since they looked at two different versions of the app. One looked at the app an hour ago, and the other just 10 minutes ago. This does not create an understanding; rather, it creates confusion, since you have two versions of the truth.

You should ask yourself whether the users benefit most from having as fresh data as possible, or whether they benefit more from having one truth. A good balance is to have one refresh per day. The users will learn this, and refer to the numbers as today's numbers and yesterday's numbers.

User synchronization is necessary to refresh data from the directory service, so that Qlik Sense is aware of any changes made to groups and users.

A task can be triggered by either a scheduler or the completion of another task. This way, you can get task chaining.



# System management

The group to the bottom left in QMC relates to system settings. Here, you can configure the nodes, engines, proxies, schedulers, repositories, sync rules, and certificates. With these, you can configure how the Qlik Sense server should work on different computers. You can do really advanced things here, but this is beyond the scope of this book.

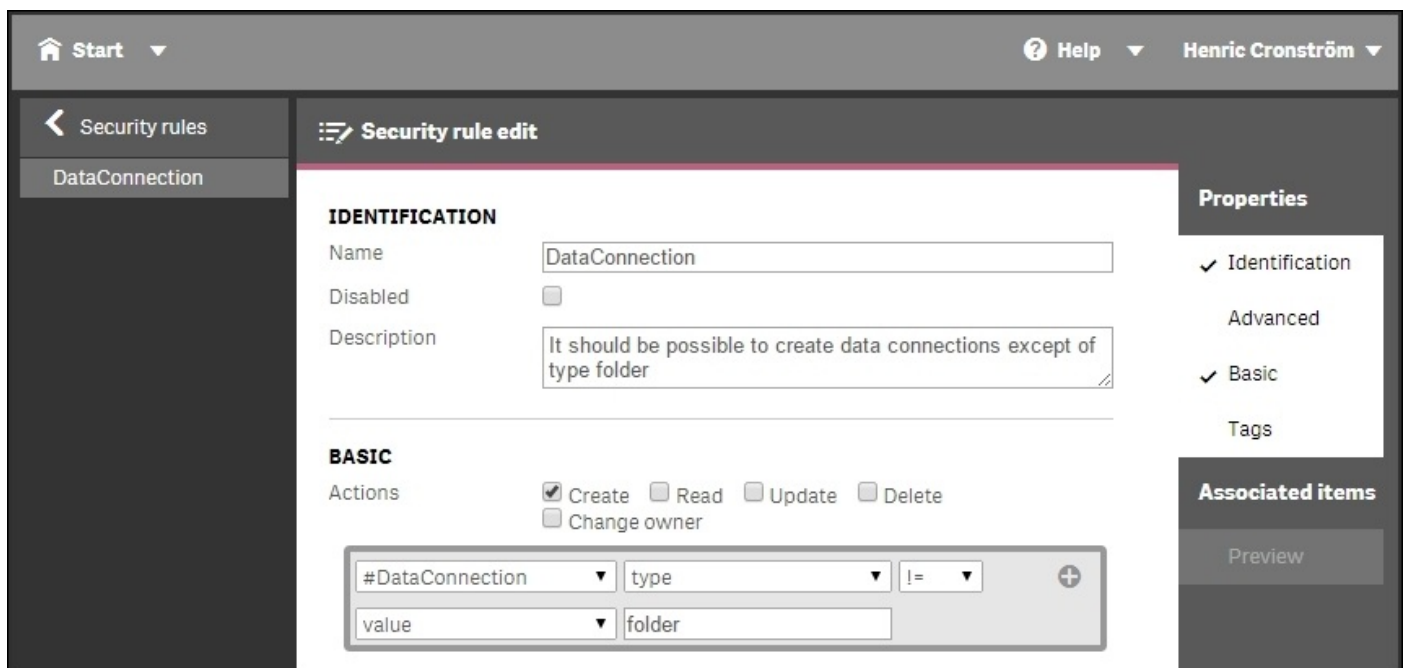
# Security rules

You can set access control for most of the preceding settings, for example, only some users should be able to see a specific application; only some users should be allowed to use a specific data connection; all users should be allowed to create data connectors to databases, but not to file folders; only some users are allowed to log in using a specific pool of login access passes; and so on.

When doing so, you should think of the following user types:

- **Developer:** These are users who are allowed to create apps, sheets, stories, objects, and who can use and create data connections
- **Contributor:** These are users who are allowed to create stories and sheets for published apps but are not allowed to create new apps
- **Consumer:** These are users who can only use apps, sheet, stories, objects, and so on but are not allowed to create content

These rules are called security rules, even though they do not always pertain to true security. They can be edited on each sheet, for example, the rules for streams can be edited on the **Streams** sheet, but there is also an overview: the security rules have a sheet of their own where they can be edited.

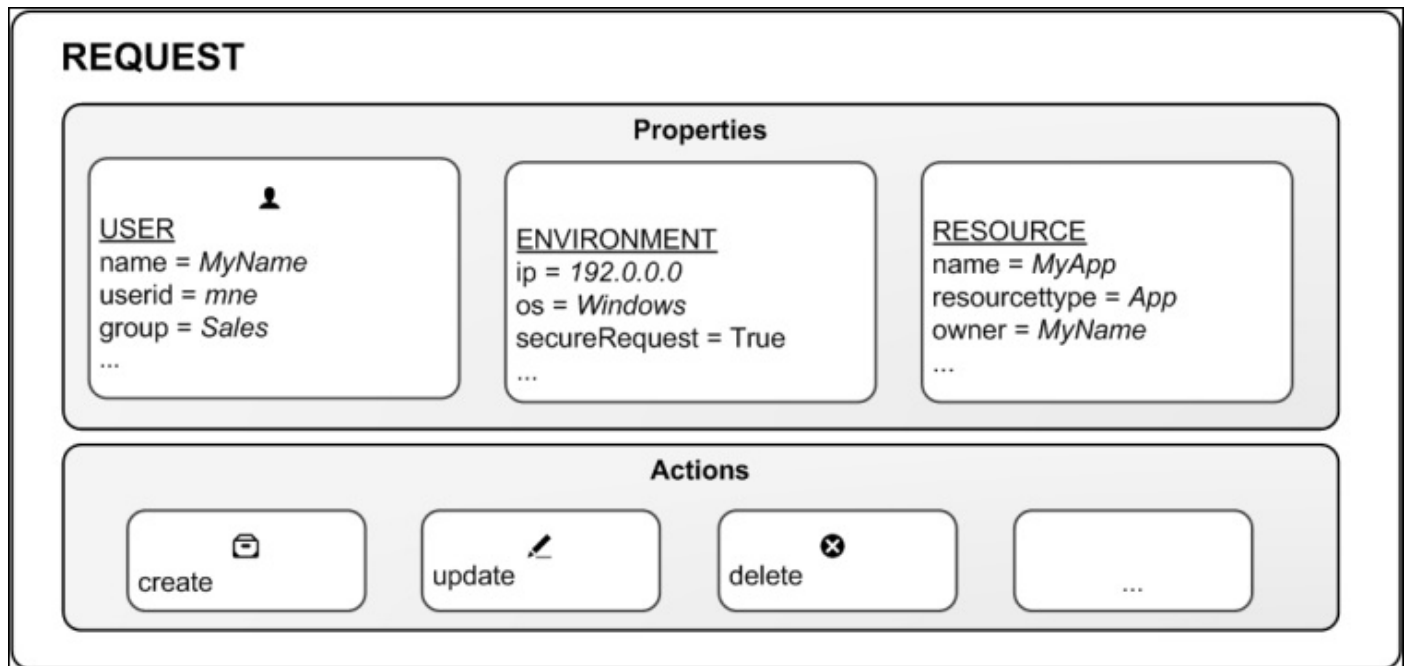


When you create a security rule using the basic interface, you create a property-value pair that grants users the right to do something. In the preceding screenshot, all users are granted the right to create data connections that aren't file folders.

The rules are property-based and the properties are used to describe the parties involved in an access request. In the usual case, the parties involved are the user making the request, the environment the request is made from, and the resource the request applies to.

Each property is defined in a property-value pair such as **group = Sales** or **resourcetype =**

**App.** Each request, in turn, includes the property-value pairs for the users, environments, and resources involved in the request together with the action that the person making the request wishes to perform on the resource, for example, create, update, or delete.



*The four components in security rules: user, environment, resource, and action*

You can create rules based on the property-value pairs. This means requests for an action on a resource are granted only if the property value of the requester matches the property-value conditions defined in a security rule for that resource.

A rule can read as a sentence in the following way: *Allow the requester to [action] the [resource] provided that [conditions].*

Each rule must describe the action and the resource or resources the action should be applied to. If you don't define any rules for a resource, no users will have access to that resource.

By design, security rules are written to include, not exclude, users. Users who are not included in security rules will be denied access. So, security rules must be created to enable users to interact with Qlik Sense content, data connections, and other resources.

Hence, the rules define when access is granted, and there is no rule that can deny a user access. If there is a rule that allows the user to do something, they are allowed to do so. So, if you want to deny a user something, you must delete the rule that grants access, or edit the rule.

# Monitoring

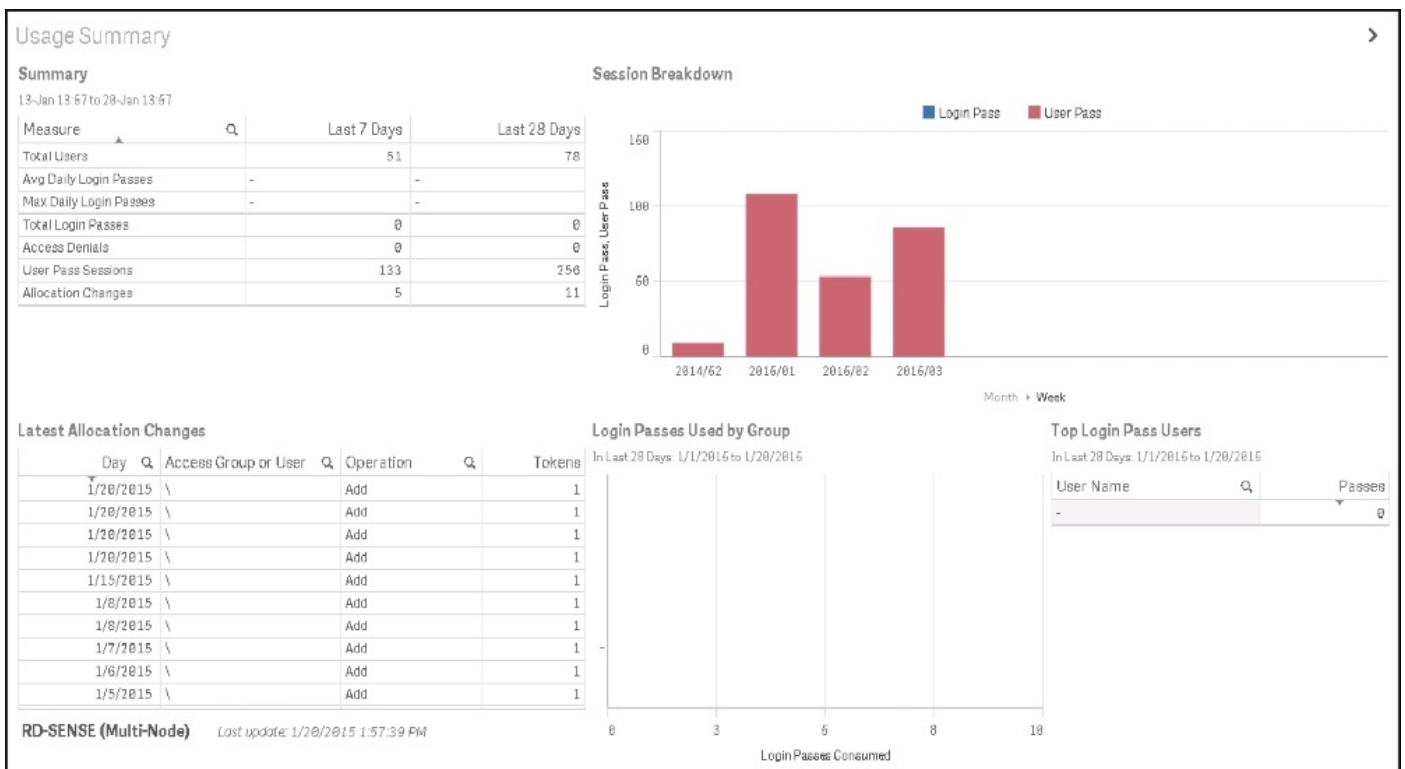
Delivered together with Qlik Sense, you will also find two monitoring Qlik Sense applications: the **License Monitor** and **Operations Monitor**. These read the log files of Qlik Sense and give you a good overview of the state of the Qlik Sense server.

The following screenshot shows the Operations Monitor:



The Operations Monitor

The following screenshot shows the License Monitor:



The License Monitor





# Security

The security in Qlik Sense consists of many parts. In the QMC, there is a system with security rules for almost everything you can do, not only data access. There are also rules to change the setup or rights to publish apps or sheets. This implies protection of the platform, that is, how the Qlik Sense platform itself is protected and how it communicates and operates.

However, security as a concept goes beyond that. So let's start from the beginning.

# Authentication and authorization

The two most basic concepts in security are authentication and authorization.

Authentication answers the question, “Who is the user and how can the user prove it?”

Authorization answers the question, “What does this specific user have access to, and what are they allowed to do?”

In Qlik Sense, authentication and authorization are two distinct, unconnected actions. In addition, the sources of information used for authentication do not have to be the same as for authorization, and vice versa.

Qlik Sense uses standard authentication protocols to authenticate every user requesting access, for example, Windows Integrated Authentication, HTTP headers, and ticketing. If you want a customized authentication, you can configure this in the proxy, but the details of this are beyond the scope of this book.

Authorization on the other hand, is the procedure of granting or denying user access to resources. A user perhaps has the right to see a resource or perhaps they don't. When it comes to data, the right to see data can be set with different granularity. A user may see an app or they may not; and once opened, the user may be restricted to see some parts of the app but not other parts.

Hence, authorization can be defined on several levels:

- Firstly, there is the administrator access control. Which rights are needed for the different roles and responsibilities of the administrators? This is controlled in the security rules as previously described.
- Secondly, there is app-level authorization. Is the user allowed to access the app? Which functions in the app is the user allowed to use, for example, printing, exporting, and creating snapshots?
- Thirdly, there is data-level authorization. Is the user allowed to see all of the data in the app or just parts of it?



# Content security

Content security is a critical aspect of setting up and managing your Qlik Sense system. QMC enables you to centrally create and manage security rules for all your Qlik Sense resources. Security rules define what a user is allowed to do with a resource, for example, read, update, create, or delete.

Additionally, there is data reduction by a section access in the script that handles data-level authorization. The section access is an app-defined, data-driven security model, intimately connected with the data model. It allows the implementation of row- and field-level data security.

In data-level authorization, the authentication information also exists in the data model (albeit in a hidden part of it). It could be, for example, a username.

The selection propagates to all the other tables in the standard QlikView manner so that the appropriate records in all tables are excluded, wherein Qlik Sense reduces the scope for this user to only the possible records. This way, the user will only see data pertaining to the countries to which they are associated.





# Summary

In this chapter, we have seen that with the QMC, you can manage a Qlik Sense installation very efficiently. It includes a wide range of functions that allow you to configure your system the way you want it, and it allows you to set access rights on not only apps but also on streams, licenses, and data connectors.

Since QMC is based on standard web technology, you can, in principle, use any browser to run it, and it integrates well with other systems used to manage software and hardware components. In addition, you can use APIs to create custom management utilities.

To end this book, we'll be looking at putting Qlik Sense into practice for active data discovery as we spend the last chapters analyzing the examples of sales, HR, travel expenses, and demographics.



# Chapter 10. Sales Discovery

Throughout this book, we have shared the driving forces in the creation of Qlik Sense and key capabilities to aid in helping organizations make better business decisions. This chapter is the first of four that will apply Qlik Sense to the challenges of analyzing sales performance within your organization. This example and many others are available for you to explore live at <http://sense-demo.qlik.com>. Please bookmark this link as additional demonstrations and examples are constantly being added and updated. Now, let's turn our attention to the following challenge of sales analysis and how Qlik Sense addresses this common business challenge.

In this chapter, we will cover the following topics:

- Common sales analysis problems
- The unique way Qlik Sense addresses these problems
- How the Sales Discovery application was built

# The business problem

Analyzing sales information can be a difficult process for any organization, and is critical to meeting sales expectations and understanding customer demand signals. What makes sales analysis so difficult is that many perspectives can be taken on the enormous amount of information that is captured during the sales process.

Some key questions include:

- Who are our top customers?
- Who are our most productive sales representatives?
- How are our high margin products selling and to whom?

The key thing here is that during the analysis process, one answered question always leads to further questions depending on the results; in other words, the analysis process's diagnostics. These paths to discovery cannot be precalculated or anticipated. With this in mind, let's take a look at how the Sales Discovery application seeks to meet these requirements.



# Application features

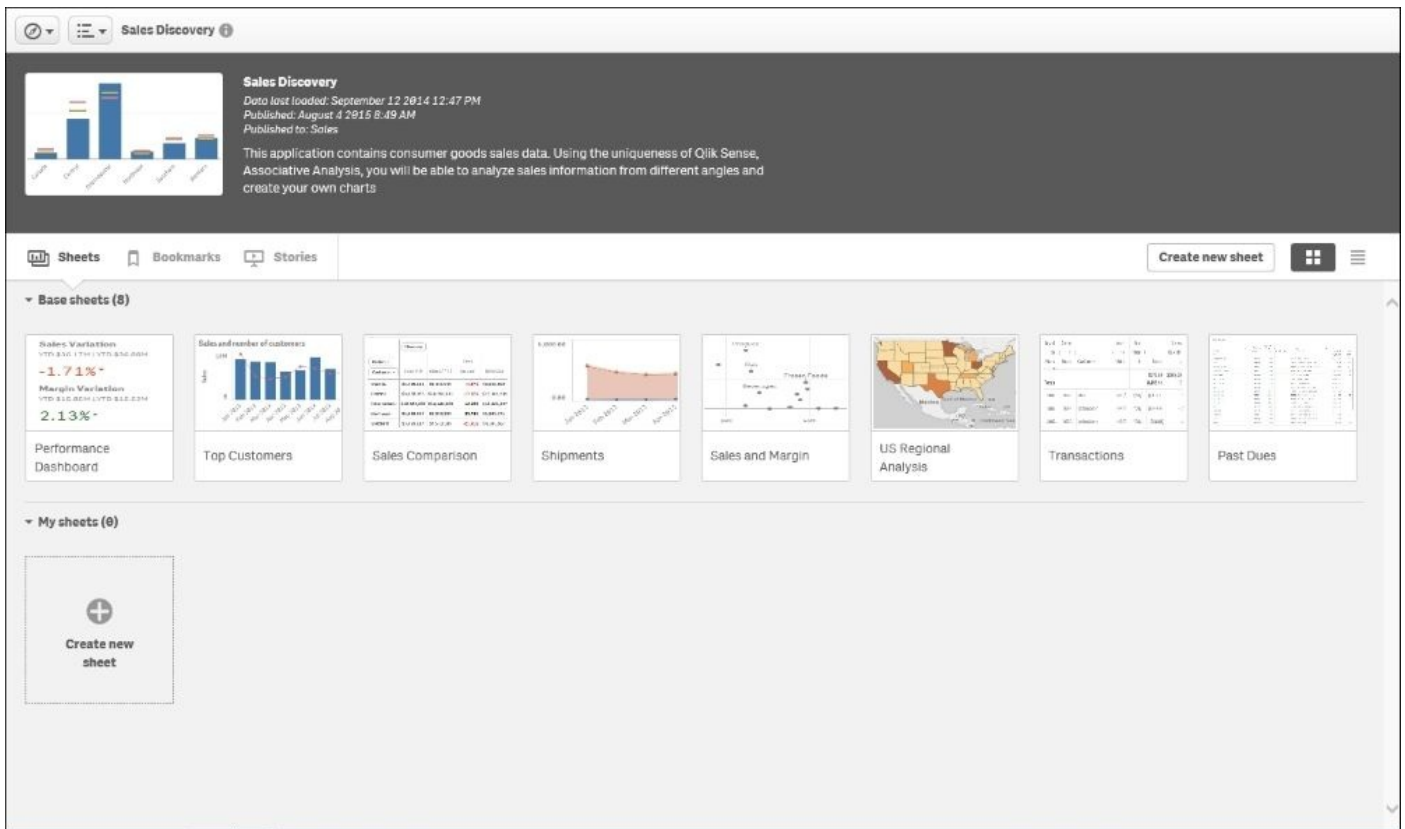
Qlik Sense's associative model allows users to answer the common questions outlined in the preceding section through the selection of elements in filter boxes, but more importantly, to drive follow-up questions. Often, this relies on Qlik Sense's ability to instantly identify the associated and nonassociated data, which is also known as The Power of Gray after the color assigned to nonassociated elements highlighted in [Chapter 3, Empowering Next Generation Data Discovery Consumers](#). The following are two key beginner questions that will drive additional questions as the analysis begins.

Key questions include:

- Who are our top customers?
  - What are these customers buying?
  - Where are these customers buying from?
  - Are the products getting there?
  - Who are our bottom five customers?
- Can we cross-promote products?
  - Who are our most productive sales representatives?
  - What products are the most productive sales representatives selling?
  - Whom are they selling to?
  - Which regions are they being sold in?

Before we begin, let's review the main sheets in the Sales Discovery application. As noted in the following screenshot, the application is made up of the **Performance Dashboard**, **Top Customers**, **Shipments**, **Sales and Margin**, **US Regional Analysis**, **Transactions**, and finally, **Past Dues** sheets:



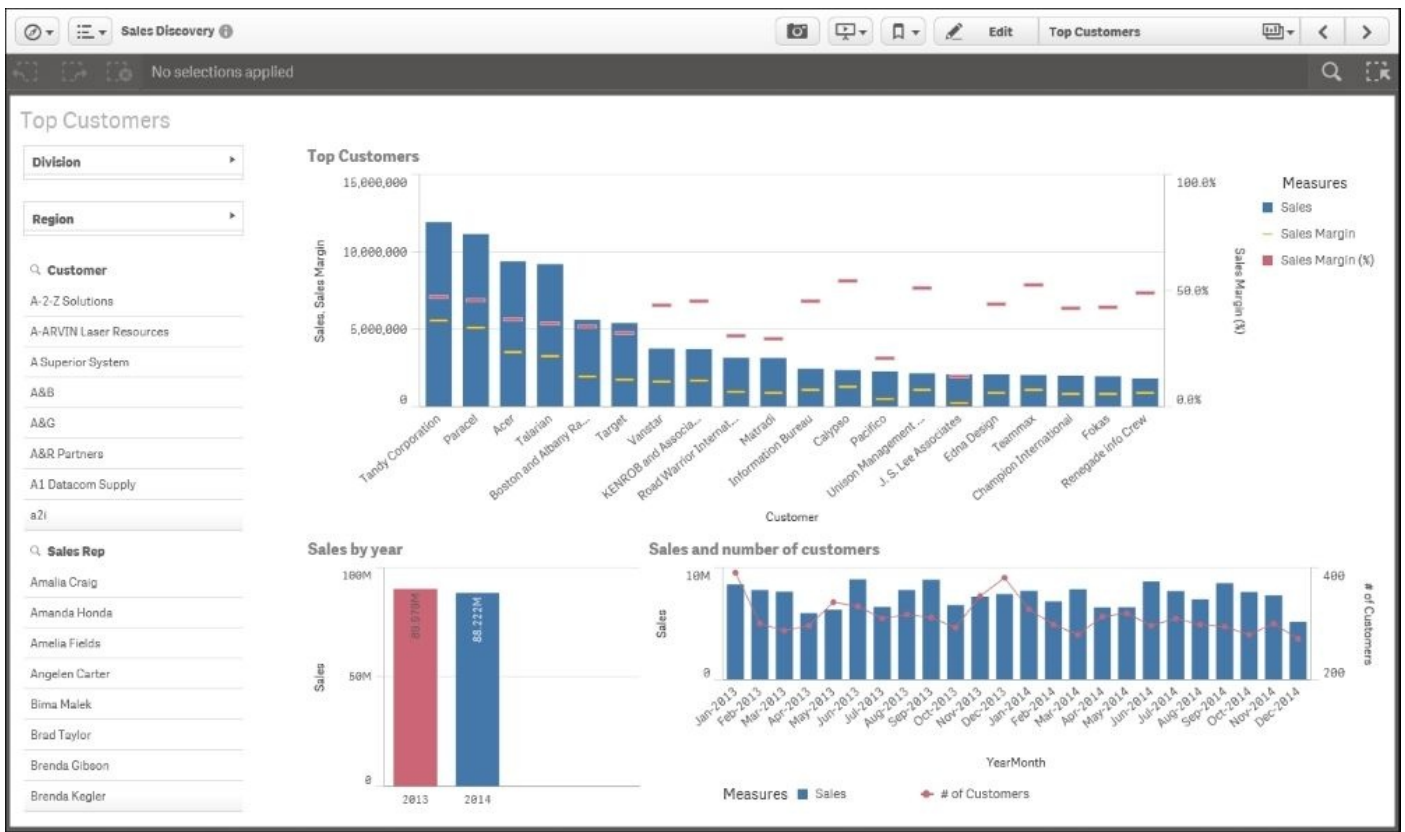


*The application overview*

Given the nature of the associative model, all filters are global, allowing a user to explore each application sheet in the context of the selected filters and associative results. Filters serve as a way to ask questions to the Qlik Sense application.

# Who are our top customers?

So, with that said, let's begin with our first question, "Who are our top customers?". This is a typical question that can be handled by a number of BI solutions in the marketplace.

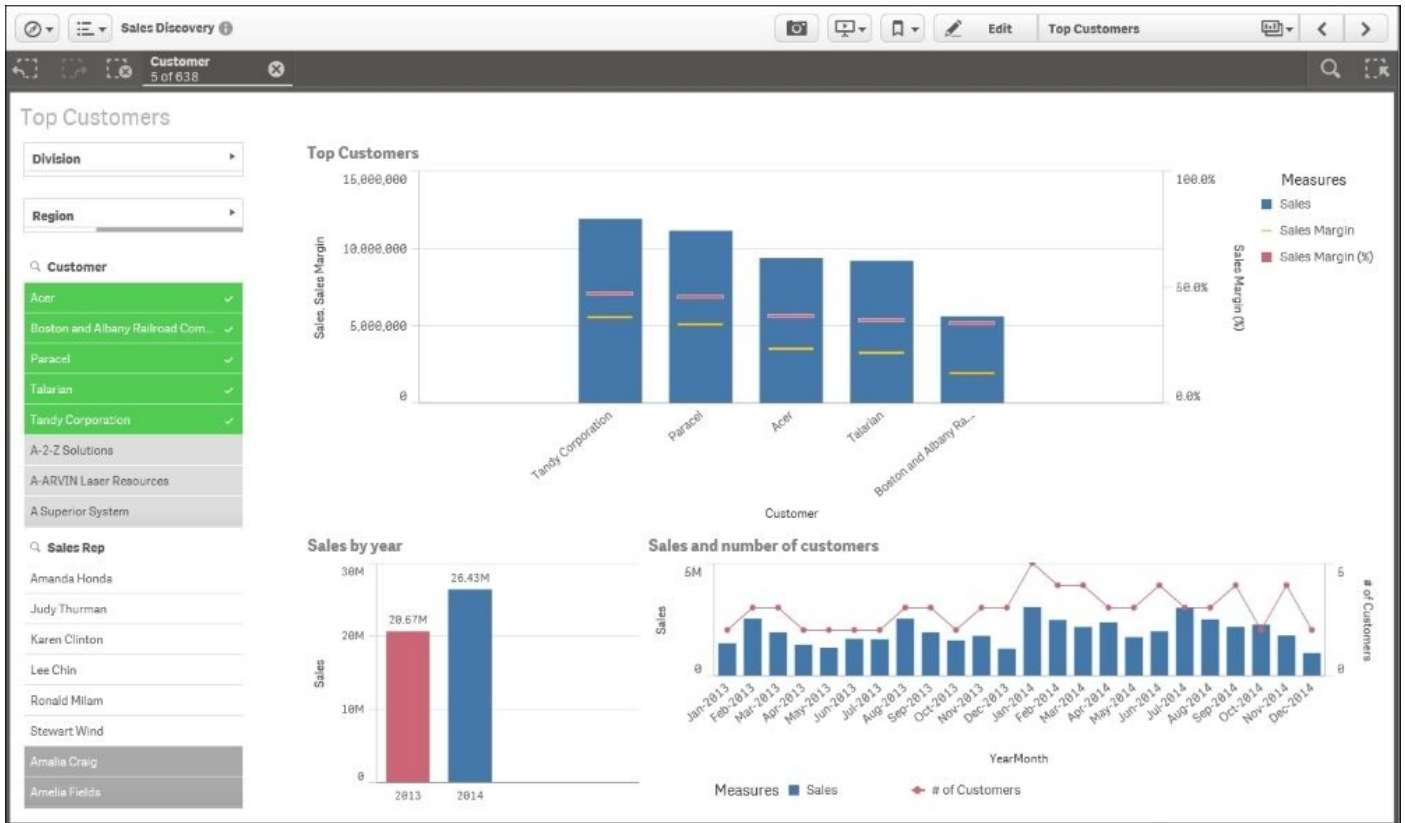


*Our top customers*

In the preceding screenshot, we can see that the top five customers in terms of sales are **Tandy Corporation, Paracel, Acer, Talarian, and Boston and Albany Railroad Company.**

# The 360-degree customer view

Now is where things get interesting in Qlik Sense and the associative experience. Once we select these customers, as shown in the following screenshot, we get a 360-degree view of them across the application. Immediately, we can see which representatives have sold to these accounts, the trended revenue, year-on-year sales, as well as in what percentage of the regions these sales were made. The percentage of the regions (noted by the green arrow) where the sales were made is highlighted in the filter list shade, which shows approximately 25 percent of the regions:

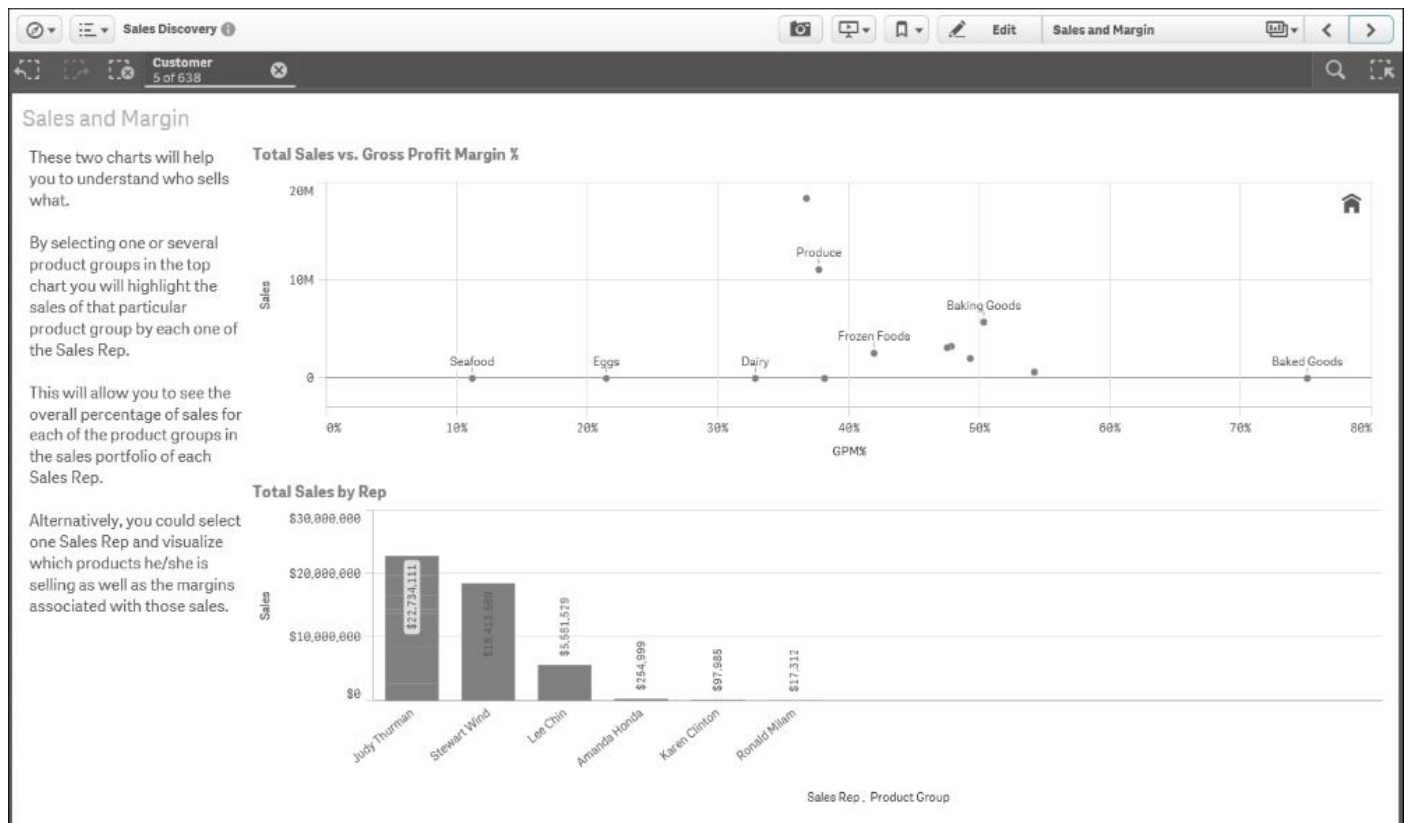


*Top five customer sales*

# Filtering customers

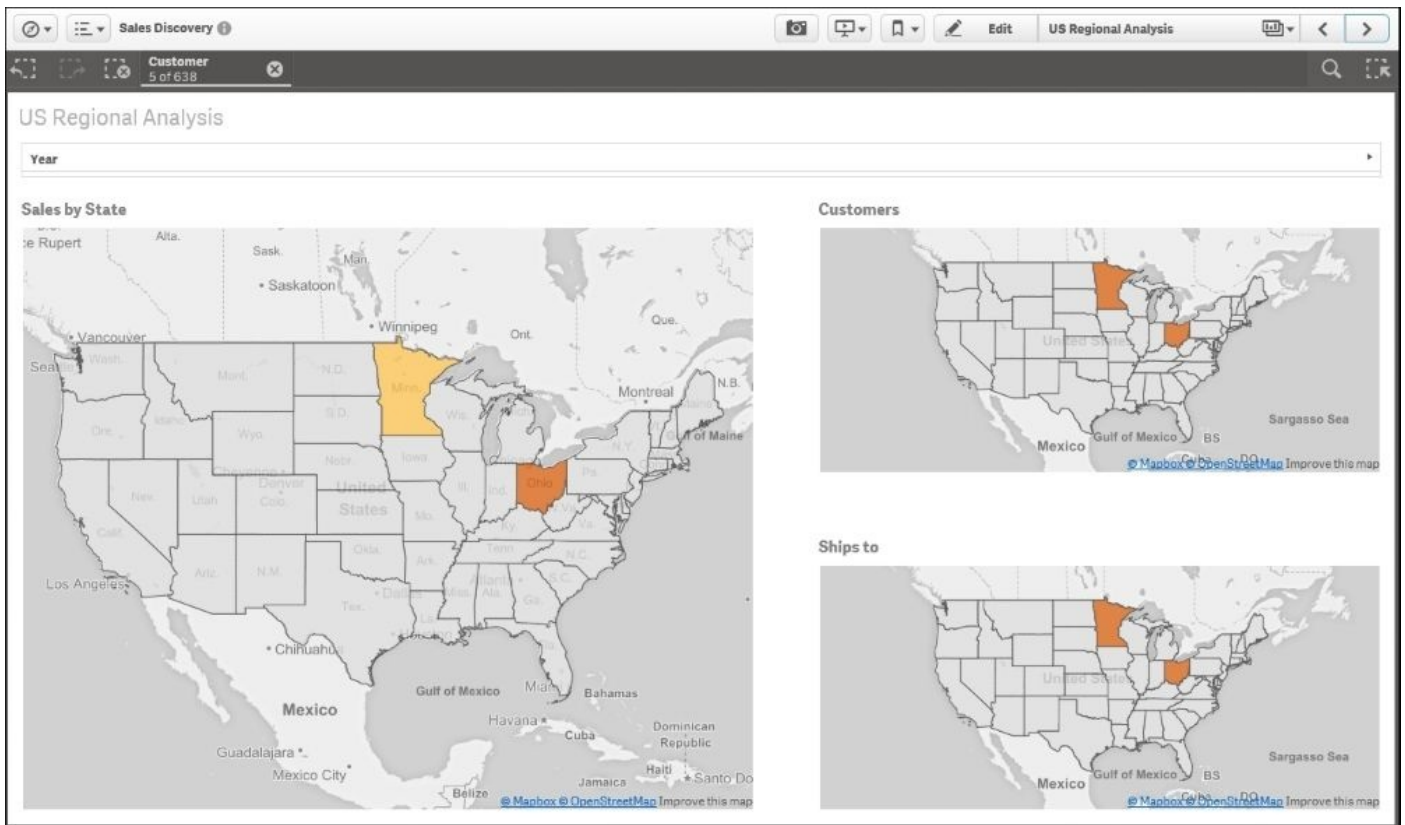
The preceding information leads naturally to the next question: what are these customers buying and from where?

Again, because of Qlik's associative indexing engine, this information is linked together automatically. Based on this, let's view the impact that filtering these top five customers has on sales and gross margins, as shown in the following screenshot. Note that the customer filter box with selections is globally available at the top of the screen. In the **Sales and Margin** sheet, we can see that **Canned Foods** and **Produce** account for the largest sales, and **Baking Goods** has the highest gross margin with just over 50 percent.



*What are these customers buying?*

As we continue our analysis, the next question that is most likely to arise is where are these sales occurring? Again, this data is available in the sales transaction, and Qlik's associative indexing engine makes this easily available and interactive within the application. Note that in the next figure, the **US Regional Analysis** sheet displays the sales by states, customers, and the important shipments as well:

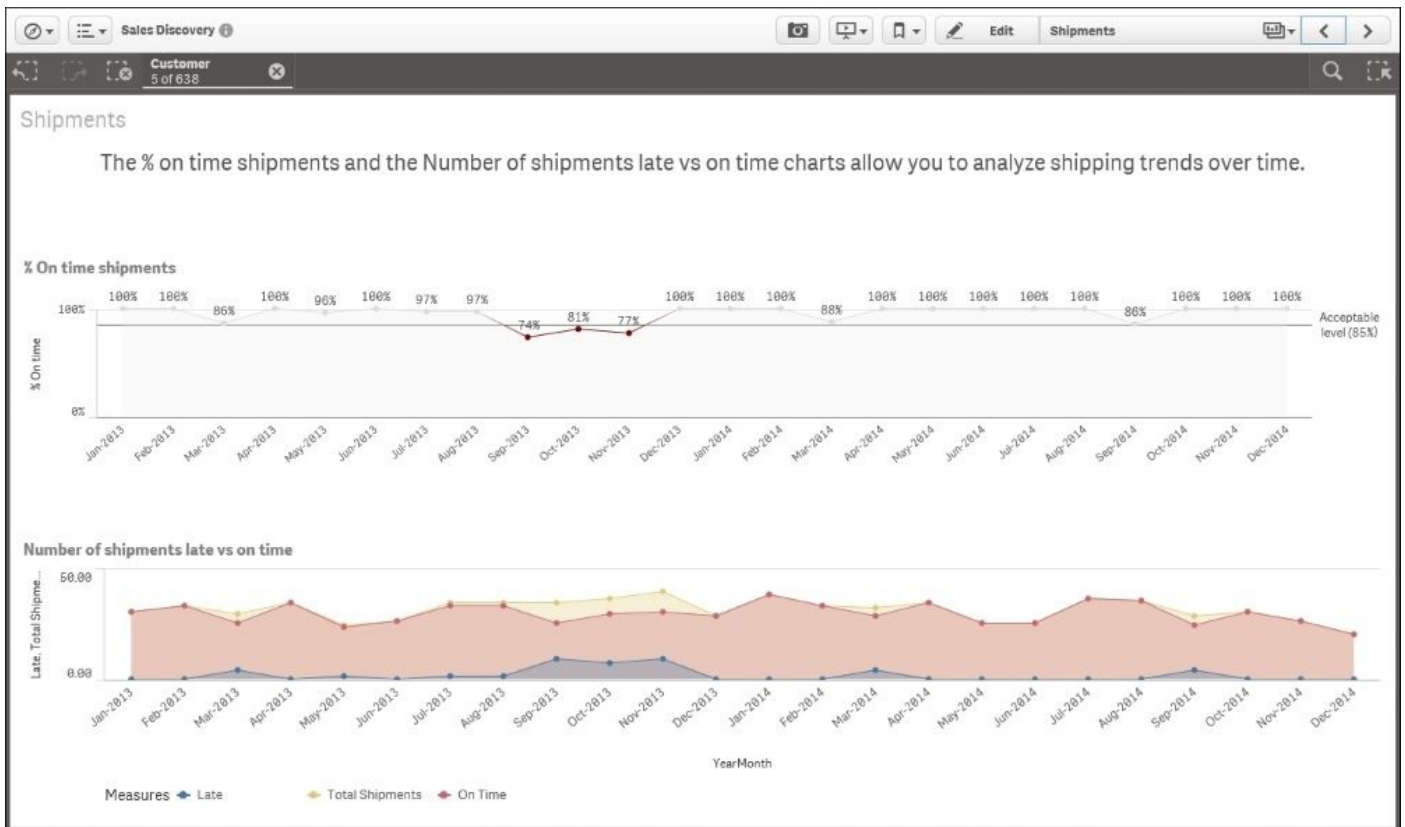


*Where are these customers buying from?*

## Reviewing shipments for top customers

We can see in the preceding screenshot that Minnesota and Ohio account for all top five customer sales that are between 5.94 million and 11.89 million. After reviewing this sheet, a number of questions can arise and be analyzed. Let's follow one specific thought on shipments. Are products getting there?

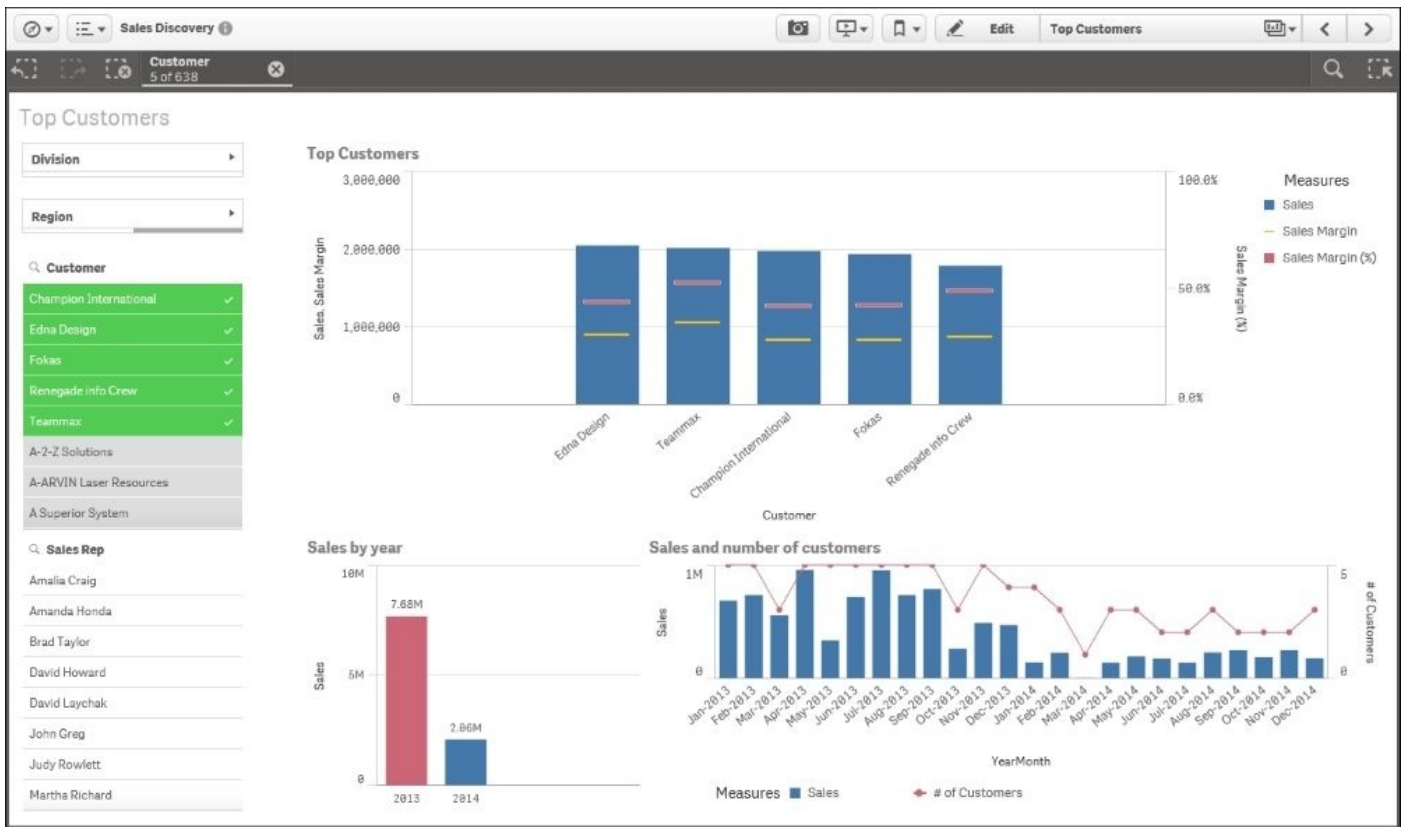
As we know, shipments play a critical role in a sales process because without shipping you cannot book revenue and continue to grow sales. With this in mind, let's turn our attention to the **Shipments** sheet as shown in the following screenshot. From here, we can see the trending shipment information on two levels: **% On time shipments** and **Number of shipments late vs on time**. Additionally, we see that the on-time shipment goal is **86%**. Based on this, we can see problems in meeting these goals in September, October, and November 2014:



*Are the products getting there?*

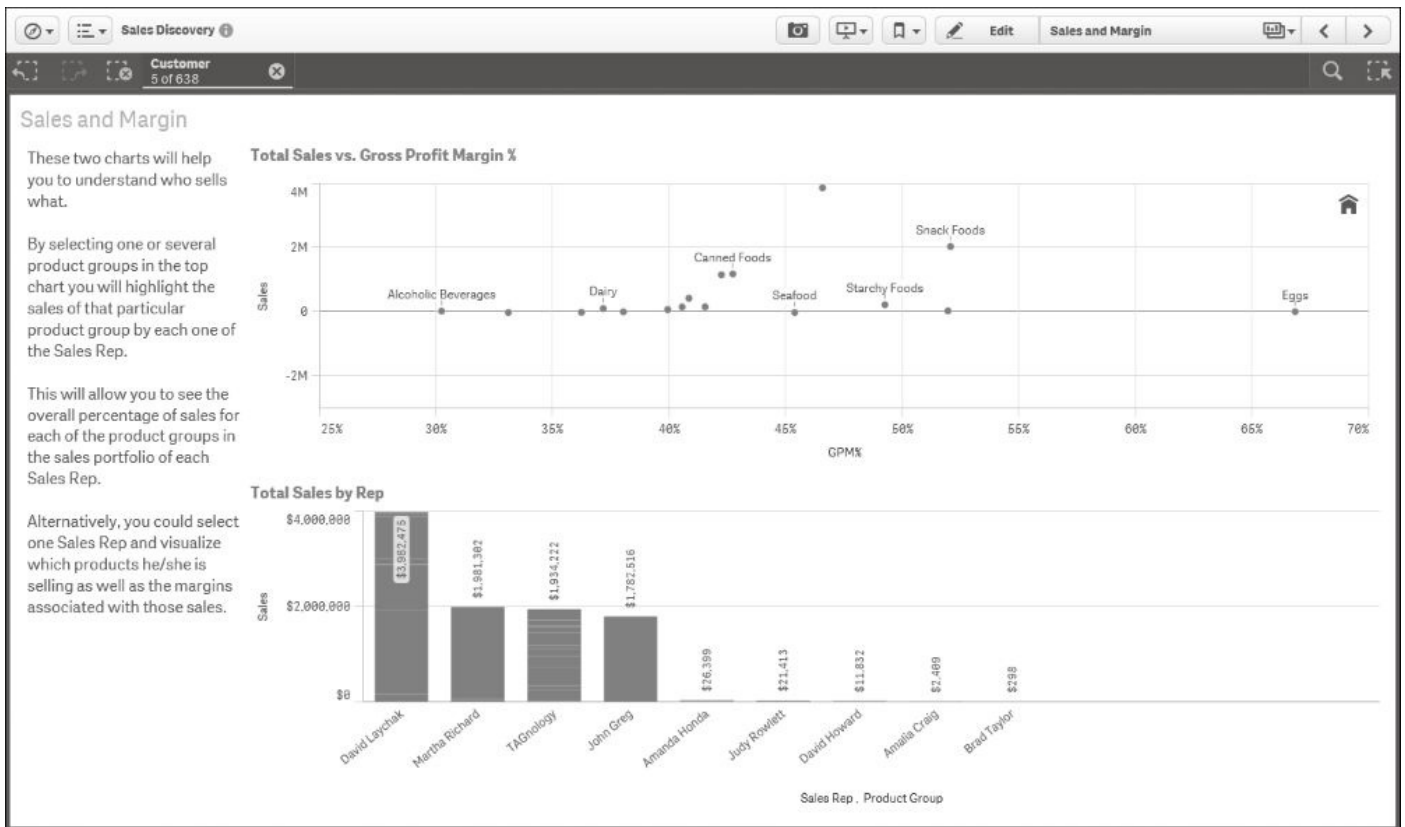
## Reviewing the bottom five customers

Now, let's turn our attention to an equally important topic: who are our bottom five customers and how can we increase sales to them? In the following screenshot, we can see the bottom five customers: **Edna Design**, **Teammax**, **Champion International**, **Fokas**, and **Renegade Info Crew**. Our sales to them are 2 million products or less and they purchase lower margin products:



*Who are our bottom five customers?*

While taking a note of this, let's dig in a bit deeper on the products they purchase. In the following screenshot, we can see that these customers purchase a large amount of **Produce** and **Snack Foods**. Now, the question arises—can we cross-promote products to increase our sales from these customers?



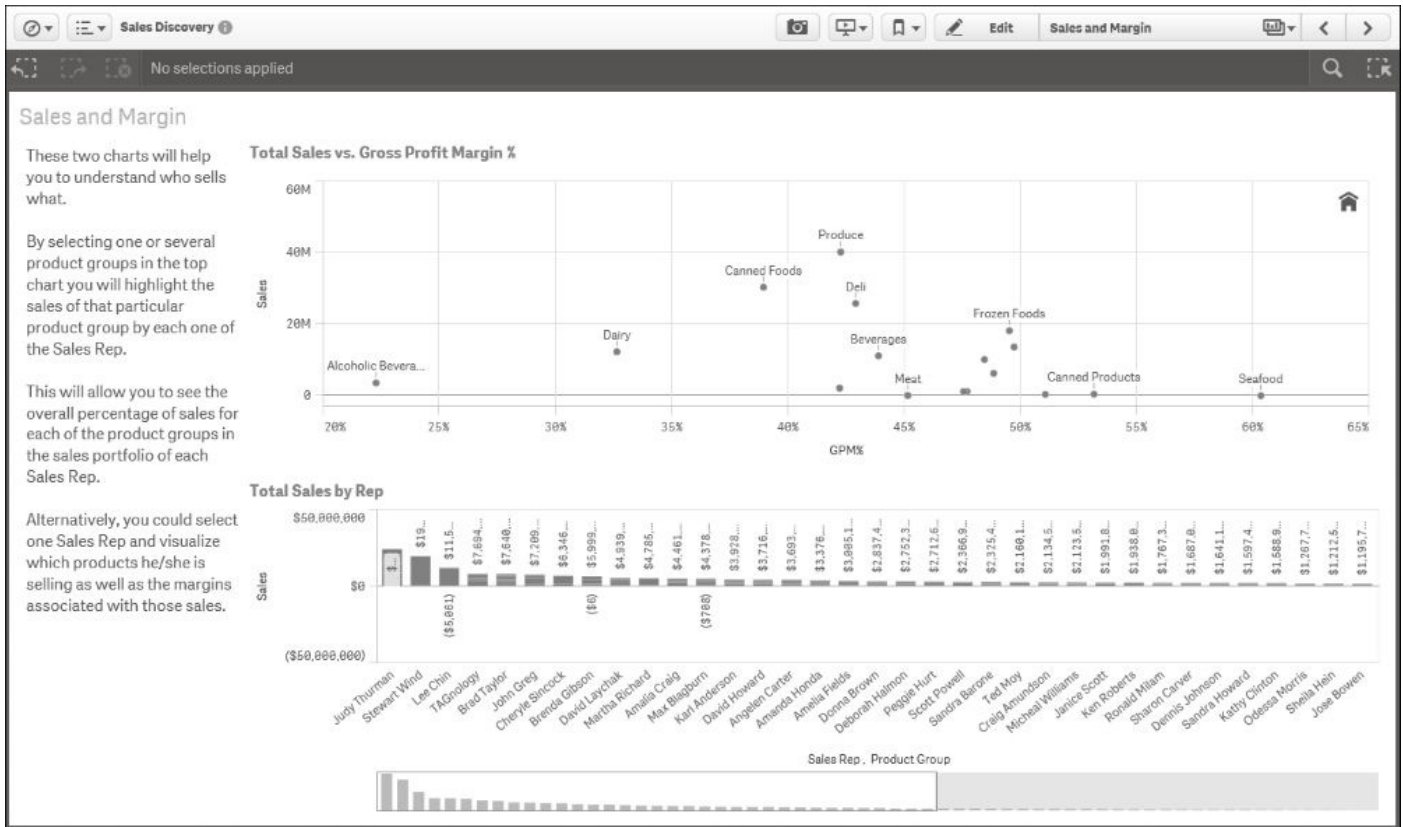
### *Can we cross-promote products?*

Based on the information gleaned, we can see some opportunities to cross-promote products. For example, with the high purchase of **Produce** by these customers, perhaps a cross-promotional program that introduces **Eggs** (at a **67%** margin) to them to supplement their produce may raise sales. Additionally, with strong sales of **Snack Foods**, perhaps we can expand the sales of **Baked Goods** (at a **52%** margin) to these customers as well. Now, let's turn our attention to the analysis of sales representatives.



# Who are our most productive sales representatives?

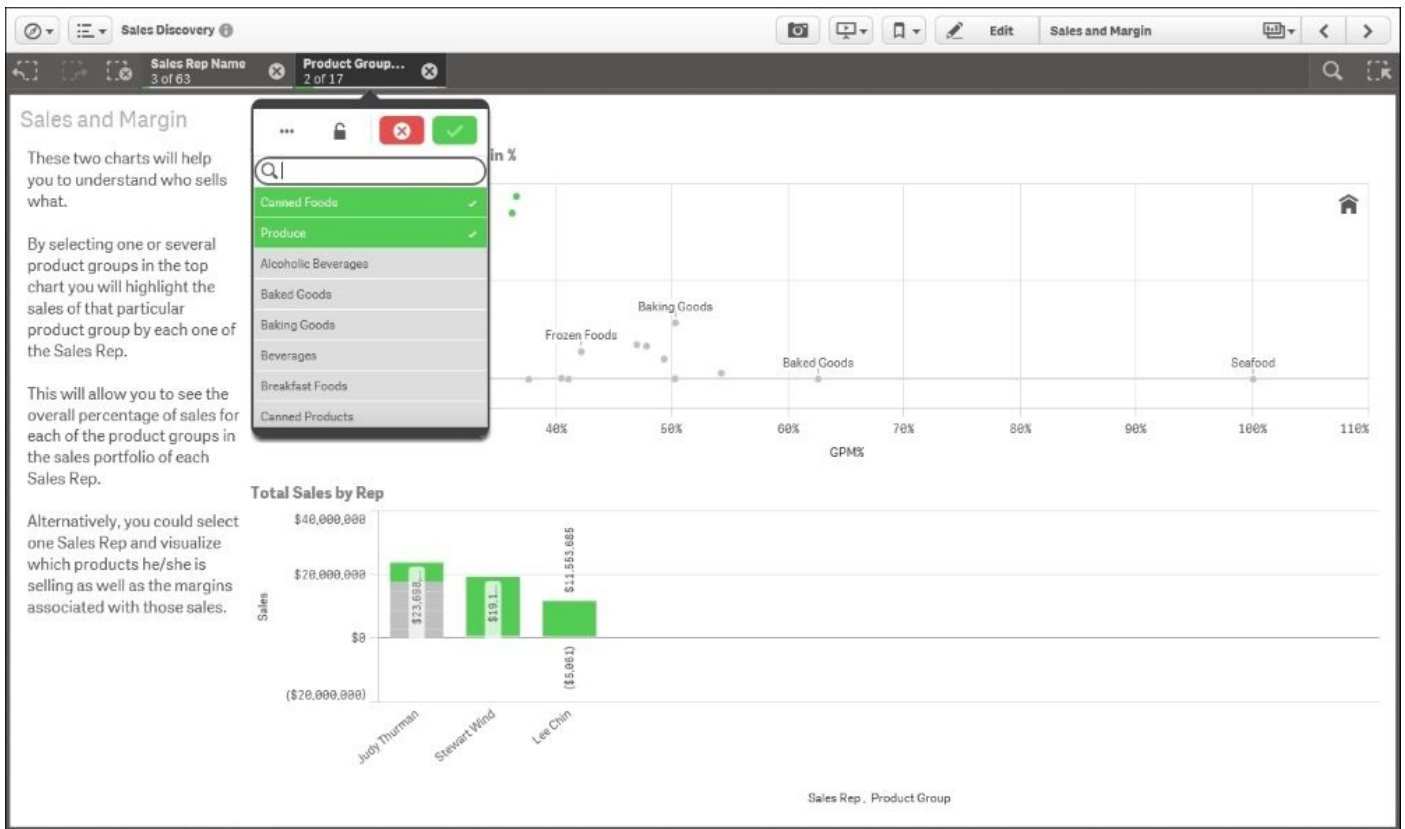
As often is the case, a key area for analysis is the performance of sales representatives. So, let's turn our attention to the **Sales and Margin** sheet in our Sales Discovery application, as shown in the following screenshot. Here, we can see that **Judy Thurman**, **Steward Wind**, and **Lee Chin** lead the sales team in terms of revenue:



*Who are our most productive sales reps?*

## Analyzing products

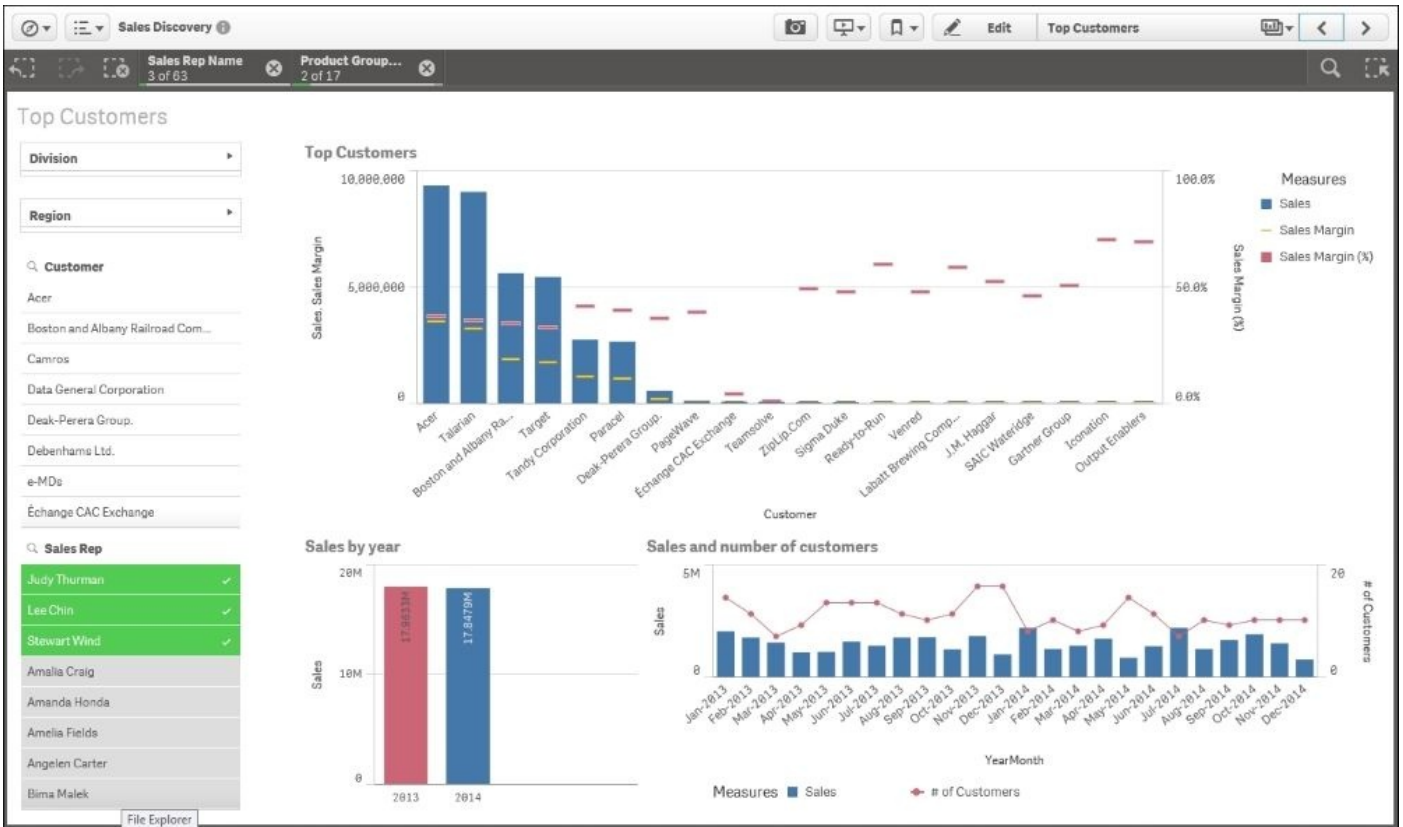
The next question that arises is what products are they selling? As we can see, **Canned Foods** and **Produce** are the top selling products. After identifying these sales representatives and top selling products, we will need to combine this information with an understanding of which customers are driving these sales.



*What products are they selling?*

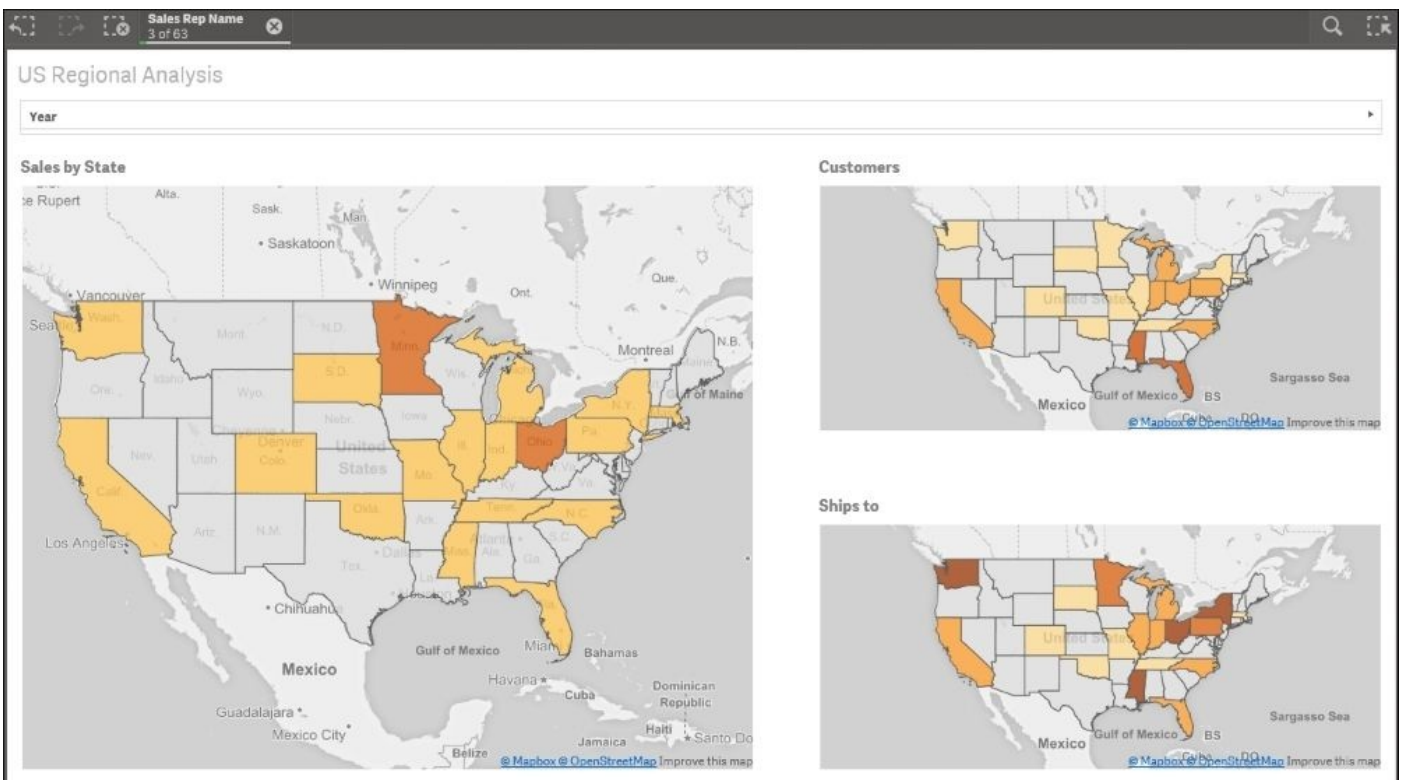
## Analyzing customer sales

Navigating back to the **Top Customers** sheet, we can see from which customers these sales are generated. Perhaps, while working with these sales representatives, additional promotions can be developed to expand the sales of products such as **Canned Products** to these customers.



*Who are they selling to?*

The final area to help improve sales representative performance is to analyze where these products are being sold. In the **US Regional Analysis** sheet, we can see that **Sales by State, Customers**, and **Ships to** are nicely dispersed, and additional information is not necessary for the next step:

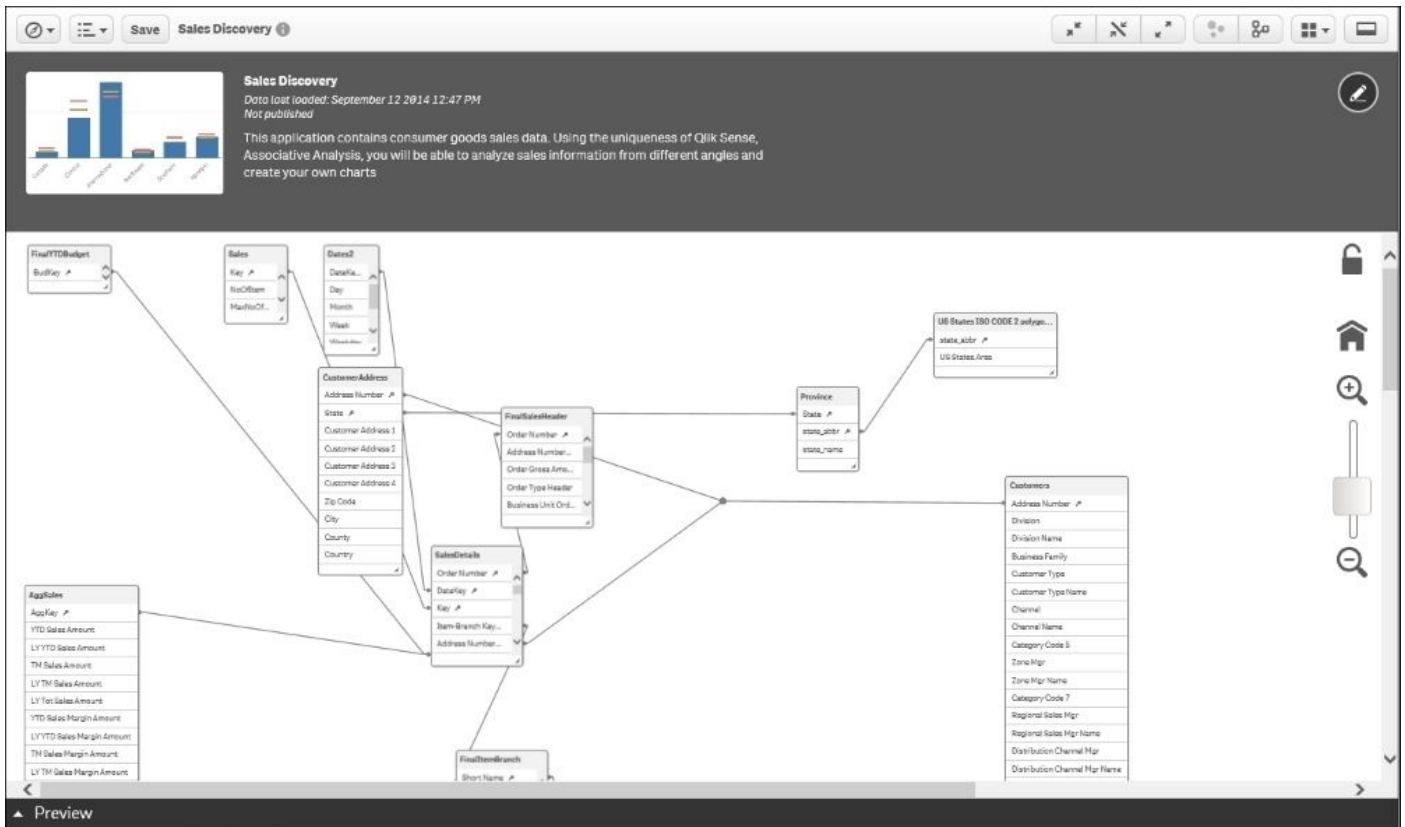


## *What regions are they selling to?*

As you can see, the Sales Discovery application provides a 360-degree view of a sales analysis. This is primarily driven by Qlik's associative indexing engine that drives all Qlik-based applications. Additionally, like most analysis processes, the path to discovery of new information cannot be prestaged but rather unfolds based on the next question asked. This is where Qlik Sense excels in enabling a level of interaction with data to drive insight and is only limited by the data that is available. Now, let's turn our attention to how this application was built.

# Building the application

Let's start our review of the heart of a Qlik Sense application, the data model. As you can see from the following screenshot, there are 12 tables in the Sales Discovery associative model. At the heart of this application is the SalesDetails table. All these tables were created through **Data Load Editor**, which was covered in [Chapter 5, Authoring Engaging Applications](#). It is worth noting that Qlik and Qlik partners provide both general-purpose connectors and specialized connectors to access a broad array of data sources.



*The Sales Discovery model*

Let's dig a bit deeper into the key tables. The key tables that drive this application are covered in the upcoming sections.

## The SalesDetails table

The SalesDetails table contains all the key information about the sales transaction for a specific order. This includes information such as the order number, date, and so on, as shown in the following screenshot:

SalesDetails		Preview of data							
Rows	207976	Order Number	DateKey	Key	Item-Branch Key	Address Number	AggKey	BudKey	Ship To
Fields	54	200286	41476	7/21/2013200286CR-10799	CR-10799	10025919	100-10025919-28401	10025919	10025919
Keys	7	205953	41452	6/27/2013205953CR-10799	CR-10799	10025919	100-10025919-28401	10025919	100840
Tags:	\$key \$numeric \$integer \$timestamp \$date \$ascii \$text	202518	41402	5/8/2013202518CR-10799	CR-10799	10025919	100-10025919-28401	10025919	100786
		202147	41402	5/8/2013202147CR-10799	CR-10799	10025919	100-10025919-28401	10025919	100786
		205954	41452	6/27/2013205954CR-10731	CR-10731	10025919	100-10025919-27550	10025919	100786
		200285	41476	7/21/2013200285CR-10384	CR-10384	10009985	173-10009985-20910	10009985	101263

## The SalesDetails table

### The Customers table

The Customers table contains all the key information about the customer: channel, region, account management, and so on:

Customers		Preview of data										
Rows	638	Address Number	Division	Division Name	Business Family	Customer Type	Customer Type Name	Channel	Channel Name	Category Code 5	Zone Mgr	Zone Mgr Name
Fields	29	10015793	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N
Keys	1	10019000	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N
Tags:	\$key \$ascii \$text \$numeric \$integer	10007120	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N
		10011981	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N
		10009656	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N
		10025248	2	Domestic	R3	G2	Domestic	C7	Distribution		ZM5	Western Zone N

## The Customers table

### The AggSales table

The AggSales table contains all the sales KPI information and is associated with the model so that sales information is available by customer, product, region, and so on:

AggSales		is Amount	LY YTD Sales Amount	TM Sales Amount	LY TM Sales Amount	LY Tot Sales Amount	YTD Sales Margin Amount	LY YTD Sales Margin Amount	TM Sales Margin Amount
Rows	64118	\$0	\$49,711	\$0	\$8,461	\$244,602	\$0	\$23,881	\$0
Fields	14	\$34	\$0	\$0	\$0	\$0	\$19	\$0	\$0
Keys	1	\$0	\$27,301	\$0	\$594	\$204,140	\$0	\$13,478	\$0
Tags:	\$key \$ascii \$text \$numeric \$integer	\$0	\$5,859	\$0	\$0	\$16,294	\$0	\$2,242	\$0
		\$0	\$27,438	\$0	\$4,498	\$68,821	\$0	\$12,512	\$0
		\$0	\$0	\$0	\$0	\$0	\$0	\$0	\$0

## The AggSales table

### US States ISO CODE 2 polygons

The US States ISO CODE 2 polygons table drives the map visualization in the **US Regional Analysis** sheet. The key field is defined by the state, which drives the associative sections, and the field US States\_Area is an imported **Keyhole Markup Language (KML)** file that contains the map's geographic information. This is stored as blob data in the model, and the map object interprets this information when used in a sheet. This table is shown in the following screenshot:



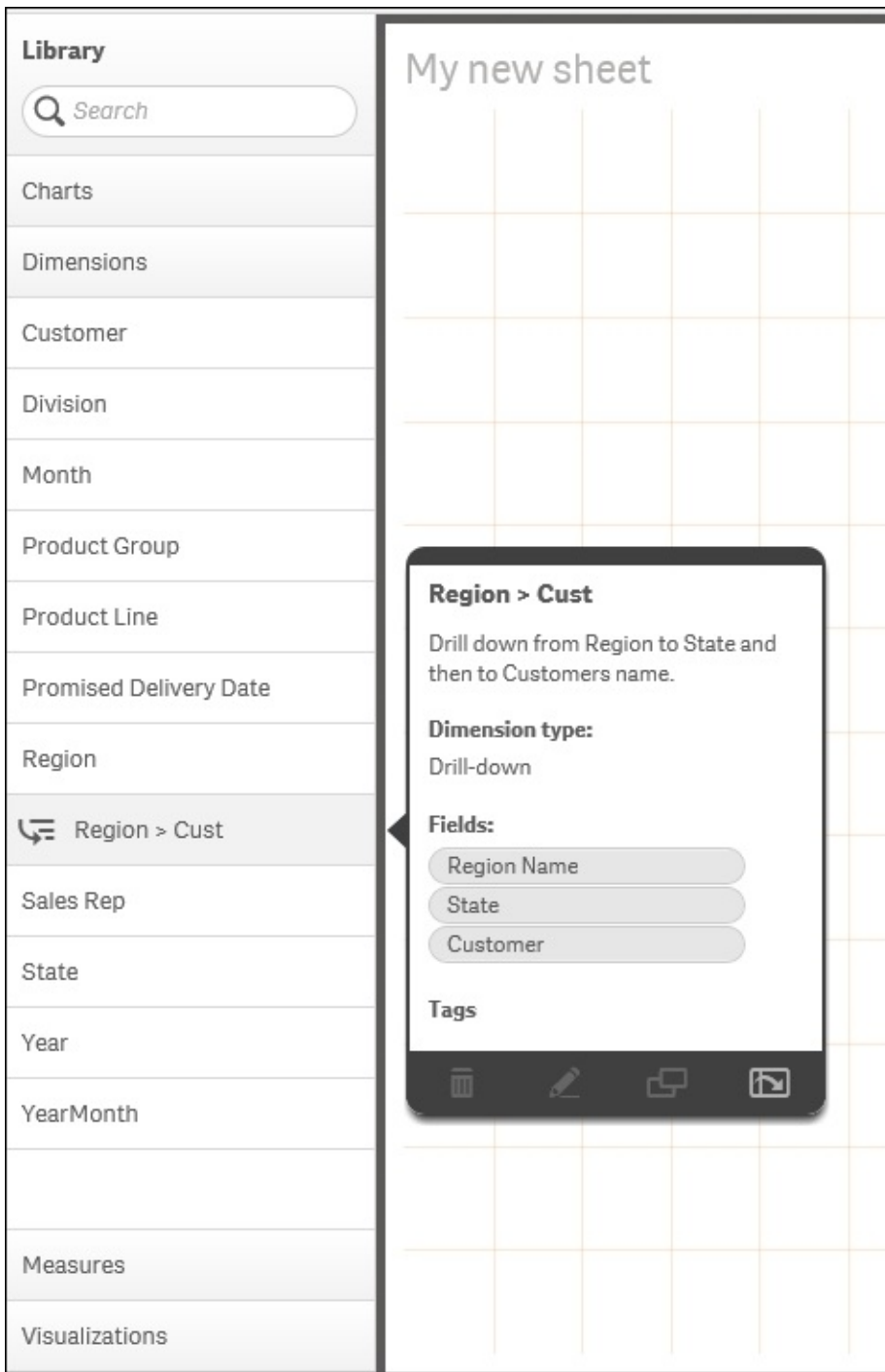
# Analyzing the Sales Discovery Library

Now, let's turn our attention to what has been exposed in the Sales Discovery Library by the developer to facilitate the creation and sharing of personal sheets.

## Dimensions

In the next screenshot, we can see the dimensions that were created. One particular dimension that needs attention is the **Region > Cust** dimension, which provides a drill-down navigation from **Region Name** to **Customer**. This capability usually requires extensive modeling or complex scripts in other BI software products, but with Qlik Sense, this is a simple selection process when creating a dimension. This is another example of the power of Qlik's associative indexing engine in action, but this time, easing the development of navigation within the application.

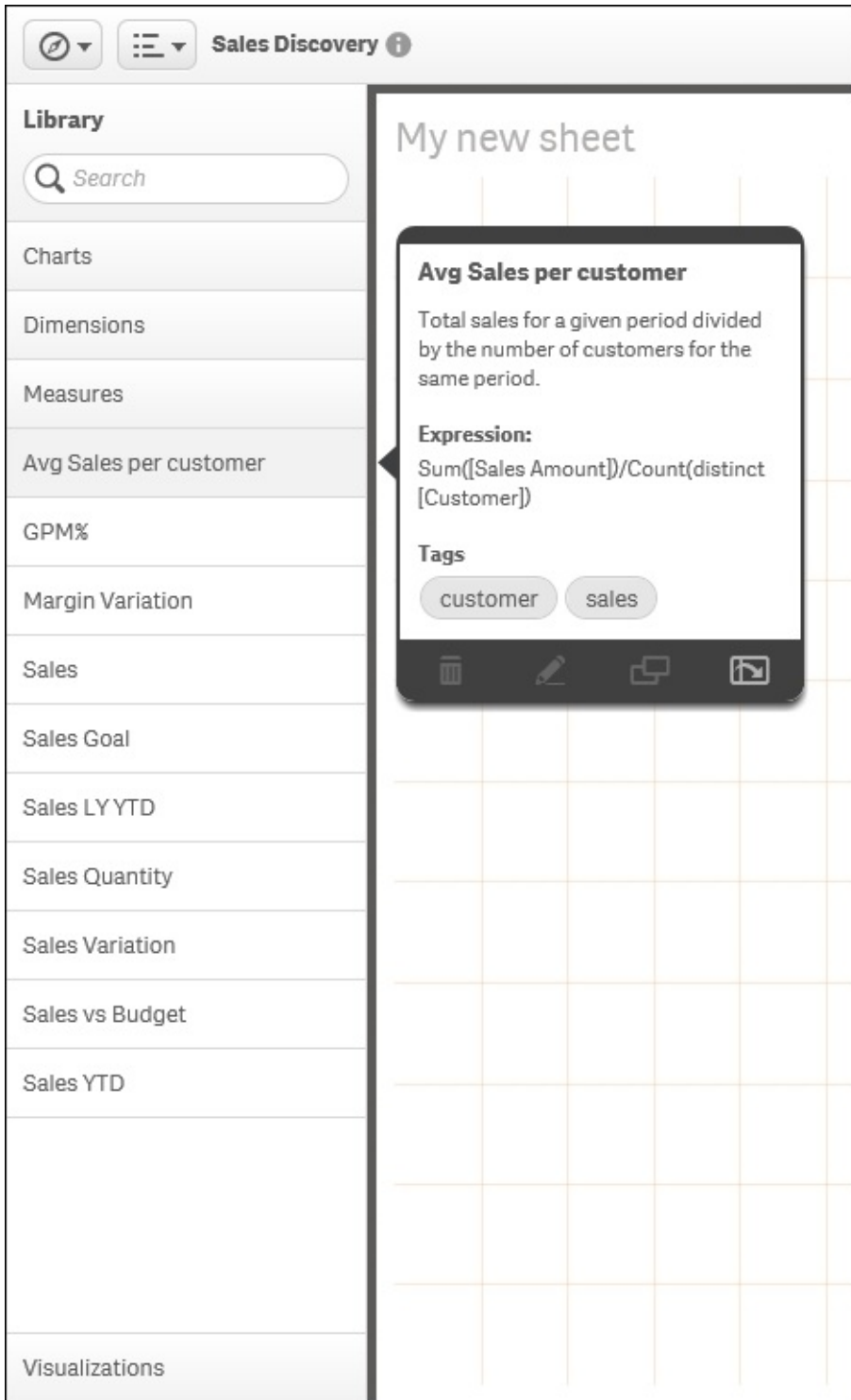




## *Dimensions*

## **Measures**

The next area to cover is **Measures**. These are calculated expressions that most often form the KPIs within an application. In the following screenshot, we can see the list of measures that are used and exposed to contributors to allow them to create private sheets. Note that hovering the pointer over any of these objects makes a preview popup appear to provide additional context. In this case, you can see how the measure is calculated. The following screenshot shows **Measures**:



### *Measures*

Additionally, the following table contains the measure definitions that directly tie to the KPIs used in this application:

Measure	Calculation
<b>Avg Sales per customer</b>	Sum ([Sales Amount]) / Count(distinct [Customer])
<b>GPM%</b>	Sum ({<[Product Group Desc] = {*>}[Sales Margin Amount]) / Sum ({<[Product Group Desc] = {*>}[Sales Amount])

<b>Margin Variation</b>	$(\text{Sum} ([\text{YTD Sales Margin Amount}]) / \text{sum} ([\text{LY YTD Sales Margin Amount}])) - 1$
<b>Sales</b>	$\text{Sum} ([\text{Sales Amount}])$
<b>Sales Goal</b>	$\text{Sum} ([\text{YTD Budget Amount}])$
<b>Sales LY YTD</b>	$\text{Sum} ([\text{LY YTD Sales Amount}])$
<b>Sales Quantity</b>	$\text{Sum} ([\text{Sales Quantity}])$
<b>Sales Variation</b>	$(\text{sum} ([\text{YTD Sales Amount}]) - \text{sum} ([\text{LY YTD Sales Amount}])) / \text{sum} ([\text{LY YTD Sales Amount}])$
<b>Sales vs Budget</b>	$\text{Sum} ([\text{YTD Sales Amount}]) / \text{Sum} ([\text{YTD Budget Amount}]) - 1$
<b>Sales YTD</b>	$\text{Sum} ([\text{YTD Sales Amount}])$

## Visualizations

The last category of objects in **Library (Master items)** is **Visualizations**. These are preformed visualizations that are typically the most popular or requested visualizations. They are defined to help facilitate a user's analysis and can be easily dragged and dropped onto a private sheet. Here, we see a trend line chart for **Number of shipments vs late vs on time**. Each of these visualizations contains predefined dimensions, measures, and chart definitions:

Sales Discovery

Library

Search

Charts

Dimensions

Measures

Visualizations

- % On time shipments
- Customers
- Filter Pane
- Margin Variation Widget
- Number of shipments vs...
- Sales and number of cus...
- Sales by State
- Sales by year
- Sales Evolution
- Sales Variation Widget
- Sales vs Budget Widget
- Sales: YTD vs LYTD vs YT...
- Ships to

My new sheet

**Number of shipments vs late vs on time**

Month	On Time	Late	Total
Jan-2013	0	3000	3000
Feb-2013	0	2500	2500
Mar-2013	0	2200	2200
Apr-2013	0	2500	2500
May-2013	1000	2800	3800

Shipment performance over time

Tags

- Late
- Month
- On Time
- Shipment
- Year

*Visualizations*



# Summary

In summary, Qlik Sense provides unique capabilities to meet the challenging task of analyzing sales data. Without the capabilities offered by Qlik, this task can be difficult due to the size of the data and the many perspectives that can be taken in trying to understand customer buying habits, sales representative productivity, and the responsive nature of the organization in meeting customer needs. Qlik's associative indexing engine powers this exploration. This means meeting these requirements is no longer challenging at all.

In the next chapter, we will explore how Qlik Sense will meet the needs of Human Resource Discovery.



# Chapter 11. Human Resource Discovery

Just like the previous chapter, this chapter will show you how to apply Qlik Sense to the challenges of analyzing real data. This chapter's example and many others are available for you to explore on <http://sense-demo.qlik.com>. Again, make sure you bookmark this link, as more demonstrations and examples are constantly being added and updated.

This chapter is about the analysis of human resources data, and it covers the following topics:

- General information about common KPIs
- What a typical data model would look like
- An example of how to use the global selector
- Examples of dimensions and measures



# The business problem

The term Human Resources analysis covers a wide area of KPIs that use data from a number of different data sources.

It could be that you want to analyze in-house data, for example, the efficiency of the recruitment process and the costs tied to it. It could just as well be analysis of external data, for example, different employee surveys or sentiment analysis on social media.

Just to give you an idea, we have compiled a list of some of the most common areas to investigate when preparing a Human Resources analysis:

- **Recruitment:** This measures the efficiency of the recruitment process, for example, what is the recruitment cost per employee? What is the average lead time to recruit?
- **Employee satisfaction and retention:** This measures employee loyalty, for example, what is the average satisfaction (as measured by a survey)? What is the employee turnover?
- **Training:** This covers the following questions as examples: What is the total expenditure on training? What percentage of the employees have gone through the training? What is the number of training hours per employee?
- **Health and safety:** This covers the following questions as examples: What is the number of accidents per year? How many employees are of adequate health and get safety training? How much does health and safety prevention cost?
- **Career and compensation:** These cover the following questions as examples: What is the average salary rate? How does it compare to the national average? How much is the salary cost compared to the sales turnover? What is the cost of social and medical insurances?

All of the preceding KPIs can be split by a dimension, month, department, position, tenure, age, and so on.

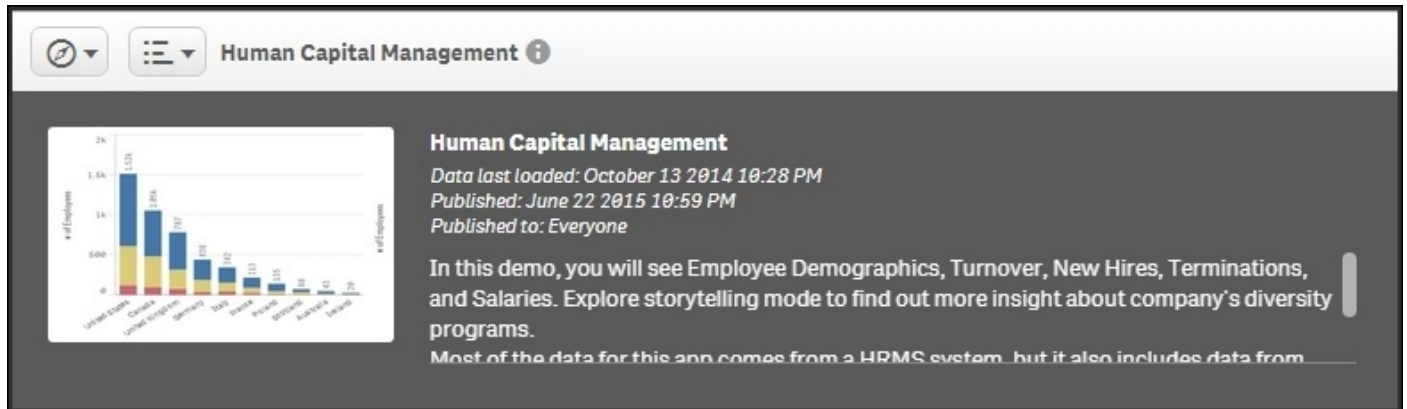
It might be that you don't have data for all the preceding KPIs, but we can assure you that if you do, you will find it worthwhile to analyze them.



# Application features

On our demo site, we have a human resource app. You can find it on <http://sense-demo.qlik.com>, by clicking on the **Human Capital Management** link. In it, you will find a subset of what you can analyze in HR data. Mainly, it analyzes training investments and employee satisfaction.

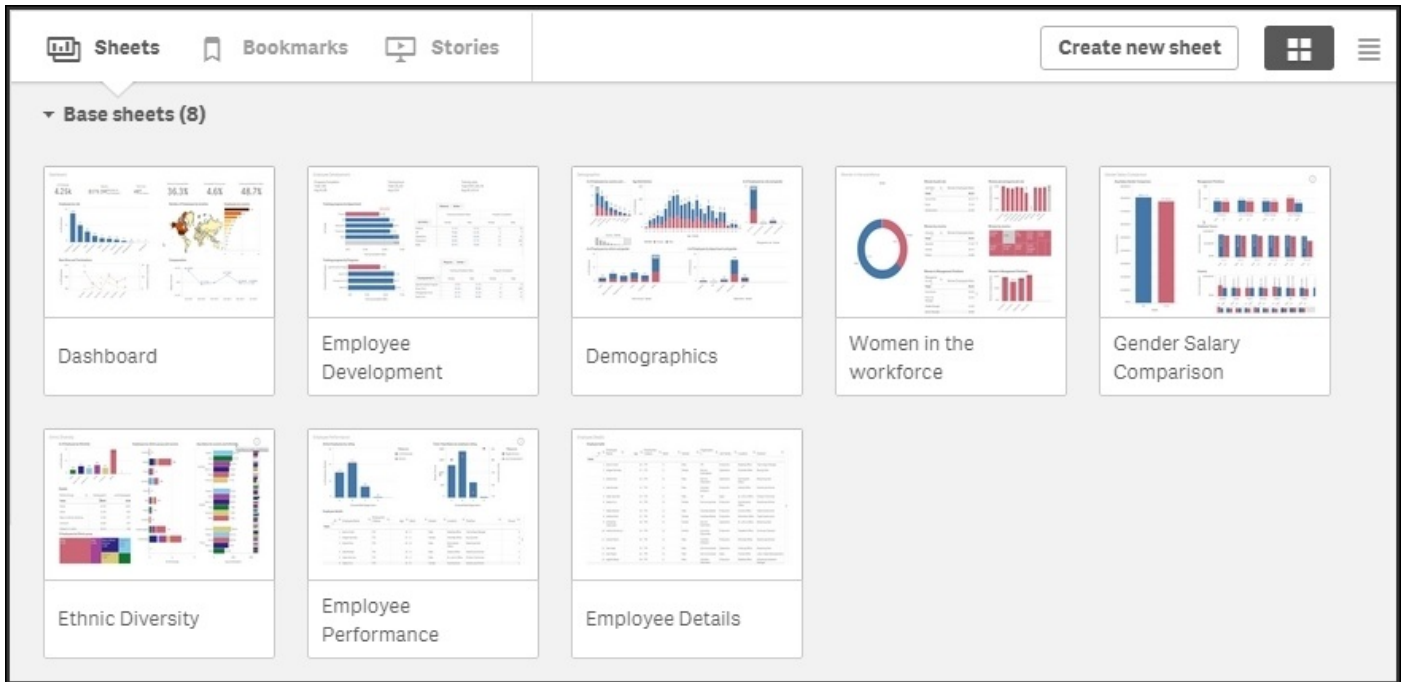
When you open the app, you will first see the app overview, with a small description of the app and a thumbnail in the form of a small bar chart, as shown in the following screenshot:



*The overview of the Human Capital Management application*

The following is this overview; you will see a number of sheets. These are created according to the dashboard analysis report principles described in [Chapter 5, Authoring Engaging Applications](#). This means the leftmost sheet is an overview, very much like a dashboard, whereas the other sheets are prepared for analysis and detailed information.

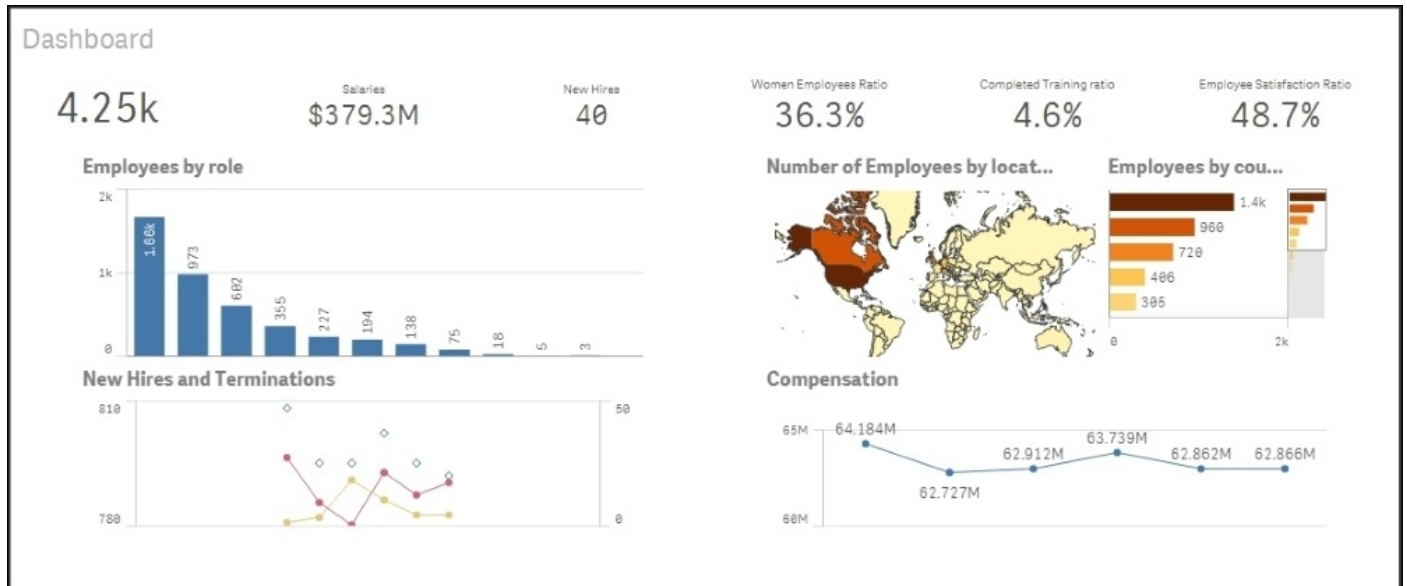
If you click on the **Stories** button, you will see that the app also contains a story—a story that can be used to present the data in the app. It can also be used as an introduction to the app the first time you open it.



*The sheets on the app overview page*

# Sheets

The first sheet is called **Dashboard**, and if you open this, you will see several key numbers, a couple of charts, and a map. This overview is designed so that you can quickly get an overview and a brief understanding of the information without having to make any selections.



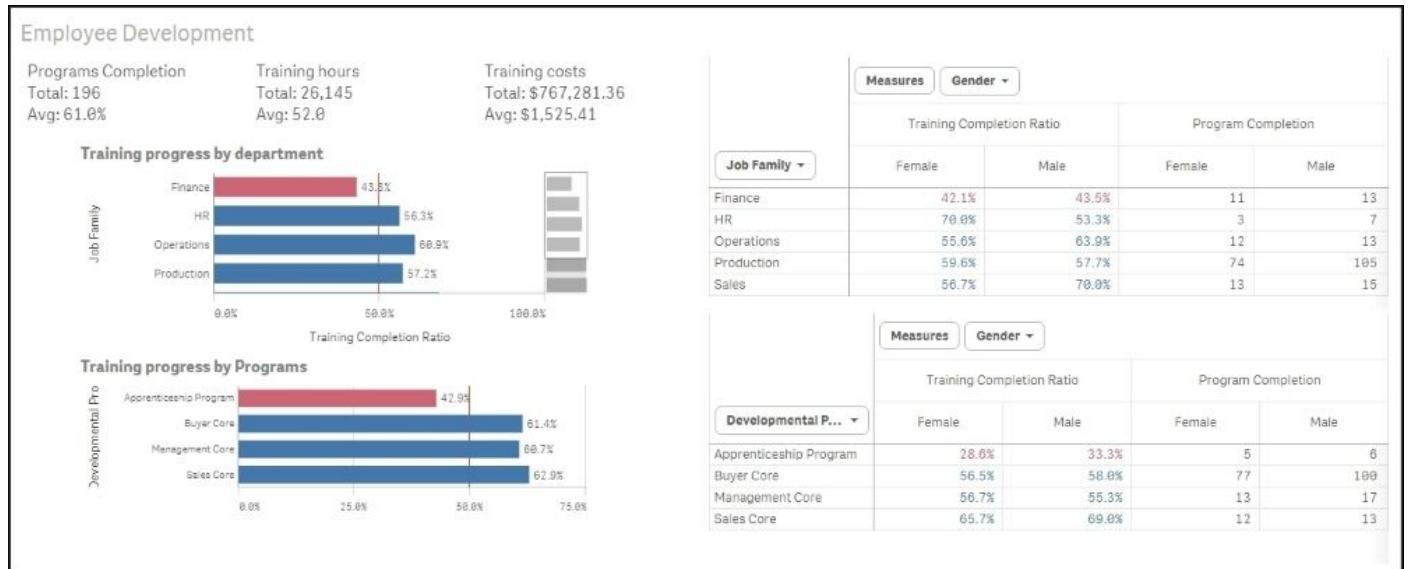
*The first sheet – the Dashboard*

The top-left chart and the map show the number of employees per role and per country. The two charts at the bottom show the number of hires, number of terminations, and total compensation over time. Note that this sheet does not contain any filter panes because it should not encourage making selections.

The other sheets contain more detailed information, ordered by topics, such as **Demographics**, **Ethnic Diversity**, and so on. The final sheet contains a table only, showing the details about what the application captures, should the user be interested in drilling down to the lowest level.

# Training costs

This application contains information that covers only some of the KPIs mentioned in the previous section. One area that it covers well is training. Hence, our first question could be what the total expenditure on training is. The answer to this can easily be found from the **Employee Development** sheet.



*The Employee Development sheet*

In the top part of the sheet, you can find a textbox containing **Total:** under **Training costs**.

The next question is what the percentage of employees who have gone through the training is. The answer to this can be found from the same sheet. All charts on this sheet show the training completion ratio, split per department or program. By clicking on a chart, you can drill down to the data and explore how the numbers vary between departments, programs, job, course type, and course name.

When you analyze data in a Qlik Sense application, you will realize that there are many ways of using it. For example, say that you want to look at the training progress by gender, to see whether there is any difference between men and women. You have already found the charts showing training progress on the **Employee Development** sheet, but these only show the progress by department and by program.

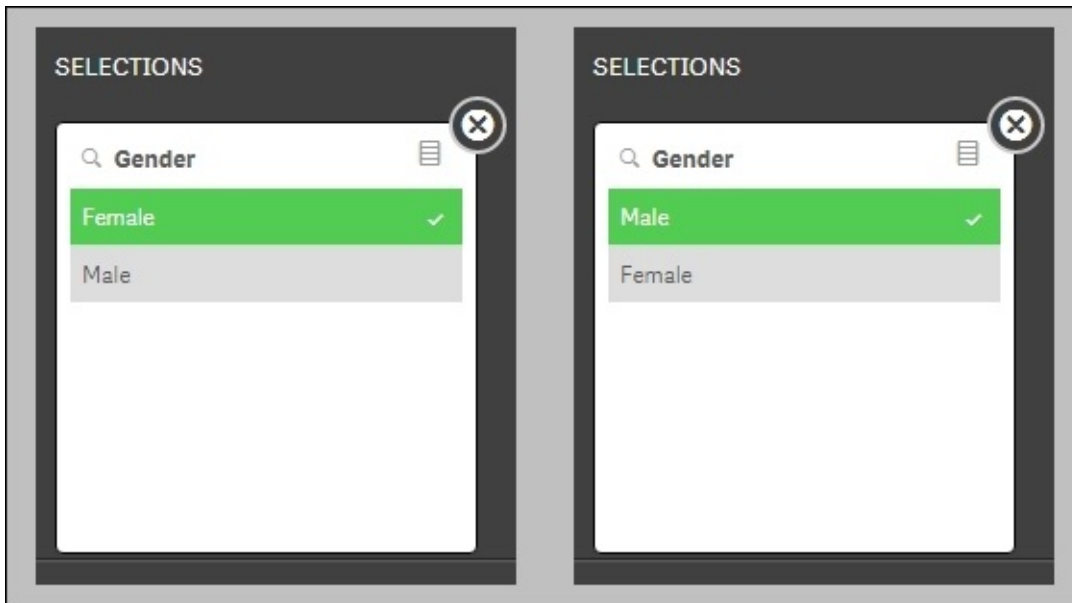


*Chart showing training progress by department*

# Using the global selector

If you have been authorized to create your own visualizations, you can simply go to edit mode, duplicate the sheet, and then drag **Gender** onto the chart, thus replacing the existing dimension with **Gender**.

But even if you aren't allowed to change anything, you can still do your analysis. You just have to do it in a different way. What you could do is use the global selector (the rightmost button in the **SELECTIONS** bar) to select **Female** first, and then **Male**, as shown in the following screenshot:



*Selecting Gender in the global selector*

When you now close the global selector and return to your **Employee Development** sheet, you can toggle between **Male** and **Female** using the Step back and Step forward buttons to the left in the **SELECTIONS** bar, as shown in the following screenshot. There's also a Clear all selections button in this bar:



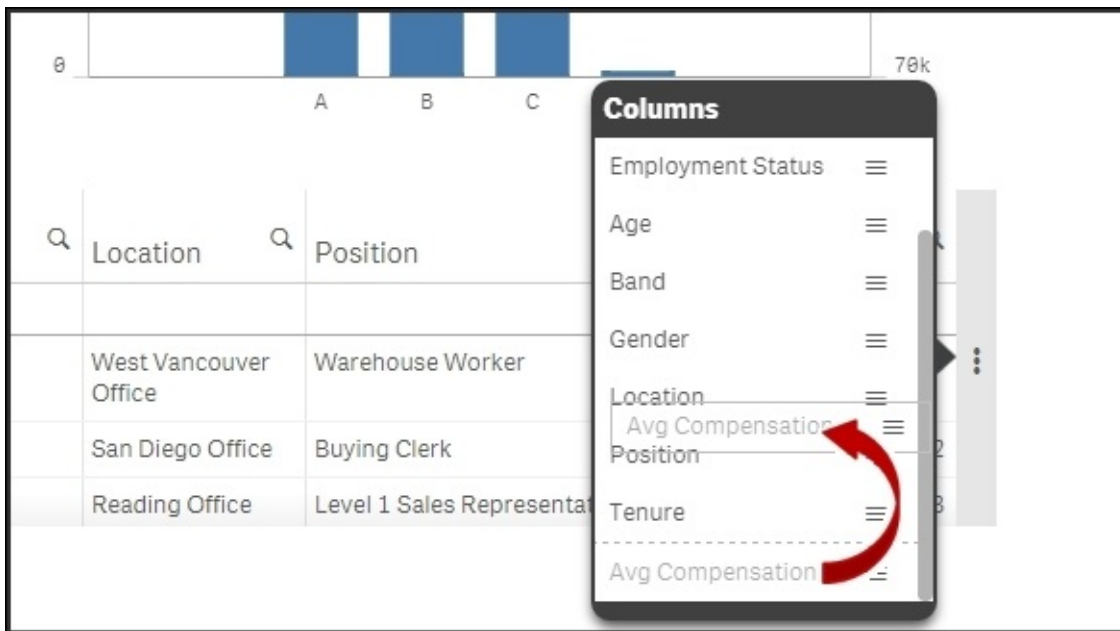
*The left part of the Selections bar has the Step back, Step forward, and Clear all selections buttons. Further right, the current selections are listed*

This way, you can see how the chart changes as you toggle between the data of male and female employees.



The next question could be about employee compensation. For this, you need to go to the **Employee Performance** sheet. Here, you will find a table showing all employees and the compensation attached to them. By clicking on **Avg Compensation** in the table, you can sort the employees in ascending or descending order, and can thus get a good overview of the span.

**Avg Compensation** is the rightmost column in the table, and due to the responsive design of Qlik Sense, it may be that this column is hidden. If so, just click the three dots on the edge of the table, and you can change the order of the columns so that this column becomes visible.

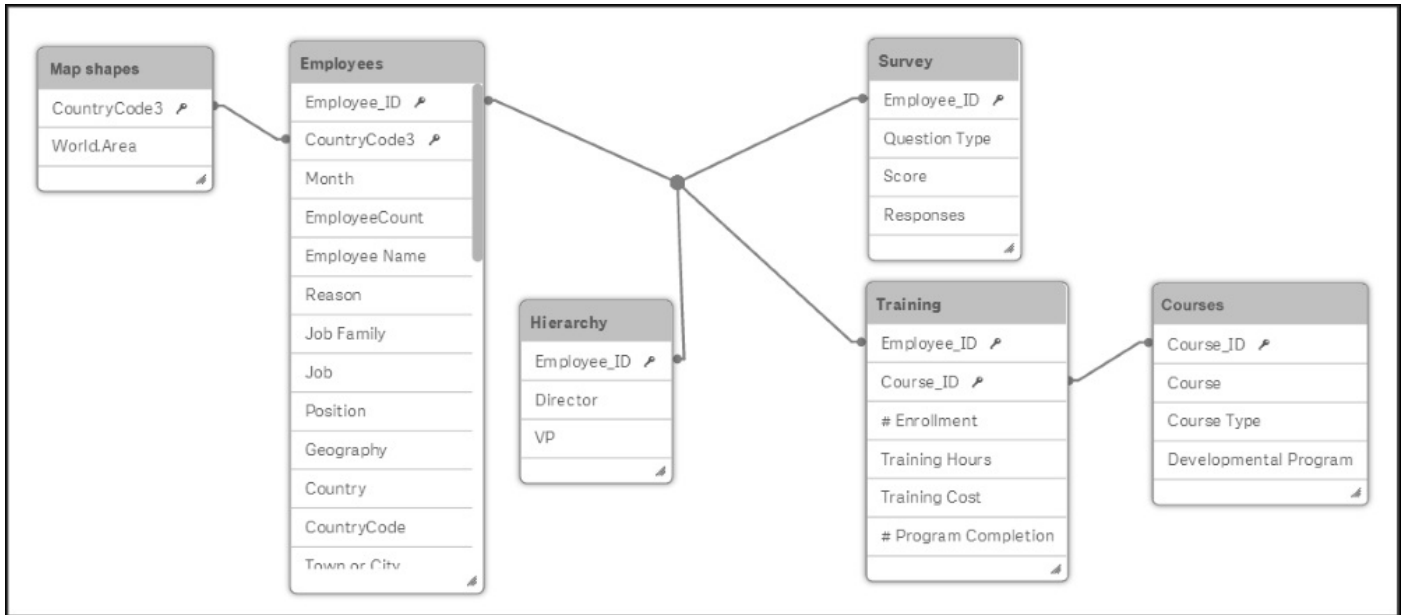


As users, we would probably also want to see a chart showing the average and total salary costs, split by department, but unfortunately, this has not been created by the app developer. However, in a real-life situation, a user should be empowered to create such charts. After all, it is impossible for an application developer to foresee all the needs of a user. Hence, this is a good example of the need for self-service data discovery.



# How the application was developed

The data model for the **Human Capital Management** application looks similar to what is shown in the following diagram:



There are six tables in this application:

- **Employees:** This is the main table, which has one record per employee and month. It contains all the relevant information about the employee such as country, position, salary, and so on. It would probably be a cleaner data model if this table were split into one table containing employee information that doesn't change over time and another table with the time-dependent information.

However, since the QIX engine analyzes the data just as efficiently either way, we don't see any great benefit in spending time structuring the data more.

Preview		Preview of data							
Employees		Employee_ID	CountryCode3	Month	EmployeeCount	Employee Name	Reason	Job Family	Job
Rows	4787	231	GBR	Jun-2014	1	Aaron Cohen	Import created action	Production	Engineering
Fields	35	1298	CAN	Jun-2014	2	Abigail Kennedy	*New Job	Operations	Buyer
Keys	2	1192	USA	Jun-2014	3	Abram Ruiz	Existing Position	Operations	Receiver
Tags	\$key \$numeric \$integer \$geoname \$ascii \$text	49	USA	Jun-2014	4	Ada Morales	Existing Position	Production	Warehouse
		1013	CAN	Jun-2014	5	Adam Garrison	Existing Position	Sales	Sales
		720	GBR	Jun-2014	6	Adara Cruz	Import created action	Production	Warehouse
		1116	CAN	Jun-2014	7	Adda Hejman	Existing Position	Production	Engineering

*Preview of the Employees table*

- **Hierarchy:** This table contains information about the hierarchy of the employee, such as who the manager of the employee is, and who the VP of the employee is.
- **Survey:** This table contains the results from an external survey made on employee satisfaction. Such surveys are usually made once in a year, so if the results from

several surveys are kept in this table, the key needs to hold information not only about the employee but also about the year in which the survey was made.

- **Training:** This table contains information about the training sessions attended. Hence, if an employee has attended two courses, two records are stored. The table also contains costs associated with the training sessions.
- **Courses:** The possible courses are stored in this separate table.
- **Map shapes:** This table lists all countries. It has one record per country and could, in principle, hold demographic information about the country. However, in this case, it only holds the map information—the shapes of the countries—used in the map object, which is in the user interface.

Note that this application has fields used for measures in *several* tables: salary can be found in the `Employees` table, cost for training can be found in the `Training` table, and ratings from the survey can be found in the `Survey` table. This is in sharp contrast to classic BI tools, where all such facts need to be in one single table, the `Facts` table.

# Dimensions

There are many fields that can be used as dimensions, and a large number of them have been added as dimensions to **Library** such as **Employee Name**, **Age Group**, **Department**, and so on.

In principle, any field that a user would want to use as a grouping symbol should be added as a dimension. However, you should not add keys with cryptic names or numbers that should be used as measures.



Dimensions
Create new
Geography
Job Family > Job
Job Role
Management role
Office
Organization
Organization Class
Organization Type
Recruiting Channel

*The dimensions in Library*

A number of measures have also been defined, for example, **# of Employees**, **# of Women**, **Attrition**, **# of New Hires**, **Avg Compensation**, and so on.

It is important that the app developer writes the formula correctly, since this is something that could be difficult for a business user. A business user doesn't always have knowledge about the data model, which is something you need in order to get all the expressions right.











In the following table, you can find some of the measures defined in this app:

Measure	Definition
<b># of Employees</b>	Count(DISTINCT [EmployeeCount])
<b># of Women</b>	Count ({<[Gender] = {'Female'}}> DISTINCT EmployeeCount)

<b>Attrition</b>	Count ({<[Terminated Employee] = {'1'}}>} DISTINCT [EmployeeCount])
<b>Avg Compensation</b>	Avg([Salary])
<b>Completed Training ratio</b>	Sum([# Program Completion]) / Count(DISTINCT EmployeeCount)
<b>Employee Satisfaction Ratio</b>	Avg(Score)
<b>New Hires ratio</b>	(Count ({< [New Hires] = {'1'} >} DISTINCT EmployeeCount) / Count(DISTINCT EmployeeCount))
<b>Terminations</b>	Count ({< [Terminated Employee] = {'1'} >} DISTINCT EmployeeCount)
<b>Wages Amount</b>	Sum(Salary)

Finally, there are also a number of visualizations added to **Library**. These are important, as they help a business user in the initial use of the app.

The most common bar charts and tree maps have been stored here: **Number of employees by role**, **Number of employees by management position**, and so on.

Visualizations	
	# of Employee by Ethnicity
	Avg Salary Gender Com...
	Employees by role
	Gender Employee Distrib...
	Management Positions - ...
	Salary difference by gen...
	Text: Compensantion
	Text: Diversity Rate
	Text: Employee Satisfact...
	Text: Number of Employ...

*The list of visualizations in Library*



# Summary

In summary, the analysis of human resource data is easy when you use Qlik Sense's unique capabilities. Such an analysis can otherwise be difficult due to multiple and disparate data sources holding human resource data. Qlik's associative indexing engine powers this exploration and analysis is made easy for the user.

In the next chapter, we will look at how Qlik Sense can be used to analyze costs, or more specifically, travel expenses.





# Chapter 12. Travel Expense Discovery

The goal of this chapter is to continue our exploration of Qlik Sense with real data, and how it meets the needs of business discovery in your organization. The Qlik Sense application chosen for this chapter is a topic near and dear to most finance departments, Travel Expense Management. Like all the applications covered in this book, feel free to explore this application live at <http://sense-demo.qlik.com>. With that said, let's turn our attention to the following challenge of travel expense management and how Qlik Sense addresses this common business challenge.

In this chapter, we will cover the following topics:

- Common travel expense analysis challenges
- The unique way Qlik Sense addresses these challenges
- How the application was built

# The business problem

Expenses are a major line item of every global company. Traveling cost is a part of every sales and service cycle. Unfortunately, most expense tools capture the transaction but do little to help gain insights about how the expenses were spent, when, and most importantly how, to reduce these expenses when possible. Some key questions include:

- How are expenses tracked versus the budget?
- What is the actual amount spent to date?
- What is our largest expense type?
- How can we reduce expenses?



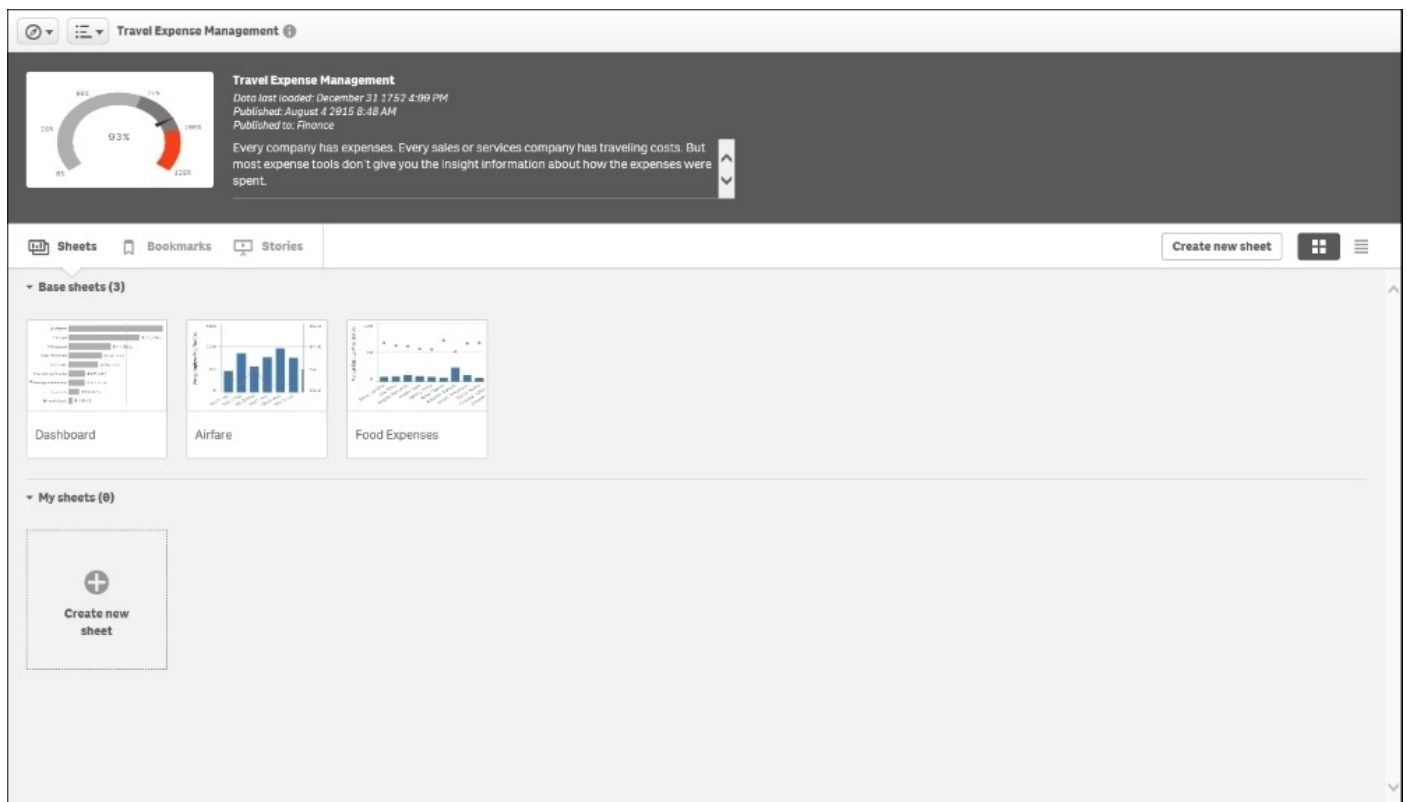
# Application features

Now, let's take a look at the unique way Qlik Sense approaches solving the business problems mentioned in the previous section. Qlik Sense's associative model allows users to answer common questions through filters, but they can also address the more important follow up questions that arise. As you may recall, this type of analysis uses "The Power of Gray", named after the color Qlik Sense assigns to nonassociated elements (potential opportunities for improvement) highlighted in [Chapter 3, Empowering Next Generation Data Discovery Consumers](#).

The key questions include:

- How are expenses tracked versus the budget?
- What is actual amount spent to date?
- Is my department over budgeted?
- What is our largest expense variance?
- What is the meal expense breakdown?
- How can we reduce expenses?

Before we begin, let's review the main sheets within the Travel Expense Management application. As noted in the following screenshot, the application is made up of three sheets: **Dashboard**, **Airfare**, and **Food Expenses**:



*The Travel Expense Management overview*

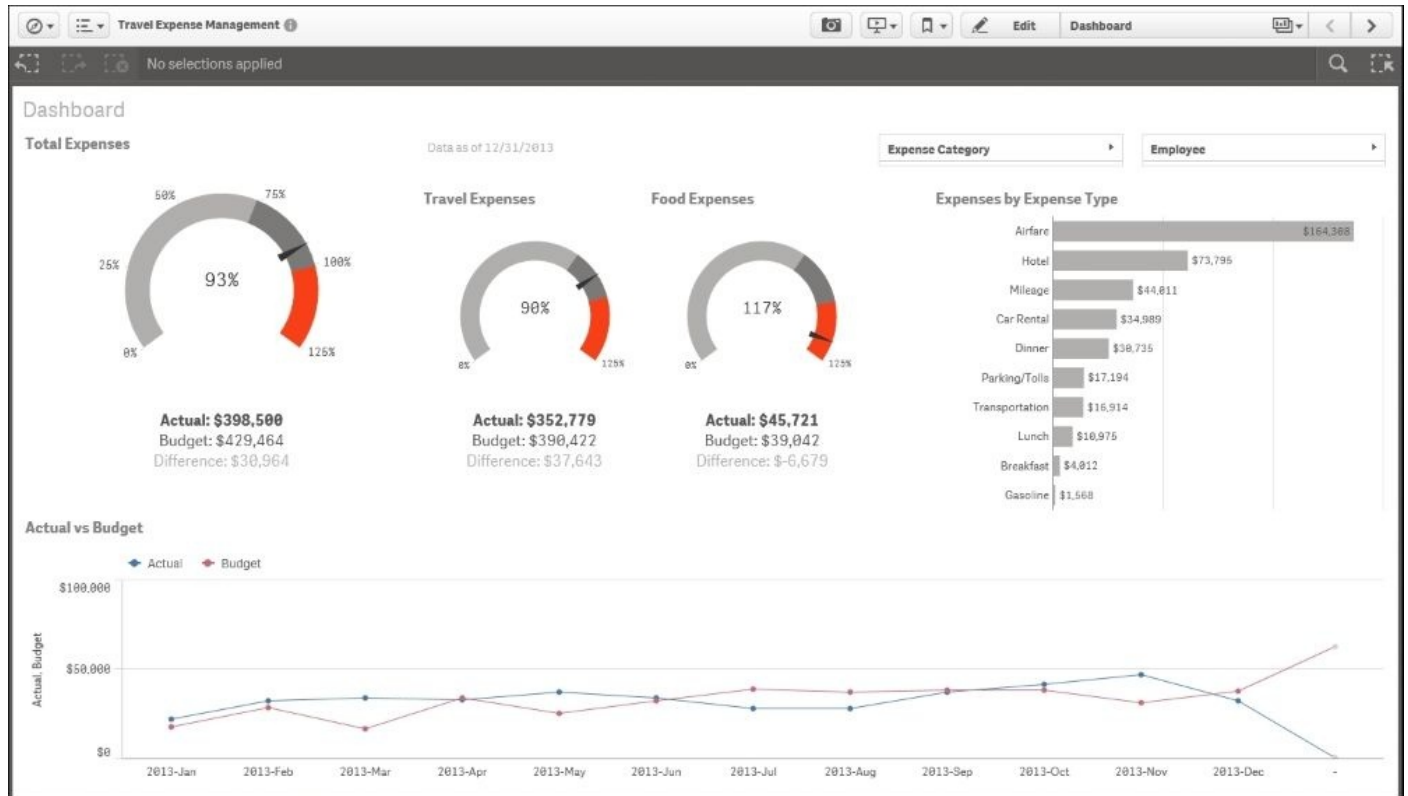
These sheets provide insights into the overall expense management, and the two largest expense categories of airfare and food. With that said, let's now turn our attention to our

first question.

# Tracking expenses

A key question is how to manage departmental expenses on a quarterly basis. How are expenses tracked versus what has been budgeted?

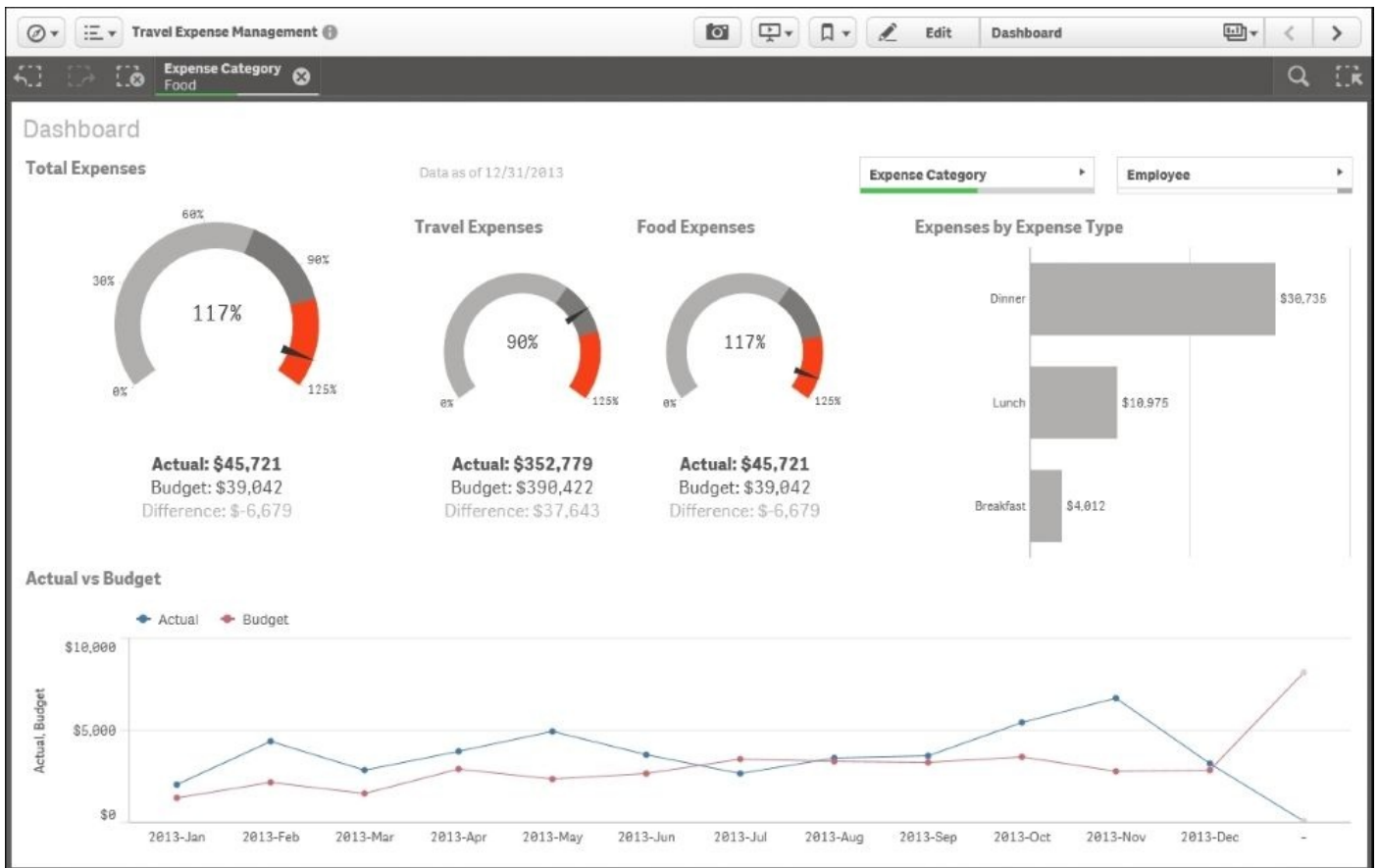
In the following screenshot, we can see in the sample application that **Total Expenses** is below budget by \$30,964. This is good news. Additionally, we see that the largest expense is **Airfare**, and what is more troubling is that **Food Expenses** is running \$6,679 over budget.



*Expenses tracked versus budget*

## Analyzing expenses overspent

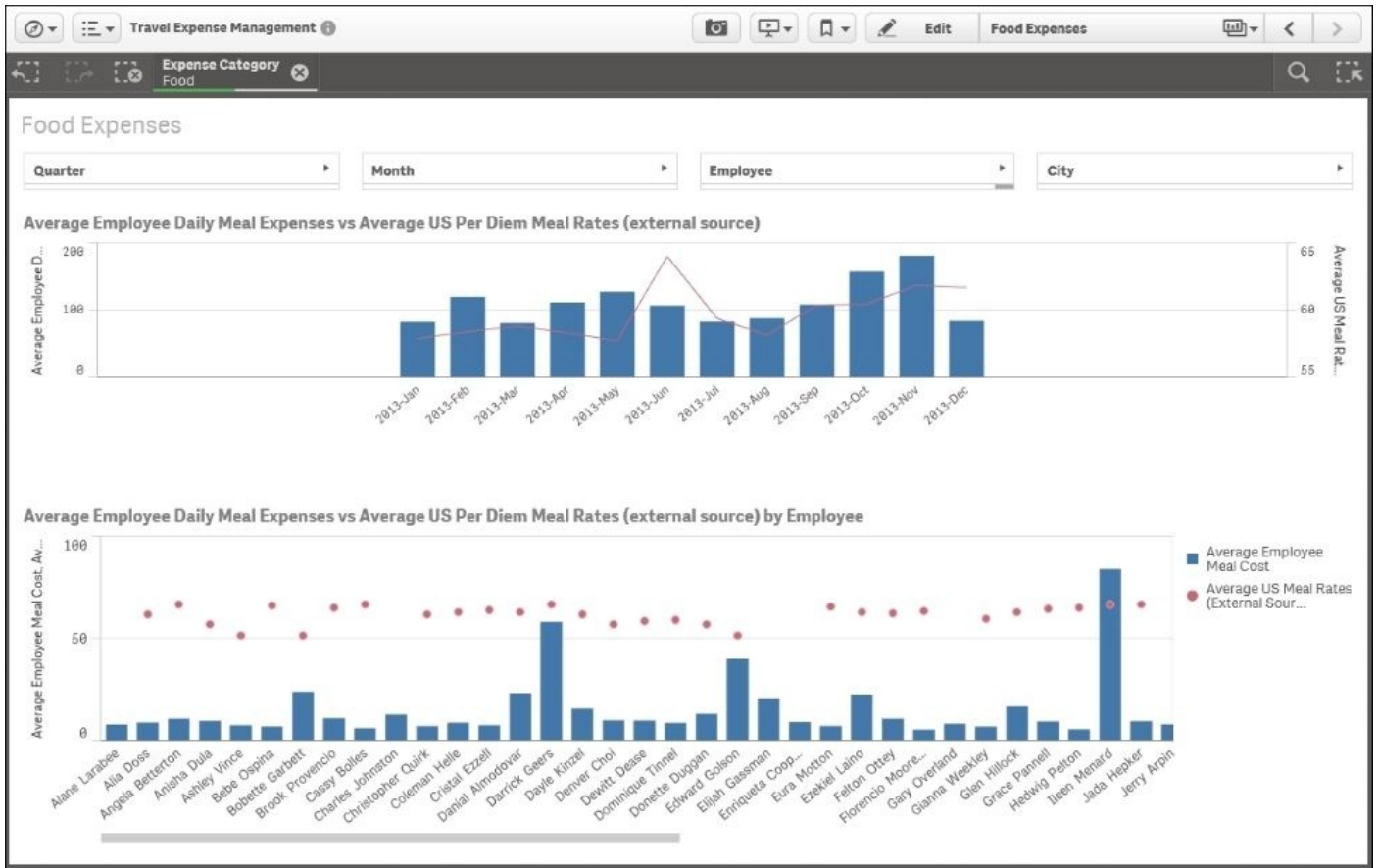
Taking a closer look at food expenses such as breakfast, lunch, and dinner, we can see that for most of 2013 (11 out of 12 months), employees spent more for meals than was budgeted. Exploring a little deeper, we can see that as you'd expect, **Dinner** takes up the majority of the expenses with \$30,735. What's more interesting is that the budget to actual variance starts to sharply grow in May, October, and November.



*The meal expense breakdown*

Now that we have highlighted a problem with food expenses, let's start to use the more detailed information that is available in the **Food Expenses** sheet, shown in the next screenshot. Additional external information is always helpful in variance analysis. In the following screenshot, we can see not only **Average Employee Daily Meal Expenses vs Average US Per Diem Meal Rates (external source)**, but also **Average Employee Daily Meal Expenses vs Average US Per Diem Meal Rates (external source) by Employee** on a monthly basis:





*Average Employee expense versus Average US Meals expenses*

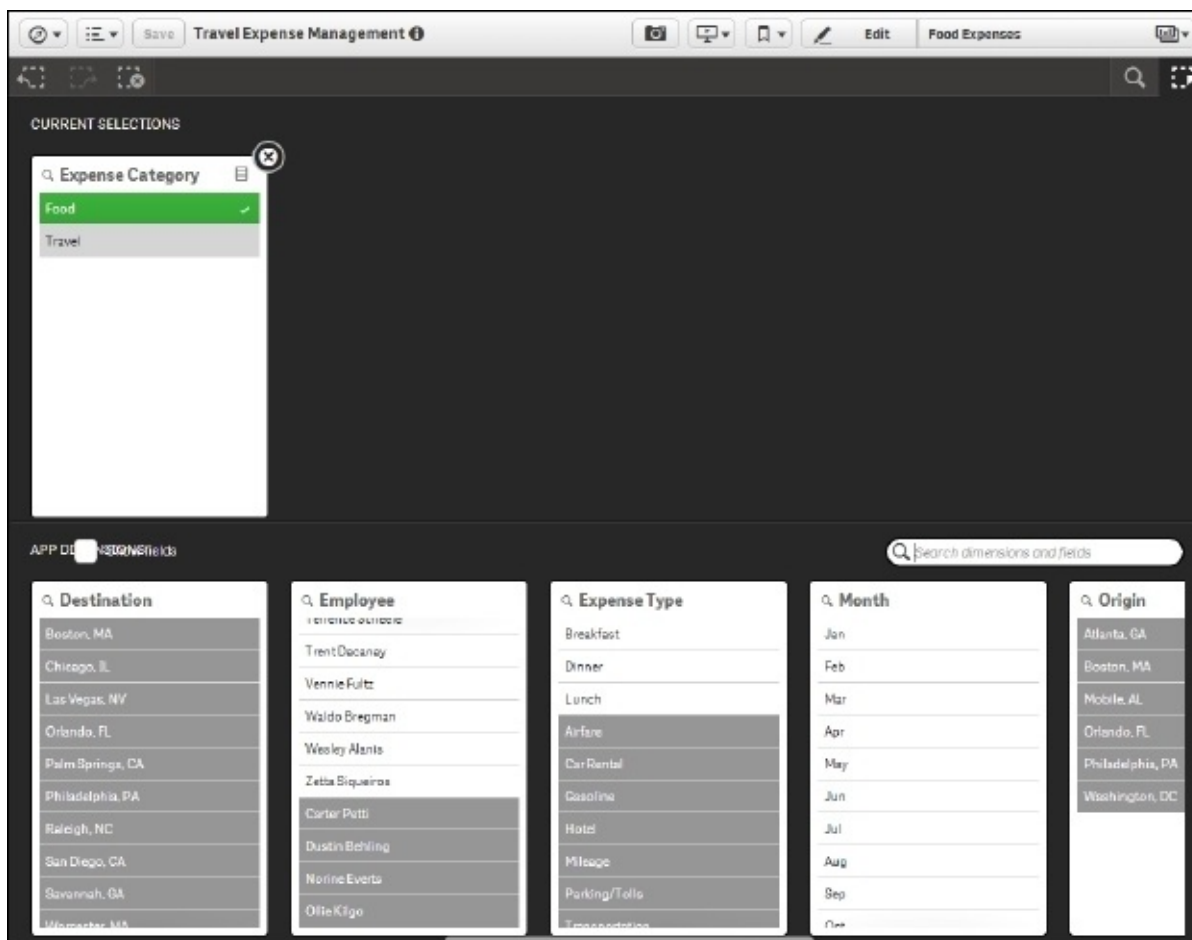
# Digging deeper into the data

Overall, the company seems to be performing well against the US average, but let's dig a bit deeper. For example, are there employees that do not spend on meals, which could be lowering the company average? To find this out, simply select the global filter (also known as the Selections tool) icon, as shown in the following screenshot:



*Global filter*

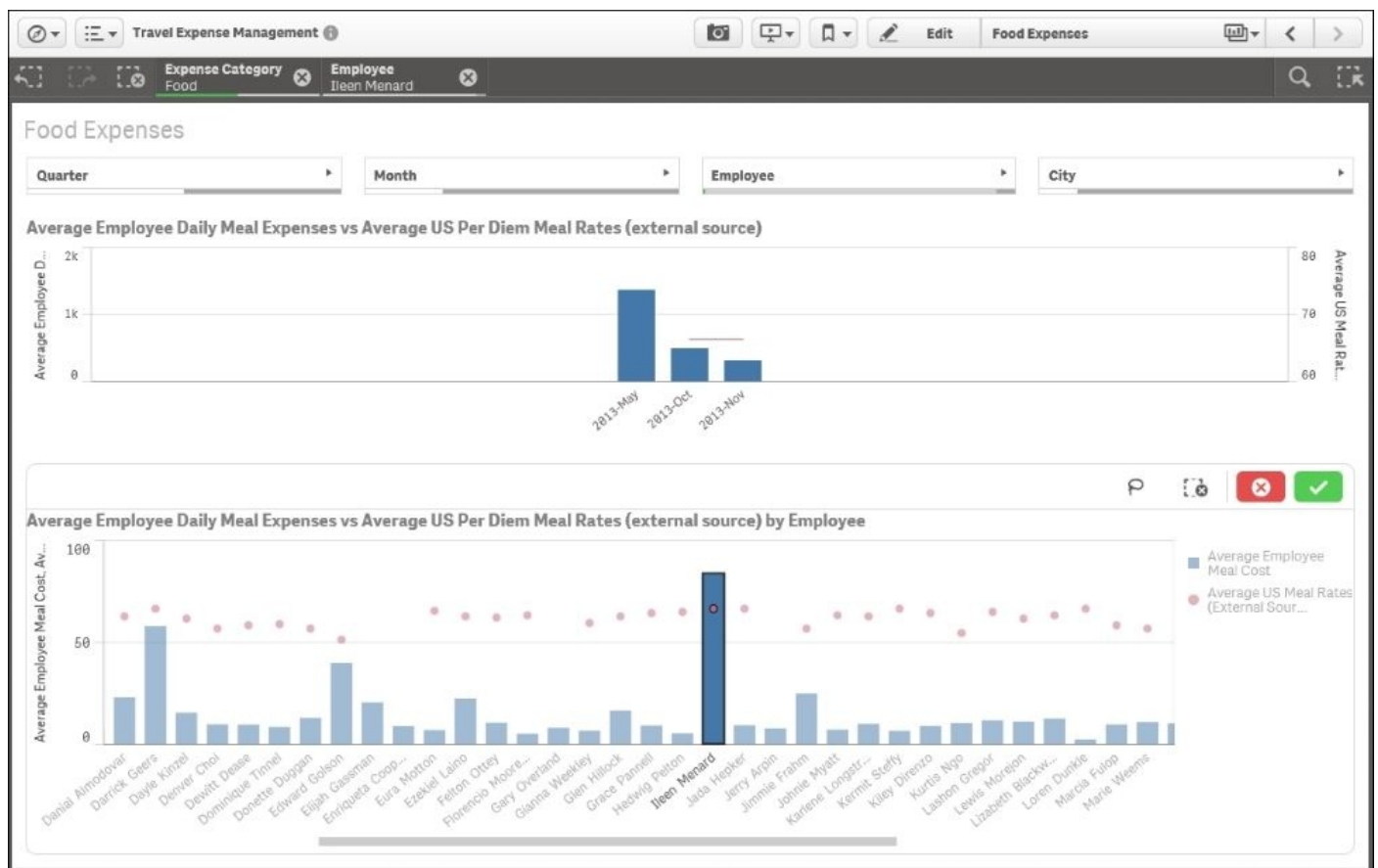
Using “The Power of Gray” (nonassociated elements), we can see in the **Employee** dimension that four employees do not spend on their meals, as shown in the following screenshot:



*Employees who are not spending on food*

Knowing this, we can exit the global filter screen and continue our employee meal analysis. What started off as a travel expense analysis has, through Qlik Sense, narrowed down the analysis to an employee meal analysis. As we scroll through the employees, as shown in the following screenshot, we immediately get to an employee (**Ileen Menard**) who has exceeded the average US per diem allowance significantly, and by selecting **Ileen**

**Menard**, you can see that May was the month with the significant variance. Additionally, the green/white/gray is shown in summary on the selection bar after the dimensions/members are selected in the global filter below:



*Employees that exceed the average for US Meals per diem?*

## Creating an analysis story for travel expenses

Now that we've completed our analysis, let's create a **Travel Expense Analysis** story so that we can share our findings through our organization.

In [Chapter 4, Contributing to Data Discovery](#), we reviewed the role of a contributor, and how to create a Qlik Sense story and publish it so that others may view their analysis. Based on the analysis discussed in the previous section, the travel expense story is made up of three sheets.

### Creating an overview

In the **Overall Expenses** sheet shown in the following screenshot, you can see the **In 2013, the company overall stayed below budget by almost \$31,000. While Travel expenses were well below budget, food expenses were \$6,679 above budget (17%)** annotation as well as the key snapshot gauges of the actual to budget performance total, **Travel Expenses**, and **Food Expenses**:

# Overall Expenses

In 2013, the company overall stayed below budget by almost \$31,000. While Travel expenses were well below budget, food expenses were \$6,679 above budget (17%).

## Total Expenses



**Actual: \$398,500**  
Budget: \$429,464  
Difference: \$30,964

## Travel Expenses



**Actual: \$352,779**  
Budget: \$390,422  
Difference: \$37,643

## Food Expenses



**Actual: \$45,721**  
Budget: \$39,042  
Difference: \$6,679

Data as of 12/31/2013

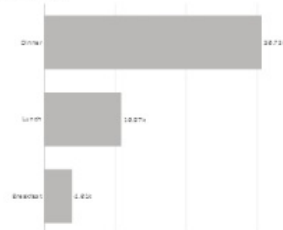
## Sharing our analysis

Now that the overview of the analysis is complete, let's move onto the next step and share what was found in analyzing **Food Expenses**. The **Food Expenses** story sheet, shown in the following screenshot, highlights that for most of 2013 (11 out of 12 months), employees spent more for meals than was budgeted:

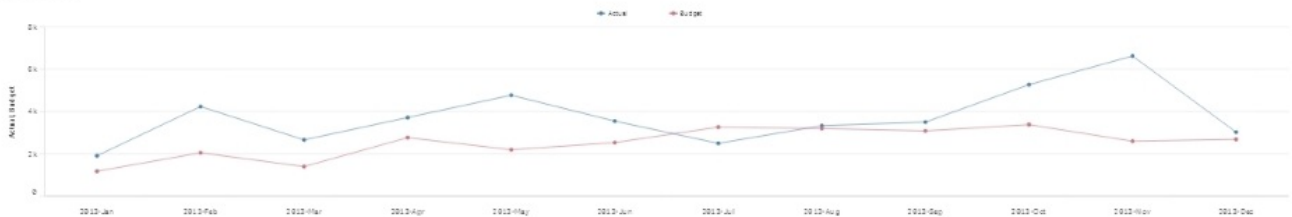
# Food Expenses

Taking a closer look at food expenses such as breakfast, lunch and dinner, we can see that for most of 2013 (11 out of 12 months), employees expenses more for meals than was budgeted.

## Expenses by Expense Type

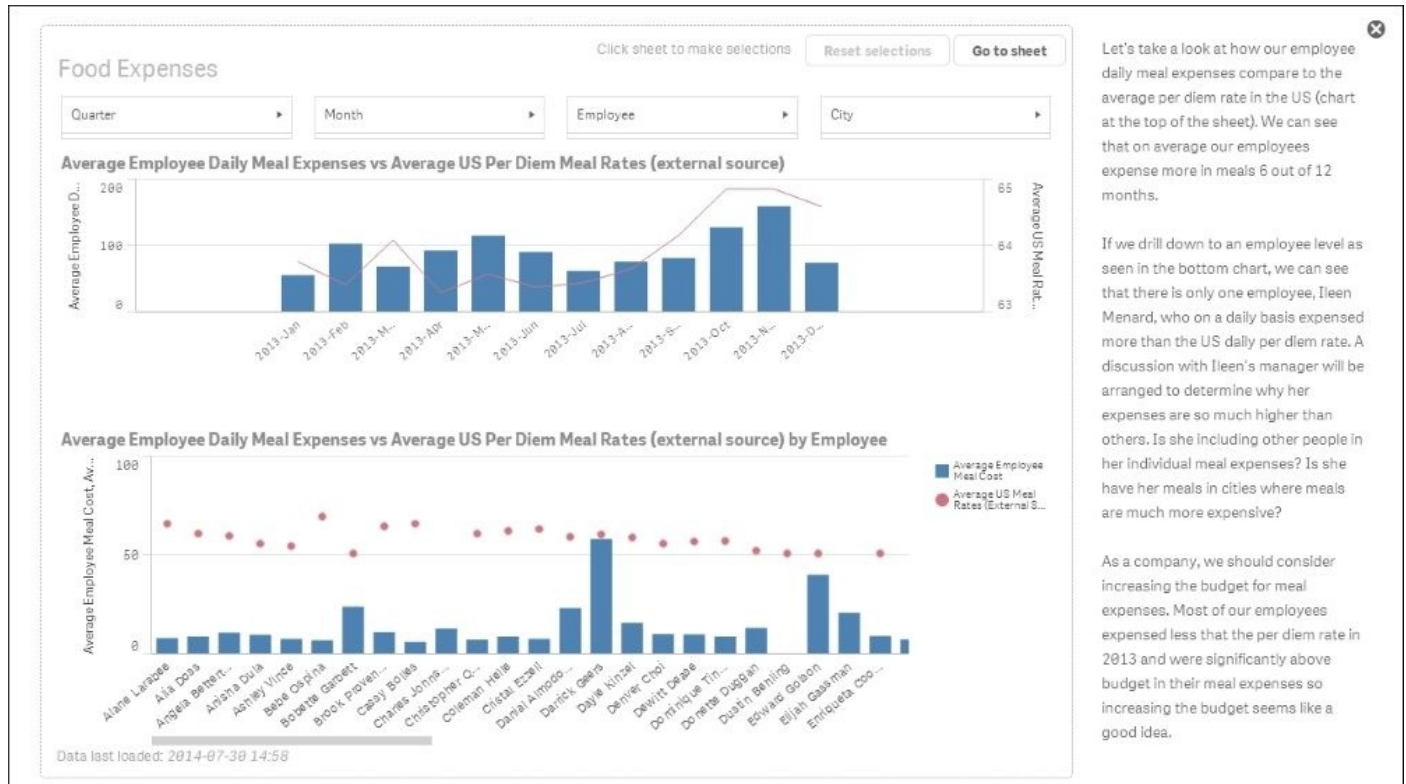


## Actual vs Budget



## Finishing the story

With these two story sheets defined, a final sheet for the story will require a bit more interaction for the viewer. As noted in [Chapter 4, Contributing to Data Discovery](#), this is achieved by embedding the **Food Expenses** sheet directly into the story, as shown in the following screenshot. This will allow the author to not only narrate the findings but also invite the viewer to explore these findings and others within the application:



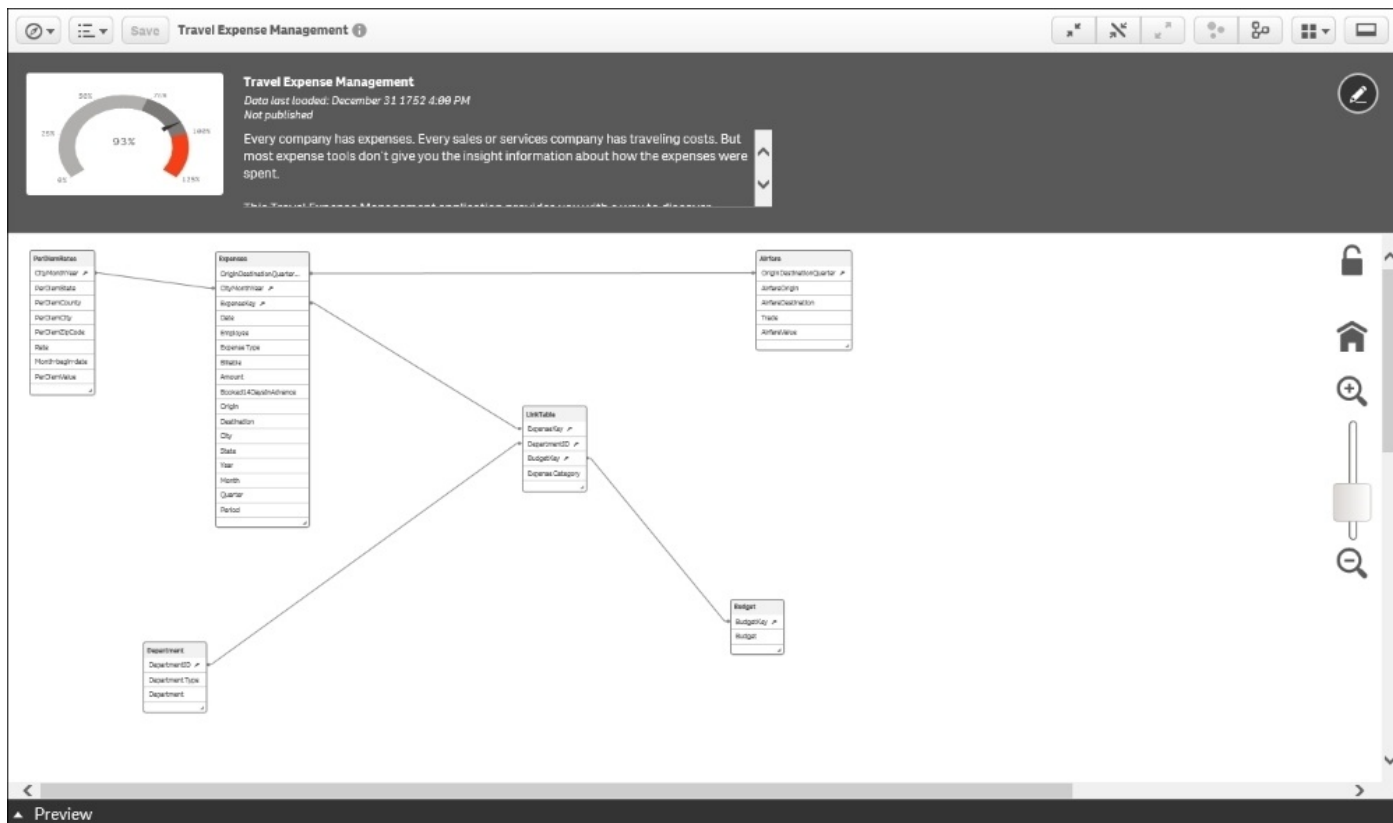
With the story created, the final annotation makes the following recommendation: **As a company, we should consider increasing the budget for meal expenses. Most of our employees expensed less than the per diem rate in 2013 and were significantly above budget in their meal expenses so increasing the budget seems like a good idea.**

Now that we have covered the application features, let's turn our attention to how it was built.



# Developing the application

Let's start our review of the heart of a Qlik Sense application, the data model. As you can see from the following screenshot, there are six tables in the Travel Expense Management associative model. At the heart of this application is the Expenses table. These tables were created through **Data Load Editor**, which was covered in [Chapter 5, Authoring Engaging Applications](#). It is worth noting that Qlik and Qlik partners provide both general-purpose connectors and specialized connectors to access a broad array of data sources.



# Examining the key tables

Let's examine the key tables.

## Expenses

The Expenses table contains all the key information (shown in the following screenshot) about the expense transaction of an employee. This includes information such as the date, employee name, expense type, and so on.

Expenses		Preview of data					
Rows	10527	CityMonthYear	OriginDestinationQuarter	ExpenseKey	Date	Employee	Expense Type
Fields	17	Las Vegas_Oct_2013	_Q4	Food_101_10/1/2013	10/17/2013	Angela Betterton	Breakfast
Keys	3	Las Vegas_Oct_2013	_Q4	Food_101_10/1/2013	10/17/2013	Angela Betterton	Dinner
Tags:	\$key \$ascii \$text \$numeric \$integer \$timestamp \$date	Las Vegas_Oct_2013	_Q4	Food_101_10/1/2013	10/17/2013	Angela Betterton	Dinner
		Las Vegas_Oct_2013	_Q4	Food_101_10/1/2013	10/17/2013	Angela Betterton	Dinner
		Philadelphia_Nov_2013	_Q4	Food_101_11/1/2013	11/7/2013	Angela Betterton	Breakfast
		Philadelphia_Nov_2013	_Q4	Food_101_11/1/2013	11/7/2013	Angela Betterton	Breakfast

*The Expenses table*

## PerDiemRates

The PerDiemRates table contains all the key information (shown in the following screenshot) about state, city, month, rates, and so on:

PerDiemRates		Preview of data				
Rows	1994533	CityMonthYear	PerDiemState	PerDiemCounty	PerDiemCity	PerDiemZipC
Fields	8	Riverhead / Ronkonkoma / Melville_Oct_2011	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005
Keys	1	Riverhead / Ronkonkoma / Melville_Nov_2011	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005
Tags:	\$key \$ascii \$text \$numeric \$integer \$timestamp \$date	Riverhead / Ronkonkoma / Melville_Dec_2011	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005
		Riverhead / Ronkonkoma / Melville_Jan_2012	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005
		Riverhead / Ronkonkoma / Melville_Feb_2012	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005
		Riverhead / Ronkonkoma / Melville_Mar_2012	New York	Suffolk	Riverhead / Ronkonkoma / Melville	005

*The PerDiemRates table*

## Airfare

The Airfare table contains all the key information (shown in the following screenshot) about the origin, destination, airfare value, and so on:



Airfare		Preview of data				
Rows	131634	<b>OriginDestinationQuarter</b>	<b>AirfareOrigin</b>	<b>AirfareDestination</b>	<b>Trade</b>	<b>AirfareValue</b>
Fields	5	Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q1	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	236.2706977
Keys	1	Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q2	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	273.1373296
Tags:	\$key \$ascil \$text \$numeric	Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q3	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	327.5836916
		Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q4	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	323.2155909
		Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q1	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	337.0201818
		Allentown/Bethlehem Area_Dallas/Fort Worth, TX_Q2	Allentown/Bethlehem Area	Dallas/Fort Worth, TX	Average Fare	318.482765

*The Airfare table*

## Department

The Department table contains all the key information (shown in the following screenshot) about the department ID, type, and department name:

Department		Preview of data		
Rows	22	<b>DepartmentID</b>	<b>Department Type</b>	<b>Department</b>
Fields	3	101	C-level	C-level
Keys	1	102	Finance	Finance
Tags:	\$key \$numeric \$integer \$ascil \$text	104	HR	Human Resources
		106	Marketing	Marketing - Corporate
		107	Sales	Support
		108	Marketing	Marketing - Field

*The Department table*

## Budget

The Budget table contains all the key information (shown in the following screenshot) about the budgeted amount using a compound key value that includes the expense type, department ID, and date:

Budget		Preview of data	
Rows	504	<b>BudgetKey</b>	<b>Budget</b>
Fields	2	Food_101_1/1/2013	18.1
Keys	1	Food_101_2/1/2013	16.4
Tags:	\$key \$ascii \$text \$numeric	Food_101_3/1/2013	27.4
		Food_101_4/1/2013	15.6
		Food_101_5/1/2013	31.7
		Food_101_6/1/2013	38.5

*The Budget table*

## LinkTable

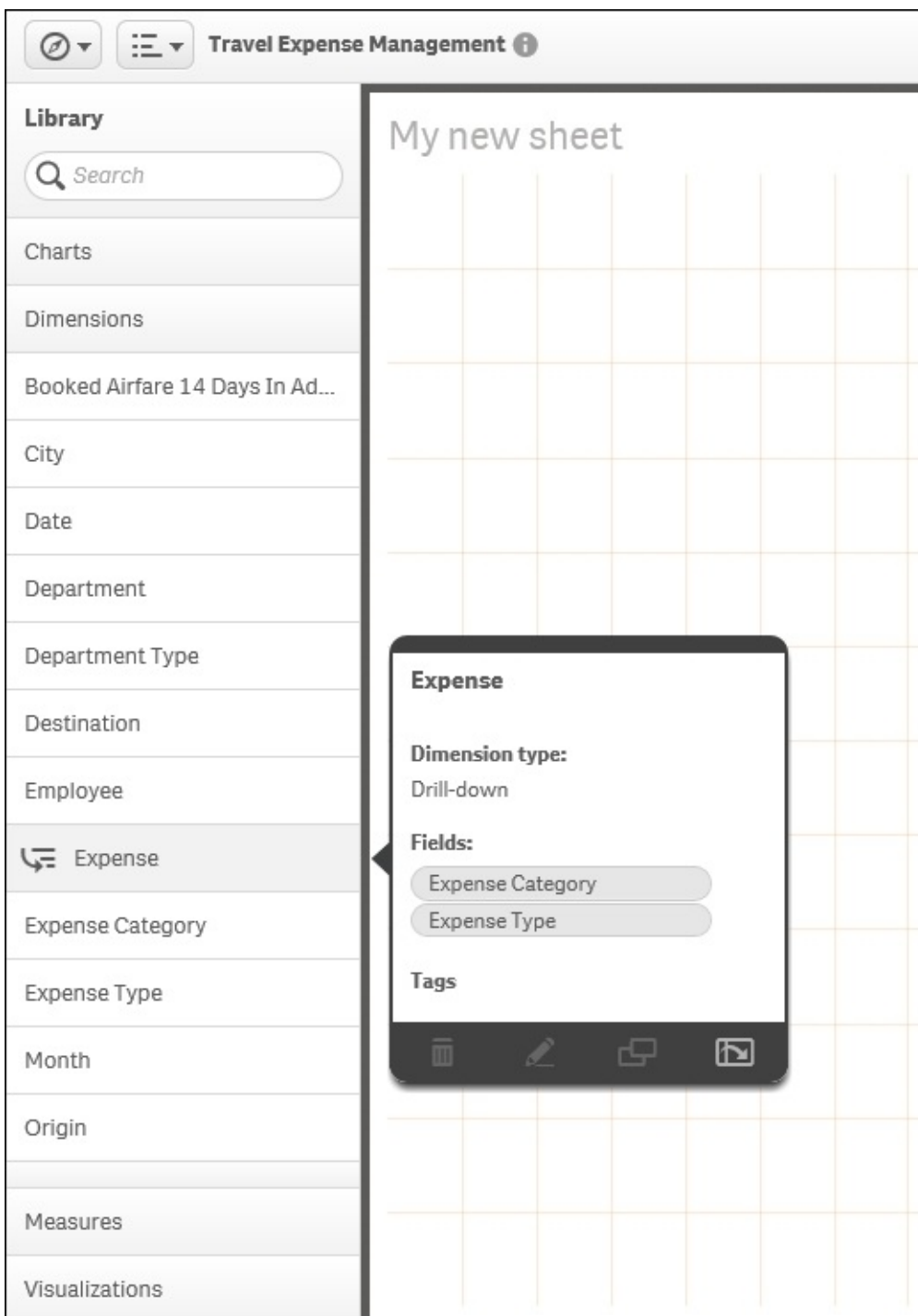
LinkTable contains all the keys (shown in the following screenshot) to link the expense, department, and budget tables:

LinkTable		Preview of data			
Rows	504	<b>ExpenseKey</b>	<b>DepartmentID</b>	<b>BudgetKey</b>	<b>Expense Category</b>
Fields	4	Travel_401_1/1/2013	401	Travel_401_1/1/2013	Travel
Keys	3	Food_401_1/1/2013	401	Food_401_1/1/2013	Food
Tags:	\$key \$ascii \$text \$numeric \$integer	Travel_301_1/1/2013	301	Travel_301_1/1/2013	Travel
		Food_301_1/1/2013	301	Food_301_1/1/2013	Food
		Travel_406_1/1/2013	406	Travel_406_1/1/2013	Travel
		Travel_106_1/1/2013	106	Travel_106_1/1/2013	Travel

*LinkTable*

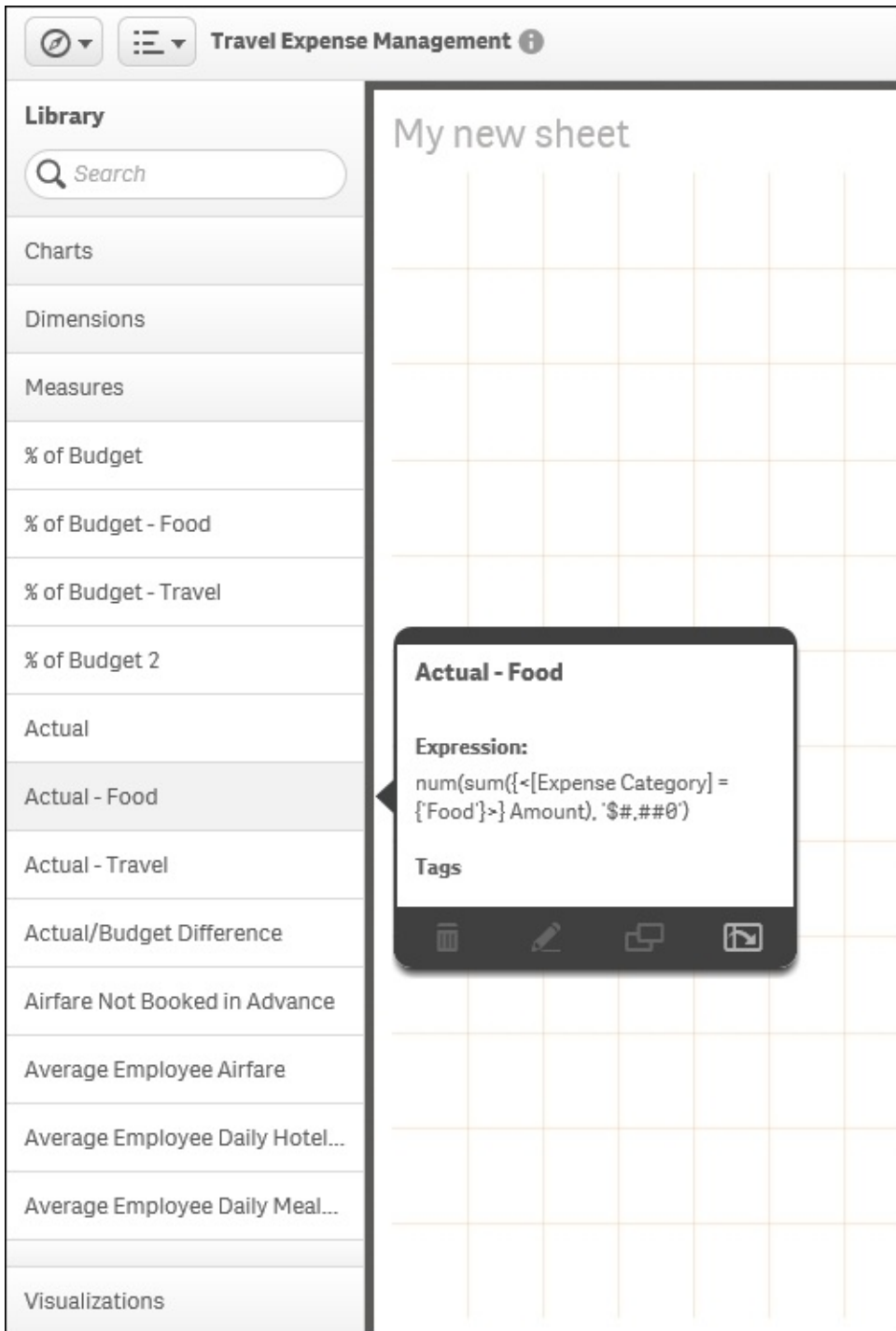
# Dimensions

Now, let's turn our attention to what has been exposed in Travel Expense Library by the developer to facilitate the creation and sharing of personal sheets. In the following screenshot, we can see the dimensions that were created. One particular dimension that is worth calling out is the **Expense** dimension, which provides a drill navigation from **ExpenseCategory** to **ExpenseType**. This capability usually requires extensive modeling or complex scripts in other BI software products, but with Qlik Sense, this is a simple selection process when creating the dimension. This is another example of the power of Qlik's associative indexing engine in action, but this time, easing the development of navigation within the application.



## Measures

The next area to cover is **Measures**. These are calculated expressions that most often form the KPIs within an application. In the following screenshot, we can see a list of measures that are used and exposed to contributors to allow them to create private sheets. Note that hovering the pointer over any of these objects makes a preview popup appear to provide additional context. In this case, you can see how the measure **Actual - Food** is calculated.



Additionally, the following table contains the measure definitions that directly tie to the KPIs used in this application. Refer to the Qlik Sense online help for additional information on the Qlik Sense function, which is available at <https://help.qlik.com>.

The measure expressions include:

Measure	Calculation
<b>% of Budget - Travel</b>	$\text{num}(\text{sum}(\{<[\text{Expense Category}] = \{\text{'Travel'}\}>\} \text{Amount}) / \text{sum}(\{<[\text{Expense Category}] = \{\text{'Travel'}\}>\} \text{Budget}), \text{'\#,##0\%'})$
<b>% of Budget 2</b>	$\text{num}((\text{Sum}(\text{Budget})/\text{sum}(\text{Amount}))-1, \text{'\#,##0\%'})\text{Amount}]$
<b>Actual</b>	$\text{Num}(\text{Sum}(\text{Amount}), \text{'\$,##0'})$
<b>Actual - Food</b>	$\text{num}(\text{sum}(\{<[\text{Expense Category}] = \{\text{'Food'}\}>\} \text{Amount}), \text{'\$,##0'})$
<b>Actual - Travel</b>	$\text{num}(\text{sum}(\{<[\text{Expense Category}] = \{\text{'Travel'}\}>\} \text{Amount}), \text{'\$,##0'})$
<b>Actual/Budget Difference</b>	$\text{Num}(\text{Sum}(\text{Budget})-\text{Sum}(\text{Amount}), \text{'\$,##0'})$
<b>Airfare Not Booked in Advance</b>	$\text{Num}(\text{Avg}(\{<[\text{Expense Type}] = \{\text{'Airfare'}\}, \text{Booked14DaysInAdvance} = \{\text{'No'}\}>\} \text{Amount}), \text{'\$,##0.00'})$
<b>Average Employee Airfare</b>	$\text{Avg}(\{<[\text{Expense Type}] = \{\text{'Airfare'}\}>\} \text{Amount})$
<b>Average Employee Daily Hotel Cost</b>	$\text{Sum}(\{<[\text{Expense Type}] = \{\text{'Hotel'}\}>\} \text{Amount})/\text{Count}(\text{DISTINCT Employee})$
<b>Average Employee Daily Meal Cost</b>	$\text{Sum}(\{<[\text{Expense Type}] = \{\text{'Breakfast'}, \text{'Lunch'}, \text{'Dinner'}\}>\} \text{Amount})/\text{Count}(\text{DISTINCT Employee})$
<b>Average US Airfare (External Source)</b>	$\text{Avg}(\{<\text{Trade} = \{\text{'Average Fare'}\}>\} \text{AirfareValue})$
<b>Average US Lodging Rates (External Source)</b>	$\text{Avg}(\{<\text{Rate} = \{\text{'Lodging'}\}>\} \text{PerDiemValue})$
<b>Average US Meal Rates (External Source)</b>	$\text{Avg}(\{<\text{Rate} = \{\text{'M\&IE'}\}>\} \text{PerDiemValue})$
<b>Avg Airfare - Largest Carrier (External Source)</b>	$\text{Avg}(\{<\text{Trade} = \{\text{'Average Fare - Largest Carrier'}\}>\} \text{AirfareValue})$
<b>Avg Airfare - Low Fare Carrier (External Source)</b>	$\text{Avg}(\{<\text{Trade} = \{\text{'Average Fare - Low Fare Carrier'}\}>\} \text{AirfareValue})$
<b>Booked Airfare in Advance</b>	$\text{Num}(\text{Avg}(\{<[\text{Expense Type}] = \{\text{'Airfare'}\}, \text{Booked14DaysInAdvance} = \{\text{'Yes'}\}>\} \text{Amount}), \text{'\$,##0.00'})$

<b>Booked Difference</b>	Num(Avg({<[Expense Type]='Airfare'}, Booked14DaysInAdvance={'Yes'}>} Amount)-Avg({<[Expense Type]='Airfare'}, Booked14DaysInAdvance={'No'}>} Amount), '\$#,##0.00')
<b>Booked Difference %</b>	Num((Avg({<[Expense Type]='Airfare'}, Booked14DaysInAdvance={'Yes'}>} Amount)/Avg({<[Expense Type]='Airfare'}, Booked14DaysInAdvance={'No'}>} Amount))-1, '#,##0.00%')
<b>Budget</b>	Num(Sum(Budget), '\$#,##0')
<b>Budget - Food</b>	num(sum({<[Expense Category] = 'Food'}>} Budget), '\$#,##0')
<b>Budget - Travel</b>	num(sum({<[Expense Category] = 'Travel'}>} Budget), '\$#,##0')
<b>Food Difference</b>	num(sum({<[Expense Category] = 'Food'}>} Budget)-sum({<[Expense Category] = 'Food'}>} Amount), '\$#,##0')
<b>Travel Difference</b>	num(sum({<[Expense Category] = 'Travel'}>} Budget)-sum({<[Expense Category] = 'Travel'}>} Amount), '\$#,##0')

## Visualizations

The last category of objects in the **Library (Master items)** is **Visualizations**. These are preformed visualizations that are typically the most popular or requested. They are defined to help facilitate a user's analysis and can be easily dragged and dropped onto a private sheet. In the following screenshot, we see a horizontal bar chart that analyzes the variance in **Booked Airfare in Advance vs Not in Advance**. Each of these visualizations contains predefined dimensions, measures, and chart definitions.

**Library**

Search

Charts

Dimensions

Measures

Visualizations

- Average Employee Daily...
- Average Employee Daily...
- Average Employee Daily...
- Average Employee Daily...
- Booked Airfare in Advan...**

### My new sheet

Category	Percentage Change
Category 1	-18.39%
Category 2	-8.37%
Category 3	9.95%
Category 4	2.00%

Tags

Navigation icons: delete, edit, copy, share

*The Travel Expense visualizations*





# Summary

In summary, Qlik Sense provides unique capabilities to meet the challenging task of analyzing and managing travel expenses. Without the capabilities offered by Qlik, this task can be difficult due to the size of the data and the many perspectives that can be taken in trying to understand airline purchasing, meal expense habits, and the impact on meeting corporate budget requirements. Qlik's associative indexing engine powers this exploration and means that meeting these requirements is no longer challenging at all.

In the next chapter, we will explore how Qlik Sense meets the needs of demographic data discovery.



# Chapter 13. Demographic Data Discovery

In this final chapter, we shall finish our exploration of real data with Qlik Sense by moving beyond the standard structures of the office and showing the full possibilities of the software for analysis of almost any kind of imaginable data. We'll therefore be looking at applying Qlik Sense to demographic data. As before, this example and many others are available for you to explore at <http://sense-demo.qlik.com>.

This chapter will cover the aspects necessary for demographic data discovery, including:

- General information about common KPIs
- Examples showing how to use the lasso selection in maps and scatter charts
- Examples of dimensions and measures

# Problem analysis

With Qlik Sense, it is possible to analyze not only business data, but rather *any* data. One great example is demographic data—statistics of countries and regions on anything from age and gender to income and life expectancy.

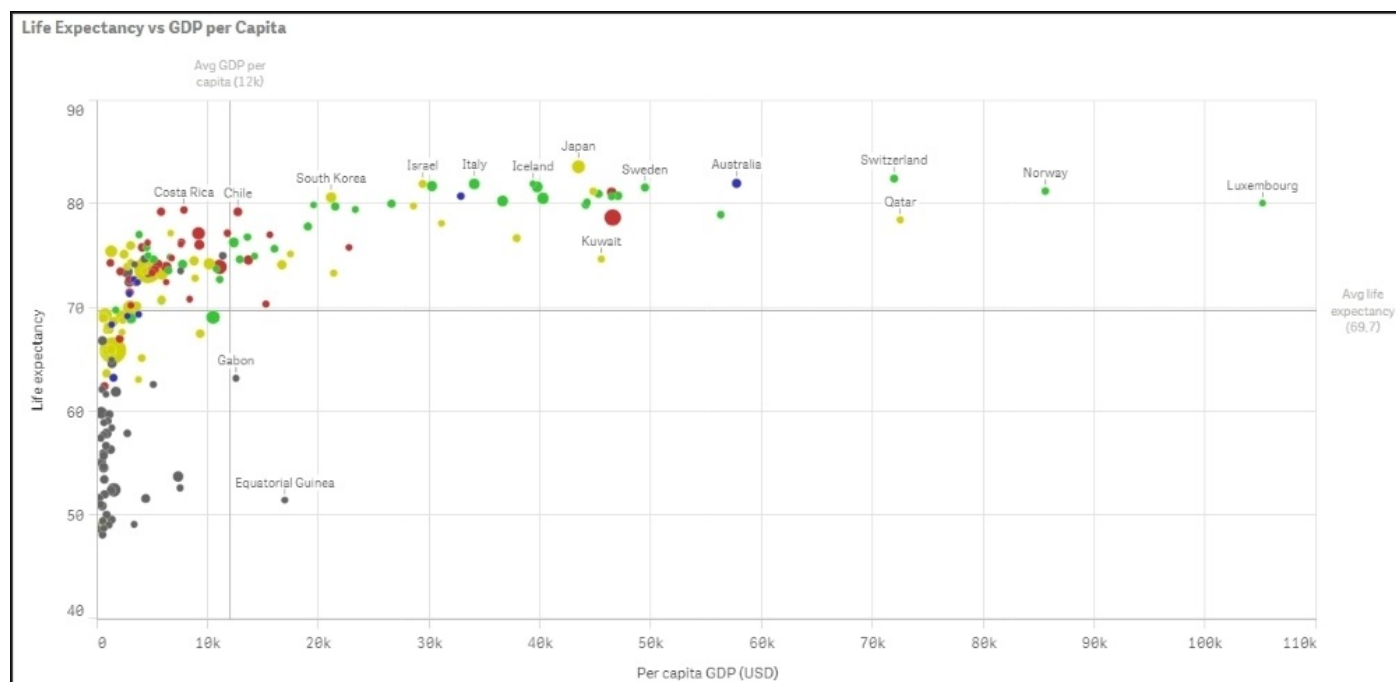
Such data can be found on a number of Internet sites and downloaded for your convenience, for example, from the following websites:

- United Nations ([data.un.org](http://data.un.org))
- Federal government of the United States ([www.data.gov](http://www.data.gov))
- European Union ([ec.europa.eu/eurostat](http://ec.europa.eu/eurostat))
- Qlik DataMarket ([www.qlik.com/us/explore/products/qlik-datamarket](http://www.qlik.com/us/explore/products/qlik-datamarket))

Demographic data is used and analyzed as-is by a number of nongovernmental organizations that need it for their activities. The common measures required are GDP per capita, population, unemployment rate, inflation, life expectancy, happiness, trade balance, labor cost, national debt, election results, and so on.

Often, interesting questions about correlations are asked; for example, how does happiness correlate with material standards and health? How are population growth and the number of children affected by factors such as life expectancy, poverty, and average salary? How has life expectancy improved over the years? If you haven't seen Hans Rosling's presentations on the Internet on this topic, we strongly recommend them. They show that data analysis is both important and fun.

Common dimensions in demographic data are country, region, gender, age group, ethnicity, and so on. An example can be seen in the following scatter chart, where you can see life expectancy and per capita GDP for different countries. Many developing countries are found in the lower-left quadrant, whereas the richer countries usually are found in the upper-right quadrant.



## *Life expectancy versus per capita GDP*

You can clearly see that the two numbers are highly correlated—the higher the GDP, the higher the life expectancy.

These measures can often also be linked to your business data to enable a deeper understanding. For instance, you can divide your country sales by the population of the country, thereby getting a relative sales number, which tells you how well you sell in that country. With this number, you can make relevant comparisons of countries of different sizes.

Alternatively, if you assume that the market space in the country is roughly proportional to the GDP, you can divide your sales by the GDP and use this number to compare market penetration between countries.

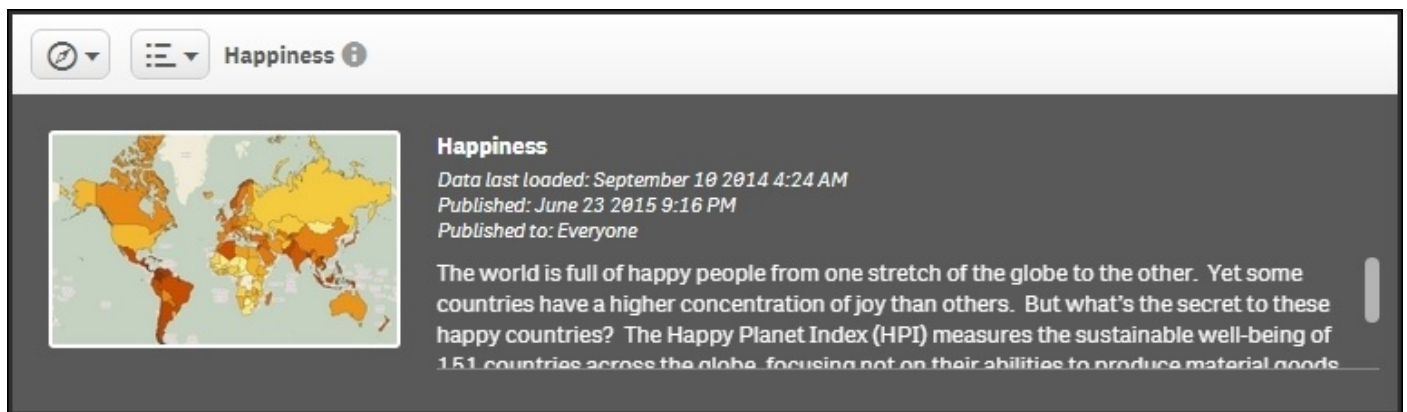
These numbers will answer questions such as, “How well are we selling in this country, given the potential?”



# Application features

On our demo site, we have an app with a number of demographic measures per country. You can find it at <http://sense-demo.qlik.com> under the name **Happiness**. It analyzes, among other demographic indexes, the **Happy Planet Index (HPI)** in a number of countries. You can learn more about this index at [www.happyplanetindex.org](http://www.happyplanetindex.org).

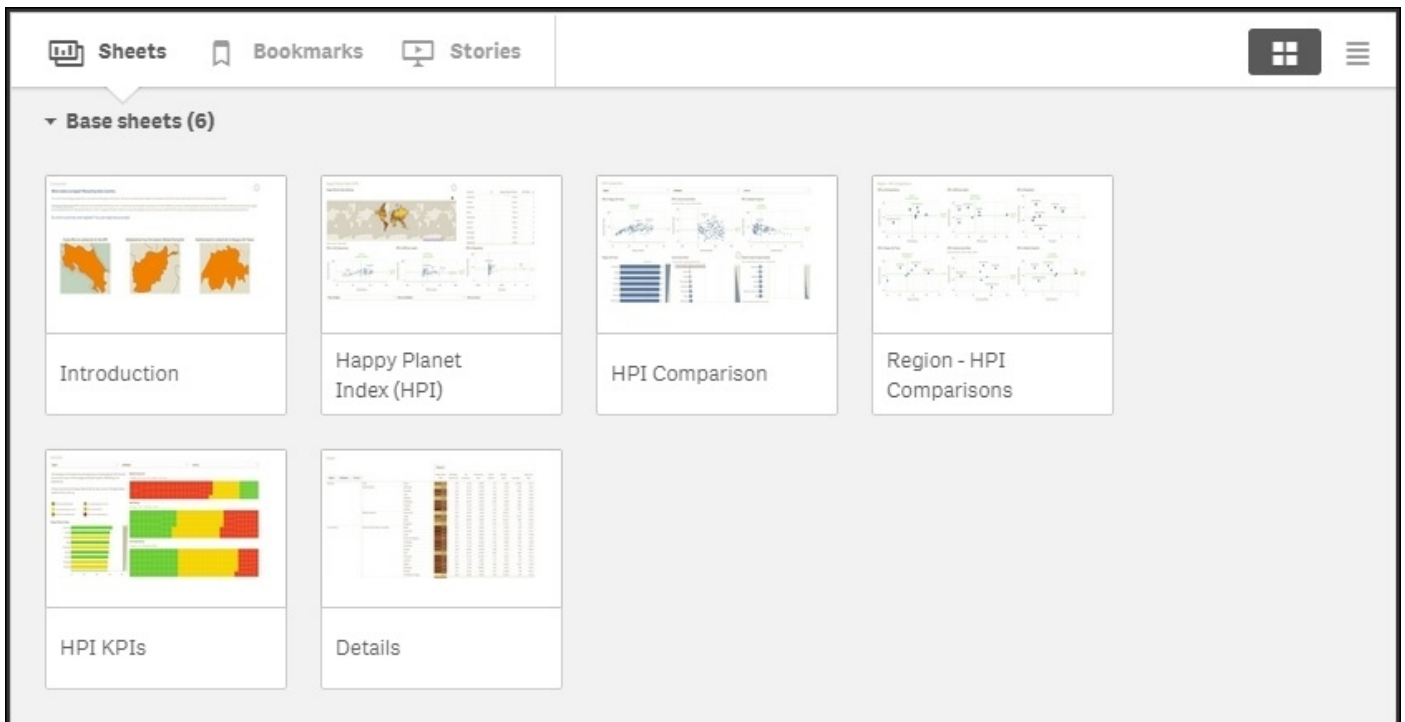
This index measures the sustainable well-being of 151 countries across the globe, focusing not on their abilities to produce material goods and services, but rather on their abilities to produce long, happy, and sustainable lives for the people who live in them. A happy life doesn't have to come at the expense of our environment, and the HPI is used to promote a policy that puts the well-being of people and the planet first.



*The app overview of the Happiness application*

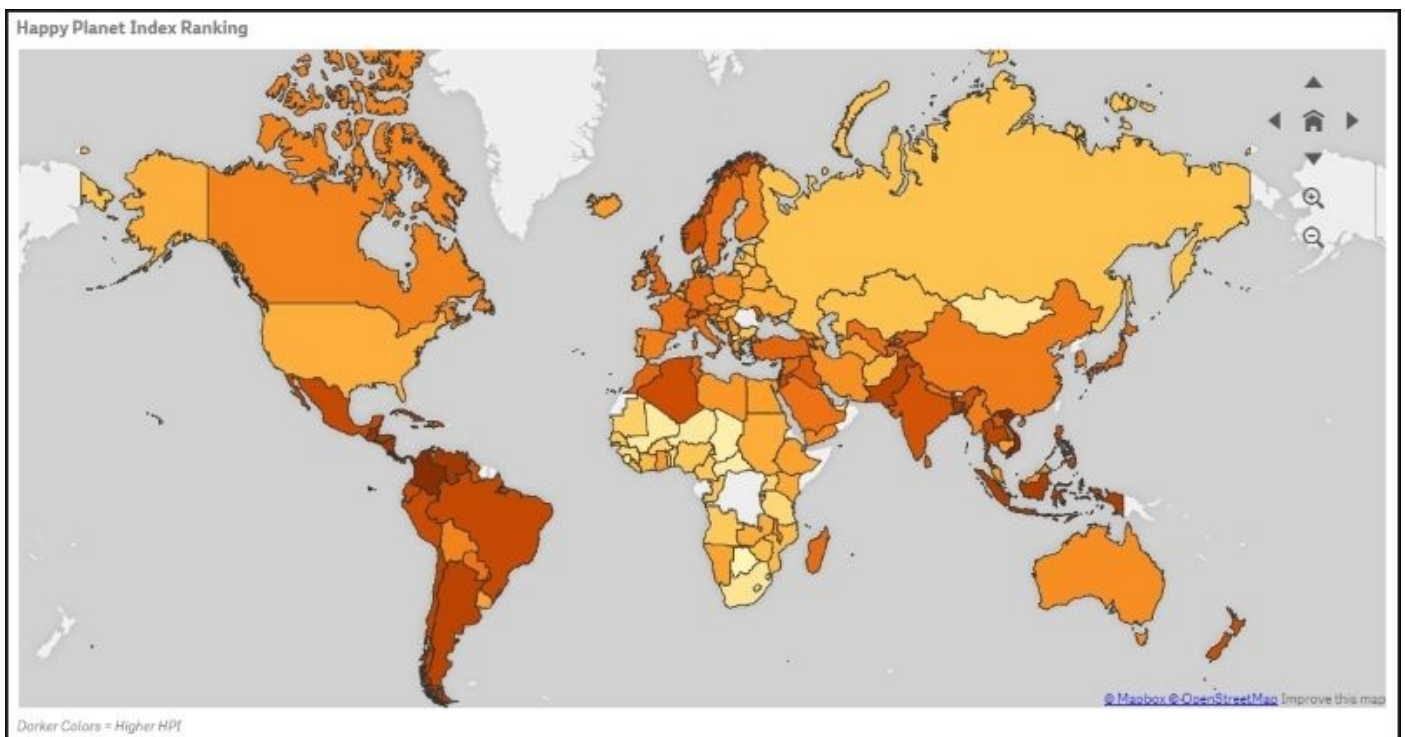
Below this overview, you will see a number of sheets. The leftmost sheet is an introduction, whereas the other sheets are prepared for analysis and detailed information.

If you click on the **Stories** button to the left, you will see that the app also contains one story—a story that can be used to present data in the app. It can also be used as an introduction to the app the first time you open it.



*The sheets on the app overview page*

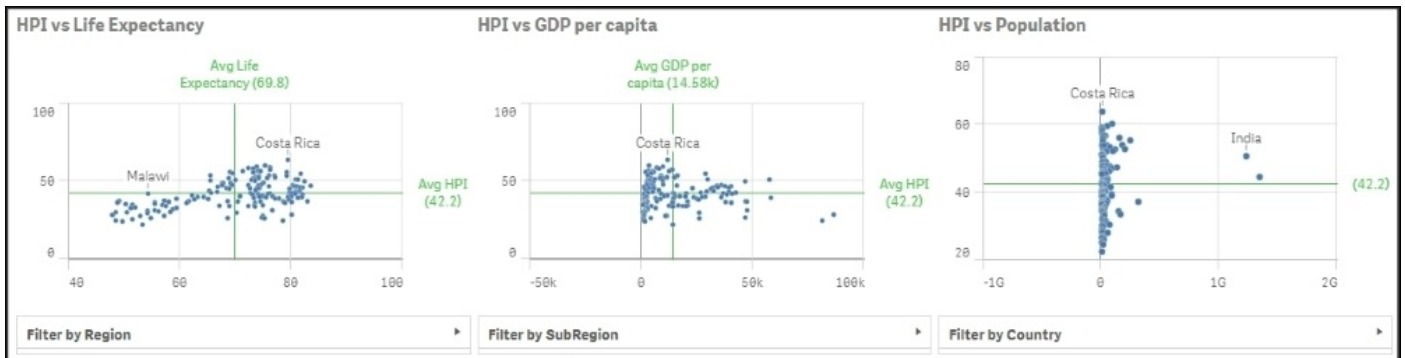
The first sheet is an introduction sheet that explains what the app is all about. The second sheet, which is the first one with traditional charts, is called **Happy Planet Index (HPI)**. On it, you will see the happiness index for all countries, first on a map, and then in a table. The countries in the map are colored according to the happiness index—the darker the color, the higher the happiness index.



*A map showing the happiness index per country*



Below the map, there are three scatter charts showing the happiness index per country, plotted against the life expectancy, GDP per capita, and total population. These three charts are excellent tools to analyze any correlation between happiness and the mentioned demographic measures.



*Scatter charts that show the correlation (or lack of correlation) between happiness and other demographic measures*

Finally, at the bottom, you have three filter panes, allowing the user to choose only a region, subregion, or country to zoom in the numbers for a specific area.

The other sheets contain additional and more detailed information, ordered by topics. The final sheet contains a table showing the details, should the user be interested in drilling down to the lowest level.

# Analysis

When looking at data in this app, the first question that pops up in the user's mind is usually, "Is there any correlation between happiness and x?" To get a qualitative answer to this, you only need to browse through the scatter charts.

On the **Happy Planet Index (HPI)** sheet, you have three scatter charts. In the leftmost chart, **HPI vs Life Expectancy**, you can see a correlation between the two measures, at least for lower life expectancies. In the other two charts, however, there is no clear correlation.

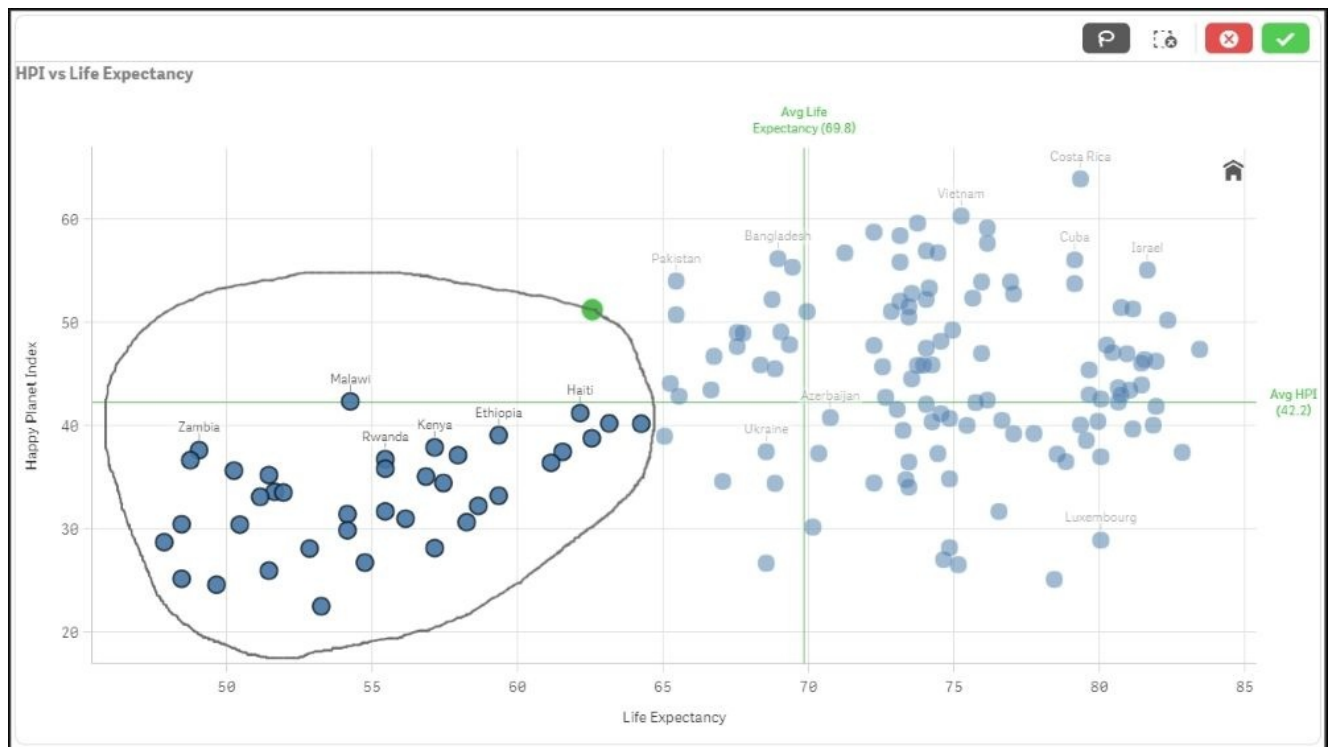
On the **HPI Comparison** sheet, you have three additional scatter charts. In the leftmost chart, **HPI vs Happy Life Years**, you can see a weak correlation between the two measures. The same is true for the rightmost chart, **HPI vs Global Footprint**, but in the chart in the middle (**HPI vs Governance**), there is no clear correlation.

However, as in all statistics, you have to be careful with your conclusions. Firstly, correlation does not imply causation. You have to look at many factors and use common sense to find the true cause and effect. In this case, it is just that the happiness index is an artificial index calculated from the life expectancy and ecological footprint among others, hence the correlation with happy life years and global footprint.

# Using the lasso selector to make selections

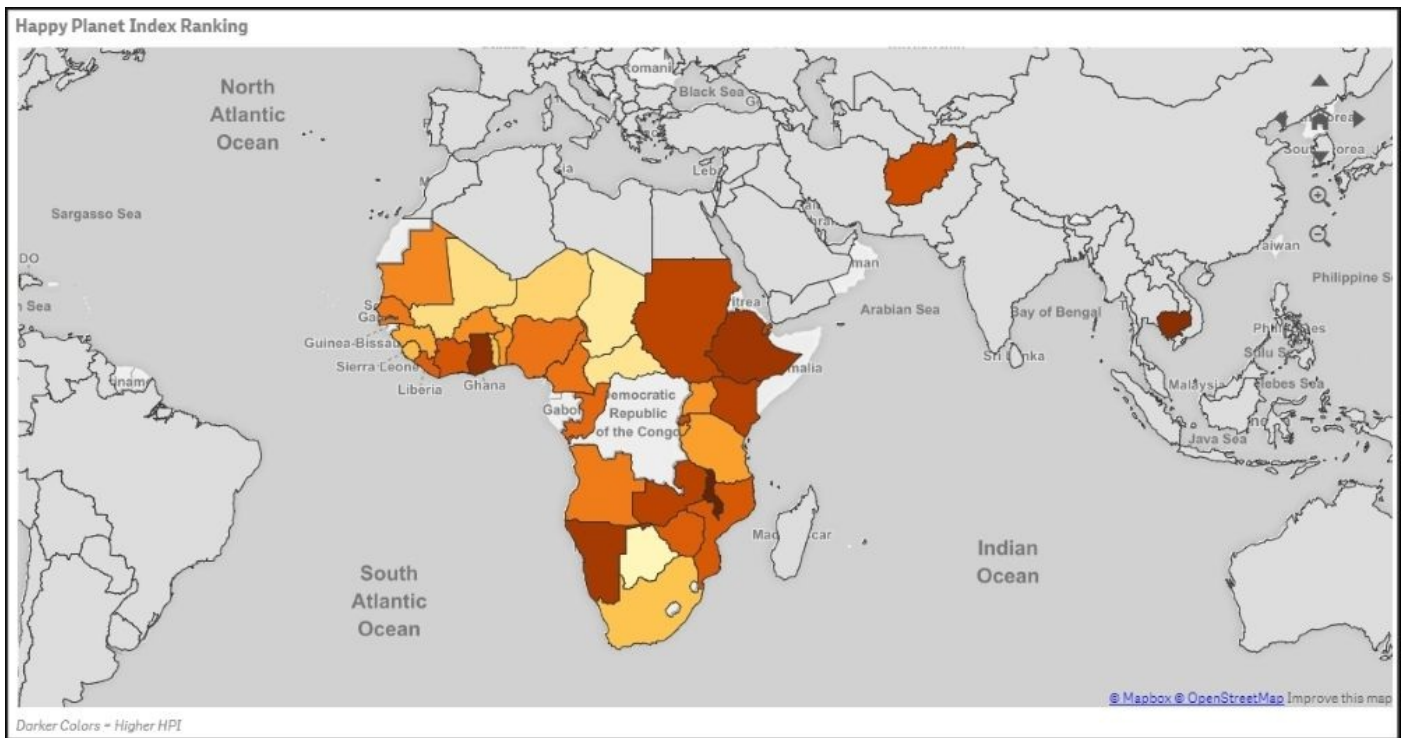
Now, let's explore the data. One question could be, "Where in the world do we find the countries with a low average life expectancy?" To answer this, you need to make a selection in the scatter chart showing life expectancy:

1. First, navigate to the **Happy Planet Index** sheet. Maximize the scatter chart that shows **HPI vs Life Expectancy** by clicking on the fullscreen arrow in the upper-right corner of the object.
2. Then, click on the chart so that the chart controls, including the lasso symbol, appear in the upper-right corner.
3. Next, click on the **Turn on lasso selection** option. Now you can draw a line around the points you want to select.
4. Finally, confirm your selection by clicking on the green tick mark in the upper-right corner.



*Lasso selection in the scatter chart*

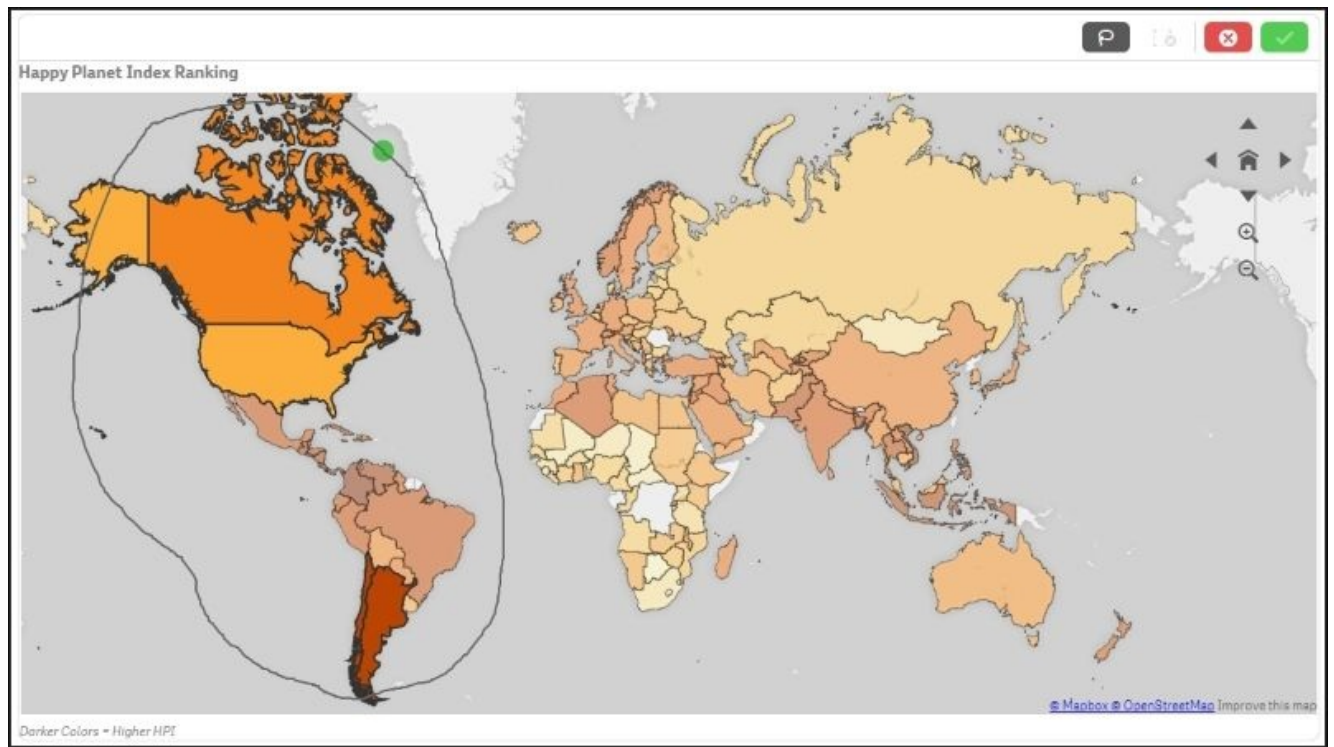
If you now look at the map, you will see where these countries appear in the world. It's predominantly Africa and South Asia. If you click on the map, you can zoom in using the scroll wheel of the mouse. You can also pan the map.



### *Countries with low life expectancy*

Of course, you can also make a selection the other way round. Use the lasso selector in the map and see how the selected countries are distributed in the scatter chart. The way to do this is as follows:

1. Maximize the map.
2. Click somewhere in the map.
3. Click on the **Turn on lasso selection** option and encircle the part of the world you want to explore.
4. Finally, confirm your selection.



*Making a lasso selection of America on the map*

# Using the global selector to make selections

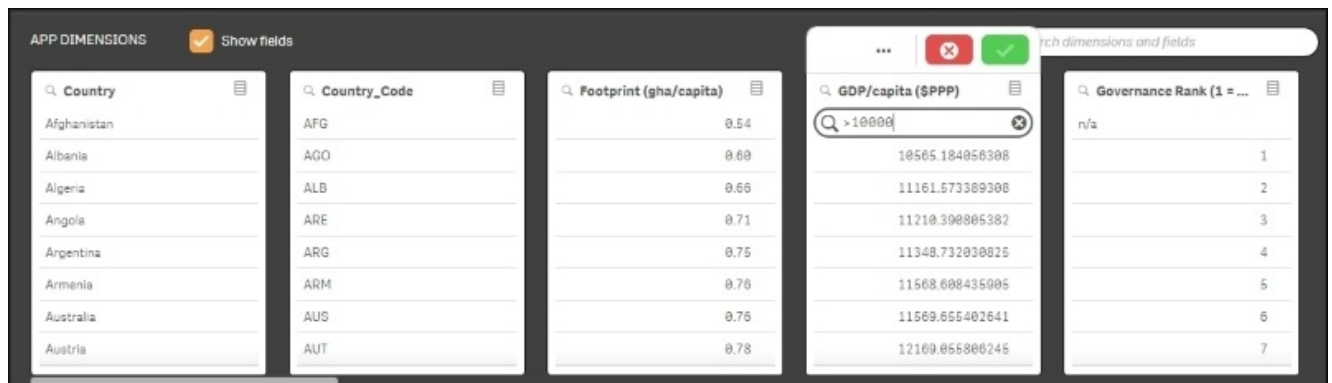
You can also use the global selector to make selections. Just click on the global selector and make selections directly in the fields.

For instance, you may have a question like this, “Where in the world do I find the richest countries?” In such a case, perform the following steps:

1. Open the global selector. (This is found to the right in the toolbar with **Selections tool** as a popup.)
2. Find a field called **GDP/capita (\$PPP)**. To do this, you might first need to check **Show fields** in the global selector.
3. Once you have found this field, you can investigate it just by scrolling. You will then see that there are some countries with less than \$400 in GDP per capita, while the richest countries have more than \$80,000 in GDP per capita.

If you want to find the countries where the GDP is greater than \$10,000, perform the following steps:

1. Click on the search icon and type >10000.
2. Confirm the search by pressing *Enter*, and then confirm the selection by clicking on the green tick mark.



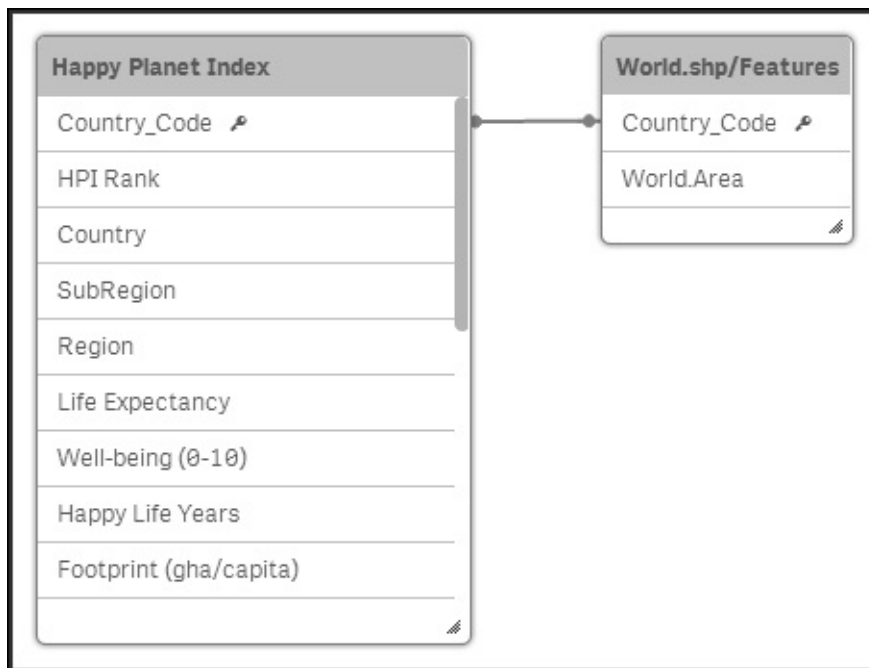
*Selecting the countries in the world with the highest GDP*

If you now close the global selector and go back to the map and the scatter charts, you will be able to see where you find the richest countries, both on the map and in the scatter charts.



# How the application was developed

The data model of the **Happiness** application is not very complicated:



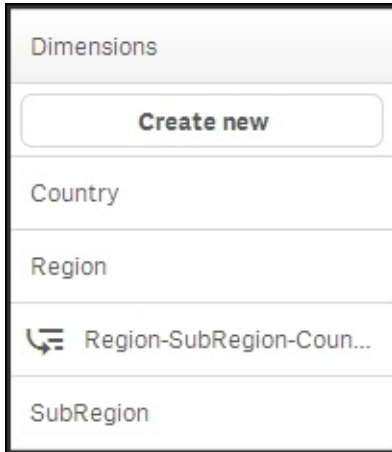
This is an extremely simple data model that only contains one table of real data, Happy Planet Index, and an additional table listing all countries, world.shp/Features. The second table has one record per country and holds the map information—the shapes of the country—used in the map object in the user interface.

In this app, the data table has exactly one record per country—a record that contains the relevant information for a given country at a given moment. However, this is not always the situation. More often, the data table contains data for countries over many points in time, for example, one record per combination of a country and a year. This will result in several lines per country.



# Dimensions

There are not many fields that can be used as dimensions. The three available fields are region, subregion, and country. The world is split into 7 regions and 19 subregions. A country can only belong to one subregion and one region. These fields have been added to **Library**. In addition, a drill-down dimension has been created from the three fields.



*The dimensions in Library*

One way of adding dimensions could be by creating buckets based on one of the measures, for example, population. Countries could then be grouped under **Large**, **Medium**, and **Small** classes, which will be stored in a new field, **Population Class**.

# Measures

A number of measures have also been defined, for example, GDP, happiness index, global footprint, life expectancy, and so on.

It is important that the app developer formulates the formulas correctly, since this is something that could be difficult for the business user. The business user doesn't always have knowledge about the data model, which is something you need in order to get all the expressions right.

In the following table, you can find some of the measures defined in this app:

Measure	Definition
<b>GDP per Capita</b>	Avg([GDP/capita])
<b>Global Footprint</b>	Avg([Footprint])
<b>Governance Rank</b>	Only[Governance Rank])
<b>Happy Life Years</b>	Only([Happy Life Years])
<b>Happy Planet Index</b>	Only([Happy Planet Index])
<b>HPI Rank</b>	Only([HPI Rank])
<b>Population</b>	Only(Population)

Several of these measures can be defined differently. How you do this is very much a matter of taste. For instance, the measures where the `Only()` function is used can also be defined using `Sum()` or `Avg()`. As long as you only have a single number, all three functions will return the same answer.

But how do you want Qlik Sense to behave when there are several countries, for example, a region that should be represented by one value? For the **Population** measure, the obvious function to use should be `Sum()`. Then the total population of the region will be shown.

But if the source data contains several years, so that a single country has several records, you don't just want to sum the population. Then you would get numbers that are much larger than they should be. Instead, you might want to use `Sum(Population)/Count(distinct Year)` to create an average over all possible years.

In addition, for a rank, you wouldn't want to use `Sum()` because it would show an incorrect number. You could use `Avg()`, which will give the average rank between the countries. An average is clearly better, but it is still not mathematically correct. Then it might be better to use `Only()`, which doesn't return an answer at all when more than one country is involved.



# Summary

The analysis of demographic data is easy when you use Qlik Sense. Obviously, this analysis can also be made with a number of other tools, since the data model is very simple. However, with Qlik Sense, it is easy to build further. Qlik's associative indexing engine powers the analysis and ensures that you can develop or change your apps quickly and easily. With Qlik Sense, data discovery and analysis is made easy.

With the end of this chapter, we have also reached the end of the book. We took you from the history of Qlik to how to develop applications, and finally gave you some examples of how applications might look.

We hope that after reading this book, you have acquired some skills that will be useful when you develop your own Qlik Sense applications. We also think you now have a better understanding of the thoughts behind Qlik Sense, and wish you good luck in your endeavors.

Welcome to the community of Qlik users!

# Index

## A

- access pass / [Tokens](#)
- analysis interface, authoring engaging applications
  - about / [The analysis interface—sheets and visualizations](#)
  - sheet, creating / [Creating a sheet](#)
  - visualizations, adding / [Adding visualizations](#)
  - dimensions, adding / [Adding dimensions and measures](#)
  - measures, adding / [Adding dimensions and measures](#)
  - bar charts, defining / [Defining bar charts](#)
  - storytelling / [Storytelling](#)
- application library, authoring engaging applications
  - about / [The application library](#)
  - fields, to be exposed / [Which fields should be exposed?](#)
  - KPIs, defining / [Defining KPIs](#)
  - library entries, creating / [Creating library entries](#)
- application life cycle
  - overview / [Overview of an application's life cycle](#)
  - application authoring, starting / [Starting application authoring](#)
  - continuing / [Continuing the application's life cycle](#)
- applications
  - managing / [Management and monitoring](#)
  - importing / [Importing and managing apps](#)
  - extensions, importing / [Importing extensions](#)
  - users / [Users and user directories](#)
  - user directories / [Users and user directories](#)
  - streams, defining / [Defining streams](#)
  - connectivity management / [Connectivity management](#)
  - tasks / [Tasks](#)
  - system management / [System management](#)
  - security rules / [Security rules](#)
  - monitoring / [Monitoring](#)
- architecture, Qlik Sense®
  - about / [The Qlik Sense® architecture](#)
  - services / [Services](#)
  - clients / [Clients](#)
  - applications / [Applications](#)
  - nodes / [Nodes](#)
  - streams / [Streams](#)
- authentication
  - about / [Authentication and authorization](#)
- authoring engaging applications

- preparations / [Preparations and requirements](#)
- requisites / [Preparations and requirements](#)
- requirement specifications / [The requirement specifications](#)
- communication problem / [The communication problem](#)
- step-wise implementation / [A step-wise implementation](#)
- process / [The process](#)
- creating / [Getting started with the app creation](#), [Creating a new app](#)
- data, loading / [Loading your data](#)
- analysis interface / [The analysis interface—sheets and visualizations](#)
- application library / [The application library](#)
- publishing / [Publishing your apps](#)
- authorization
  - about / [Authentication and authorization](#)

# B

- bar chart
  - about / [The bar chart](#)
- best practices, data visualization
  - about / [Best practices in data visualization](#)
  - dashboard / [Dashboard](#)
  - analysis / [Analysis](#)
  - reporting / [Reporting](#)
  - sheet, structuring / [Structuring a sheet](#)
  - graphs / [Graphs and other visualizations](#)
  - dimensions / [Dimensions and measures](#)
  - measures / [Dimensions and measures](#)
- bookmarks
  - about / [Bookmarks](#)
- Business Discovery / [Continuing disruption](#)
- business intelligence (BI) / [Continuing disruption](#)
  - evolution / [Data discovery—the evolution of BI](#)
- business problem
  - about / [The business problem](#)

# C

- clients, Qlik Sense®
  - about / [Clients](#)
  - hub / [Clients](#)
  - Qlik Management Console (QMC) / [Clients](#)
- cloud
  - features / [Why use the cloud?](#)
  - content / [Cloud content](#)
- cloud sharing
  - about / [Cloud sharing](#)
- cluster
  - about / [Nodes](#)
- com-qliktech-peoplechart.js file / [Extending the Qlik Sense® client](#)
- com-qliktech-peoplechart.qext file / [Extending the Qlik Sense® client](#)
- components, applications
  - sheets / [Sheets](#)
  - bookmarks / [Bookmarks](#)
  - data storytelling / [Data storytelling](#)
- connectivity / [Connectivity management](#)
- content security
  - about / [Content security](#)



# D

- data, structuring
  - about / [Structuring your data](#)
  - normalization / [Normalization](#)
  - star schema / [Star schema and snowflake schema](#)
  - snowflake schema / [Star schema and snowflake schema](#)
- data discovery / [Continuing disruption](#)
  - about / [What is data discovery?](#), [Data discovery—the evolution of BI](#)
  - empowered user / [The empowered user](#)
  - data interaction / [Interaction with data](#)
  - traditional business intelligence architecture / [Traditional business intelligence architecture](#)
  - contributing to / [Realities of the data discovery contributor](#)
  - private bookmarks, creating / [Creating private bookmarks](#)
  - private sheets, creating / [Creating and sharing private sheets](#), [Creating a private sheet](#)
  - private sheets, sharing / [Creating and sharing private sheets](#)
  - private sheets, publishing / [Publishing a private sheet](#)
  - new sheet, creating / [Creating a new sheet](#)
- data discovery consumption requirements
  - about / [Data discovery consumption requirements](#)
  - hub / [Introducing the hub](#)
  - streams / [Introducing streams](#)
  - application components, exploring / [Exploring the components of the application](#)
  - associative experience, leveraging / [Navigating and leveraging the associative experience](#)
  - associative experience, navigating / [Navigating and leveraging the associative experience](#)
  - Library, extending with / [Extending with Library](#)
- data loading, authoring engaging applications
  - about / [Loading your data](#)
  - additional tables, loading / [Loading additional tables](#)
  - Data load editor, using / [Using the Data load editor](#)
  - database connection, creating / [Creating a database connection](#)
  - data connectors / [Data connectors](#)
- data model viewer
  - about / [The data model viewer](#)
  - preview mode, using / [Using preview mode](#)
- data storytelling
  - about / [Data storytelling](#)
- data visualization
  - best practices / [Best practices in data visualization](#)

- Demographic Data Discover application
  - problem analysis / [Problem analysis](#)
  - features / [Application features](#)
  - analysis / [Analysis](#)
  - lasso selector, for making selections / [Using the lasso selector to make selections](#)
  - global selector, for making selections / [Using the global selector to make selections](#)
  - developing / [How the application was developed](#)
  - dimensions / [Dimensions](#)
  - measures / [Measures](#)
- deployment, Qlik Sense®
  - about / [Deployment and licensing](#)
  - single node / [Single node or multinode](#)
  - multinode / [Single node or multinode](#)
  - license / [License and access passes](#)
  - access passes / [License and access passes](#)
  - tokens / [Tokens](#)
  - access rules / [Access rules](#)
- dimension / [Adding dimensions and measures](#)
- dimensions, Travel Expense Discovery application
  - about / [Dimensions](#)
  - measures / [Measures](#)
  - visualizations / [Visualizations](#)

# E

- Engine API Explorer
  - about / [Engine API Explorer](#)

# F

- fields, used as dimensions
  - nominals / [Dimensions and measures](#)
  - ordinals / [Dimensions and measures](#)
  - intervals / [Dimensions and measures](#)
  - ratios / [Dimensions and measures](#)

# G

- gauge object
  - about / [The gauge](#)
- geographical map
  - about / [The geographical map](#)
- Global Defined Data Sources / [What makes up a Qlik Sense® application?](#)
- graph types, data visualization
  - bar chart / [The bar chart](#)
  - pie chart / [The pie chart](#)
  - line chart / [The line chart](#)
  - KPI object / [The KPI object](#)
  - gauge object / [The gauge](#)
  - scatter chart / [The scatter chart](#)
  - tree map / [The tree map](#)
  - geographical map / [The geographical map](#)
  - tables / [Tables](#)
  - sorting / [Sorting and colors](#)
  - colors / [Sorting and colors](#)

# H

- hub / [Overview of an application's life cycle](#)
  - about / [Introducing the hub](#), [Clients](#)
- human resource app
  - recruitment / [The business problem](#)
  - employee satisfaction and retention / [The business problem](#)
  - training / [The business problem](#)
  - health and safety / [The business problem](#)
  - career and compensation / [The business problem](#)
  - features / [Application features](#)
  - sheet / [Sheets](#)
  - costs, training / [Training costs](#)
  - global selector, using / [Using the global selector](#)
  - developing / [How the application was developed](#)
  - Employees table / [How the application was developed](#)
  - Hierarchy table / [How the application was developed](#)
  - Survey table / [How the application was developed](#)
  - Training table / [How the application was developed](#)
  - Courses table / [How the application was developed](#)
  - Map shapes table / [How the application was developed](#)
  - dimensions / [Dimensions](#)
  - measures / [Dimensions](#)
- Human Resources analysis
  - about / [The business problem](#)
  - preparing / [The business problem](#)

# K

- key fields / [The process](#)
- key tables, Travel Expense Discovery application
  - Expenses table / [Expenses](#)
  - PerDiemsRates table / [PerDiemRates](#)
  - Airfare table / [Airfare](#)
  - Department table / [Department](#)
  - Budget table / [Budget](#)
  - LinkTable / [LinkTable](#)
- KPI object
  - about / [The KPI object](#)

# L

- Library / [What makes up a Qlik Sense® application?](#)
- License Monitor / [Monitoring](#)
- line chart
  - about / [The line chart](#)
- Load Script / [What makes up a Qlik Sense® application?](#)
- login access pass / [Tokens](#)



# M

- Mashup editor
  - about / [Web mashups](#)
  - pre-built templates / [Web mashups](#)
- measure / [Adding dimensions and measures](#)
- My Workspace / [Starting application authoring](#)

# N

- new sheet, data discovery
  - creating / [Creating a new sheet](#)
  - predefined visualization, adding to / [Adding a predefined visualization to a new sheet](#)
  - Combo chart object, creating / [Creating a Combo chart object](#)
  - publishing / [Publishing a private sheet](#)
- node
  - about / [Nodes](#)

# O

- Open Database Connectivity (ODBC) / [Creating a database connection](#)
- Operations Monitor / [Monitoring](#)

# P

- peoplechart-properties.js file / [Extending the Qlik Sense® client](#)
- peoplechart.css file / [Extending the Qlik Sense® client](#)
- pie chart
  - about / [The pie chart](#)

# Q

- QIX engine
  - about / [The QIX engine](#)
- Qlik Branch
  - about / [Developer community – Qlik Branch](#)
  - URL / [Developer community – Qlik Branch](#)
- Qlik Developer Hub
  - about / [Qlik® Dev Hub](#)
  - single configurator / [Qlik® Dev Hub](#)
  - extension editor / [Qlik® Dev Hub](#)
  - mashup editor / [Qlik® Dev Hub](#)
  - Engine API Explorer / [Qlik® Dev Hub](#)
  - web mashups / [Web mashups](#)
- Qlik Management Console (QMC)
  - about / [Clients](#)
- QlikMarket® data
  - using / [Using the Qlik DataMarket® content](#)
  - adding / [Adding the QlikMarket® data](#)
- Qlik Sense Management Console (QMC) / [Overview of an application's life cycle](#)
- Qlik Sense®
  - about / [Qlik Sense® and the QlikView.Next project](#)
  - defining / [Making sense of modern business](#)
  - architecture / [The Qlik Sense® architecture](#)
  - deployment / [Deployment and licensing](#)
  - licensing / [Deployment and licensing](#)
- Qlik Sense® application
  - components / [What makes up a Qlik Sense® application?](#)
  - sharing / [Sharing an application](#)
  - navigation / [Navigation](#)
  - smart visualizations / [Smart visualizations](#)
  - global search / [Global search](#)
  - global filtering / [Global filtering](#)
- Qlik Sense® apps, using in cloud
  - about / [Using Qlik Sense® apps in the cloud](#)
  - app, uploading from desktop / [Uploading an app from the desktop](#)
  - app, creating in Qlik Sense® Cloud / [Creating an app in Qlik Sense® Cloud](#)
  - app, sharing in Qlik Sense® Cloud / [Sharing an app in Qlik Sense® Cloud](#)
  - apps, maintaining / [Maintaining Qlik Sense® Cloud apps](#)
- Qlik Sense® client
  - extending / [Extending the Qlik Sense® client](#)
- Qlik Sense® Cloud
  - about / [Cloud sharing](#)
- Qlik Sense® data model

- about / [The Qlik Sense® data model](#)
- multitable data model, creating / [Creating a multitable data model](#)
- tables, linking / [Linking tables](#)
- pitfalls / [Pitfalls in the data model](#)
- QlikView.Next project
  - about / [Qlik Sense® and the QlikView.Next project](#)
  - themes / [Qlik Sense® and the QlikView.Next project](#)
- QlikView® applications
  - migrating, from Qlik Sense® / [Migrating applications from QlikView® to Qlik Sense®](#)
  - script changes / [Changes to the script](#)
  - user interface changes / [Changes to the user interface](#)
- Qlik® way
  - about / [The Qlik® way](#)
  - color coding / [Color coding](#)
  - freedom of data navigation / [Freedom of data navigation](#)
  - calculation on demand / [Calculation on demand](#)

# R

- requirement specifications, authoring engaging applications
  - data / [The requirement specifications](#)
  - KPIs / [The requirement specifications](#)
  - dimensions / [The requirement specifications](#)
  - security / [The requirement specifications](#)
- Responsive Web Design (RWD) / [Data discovery consumption requirements](#)

# S

- Sales Discovery application
  - business problem, analyzing / [The business problem](#)
  - features / [Application features](#)
  - application features / [Application features](#)
  - top customers, analyzing / [Who are our top customers?](#)
  - 360-degree customer view / [The 360-degree customer view](#)
  - customers, filtering / [Filtering customers](#)
  - shipments for top customers, reviewing / [Reviewing shipments for top customers](#)
  - bottom five customers, reviewing / [Reviewing the bottom five customers](#)
  - productive sales representatives, analyzing / [Who are our most productive sales representatives?](#)
  - products, analyzing / [Analyzing products](#)
  - customer sales, analyzing / [Analyzing customer sales](#)
  - building / [Building the application](#)
  - SalesDetails table / [The SalesDetails table](#)
  - Customers table / [The Customers table](#)
  - AggSales table / [The AggSales table](#)
  - US States ISO CODE 2 polygons table / [US States ISO CODE 2 polygons](#)
- Sales Discovery Library
  - analyzing / [Analyzing the Sales Discovery Library](#)
  - dimensions / [Dimensions](#)
  - measures / [Measures](#)
  - visualizations / [Visualizations](#)
- scatter chart
  - about / [The scatter chart](#)
- security
  - about / [Security](#)
  - authentication / [Authentication and authorization](#)
  - authorization / [Authentication and authorization](#)
  - content security / [Content security](#)
- services, Qlik Sense®
  - Qlik Sense Engine Service (The QIX engine) / [Services](#)
  - Qlik Sense Printing Service / [Services](#)
  - Qlik Sense Proxy Service (QPS) / [Services](#)
  - Qlik Sense Repository Service (QRS) / [Services](#)
  - Qlik Sense Repository Database / [Services](#)
  - Qlik Sense Scheduler Service (QSS) / [Services](#)
  - Qlik Sense Service Dispatcher / [Services](#)
- sheets
  - about / [Sheets](#)
  - base sheets / [Sheets](#)



- my sheets / [Sheets](#)
- community sheets / [Sheets](#)
- stories
  - creating / [Creating and sharing stories](#)
  - defining / [Defining a story](#)
  - snapshots, creating / [Creating snapshots](#)
  - text, adding / [Adding text](#)
  - shapes, adding / [Adding shapes](#)
  - media library / [Media library](#)
  - publishing / [Publishing your story](#)
- storytelling
  - about / [Storytelling](#)
- streams
  - about / [Introducing streams](#), [Streams](#)
- Streams / [Starting application authoring](#)

# T

- tables
  - about / [Tables](#)
  - standard table / [Tables](#)
  - pivot table / [Tables](#)
- tasks
  - about / [Tasks](#)
  - reload tasks / [Tasks](#)
  - user synchronizations / [Tasks](#)
- tokens / [Tokens](#)
- Travel Expense Discovery application
  - business problem, analyzing / [The business problem](#)
  - features / [Application features](#)
  - expenses, tracking / [Tracking expenses](#)
  - expenses overspent, analyzing / [Analyzing expenses overspent](#)
  - data, analyzing / [Digging deeper into the data](#)
  - travel expense story, creating / [Creating an analysis story for travel expenses](#)
  - developing / [Developing the application](#)
  - key tables, examining / [Examining the key tables](#)
  - dimensions / [Dimensions](#)
- travel expense story
  - creating / [Creating an analysis story for travel expenses](#)
  - overview, creating / [Creating an overview](#)
  - analysis, sharing / [Sharing our analysis](#)
  - finishing / [Finishing the story](#)
- tree map
  - about / [The tree map](#)

# U

- user types
  - developer / [Security rules](#)
  - contributor / [Security rules](#)
  - consumer / [Security rules](#)

# W

- web mashups
  - about / [Web mashups](#)