

Global Dashboard

USER MANUAL GUIDELINES

2018

**Revision History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Version** | **Release Date** | **Author** | **Reviewer** | **Revised Sections** | **Remarks** |
| 1 | 1 |  | Saurabh Goel |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |
| --- |
|  |

Table of Contents

[1. Introduction 3](#_Toc512938905)

[2. Qlik View Dashboard Schema Flow Diagram 3](#_Toc512938906)

[3. Qlik view web dashboard 4](#_Toc512938907)

[4. ODBC Driver for Connection 5](#_Toc512938908)

[5. Client specific DB schema 11](#_Toc512938909)

[6. Cubes tables 12](#_Toc512938910)

[7. Mirror DB 12](#_Toc512938911)

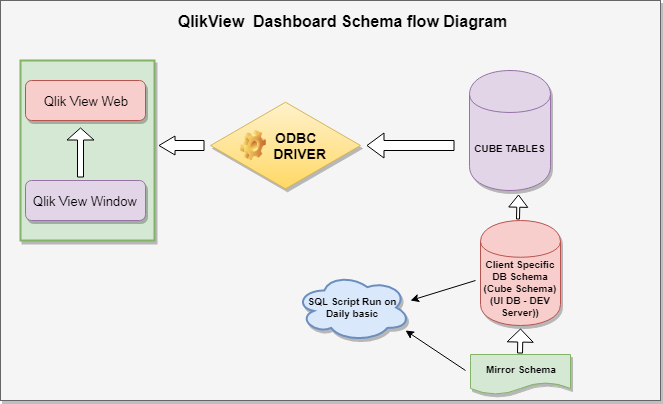
[8. Script Execution on daily/Weekly basic 13](#_Toc512938912)

1. Introduction

Qlik View dashboard is basically used by Admin user, which is internal to the system. In this document we are showing dashboard schema of Qlik view system. It consists of data flow from DB tables to Web dashboard. Main component of the system are given below.

* Qlik View Web Dashboard
* ODBC Driver for connection
* Cubes tables
* Client specific DB schema
* Mirror schema
* Script execution on Daily basic

1. Qlik View Dashboard Schema Flow Diagram



# **Qlik view web dashboard**

QlikView is a leading Business Discovery Platform. It is very powerful in visually analysing the relationships between data. It does in-memory data processing and stores the data in the report itself that it creates. It can read data from numerous sources including files and relational databases. It is used by businesses to get deeper insight by doing advanced analytics on the data they have.

Features of QlikView

QlikView has patented technology, which enables it to have many features that are useful in creating advanced reports from multiple data sources quickly. Following is a list of features that makes QlikView very unique.

* Data Association is maintained automatically − QlikView automatically recognizes the relationship between each piece of data that is present in a dataset. Users need not preconfigure the relationship between different data entities.
* Data is held in memory for multiple users, for a super-fast user experience − The structure, data and calculations of a report are all held in the memory (RAM) of the server.
* Aggregations are calculated on the fly as needed − As the data is held in memory, calculations are done on the fly. No need of storing pre-calculated aggregate values.
* Data is compressed to 10% of its original size − QlikView heavily uses data dictionary. Only essential bits of data in memory is required for any analysis. Hence, it compresses the original data to a very small size.
* Visual relationship using colors − The relationship between data is not shown by arrow or lines but by colors. Selecting a piece of data gives specific colors to the related data and another color to unrelated data.
* Direct and Indirect searches − Instead of giving the direct value a user is looking for, they can input some related data and get the exact result because of the data association. Of course, they can also search for a value directly.

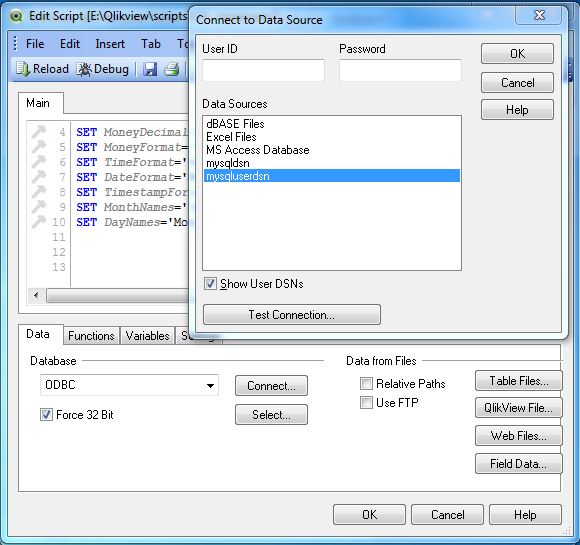
# **ODBC Driver for Connection**

QlikView can connect to most of the popular databases like MySQL, SQL Server, Oracle, Postgress etc. It can fetch data and table structures into QlikView environment and store the results in its memory for further analysis. The steps to connect to any of these databases involves creating an ODBC connection using a DSN and then using this DSN to fetch the data.

For this tutorial, we will be connecting to MySQL database. This tutorial assumes you have a MySQL environment available. Create an ODBC DSN (Data Source Name) for MySQL, following these steps − [to create DSN.](https://dev.mysql.com/doc/connector-odbc/en/connector-odbc-configuration-dsn-windows.html)Name the DSN as mysqluserdsn or you may prefer to use the existing one if you have already created a DSN for MySql.

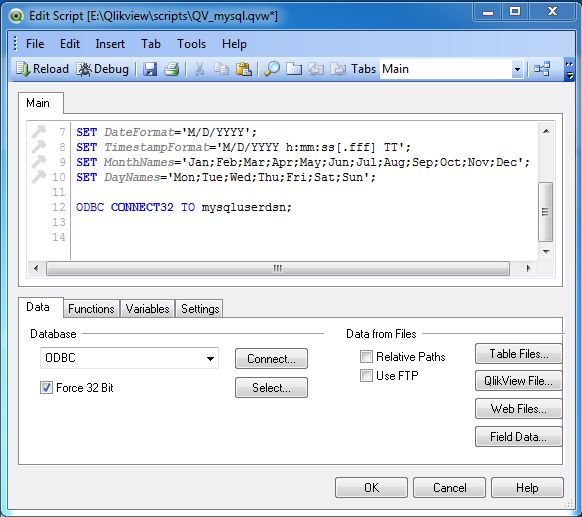
Connecting to the Database

For this chapter we will use the MySql inbuilt database named sakila. We create a new QlikView document and open the script editor (pressing Control+E). Under the tab Data, we locate the section named Database. Choose ODBC from the drop down list and click Connect. The following window opens. Choose the DSN named mysqluserdns and click Test Connection. The message Connection Test succeeded should appear.



**Verifying Connection to Database**

On successful connection, the screen given below appears showing the connection to the DB in the main window of the script editor.



**Select Database Table**

Click Select iin the above window to get the list of tables and columns. Here as we have created the DSN with sakila as the default database we get the list of tables and columns from this database. We can choose another database from the database drop down list as shown in the screenshot given below. We will continue using the sakila database for this chapter.

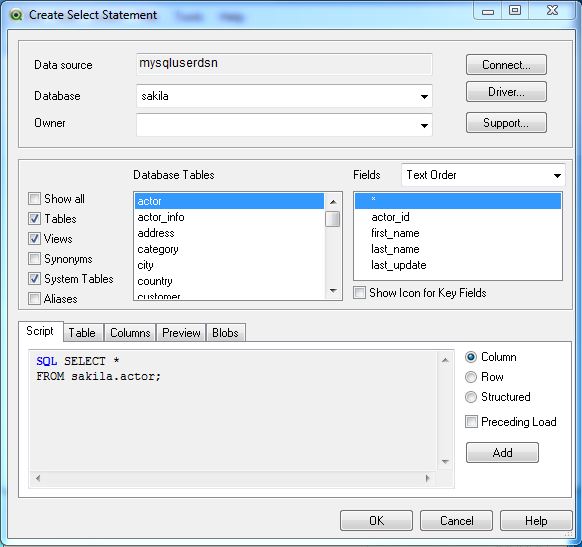
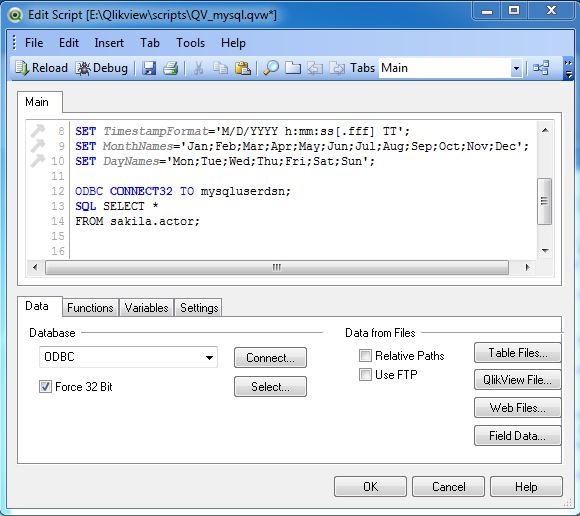


Table Loader Script

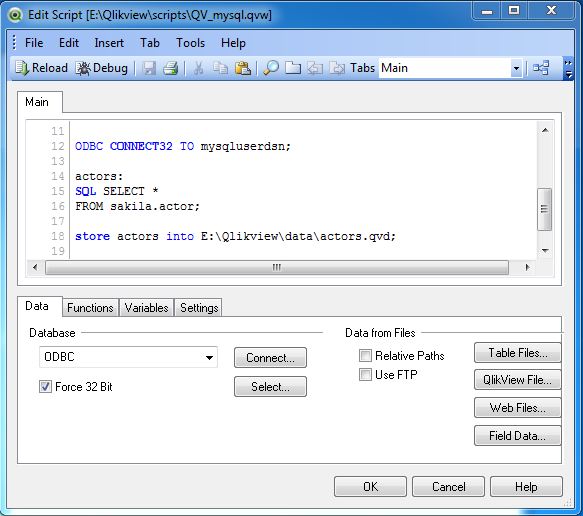
On Clicking OK in the above window, we get back to the main script editor showing the script for using the table named actor.



Showing the Result in qvd File

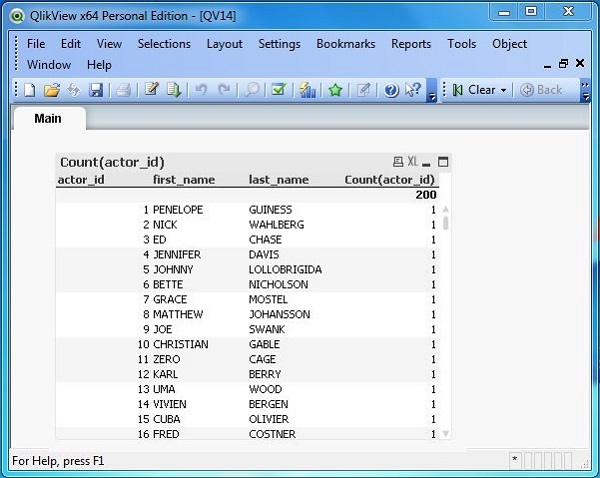
Now the data loaded into QlikView document needs to be stored permanently to be analyzed further. For this, we will edit the script to store the data in the form of a qvd file. Press Control+E to open the edit script window and write the following code.

In the code, we give appropriate names to the columns and mention the table name above the load statement. In addition, we give a path where the generated qvd file will be stored. Save this file as QV\_mysql.qvw



**Using the qvd File**

The qvd file can be loaded into the main document and used to create graphs and tables for further analysis. Press Control+R to reload the QV\_mysql.qvw file and click Next in the chart wizard. Choose the straight table to be created with actor\_id, first\_name, last\_name as the dimensions and count of actor\_id as the expression. A chart appears as given below.



# **Client specific DB schema**

Qlik view dashboard is available for five revenue leakage clients.

1. AHS
2. LP
3. CP
4. Mercy
5. MSH

Below URL consist of folder of each client.

<https://svn.operasolutions.com/svn/RevLeakage/Utilities/Qlikview/customer>

Each client specific folder has below files

* Cube table schema for all analysis
* Mirrors table schema for all analysis
* Daily run script file (run.sh)
* Weekly run script file (runWeekly.sh)

# **Cubes tables**

Web dashboard get the data from Cubes table with the help of ODBC driver and show in the dashboard. Cube tables get the data from client specific DB schema in Mirror DB. Sql script is running on daily/weekly basic to fill the cubes tables.

As a naming convention prefix with **CB** are cubes tables.

**Use case of LP client for ACOOUNT\_VOLUME\_ANALYSIS**

Below script file consist of Cubes table and Mirror tables used in above analysis.

**(**<https://svn.operasolutions.com/svn/RevLeakage/Utilities/Qlikview/customer/LP/ACCOUNT_VOLUME_ANALYSIS.sql>**)**

**CB\_ACCOUNT\_VOLUME\_ANALYSIS - Cube table used in above analysis.**

# **Mirror DB**

Mirror DB tables is responsible to populate cubes table which is further required to show data on dashboard.

**Use case of LP client for ACOOUNT\_VOLUME\_ANALYSIS**

Below script file consist of Mirror DB tables used in above analysis.

**(**<https://svn.operasolutions.com/svn/RevLeakage/Utilities/Qlikview/customer/LP/ACCOUNT_VOLUME_ANALYSIS.sql>**)**

Below Mirror tables used in above analysis.

* RL\_LP\_Mirror.T\_PATIENT
* RL\_LP\_Mirror.T\_CHARGES
* RL\_LP\_Mirror.T\_HOSPITAL
* RL\_LP\_Mirror.T\_PREDICTIONS\_PRE
* RL\_LP\_Mirror.T\_PREDICTIONS\_POST
* RL\_LP\_Mirror.T\_INSURANCE
* RL\_LP\_Mirror.T\_PAYMENTS
* RL\_LP\_Mirror.T\_ADJUSTMENTS
* RL\_LP\_Mirror.DAILY\_SNAPSHOT\_SCORABLE

# **Script Execution on daily/Weekly basic**

Some script required to run on daily basis and some weekly. Below script files are available in each client folder

* run.sh – (Data update on daily basis)
* runWeekly.sh (Data update on weekly basis)

**run**.sh –

#### cd /home/tomcat/LP

#### svn update

#### chmod 777 \*

#### mysql -v -v -v -u application -papp@123 RL\_LP\_UI < /home/tomcat/LP/HITRATE\_DASHBOARD.sql >> /home/tomcat/LP/HITRATE\_DASHBOARD.log

runWeekly.sh

#### cd /home/tomcat/LP

#### svn update

#### chmod 777 \*

#### mysql -v -v -v -u application -papp@123 RL\_LP\_UI < /home/tomcat/LP/ACCOUNT\_VOLUME\_ANALYSIS.sql >> /home/tomcat/LP/ACCOUNT\_VOLUME\_ANALYSIS.log

#### mysql -v -v -v -u application -papp@123 RL\_LP\_UI < /home/tomcat/LP/PREDICTION\_ANALYSIS.sql >> /home/tomcat/LP/PREDICTION\_ANALYSIS.log

#### mysql -v -v -v -u application -papp@123 RL\_LP\_UI < /home/tomcat/LP/CHARGES\_ANALYSIS.sql >> /home/tomcat/LP/CHARGES\_ANALYSIS.log