QlikView

Best Practices

The QlikView Data Model & Working with Multiple Events

Version: 1
Date: 12/04/2007
Author(s): STB/ABY

"A best practice is a technique or methodology that, through experience and research, has proven to reliably lead to a desired result."

Contents

CONTENTS 2

WHAT IS THE QLIKVIEW DATA MODEL 3

THE STAR SCHEMA APPROACH 4

MULTIPLE FACT TABLES/MULTIPLE EVENTS 5

CONCATENATE, One Fact Table Solution 5

Joining Fact Tables Solution 6

Central Link Table Solution (Event Space) 6

AGGREGATION 7

What is the QlikView Data Model

When you load your data in to the QlikView application, a data model will be created based on the tables and columns you have in your script and also the names of the columns and any resident loads and joins you have previously defined.

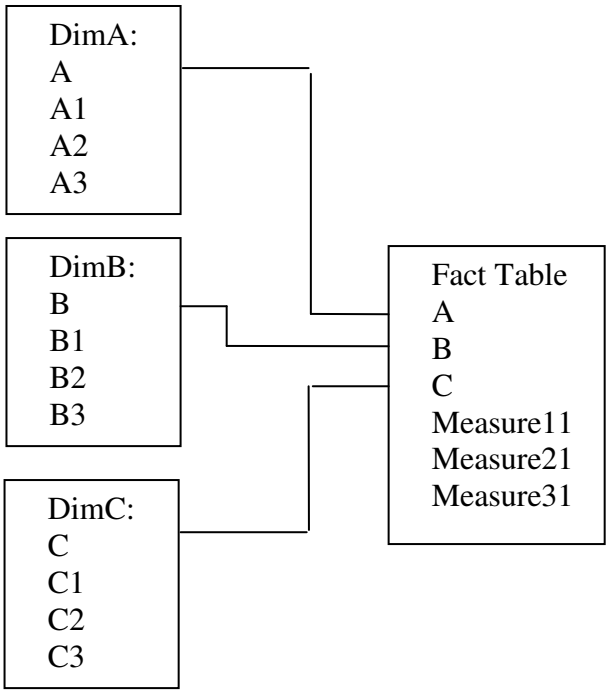
You will of course be driven by the type and structure of your data sources. These sources and the underling data will have to be maipulated within the script to deliver the Data Model that best
suites your data for both performance and usability.

The Star Schema Approach

The standard layout and structure of data presentation is the Star Schema. QlikView is generally most efficient when working in this space.

> The **star schema** (sometimes referenced as star join schema) is the simplest data warehouse schema, consisting of a single "fact table" with a compound primary key, with one segment for each "dimension" and with additional columns of additive, numeric facts. The name star schema is derived from the fact that the schema diagram is shaped like a star.
> **(Source, Wikipedia - http://en.wikipedia.org/wiki/Star_schema)**

Within a Star schema model, the event data (transactions) reside in a central “Fact Table” and the attributes of the event reside in separate “dimension tables”. The diagram below shows the basic layout…



This model works well in a simplistic, single event scenario. But as QlikView can handle multiple data sources from many different source systems and files, we have to work with multiple event scenarios, or many fact tables.

Multiple Fact Tables/Multiple Events

CONCATENATE, One Fact Table Solution

The easiest solution is to link fact tables together but make the dimension tables event specific. You can link/merge the Fact tables by using the CONCATENATE function.

Forced Concatenation

If two or more tables do not have exactly the same set of fields, it is still possible to force QlikView to concatenate the two tables. This is done with the **concatenate** prefix in the script, which concatenates a table with another named table or with the last previously created logical table.

load a, b, c from table1.csv;
concatenate load a, c from table2,csv;

The resulting logical table has the fields a, b and c. The number of records in the resulting table is the sum of the numbers of records in table 1 and table 2. The value of field b in the records coming from table 2 is NULL.

The names of the fields must be exactly the same.
Unless a table name of a previously loaded table is specified in the **concatenate** statement the **concatenate** prefix uses the last previously created logical table. The order of the two statements is thus *not* arbitrary.

When concatenating these tables you may wish to add a manual reference column that gives you a dimension to be able to split the merged fact table. The example below shows a merging of two tables STOCK and SALES.

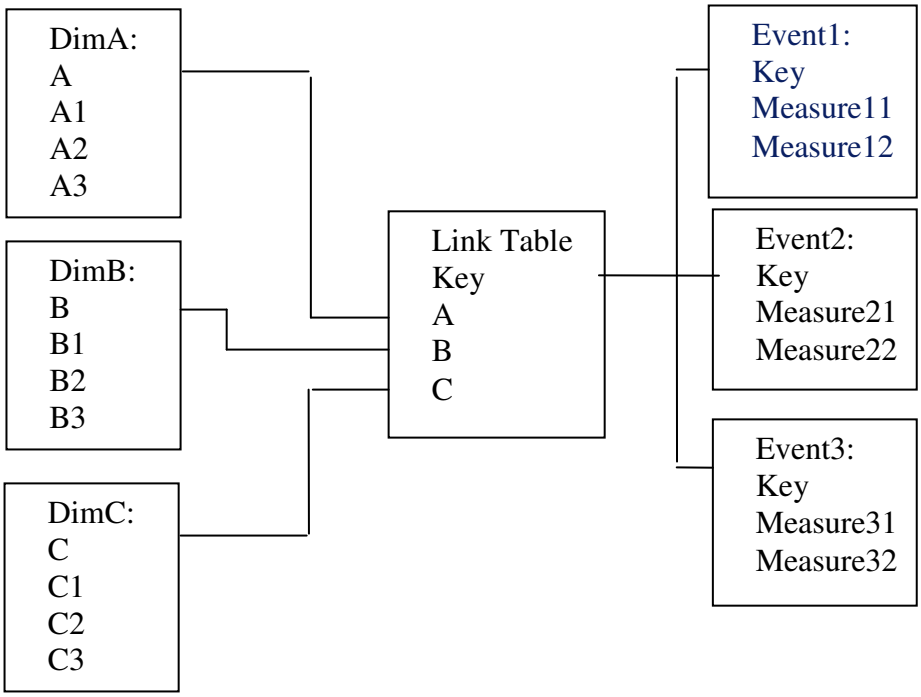
```
Transactions:
LOAD 'Sales' as Type,
     Account_Number,
     Order_Date,
     Sales_Amount
FROM Sales.qvd (qvd);
CONCATENATE (Transactions)
LOAD 'Stock' as Type,
     Account_Number,
     Order_Date,
     Stock_Amount
FROM Stock.qvd (qvd);
```

Joining Fact Tables Solution

Another solution is to force a join between the different fact tables to essentially create one large core fact table. This explicit join is a costly operation as the event specific dimensions can be statistically unrelated and the resultant fact table will grow to contain all possible values. Joining data may also mean that you remove data you actually require in your analysis. As in ANSI SQL, joining tables will require data to be present in one of the tables and that table will be the driver for the data appearing in the final joined table.

Central Link Table Solution (Event Space)

In the event of multiple fact tables QlikView In-Memory Technology allows us to create a central link table that only contains the existing data combinations. Instead of *Joining* the tables the event dimensions can be merged (CONCATENATED) in to one central Link table. This link table can then be linked back to the event measures one side and the dimension tables on the other.



Aggregation

Another QlikView feature that improves flexibility and also makes the Link Table topology even more useful is the fact that aggregations can be done both in the Script (pre-defined) and in the front end application (GUI, front end objects). The standard OLAP way of working would require predefined aggregation in the measures to create the cube. The OLAP cube then has to pre-aggregate at every level within the predefined dimensions set up in the OLAP model.

In summary, you do not have to apply any aggregation within the creation of the QlikView script if not necessary, any SUM, TOTAL etc can be performed at the GUI development stage. This enables a user to have core detail (transaction values) within an application for detailed analysis.