Early Access

QLIKVIEW

for Ninjas

BY: Rajesh Pillai

QlikView for Ninjas

The first and ultimate step to the exciting world of QlikView

Rajesh Pillai

This book is for sale at http://leanpub.com/qlikviewforninjas

This version was published on 2015-12-25

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Also By Rajesh Pillai

QlikView Recipes

This book is dedicated to my son Rohan, my nephew Tanuv who being just 9.5 years has started with CSS3, HTML and JavaScript, my wife Radhika and my parents and well wishers. This book is also dedicated to all the participants who attended my QlikView training and has given some really constructive feedback based on which this series of books have been structured.

Contents

10 – Mapping Tables

In Qlikview it is always advisable to minimize the number of tables in the data model.

Mapping Tables are used in QlikView to clean up the data model.

Tables with just 2 columns can be removed and columns of that table can be mapped to another table.

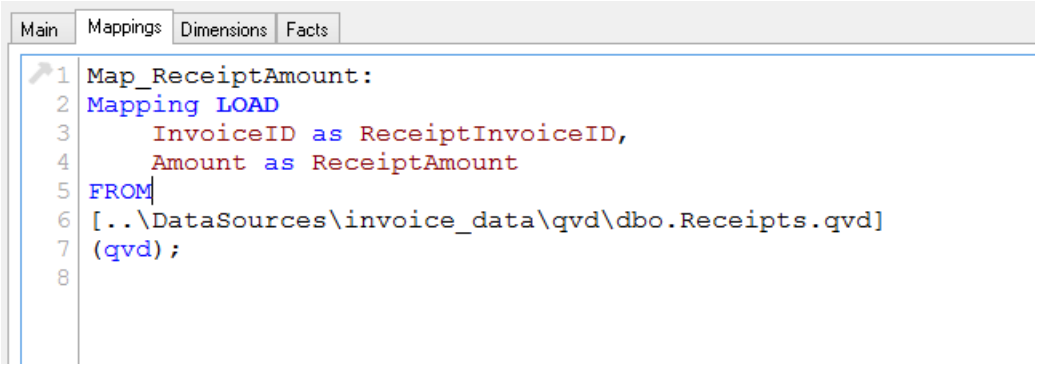
Mapping tables are stored in a separate area in memory and used only as mapping tables during script execution. After script execution they will be automatically dropped.

So, what does the function do? Well, basically it is just a lookup function – it takes one value as input, checks if this value exists in a mapping table and returns the corresponding value from the second column in the mapping table. Think of it as a translation function or a dictionary, where the translation is fetched from a pre-defined, two-column table.

So, we have to calculate the total invoice amount, the receipt amount and the pending amount. So, let's first create a mapping table that contains the InvoiceID and the Amount that is received against that invoice.

Fire up your script editor (Ctrl + E).

1. Create a new tab and name it "Mapping". Promote this tab as the second tab, just after the "Main" tab. Since this is a mapping table this table has to be loaded prior to any other tables that will use this mapping.
2. Add the below script.



1. Now, we can use the ApplyMap function in the load statement of the tables where we require this mapping. In this case we will use this function in the InvoiceDetail table.

2. Go to the tab where the InvoiceDetail table is loaded and add the below script just after the last loaded field.

ApplyMap ('Map_ReceiptAmount',InvoiceID,0) as ReceiptAmount

The full script will now look like the below figure.

```
[InvoiceDetails]:
LOAD InvoiceDetailID,
     Sn,
     Description,
     Unit,
     Rate,
     Unit * Rate as InvoiceLineAmount,
     InvoiceID,
     TimeStamp,
     ApplyMap ('Map_ReceiptAmount',InvoiceID,0) as ReceiptAmount
FROM
[..\DataSources\invoice_data\qvd\dbo.InvoiceDetails.qvd]
(qvd);
```

NOTE: ApplyMap function takes 3 parameters. The first one is the name of the Mapping table, the key field on which the lookup needs to be done, and the third optional parameter is the default value, if the key field is not found. For numeric fields we use 0 to avoid any calculation issues, and for text fields we can use any meaningful data for that table indicating missing values, as required.

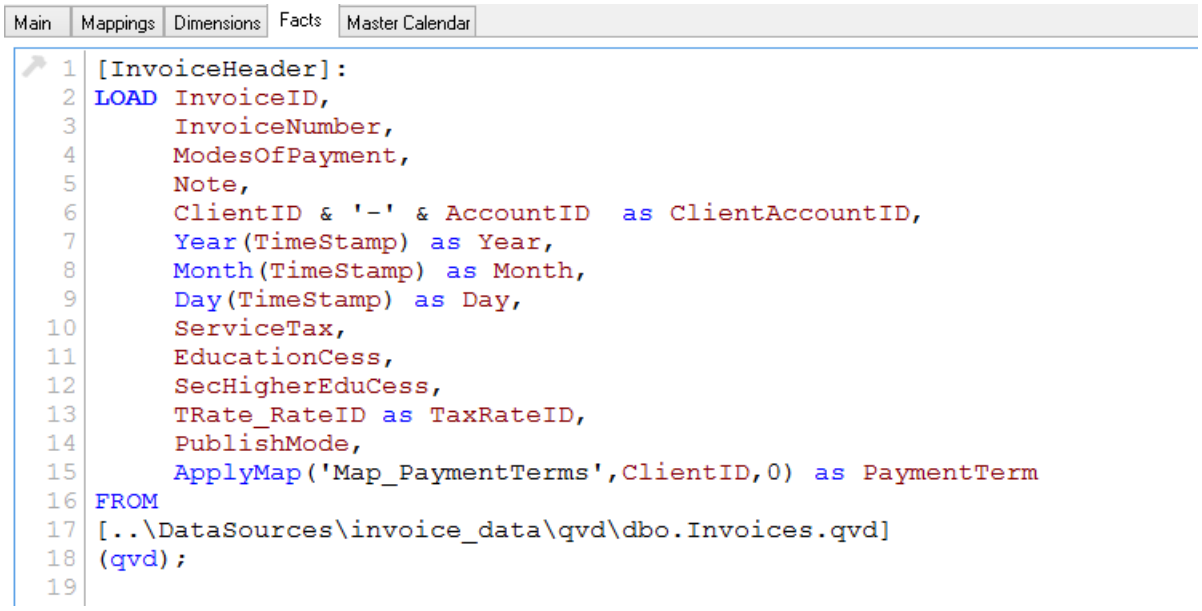
1. Reload and check the table viewer (Ctrl + T). You will see a new column ReceiptAmount added to the invoice table. Also, note, the mapping table 'Map_ReceiptAmount' is not present in the data model.

Exercise:

Load the Payment terms from Clients.qvd as a mapping table and use that in the Invoice Details table.

The final figure is given below for your reference. In the mapping table tab add the below script.

In the Invoices table update the script as shown below.



The full script in sequence is shown below for your reference. Note, the order of scripts are important.

///$tab Mappings

Map_ReceiptAmount:

Mapping LOAD

```
1  InvoiceID as ReceiptInvoiceID,
```

Amount as ReceiptAmount

FROM

[..\DataSources\invoice_data\qvd\dbo.Receipts.qvd]

(qvd);

Map_PaymentTerms:

Mapping load

ClientID,

PaymentTerms

FROM

[..\DataSources\invoice_data\qvd\dbo.Clients.qvd]

(qvd);

///$tab Dimensions

LOAD AccountID,

AccountCategoryID,
Name as AccountName,
Address,
NameOfBank,
Branch,
AccountNo,
MICR,
IFSC,
Initials,
PAN,
ServiceTaxRegistration
FROM
[..\DataSources\invoice_data\qvd\dbo.Accounts.qvd]
(qvd);
LOAD

```
1  AccountID,
```

ClientID & '-' & AccountID as ClientAccountID,

```
1  Name as ClientName,
```

ContactPerson,
PaymentTerms,
Tax,
Street,
City,
State,
Country,
PinCode
FROM
[..\DataSources\invoice_data\qvd\dbo.Clients.qvd]

```
(qvd);
LOAD Id as AccountCategoryID,
Name as CategoryName
FROM
[..\DataSources\invoice_data\qvd\dbo.AccountCategory.qvd]
(qvd);
// Load the Users Table
LOAD Id as UserId,
UserName,
FirstName,
LastName,
IsAdmin,
AccountID
FROM
[..\DataSources\invoice_data\qvd\dbo.Users.qvd]
(qvd);
QUALIFY *;
UNQUALIFY UserId;
LOAD Id,
UserId,
City,
Pincode,
State,
Country
FROM
[..\DataSources\invoice_data\qvd\dbo.UserAddresses.qvd]
(qvd);
UNQUALIFY *;
// Load the Invoice publish status table
LOAD Id as PublishMode,
```

```
Status as PubishStatus
FROM
[..\DataSources\invoice_data\qvd\dbo.InvoicePublishMode.qvd]
(qvd);
///$tab Facts
LOAD InvoiceID, InvoiceNumber, ModesOfPayment,
Note,
ClientID,
ClientID & '-' & AccountID as ClientAccountID,
Year(TimeStamp) as Year,
Month(TimeStamp) as Month,
Day(TimeStamp) as Day,
ServiceTax,
EducationCess,
SecHigherEduCess,
TRate_RateID as TaxRateID,
PublishMode,
ApplyMap('Map_PaymentTerms',ClientID,0) as PaymentTerm
FROM
[..\DataSources\invoice_data\qvd\dbo.Invoices.qvd]
(qvd);
LOAD InvoiceDetailID,
Sn,
Description,
Unit,
Rate,
Unit * Rate as InvoiceLineAmount,
InvoiceID,
TimeStamp,
ApplyMap ('Map_ReceiptAmount',InvoiceID,0) as ReceiptAmount
```

```
FROM
[..\DataSources\invoice_data\qvd\dbo.InvoiceDetails.qvd]
(qvd);
// Load Receipts table
LOAD ReceiptID,
Date,
Amount,
ChequeNo,
Discount,
Note as ReceiptNote,
InvoiceID
FROM
[..\DataSources\invoice_data\qvd\dbo.Receipts.qvd]
(qvd);
```