Business Intelligence Solution Providers
Specialized in creating talent resource pool

Getting Started with QlikView Part XIII
Addressing the Data model issue (Synthetic Key issue)

Description:

BISP is committed to provide BEST learning material to the beginners and advance learners. In the same series, we have prepared a complete end-to end Hands-on Beginner's Guide for Qlikview Report, Synthetic key issue, Subfield Function. The document focuses on script editor. Join our professional training program and learn from experts.

History:

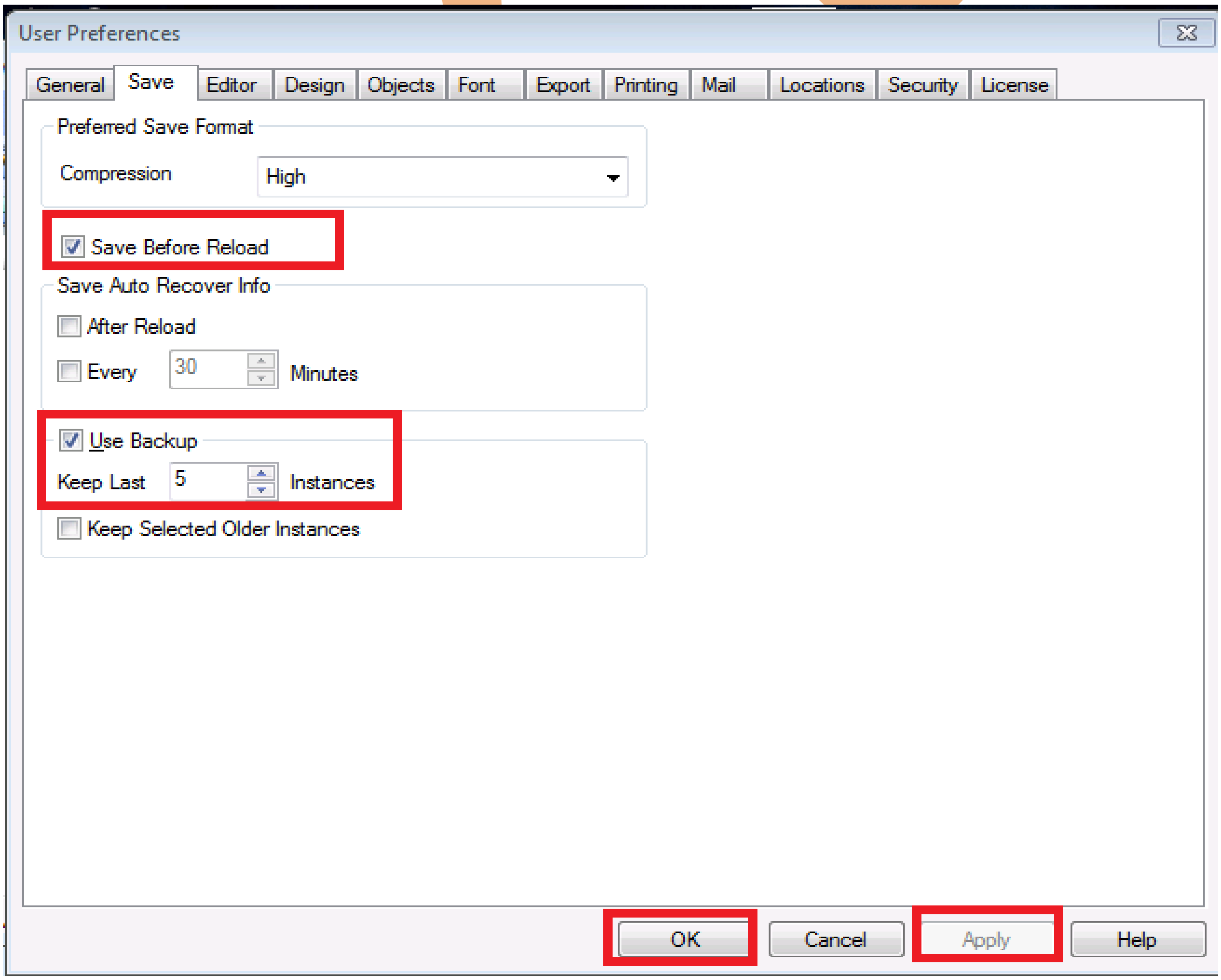
Version	Description Change	Author	Publish Date
0.1	Initial Draft	Surbhi sahu	21st Aug 2012
0.1	Review#1	Rajkumar N	29th Aug 2012

Table of Contents

1) Aspects related to reloading the application:

As we familiar about the reload function is to load the data from different data sources into the QlikView data model. By default, when QlikView encounters errors during the reload of a document, it automatically closes the document and reloads the last saved version of the file / data. It can be a frustrating experience when you have just written a lot of script, only to see all of it lost because you forgot a semicolon somewhere and didn't save the script which you have written all this time. And you forgot to save the script before reloading the data.

One way to avoid this problem is by always first saving your script before reloading. This can be done by 1) File > Save Entire Document from the menu, 2) Ctrl + S, 3) Clicking on the Save icon in the toolbar, 4) Another more safe way is to set in QlikView to automatically save the file before each reload. To do this, close the script editor and open the User Preferences menu by selecting Settings > User Preferences from the menu, or by pressing Ctrl + Alt + U. In the menu, select the Save tab and tick the checkbox labeled Save Before Reload. It is also advisable to tick the checkbox Use Backup and set the field Keep Last Instances to 5. This last option ensures that the last 5 versions of the QlikView file are kept as backup.



Preferred Save Format

Compression: High

☑ Save Before Reload

Save Auto Recover Info

☐ After Reload

☐ Every 30 Minutes

☑ Use Backup

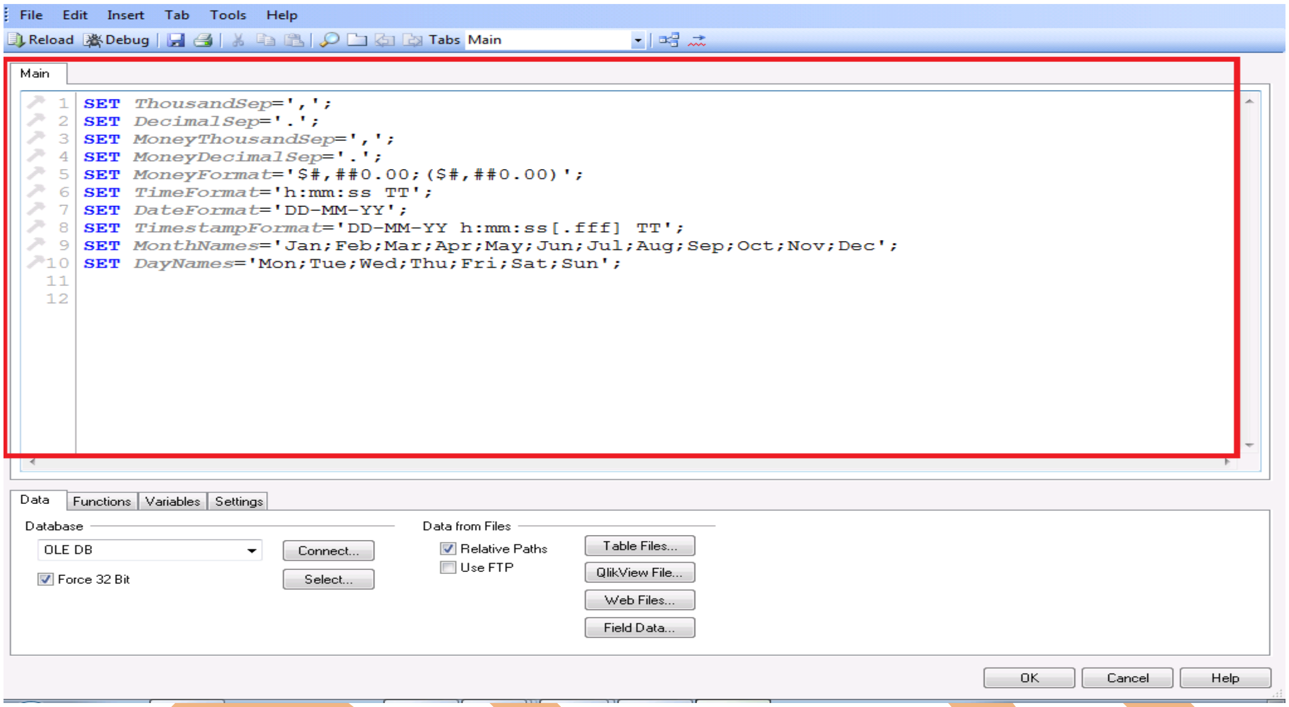
Keep Last 5 Instances

☐ Keep Selected Older Instances

Then->Apply->ok

2) The Script Editor

Script editor is the portion where the script are written and tables are loaded in application. This is mainly termed as the “Heart of QlikView”. All the ETL work is done here.



```
SET ThousandSep=',';
SET DecimalSep='.';
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$#,##0.00;($#,##0.00)';
SET TimeFormat='h:mm:ss TT';
SET DateFormat='DD-MM-YY';
SET TimestampFormat='DD-MM-YY h:mm:ss[.fff] TT';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

OLE DB

☑ Force 32 Bit

☑ Relative Paths

☐ Use FTP

The script editor consists of the following areas:

- A menu bar
- A toolbar
- A script pane
- A tool pane

2.1)Menu and toolbar

The menu offers a wide range of options

File | Reload= Runs the entire script to reload the data File |

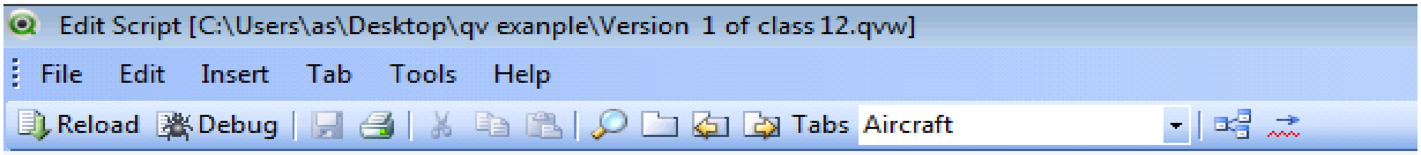
Save Entire Document= Saves the entire document, not just the

Tab | Add Tab = script QlikView scripts can be organized using tabs; this function adds a new tab

Tab | Rename= Renames the currently selected tab

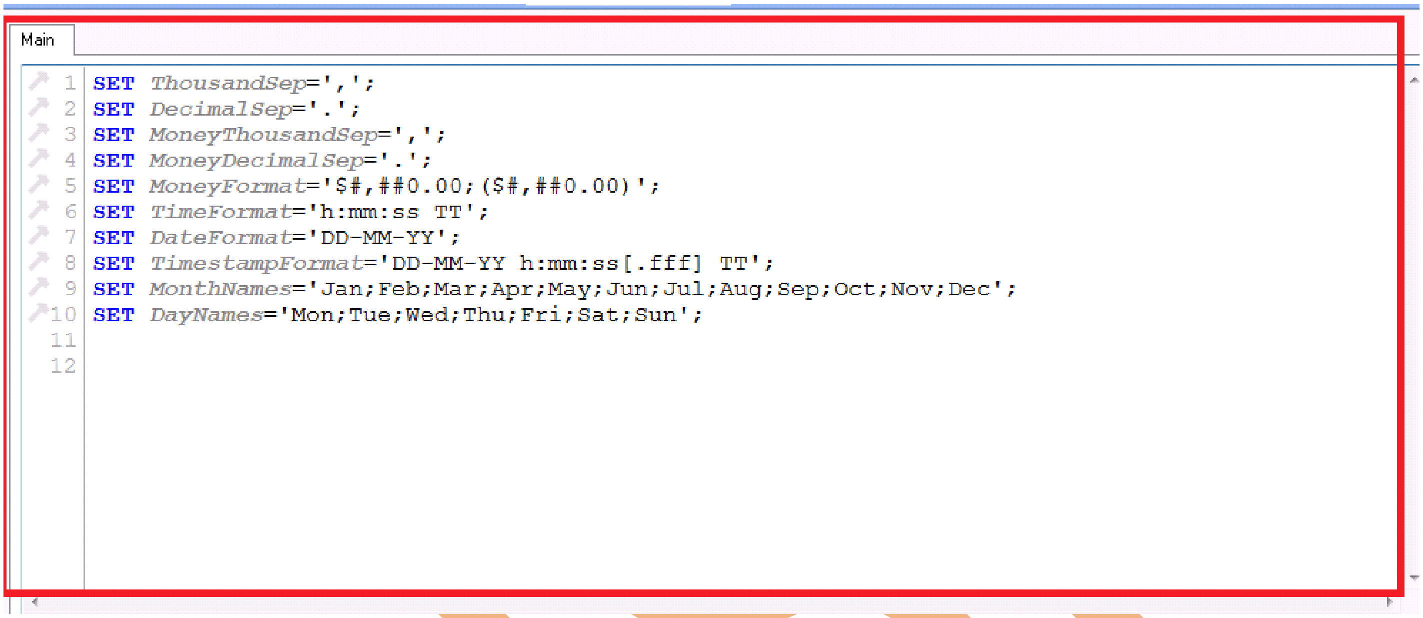
Tab | Promote= Moves the currently selected tab to the left

Tab | Demote= Moves the currently selected tab to the right


Edit Script [C:\Users\as\Desktop\qv exanple\Version 1 of class 12.qvw]
File
Edit
Insert
Tab
Tools
Help
Reload
Debug
Tabs
Aircraft

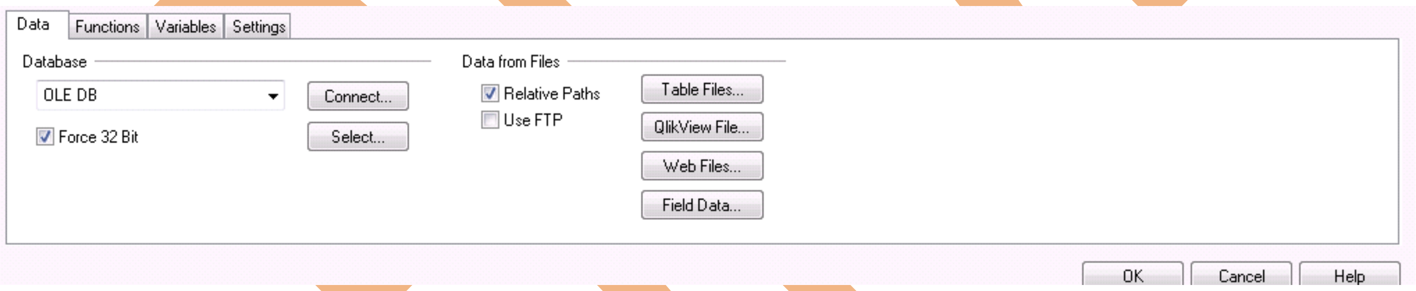
2.2) Script pane

This is the main working area of the script editor. When a new QlikView data model is created, the new script is populated with a Main tab and a few lines where number interpretation variables are added. These lines tell QlikView how to interpret various numbers and are generated automatically based on your operating system settings.



```
SET ThousandSep=',';
SET DecimalSep='.';
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$#,##0.00;($#,##0.00)';
SET TimeFormat='h:mm:ss TT';
SET DateFormat='DD-MM-YY';
SET TimestampFormat='DD-MM-YY h:mm:ss[.fff] TT';
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
```

2.3) Tool pane



Database: OLE DB

☑ Force 32 Bit

☑ Relative Paths

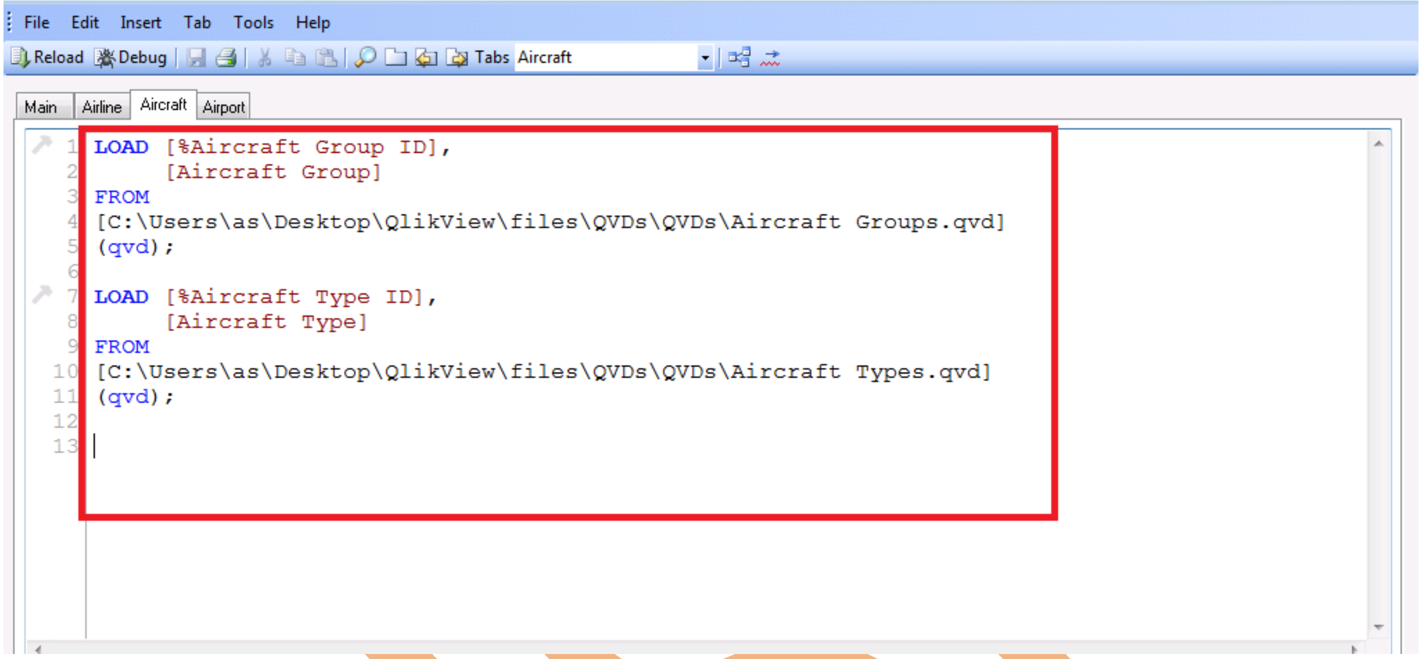
□ Use FTP

- ➢ The Data tab gives QV the flexibility to get / connect to different data sources to get data.
- ➢ The Functions tab gives a categorized overview of the script functions available in QlikView.
- ➢ The Variables tab shows user and system variables. This tab is populated after each reload, so in a new document it will be blank.
- ➢ The Settings tab contains some additional settings with regards to system access and password scrambling.

2.4) Script statement

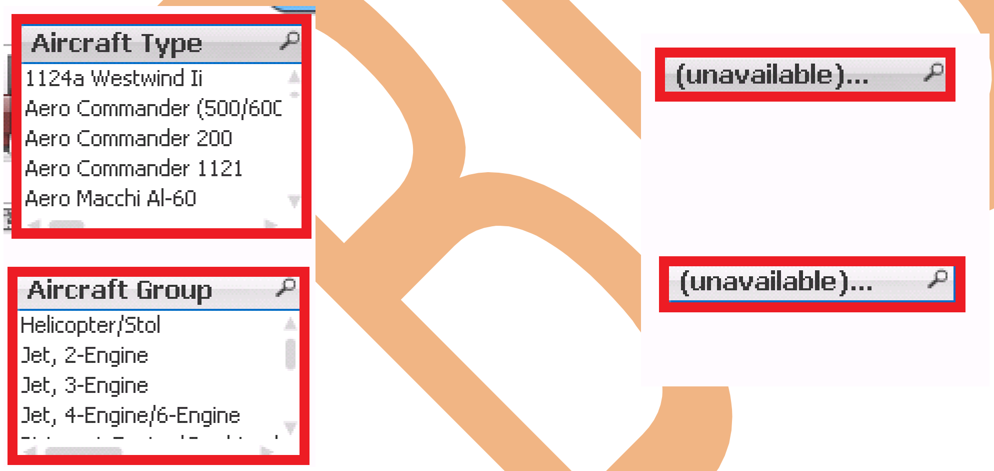
A QlikView script is made up of a sequence of LOAD / SQL statements. These statements are typically used to either manipulate the data, or to conditionally control the way in which the script is executed. For example, we may want to combine two tables together, or skip over a part of a script if a condition is not met.

We look some other actions which we can perform through the QlikView


File Edit Insert Tab Tools Help
Reload Debug Tabs Aircraft
Main Airline Aircraft Airport

```
LOAD [%Aircraft Group ID],
     [Aircraft Group]
FROM
[C:\Users\as\Desktop\QlikView\files\QVDs\QVDs\Aircraft Groups.qvd]
(qvd);

LOAD [%Aircraft Type ID],
     [Aircraft Type]
FROM
[C:\Users\as\Desktop\QlikView\files\QVDs\QVDs\Aircraft Types.qvd]
(qvd);
```

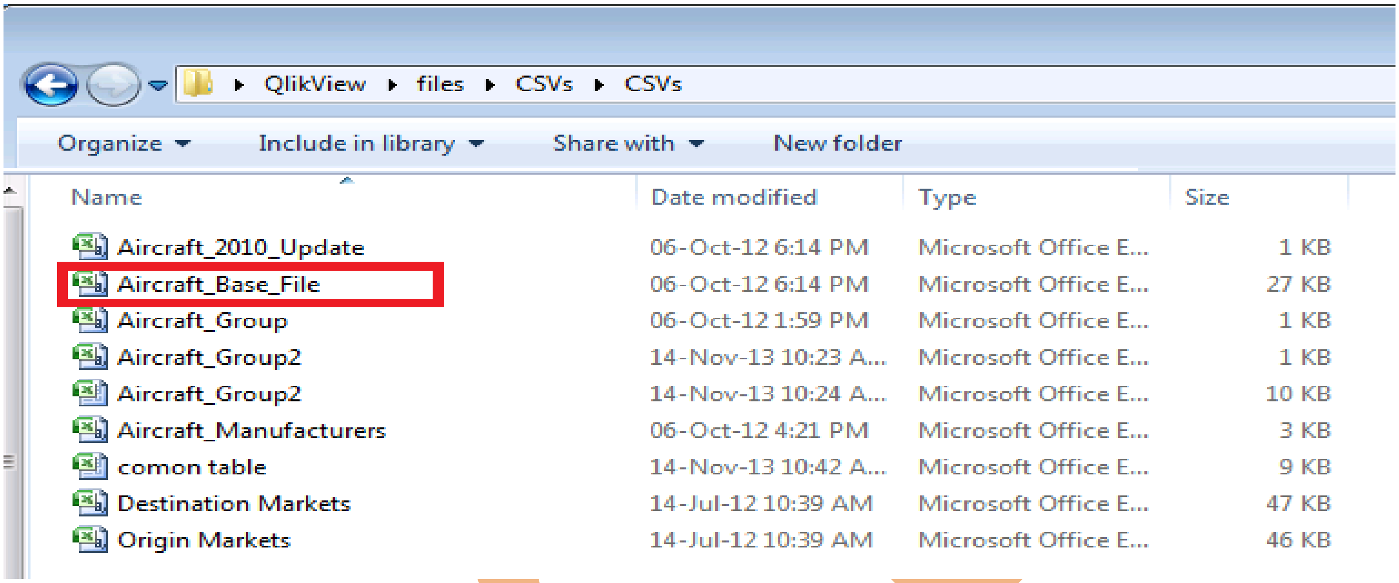
Here you can see in the script editor window the script is here for the table Aircraft Groups and Aircraft Types If we erase the script from here this will disappear the data from both tables in the sheet. Look here the both fields are shown as (unavailable).



3) Loading the aircraft information

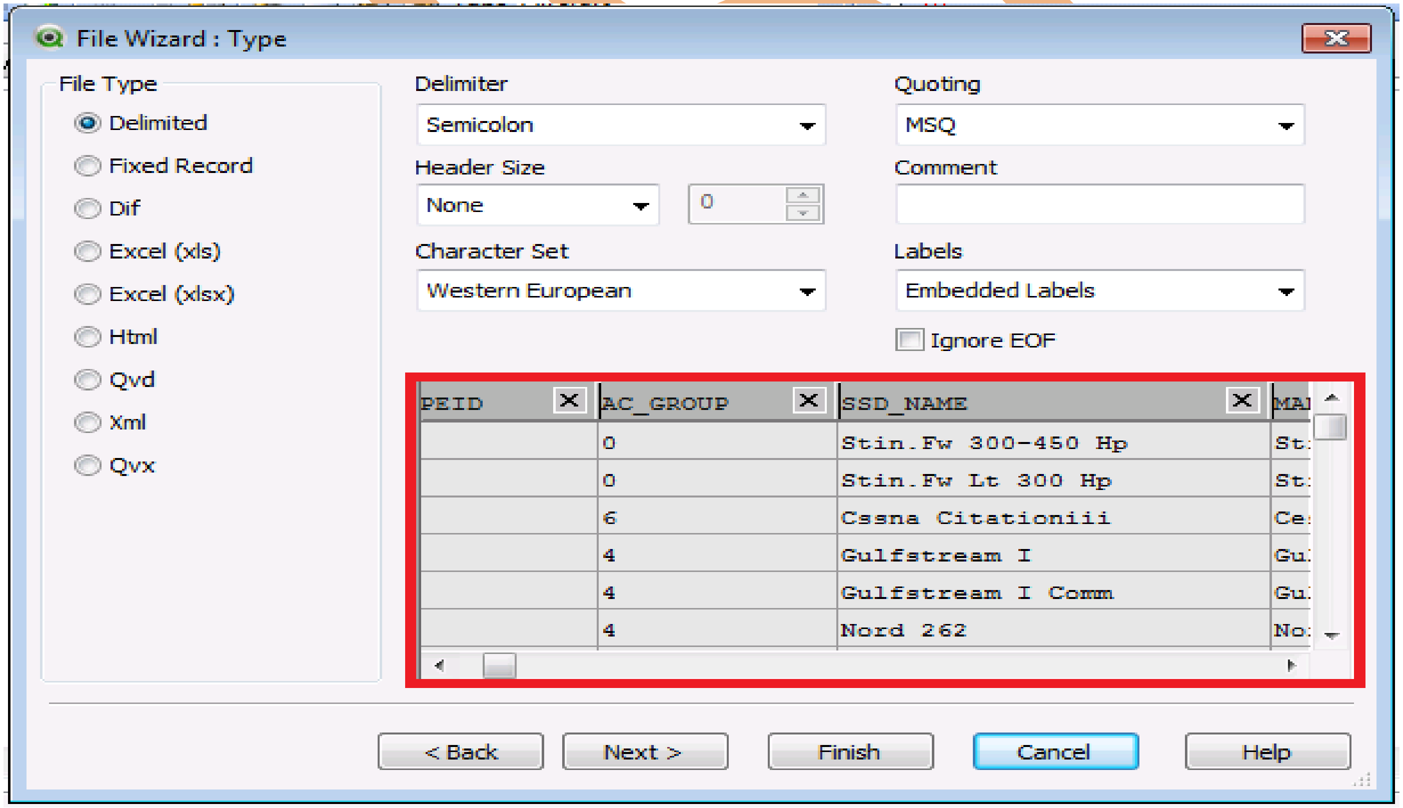
Loading of aircraft information using these steps:

Step1: Choose the file for upload by clicking on -> Table files in the Tool Pane > Browse the location of the particular file which we want to load, select "Aircraft_Base_File.csv" -> OK basically this file consists of all the updates prior to the year 2010. We have loaded it in to the application.



Name	Date modified	Type	Size
Aircraft_2010_Update	06-Oct-12 6:14 PM	Microsoft Office E...	1 KB
Aircraft_Base_File	06-Oct-12 6:14 PM	Microsoft Office E...	27 KB
Aircraft_Group	06-Oct-12 1:59 PM	Microsoft Office E...	1 KB
Aircraft_Group2	14-Nov-13 10:23 A...	Microsoft Office E...	1 KB
Aircraft_Group2	14-Nov-13 10:24 A...	Microsoft Office E...	10 KB
Aircraft_Manufacturers	06-Oct-12 4:21 PM	Microsoft Office E...	3 KB
comon table	14-Nov-13 10:42 A...	Microsoft Office E...	9 KB
Destination Markets	14-Jul-12 10:39 AM	Microsoft Office E...	47 KB
Origin Markets	14-Jul-12 10:39 AM	Microsoft Office E...	46 KB

Step 2: Now the table will be loaded in the QlikView



File Type
- ● Delimited
- ○ Fixed Record
- ○ Dif
- ○ Excel (xls)
- ○ Excel (xlsx)
- ○ Html
- ○ Qvd
- ○ Xml
- ○ Qvx

Delimiter: Semicolon

Quoting: MSQ

Header Size: None 0

Comment:

Character Set: Western European

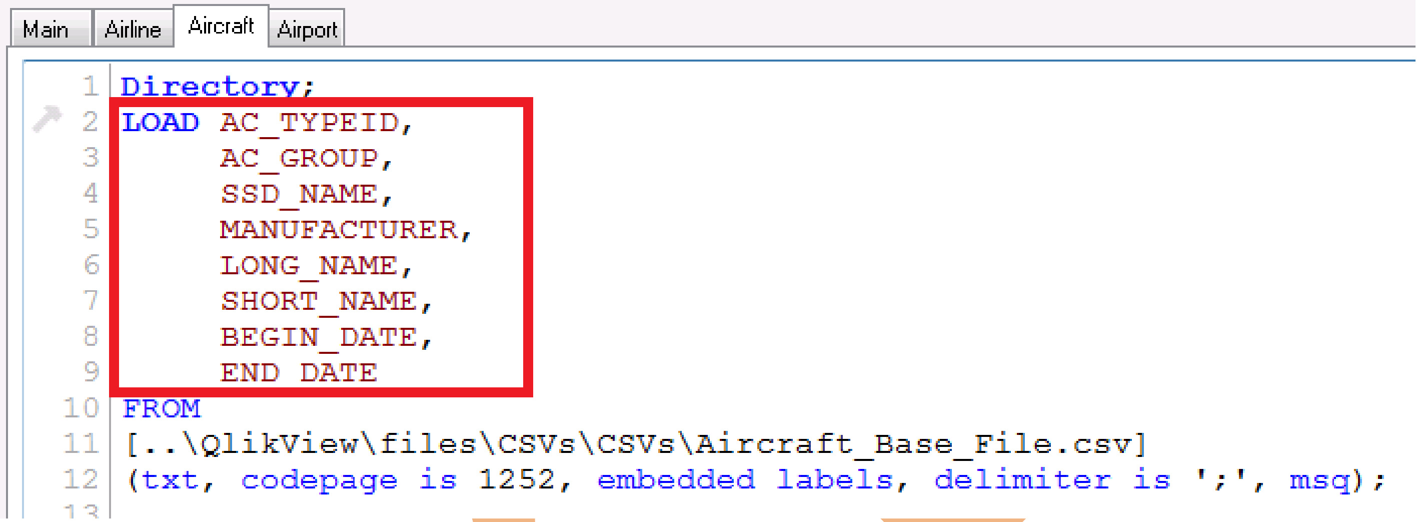
Labels: Embedded Labels

□ Ignore EOF

PEID	AC_GROUP	SSD_NAME	MA
	0	Stin.Fw 300-450 Hp	St:
	0	Stin.Fw Lt 300 Hp	St:
	6	Cssna Citationiii	Ce:
	4	Gulfstream I	Gu:
	4	Gulfstream I Comm	Gu:
	4	Nord 262	No:

The QV File Wizard has identified the file type as “Delimited”, which we are loading > Finish.

Step 4:

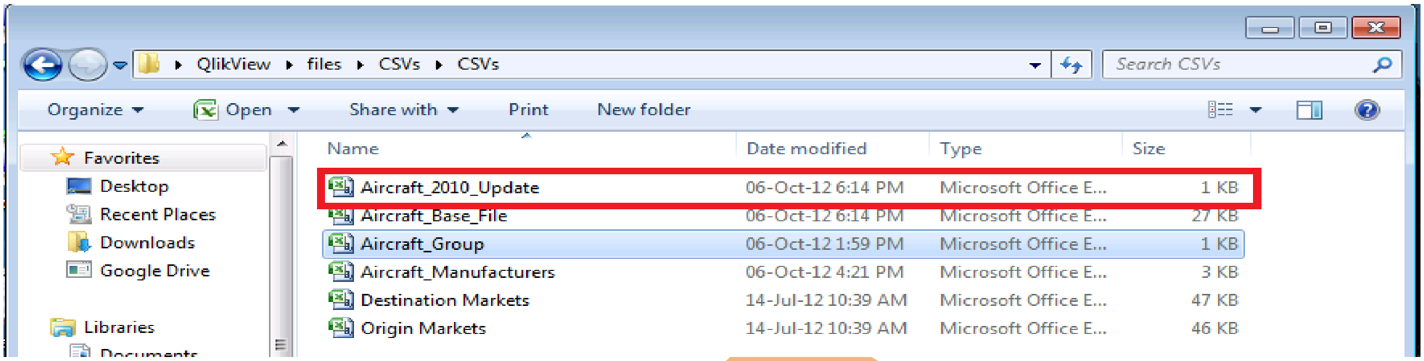


```
Directory;
LOAD AC_TYPEID,
     AC_GROUP,
     SSD_NAME,
     MANUFACTURER,
     LONG_NAME,
     SHORT_NAME,
     BEGIN_DATE,
     END_DATE
FROM
[..\QlikView\files\CSVs\CSVs\Aircraft_Base_File.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);
```

Here the script is loaded in the table but the name of the fields name are not properly maintained (so as to maintain joins between the tables) so we rename the fields by using ‘as’ keyword.

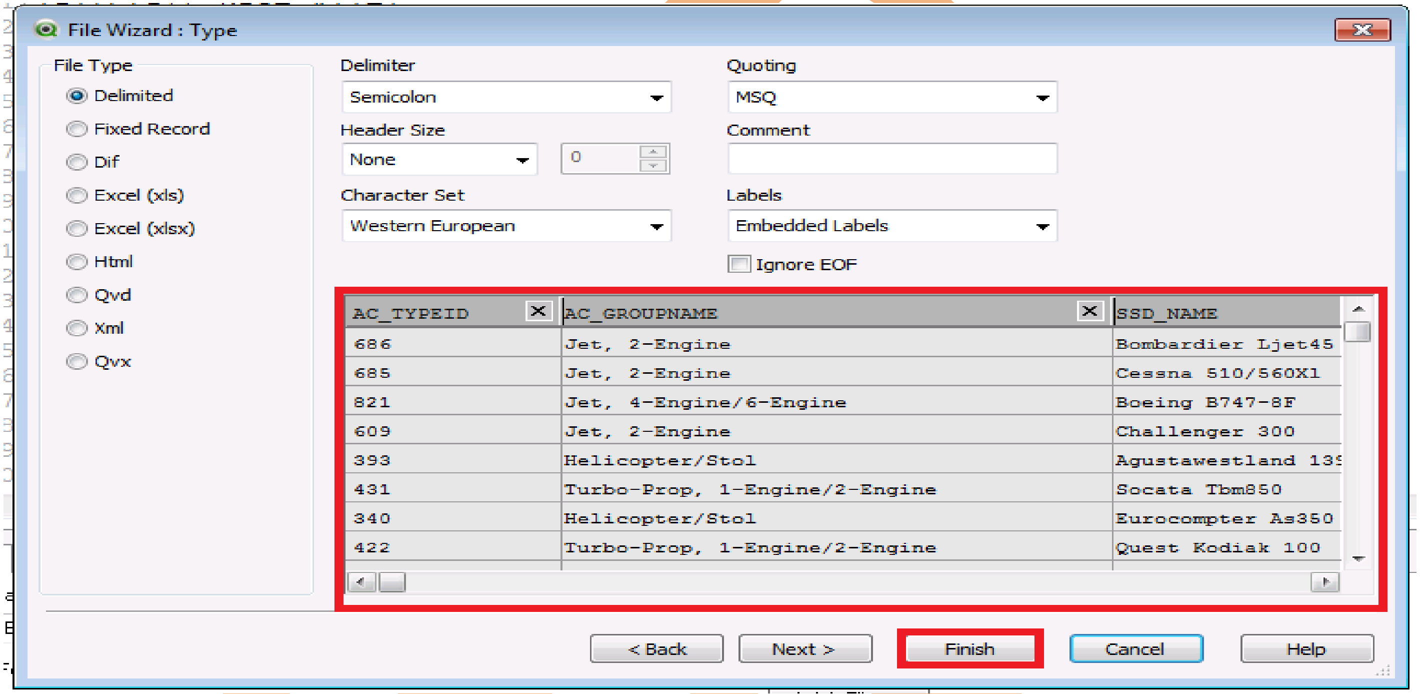
```
[Aircraft Types]:
LOAD AC_TYPEID as [%Aircraft Type ID],
    AC_GROUP as [%Aircraft Group Type],
    SSD_NAME as [Aircraft Name],
    MANUFACTURER as [Aircraft Manufacturer],
    LONG_NAME as [Aircraft Name Full],
    SHORT_NAME as [Aircraft Name Abbreviated],
    BEGIN_DATE as [Aircraft Begin Date],
    END_DATE as [Aircraft End Date]
FROM
[..\Data Files\CSVs\Aircraft_Base_File.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);
```

Step 5: Now we are moving through our next table which is “Aicraft_2010_Update.csv” which gives the information about the updates of 2010 year and later.



Name	Date modified	Type	Size
Aircraft_2010_Update	06-Oct-12 6:14 PM	Microsoft Office E...	1 KB
Aircraft_Base_File	06-Oct-12 6:14 PM	Microsoft Office E...	27 KB
Aircraft_Group	06-Oct-12 1:59 PM	Microsoft Office E...	1 KB
Aircraft_Manufacturers	06-Oct-12 4:21 PM	Microsoft Office E...	3 KB
Destination Markets	14-Jul-12 10:39 AM	Microsoft Office E...	47 KB
Origin Markets	14-Jul-12 10:39 AM	Microsoft Office E...	46 KB

Step 6: Now the table will be loaded > Finish



File Type
● Delimited
○ Fixed Record
○ Dif
○ Excel (xls)
○ Excel (xlsx)
○ Html
○ Qvd
○ Xml
○ Qvx

Delimiter: Semicolon
Quoting: MSQ
Header Size: None 0
Comment:
Character Set: Western European
Labels: Embedded Labels
□ Ignore EOF

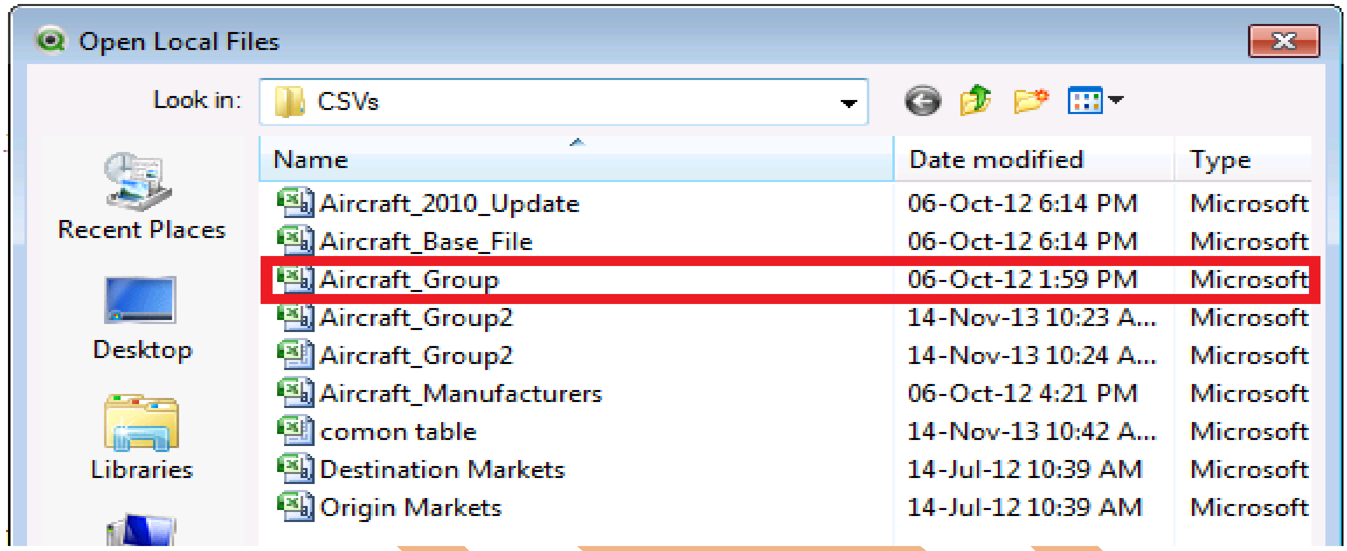
AC_TYPEID	AC_GROUPNAME	SSD_NAME
686	Jet, 2-Engine	Bombardier Ljet45
685	Jet, 2-Engine	Cessna 510/560X1
821	Jet, 4-Engine/6-Engine	Boeing B747-8F
609	Jet, 2-Engine	Challenger 300
393	Helicopter/Stol	Agustawestland 139
431	Turbo-Prop, 1-Engine/2-Engine	Socata Tbm850
340	Helicopter/Stol	Eurocompter As350
422	Turbo-Prop, 1-Engine/2-Engine	Quest Kodiak 100

Step 7: Then Rename the columns of the table “Aircraft_2010_Update” using ‘as’ keyword as shown here.

```
[Aircraft Types 2010]:
LOAD AC_TYPEID as [%Aircraft Type ID],
    AC_GROUPNAME,
    SSD_NAME as [Aircraft Name],
    MANUFACTURER as [Aircraft Manufacturer],
    LONG_NAME as [Aircraft Name Full],
    SHORT_NAME as [Aircraft Name Abbreviated],
    BEGIN_DATE as [Aircraft Begin Date],
    END_DATE as [Aircraft End Date]
FROM
```

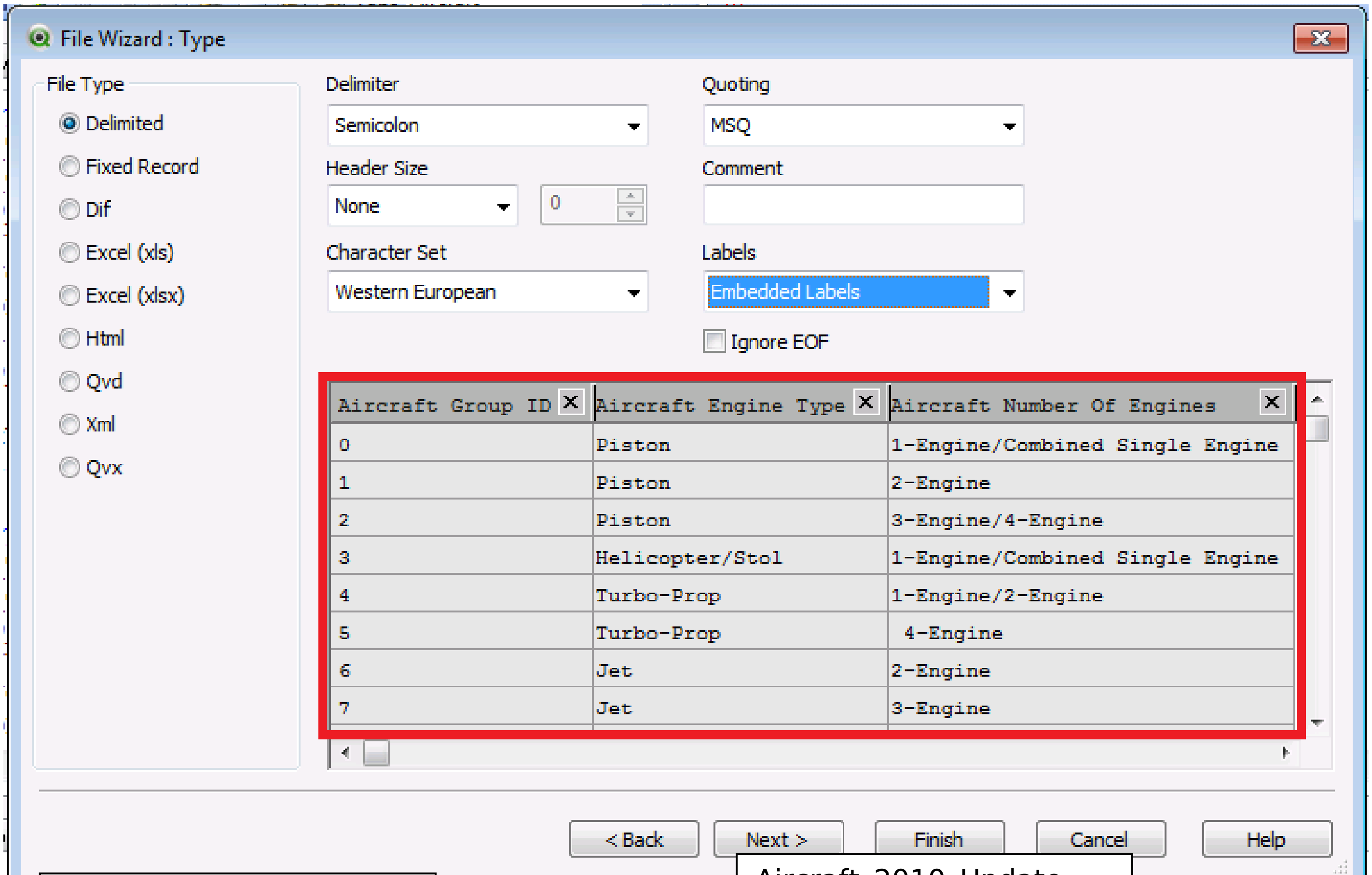
[..\Data Files\CSVs\Aircraft_2010_Update.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);

Step 8: Then the Table Aircraft Group will be loaded in the QlikView application. This Field contains the information about Aircraft Group ID ,Aircraft Engine type and Aircraft number of Engine. For this browse the location and open the particular file.



Name	Date modified	Type
Aircraft_2010_Update	06-Oct-12 6:14 PM	Microsoft
Aircraft_Base_File	06-Oct-12 6:14 PM	Microsoft
Aircraft_Group	06-Oct-12 1:59 PM	Microsoft
Aircraft_Group2	14-Nov-13 10:23 A...	Microsoft
Aircraft_Group2	14-Nov-13 10:24 A...	Microsoft
Aircraft_Manufacturers	06-Oct-12 4:21 PM	Microsoft
comon table	14-Nov-13 10:42 A...	Microsoft
Destination Markets	14-Jul-12 10:39 AM	Microsoft
Origin Markets	14-Jul-12 10:39 AM	Microsoft

Step 9:Now the data will be loaded in application-> Finish



File Type

- ● Delimited
- ○ Fixed Record
- ○ Dif
- ○ Excel (xls)
- ○ Excel (xlsx)
- ○ Html
- ○ Qvd
- ○ Xml
- ○ Qvx

Delimiter: Semicolon

Quoting: MSQ

Header Size: None 0

Comment:

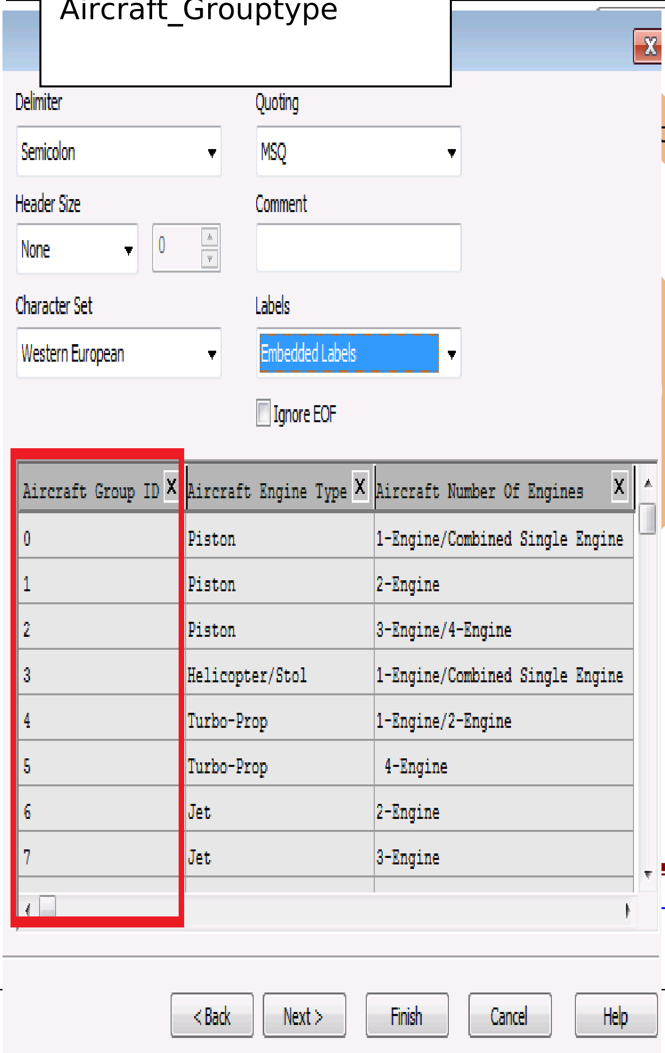
Character Set: Western European

Labels: Embedded Labels

□ Ignore EOF

Aircraft Group ID	Aircraft Engine Type	Aircraft Number Of Engines
0	Piston	1-Engine/Combined Single Engine
1	Piston	2-Engine
2	Piston	3-Engine/4-Engine
3	Helicopter/Stol	1-Engine/Combined Single Engine
4	Turbo-Prop	1-Engine/2-Engine
5	Turbo-Prop	4-Engine
6	Jet	2-Engine
7	Jet	3-Engine

Aircraft_Grouptype



Delimiter: Semicolon

Quoting: MSQ

Header Size: None 0

Comment:

Character Set: Western European

Labels: Embedded Labels

□ Ignore EOF

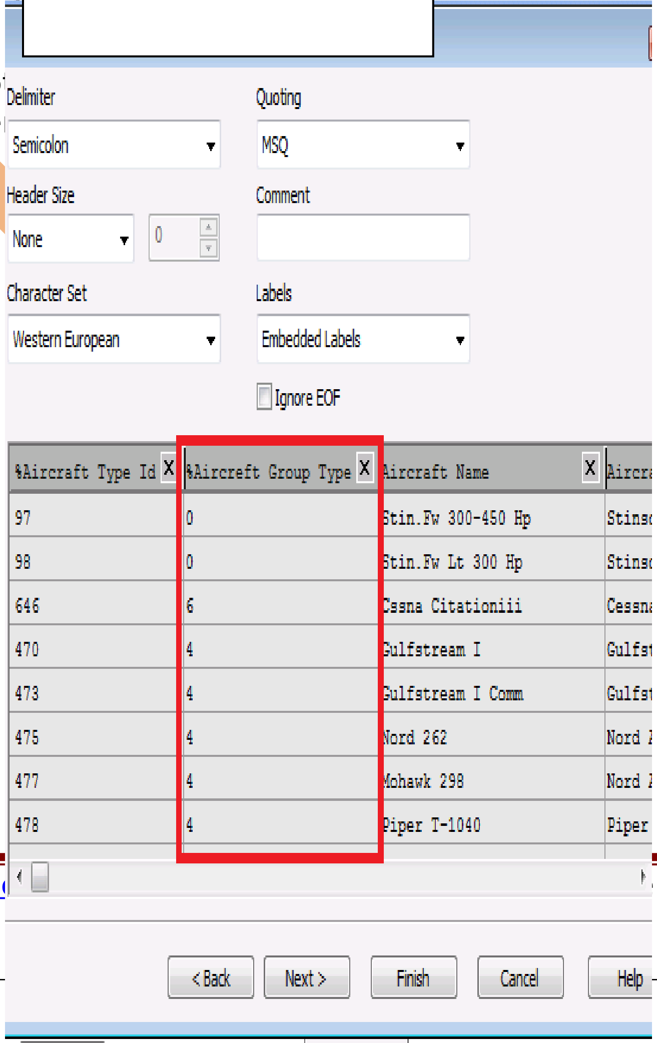
Aircraft Group ID	Aircraft Engine Type	Aircraft Number Of Engines
0	Piston	1-Engine/Combined Single Engine
1	Piston	2-Engine
2	Piston	3-Engine/4-Engine
3	Helicopter/Stol	1-Engine/Combined Single Engine
4	Turbo-Prop	1-Engine/2-Engine
5	Turbo-Prop	4-Engine
6	Jet	2-Engine
7	Jet	3-Engine

scrip

t diffe

inigs.(

Aircraft_2010_Update



Delimiter: Semicolon

Quoting: MSQ

Header Size: None 0

Comment:

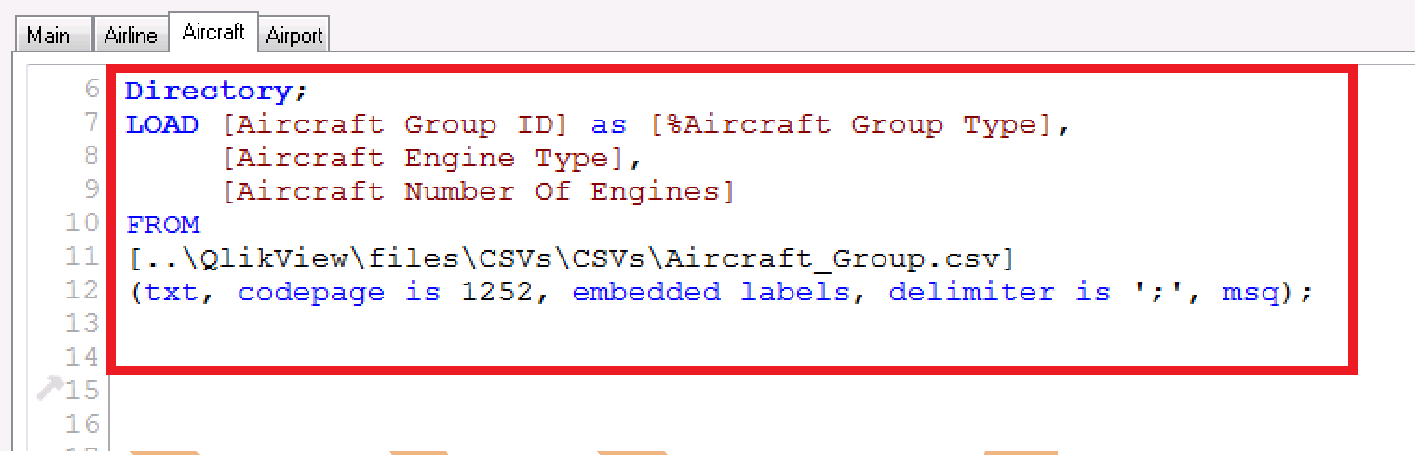
Character Set: Western European

Labels: Embedded Labels

□ Ignore EOF

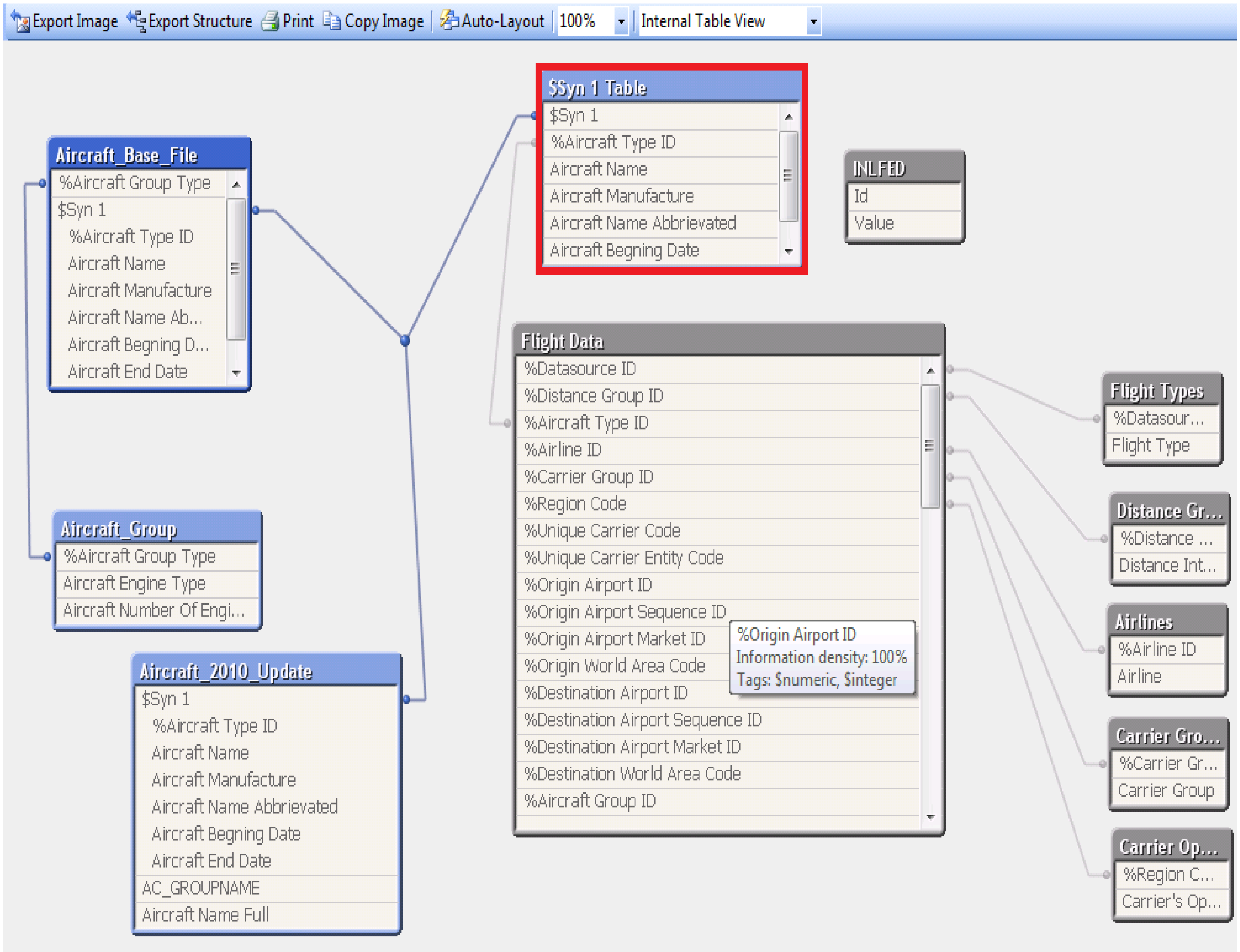
%Aircraft Type Id	%Aircreft Group Type	Aircraft Name	Aircra
97	0	Stin.Fw 300-450 Hp	Stins
98	0	Stin.Fw Lt 300 Hp	Stins
646	6	Cssna Citationiii	Cessna
470	4	Gulfstream I	Gulfst
473	4	Gulfstream I Comm	Gulfst
475	4	Nord 262	Nord
477	4	Mohawk 298	Nord
478	4	Piper T-1040	Piper

So, it's very illogical to avoid this as it will have large no of records. We should reduce our data as much as possible. So that we can easily manage our data and can reduce the extra statement using 'as' keyword which can be applied in the both table. And joining these 2 tables with this column / field name.



```
Directory;
LOAD [Aircraft Group ID] as [%Aircraft Group Type],
     [Aircraft Engine Type],
     [Aircraft Number Of Engines]
FROM
[..\QlikView\files\CSVs\CSVs\Aircraft_Group.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);
```

we used as Aircraft Group Id in %Aircraft Group Type these both considered as a single column. Now loaded the data. Have a look at the table viewer.

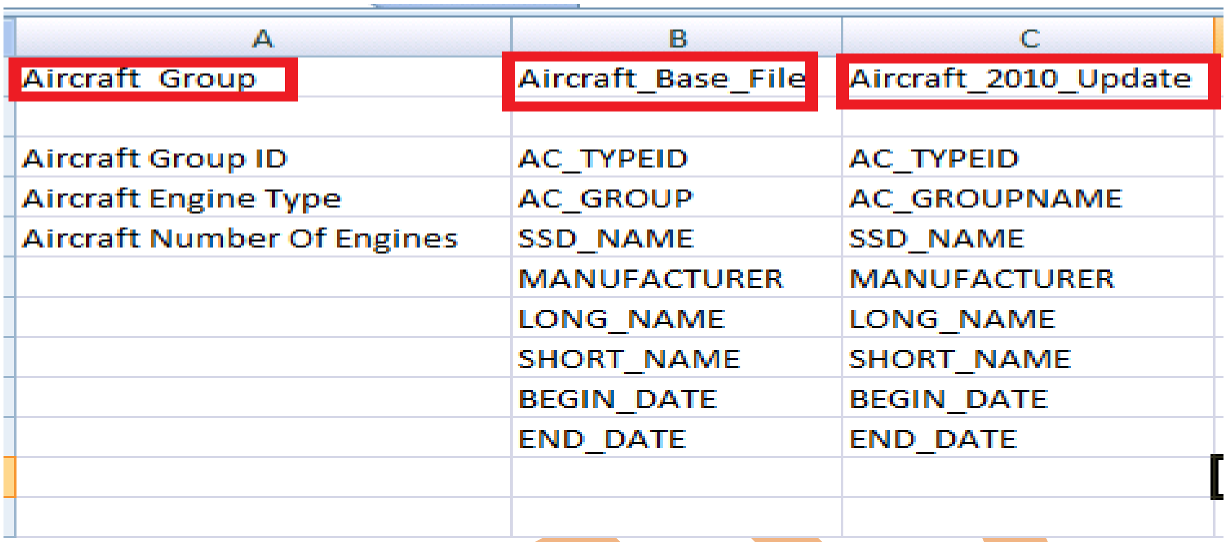


Look here the field $syn1 Table which we have not import but exist in the table viewer window. This is the Synthetic key which is auto-matically generated.

4)Synthetic key issue

Synthetic table and Synthetic keys gets created when there are more than one common fields between the 2 or more tables. QlikView join tables based on the common fields names in the table and if there is more than one common field then QlikView creates Synthetic tables. Synthetic keys are not beneficial in data model because they are resource intensive and may slow down calculations. They also make data model hard to understand. Synthetic keys should always be resolved.

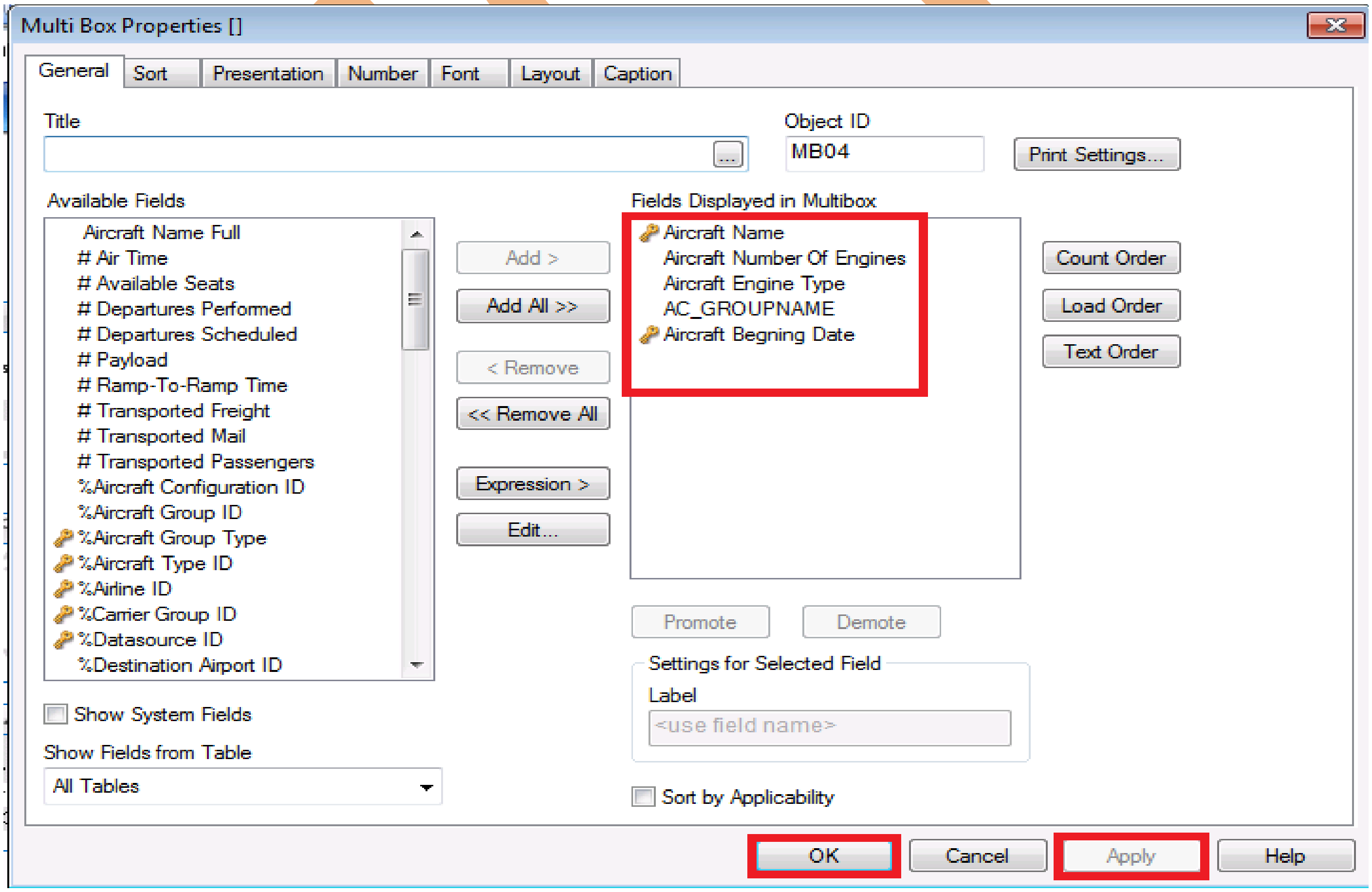
Look the three tables which we have loaded in the table having many fields common as we see Aircraft_Base_File having almost fields similar to Aircarft_2010_Update since in table field and these two contain one field similar to the table Aircraft Group is Aircraft Group ID.



A	B	C
Aircraft_Group	Aircraft_Base_File	Aircraft_2010_Update
Aircraft Group ID	AC_TYPEID	AC_TYPEID
Aircraft Engine Type	AC_GROUP	AC_GROUPNAME
Aircraft Number Of Engines	SSD_NAME	SSD_NAME
	MANUFACTURER	MANUFACTURER
	LONG_NAME	LONG_NAME
	SHORT_NAME	SHORT_NAME
	BEGIN_DATE	BEGIN_DATE
	END_DATE	END_DATE

The problem generated due to synthetic key

For looking the problem create a multibox ->right click on the sheet ->go to new sheet objects ->multibox.



Multi Box Properties []

Title:

Object ID: MB04

Available Fields:

- Aircraft Name Full
- # Air Time
- # Available Seats
- # Departures Performed
- # Departures Scheduled
- # Payload
- # Ramp-To-Ramp Time
- # Transported Freight
- # Transported Mail
- # Transported Passengers
- %Aircraft Configuration ID
- %Aircraft Group ID
- %Aircraft Group Type
- %Aircraft Type ID
- %Airline ID
- %Carrier Group ID
- %Datasource ID
- %Destination Airport ID

Fields Displayed in Multibox:

- Aircraft Name
- Aircraft Number Of Engines
- Aircraft Engine Type
- AC_GROUPNAME
- Aircraft Begning Date

□ Show System Fields

Show Fields from Table: All Tables

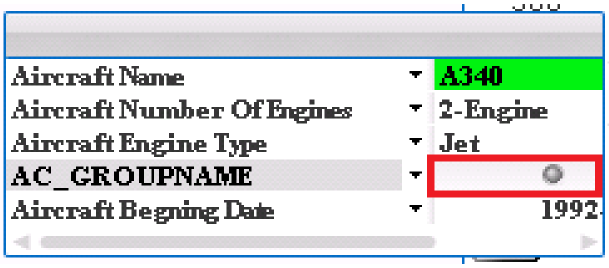
Settings for Selected Field

Label: <use field name>

□ Sort by Applicability

Now five fields are selected those are Aircraft Name,Aircraft number of engine,Aircraft engine type

Aircraft beging date,AC_GROUPNAME.now->Apply->ok.



Aircraft Name	A340
Aircraft Number Of Engines	2-Engine
Aircraft Engine Type	Jet
AC_GROUPNAME	●
Aircraft Begning Date	1992

look here the data on AC_GROUPNAME field disappeared due to generation of synthetic keys here we have two ways to resolve this issue.

⇒ Using concatenate statement

⇒ Using joins

6)Concatenate statement

The CONCATENATE statement is also a prefix to the LOAD statement, but instead of matching and merging rows between tables, this statement appends the rows of one table to another table.

Example Look here we have table Table1 and Table 2

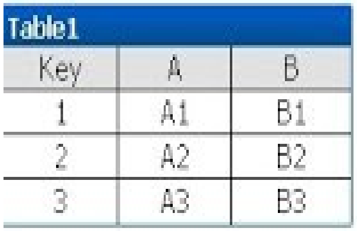


Table1

Key	A	B
1	A1	B1
2	A2	B2
3	A3	B3

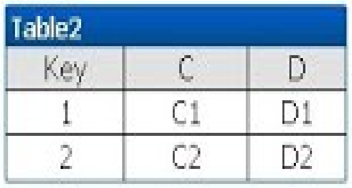
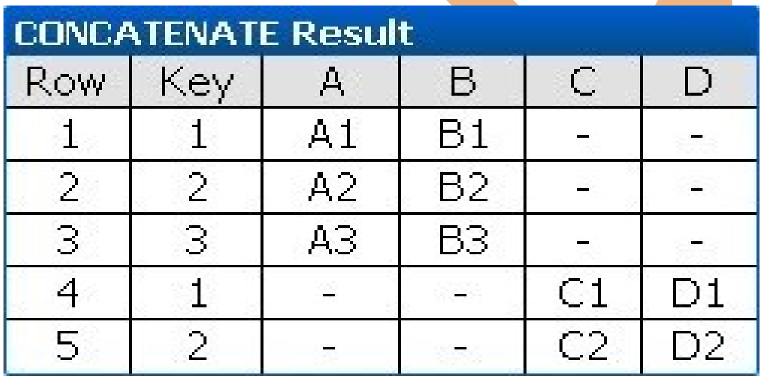


Table2

Key	C	D
1	C1	D1
2	C2	D2

Concatenate appends the rows of one table to another. Concatenate never merges any rows. The number of rows in a concatenated table is always the sum of the rows from the two input tables.

CONCATENATE (Table1) LOAD * RESIDENT Table2;



CONCATENATE Result

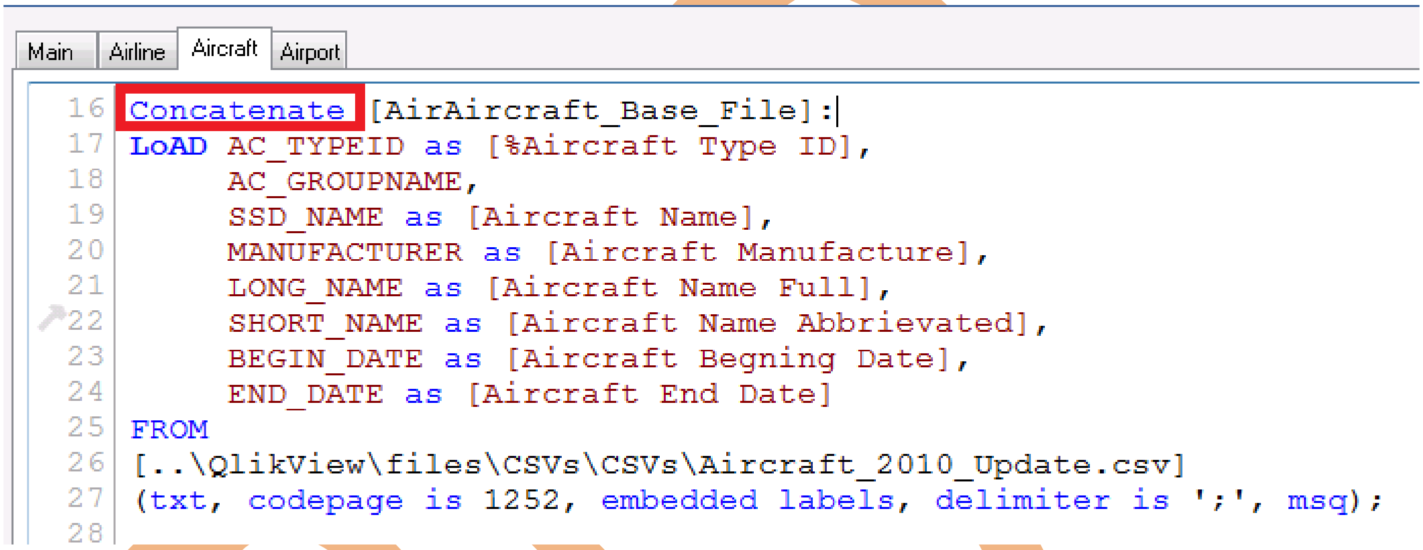
Row	Key	A	B	C	D
1	1	A1	B1	-	-
2	2	A2	B2	-	-
3	3	A3	B3	-	-
4	1	-	-	C1	D1
5	2	-	-	C2	D2

Rows are like Key values and are not merged together. The rows from Table2 are simply appended to Table1. Because the tables have different fields, rows will have null values for the fields from the "other" table.

Here we want the whole information in this the table Aircraft_2010_Update contain information of 2010 and Aircraft_Base_File information before 2010 we want to perform

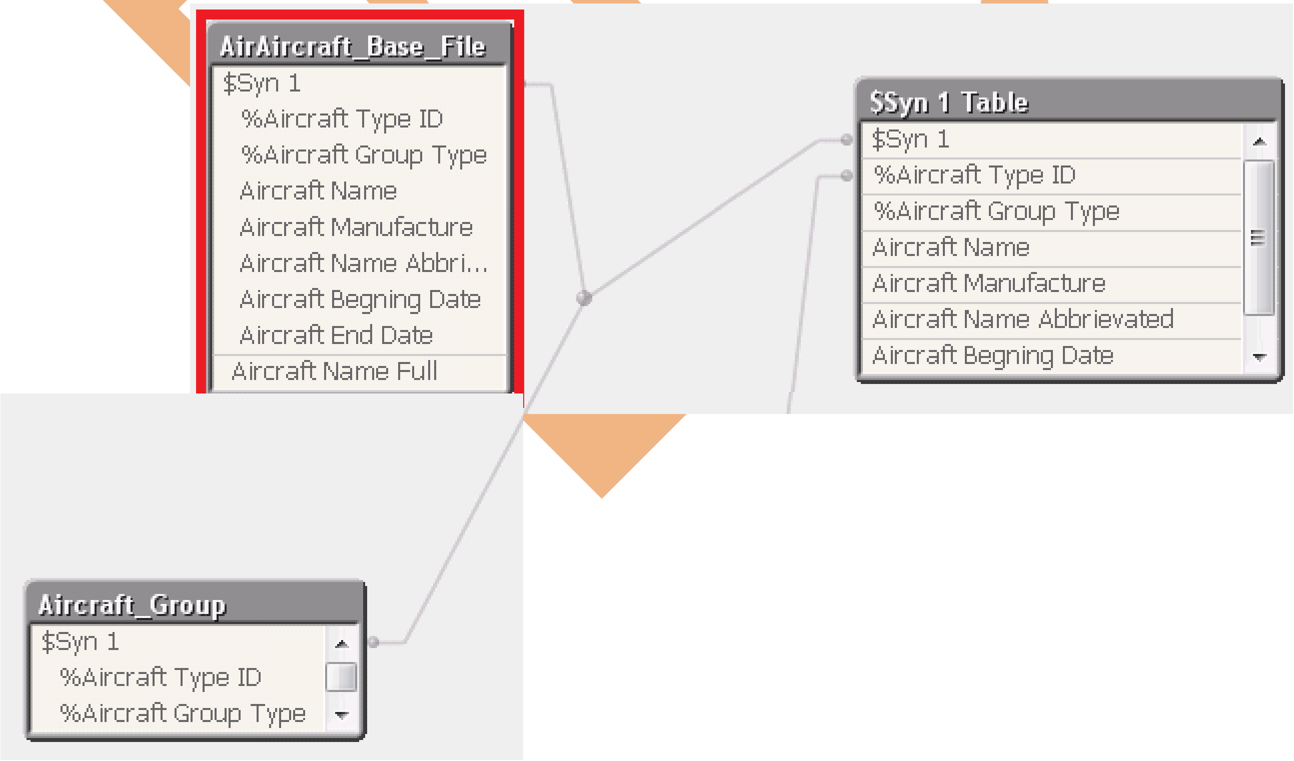
Aircaft_Base_File= Aircraft_Base_File(previous)+ Aircraft_2010_Update

we want that the data of 2010 append to the base file.



```
Concatenate [AirAircraft_Base_File]:
LoAD AC_TYPEID as [%Aircraft Type ID],
     AC_GROUPNAME,
     SSD_NAME as [Aircraft Name],
     MANUFACTURER as [Aircraft Manufacture],
     LONG_NAME as [Aircraft Name Full],
     SHORT_NAME as [Aircraft Name Abbrievated],
     BEGIN_DATE as [Aircraft Begning Date],
     END_DATE as [Aircraft End Date]
FROM
[..\QlikView\files\CSVs\CSVs\Aircraft_2010_Update.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);
```

->Reload the table->go to the table viewer



From the table viewer window the table Aircraft_2010_Update disappear and concatenates into the table Aircraft_Base_File.

7) Join statement

The JOIN statement is a prefix to the LOAD statement. It is used to join the table that is being loaded to a previously loaded table. The two tables are joined using a natural join, this means that the columns in both tables are compared and the join is made over those columns that have the same column names. This means that if multiple columns are shared between tables, the match will be made over the distinct combinations of those columns.

Example we have 2 table Table1 and Table 2

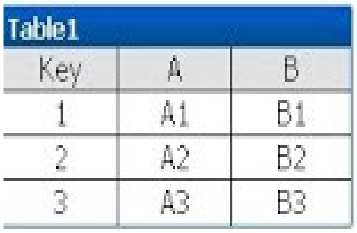


Table1

Key	A	B
1	A1	B1
2	A2	B2
3	A3	B3

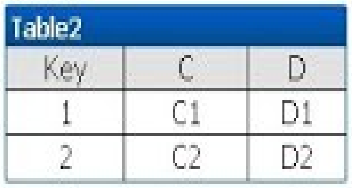
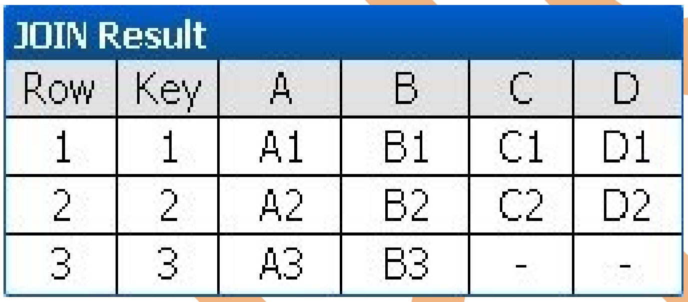


Table2

Key	C	D
1	C1	D1
2	C2	D2

Now the join will result

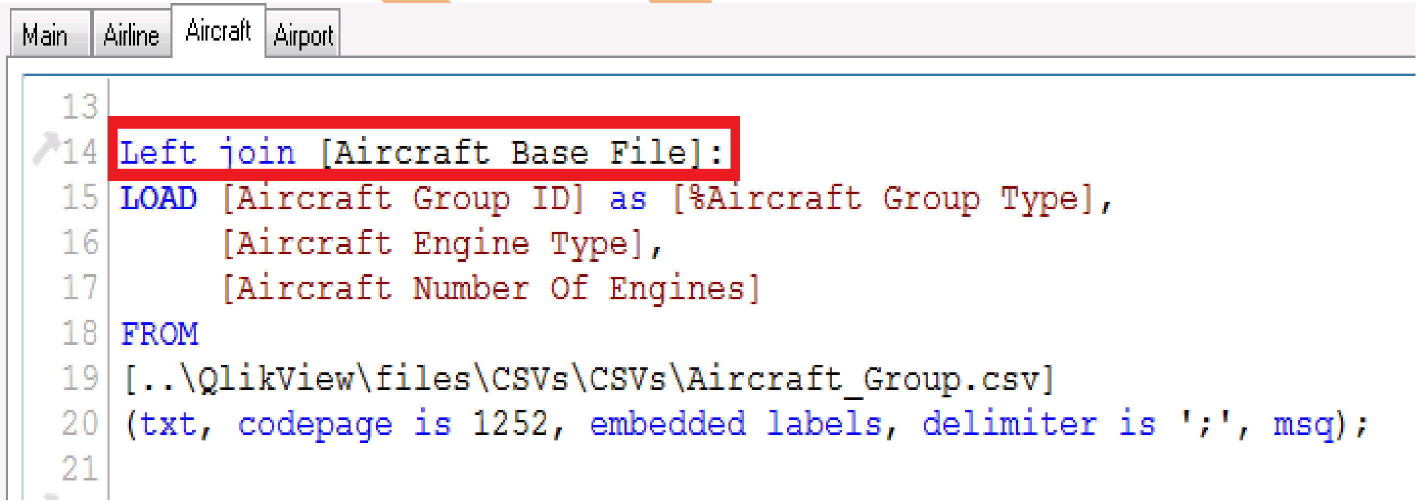
CONCATENATE (Table1) LOAD * RESIDENT Table2;



JOIN Result

Row	Key	A	B	C	D
1	1	A1	B1	C1	D1
2	2	A2	B2	C2	D2
3	3	A3	B3	-	-

Here we have two tables as Aircraft_group and Aircraft_Base_File in this we have previously seen that the data in the 2 fields as Aircraft Group ID and %Aircraft Group Type is same so we want to merge this row of the table Aircraft_group



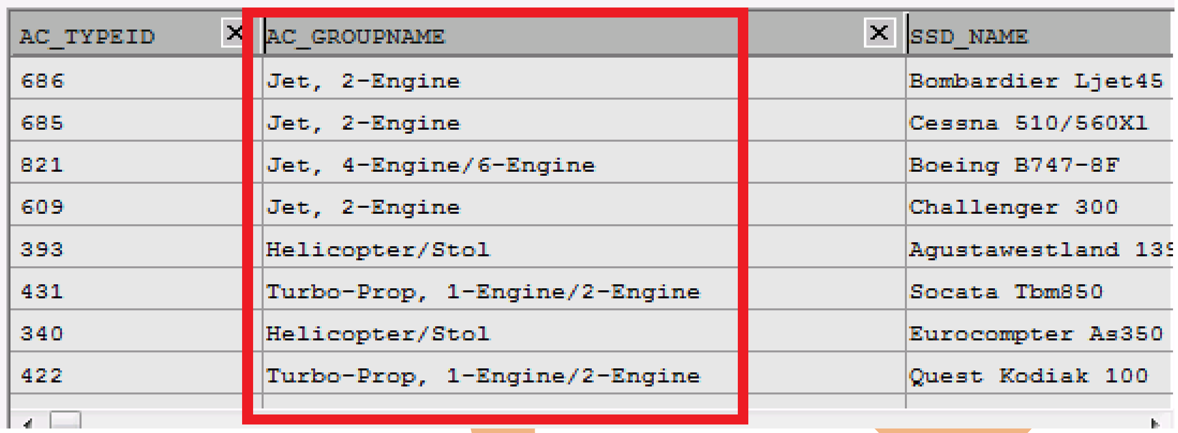
```

Left join [Aircraft Base File]:
LOAD [Aircraft Group ID] as [%Aircraft Group Type],
     [Aircraft Engine Type],
     [Aircraft Number Of Engines]
FROM
[..\QlikView\files\CSVs\CSVs\Aircraft_Group.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);

```

This will merge the row in the table Aircraft_base_File.

8)Subfield Function

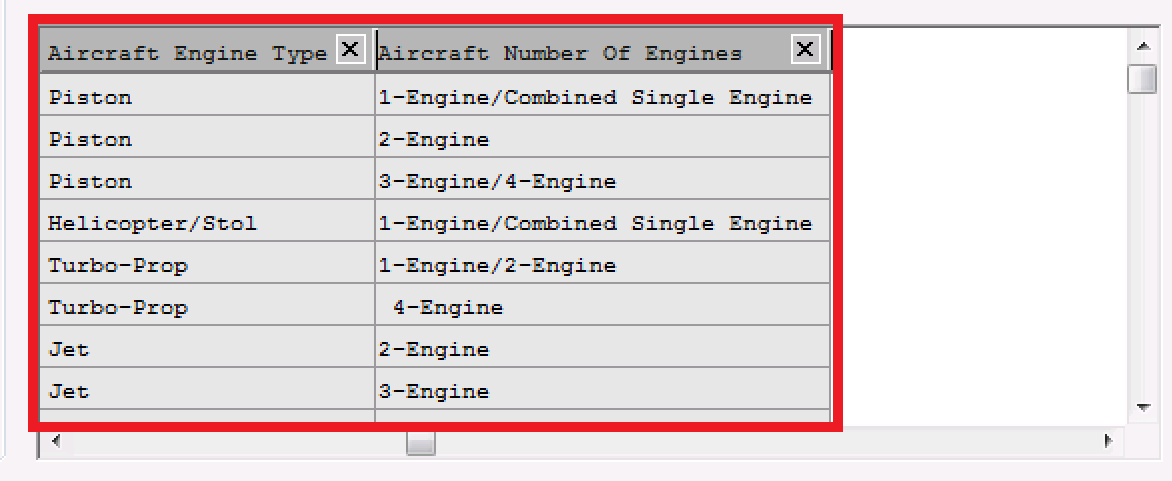
Subfield function are used for splitting the field via subfield function we can split two fields within a single field.

Look the field AC_GROUPNAME of Aircraft base table



AC_TYPEID	AC_GROUPNAME	SSD_NAME
686	Jet, 2-Engine	Bombardier Ljet45
685	Jet, 2-Engine	Cessna 510/560Xl
821	Jet, 4-Engine/6-Engine	Boeing B747-8F
609	Jet, 2-Engine	Challenger 300
393	Helicopter/Stol	Agustawestland 13
431	Turbo-Prop, 1-Engine/2-Engine	Socata Tbm850
340	Helicopter/Stol	Eurocompter As350
422	Turbo-Prop, 1-Engine/2-Engine	Quest Kodiak 100

And these two fields of table Aircraft_Group



Aircraft Engine Type	Aircraft Number Of Engines
Piston	1-Engine/Combined Single Engine
Piston	2-Engine
Piston	3-Engine/4-Engine
Helicopter/Stol	1-Engine/Combined Single Engine
Turbo-Prop	1-Engine/2-Engine
Turbo-Prop	4-Engine
Jet	2-Engine
Jet	3-Engine

Here the field AC_GROUPNAME from table Aircraft_Base_File covers the information of Aircraft Engine Type and Aircraft Number OF Engines of table Aircraft Group individually so we can split these two cploumns within a single table.

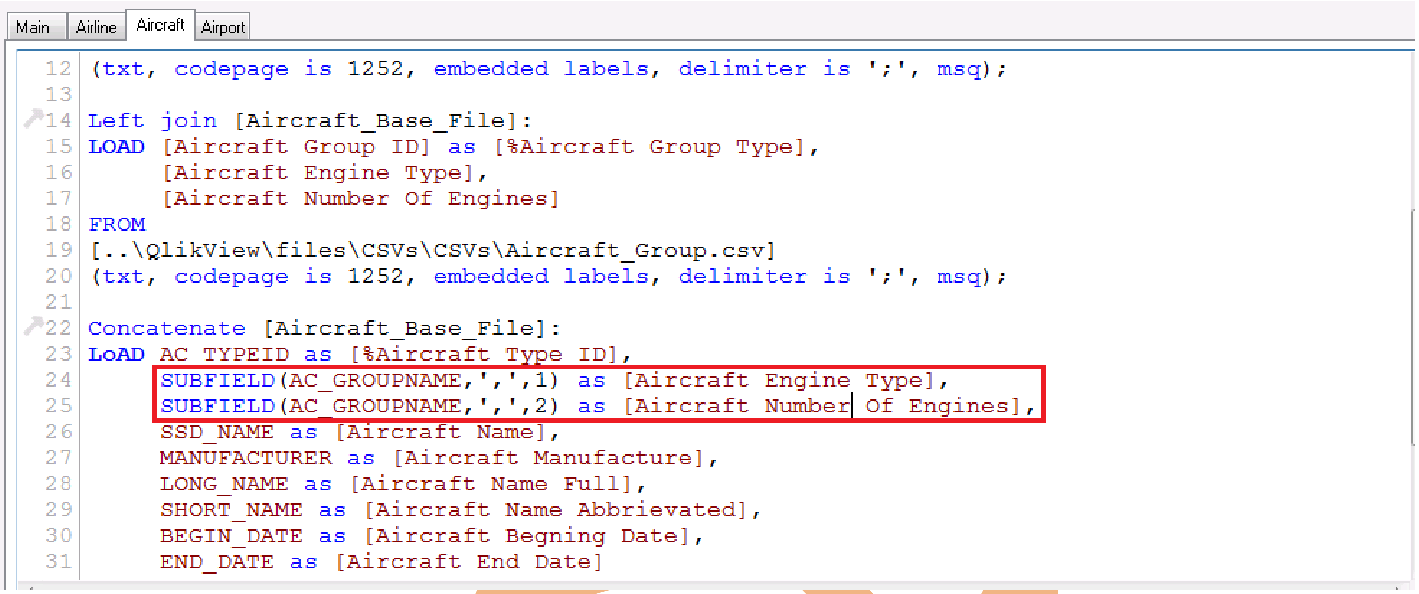
AC_GROUPNAME = Aircraft Engine Type+Aircraft Number OF Engines

This function is very case sensitive, in field AC_GROUPNAME information separated by comma(,) in syntax as we are loading the .csv file which is a comma separated file.

SUBFIELD(AC_GROUPNAME,',''1) as Aircraft Engine Type

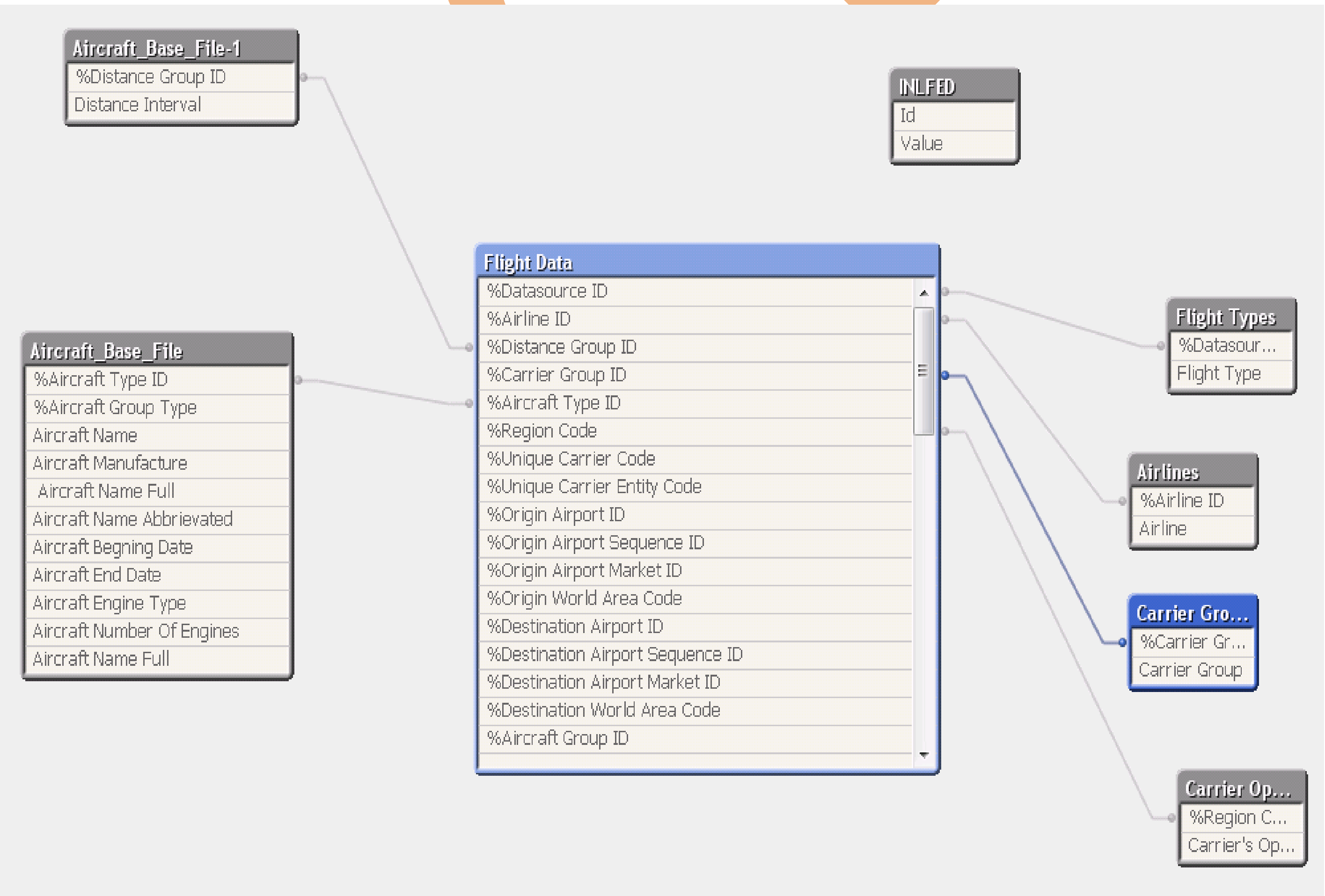
SUBFIELD(AC_GROUPNAME,',''2) as Aircraft Number OF Engines

are numbered as 1 and 2 to identify the fields which are going to get split.



```
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);

Left join [Aircraft_Base_File]:
LOAD [Aircraft Group ID] as [%Aircraft Group Type],
     [Aircraft Engine Type],
     [Aircraft Number Of Engines]
FROM
[..\QlikView\files\CSVs\CSVs\Aircraft_Group.csv]
(txt, codepage is 1252, embedded labels, delimiter is ';', msq);

Concatenate [Aircraft_Base_File]:
LoAD AC_TYPEID as [%Aircraft Type ID],
     SUBFIELD(AC_GROUPNAME,',',1) as [Aircraft Engine Type],
     SUBFIELD(AC_GROUPNAME,',',2) as [Aircraft Number Of Engines],
     SSD_NAME as [Aircraft Name],
     MANUFACTURER as [Aircraft Manufacture],
     LONG_NAME as [Aircraft Name Full],
     SHORT_NAME as [Aircraft Name Abbrievated],
     BEGIN_DATE as [Aircraft Begning Date],
     END_DATE as [Aircraft End Date]
```

Now reload the table and go to the table viewer window. Look here the synthetic key resolved and data managed in formatted way.


Aircraft_Base_File-1
%Distance Group ID
Distance Interval
INLFED
Id
Value
Flight Data
%Datasource ID
%Airline ID
%Distance Group ID
%Carrier Group ID
%Aircraft Type ID
%Region Code
%Unique Carrier Code
%Unique Carrier Entity Code
%Origin Airport ID
%Origin Airport Sequence ID
%Origin Airport Market ID
%Origin World Area Code
%Destination Airport ID
%Destination Airport Sequence ID
%Destination Airport Market ID
%Destination World Area Code
%Aircraft Group ID
Aircraft_Base_File
%Aircraft Type ID
%Aircraft Group Type
Aircraft Name
Aircraft Manufacture
Aircraft Name Full
Aircraft Name Abbrievated
Aircraft Begning Date
Aircraft End Date
Aircraft Engine Type
Aircraft Number Of Engines
Aircraft Name Full
Flight Types
%Datasour...
Flight Type
Airlines
%Airline ID
Airline
Carrier Gro...
%Carrier Gr...
Carrier Group
Carrier Op...
%Region C...
Carrier's Op...