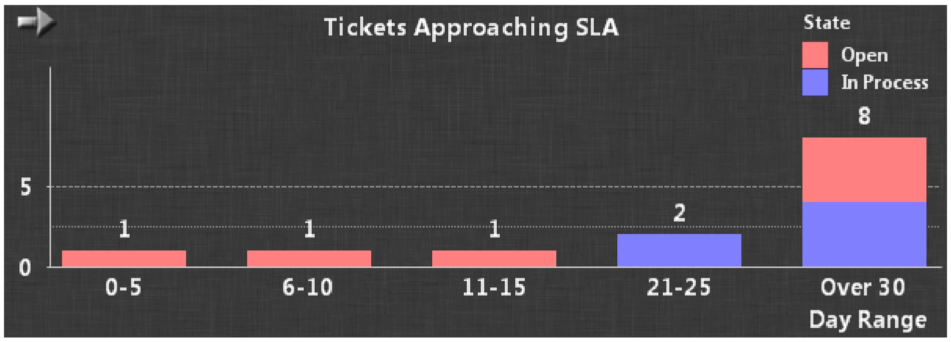
INTERVAL MATCH

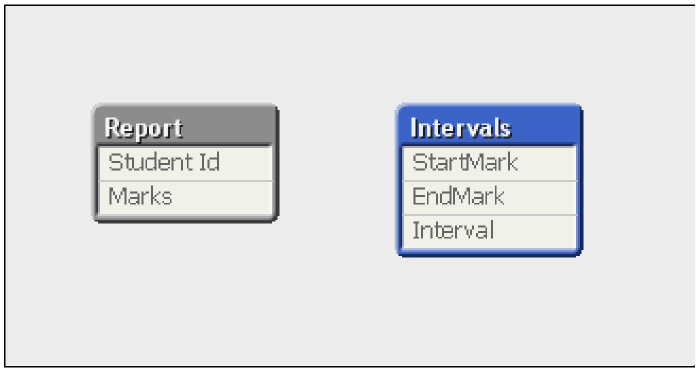
Why do we need intervalmatch() ?
When we want to link a number to a range in sql, we use "between" keyword. In qlikview there is no function as between. Here we use the interval match function for the same.

Client Requirement:
The project was related to ticketing system.
We had two fields, Ticket Request Number and SLA(in days). The requirement was to get the count of tickets in different ranges of approaching SLA days. As you can see below, the source is in table form and the chart o/p was the requirement.

Request No.	SLA
R1	39
R2	39
R3	22
R4	22
R5	12
R6	191
R7	191
R8	129
R9	8
R10	191
R11	146
R12	39
R13	3



What is intervalmatch() ?
It is a System function in Qlikview used for linking a number to a range.
For example, if we have student marks on one hand and four intervals on the other, then using IntervalMatch() we can find out number of students in each interval.



How to use IntervalMatch()?

Steps to perform Interval Match:

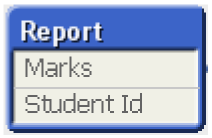
Here, I am taking an excel source having two fields - Student Id and marks. The excel sheet named intervalmatch.xlsx is stored in local D: drive inside kshop folder.

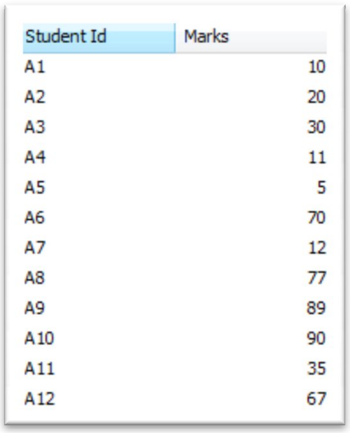
1. Load the excel into qlikview

   Following is the code which needs to be written in script editor (Ctrl + E).

```
Report:
LOAD[Student Id],
Marks
FROM
D:\Kshop\intervalmatch.xlsx
(ooxml, embedded labels, table is Sheet1);
```

   Reload after typing in the code.
   After successful reload, following "Report" table is created.





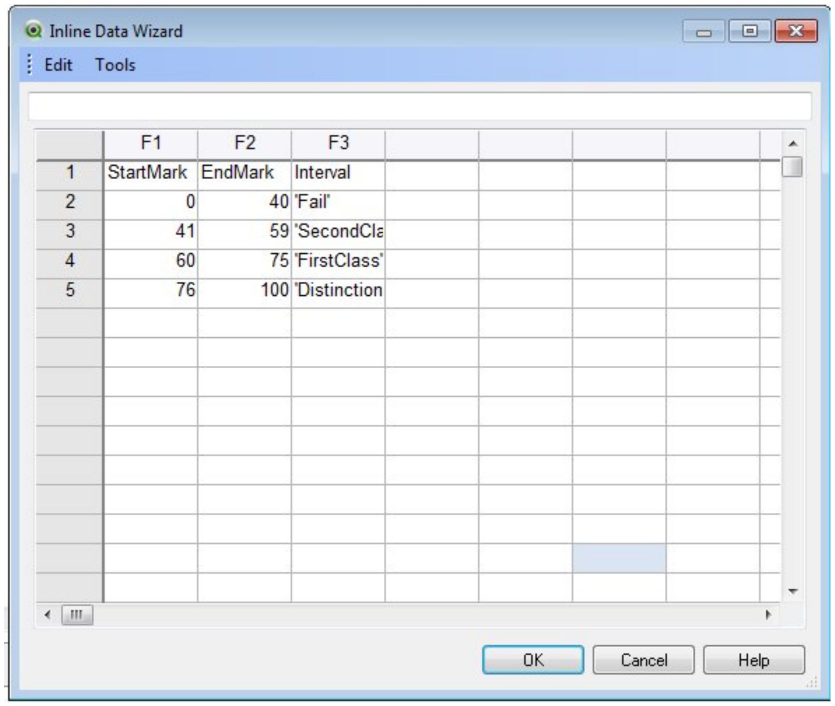
Student Id	Marks
A1	10
A2	20
A3	30
A4	11
A5	5
A6	70
A7	12
A8	77
A9	89
A10	90
A11	35
A12	67

2. Create interval table manually using Inline table concept

   Inline Table: To create a table manually in qlikview, we go for inline table.
   Edit Script (Ctrl+E) -> Insert (menu) -> Load Statement -> inline table

   Insert column names and data manually as shown in the image below.

   The intervals table should have three fields
   - First one for start of the Interval
   - Second one for End of the interval
   - Third one for the interval name



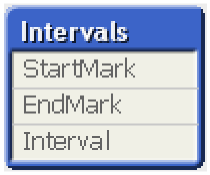
	F1	F2	F3
1	StartMark	EndMark	Interval
2	0	40	'Fail'
3	41	59	'SecondCla
4	60	75	'FirstClass'
5	76	100	'Distinction

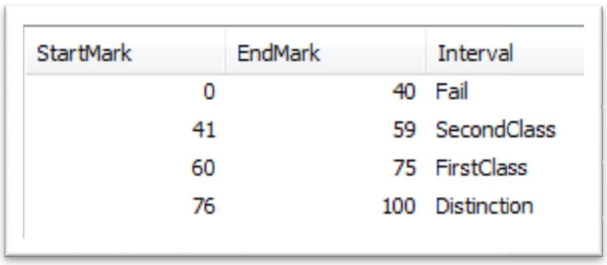
The following code comes up in the script automatically.

```
Load * inline
[F1,F2,F3
StartMark,EndMark,Interval
0,40,'Fail'
41,59,'SecondClass'
60,75,'FirstClass'
76,100,'Distinction'
];
```

Modify the above code as per the need. Here, I have given table name and removed default column names.

```
Intervals:
Load * inline
[
StartMark,EndMark,Interval
0,40,'Fail'
41,59,'SecondClass'
60,75,'FirstClass'
76,100,'Distinction'
];
```





StartMark	EndMark	Interval
0	40	Fail
41	59	SecondClass
60	75	FirstClass
76	100	Distinction

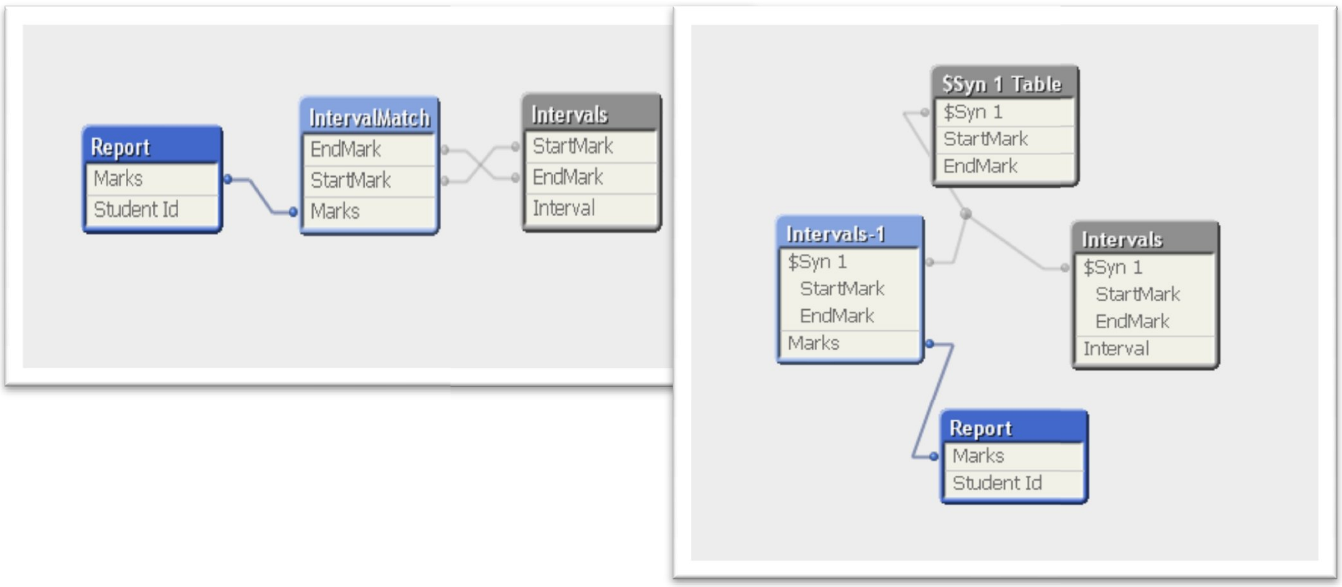
3. Load the intervalmatch table

This table acts as the mediator between the two tables.
IntervalMatch function can be put in front of a Load or a Select statement.
Inside the parenthesis, the field should be the one on which we are performing interval Match.

Here, Marks is the field on which we are performing the interval match.
Following is the code which has to be written in the script.

```
IntervalMatch:
IntervalMatch(Marks)
LoadStartMark,EndMark
Resident Intervals;
```

After loading these three tables the data model would look like below:



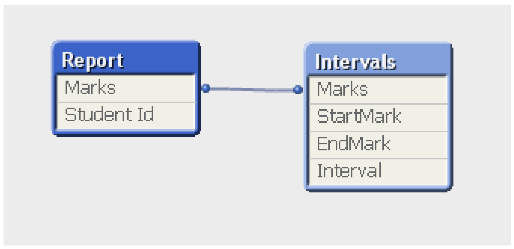
4. Remove the Synthetic key

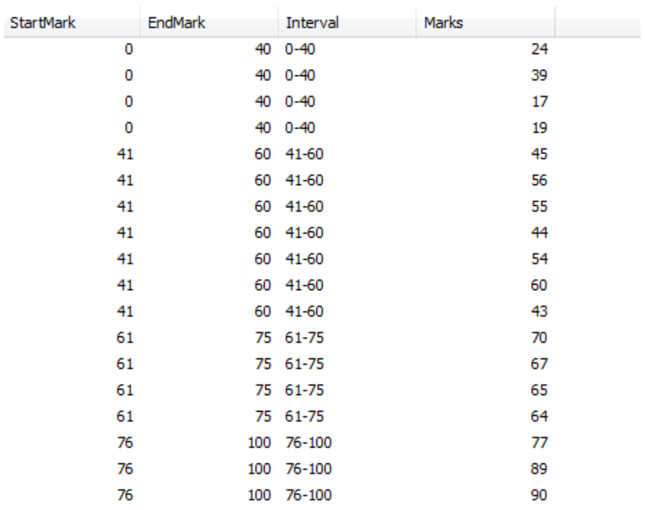
Synthetic keys :
When two or more input tables have two or more fields in common, thisimplies a composite key relationship. QlikView handles this through synthetickeys. These keys are anonymous fields that represent all occurringcombinations of the composite key. When the number of composite keysincreases, depending on data amounts, table structure and other factors,QlikView may or may not handle them gracefully. QlikView may end upusing excessive amount of time and/or memory. Unfortunately the actuallimitations are virtually impossible to predict, which leaves only trial anderror as a practical method to determine them.

By doing a simple Left join between intervals table and interval match synthetic key can be removed.
Here, left join is done between Intervals table and IntervalMatch table.

```
Intervals :
Load * inline
[
StartMark,EndMark,Interval
0,40,'Fail'
41,60,'SecondClass'
61,75,'FirstClass'
76,100,'Distiction'
];

IntervalMatch:

LeftJoin(Intervals)
IntervalMatch(Marks)

LoadStartMark,EndMark
Resident Intervals;
```

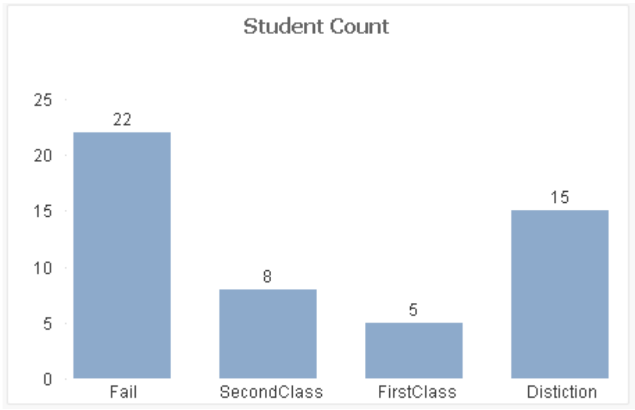
Now the data diagram looks as below:





StartMark	EndMark	Interval	Marks
0	40	0-40	24
0	40	0-40	39
0	40	0-40	17
0	40	0-40	19
41	60	41-60	45
41	60	41-60	56
41	60	41-60	55
41	60	41-60	44
41	60	41-60	54
41	60	41-60	60
41	60	41-60	43
61	75	61-75	70
61	75	61-75	67
61	75	61-75	65
61	75	61-75	64
76	100	76-100	77
76	100	76-100	89
76	100	76-100	90

As we can see from the above image, we have one more column “interval” in the intervals table.
We can use above table to create the following chart.
In chart properties (right click on chart -> properties)
Dimension tab - > Interval
Expression tab - > Add Expression ->Count([Student Id]



Alternatives forInterval Match

- Class function
- Nested if

Class Function:

Syntax: Class(expression, interval [, label [, offset]])

Unlike Interval match Class function does not allow us to set the intervals. We need to mention the interval value in the expression. If we sat a value of 5, it creates intervals keeping the value 5 constant.

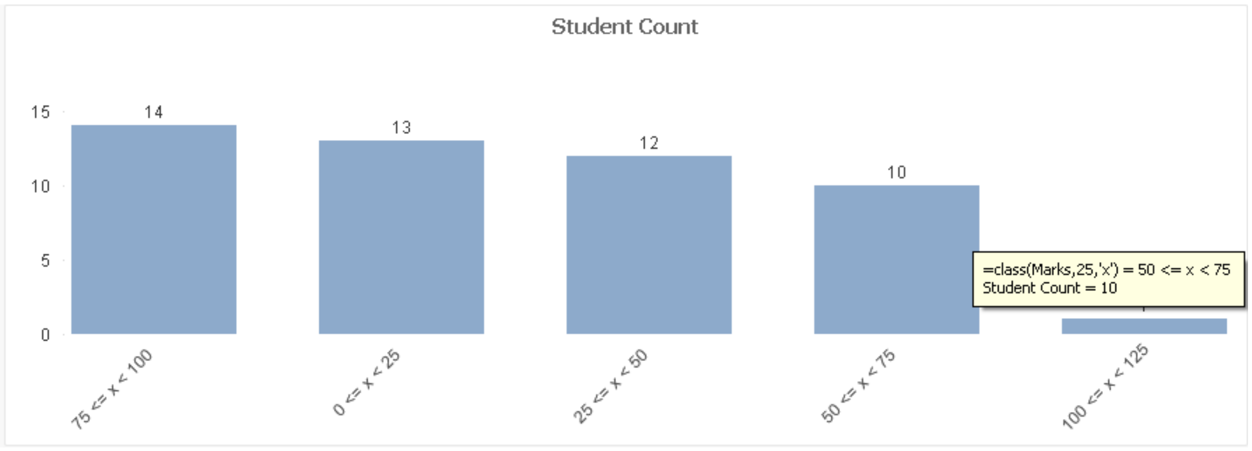
1. Load report table

```
Report:
LOAD[Student Id],
Marks
FROM
D:\Kshop\intervalmatch.xlsx
```

2. Create a chart with following Dimension and Expression

   Dimension - >Class(Marks,25)

   Expression - > Count([Student Id]



Nested If

This is done in two steps:

1. Create a table and name it as report

```
Report:
LOAD[Student Id],
Marks
FROM
D:\Kshop\intervalmatch.xlsx
```

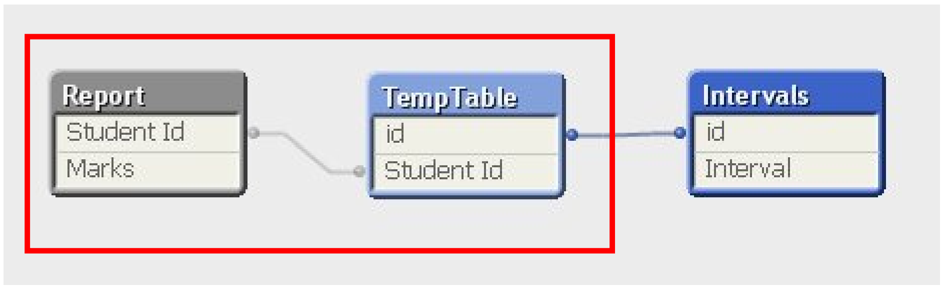
2. Create an inline table for intervals

```
Intervals :
Load * inline
[
id,Interval
1,'Fail'
2,'SecondClass'
3,'FirstClass'
4,'Distiction'
];
```

3. Create a temporary table using nested if fordefining id for each student which matches with the id in the Intervals table

```
TempTable:
Load
[Student Id] ,
if(Marks>=0 ANDMarks<=40,1,
if(Marks>=41 ANDMarks<=60,2,
if(Marks>=61 ANDMarks<=75,3,
if(Marks>=75 ANDMarks<=40,4,Marks
     ))))asid

resident Report;
```

The data diagrams looks as below:

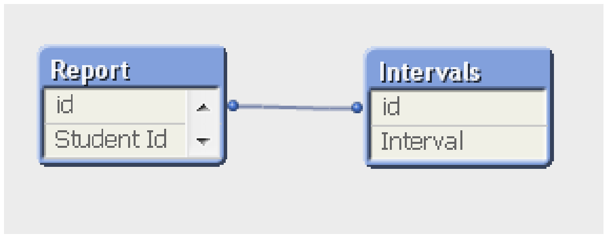


The two tables Report and TempTable can be joined to create single table. It can be done by adding one more line in the above code

The final code becomes

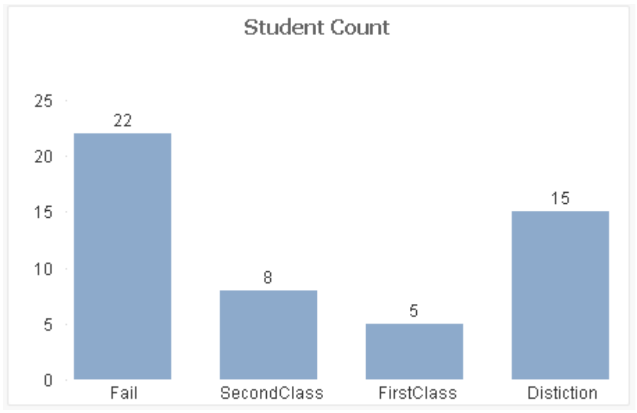
```
TempTable:
innerJoin(Report)
Load
[Student Id] ,
if(Marks>=0 ANDMarks<=40,1,
if(Marks>=41 ANDMarks<=60,2,
if(Marks>=61 ANDMarks<=75,3,
if(Marks>=75 ANDMarks<=100,4,Marks
     ))))asid

resident Report;
```

The final data diagrams looks as below:



We can also join the above two tables, it entirely depends how business wants it.

Using the above two tables we can create the following chart


Student Count
Fail
SecondClass
FirstClass
Distiction