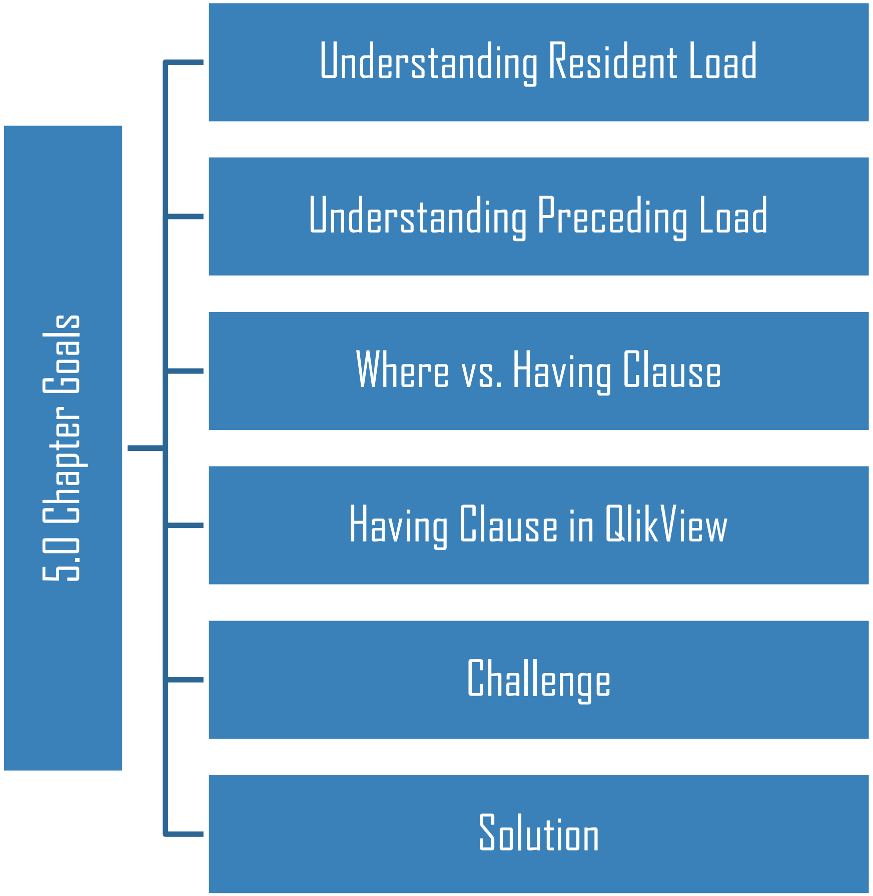
5. Resident Load vs Preceding Load

VERTECHS.IO
TEACH..LEARN..INSPIRE

5.0 Chapter Goals

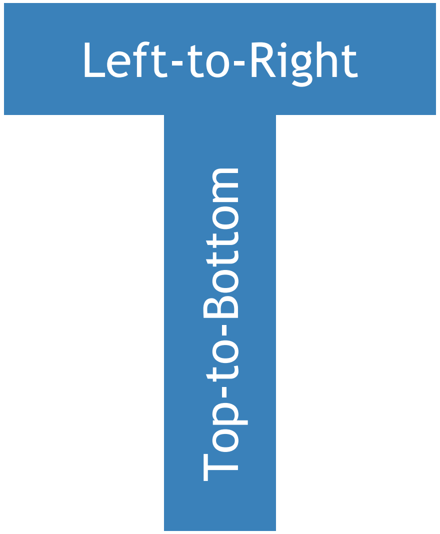


5.0 Chapter Goals

- Understanding Resident Load
- Understanding Preceding Load
- Where vs. Having Clause
- Having Clause in QlikView
- Challenge
- Solution

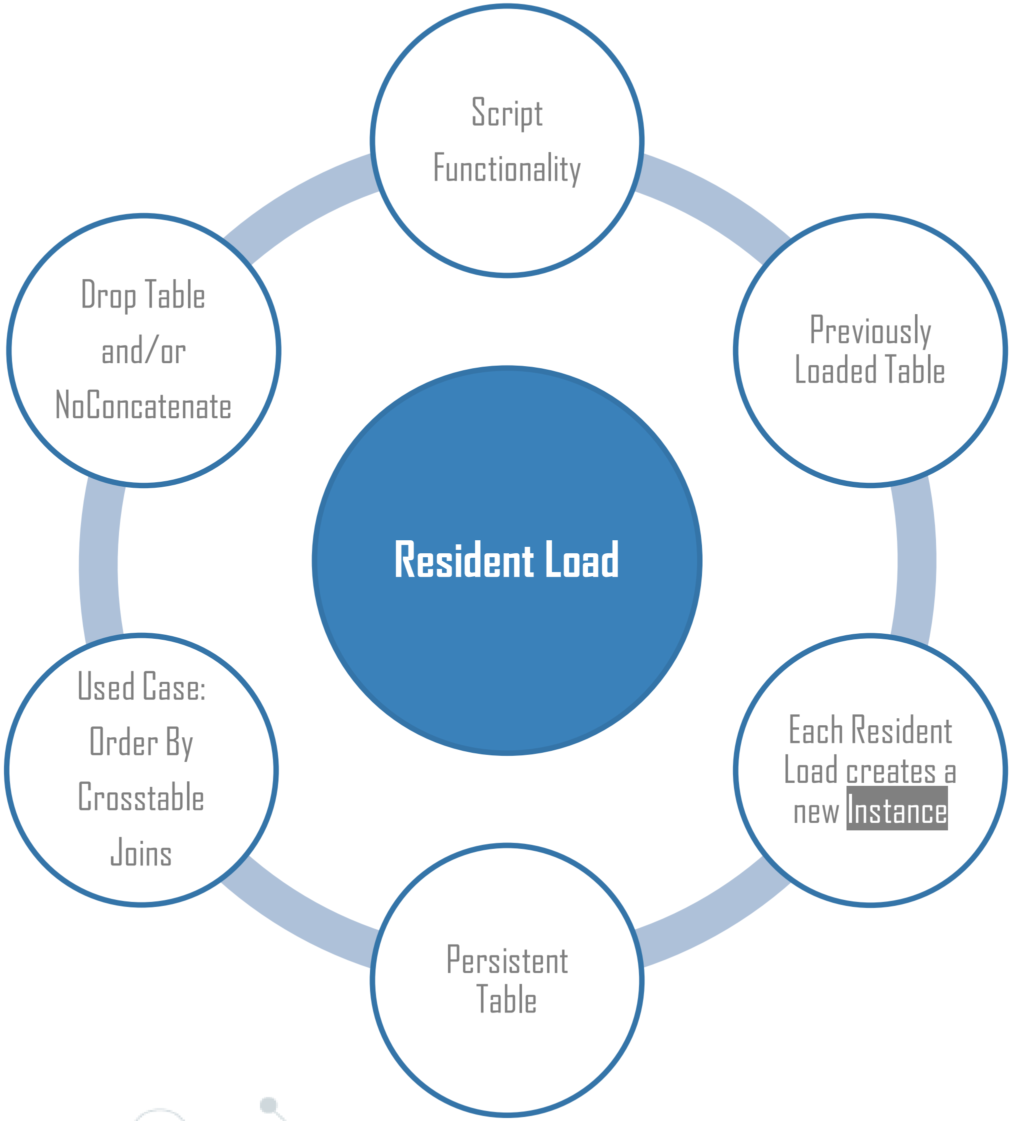
5.1: Understanding Script Execution

- T-Shape Execution:



- Statement execution and not line-by-line execution
- No logical division for tabs
- Preceding Load: Breaks the Top-to-Bottom rule

5.2: Understanding Resident Load



Resident Load

Script Functionality

Previously Loaded Table

Each Resident Load creates a new Instance

Persistent Table

Used Case: Order By Crosstable Joins

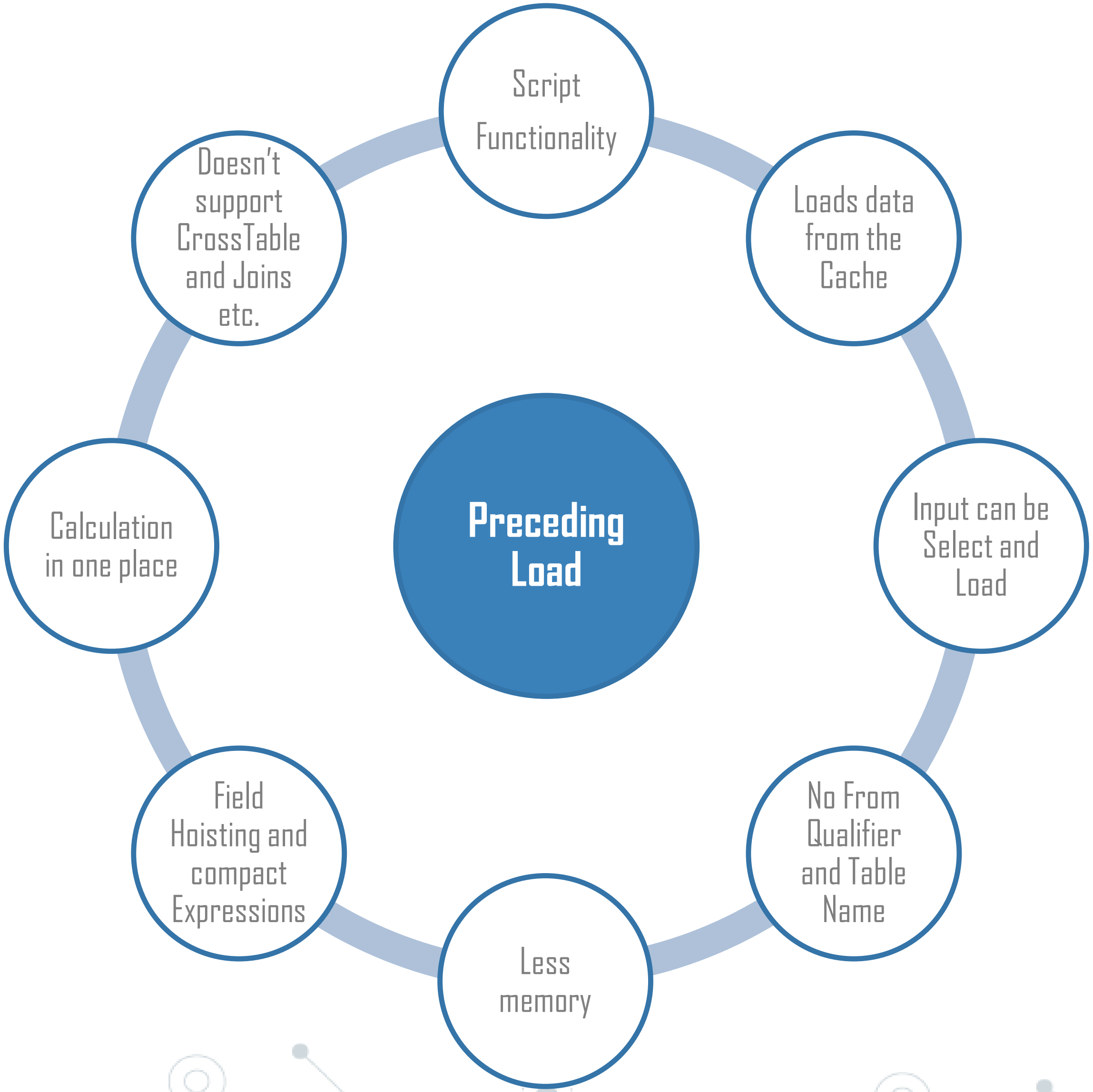
Drop Table and/or NoConcatenate

5.2: Understanding Resident Load

Demo - Resident Load table

- Order Table has incorrect Courier Website format
- Shipped field has {-1, 0} but users need to have {Yes, No} to filter & search

5.3: Understanding Preceding Load



Preceding Load

Script Functionality

Loads data from the Cache

Input can be Select and Load

No From Qualifier and Table Name

Less memory

Field Hoisting and compact Expressions

Calculation in one place

Doesn't support CrossTable and Joins etc.

5.5: Difference between WHERE and HAVING clause

Demo - WHERE vs. HAVING clause

- A WHERE clause is used is filter records from a result. The filter occurs before any groupings are made

```
SELECT   COUNT(SalesOrderID)
FROM     Sales.SalesOrderDetail
WHERE    UnitPrice > 200
```

- A HAVING clause is used to filter values from a group

```
SELECT   SalesOrderID,
SUM(UnitPrice * OrderQty) AS TotalPrice
FROM     Sales.SalesOrderDetail
GROUP BY SalesOrderID
HAVING   SalesOrderID > 50000
```

5.6: How to implement HAVING clause in QlikView

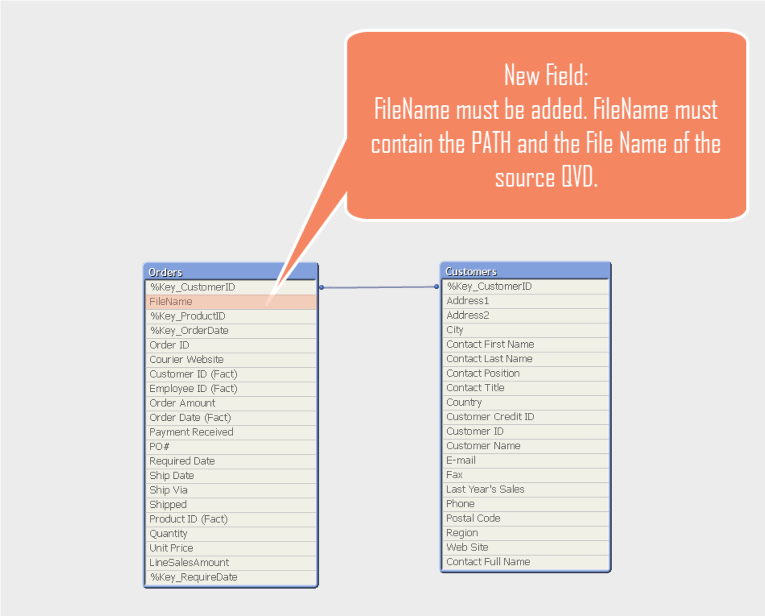
Demo - HAVING clause in QlikView

- There is NO HAVING clause in QlikView
- Preceding Load with WHERE clause can be used as HAVING clause

5.7: Challenge

Challenge:

5.7: You have been provided with five QVD files ***(Orders_YYYYMMDD)***, QVW file and the script. After reloading the script we see Synthetic Table with Synthetic Keys instead we wanted to perform auto concatenation of all the Order Tables. Please go ahead and correct the script which generates one Order table after reloading. Also, add a new field which shows the file name for each of the QVD files loaded.


New Field:
FileName must be added. FileName must contain the PATH and the File Name of the source QVD.

Rules:

- *You MUST use the wildcard load as shown in the script [Orders_2017*]*
- *You can't use forced concatenation on individual table*
- *You MUST create the %Key_RequireDate either in Preceding Load or Resident Load ONLY*

5.8: Solution - Resident Load vs Preceding Load

Solution: