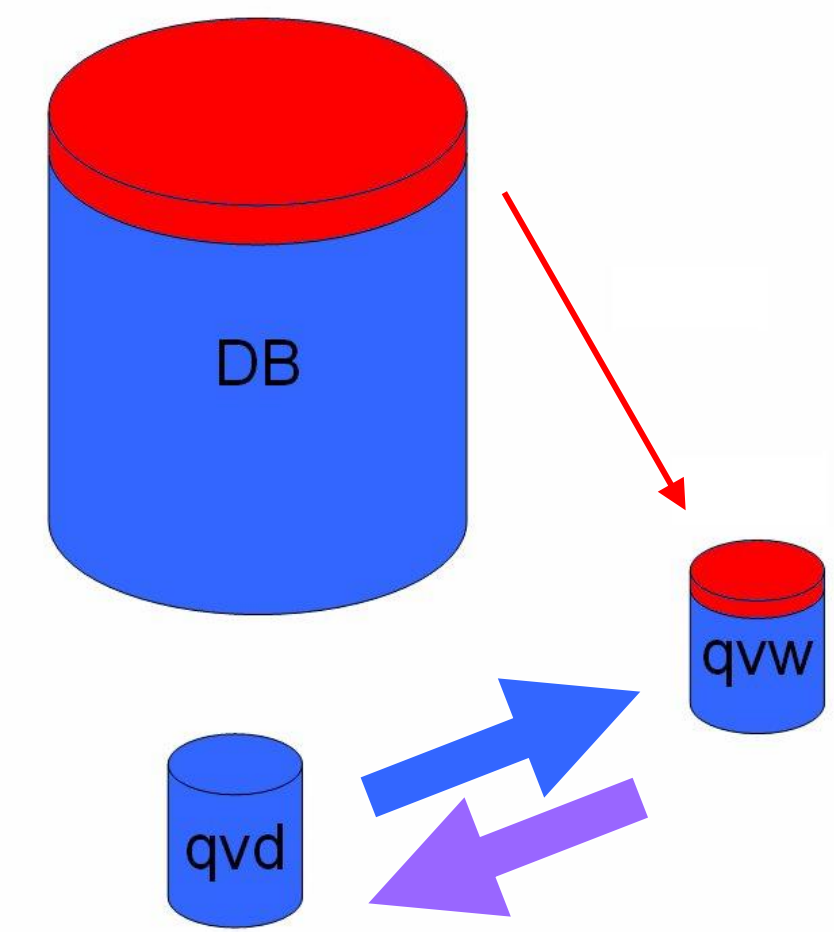
Incremental Load

using qvd files

Incremental Load

- Load new data from Database table (slow, but few records)
- Load old data from QVD file (many records, but fast)
- Create new QVD file
- Procedure must be repeated for each table



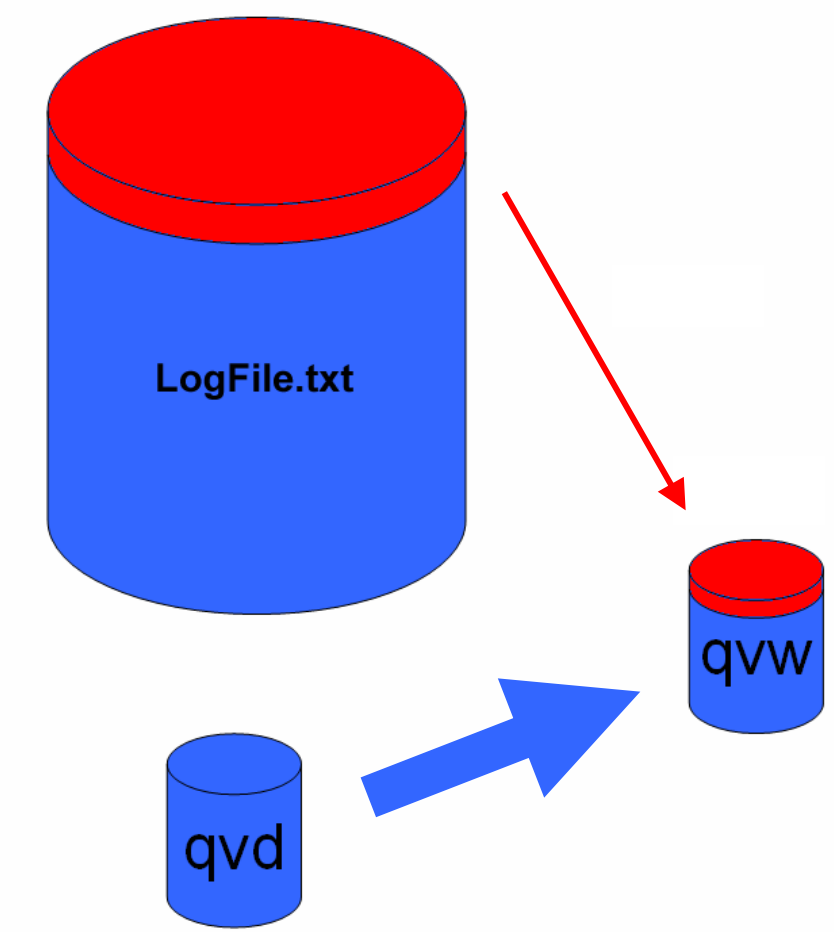
Different DB-changes

If source allows …

1) Append only. (Logfiles)
2) Insert only. (No Update or Delete)
3) Insert and Update. (No Delete)
4) Insert, Update and Delete.

1) Append only

- Must be Log file
- Loads records added in the end of the file

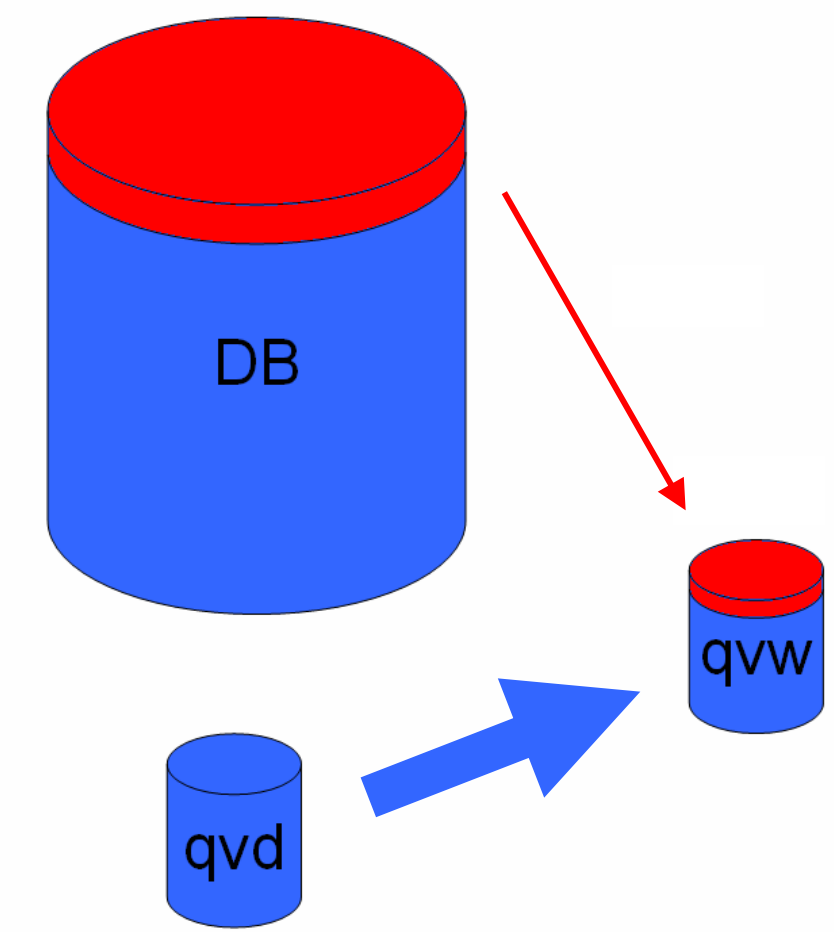


1) Append only

Buffer (Incremental)
Load * **From** LogFile.txt
(ansi, txt, delimiter is '\t', embedded labels);

2) Insert only

- Can be any DB
- Loads INSERTed records
- Needs the field *ModificationDate*



2) Insert only

QV_Table:
SQL SELECT PrimaryKey, X, Y **FROM** DB_TABLE
WHERE ModificationTime >= ***#$(LastExecTime)#***;

2) Insert only

QV_Table:
SQL SELECT PrimaryKey, X, Y **FROM** DB_TABLE
WHERE ModificationTime >= ***#$(LastExecTime)#***;

Concatenate
LOAD PrimaryKey, X, Y **FROM** File.QVD;

2) Insert only

QV_Table:
SQL SELECT PrimaryKey, X, Y **FROM** DB_TABLE
WHERE ModificationTime >= ***#$(LastExecTime)#***;

Concatenate
LOAD PrimaryKey, X, Y **FROM** File.QVD;

STORE QV_Table **INTO** File.QVD;

2) Insert only

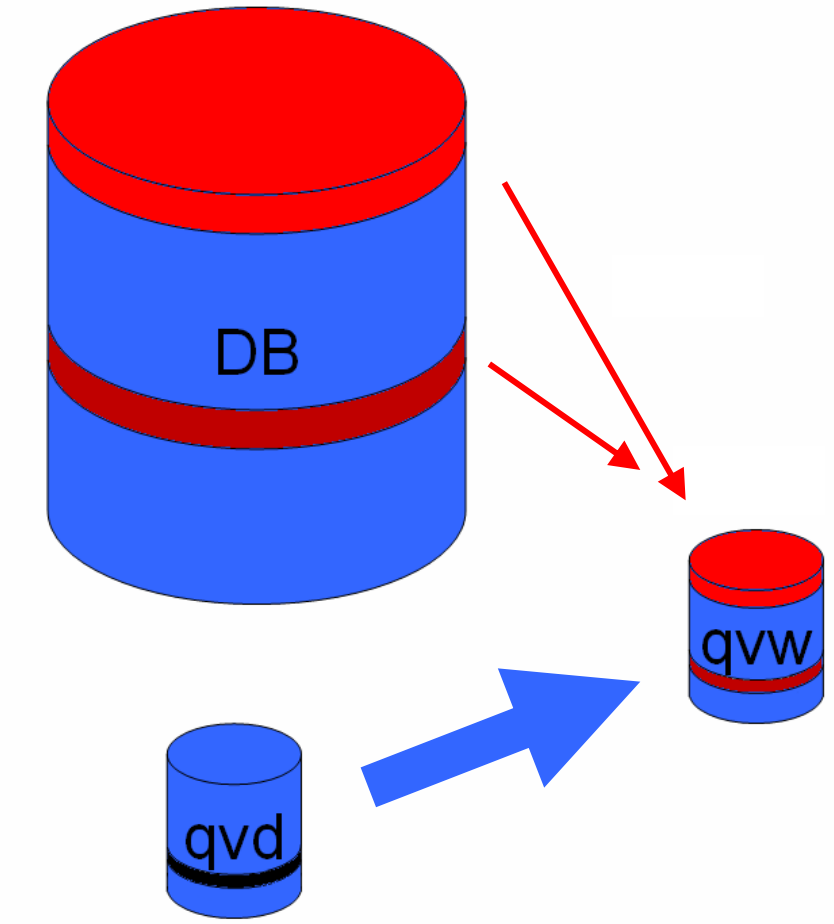
QV_Table:
SQL SELECT PrimaryKey, X, Y **FROM** DB_TABLE
WHERE ModificationTime >= ***#$(LastExecTime)#***
  AND ModificationTime < ***#$(BeginningThisExecTime)#***;

Concatenate
LOAD PrimaryKey, X, Y **FROM** File.QVD;

STORE QV_Table **INTO** File.QVD;

3) Insert and Update

- Can be any DB
- Loads INSERTed and UPDATEd records
- Needs the fields *ModificationDate* and *PrimaryKey*



3) Insert and Update

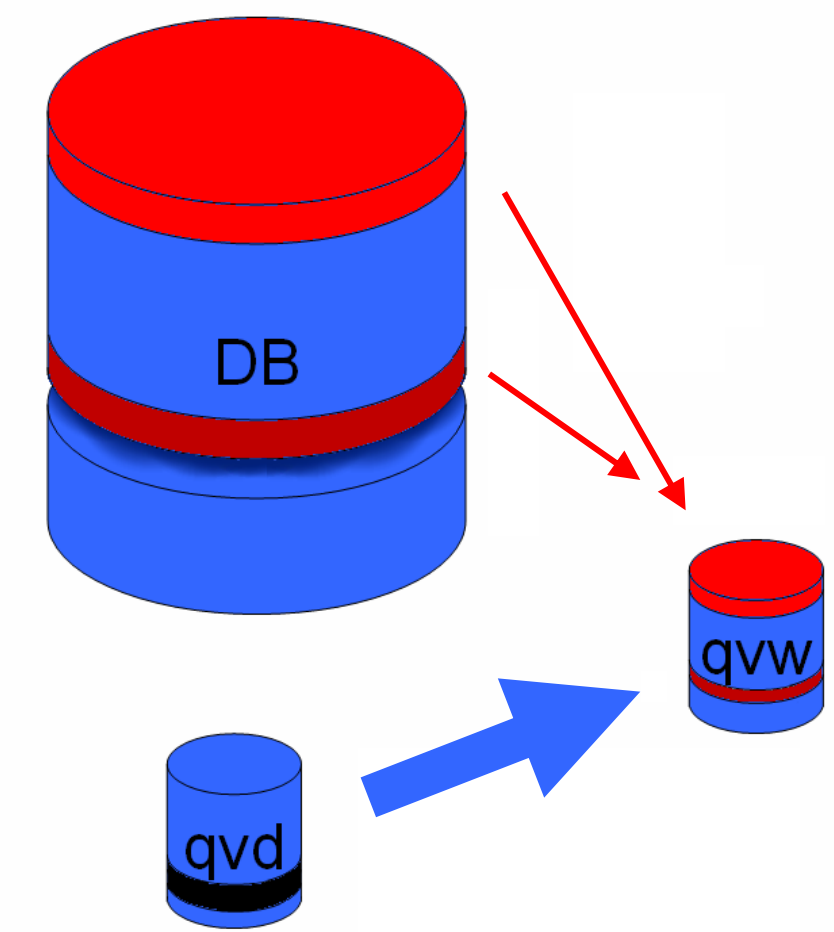
```
QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#;

Concatenate
LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT Exists(PrimaryKey);

STORE QV_Table INTO File.QVD;
```

4) Insert, Update and Delete

- Can be any DB
- Loads INSERTed and UPDATEd records
- Removes DELETEd records
- Needs the fields *ModificationDate* and *PrimaryKey*
- Tricky to implement



4) Insert, Update and Delete

QV_Table:
SQL SELECT PrimaryKey, X, Y **FROM** DB_TABLE
WHERE ModificationTime >= ***#$(LastExecTime)#***;

Concatenate
LOAD PrimaryKey, X, Y **FROM** File.QVD
WHERE NOT EXISTS(PrimaryKey);

Inner Join
SQL SELECT PrimaryKey **FROM** DB_TABLE;

STORE QV_Table **INTO** File.QVD;

4) Insert, Update and Delete

```
ListOfDeletedEntries:
SQL SELECT PrimaryKey AS Deleted FROM DB_TABLE
WHERE DeletionFlag = 1;

QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#;

Concatenate
LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT Exists(PrimaryKey)
  AND NOT Exists(Deleted,PrimaryKey);

Drop Table ListOfDeletedEntries;

STORE QV_Table INTO File.QVD;
```

LastExecutionTime & Error handling

```
Let ThisExecTime = ReloadTime();

{ Load sequence }

If ScriptErrorCount = 0 then
        Let LastExecTime = ThisExecTime;
End If
```

Final Script

```
Let ThisExecTime = ReloadTime();

QV_Table:
SQL SELECT PrimaryKey, X, Y FROM DB_TABLE
WHERE ModificationTime >= #$(LastExecTime)#
    AND ModificationTime < #$(ThisExecTime)#;

Concatenate LOAD PrimaryKey, X, Y FROM File.QVD
WHERE NOT EXISTS(PrimaryKey);

Inner Join SQL SELECT PrimaryKey FROM DB_TABLE;

If ScriptErrorCount = 0 then
        STORE QV_Table INTO File.QVD;
        Let LastExecTime = ThisExecTime;
End If
```