Understanding & Best Practices with the QVD File Format

We will begin at the top of the hour…

Rob Patterson
Senior Manager, User Enablement
QlikTech

Rob Wunderlich
Senior QlikView Consultant
Factory Information Management Solutions

Factory

Information Management Solutions

QlikLearn

Understanding & Best Practices with the QVD File Format

What?

Why?

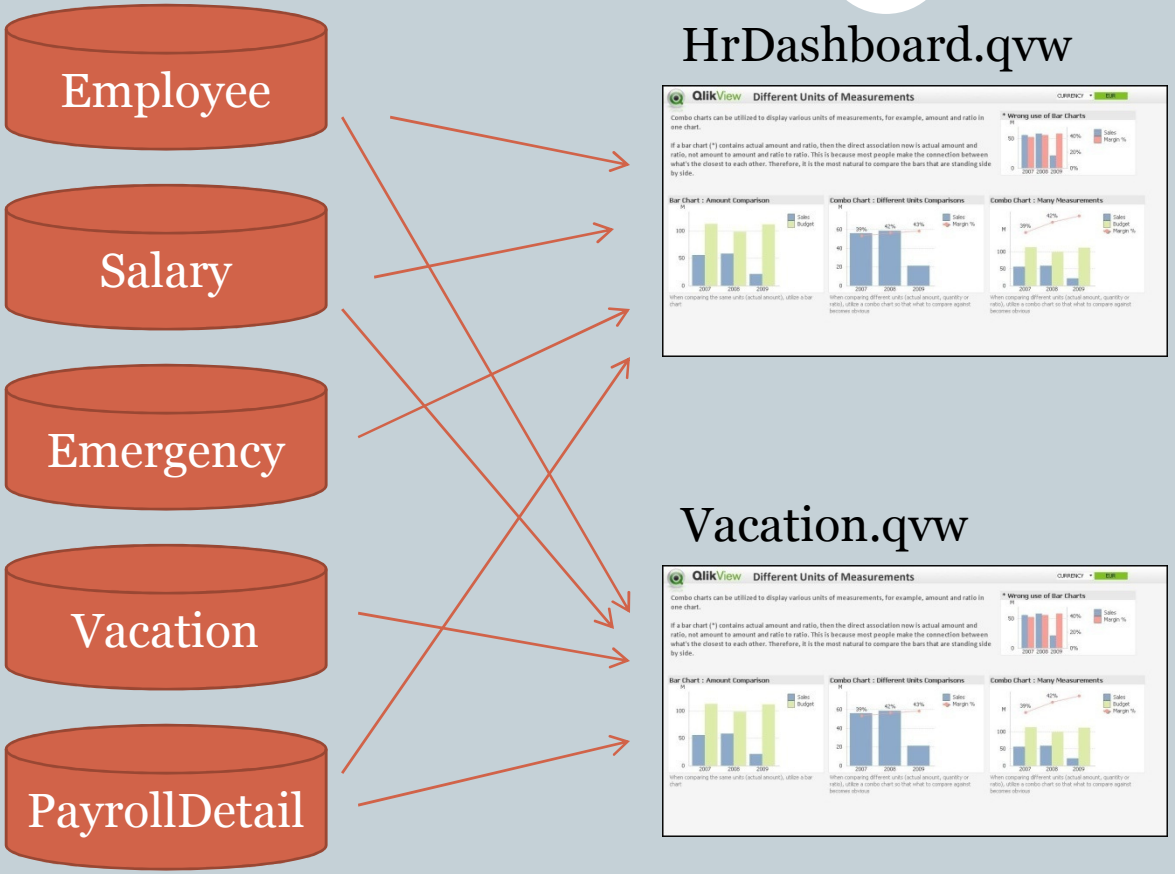
How?

Architectures

WHAT?

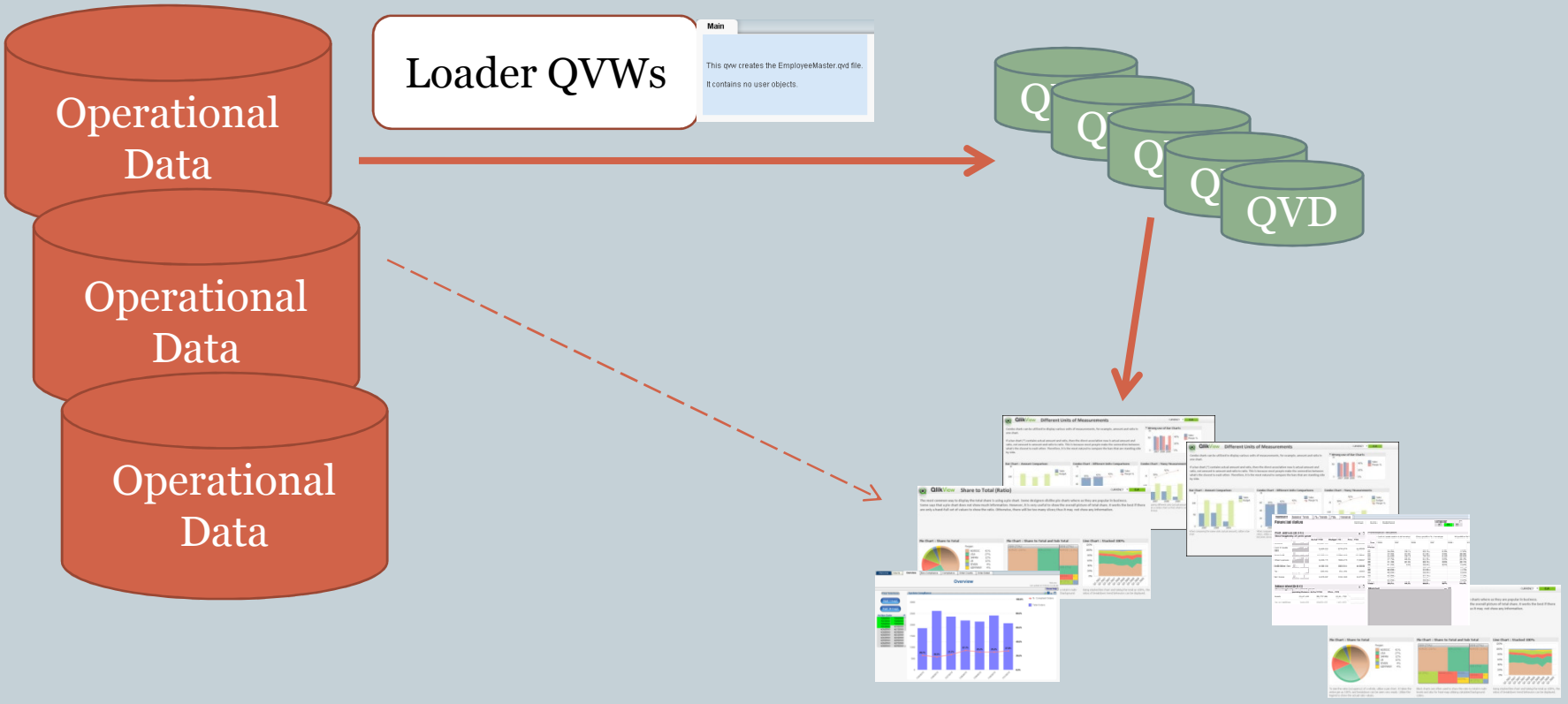
Basic "Single Tier" Model

Datasource



A *.qvw contains the ETL script, loaded data & screen objects.

Two-Tier Concept

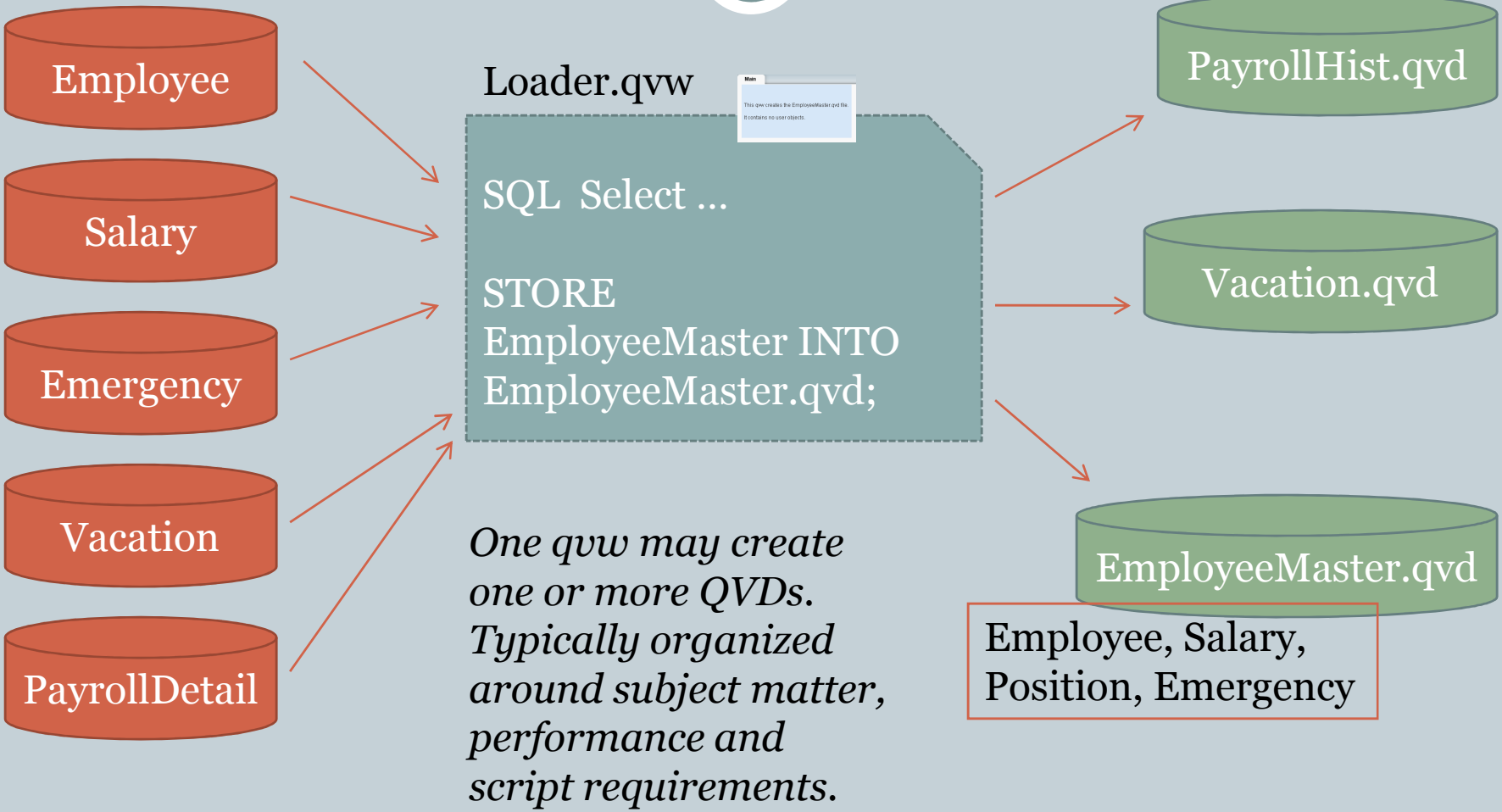


User facing qvws

QVDs

- **Q**lik**V**iew **D**ata file.
- One ***Qlikview*** table per *.qvd file.
- Loads very fast.
- Provides reusable “extract once” single version of ETL’ed data.
- May be used to provide history beyond current datamart.

Creating QVDs

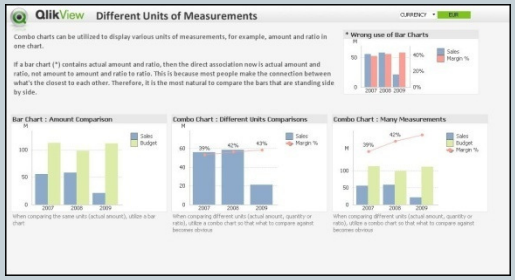


Consuming QVDs

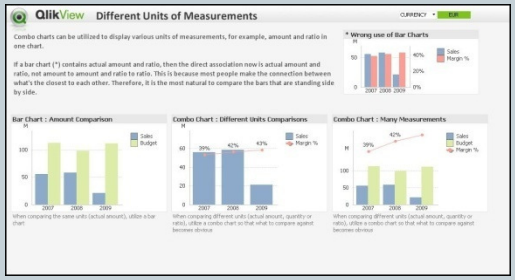
EmployeeMaster.qvd

Vacation.qvd

HrDashboard.qvw



Vacation.qvw



WHY?

QVD Benefits

- Loads fast
- Reduced DBMS load & improved scalability
- Ease of maintenance
- Separation of Roles
- De-normalization
- Incremental Load
- Data consistency, SVT
- QVD Data Dictionary

Loads Fast!

- Typically 10 to 100 times faster than loading from DBMS.

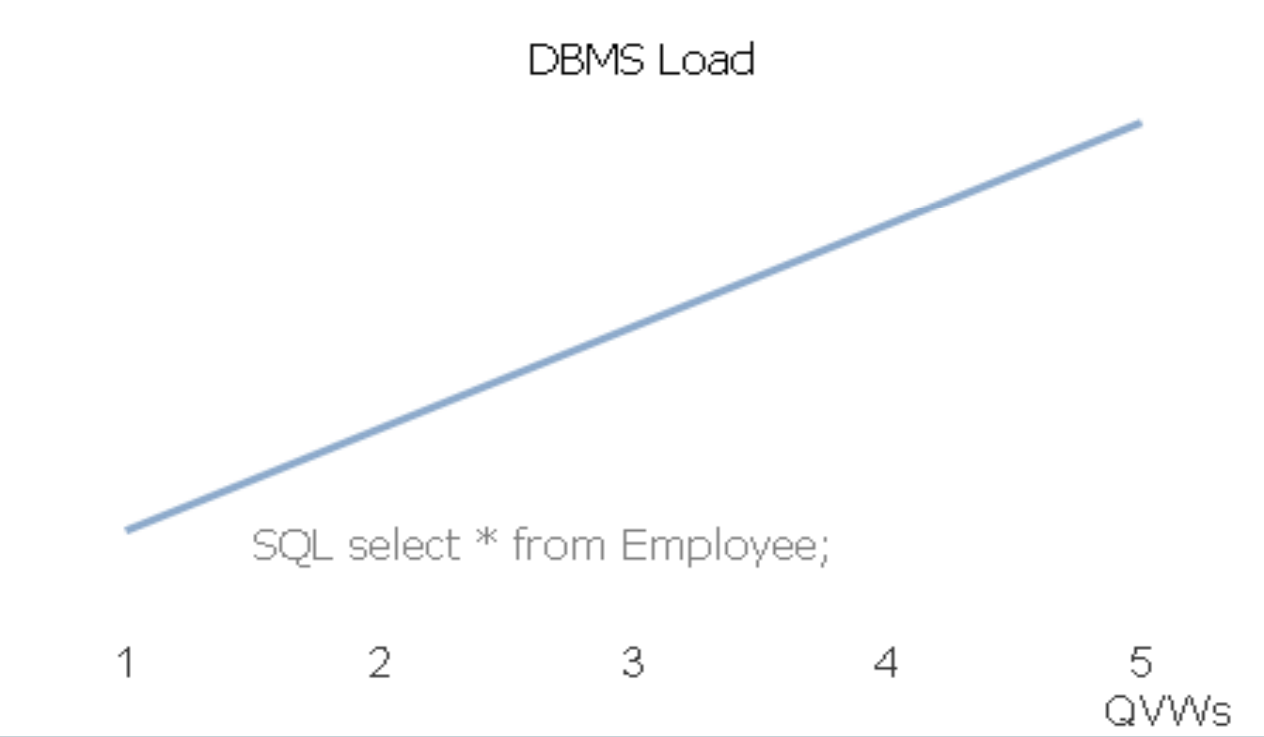
DBMS – 20 minutes

QVD – 30 seconds!

- Throughput is based on a number of factors including number of fields, network configuration and hard disk speed.

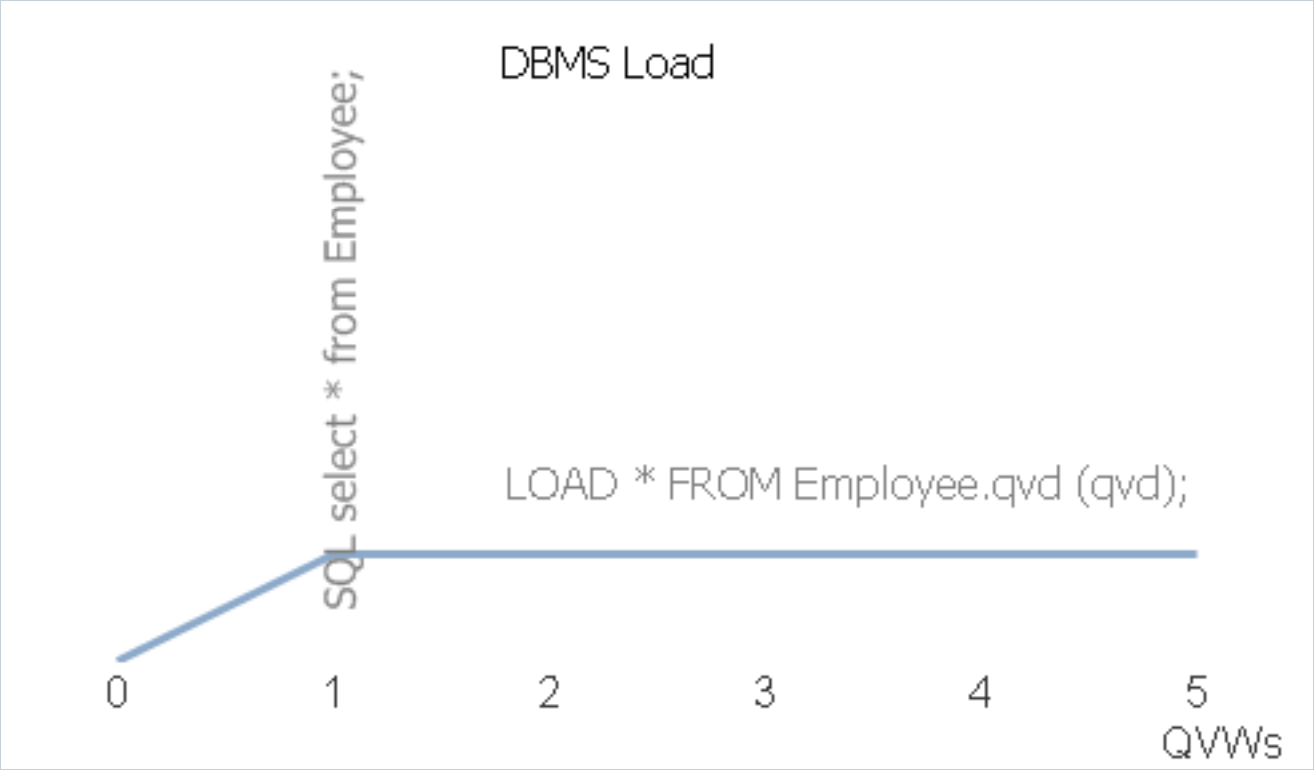
Reduced DBMS Load & Improved Scalability

- Every QVW repeats the SQL Select…



Reduced DBMS Load & Improved Scalability

- Or…one SQL Select, then load from QVD



Ease of Maintenance – Redundant Scripting –

QlikLearn

- A new business rule….

  Change all occurrences of CostCenter "3502" to"3501".

- Revenue.qvw

```
// Remap CostCenter 3502
if(CostCenter='3502', '3501', CostCenter) as CostCenter
```

- Sales.qvw

```
// Remap CostCenter 3502
if(CostCenter='3502', '3501', CostCenter) as CostCenter
```

- Commission.qvw

```
// Remap CostCenter 3502
if(CostCenter='3502', '3501', CostCenter) as CostCenter
```

Ease of Maintenance
– Eliminate Redundant Script –

QlikLearn

- LoaderCostCenter.qvw

```
// Remap CostCenter 3502
if(CostCenter='3502', '3501', CostCenter) as CostCenter

STORE CostCenterTable INTO CostCenter.qvd;
```

- Revenue.qvw

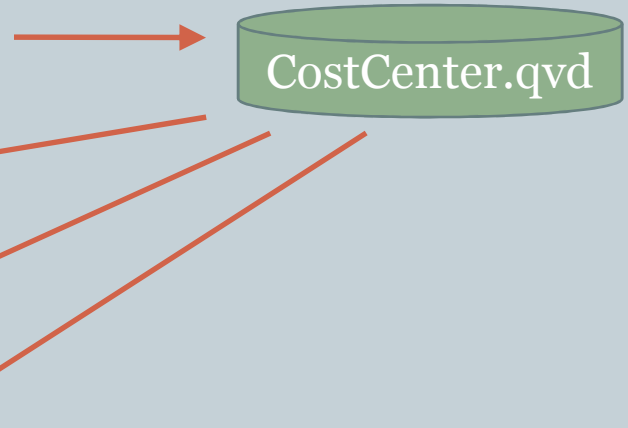
```
LOAD * FROM CostCenter.qvd (qvd);
```

- Sales.qvw

```
LOAD * FROM CostCenter.qvd (qvd);
```

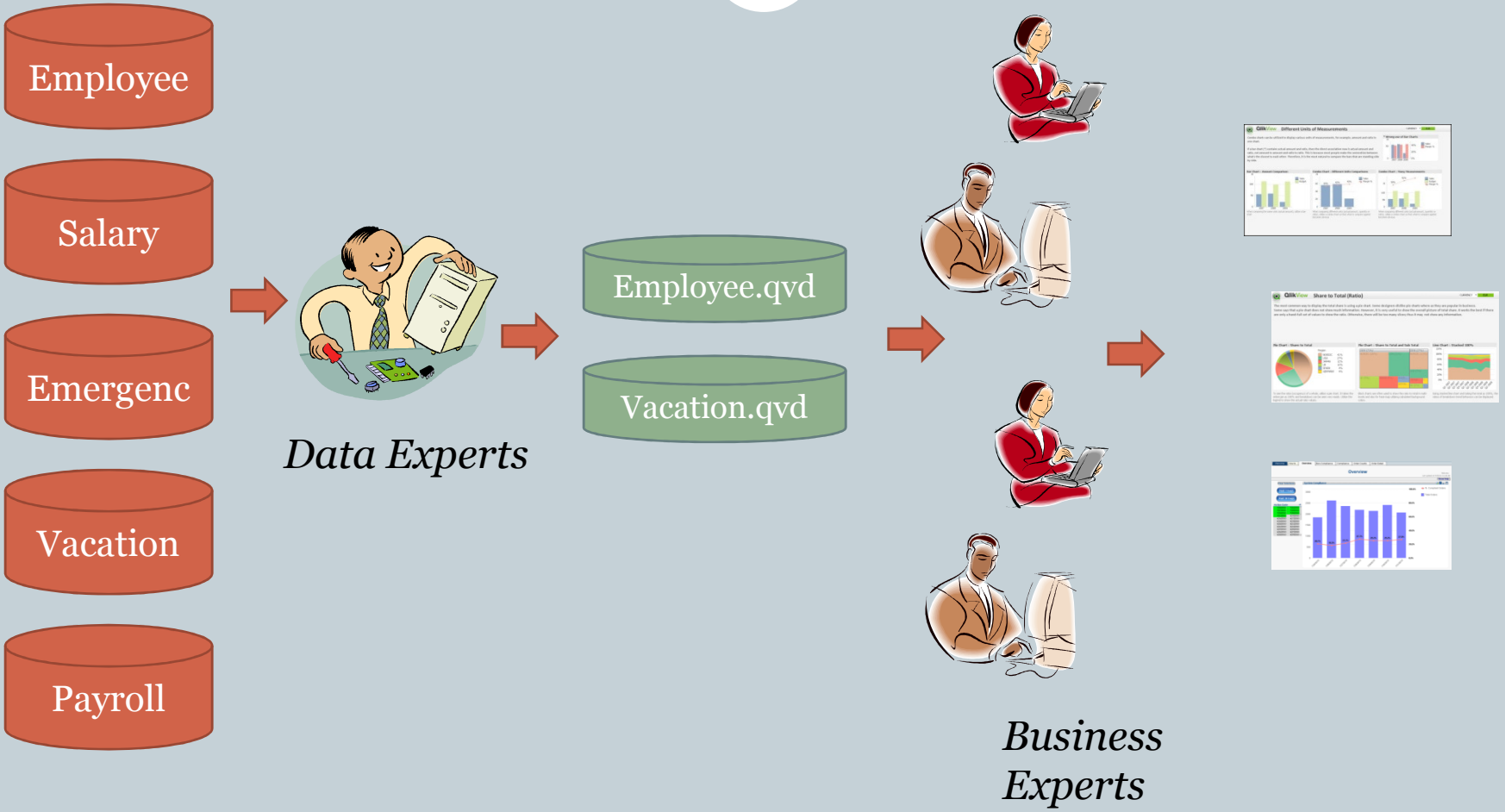
- Commission.qvw

```
LOAD * FROM CostCenter.qvd (qvd);
```



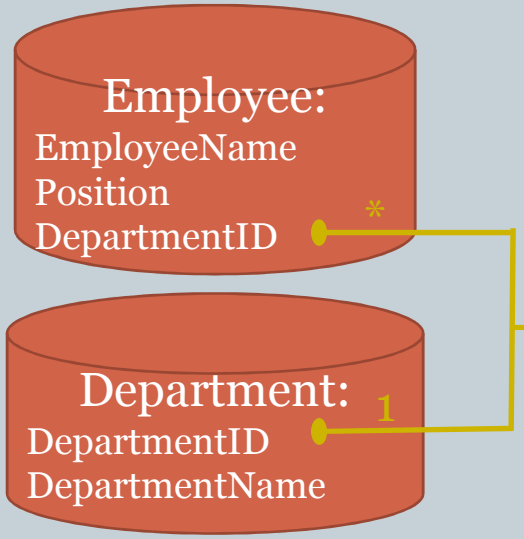
QlikLearn

Separation of Roles



QlikLearn

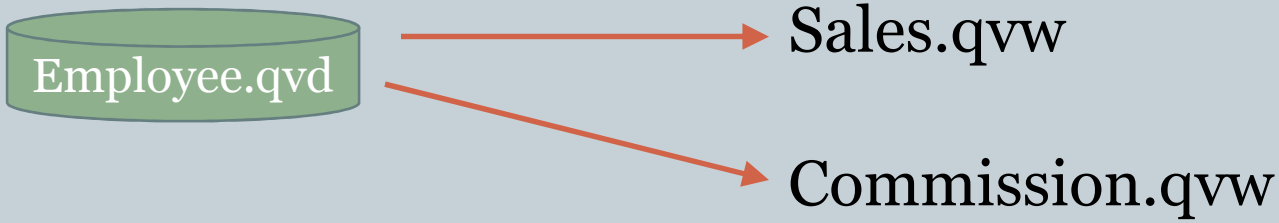
Denormalization




```
LEFT JOIN (Employee)
SQL SELECT DepartmentID,
DepartmentName
FROM dbo.Department ;
```

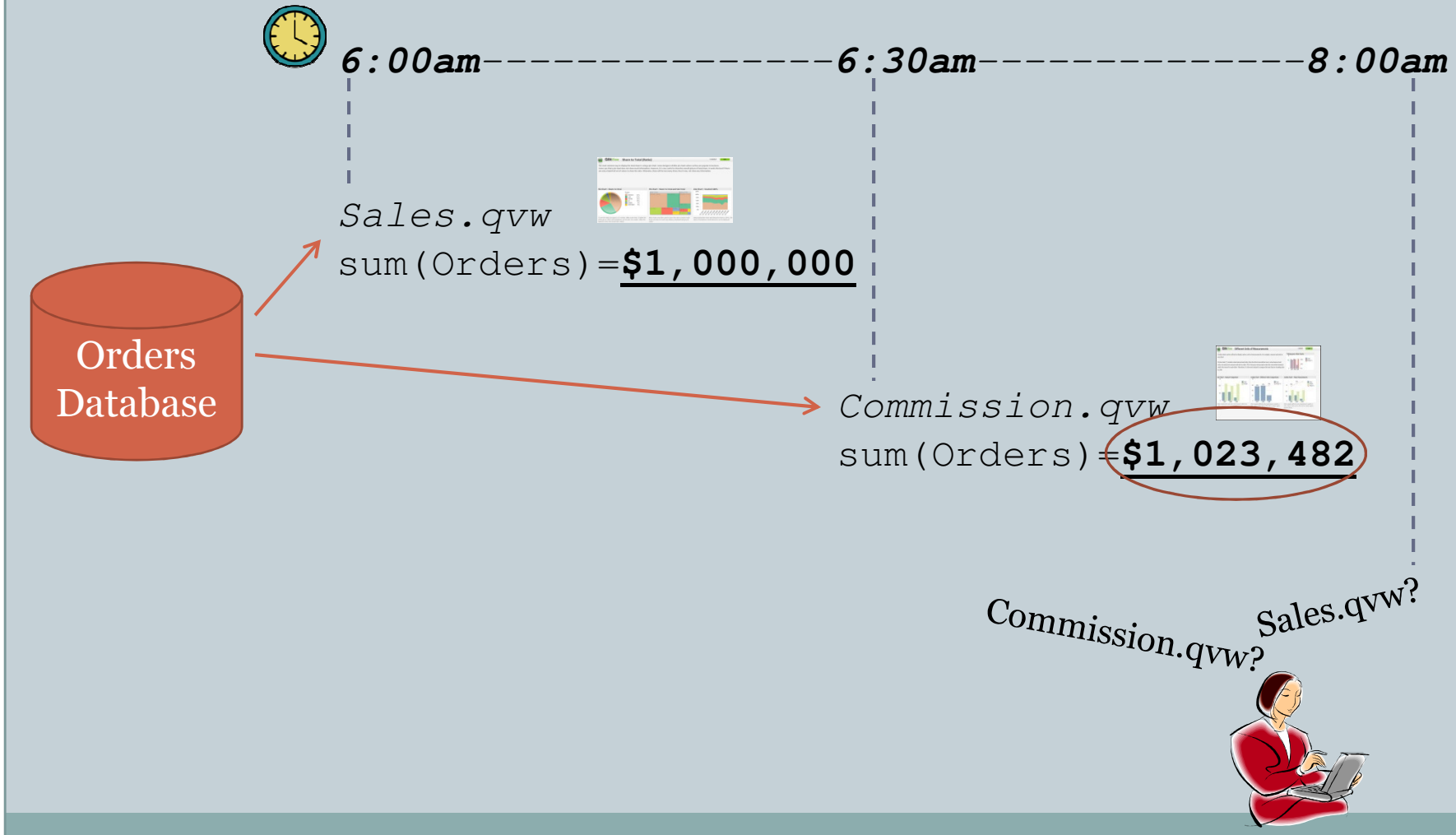
EmployeeName	DepartmentName
Alan Haight	Customer Service
Joe Boxer	Sales
Sally Smith	Sales

```
STORE Employee INTO Employee.qvd;
```



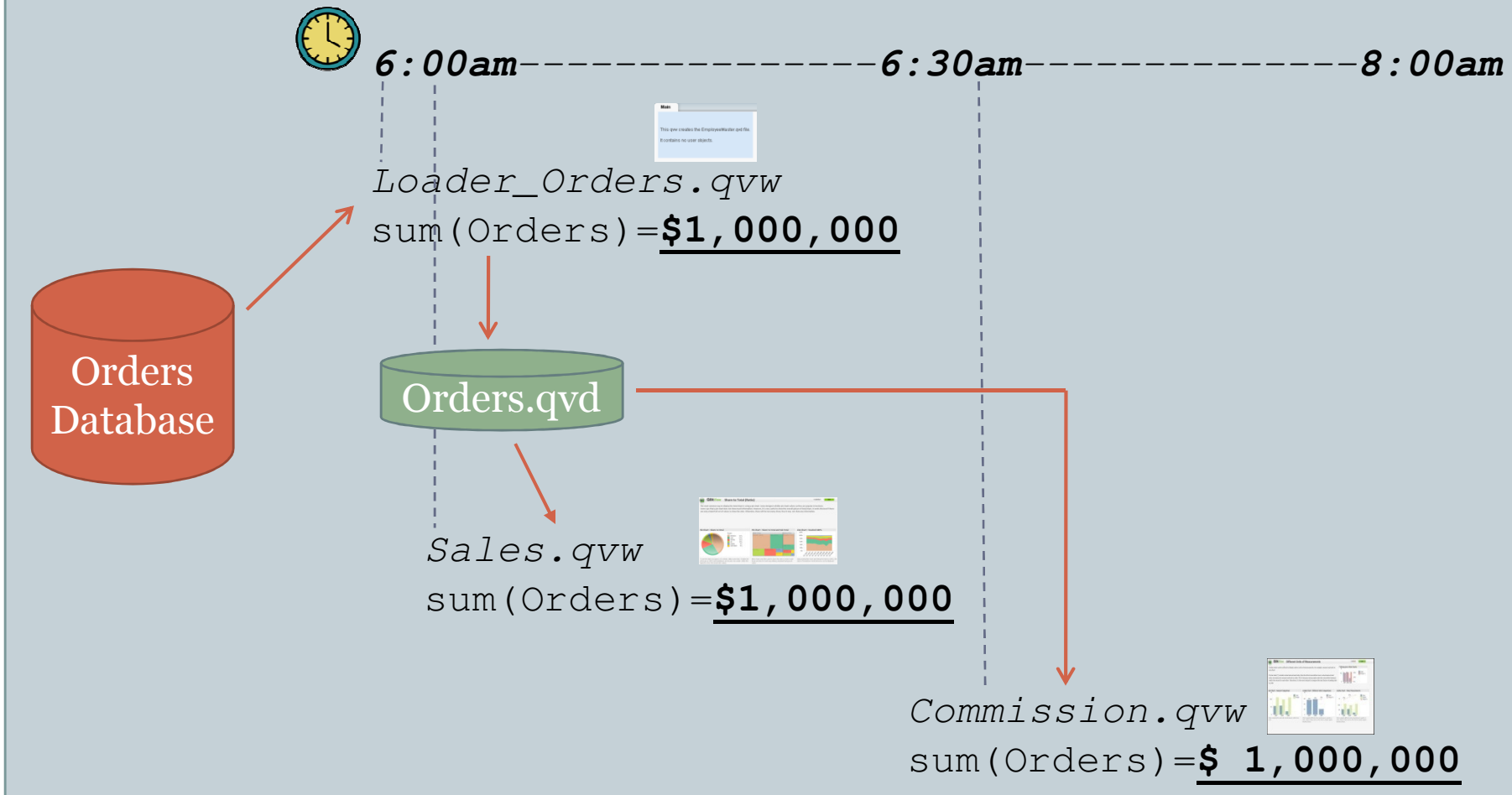
QlikLearn

Data consistency, SVT



QlikLearn

QVD Data consistency, SVT



Incremental Load

- Fetch only new or changed rows from DBMS
- New rows are merged with existing rows in QVD
- Greatly reduces load on DBMS

Full Table: 10M rows, 30 minutes

Changed data, 50k rows, 1 minute

- Reduced load time may enable intra-day reloads

History & Snapshots

- QVDs may be used to store data longer than DBMS
- Some applications may require periodic snapshots for auditing or delta reporting:
  - `Customers_2010_06.qvd`
  - `Customers_2010_07.qvd`

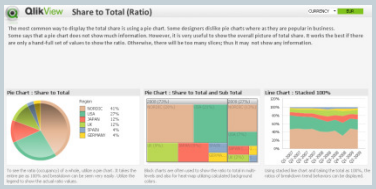
How?

Creating QVDs

- QVDs may be created in three ways:
  - Explicitly using STORE script statement
  - Implicitly using BUFFER script statement
  - Via UI chart Export

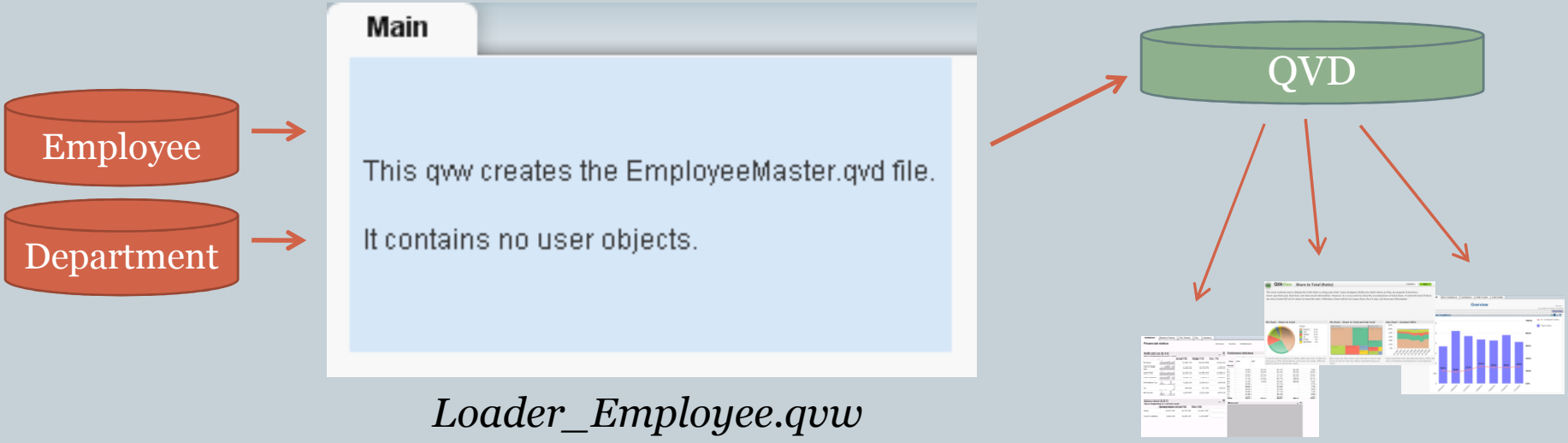
Creating QVD - STORE

- STORE may be done in script of user-facing qvw



HrDashboard.qvw

- More typically done in special purpose "Loader" qvw



Loader_Employee.qvw

STORE Syntax

- **STORE *table* INTO *filename*.qvd (qvd|txt);**
- Default is "(qvd)" – typically not specified
- "(txt)" creates a CSV file
- ***filename*** may include path – `data\employee.qvd;`
- ***filename*** respects Directory statement. e.g.

```
Directory data;
STORE Employee INTO employee.qvd;
```

Writes to **data\employee.qvd**

Creating QVD - BUFFER

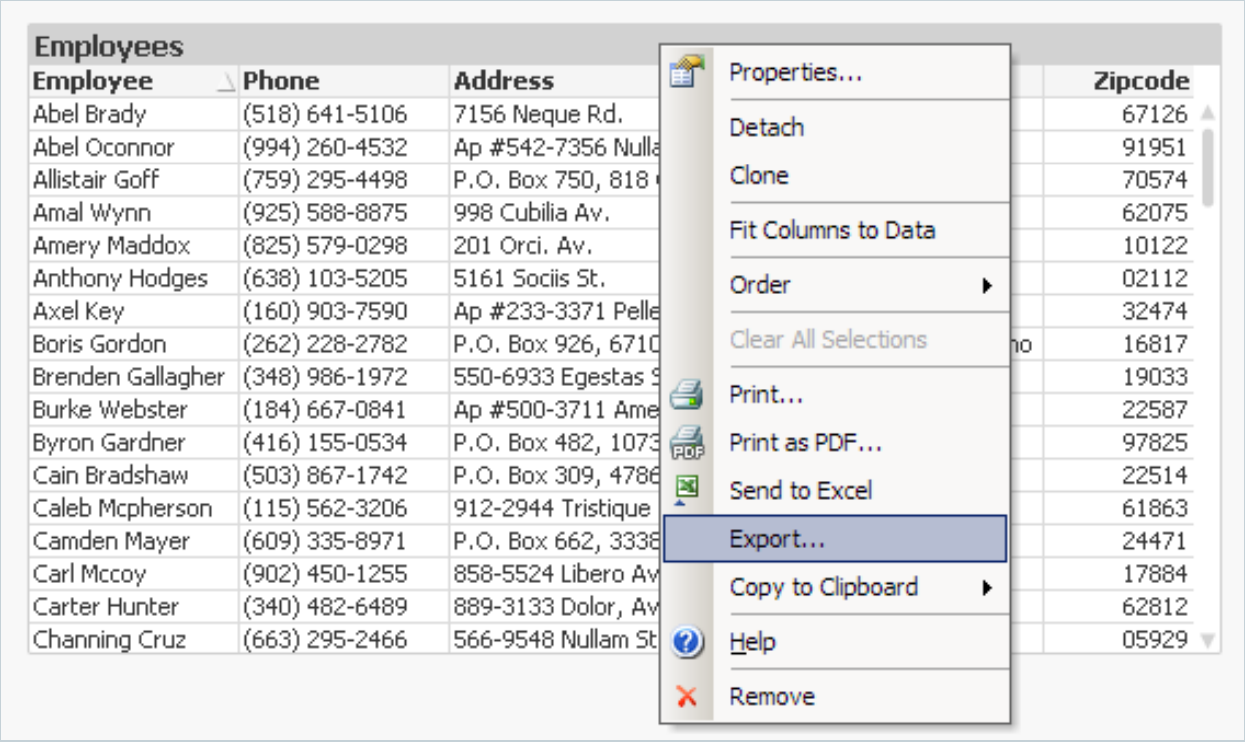
- BUFFER prefix on LOAD signifies that a QVD should be automatically created on first execution, and read from on subsequent executions.

```
BUFFER LOAD * FROM Access*.log;
```

- QVD is created in a special location. The QVD is automatically dropped and recreated if the LOAD statement changes.
- Very useful with text files (like logs) where incremental load is “automatic”.
- Less useful with DBMS tables, but often used to speed development.

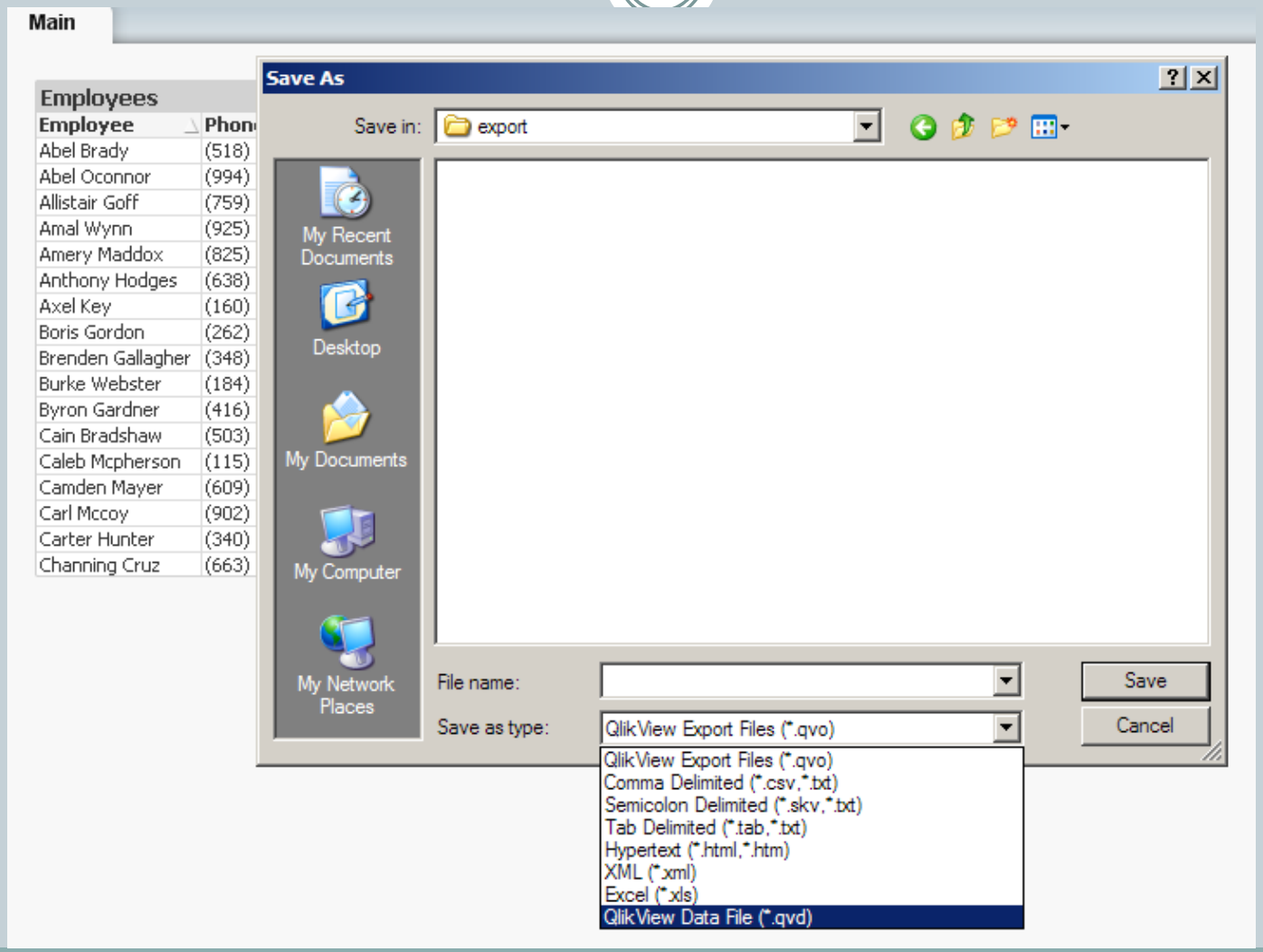
Creating QVD – UI Export

- Charts (not TableBoxes) may be exported to QVD using context menu Export



QlikLearn

Creating QVD – UI Export

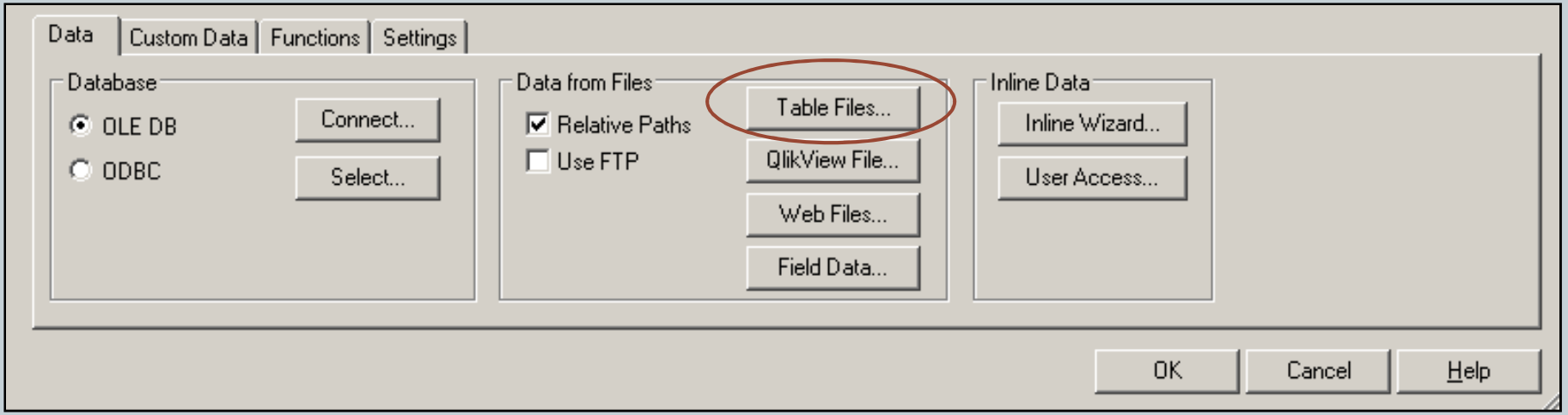


Loading QVDs

- QVDs may be loaded
  - Explicity – `LOAD * FROM data.qvd (qvd);`
  - Implictly – `BUFFER LOAD …`

Loading QVDs Explicit

- Script wizard support provided by Table Files button



Loading QVDs Explicit

- **LOAD *fieldlist* FROM filename.qvd (qvd);**
- **(qvd) parameter is critical. If ommitted, script error or unexpected (xml) data.**
- **filename** respects Directory statement. e.g.

```
Directory data;
LOAD * FROM Employee.qvd (qvd);
```

Reads from **data\employee.qvd**

Loading QVD - BUFFER

```
BUFFER LOAD * FROM Access*.log;
```

- Will load from QVD if it exists, otherwise will read FROM source.
- If source is text files, will read only new data.

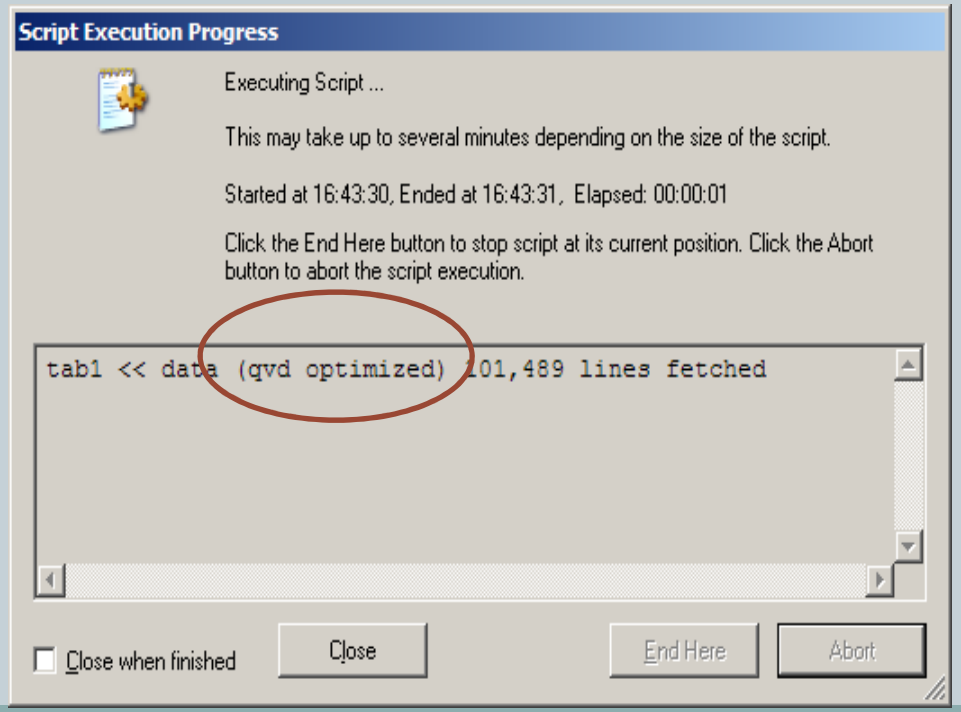
Loading QVD - BUFFER

- May be used to reload DBMS tables in the same qvw on different frequencies.
- This qvw is reloaded hourly.

```
Orders:
// Orders are loaded on every reload
SQL SELECT * FROM Orders;
Customers:
// Customers are loaded only once per day
BUFFER (stale after 1 days) SELECT * FROM
  Customers;
```

Optimized Load

- **Optimized load (super fast) will occur if no transformations or WHERE predicates. Optimized load indicated by message in script progress window.**



QlikLearn

Optimized Load Limitations

- Optimized Load Limits
  1) `F1 as newfield` – yes, rename allowed
  2) `F1 * 2 as newfield` – no, transformation
  3) `1 as newfield` – no, cannot add new field
  4) `where exists(F1)` – yes, single field exists()
  5) `where exists(F1, F3)` – no, multi-field exists()
  6) `F1 as newfield where exists(F1)` – no

QVD as Map

- QVDs may be used in MAPPING LOAD except…
- Optimized load cannot be used. No error, but no mapping results.
- Solution is to force a non-optimized load:

```
EmpMap:
MAPPING LOAD EmpID, EmpName
FROM Emp.qvd (qvd)
WHERE 1=1
;
```

QVD Functions

- Several useful script functions obtain metadata from a QVD.
- Documented in “File Functions” section of Ref Guide. Examples:
  - **QvdNoOfRecords(*filename*)**
  - **QvdNoOfFields(*filename*)**
- Qvd Functions do not respect Directory statement. Specify full path relative to qvw.
- See Ref Guide for complete list

QVD Dictionary

- QVD Functions may be used with a filelist loop to create a dictionary of QVDs for reference

QVD Name	QVD Table Name	QVD Create Time	QVD Record Count	QVD File Size (KB)	QVD Field Count
Categories.qvd	Categories	2010-08-31 8:47 pm	8	171	4
Customers.qvd	Customers	2010-08-31 8:47 pm	91	17	11
Employees.qvd	Employees	2010-08-31 8:47 pm	9	393	18
EmployeeTerritori...	EmployeeTerrit...	2010-08-31 8:47 pm	49	2	2
Invoices.qvd	Invoices	2010-08-31 8:47 pm	2,155	127	26
OrderDetails.qvd	Order Details	2010-08-31 8:47 pm	2,155	19	5
Orders.qvd	Orders	2010-08-31 8:47 pm	830	67	14
Products.qvd	Products	2010-08-31 8:47 pm	77	10	10
Region.qvd	Region	2010-08-31 8:47 pm	4	2	2
Shippers.qvd	Shippers	2010-08-31 8:47 pm	3	2	3
Suppliers.qvd	Suppliers	2010-08-31 8:47 pm	29	10	12
Territories.qvd	Territories	2010-08-31 8:47 pm	53	3	3

QVD Dictionary

- QVD Field Analysis

QVD Field Name	
Address	4
BirthDate	1
CategoryID	2
CategoryName	1
City	4
CompanyName	3
ContactName	2
ContactTitle	2
Country	4
CustomerID	3
CustomerName	1
Description	1
Discontinued	1
Discount	2
EmployeeID	3
ExtendedPrice	1
Extension	1
Fax	2
FirstName	1
Freight	2
HireDate	1
HomePage	1
HomePhone	1
LastName	1

Field Occurance

QVD Field Name	QVD Name	Categories	Customers	Employees	EmployeeTerritories	Invoices	OrderDetails	Orde
Address			X	X		X		
BirthDate				X				
CategoryID		X						
CategoryName		X						
City			X	X		X		
CompanyName			X					
ContactName			X					
ContactTitle			X					
Country			X	X		X		
CustomerID			X			X		
CustomerName						X		
Description		X						
Discontinued								
Discount						X	X	
EmployeeID				X	X			
ExtendedPrice						X		
Extension				X				
Fax			X					
FirstName				X				
Freight						X		
HireDate				X				
HomePage								

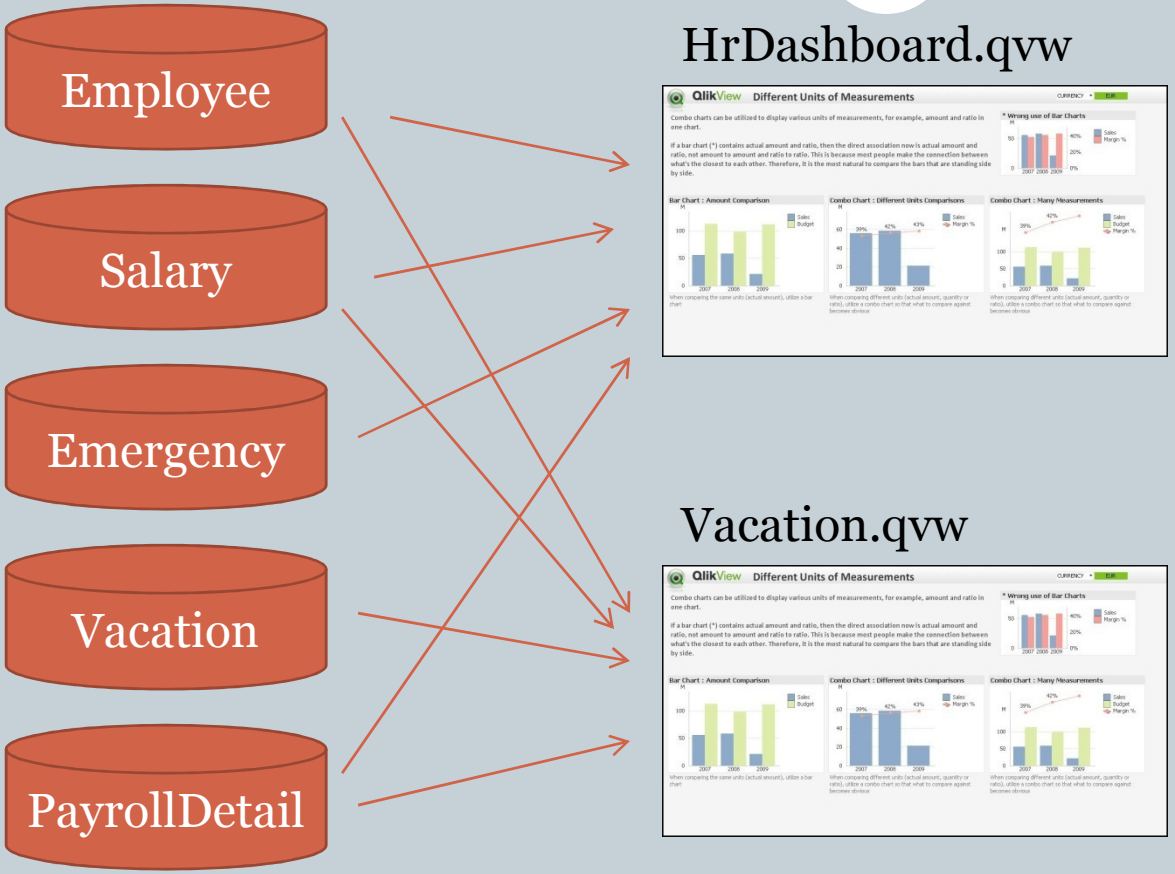
QVX

- New with V10 -- **Q**lik**V**iew data e**X**change
- “Open variant of QVD”
- Not quite as fast as QVD
- May be used to create high performance input files from sources where ODBC drivers are not available.
- Read by standard LOAD script statement.
- MetaFunctions **--QvdNoOfFields(** ***filename*** **)**

ARCHITECTURES

Basic "Single Tier" Model

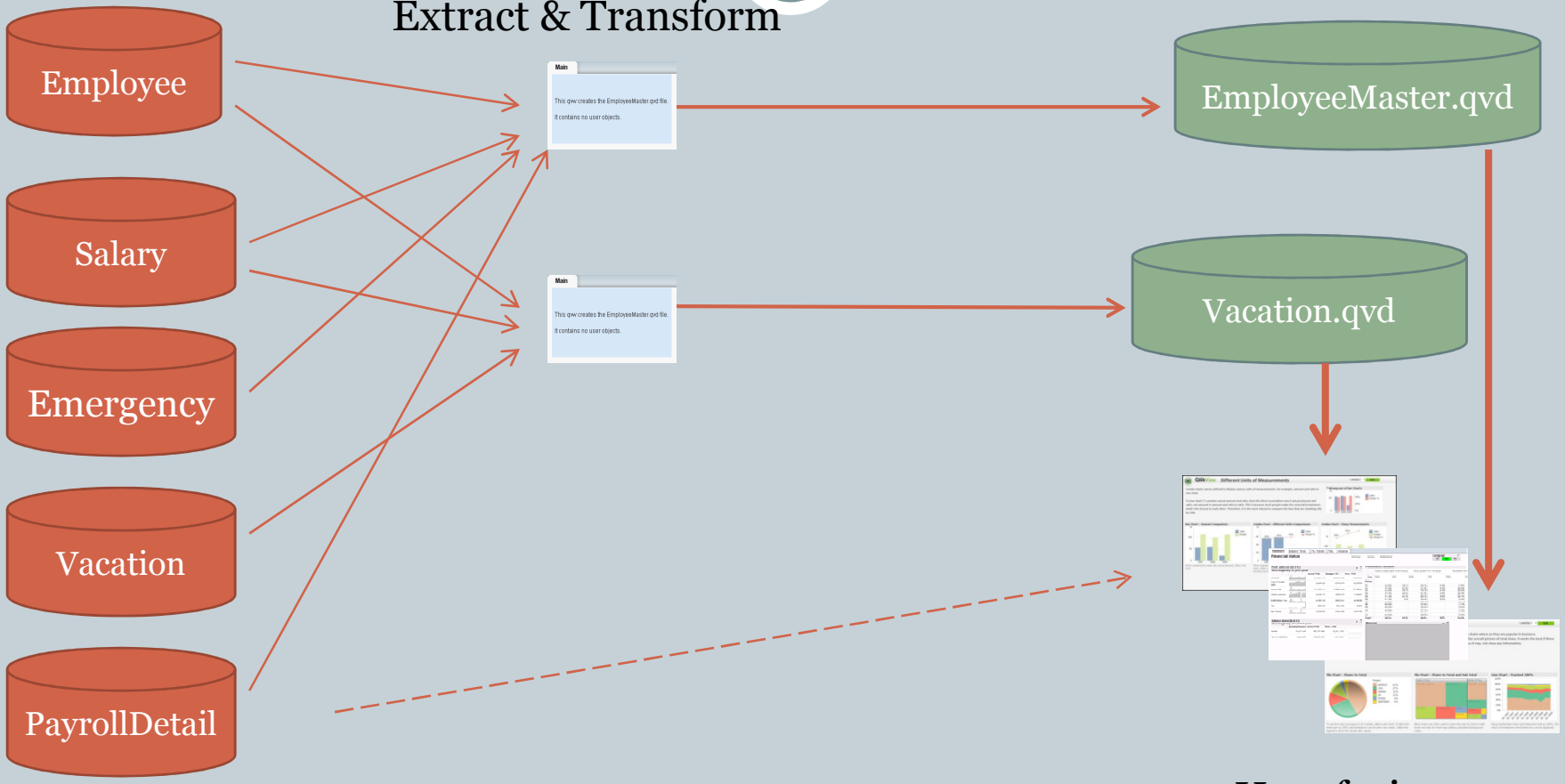
Datasource



A *.qvw contains the ETL script, loaded data & screen objects.

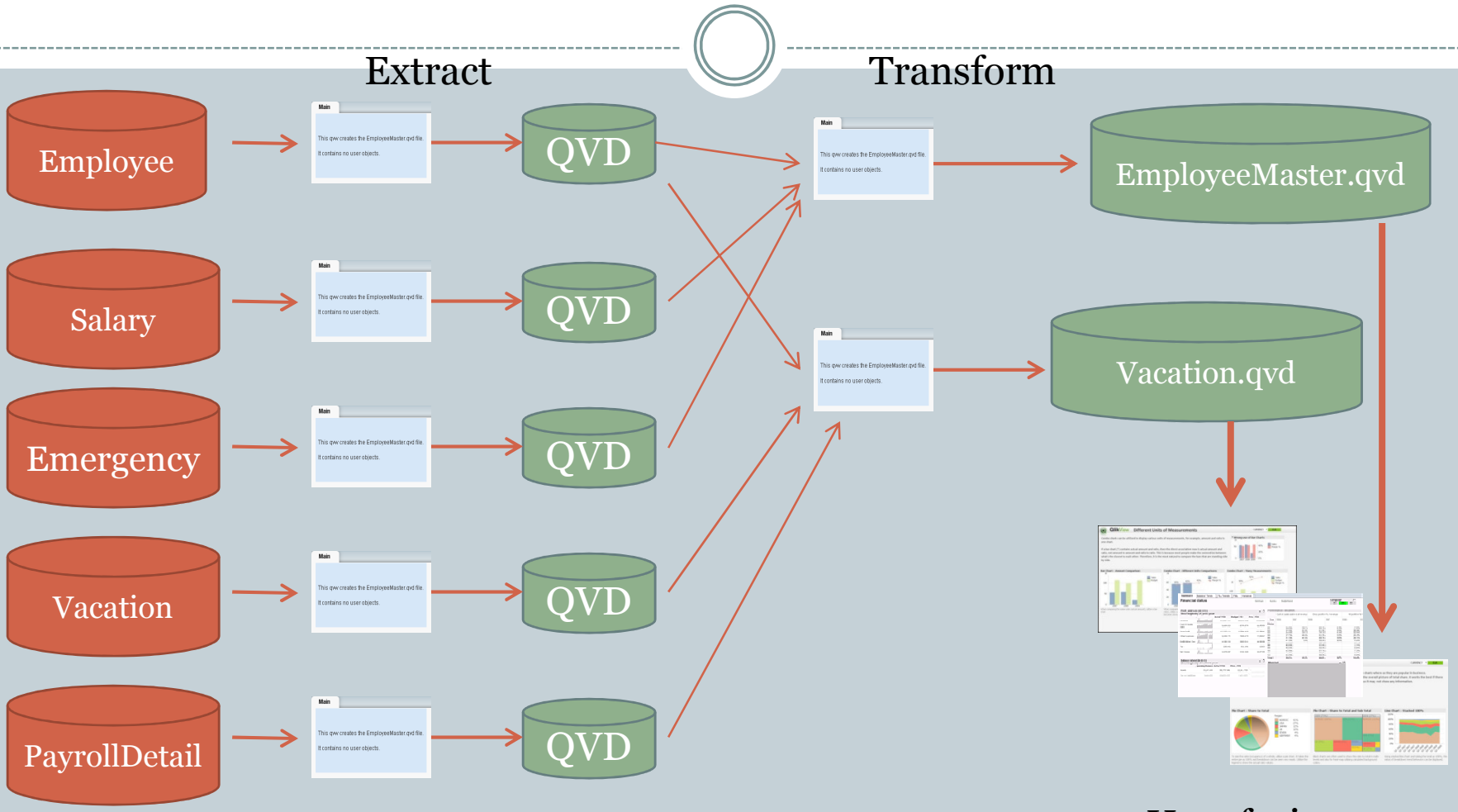
QlikLearn

Two Tier Model



QlikLearn

Three Tier Model



QlikLearn

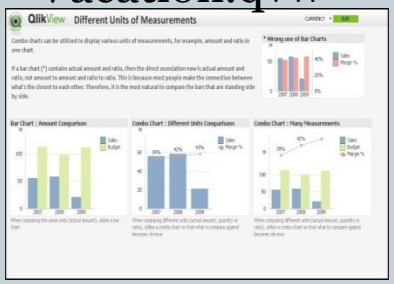
Bonus -- QlikMarts

QVWs may also be consumed as datasources.
Script: `BINARY Vacation.qvw;`

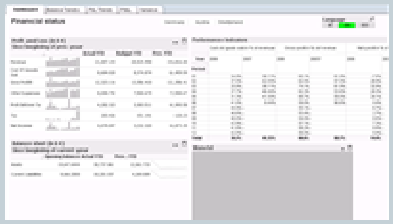
EmployeeMaster.qvd

Vacation.qvd

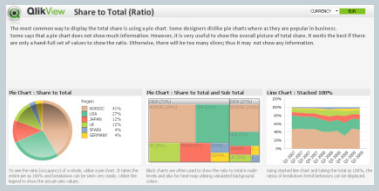
Vacation.qvw



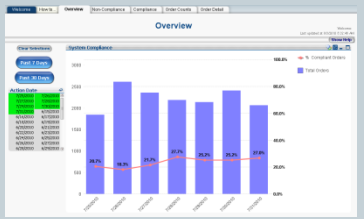
Repurpose the data model of a qvw for different UI audiences.



VacationByDept.qvw



VacationForecast.qvw



VacationSelfServe.qvw

Questions and Next Steps

Learn More about our QVD Data Model Pack at…

www.factory-ims.com

Questions?

Please use the Q&A feature in WebEx

QlikLearn

Contact Us

QlikView

Rob Patterson
Senior Manager, User Enablement

rpn@qliktech.com

Rob Wunderlich
Senior QlikView Consultant

rwunderlich@factory-ims.com

community.qlikview.com

www.factory-ims.com

QlikLearn