



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?

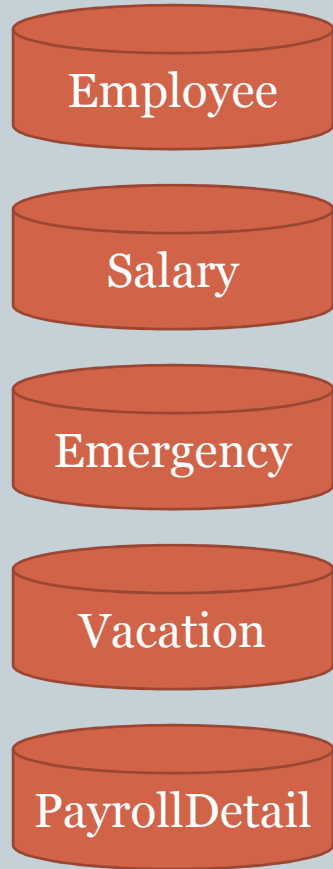
ARCHITECTURES



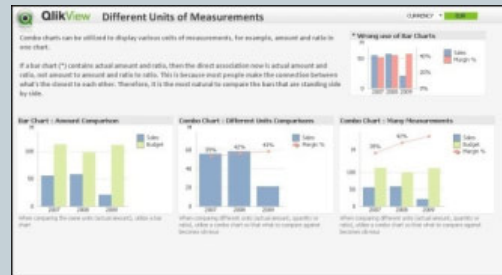
WHAT?

Basic “Single Tier” Model

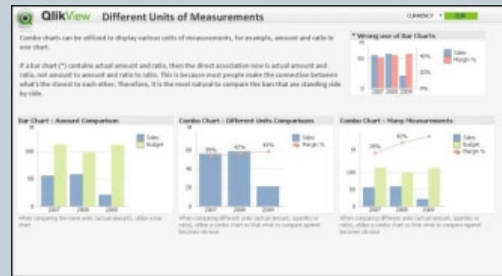
Datasource



HrDashboard.qvw

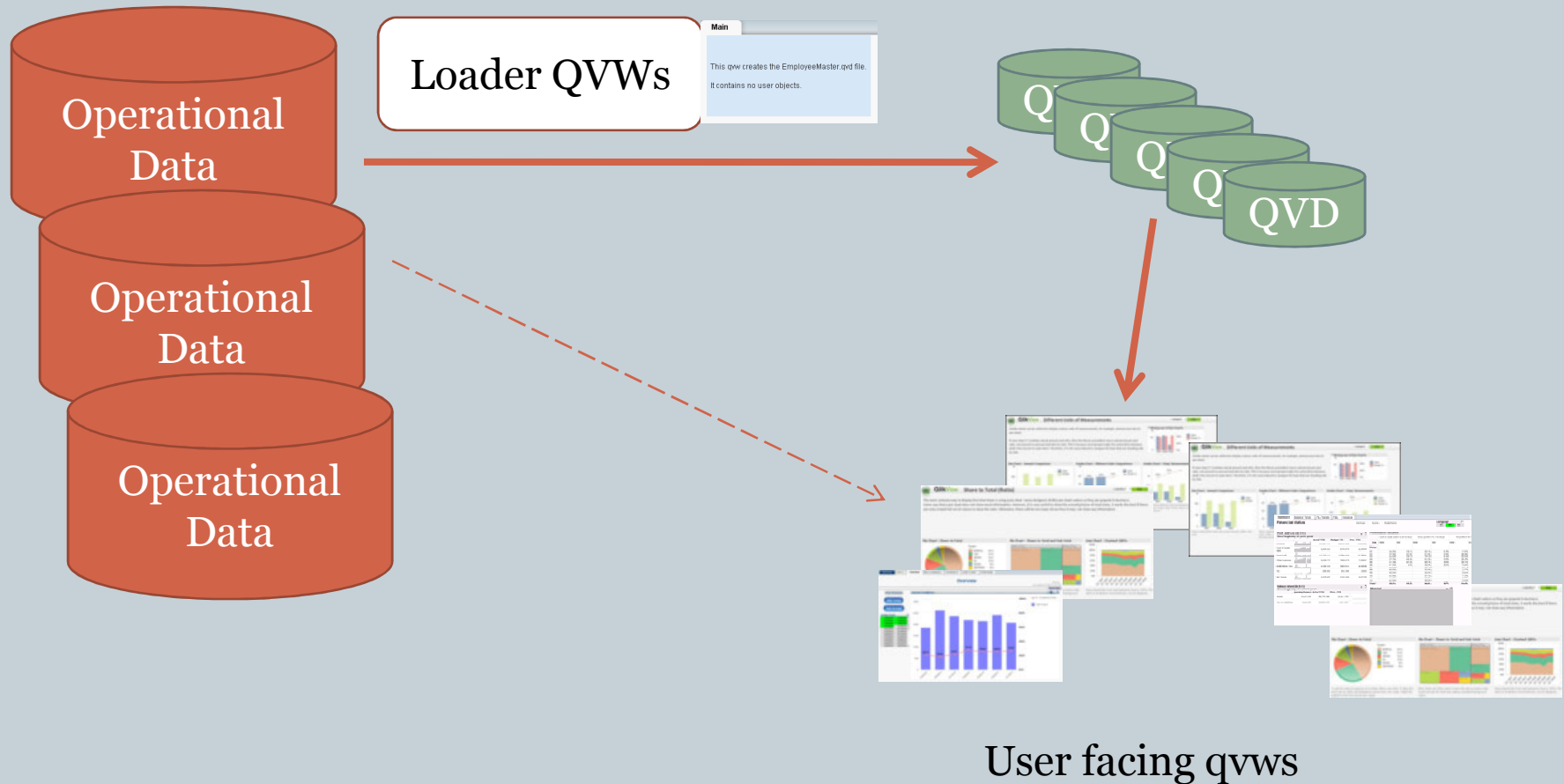


Vacation.qvw



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

Two-Tier Concept

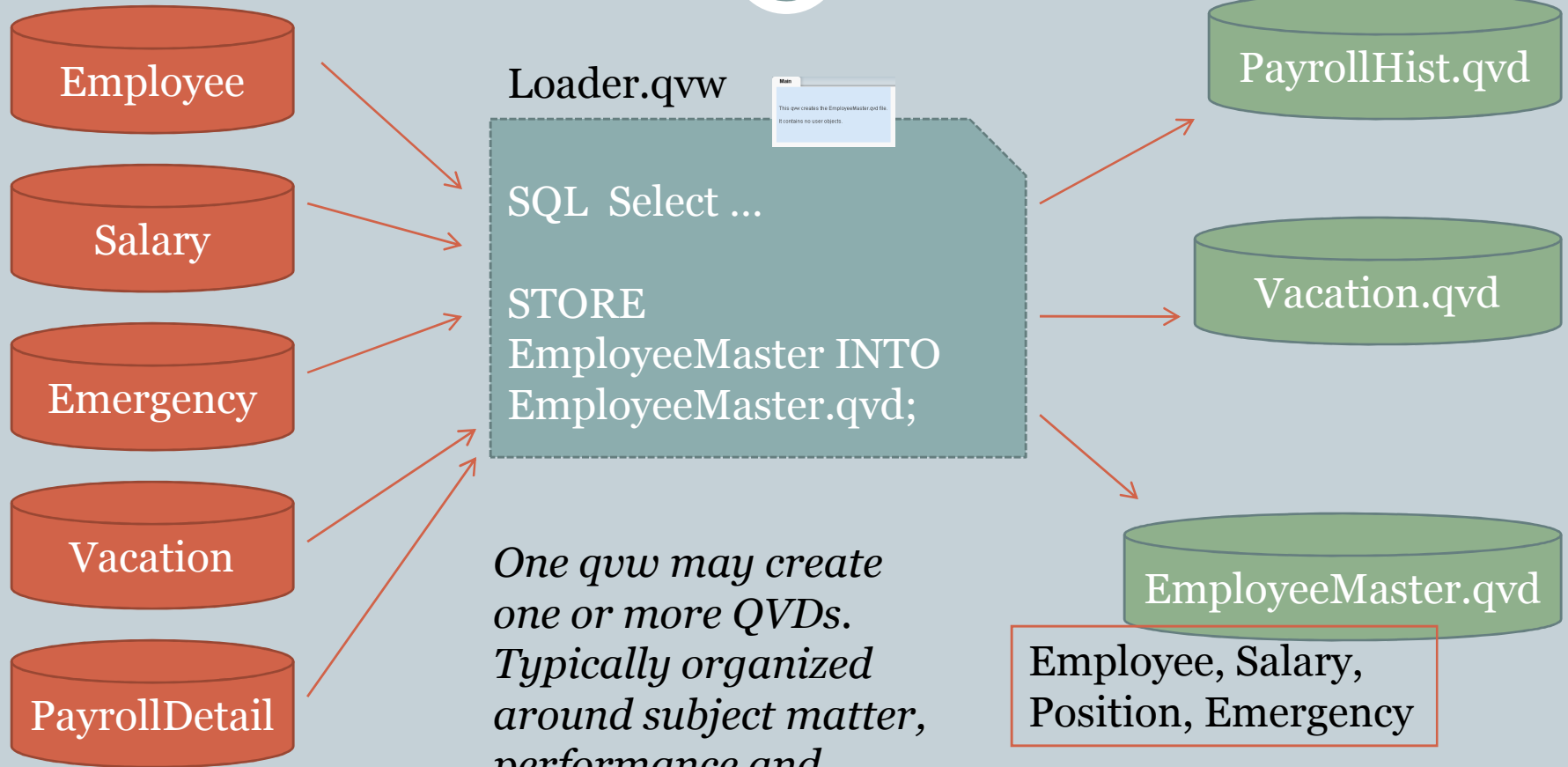


QVDs



- **QlikView Data** 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



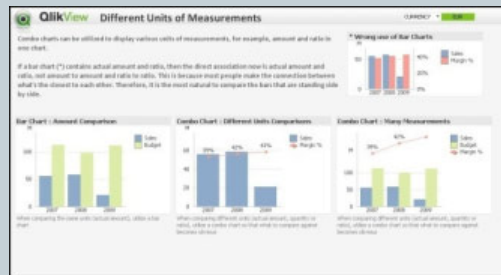
One qvw may create one or more QVDs. Typically organized around subject matter, performance and script requirements.

Employee, Salary, Position, Emergency

Consuming QVDs



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

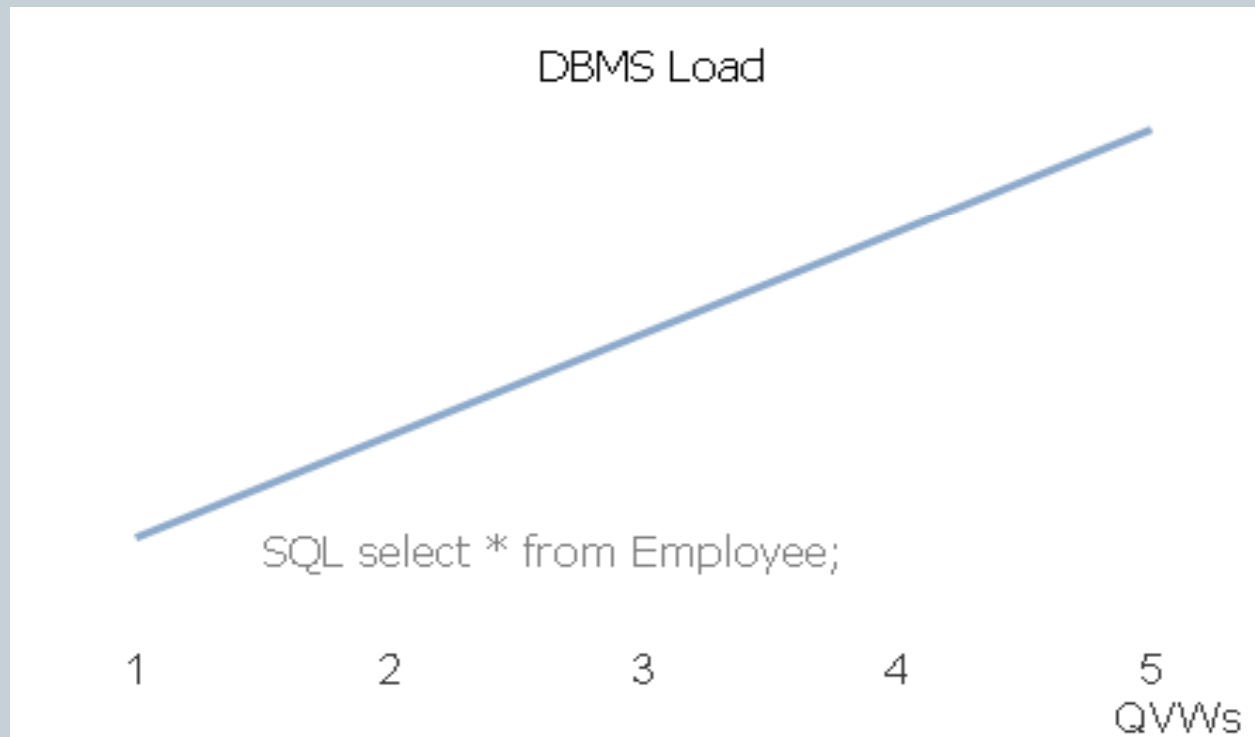
QVD – 30 seconds!

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

Reduced DBMS Load & Improved Scalability



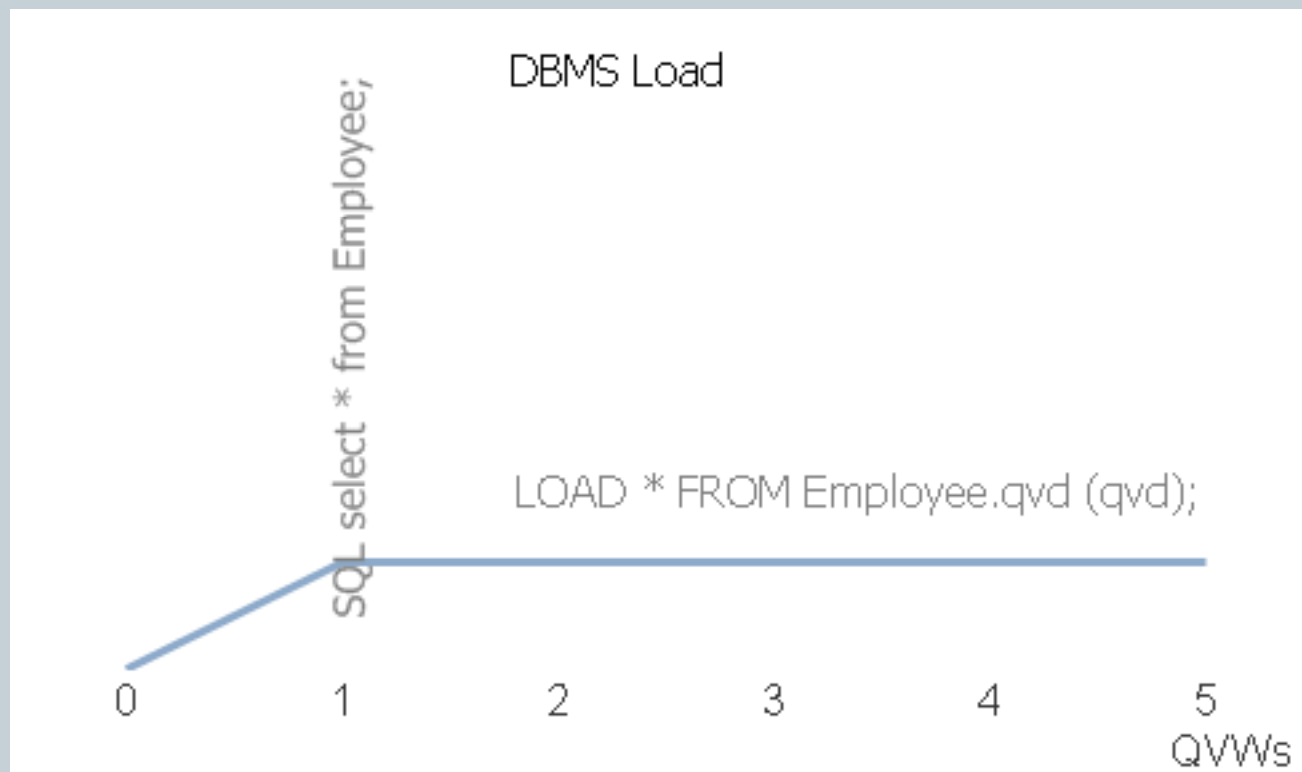
- Every QVW repeats the SQL Select...



Reduced DBMS Load & Improved Scalability



- Or...one SQL Select, then load from QVD



Ease of Maintenance

– Redundant Scripting –

- 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 –

- 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

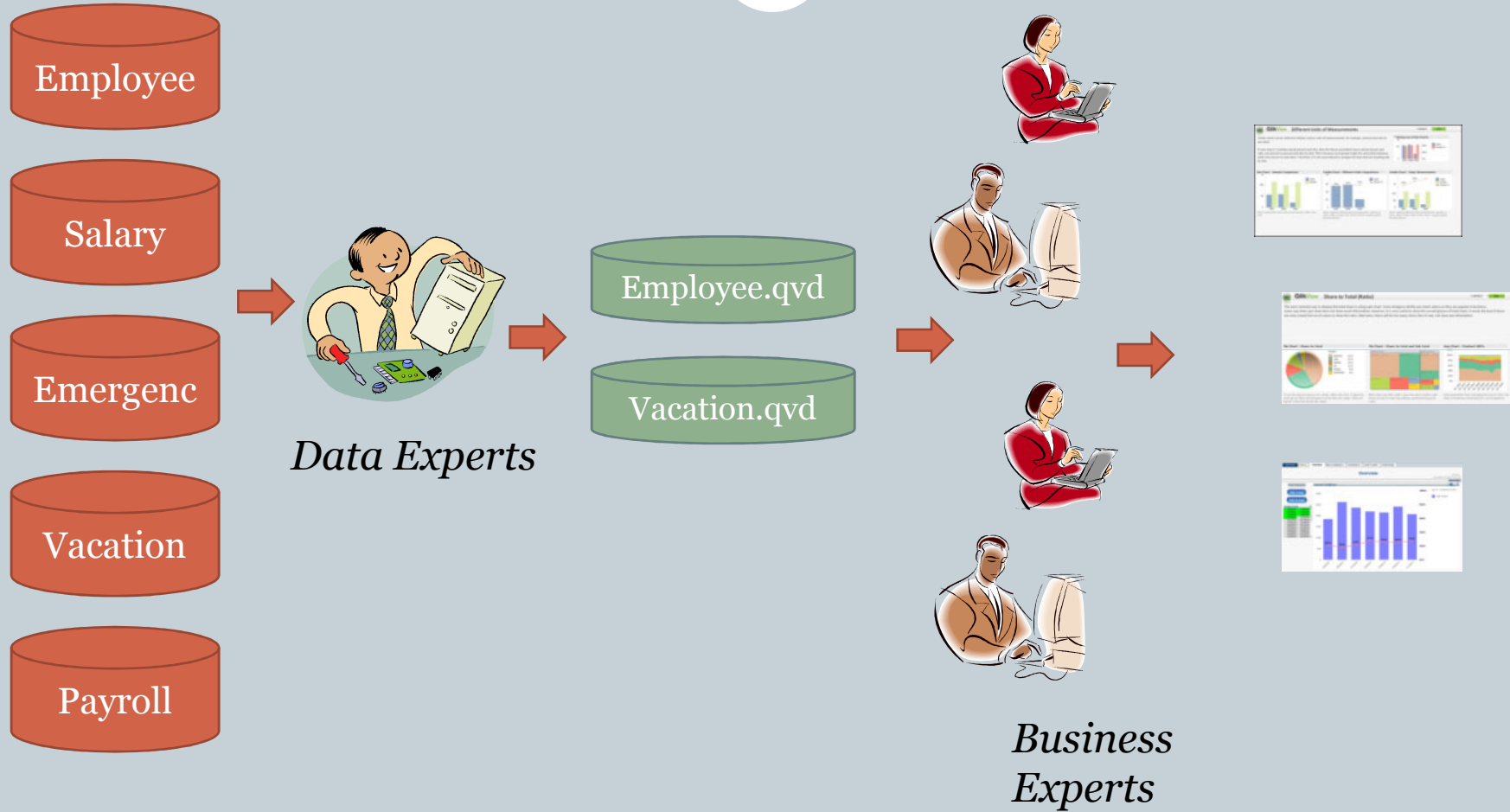
```
LOAD * FROM CostCenter.qvd (qvd);
```

- Commission.qvw

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



Separation of Roles



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;
```



Sales.qvw

Commission.qvw

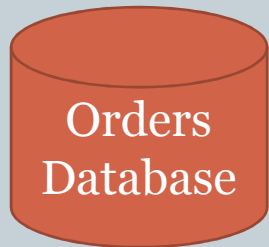
Data consistency, SVT



6:00am

6:30am

8:00am



Sales.qvw
sum(Orders) = \$1,000,000



Commission.qvw
sum(Orders) = \$1,023,482



Commission.qvw?
Sales.qvw?



QVD Data consistency, SVT



6:00am

6:30am

8:00am

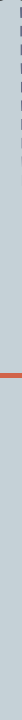


Loader_Orders.qvw
sum(Orders) = \$1,000,000



Sales.qvw
sum(Orders) = \$1,000,000

Commission.qvw
sum(Orders) = \$ 1,000,000



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

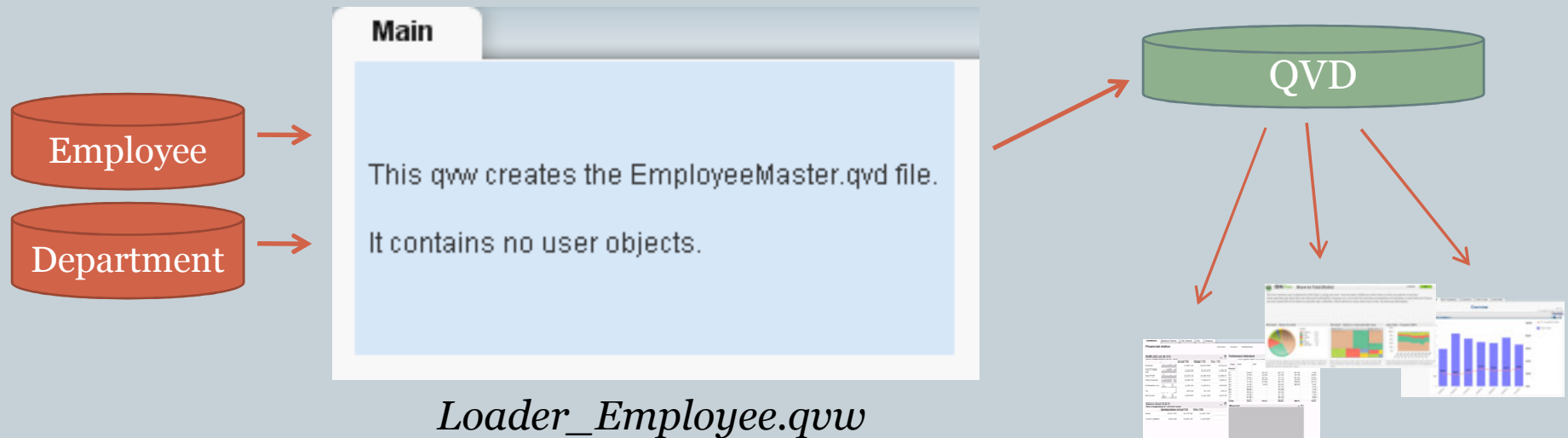
Creating QVD - STORE

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



HrDashboard.qvw

- More typically done in special purpose “Loader” qvw



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



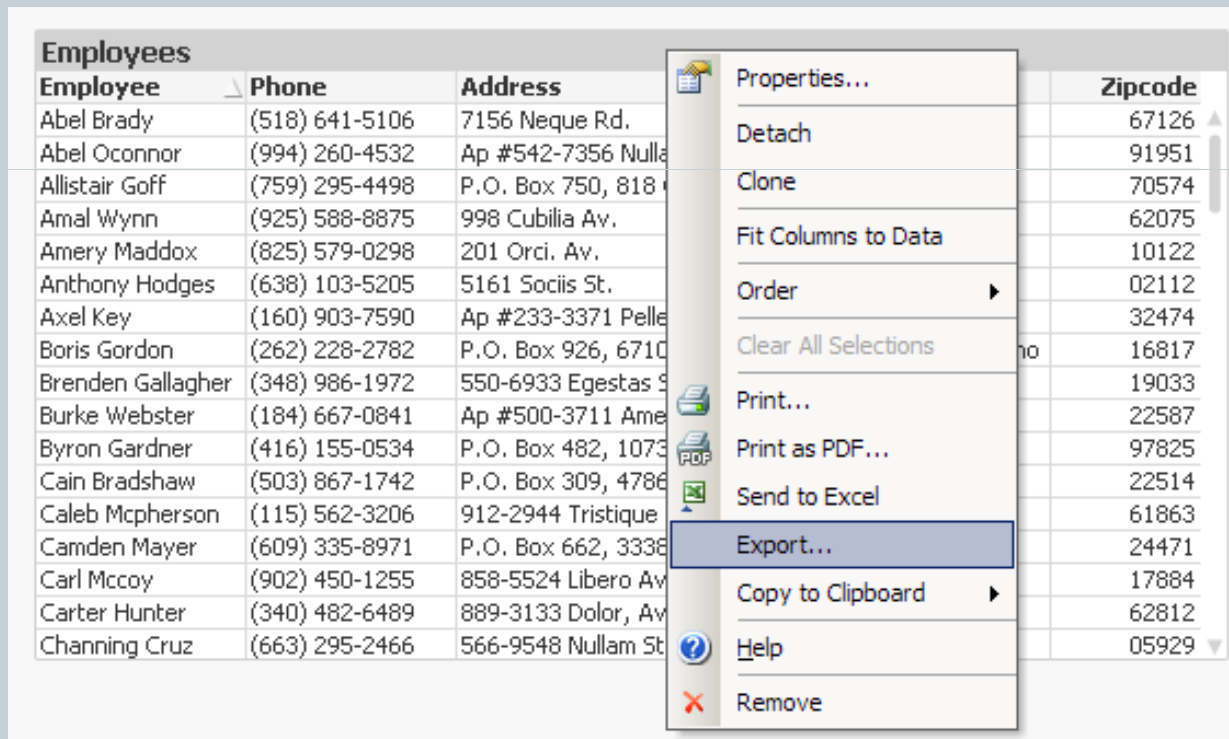
- BUFFER prefix on LOAD signifies that a QVD should be automatically created on first execution, and read from on subsequent executions.

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



Employee	Phone	Address	Zipcode
Abel Brady	(518) 641-5106	7156 Neque Rd.	67126
Abel Oconnor	(994) 260-4532	Ap #542-7356 Nulla	91951
Allistair Goff	(759) 295-4498	P.O. Box 750, 818	70574
Amal Wynn	(925) 588-8875	998 Cubilia Av.	62075
Amery Maddox	(825) 579-0298	201 Orci. Av.	10122
Anthony Hodges	(638) 103-5205	5161 Sociis St.	02112
Axel Key	(160) 903-7590	Ap #233-3371 Pelle	32474
Boris Gordon	(262) 228-2782	P.O. Box 926, 6710	16817
Brenden Gallagher	(348) 986-1972	550-6933 Egestas S	19033
Burke Webster	(184) 667-0841	Ap #500-3711 Ame	22587
Byron Gardner	(416) 155-0534	P.O. Box 482, 1073	97825
Cain Bradshaw	(503) 867-1742	P.O. Box 309, 4786	22514
Caleb Mcpherson	(115) 562-3206	912-2944 Tristique	61863
Camden Mayer	(609) 335-8971	P.O. Box 662, 3338	24471
Carl Mccoy	(902) 450-1255	858-5524 Libero Av	17884
Carter Hunter	(340) 482-6489	889-3133 Dolor, Av	62812
Channing Cruz	(663) 295-2466	566-9548 Nullam St	05929

Creating QVD – UI Export



Main

Employee	Phone
Abel Brady	(518)
Abel Oconnor	(994)
Allistair Goff	(759)
Amal Wynn	(925)
Amery Maddox	(825)
Anthony Hodges	(638)
Axel Key	(160)
Boris Gordon	(262)
Brenden Gallagher	(348)
Burke Webster	(184)
Byron Gardner	(416)
Cain Bradshaw	(503)
Caleb Mcpherson	(115)
Camden Mayer	(609)
Carl Mccoy	(902)
Carter Hunter	(340)
Channing Cruz	(663)

Save As [?] [X]

Save in: export

My Recent Documents

Desktop

My Documents

My Computer

My Network Places

File name:

Save as type: QlikView Export Files (*.qvo)

- QlikView Export Files (*.qvo)
- Comma Delimited (*.csv,*.txt)
- Semicolon Delimited (*.skv,*.txt)
- Tab Delimited (*.tab,*.txt)
- Hypertext (*.html,*.htm)
- XML (*.xml)
- Excel (*.xls)
- QlikView Data File (*.qvd)**

Save

Cancel

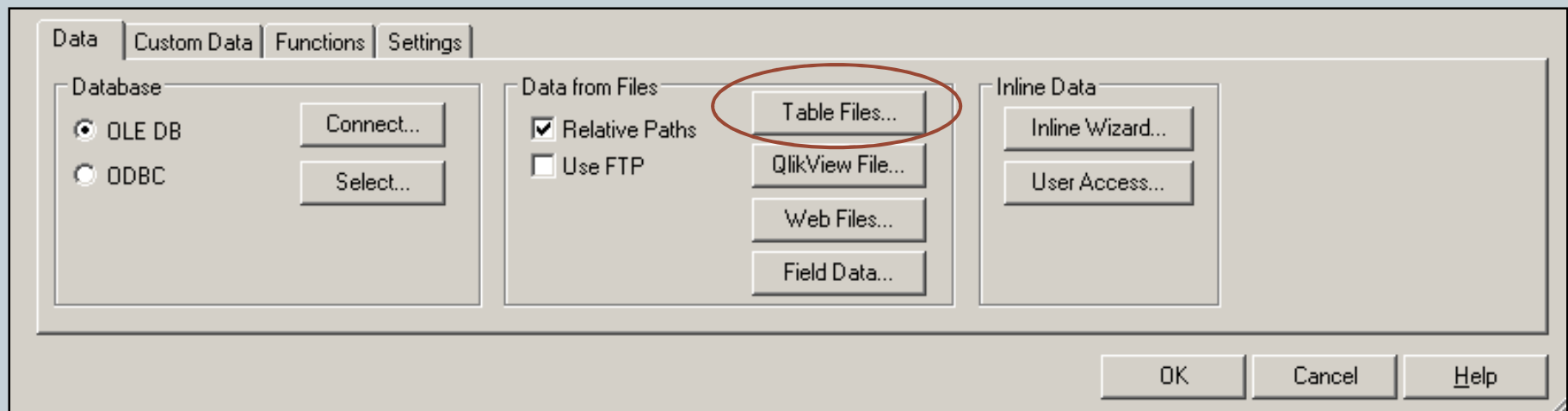
Loading QVDs



- QVDs may be loaded
 - Explicitly – **LOAD * FROM data.qvd (qvd) ;**
 - Implicitly – **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 omitted, 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;

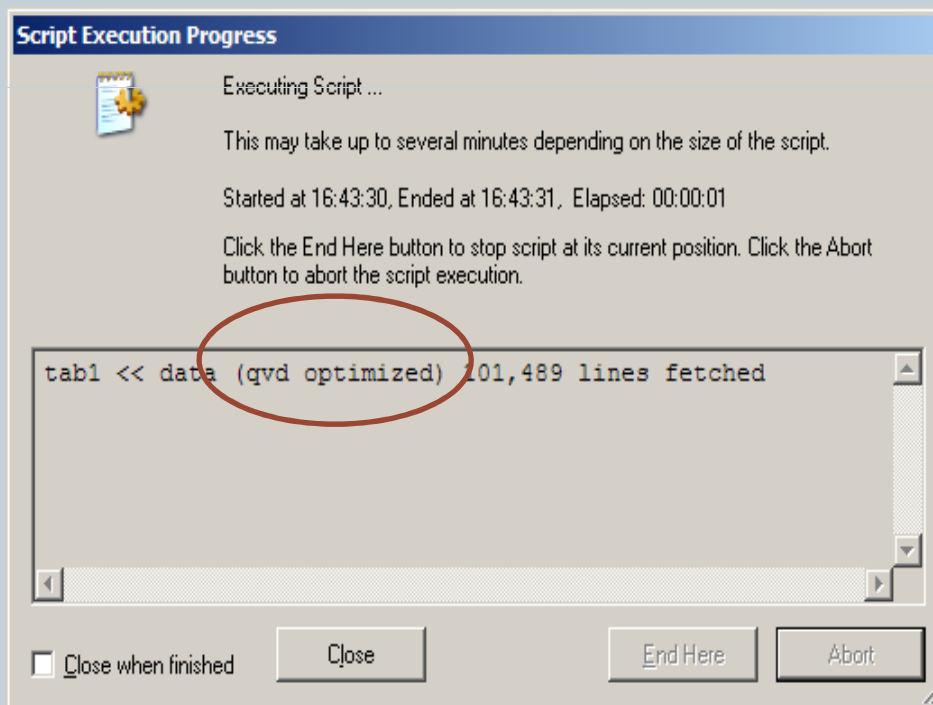
Customers:

// 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.



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		Field Occurance							
QVD Field Name		QVD Name	Categories	Customers	Employees	EmployeeTerritories	Invoices	OrderDetails	Orders
Address	4	Address		X	X		X		
BirthDate	1	BirthDate			X				
CategoryID	2	CategoryID	X						
CategoryName	1	CategoryName	X						
City	4	City		X	X		X		
CompanyName	3	CompanyName		X					
ContactName	2	ContactName		X					
ContactTitle	2	ContactTitle		X					
Country	4	Country		X	X		X		
CustomerID	3	CustomerID		X			X		
CustomerName	1	CustomerName					X		
Description	1	Description	X						
Discontinued	1	Discontinued							
Discount	2	Discount					X	X	
EmployeeID	3	EmployeeID			X	X			
ExtendedPrice	1	ExtendedPrice					X		
Extension	1	Extension			X				
Fax	2	Fax		X					
FirstName	1	FirstName			X				
Freight	2	Freight					X		
HireDate	1	HireDate			X				
HomePage	1	HomePage							
HomePhone	1	HomePhone							
LastName	1	LastName							

QVX



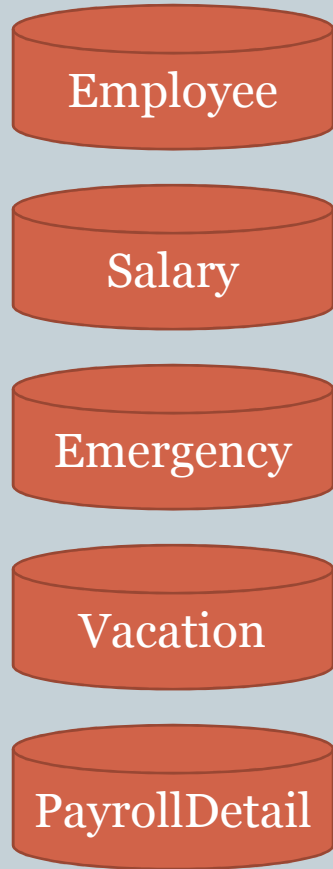
- New with V10 -- **QlikView** data e**X**change
- “Open variant of QVD”
- Not quite as fast as QVD
- May be used to create high performance input files from sources where ODBC drivers are not available.
- Read by standard LOAD script statement.
- MetaFunctions --**QvdNoOfFields(*filename*)**



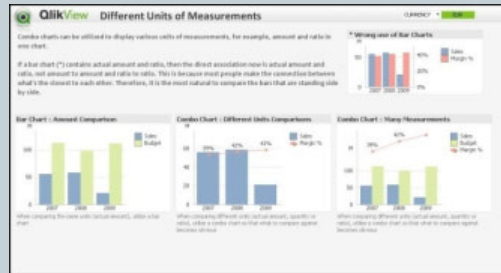
ARCHITECTURES

Basic “Single Tier” Model

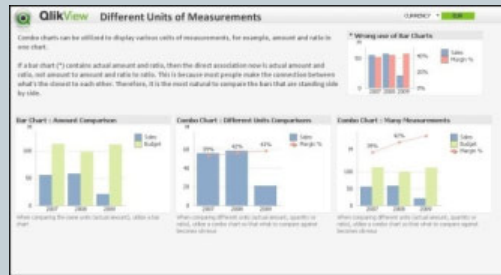
Datasource



HrDashboard.qvw



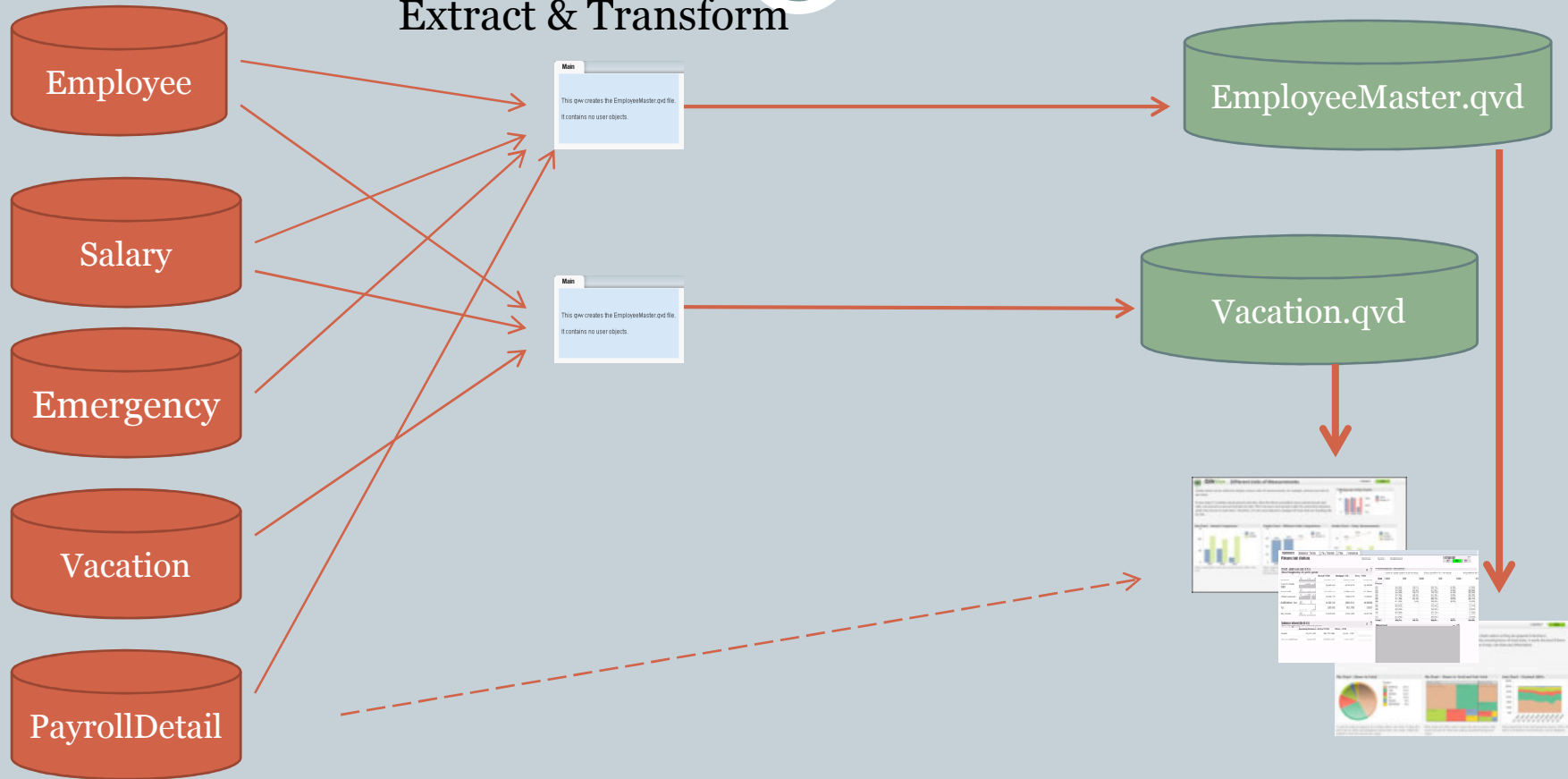
Vacation.qvw



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

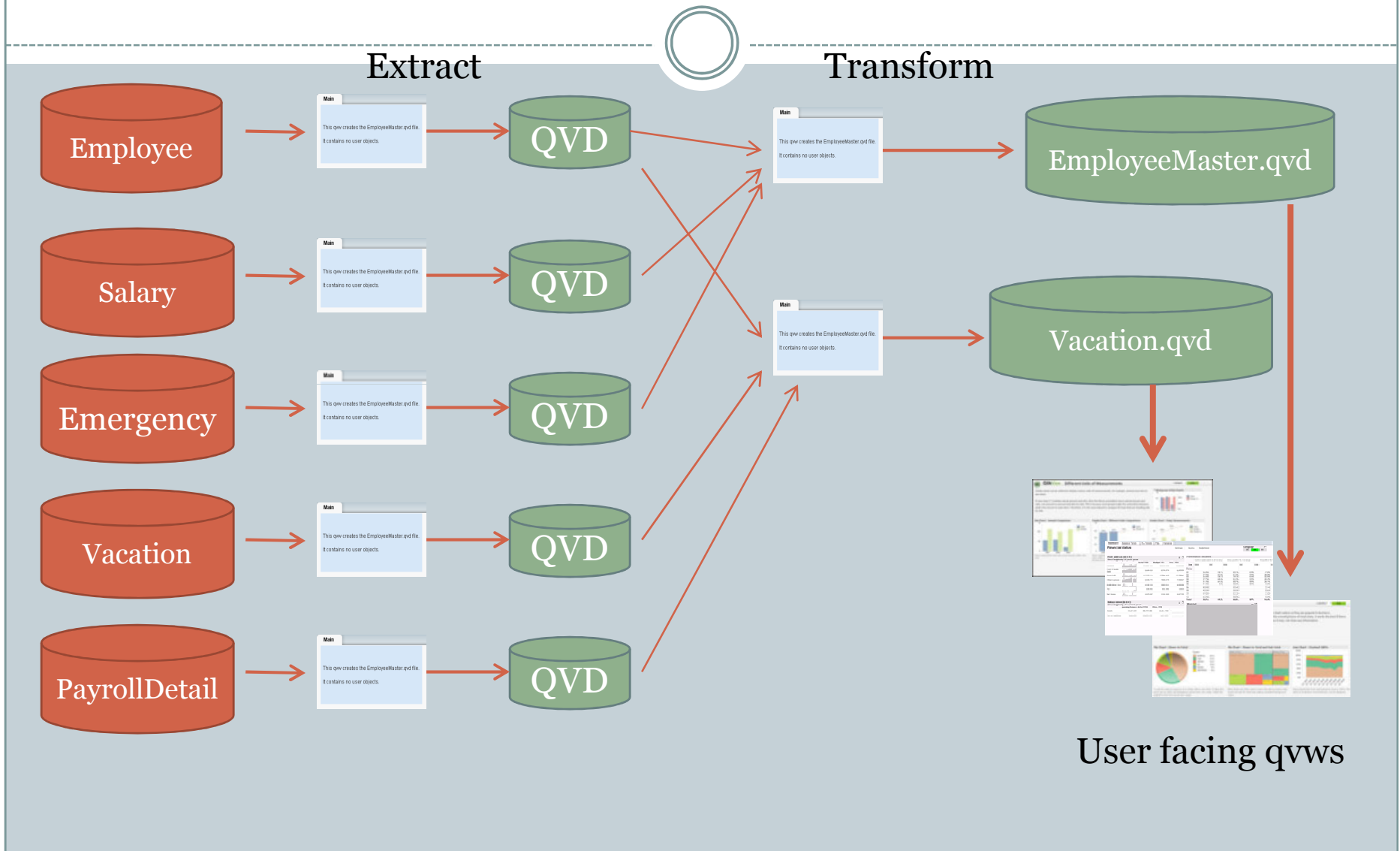
Two Tier Model

Extract & Transform



User facing qvws

Three Tier Model



Bonus -- QlikMarts

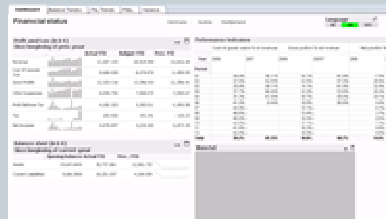
QVWs may also be consumed as datasources.
 Script: BINARY Vacation.qvw;



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

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