Best Practices in Data Modeling

Dan English

QlikView

Objectives

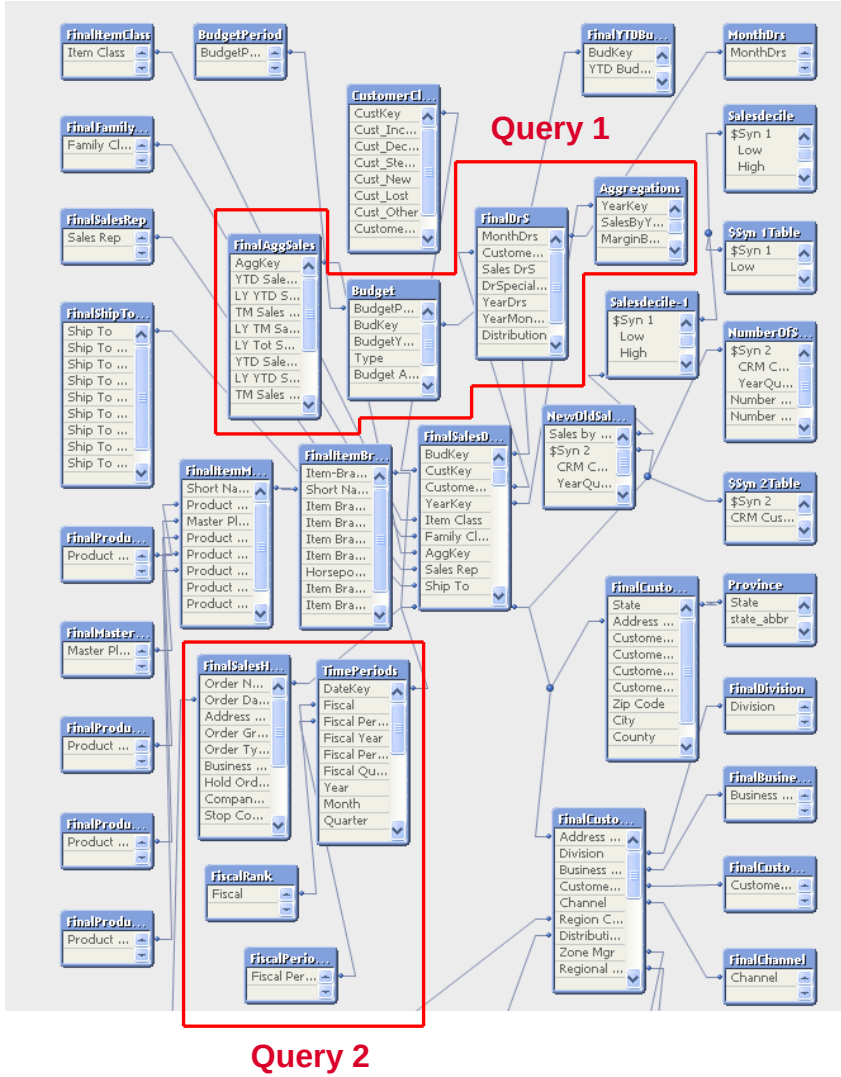
- Understand how QlikView is Different from SQL
- Understand How QlikView works with(out) a Data Warehouse
- Not Throw Baby out with the Bathwater
- Adopt Applicable Data Modeling Best Practices
- Know Where to Go for More Information

QlikView is not SQL (SQL Schemas)

SQL take a large schema and queries a subset of tables.

Each query creates a temporary "Schema" of only a few tables.

Query result sets are independent of each other.

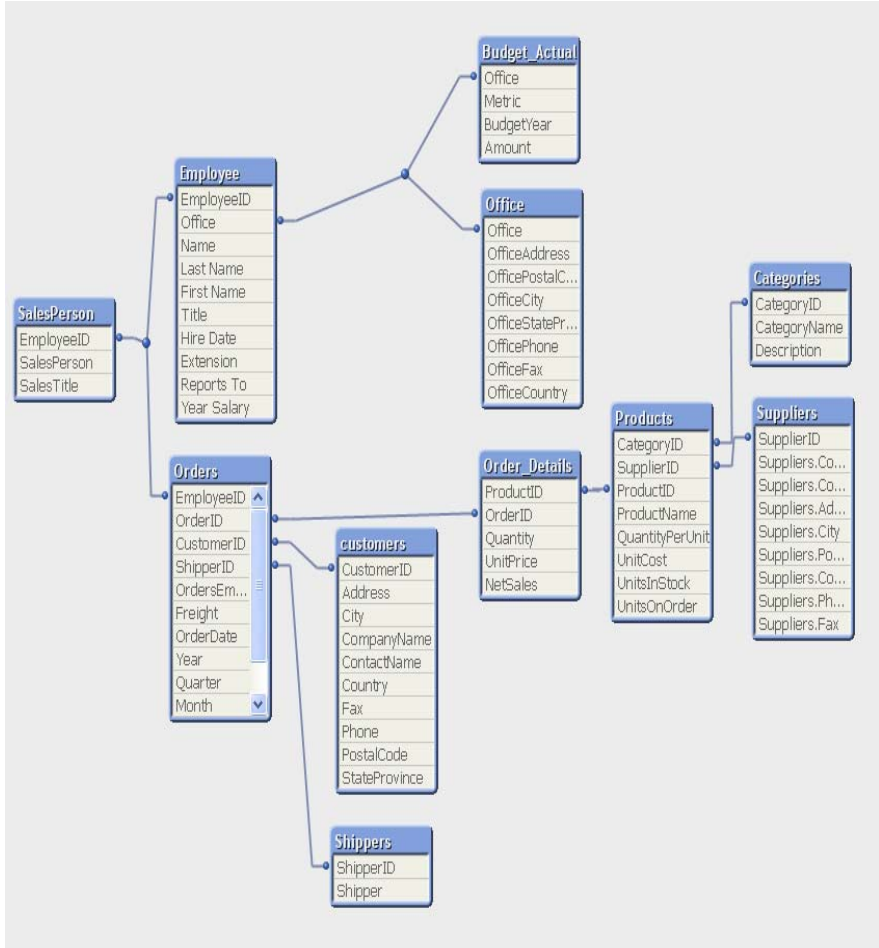


QlikView is not SQL (QV Schemas)

QlikView builds a smaller and more reporting friendly schema from the transactional database.

This schema is persistent and reacts as a whole to user “queries”.

A selection affects the entire schema.



QlikView is not SQL (Aggregation and Granularity)

Store Table

Store	SqrFootage
A	1000
B	800

Sales Table

Store	Prod	Price	Date
A	1	$1.25	1/1/2006
A	2	$0.75	1/2/2006
A	3	$2.50	1/3/2006
B	1	$1.25	1/4/2006
B	2	$0.75	1/5/2006

Select * From Store, Sales Where Store.Store = Sales.Store will return:

SqrFootage	Store	Prod	Price	Date
1000	A	1	$1.25	1/1/2006
1000	A	2	$0.75	1/1/2006
1000	A	3	$2.50	1/1/2006
800	B	1	$1.25	1/1/2006
800	B	2	$0.75	1/1/2006

Sum(SqrFootage) will return: **4600**

If you want the accurate Sum of SqrFootage in SQL you can not join on the Sales table in the same Query!

QlikView is not SQL (Benefits)

- QlikView allows you to see the results of a selection across the entire schema not just a limited subset of tables.

QlikView is not SQL (Benefits)

- QlikView allows you to see the results of a selection across the entire schema not just a limited subset of tables.

- QlikView will aggregate at the lowest level of granularity in the *expression* not the lowest level of granularity in the *schema* (query) like SQL.

QlikView is not SQL (Benefits)

- QlikView allows you to see the results of a selection across the entire schema not just a limited subset of tables.
- QlikView will aggregate at the lowest level of granularity in the *expression* not the lowest level of granularity in the *schema* (query) like SQL.
- This means that QlikView will allow a user to interact with a broader range of data than will ever be possible in SQL!

QlikView is not SQL (Challenges)

- Several SQL queries can join different tables together in completely different manners.
- In QlikView there is only ever One way tables join in any one QlikView file.
- This means that Schema design is *much* more important in QlikView!

A Word about Requirements

- Requirements will always inform your schema design.

A Word about Requirements

- Requirements will always inform your schema design.

- If you do not fully understand your requirements and these requirements are not thoroughly documented you are not ready to begin scripting. *No exceptions!*

A Word about Requirements

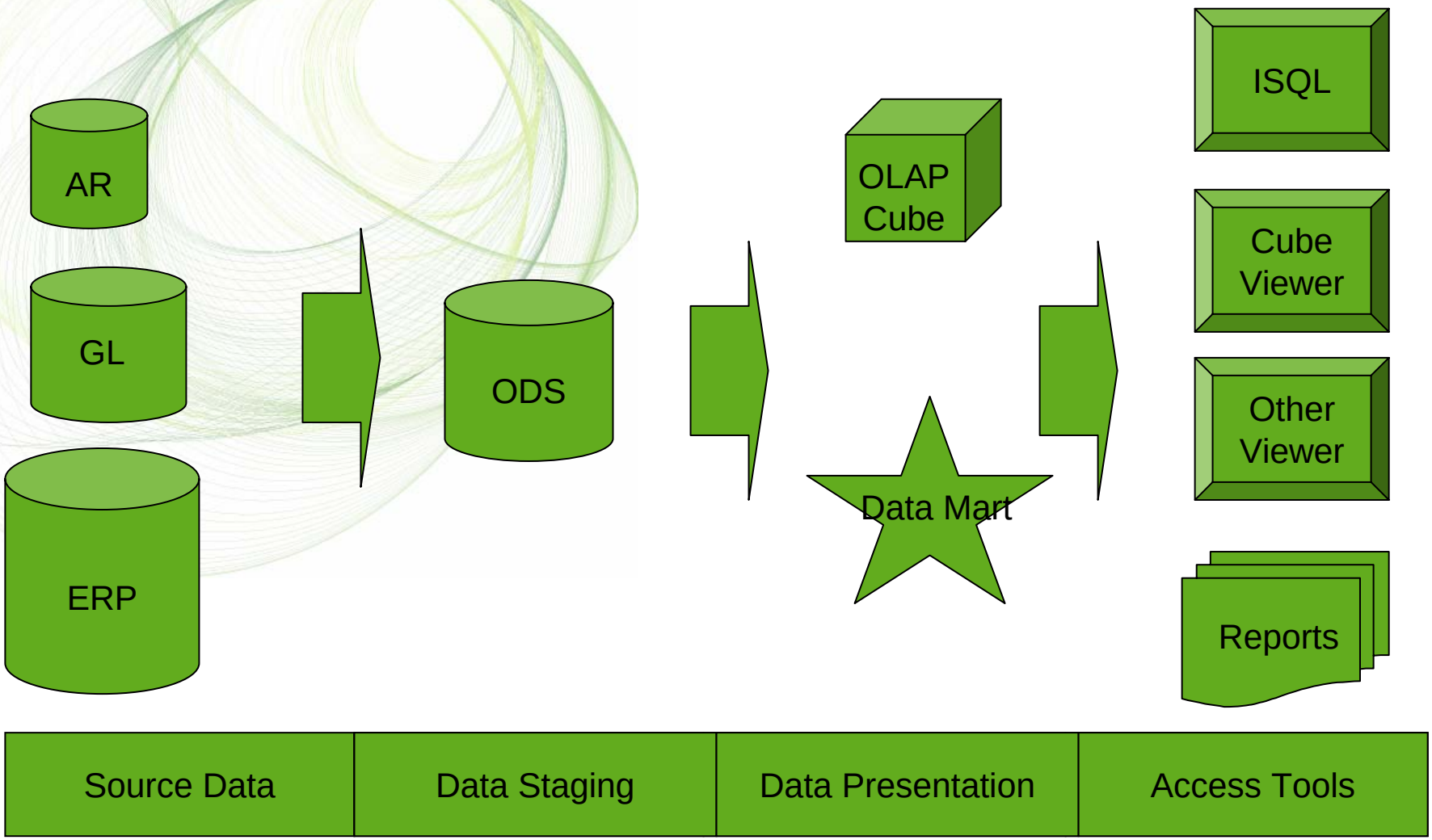
- Requirements will always inform your schema design.
- If you do not fully understand your requirements and these requirements are not thoroughly documented you are not ready to begin scripting. *No exceptions.*
- Requirements are focused in the problem domain; not the solution domain.

A Word about Requirements

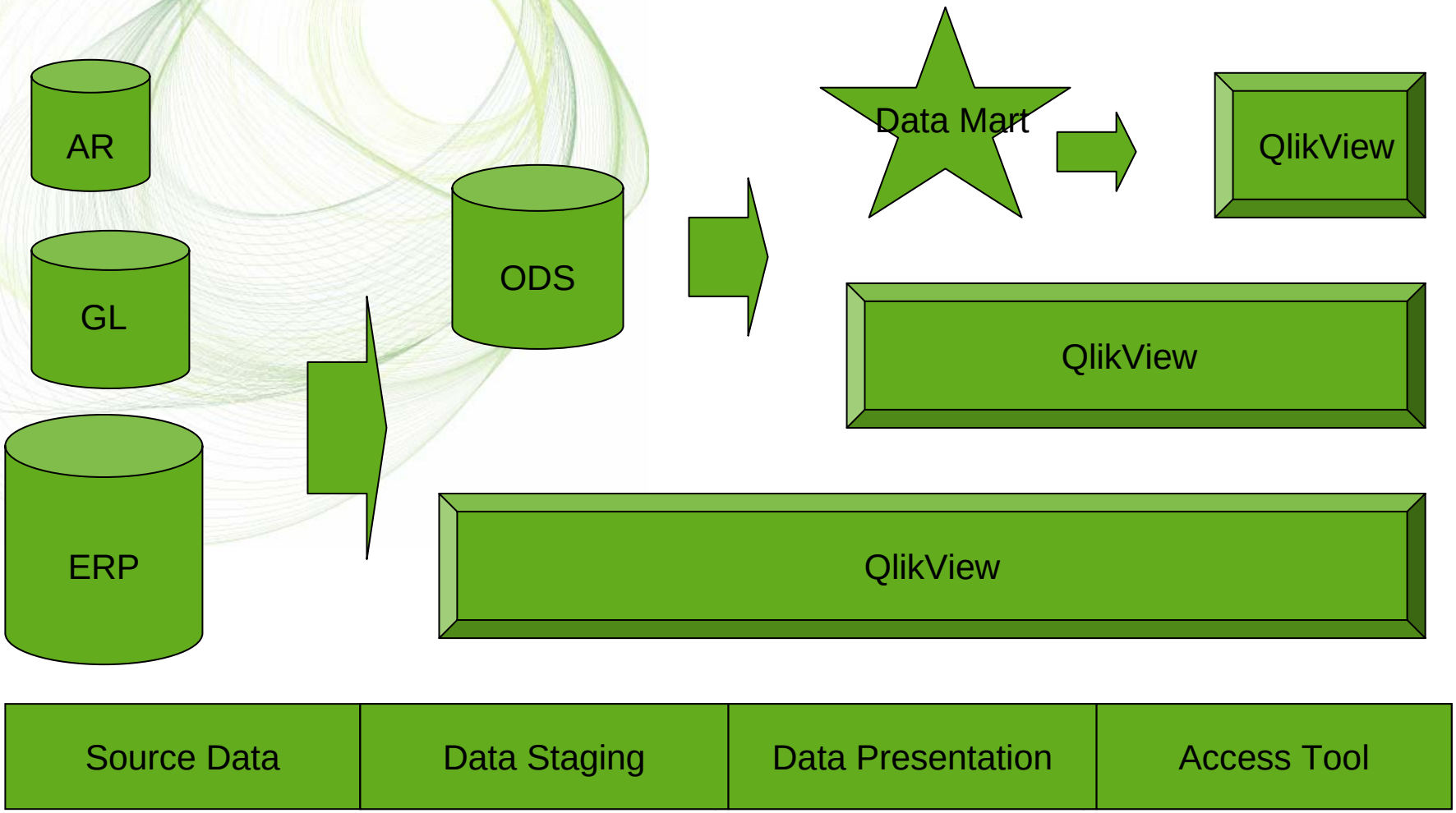
- Requirements will always inform your schema design.
- If you do not fully understand your requirements and these requirements are not thoroughly documented you are not ready to begin scripting. *No exceptions.*
- Requirements are focused in the problem domain; not the solution domain.
- Most Schema design questions are not really schema design questions they are really requirements questions.

The Traditional Data Warehouse

QlikView



How QlikView Can Be Used



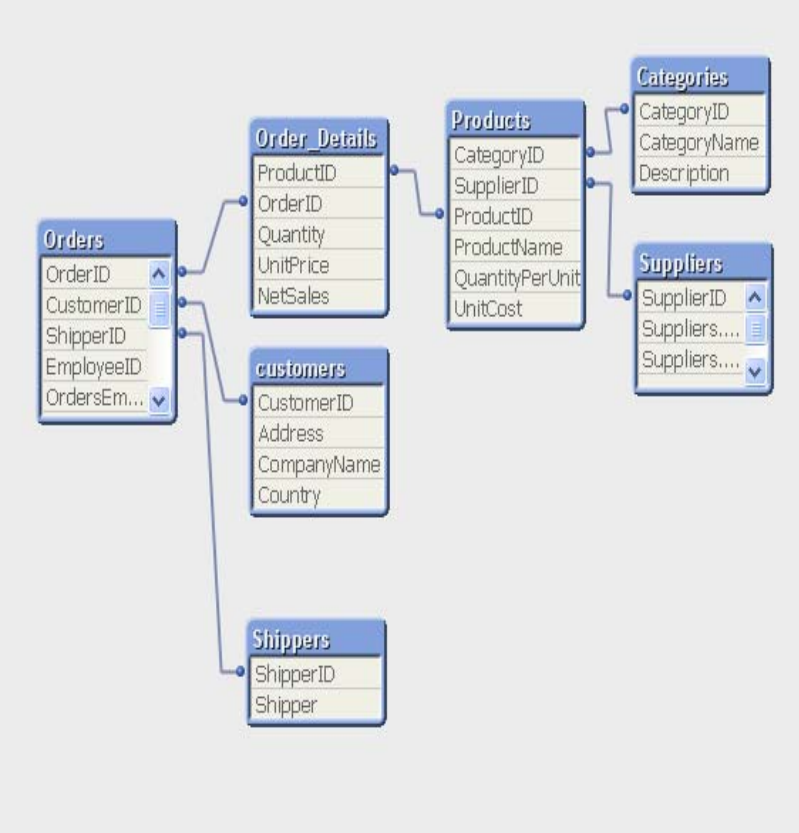
Observations

- There Is No One Best Data Modeling Best Practice.
- Data Modeling Is Entirely Dependant on Requirements
  - Systems, Skill Sets, Security, Functionality, Flexibility, Time, Money, and Above all… Business Requirements!
- Likewise Best Practices are not Universal
- Apply Best Practices Situationaly
- Sometimes (Gasp!) even QlikView may not be the Right Tool

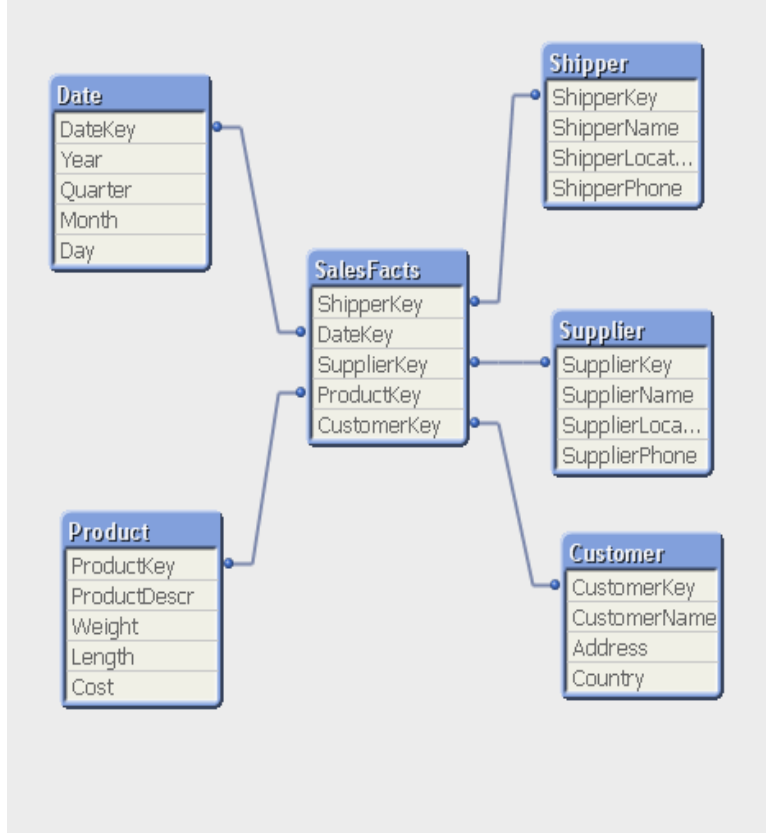
Relational vs. Dimensional Modeling

QlikView

Relational



Dimensional



Relational vs. Dimensional Modeling

QlikView

Relational

- Complex Schemas
- Efficient Data Storage
- Schema Quicker to Build
- Schema Easier to Maintain
- Queries More Complicated
- Confuses End Users

Dimensional

- Simpler Schemas
- Less Normalized
- Schema Complex to Build
- Schema Complex to Maintain
- Simpler Queries
- Understood by End-users

4 Steps to Dimensional Modeling

QlikView

1. Select the business process to model.

2. Declare the grain of the business process.

   Ex. One trip, One Segment, One Flight, One historical booking record

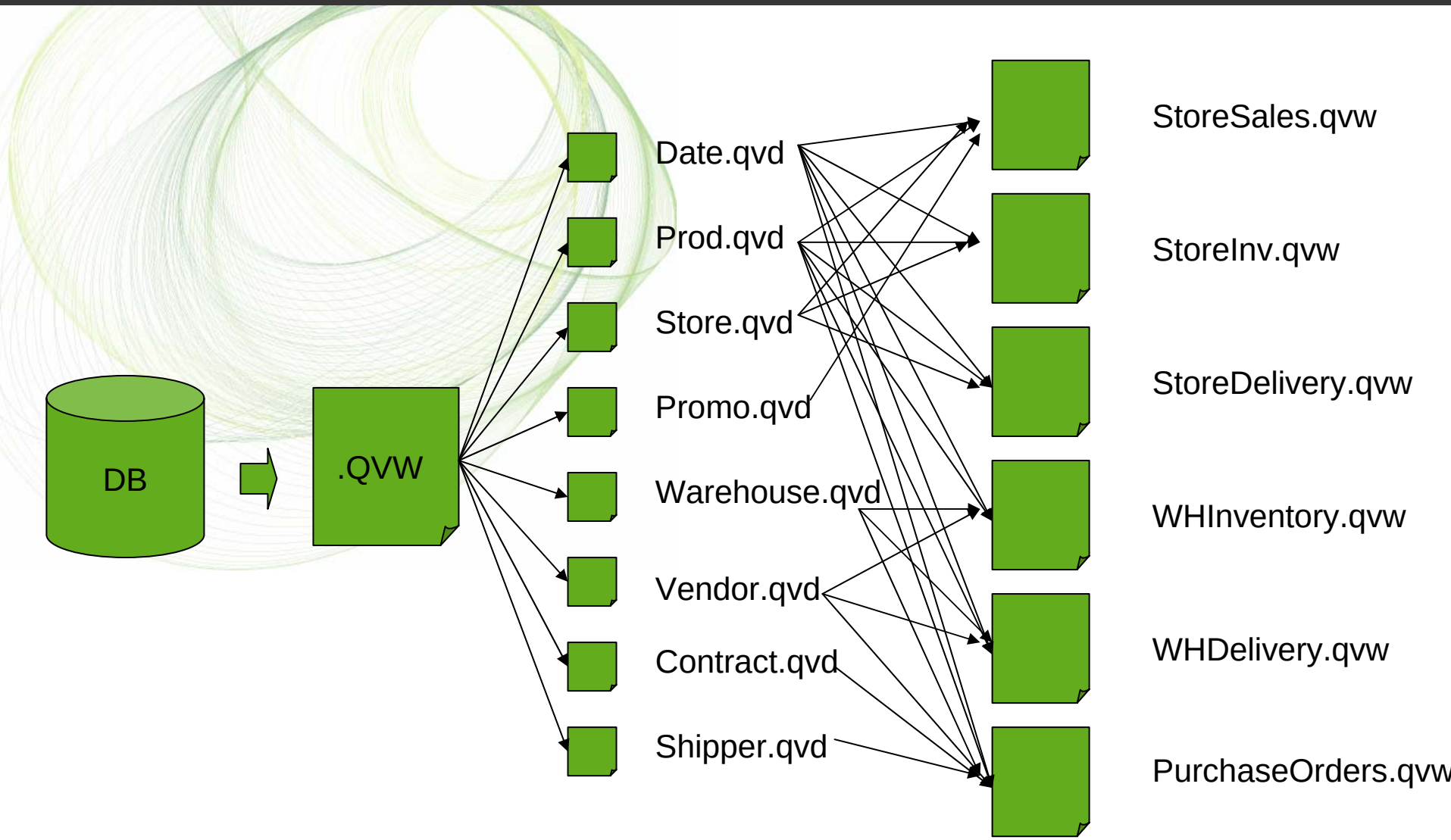
3. Choose the dimensions that apply to each fact table row.

4. Identify the numeric facts that will populate each fact table row.

Multiple Star Schemas and Conformed Dimensions

	Common Dimensions							
Business Process	Date	Product	Store	Promotion	Warehouse	Vendor	Contract	Shipper
Store Sales	X	X	X	X				
Store Inventory	X	X	X					
Store Deliveries	X	X	X					
Warehouse Inventory	X	X			X	X		
Warehouse Delivery	X	X			X	X		
Purchase Orders	X	X			X	X	X	X

Using QVD Files to Conform Dimensions



Slowly Changing Dimensions

- Dimension values change over time in relationship to each other.
- Classic example: Sales Force Territory Reorganization
- Postal code 24829 was in territory A1 but as of June 1st 2006 it moved to territory D3.

Slowly Changing Dimensions

Three way to deal with this

1. Overwrite Original Value

   Very Easy - Now all sales for 28429 roll into D3 regardless of date

2. Add a Dimension Row (requires surrogate key)

   Preserves history

FakeKey	PostalCode	TerrID
123	28429	A1
124	28429	D3

3. Add a Dimension Field

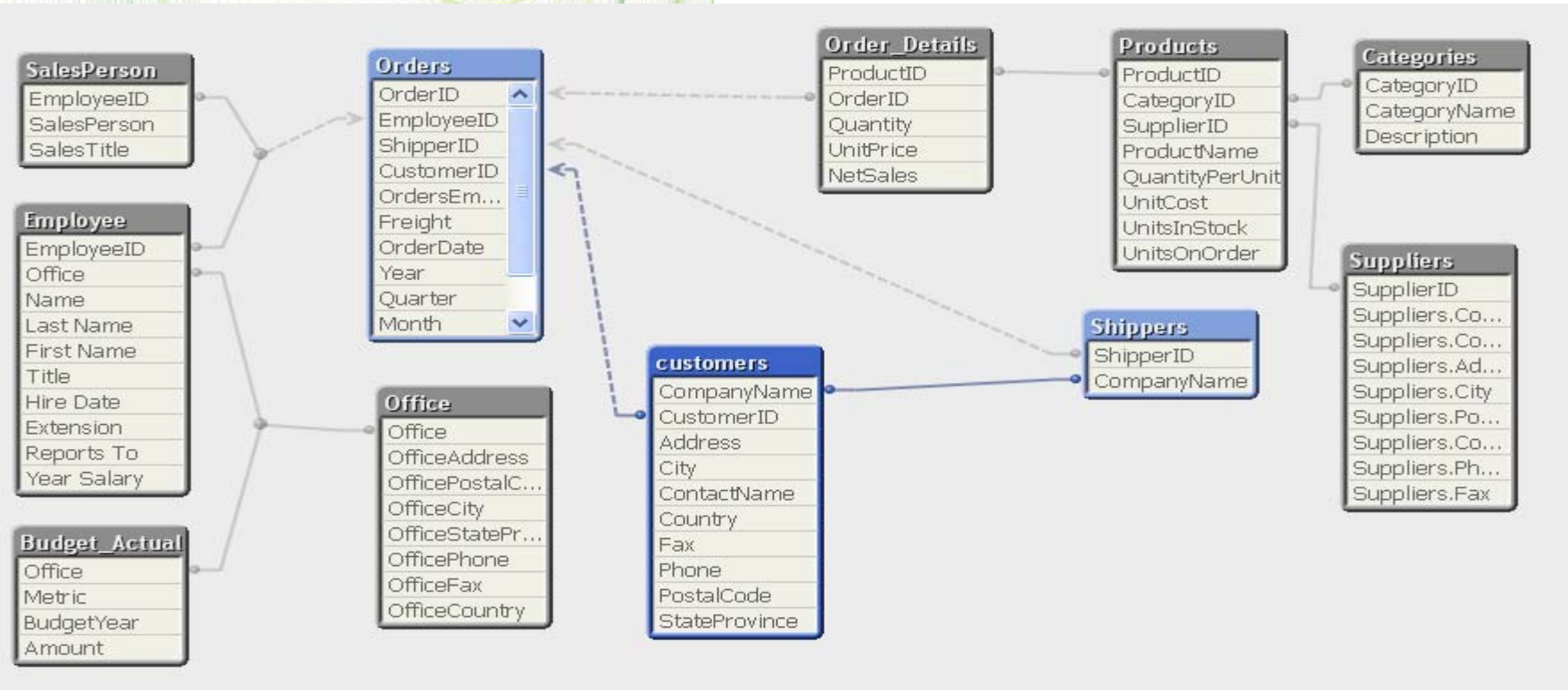
   Allows Comparison

FakeKey	PostalCode	TerrID	OldTerrID
123	28429	D3	A1

Possible to combine solutions

Circular References

Anytime you enclose area in the table viewer you will encounter a circular reference.



Circular References

Circular References are common in QlikView because you get only one set of join relationships per QlikView file.

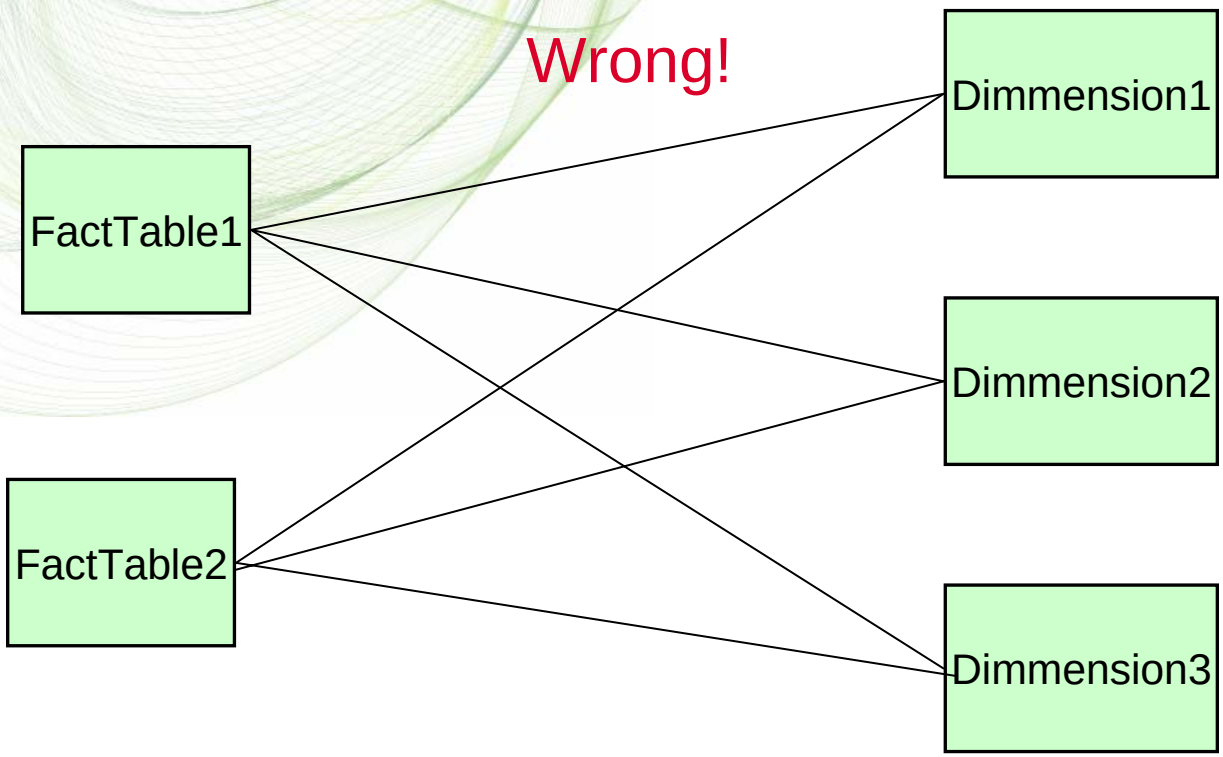
When you get a circular reference ask yourself if you could live without one of the joins. If you can, cut it.

Otherwise you may have to resort to concatenation or a link table to remove the circular reference.

Don’t kill yourself with technical link tables if you don’t have to!

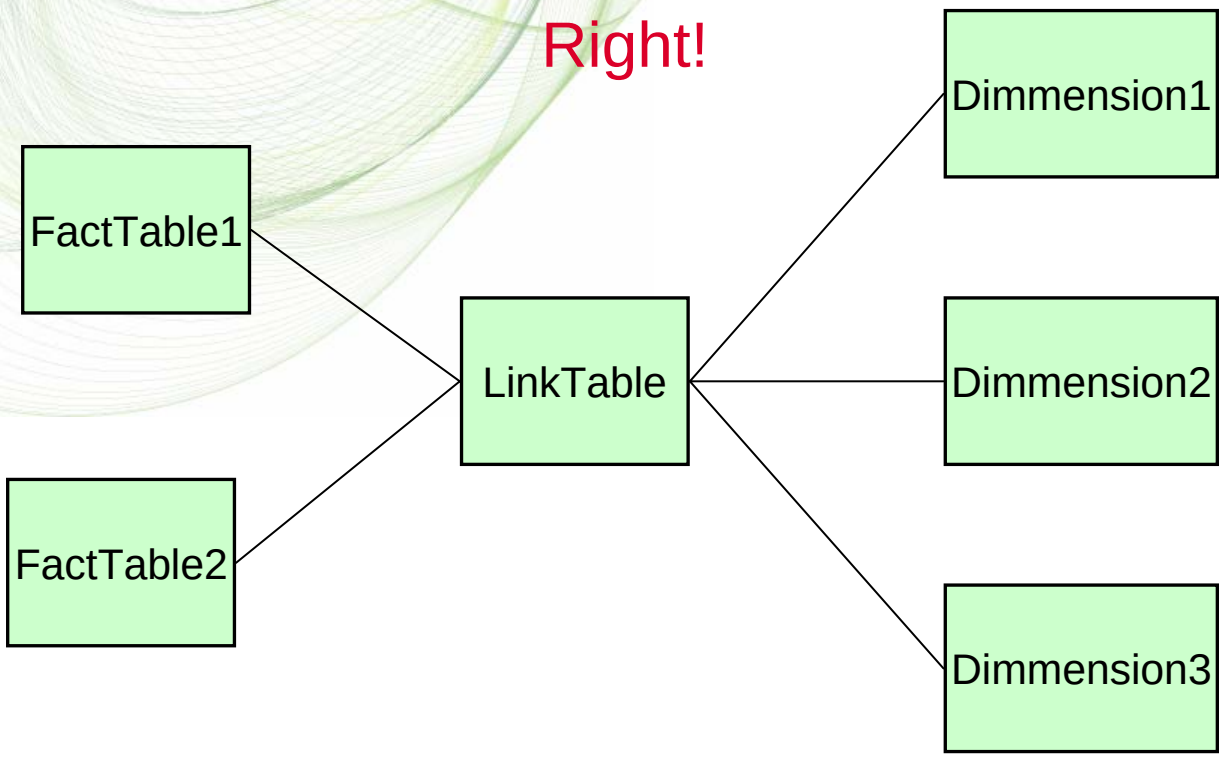
Link Tables

Link tables essentially allow you to join two or more fact tables against a common set of dimensions without the usual circular references.



Link Tables

Link tables essentially allow you to join two or more fact tables against a common set of dimensions without the usual circular references.



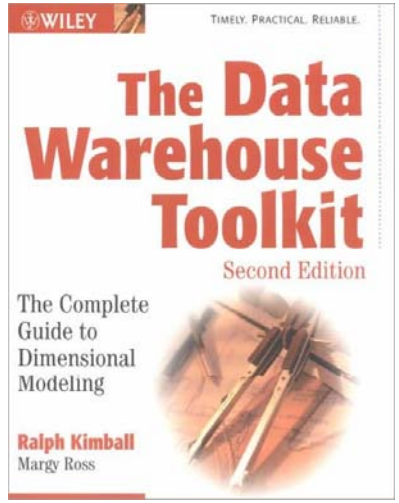
Last Words

- If your end users reject your application then you have failed, regardless of your technical execution.
- End user requirements and end user experience should always dictate your approach to developing QlikView applications, including data modeling.
- Many data warehousing techniques and best practices are directly applicable to QlikView data modeling.
- Data modeling had been ongoing for many years brilliant minds have contributed to the field; we don’t always need to reinvent the wheel.

Recommended Resources

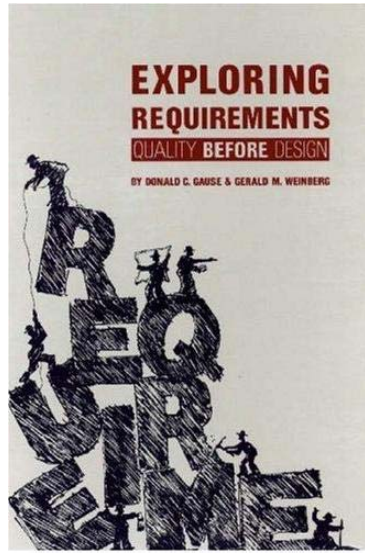
Data Modeling:

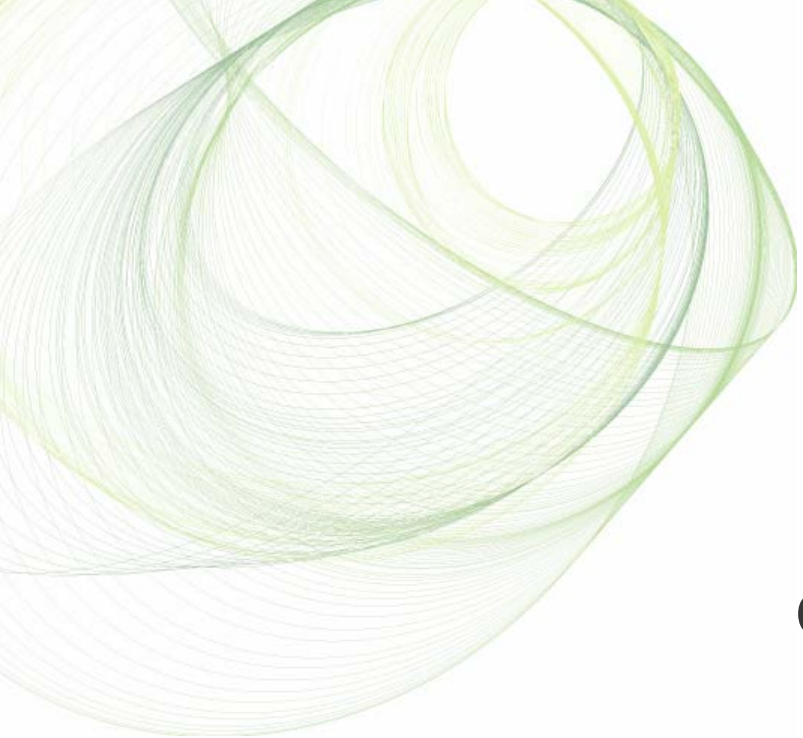
The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd Edition) – Ralph Kimball, Margy Ross – Wiley – ISBN: 0471200247



Requirements Gathering:

Exploring Requirements: Quality before Design – Donald C. Gause, Gerald M. Weinberg – Dorset House - ISBN: 0932633137





QlikView

Questions?

Thank You!

Dan English