NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

3.0 Introduction to Dimensional Modeling

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

Module 2 of the Business Intelligence and Analytics Track of UP NEC and the UP Center of Business Intelligence

Outline for This Training

1. Introduction to Data Warehousing
2. DW Lifecycle and Project Management
   - Case Study on DW PM
3. **Dimensional Modeling**
4. Designing Fact Tables
5. Designing Dimension Tables
   - Case Study on Dimension Modeling
6. Extraction Transformation and Loading
   - Case Study on ETL Planning
7. Transformation and Loading Methodologies
   - Case Study on ETL

Outline for This Session

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Examples of Dimensional Modeling
- Fact and Dimension Tables
- Designing the Dimension Model

Inmon versus Kimball Paradigm

- Two Models for Data Warehouses
  - Imnon Model
  - Kimball Model

Inmon versus Kimball Paradigm

- Inmon Model
  - Consists of all databases and information systems in an organization
  - Also called the CIF (Corporate Information Factory)
  - Defines overall database environment as:
    - Operational
    - Atomic data warehouse
    - Departmental
    - Individual
  - The Warehouse is part of the bigger whole (CIF)

Inmon versus Kimball Paradigm

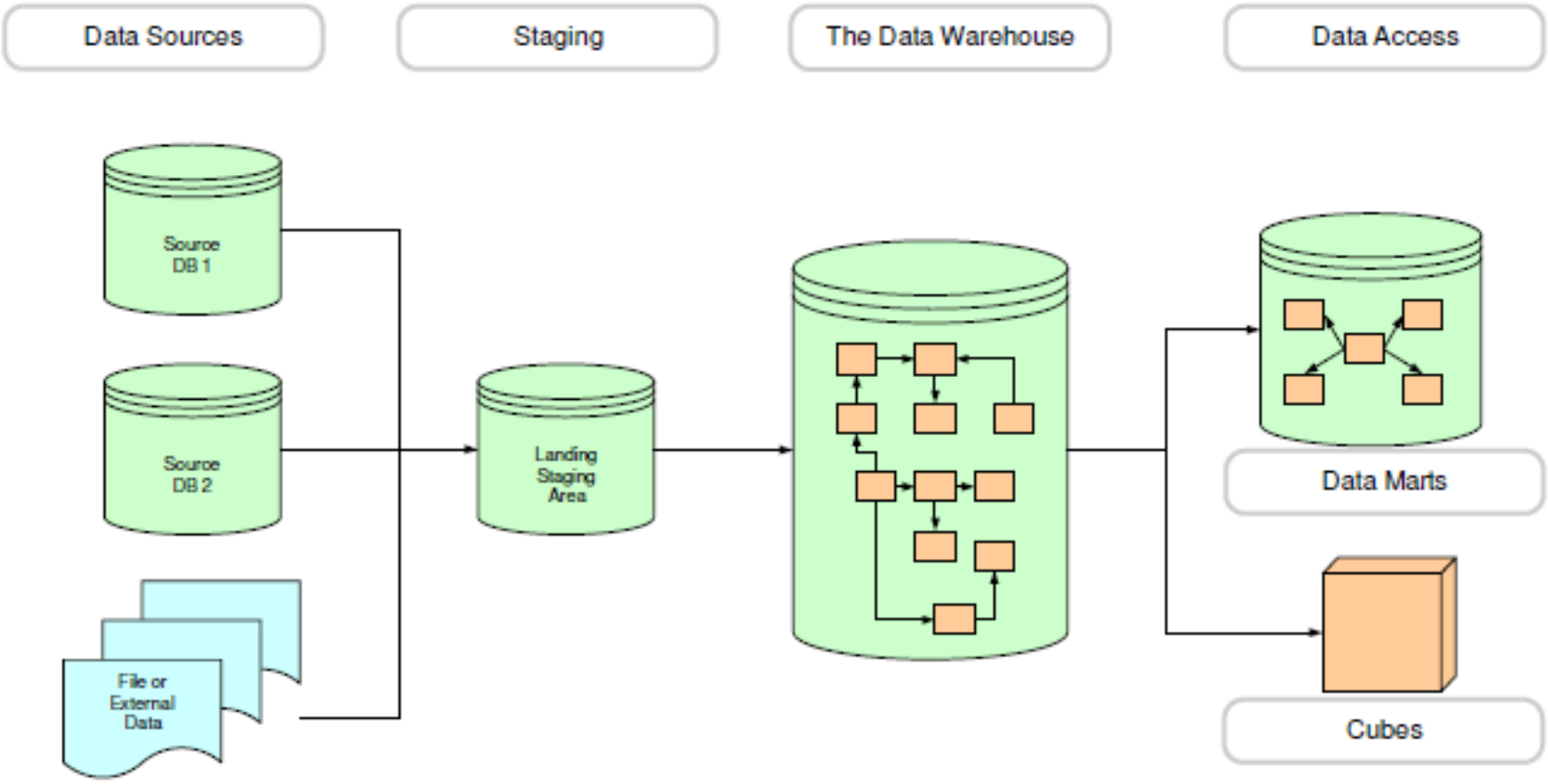


Figure 3.1: Inmon Model

Inmon versus Kimball Paradigm

- Kimball Model
  - The Dimensional Data Model
    - Does not adhere to normalization theory
    - Starts with tables
      - Numeric Tables
      - Context Tables
    - User accessible

Inmon versus Kimball Paradigm

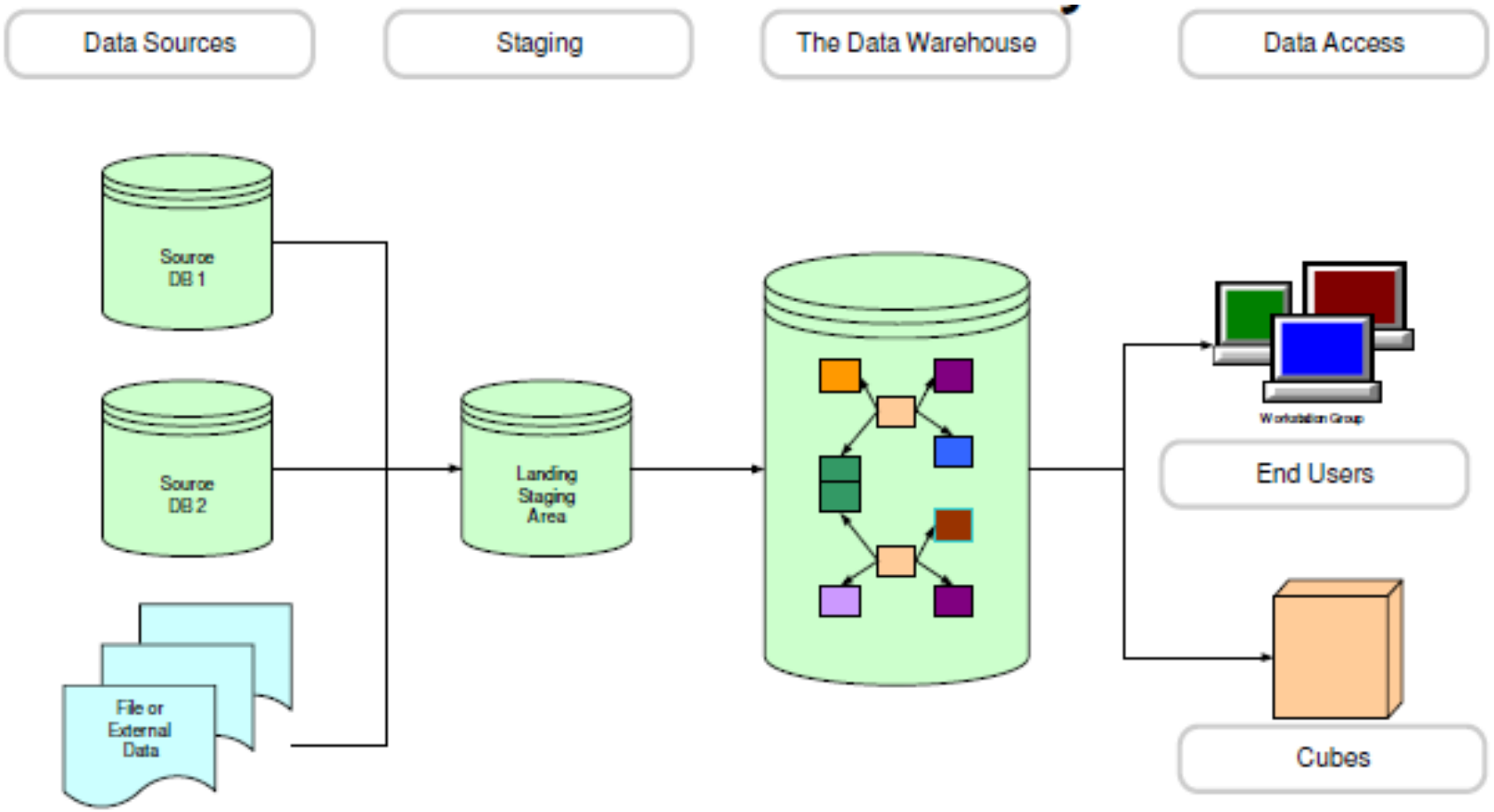


Figure 3.2: Kimball Model

Inmon versus Kimball Paradigm

Table 3.1: Comparison of the Inmon and Kimball Model

	Inmon	Kimball
Overall Approach	Top-Down	Bottom-Up
Complexity of Method	Complex	Simple
Data Orientation	Data Driven	Process Oriented
Tools	Traditional ERDs	Dimensional Modeling
End User Accessibility	Low	High

Inmon versus Kimball Paradigm

Table 3.2: Philosophy Comparison of the Inmon and Kimball Model

	Inmon	Kimball
Primary Audience	IT	End Users
Objective	Deliver a Sound Technical Solution Based on Proven Methods	Deliver a Solution that makes it easy for end users to directly query data

Inmon versus Kimball Paradigm

Table 3.3: How to Choose, Inmon versus Kimball Model?

	Favors Inmon	Favors Kimball
Planning Horizon	Strategic	Tactical
Data Integration Requirements	Enterprise-Wide Integration	Individual Business Areas
Time to Delivery	Longer Start-up Time	Need for First Data Warehouse is Urgent
Cost	Higher start-up costs, with lower subsequent project dev costs	Lower start-up costs with each subsequent project costs the same
Staffing Requirements	Large Teams of Specialists	Small Teams of Generalists

Outline for This Session

- Inmon versus Kimball Paradigm
- **What is Dimensional Modeling?**
- Why not Relational Modeling?
- Examples of Dimensional Modeling
- Fact and Dimension Tables
- Designing the Dimension Model

What is Dimensional Modeling?

- Dimensional modeling is a logical design technique for structuring data so such that
  - It is intuitive for business users
  - And delivers fast query performance.
- Widely accepted as the preferred approach for DW presentation.
- Simplicity is fundamental to usefulness.
- Allows software to easily navigate databases.

What is Dimensional Modeling?

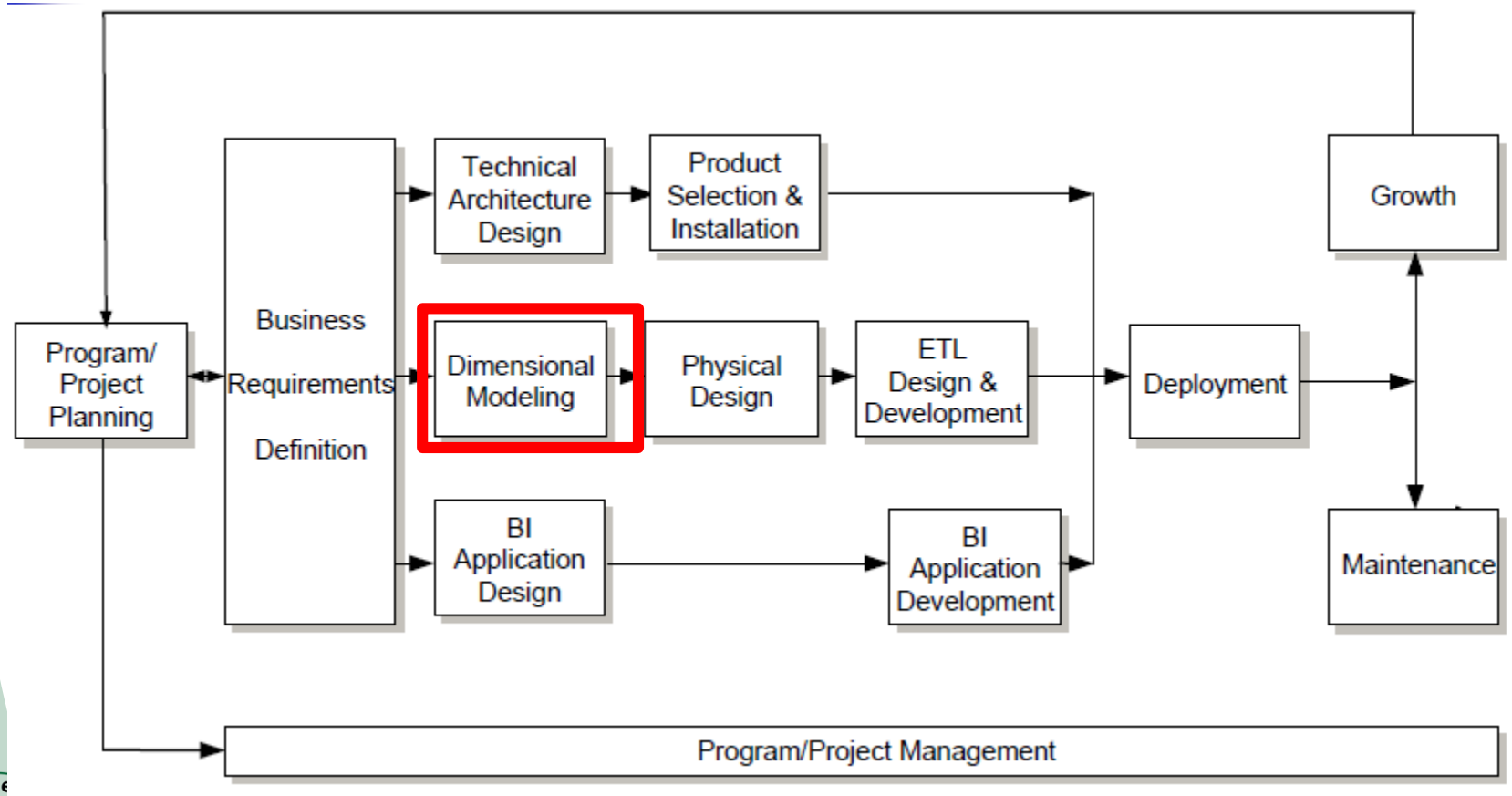


Figure 3.3: The Kimball Lifecycle

What is Dimensional Modeling?

Definition 3.1: Dimensional Modeling

- Divides world into measurements and context.
- Measurements are numeric values called facts.
- Context intuitively divided into clumps called dimensions.
- Dimensions describe the “who, what, where, when, why, and how” of the facts.

What is Dimensional Modeling?

Definition 3.2: Dimensional Model

- A dimensional model consists of a fact table containing measurements surrounded by a halo of dimension tables containing textual context.
- Known as a star join.
- Known as a star schema when stored in a relational database (RDBMS).

What is Dimensional Modeling?

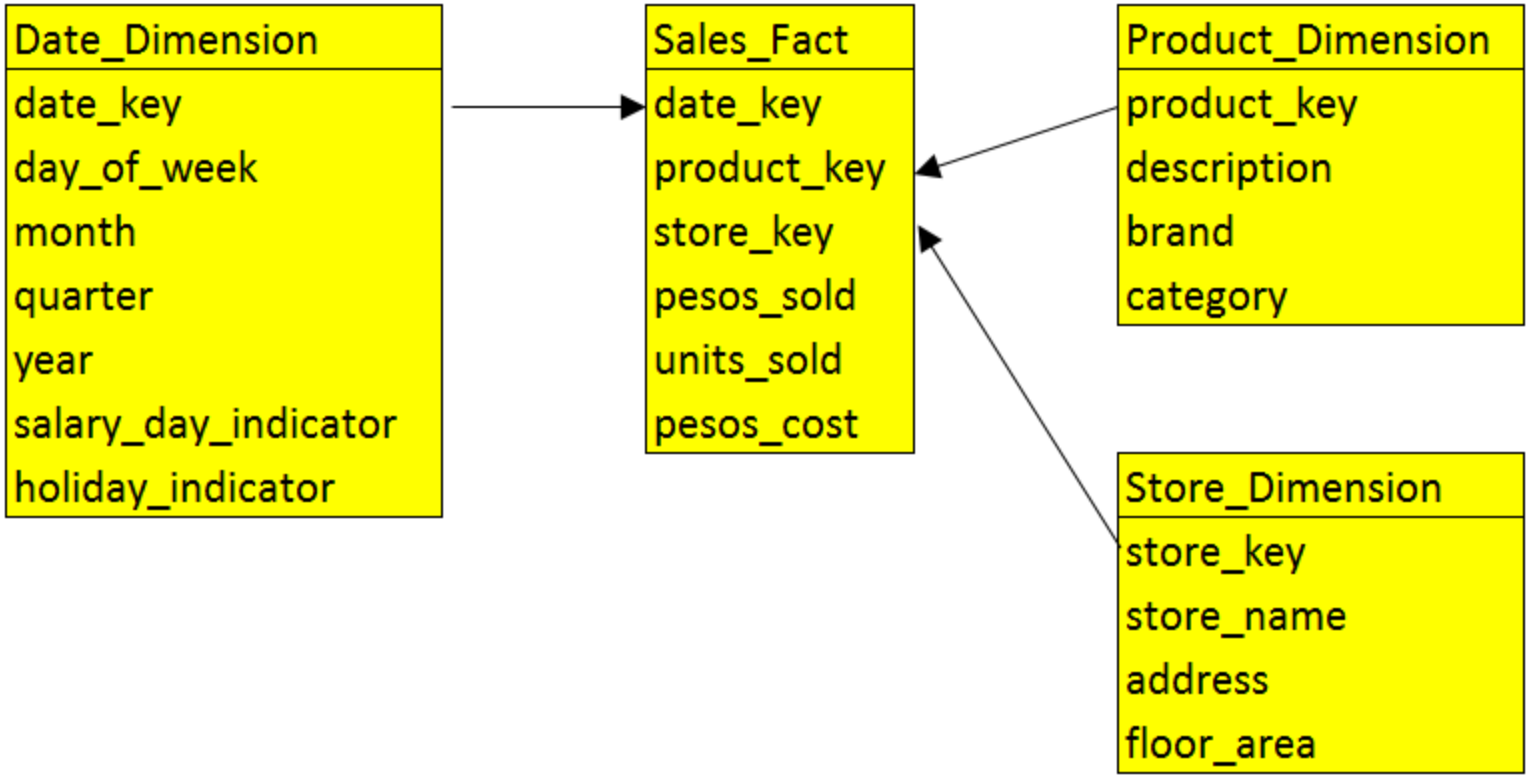


Figure 3.4: Typical Dimensional Model

Standard SQL Query Template

```
SELECT p.brand, sum(f.pesos_sold),
sum(f.units_sold)
FROM sales_fact f, product_dim p, date_dim d
WHERE f.productkey = p.productkey
and f.datekey = d.datekey
and d.quarter = '1 Q 2015'
GROUP BY p.brand
ORDER BY p.brand
```

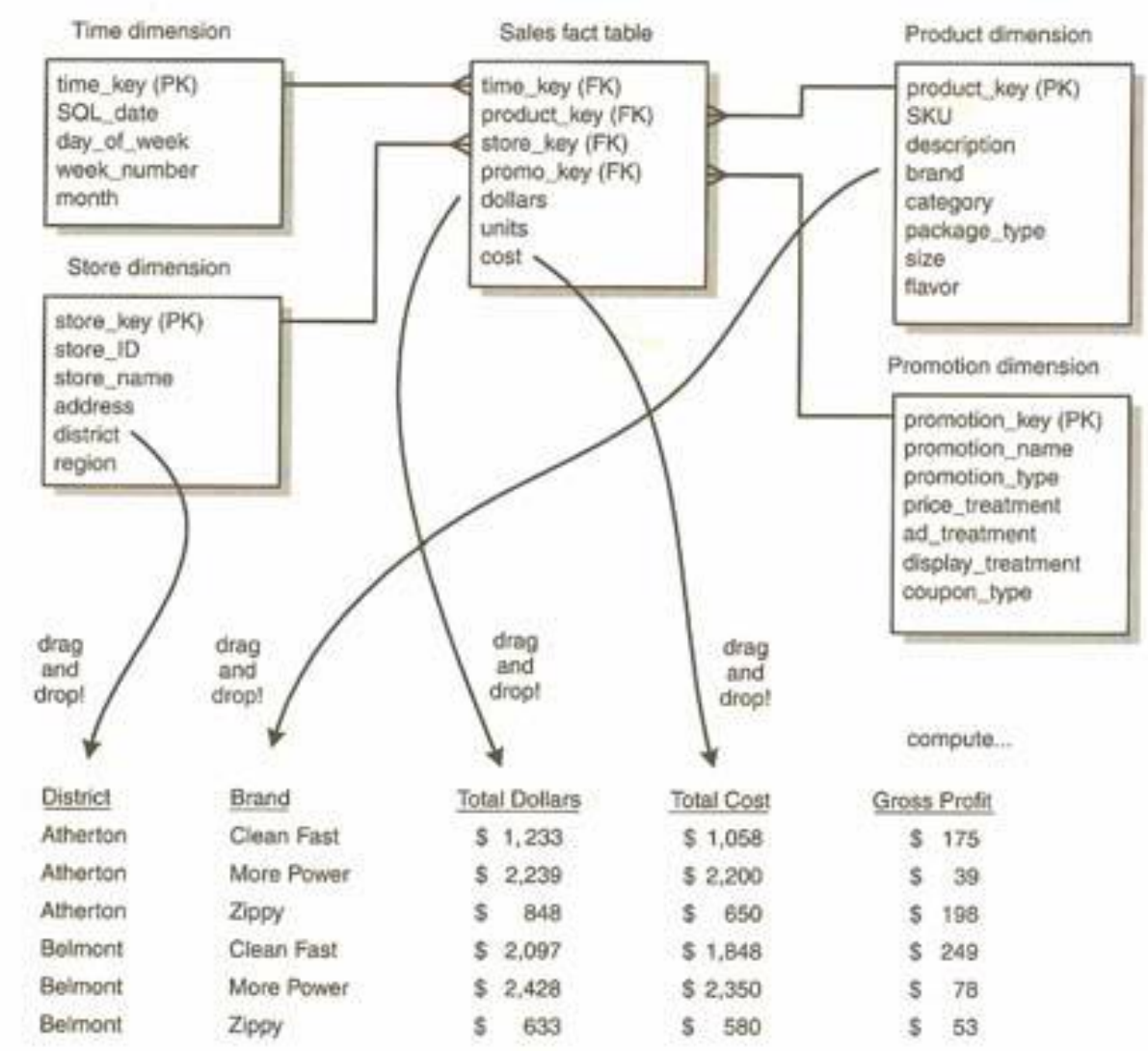
Typical Dimensional Answer Set

Brand	Pesos Sales	Unit Sales
Axon	780	263
Framis	1044	509
Widget	213	444
Zapper	95	39

Dimension Attribute

Fact Table Metrics

Creating a Report by Drag and Drop



Relating a Star Schema to a Report

- Drilling down = “give me more detail” by adding a row header (to an existing SQL request)
- Real drill down can mix hierarchical and non-hierarchical attributes from all available dimensions

Dimension Attributes Yield Interesting Results

- Dimension attributes are the source of most interesting constraints
- Examples
  - Slice sales by product category, by region, by barangay
  - Analyze sales effectiveness on radio promotions via the AdType attribute in Promotions dimension

Outline for This Session

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- **Why not Relational Modeling?**
- Examples of Dimensional Modeling
- Fact and Dimension Tables
- Designing the Dimension Model

Two Paradigms

- Relational Modelling
- Dimensional Modelling

Review: Relational Modeling

- Widely used method in most databases nowadays
- Data is divided into discrete entities
  - each of which becomes a relational database table called an entity
- Models are shown in two forms – logical and physical
- Logical models are designed to be independent of any particular RDBMS.
  - The “tables” in a logical model are called entities. The “columns” are called attributes.

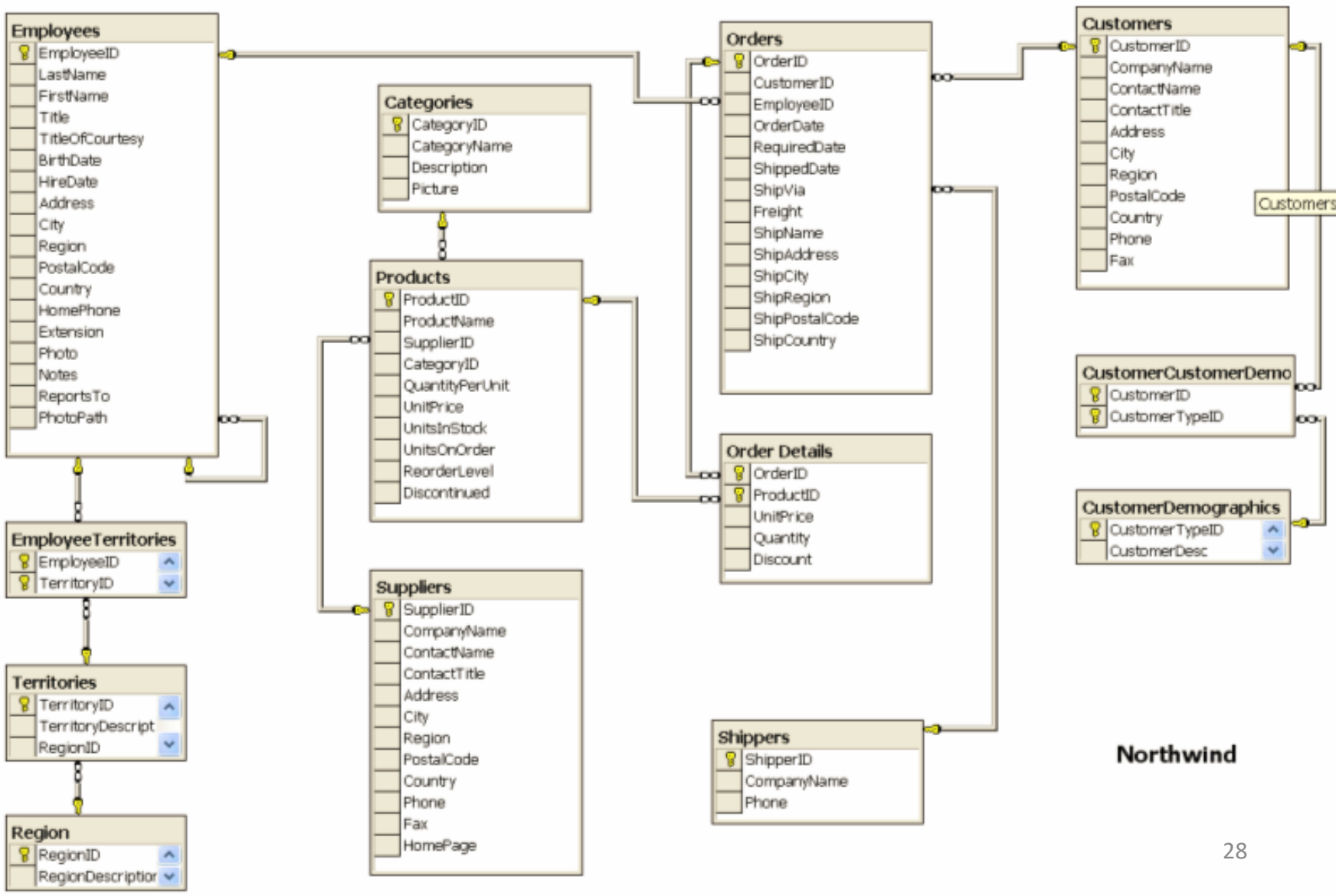
Review: Relational Modeling

- Physical models are derived from logical models but are specific to a given RDBMS.
- Each entity has a unique identifier known as its primary key.
- The primary key consists of one or more attributes/columns.

Normalized Models

- Designed to eliminate redundancies. Other than keys, each attribute may appear in only one table.
- Design objective: a Third Normal Form (3NF) model.
- Modeling business processes results in numerous data entities/tables and a spaghetti-like interweaving of relationships among them.
  - Some ERP systems have tens of thousands of tables.
  - Even a small model can be challenging.

Northwind Normalized Model



Normalized Models NOT Good for DW Systems

- Not usable by end-users – too complicated and confusing
- Not usable for DW queries – performance too slow (many joins)

Normalized Models Best for Operational Systems

- Normalized models essential to good operational systems
  - Excellent for capturing and understanding the business (rules)
    - One PO, multiple Line Items
  - Great for speed when processing individual transactions

Observations on Relational Models

- Normalized models look very different from dimensional models
  - Normalized models confuse business users
  - Business users see their business in dimensional models
- Dimensional models may contain more content than normalized models
  - History
  - Enhanced with content from external sources

Two Key Benefits of Dimensional Modeling à la Kimball

- Understandability
  - Model must be easily understood by business users
  - Yet represent complexities of the business
- Performance
  - Fast response to queries that summarize millions of rows is essential
  - Limiting models to single level joins rather than multi-level joins
  - Denormalization has a significant impact on performance

Benefits of Dimensional Models

- Predictable, Standard Framework
  - Users recognize that this is "their business"
  - Report writers, query tools, and user interfaces can be built into BI tools
  - Makes user interfaces more understandable
  - Makes processing more efficient

Benefits of Dimensional Models

- Gracefully Extensible to Accommodate Change
  - Existing tables can be changed by adding new data rows
    - Data should not have to be reloaded
  - No query tool or reporting tool has to be reprogrammed
  - Old BI applications continue to run without yielding different results

Benefits of Dimensional Models

- Star Join Schema is Symmetrical
  - Every dimension is equivalent
  - All dimensions symmetrically equal entry points to the fact table
    - No concern about order in selecting tables
  - Logical design can be done nearly independent of expected query patterns
    - Future queries not thought of can be accommodated easily
  - User interfaces, query strategies, and SQL generated are all symmetrical

Benefits of Dimensional Models

- Standard Approaches for Common Modeling Situations
  - Role-playing dimensions
    - Sales Date versus Received Date
  - Slowly changing dimensions
  - Heterogeneous products
    - Need to track lines of business together
    - But each LOB product set is highly idiosyncratic
  - And more…

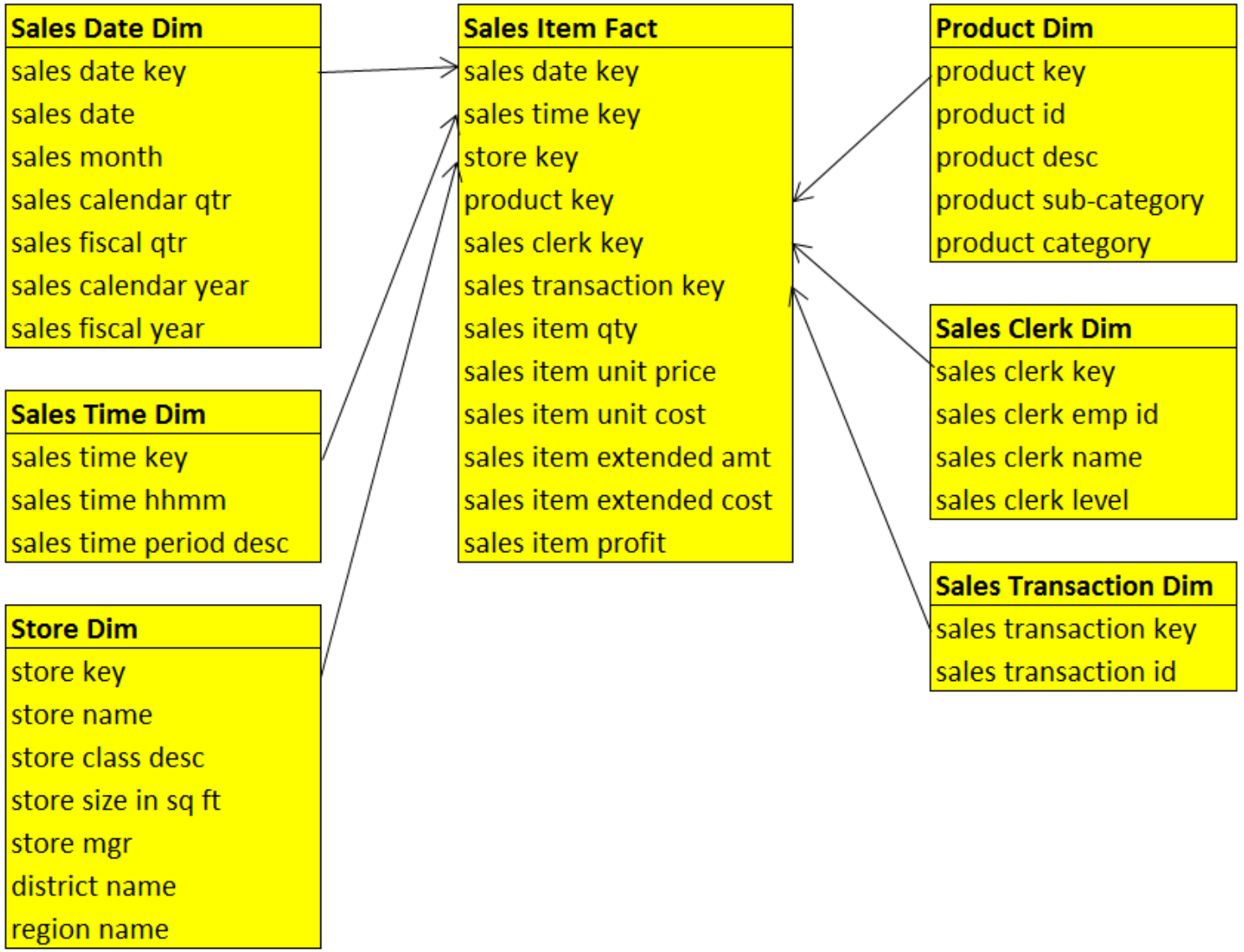
Benefits of Dimensional Models

- Aggregate Management
  - Aggregate tables are summary tables
    - Example: monthly sales fact table with month dimension
  - A sound aggregate strategy is essential to good performance and economic processing

Outline for This Session

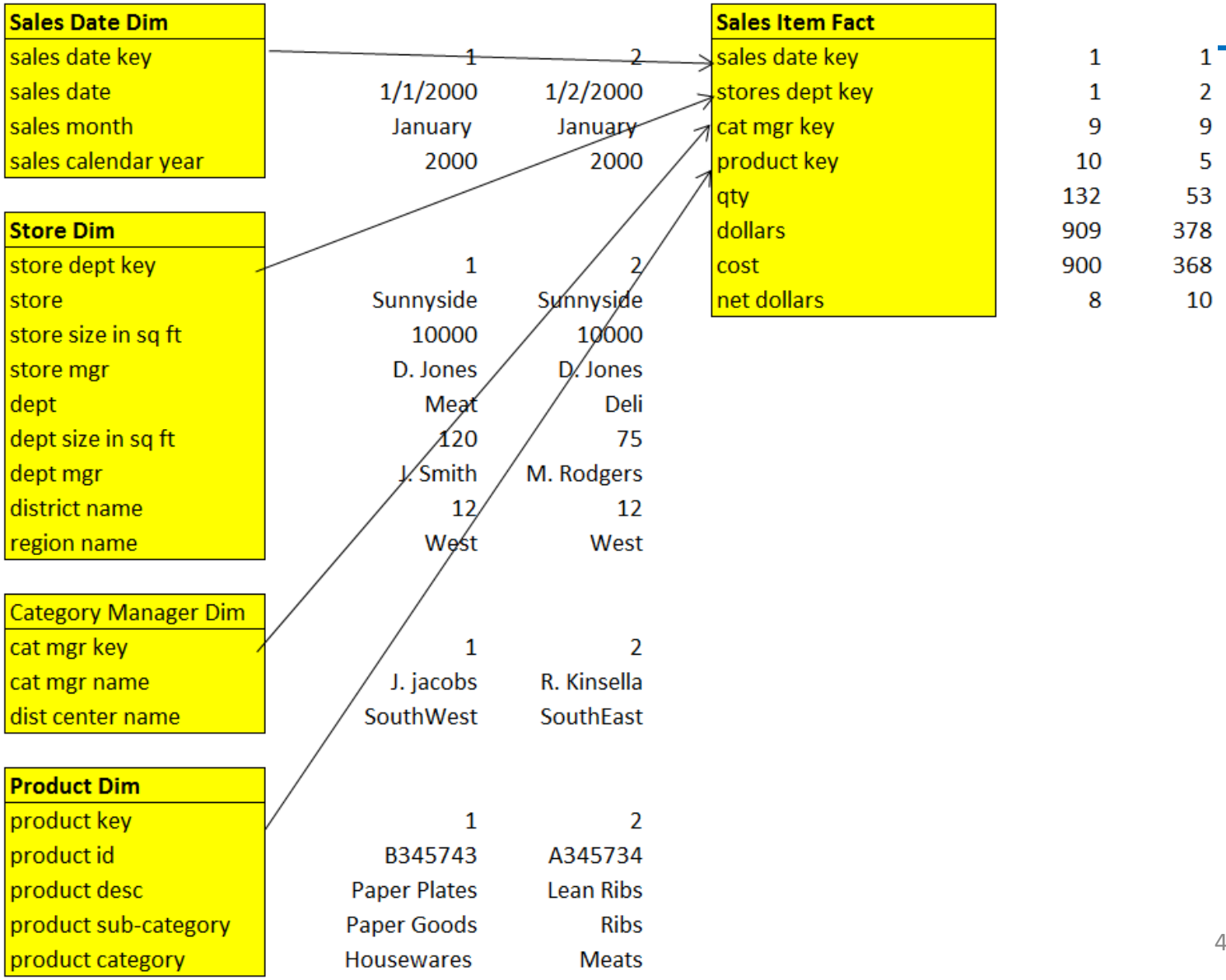
- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- **Examples of Dimensional Modeling**
- Fact and Dimension Tables
- Designing the Dimension Model

Star Schema Example



With Dimension Families

Sample Data



Sales Date Dim		
sales date key	1	2
sales date	1/1/2000	1/2/2000
sales month	January	January
sales calendar year	2000	2000

Store Dim		
store dept key	1	2
store	Sunnyside	Sunnyside
store size in sq ft	10000	10000
store mgr	D. Jones	D. Jones
dept	Meat	Deli
dept size in sq ft	120	75
dept mgr	J. Smith	M. Rodgers
district name	12	12
region name	West	West

Category Manager Dim		
cat mgr key	1	2
cat mgr name	J. jacobs	R. Kinsella
dist center name	SouthWest	SouthEast

Product Dim		
product key	1	2
product id	B345743	A345734
product desc	Paper Plates	Lean Ribs
product sub-category	Paper Goods	Ribs
product category	Housewares	Meats

Sales Item Fact		
sales date key	1	1
stores dept key	1	2
cat mgr key	9	9
product key	10	5
qty	132	53
dollars	909	378
cost	900	368
net dollars	8	10

Outline for This Session

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Examples of Dimensional Modeling
- **Fact and Dimension Tables**
- Designing the Dimension Model

Sample Fact Table Rows

Product Sales Fact							
Sales Date Key	Store Dept Key	Dist Ctr Key	Product Key	Qty	Sales Dollars	Cost	Net Dollars
343	34	1	23454	234	534	530	4
343	45	2	28343	745	756	435	23
…	…	…	…	…	…	…	…
344	34	1	23454	3	521	493	28
344	45	2	28343	4	689	542	53
…	…	…	…	…	…	…	…

Sample Dimension Table

Product Dim				
Product Key	Product Id	Product Desc	Product Sub Category	Product Category
0	Invalid Product	Invalid Product	Invalid Product	Invalid Product
-1	Unknown Product	Unknown Product	Unknown Product	Unknown Product
-2	Product is Not applicable	Product is Not applicable	Product is Not applicable	Product is Not applicable
1	A45723EF	Jonathon Apples	Apples	Fruit
2	C48723EF	Bartlett Pears	Pears	Fruit
3	F45751EF	12 oz Oatmeal	Oatmeal	Cereal
4	G12723EF	24 oz Oatmeal	Oatmeal	Cereal
5	A12433EF	6 pk Individual Oatmeal	Oatmeal	Cereal
6	C45723EF	Pound Sirloin Steak	Steak	Meat

Sample Dimension Table

Sales Date Dim			
Sales Date Key	Sales Date	Sales Date Month	Sales Date Year
0	'Invalid Sales Date'	'Invalid Sales Date'	'Invalid Sales Date'
-1	'Unknown Sales Date'	'Unknown Sales Date'	'Unknown Sales Date'
-2	'Sales Date Not Applicable'	'Sales Date Not Applicable'	'Sales Date Not Applicable'
-3	'Sales Date To Be Determined'	'Sales Date To Be Determined'	'Sales Date To Be Determined'
1	'01/01/2000'	January	2000
2	'01/02/2000'	January	2000
…	'…'	'…'	2000
34	'02/01/2000'	February	2000
…	….	…	2000
369	'01/01/2001'	January	2001
370	'01/02/2001'	January	2001
…	'…'	'…'	2001

Sample Queries

- What was the best selling product category last week?

```
SELECT product_category, sum(sales _dollars)
FROM sales_fact sf, sales_date sd, product p
WHERE last_week_ind = 'Y' and <JOIN
   Statements>
GROUP by product_category having
   rank(sum(sales_dollars)) <2
```

Sample Queries

- Which stores sold the most of product category 'ABC' last week?

```
SELECT store, sum(sales_dollars)
FROM sales_fact sf, sales_date sd, product p
    where last_week_ind = 'Y' AND
    product_category = 'ABC' and <JOIN
    Statements>
GROUP BY store having rank(sum(sales_ dollars))
    <6
```

Sample Report

- Business Analysis
  - How did profit last month equate to store size?
- Report

store size	store name	size rank	profit rank
50,000 sq ft	Northern	1	4
30,000 sq ft	Southern	2	3
25,000 sq ft	Central	3	1
20,000 sq ft	Westside	4	2

Outline for This Session

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Examples of Dimensional Modeling
- Fact and Dimension Tables
- **Designing the Dimension Model**

Designing the Dimensional Model Steps

- Establishing Naming Conventions
- Do the Four-Step Dimensional Modeling Process
- Document the High Level Data Model Diagram
- Define the Data Sources
- Document the Detailed Table Designs
- Develop Detailed Bus Matrix
- Identify, Track, and Resolve Issues

Establishing Naming Conventions

- Use descriptive and consistent data names. Reasons:
  - Names become column headers in reports. Column names must be non-redundant. Example: not just City, but Customer City or Supplier City
- Use standard naming convention
  - PrimeWord_ZeroOrMoreQualifiers_ClassWord
    - Dimension names – product_key, product_category_code, product_category_name
    - Fact names – item_amount, order_amount
- Know the naming rules of your RDBMS
  - ProductKey, ProductCategoryCode, …

Four Step Table Design Process

1. Choose the Business Process
2. Declare the Grain
3. Identify the Dimensions
4. Identify the Facts

Document the High Level Data Model Diagram

- High Level Data Model Diagram
  - Used to communicate and validate with business users and senior management
  - Always follow the same convention in arranging dimensions around the fact table, e.g., start with the date at the top
  - Use the same arrangement with aggregates or omit or gray out unused dimensions and substitute the names of shrunken dimensions for others
  - See exhibit 5

Define the Data Sources

- This is sometimes known as the Application Architecture
- Often much more extensive descriptions are very helpful if you have many sources
- See **exhibit 6**

Document the Detailed Table Designs

- Document the detailed dimension worksheet
  - Known as a Source-to-Target Map
  - See Exhibit 7
- Note that spreadsheets are used extensively in metadata documentation

Develop Detailed Bus Matrix

- Bus matrix makes several things articulate and obvious
  - Business processes have several fact tables
  - Explicit granularity for fact tables
  - Named facts for fact tables
  - Reusable conformed dimensions
- See exhibit 8

Identify, Track, and Resolve Issues

- Issues continually arise as the team works among its members and with business participants
- Important to identify, track, and resolve these issues
  - See issues log
- Assign someone to capture and track issues that arise at meetings or in discussions

Outline for This Session

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Examples of Dimensional Modeling
- Fact and Dimension Tables
- Designing the Dimension Model

References

- Kimball, Ralph, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker, *The Data Warehouse Life Cycle Toolkit, Second Edition*, Wiley, 2008, ISBN 978-0-470-14977-5
- Schmitz, Michael D. UCI Irvine Data Warehousing Notes (2014), High Performance Data Warehousing
- Simon, Alan. CIS 391 PPT Slides
- Jeltema ,Bernie, UCI Irvine Data Warehousing Notes (2014), Strategic Frameworks, Inc.