

Dimensional Modeling

Prepared by Dr. Amjad Qtaish

DS601
Data Mining

Lecture 10



Lecture: #10

Dimensional Modeling



Dr. Amjad Qtaish

Faculty of Informatics
Midocean University



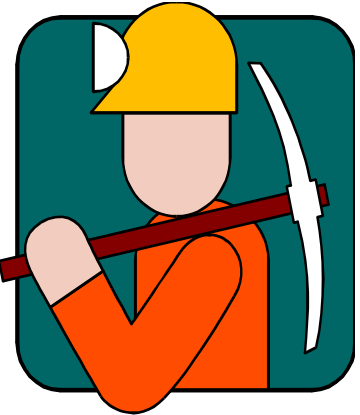
Data Mining (DS601)



Dimensional Modeling

- **Inmon versus Kimball Paradigm** 
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Relational Model vs Dimensional Model
- Fact and Dimension Tables

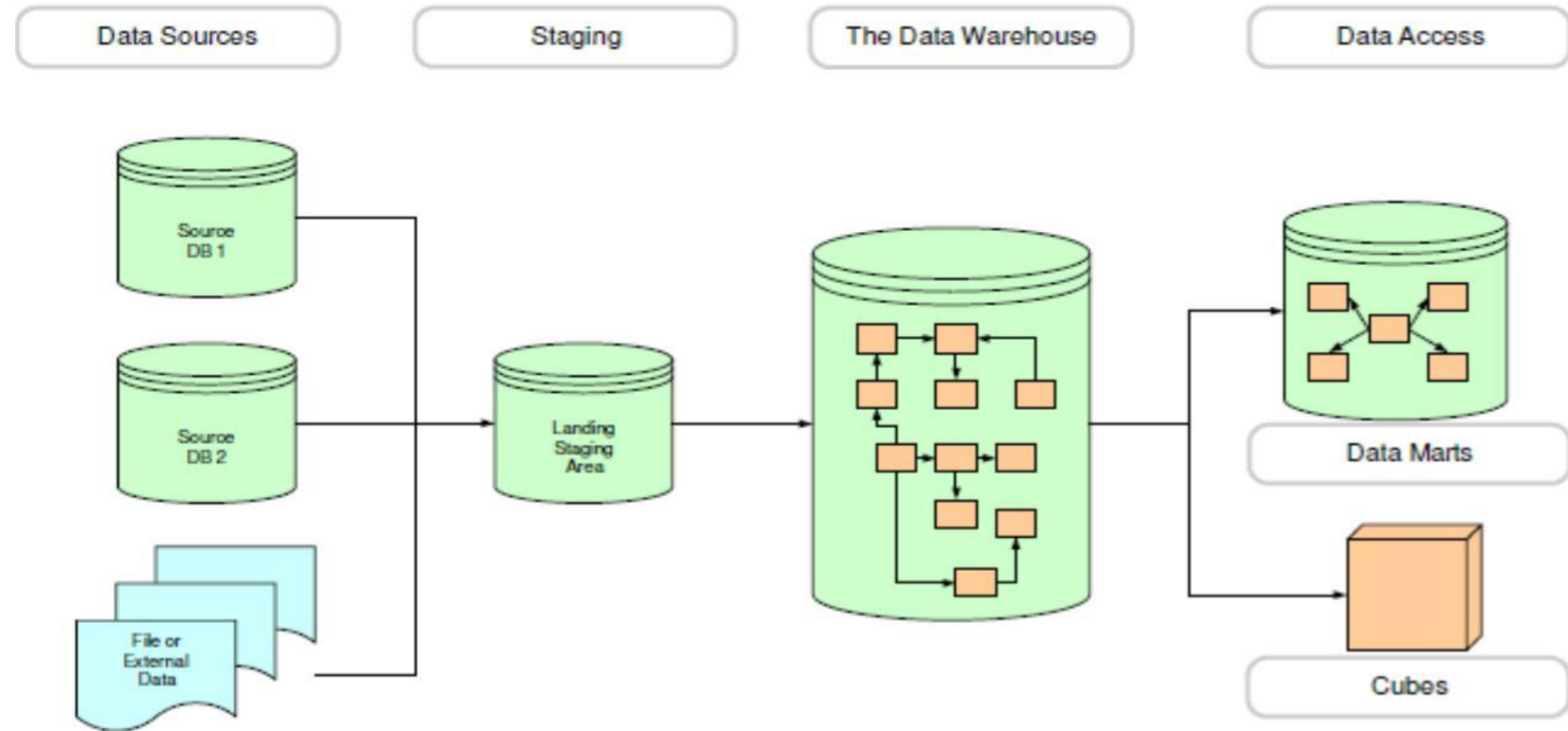
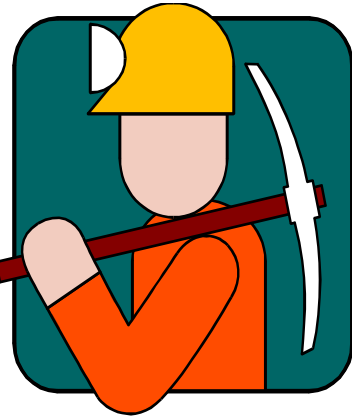




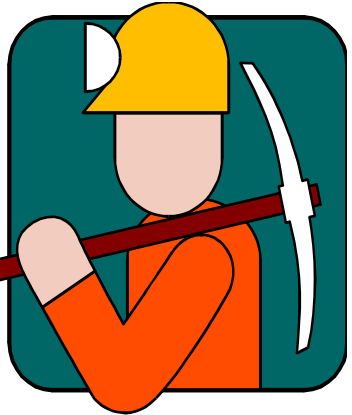
- Two Models for Data Warehouses
 - Inmon Model
 - Kimball Model
- **Inmon Model**
 - Consists of **all databases and information systems** in an organization
 - Also called the CIF (Corporate Information Factory)
 - Defines overall database environment as:
 - Operational
 - Atomic data warehouse
 -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



Inmon versus Kimball Paradigm



Inmon Model

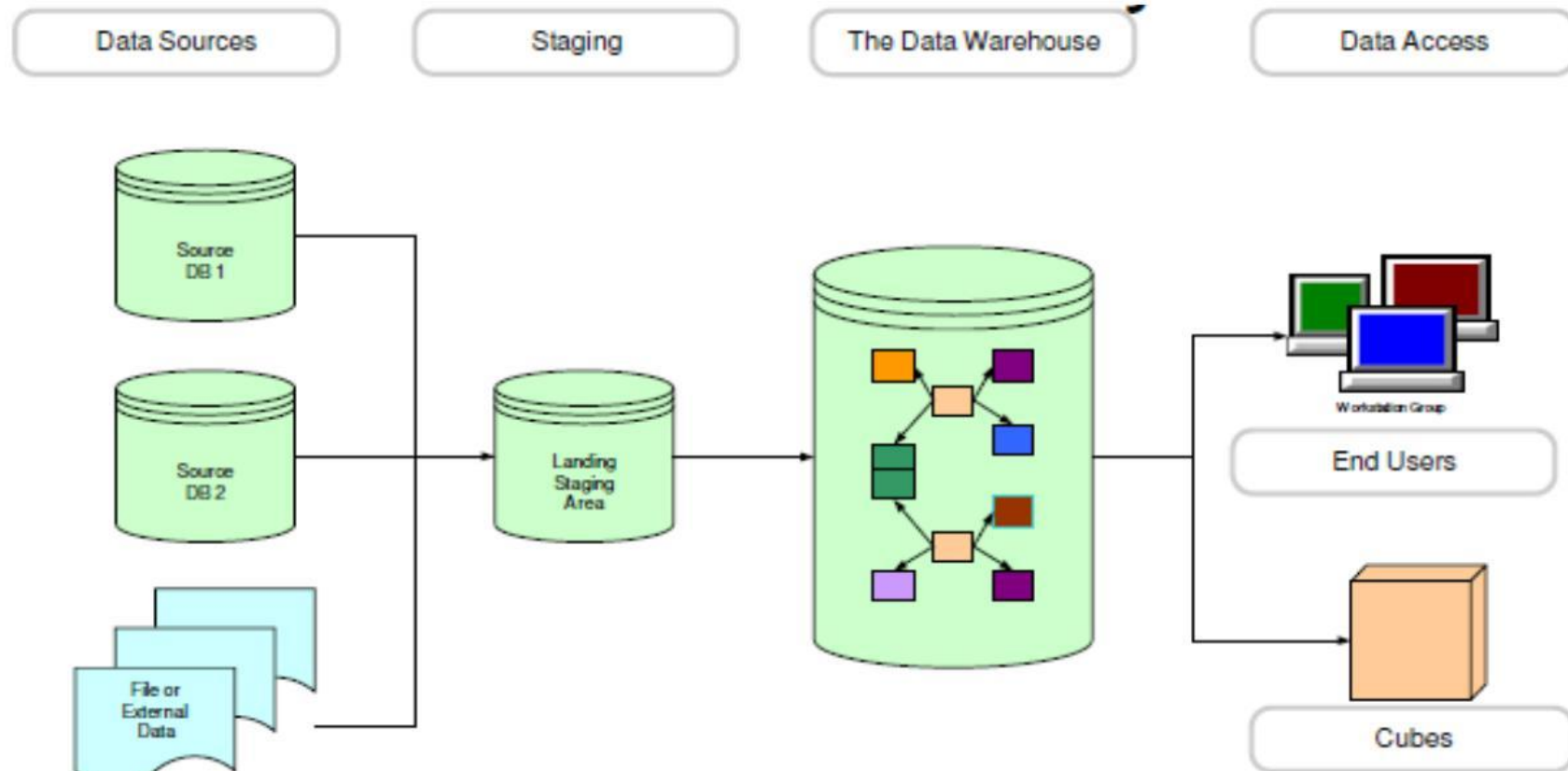


- Kimball Model
 - The Dimensional Data Model
 - Does not adhere to normalization theory
 - Starts with **tables**
 - Numeric Tables
 - Context Tables
 - **User accessible**

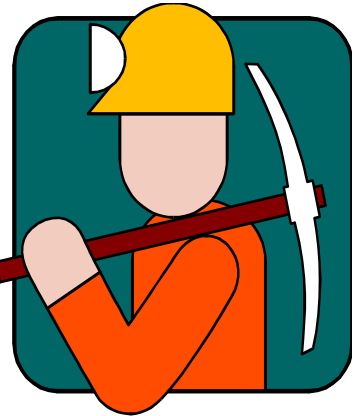




Inmon versus Kimball Paradigm

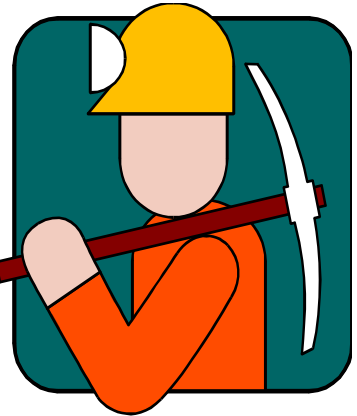


Kimball Model



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

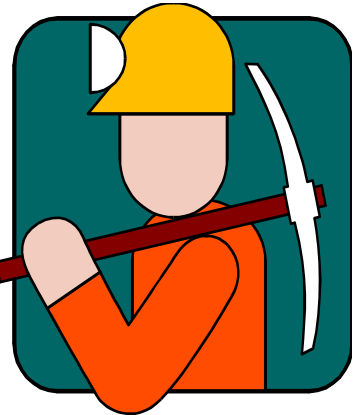


Philosophy Comparison of the Inmon and Kimball Model

	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



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



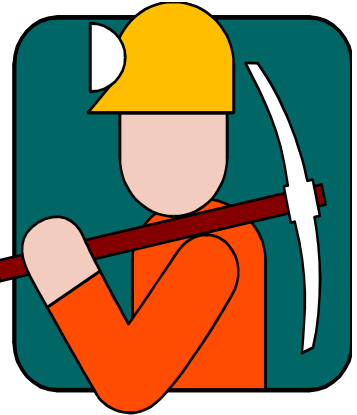
Dimensional Modeling

- Inmon versus Kimball Paradigm
- **What is Dimensional Modeling?**
- Why not Relational Modeling?
- Relational Model vs Dimensional Model
- Fact and Dimension Tables





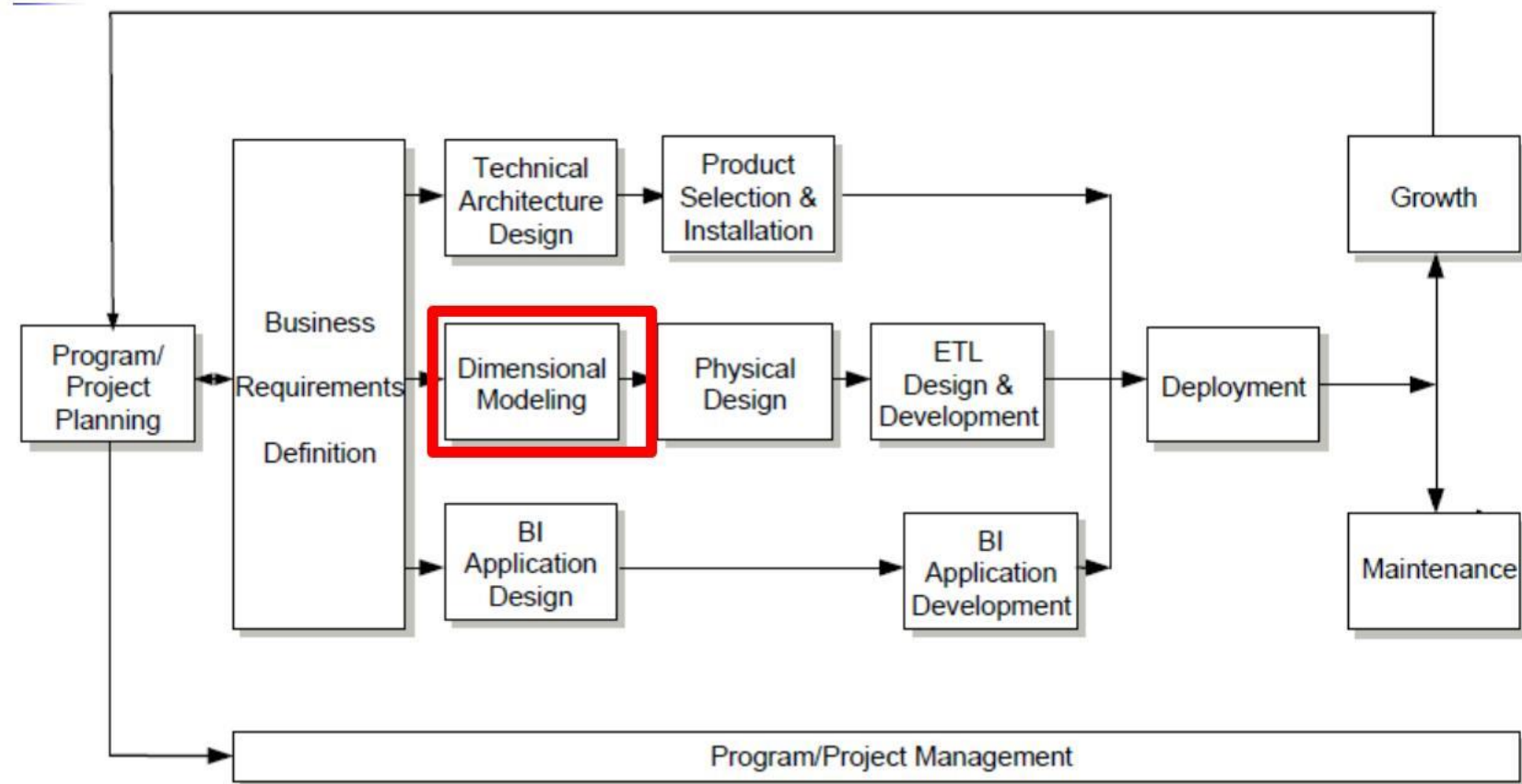
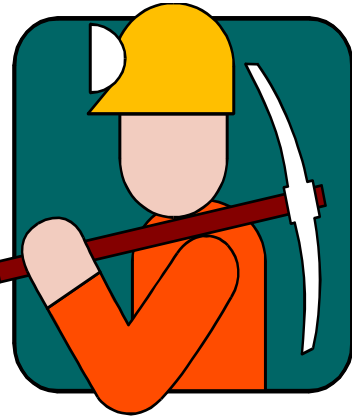
What is Dimensional Modeling?



- 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.



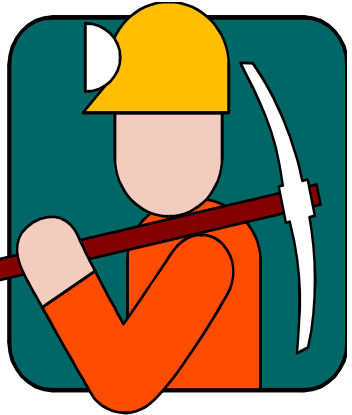
What is Dimensional Modeling?



The Kimball Lifecycle



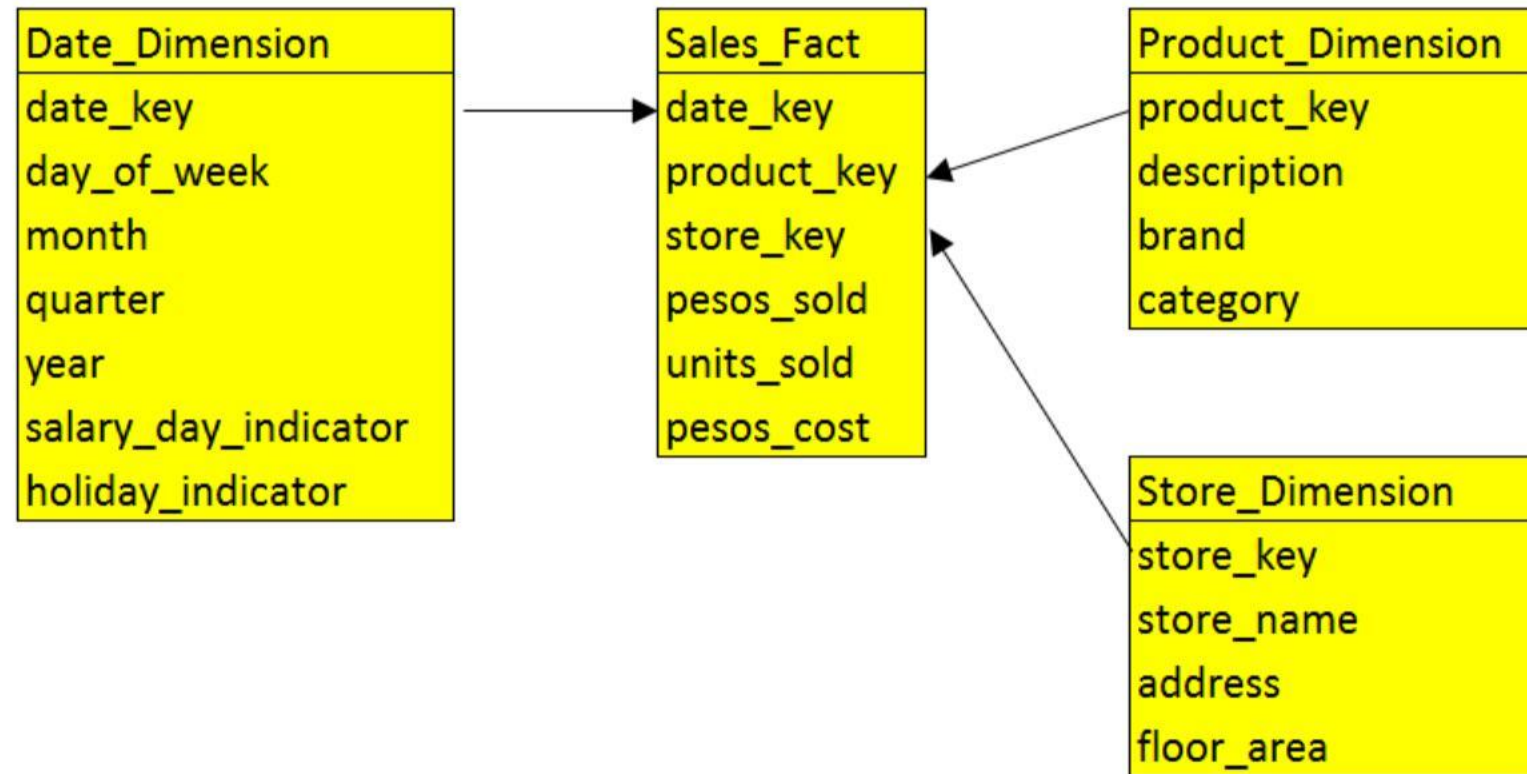
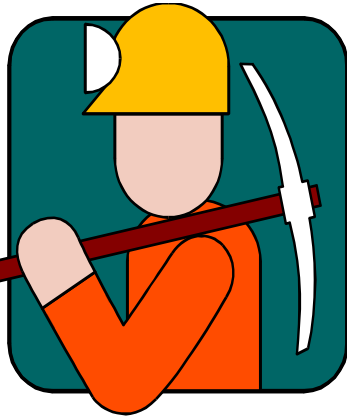
What is Dimensional Modeling?



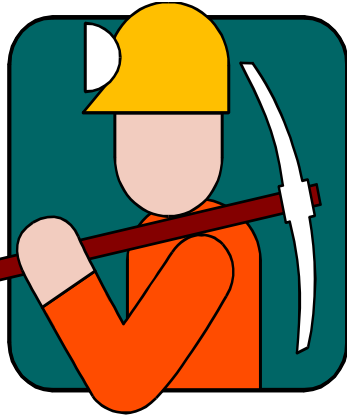
- Divides world into **measurements** and **context**.
- Measurements are numeric values called **facts**.
- Context intuitively divided into clumps called **dimensions**.
- Dimensions describe the “who, what, where, when, why, and how” of the facts.
- A **dimensional model** consists of a fact table 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).



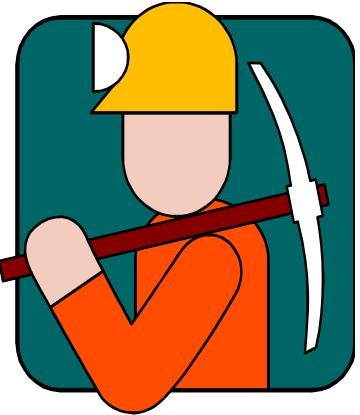
What is Dimensional Modeling?



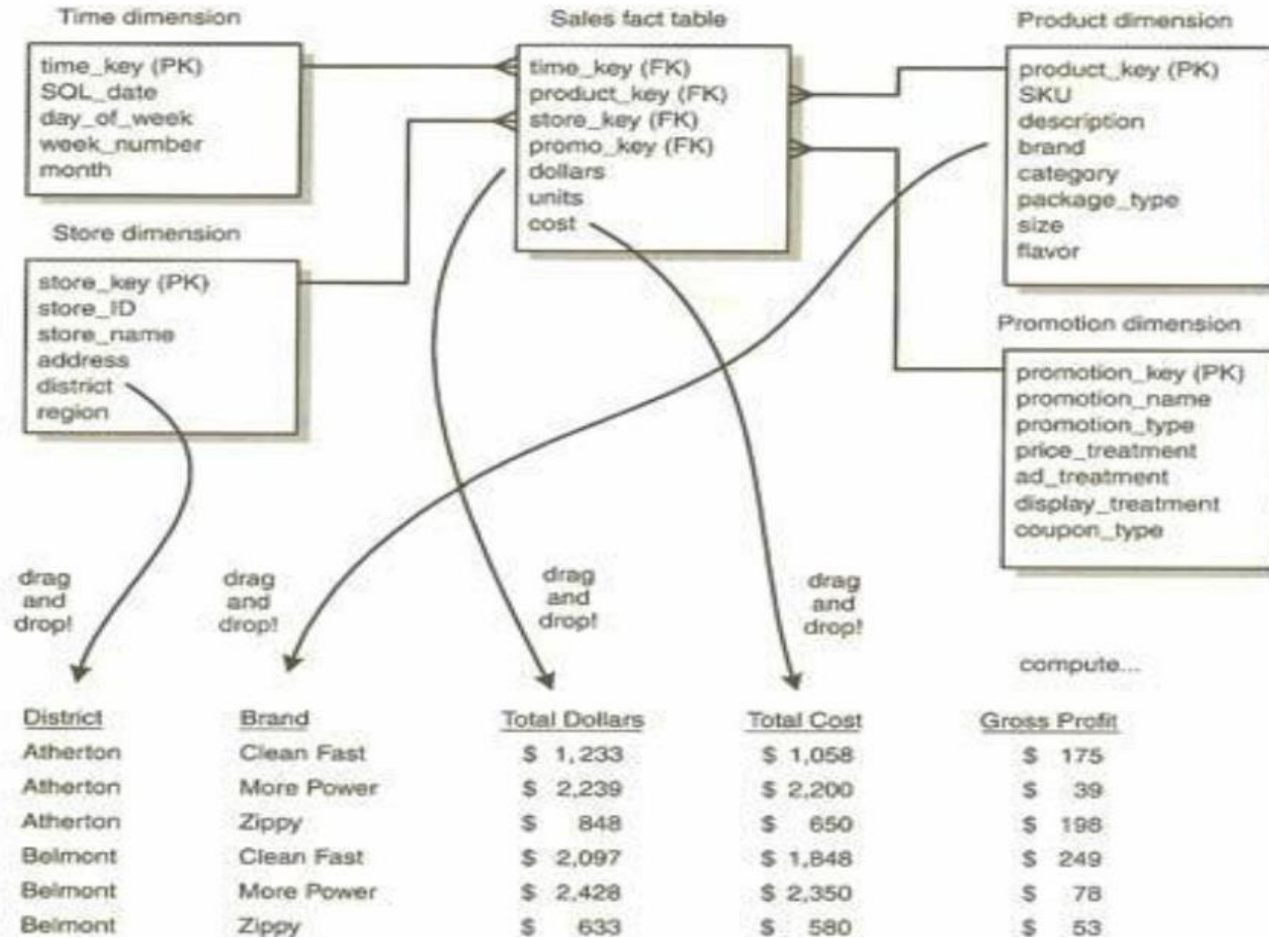
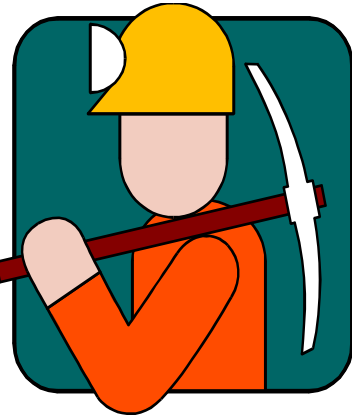
Typical Dimensional Model

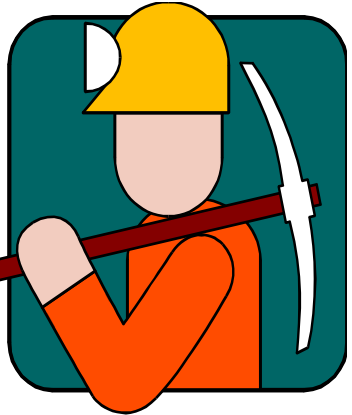


```
SELECT p.brand, sum(f.pesos_sold),  
       sum(f.units_sold)  
FROM sales_fact f, product_dim p, date_dim d  
WHERE f.productkey = p.productkey  
      and f.datekey = d.datekey  
      and d.quarter = '1 Q 2015'  
GROUP BY p.brand  
ORDER BY p.brand
```



Axon	780	263
Framis	1044	509
Widget	213	444
Zapper	95	39
Dimension Attribute	Fact Table Metrics	

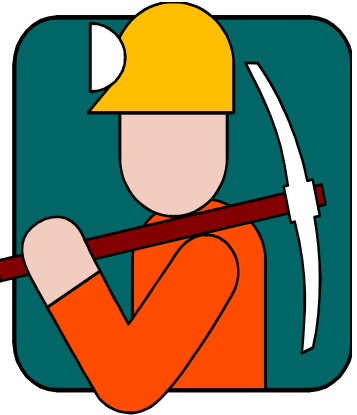




- **Drilling down** = “give me more detail” by adding a **row header** (to an existing SQL request)
- Real drill down can mix hierarchical and non-hierarchical attributes from **all available dimensions**

Dimension Attributes Yield Interesting Results

Chapter 10



- 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

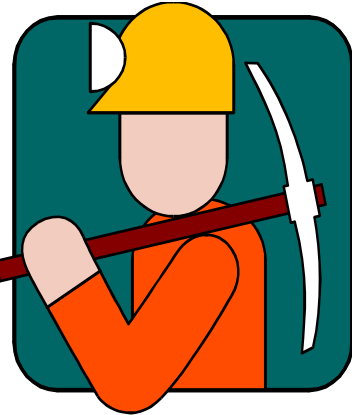




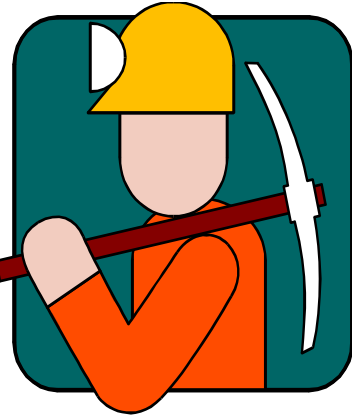
Dimensional Modeling

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- **Why not Relational Modeling?** 
- Relational Model vs Dimensional Model
- Fact and Dimension Tables

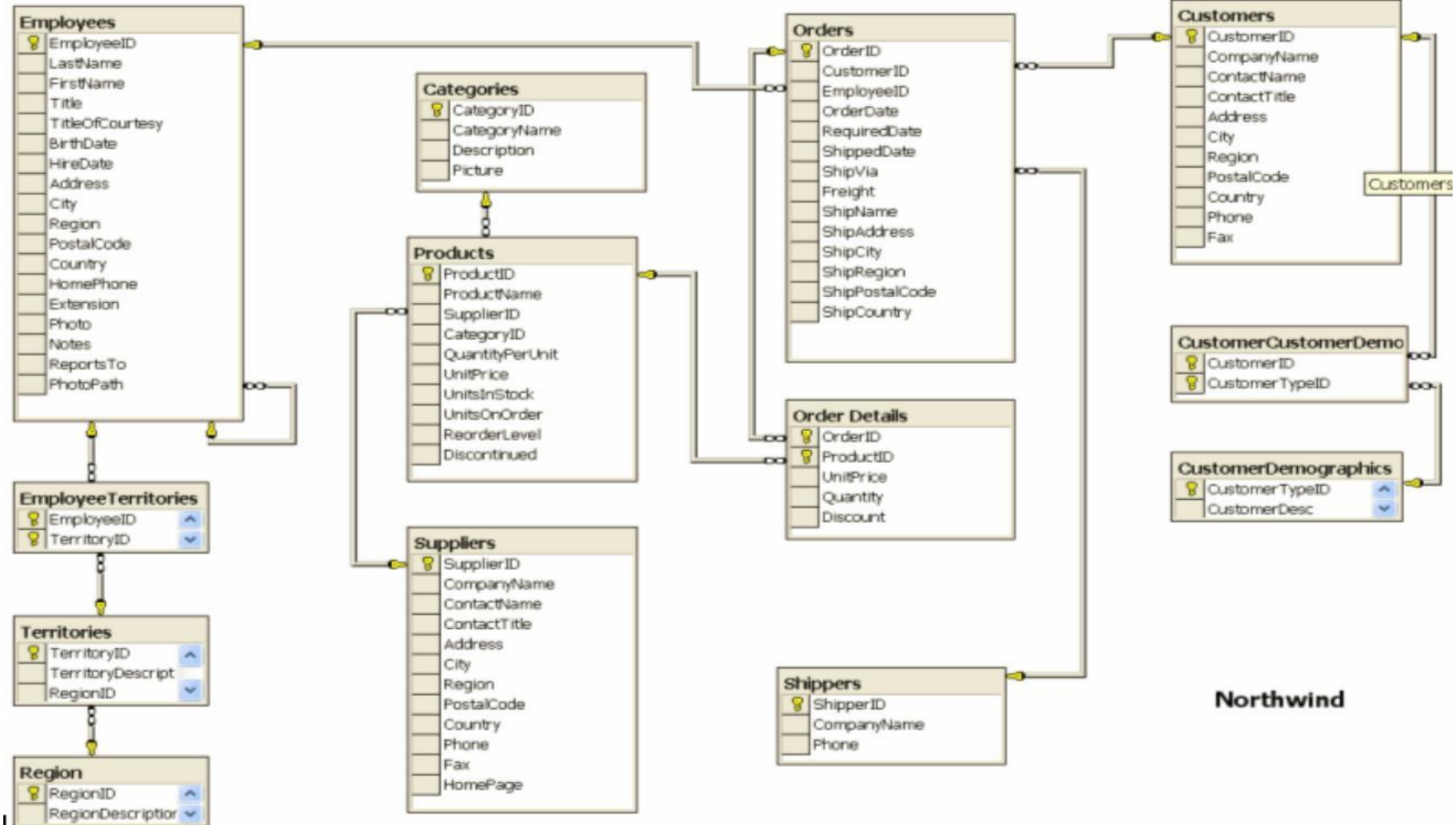
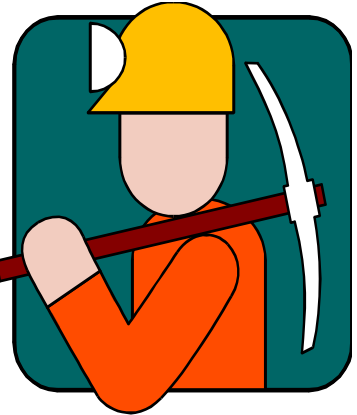




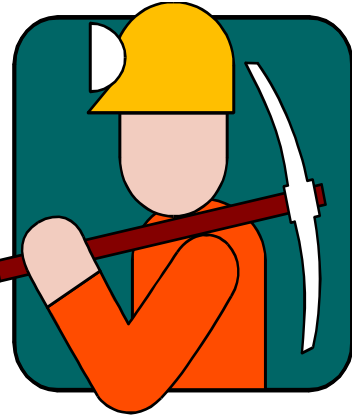
- Widely used method in most databases nowadays
- Data is divided into discrete entities
 - each of which becomes a relational database table called an **entity**
- Models are shown in two forms – **logical and physical**
- Logical models are designed to be **independent** of any particular RDBMS.
 -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



- 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.

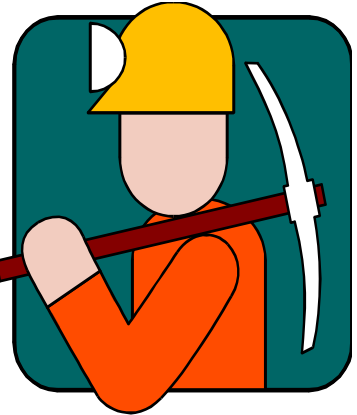


Northwind



- Normalized Models NO Good for DW Systems
- Not usable by end-users – too complicated and confusing
- Not usable for DW queries – performance too slow (many
- 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

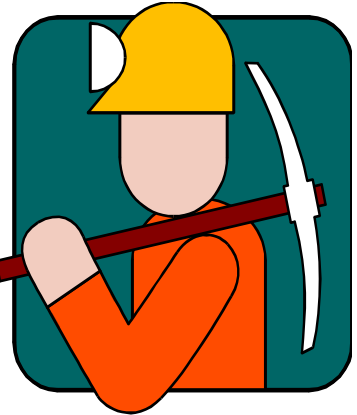




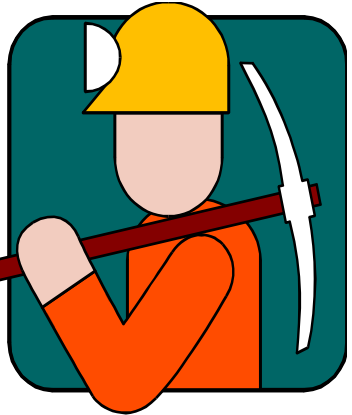
- Normalized models look very different from dimensional models
 - Normalized models confuse business users
 - Business users see their business in dimensional models
- Dimensional models may contain more content than normalized models
 - History
 - Enhanced with content from external sources



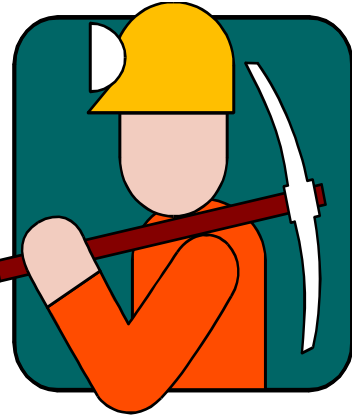
Two Key Benefits of Dimensional Modeling à la Kimball



- 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**



- 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
- 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

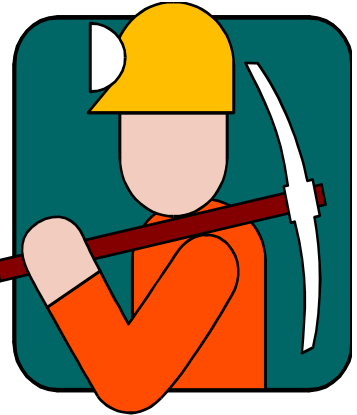


- Star Join Schema is **Symmetrical**
 - Every dimension is equivalent
 - All dimensions symmetrically equal entry points to the fact table
 - No concern about order in selecting tables
 - Logical design can be done nearly independent of expected query patterns
 - Future queries not thought of can be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



- 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

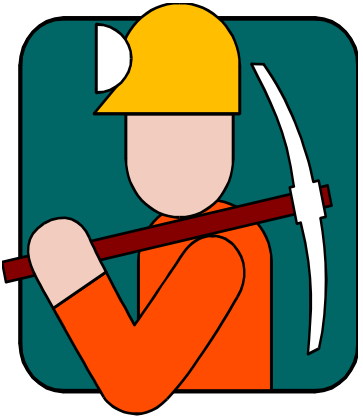




Dimensional Modeling

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- **Relational Model vs Dimensional Model**
- Fact and Dimension Tables





CUSTOMER
customer_ID (PK)
customer_name
purchase_profile
credit_profile
address

STORE
store_ID (PK)
store_name
address
district
floor_type

CLERK
clerk_id (PK)
clerk_name
clerk_grade

ERD

ORDER
order_num (PK)
customer_ID (FK)
store_ID (FK)
clerk_ID (FK)
date

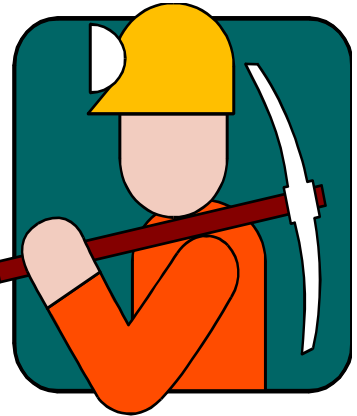
PRODUCT
SKU (PK)
description
brand
category

ORDER-LINE
order_num (PK) (FK)
SKU (PK) (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost

PROMOTION
promotion_NUM (PK)
promotion_name
price_type
ad_type



Dimensional Model



TIME

time_key (PK)
SQL_date
day_of_week
month

STORE

store_key (PK)
store_ID
store_name
address
district
floor_type

CLERK

clerk_key (PK)
clerk_id
clerk_name
clerk_grade

DIMENSIONAL MODEL

FACT

time_key (FK)
store_key (FK)
clerk_key (FK)
product_key (FK)
customer_key (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost

PRODUCT

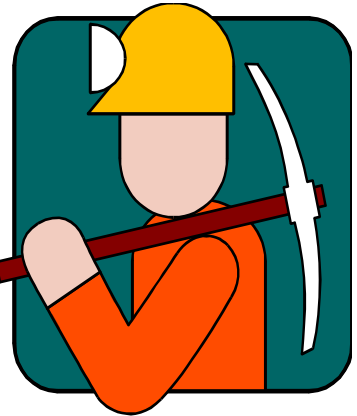
product_key (PK)
SKU
description
brand
category

CUSTOMER

customer_key (PK)
customer_name
purchase_profile
credit_profile
address

PROMOTION

promotion_key (PK)
promotion_name
price_type
ad_type



Reltional Model

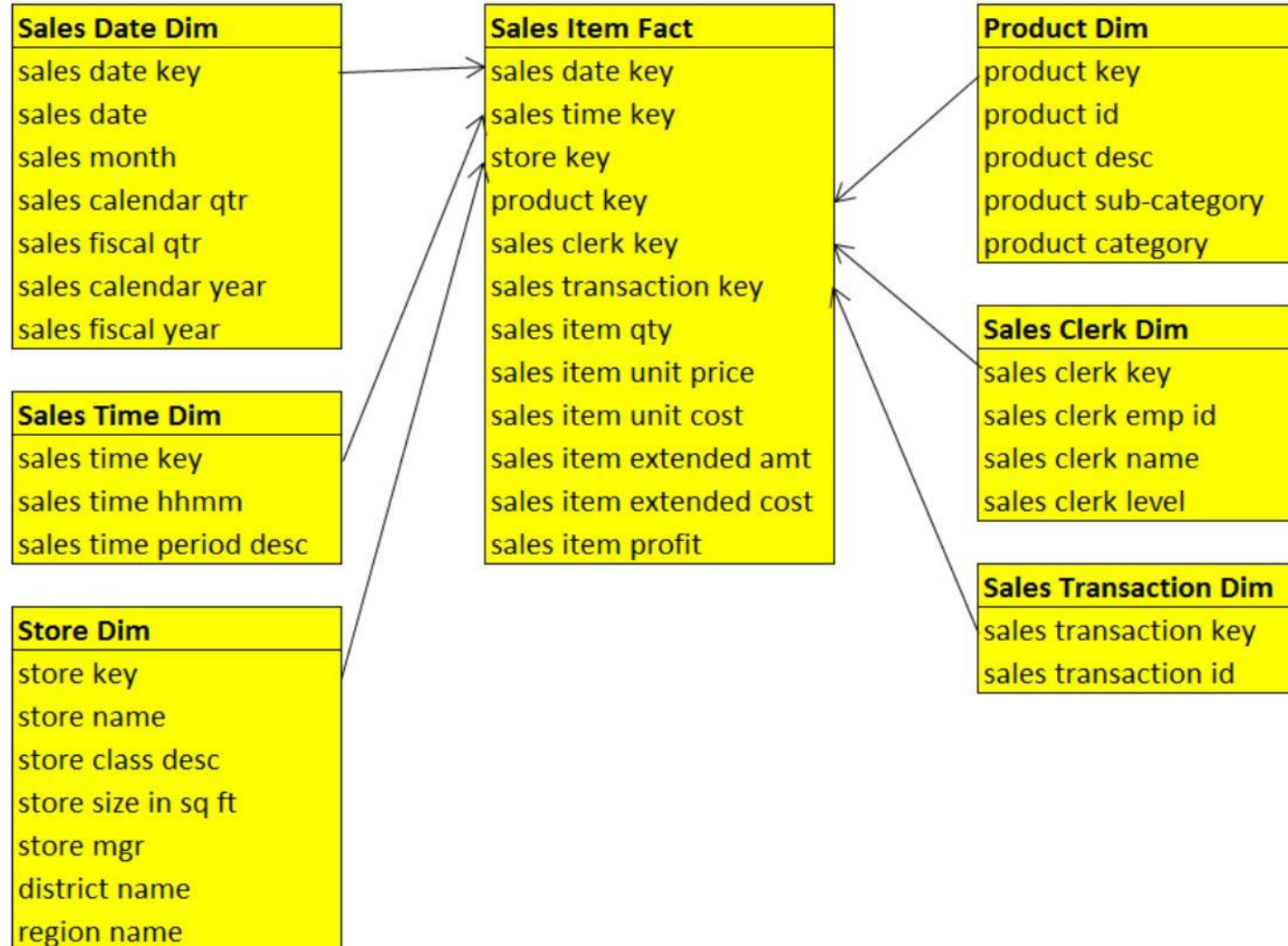
- One table per entity
- Minimize data redundancy
- Optimize update
- The Transaction Processing Model

Dimensional Model

- One fact table for data organization
- Maximize understandability
- Optimized for retrieval
- The data warehousing model



Star Schema Example

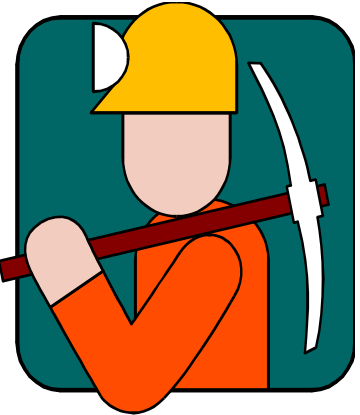




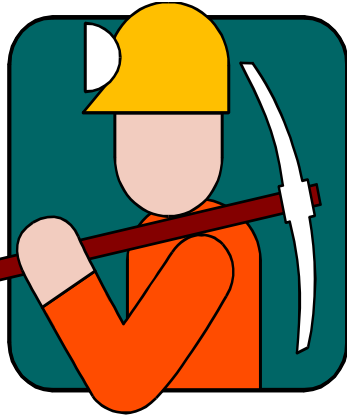
Dimensional Modeling

- Inmon versus Kimball Paradigm
- What is Dimensional Modeling?
- Why not Relational Modeling?
- Relational Model vs Dimensional Model
- **Fact and Dimension Tables** 

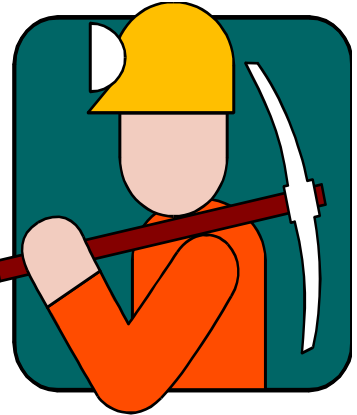




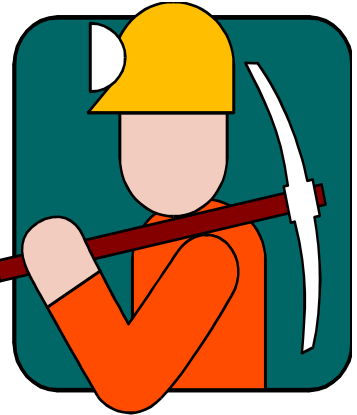
Product Sales Fact							
Sales Date Key	Store Dept Key	Dist Ctr Key	Product Key	Qty	Sales Dollars	Cost	Net Dollars
343	34	1	23454	234	534	530	4
343	45	2	28343	745	756	435	23
...	...	...	...	...	...	...	...
344	34	1	23454	3	521	493	28
344	45	2	28343	4	689	542	53
...	...	...	...	...	...	...	...



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



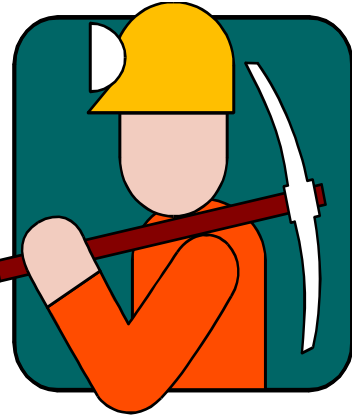
Sales Date Dim			
Sales Date Key	Sales Date	Sales Date Month	Sales Date Year
0	'Invalid Sales Date'	'Invalid Sales Date'	'Invalid Sales Date'
-1	'Unknown Sales Date'	'Unknown Sales Date'	'Unknown Sales Date'
-2	'Sales Date Not Applicable'	'Sales Date Not Applicable'	'Sales Date Not Applicable'
-3	'Sales 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



- 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
```



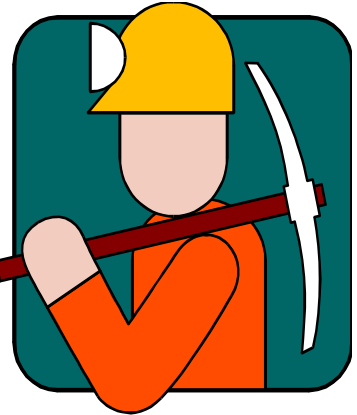


- Business Analysis
 - How did profit last month equate to store size?
- Report

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



A. Cloud Data Warehousing



- Platforms like **Snowflake**, **Google BigQuery**, **Amazon Redshift**, and **Databricks** now dominate.
- They support **massively parallel processing (MPP)** and **separation of compute and storage**, making star schemas scalable and cost-effective.

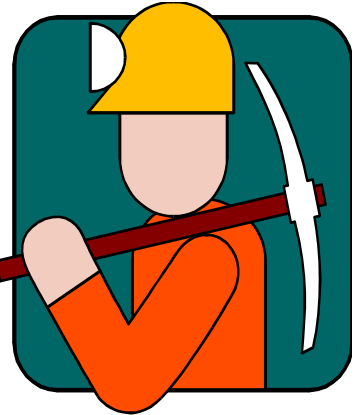
B. Modern Data Modeling Techniques



- **Data Vault 2.0**: Combines the scalability of Inmon with the agility of Kimball, using *hubs*, *links*, and *satellites* for historical tracking.
- **Lakehouse Architecture**: Integrates data lakes (unstructured) and warehouses (structured) using open formats (e.g., **Delta Lake**, **Apache Iceberg**, **Hudi**).
- **Semantic Models (dbt Semantic Layer, Cube.js)**: Provide abstraction layers for business metrics without duplicating data.



C. Automation and Governance



- **Data modeling automation tools:** dbt (Data Build Tool), ERwin, Starburst, and Apache Atlas for metadata management.
- **Data lineage and observability platforms** (Monte Carlo, Soda, Metaphor) integrate with dimensional models for traceability.

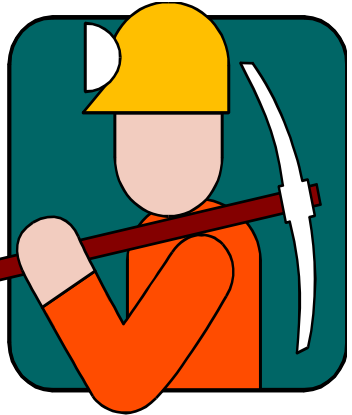
D. AI-Driven Analytics



- **AI-based query optimization** and **natural language querying** (e.g., Microsoft Fabric's Copilot, Google Gemini for BigQuery) let users interact with dimensional data using conversational interfaces.
- **Automated dimensional modeling** using AI helps infer facts and dimensions from source systems.



E. Real-Time and Streaming Analytics



- Integration of **streaming data** (Kafka, Flink, Materialize) with dimensional models for near-real-time OLAP (e.g., ClickHouse, Rockset).





Thank You