DWH Theory

CS 459 Business Intelligence

DWH = A very large database

- Data warehousing begins at around a Terabyte (TB) and extends up to several Petabytes (PB)
- **Distributed Architecture:** It Spans Over Several Servers.. Requiring Impressive Amount of Computing Power @ Enterprise Level
- Today → Creating a DWH on a single machine is also possible.

What is a Data Warehouse?

- More specifically, a **collective data repository**
- Containing ***snapshots*** of the operational data (history)
- Obtained through data cleansing (Extract-Transform-Load (ETL) process)
- Useful for **data driven decision-making and analytics**

Decision Making

- Different levels → varying decisions → diverse informational needs

Decision Making

Long-Term Decision Making in BI

- **Strategical**: Plan of Attack on a Large Time Period (3-5 years)
  - What do we do?
  - For whom do we do it?
  - How do we excel?

Short-Term Decision Making in BI

- **Tactical**: Plan of Attack on a Small Time Period (<1 year)
  - What various departments need to do for the organization to be successful in the future
- **Operational**: Link up Strategic Goals and Tactical Moves
  - Where are we now?
  - Where do we want to be?
  - How do we get there?
  - How do we measure our progress?

Plan Execution Path

BI Execution Path

Implementation in BI

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

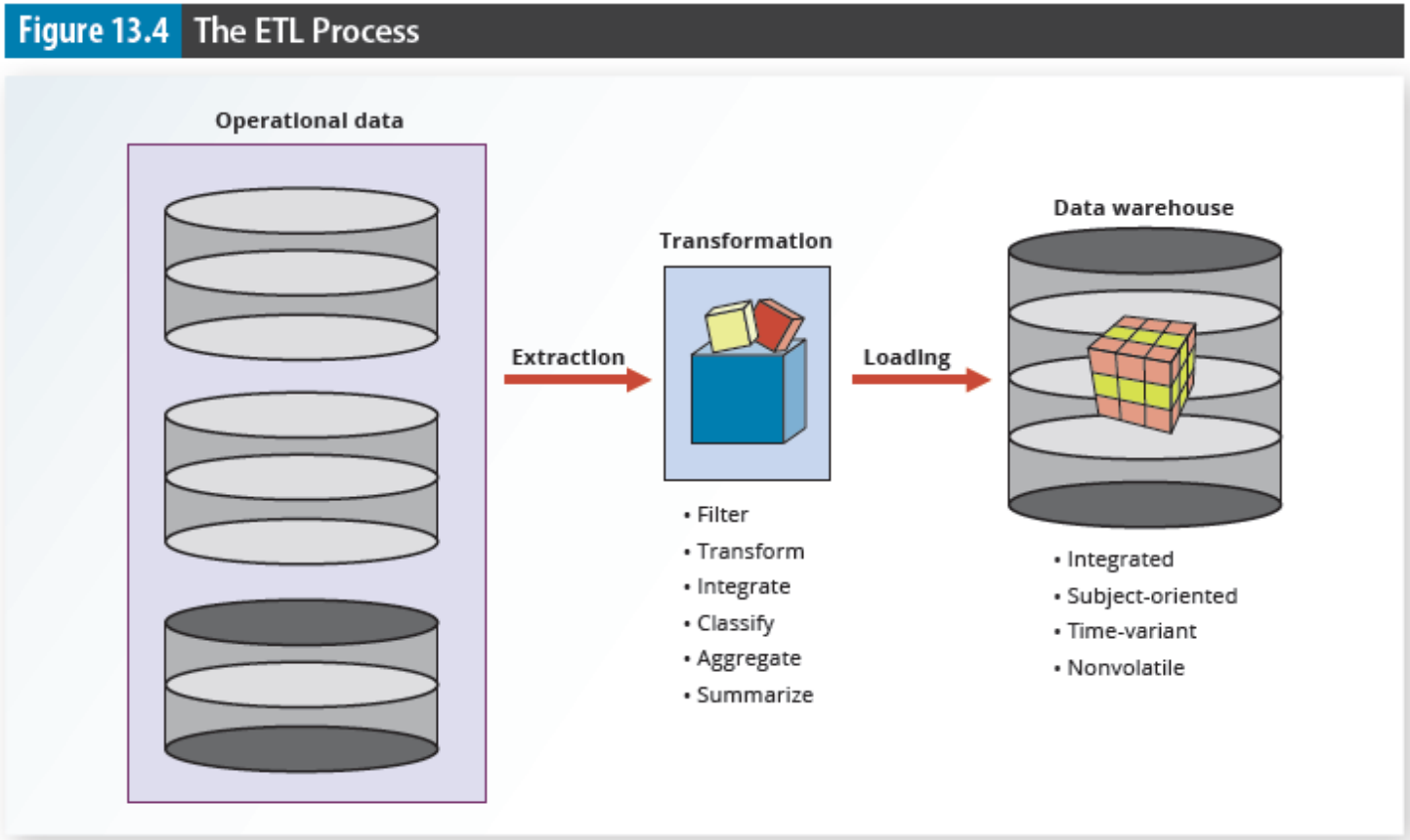
OLTP vs OLAP

	OLTP	OLAP
Objective	Day to day operation	Decision support
Database Design	Application oriented	Subject oriented
Data	Current, relational data	Historical, summarized multi-dimensional, consolidated
Usage	Repetitive	Ad-hoc
Transaction count	Large	Small
Transaction time	Less	More
Complexity	Short and simple transaction	Complex query
DB Size	Small (Gigabytes)	Very large (Terabytes)

Data Warehouse – Expert Definition

- **Ralph Kimball:** "a copy of transaction data specifically structured for query and analysis"
- **Bill Inmon:** "A data warehouse is a:
  - Subject oriented
  - Integrated
  - Non-volatile
  - Time variant

collection of data in support of management's decisions.

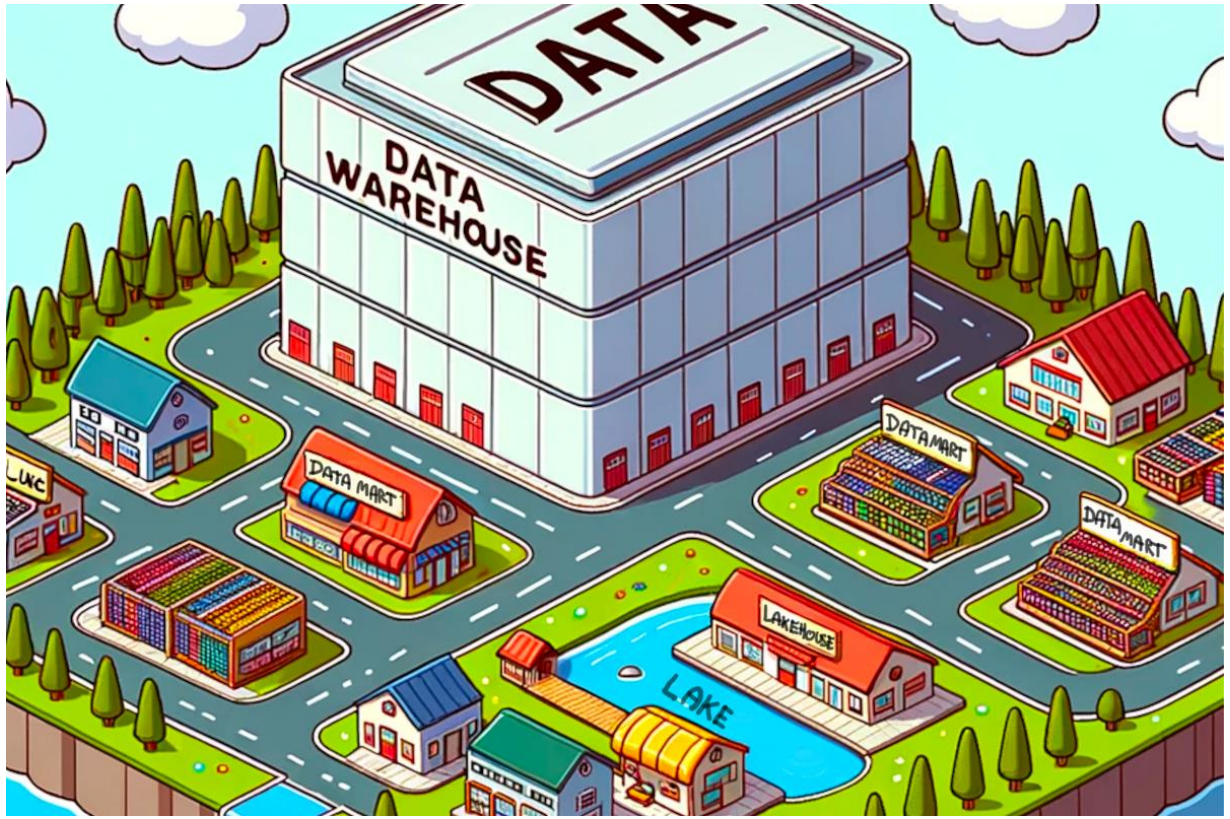


The Fathers of Data Warehousing

	W.H. Inmon	Ralph Kimball
The "Father" of…	Data Warehousing	Business Intelligence
Million Dollar Idea:	"Corporate Information Factory"	"Kimball Lifecycle"
"Data Warehouse" Definition	Strict. Subject-oriented summarized data.	Loose. Any query able data.
Approach: *How is the Data Warehouse built?*	As a whole, over time (Waterfall, **Top-down**)	In parts, by business process (Iterative, **Bottom-up**)

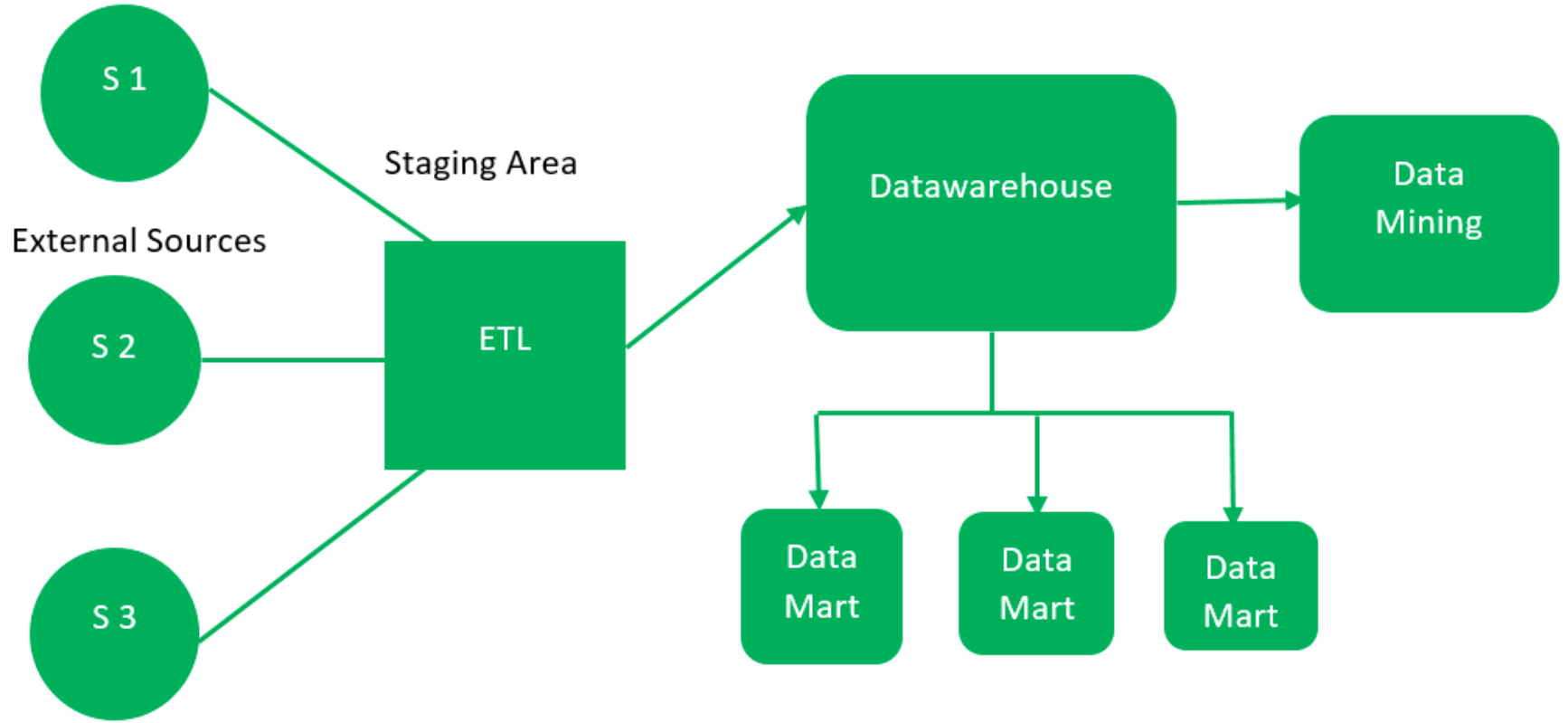
The Data Mart

- Single-subject subset of the data warehouse
- Provides Decision support to small group
- Address local or departmental needs



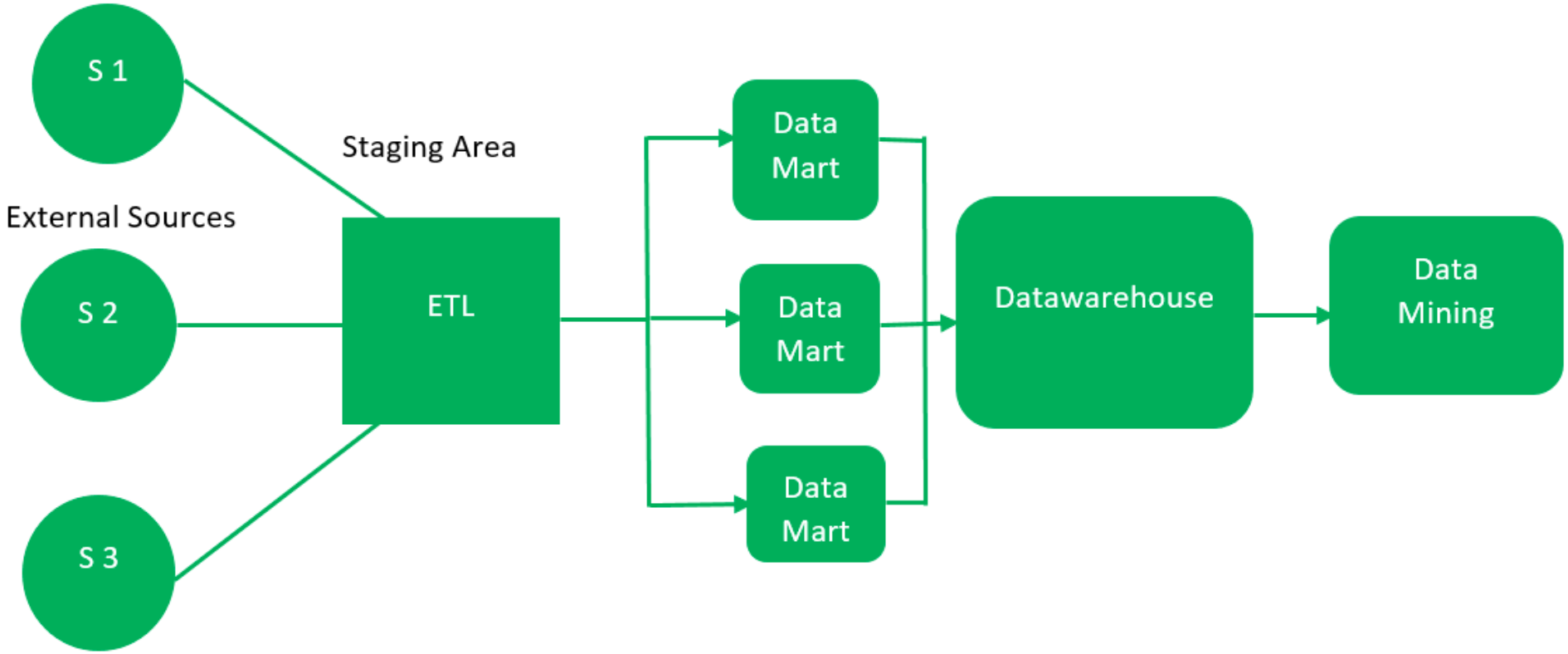
Top - Down Inmon Approach

- Extract – Transform and Load into the DW.
- Data marts constructed from the DW accessing a single business area



Bottom - Up Kimball Approach

- Extract – Transform and Load into Data Marts accessing a single business area.
- Data Marts are integrated into the DW.



Which approach to pick?

- Is there a need to see the *overall picture* or are *multiple functions* involved?
- **Inmon /Top-down**

- Is there a need to only few *domain-specific picture* or analysis is only needed at a *small scale*?
- **Kimball/ Bottom-up**

Inmon's DWH

Subject-Oriented

Integrated

A Data Warehouse is

Non-Volatile

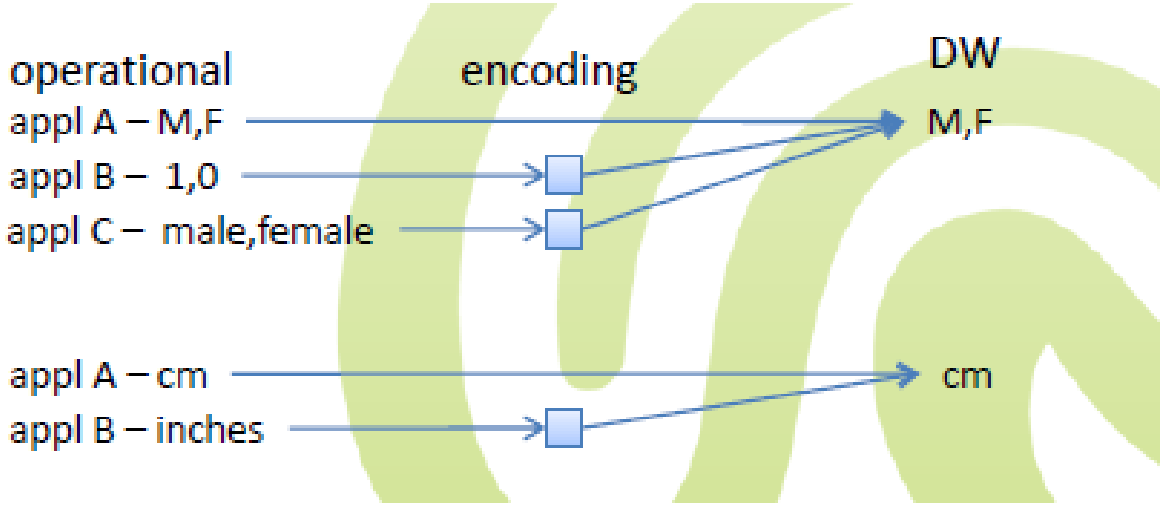
Time Variant

Defining Data Warehouse

Subject-Oriented

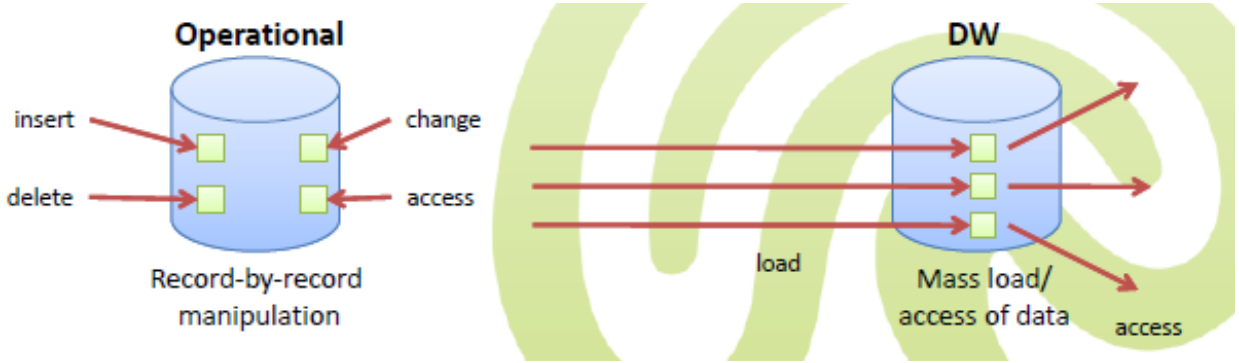
- The data in the DW is organized in such a way that all the data elements relating to the same real-world event or object are **linked together**
- Typical subject areas in DWs are Customer, Product, Order, Claim, Account,…
- Example: customer as **central subject** in some DW
  - The complete DW is **organized** by customer
  - It may consist of hundreds or more physical tables that are related

Integrated



- A DW consolidates data from **most or all** of an organization's operational systems, ensuring *consistency* for accurate analysis and reporting.
- **Key aspects of integration**:
  - Data standardization (uniform formatting)
  - Measurement consistency (alignment of units – e.g. kilograms vs pounds)
  - Conflict resolution (duplicates/ conflicting keys)
  - Global consistency (discrepancies in naming or representation across data sources

Non-volatile



- **Static and Read-Only:** Once stored, data in the DW is rarely updated or deleted. It remains static and read-only for consistent reporting.
- **No Overwrites:** Instead of modifying existing records, changes result in the creation of new versions or snapshots.
- **Load-Only Process:** Data is loaded into the DW but not updated, ensuring historical accuracy.
- **Key Difference**:
  - *Operational Systems:* Data is frequently modified (insert, update, delete).
  - *Data Warehouse:* Data is retained as-is, with changes recorded as new entries.

Time Variant

- **Historical Perspective:** Data in a DW is tied to specific time periods, enabling analysis from a historical viewpoint.
- The changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time. → ***New snapshots are taken***
- **Time Horizons**:
  - Operational Systems: Typically maintain data for 60–90 days.
  - Data Warehouses: Retain data for 5–10 years or more to support long-term trend analysis.

Subject-Oriented

Data collected relates to a particular subject (e.g sales, customer, etc.)

Integrated

Data has been standardized regardless of how it is stored in the source systems.

Data Warehouse

Non-Volatile

Data in the DW is hardly ever over-written or deleted - once committed, the data is static, read-only, and retained for future reporting

Time Variant

The data collected changes with time. Newer snapshots record the updates.

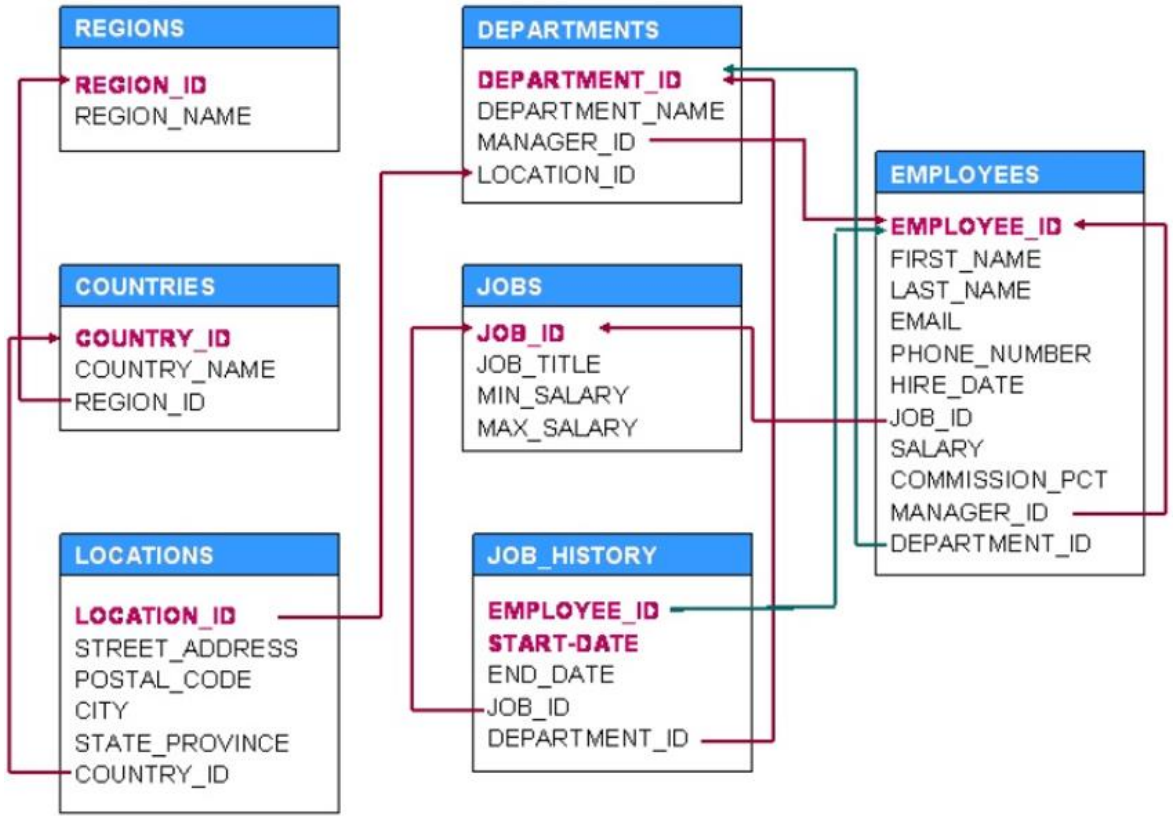
Defining Data Warehouse

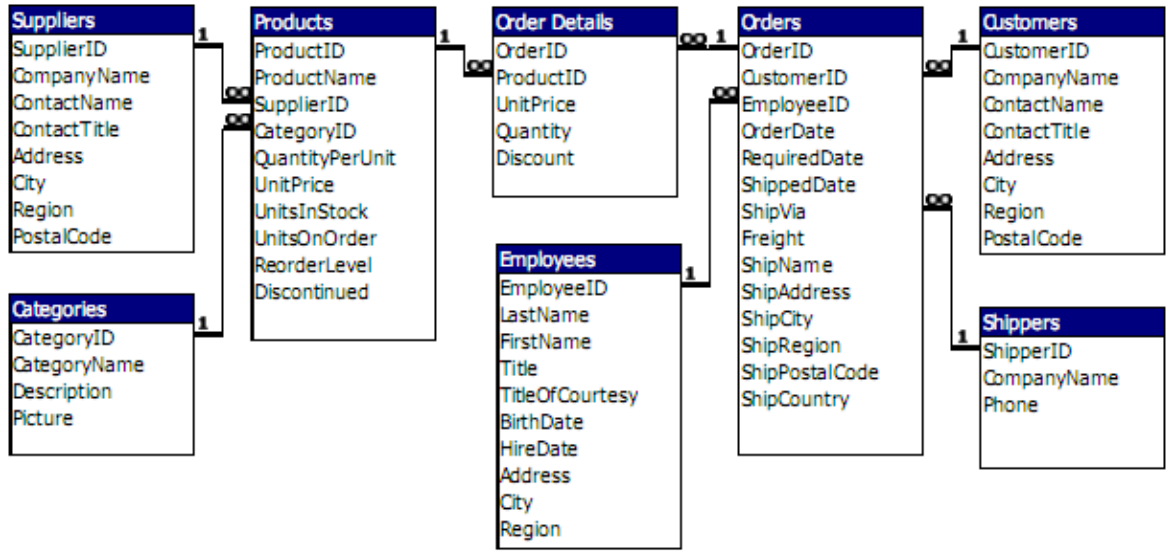
Construct an Orders ERD

- **Required tables**
  - Orders
  - Employees
  - Customer
  - Item
  - Warehouse

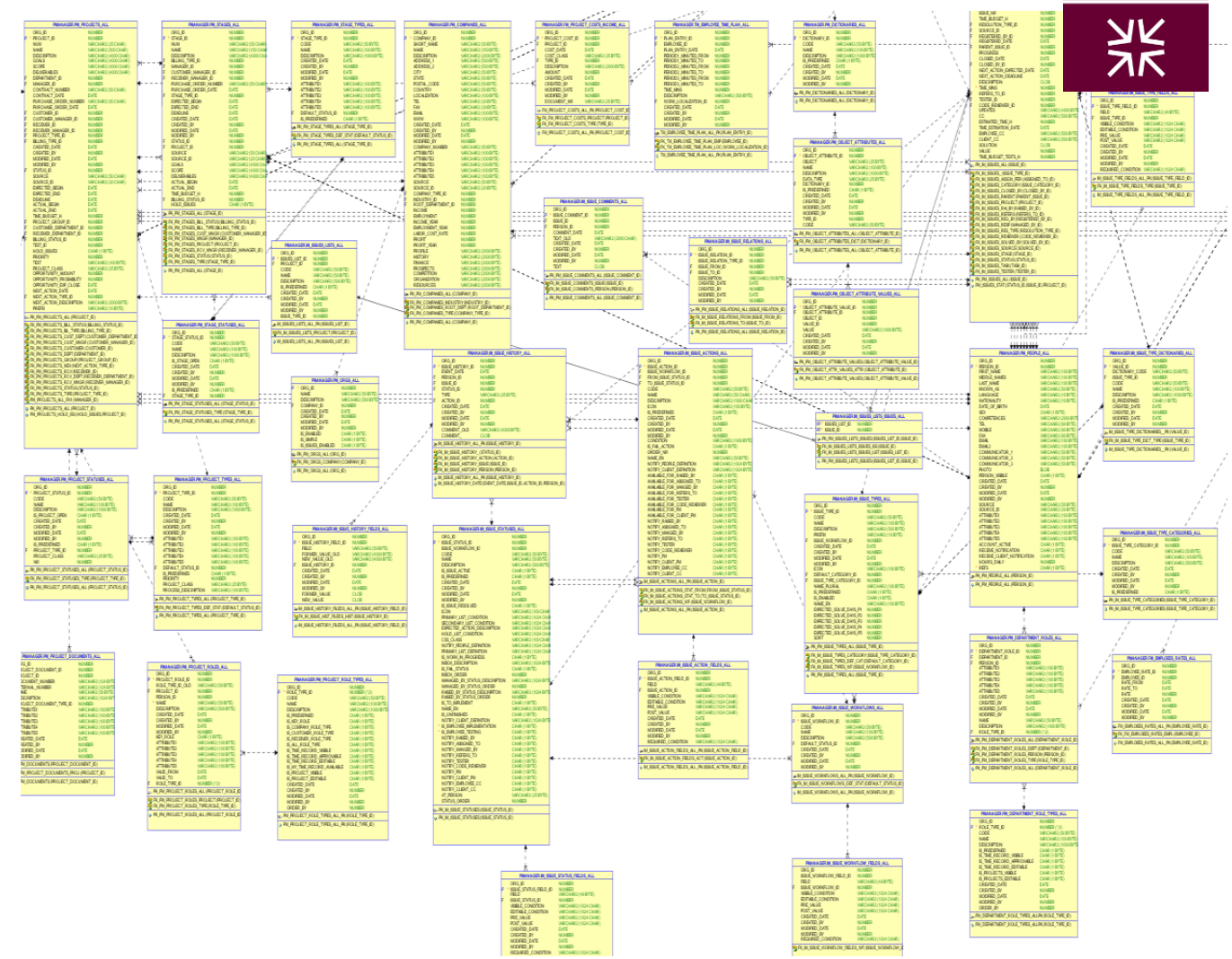
- **Relations:**
  - One order is placed by a single customer and completed by a single employee.
  - Employees can be assigned many orders.
  - Customers can place multiple orders.
  - A single order may have multiple items and items can be ordered in multiple orders.
  - The order is fulfilled by a single warehouse

Naïve Databases





Database Spaghetti



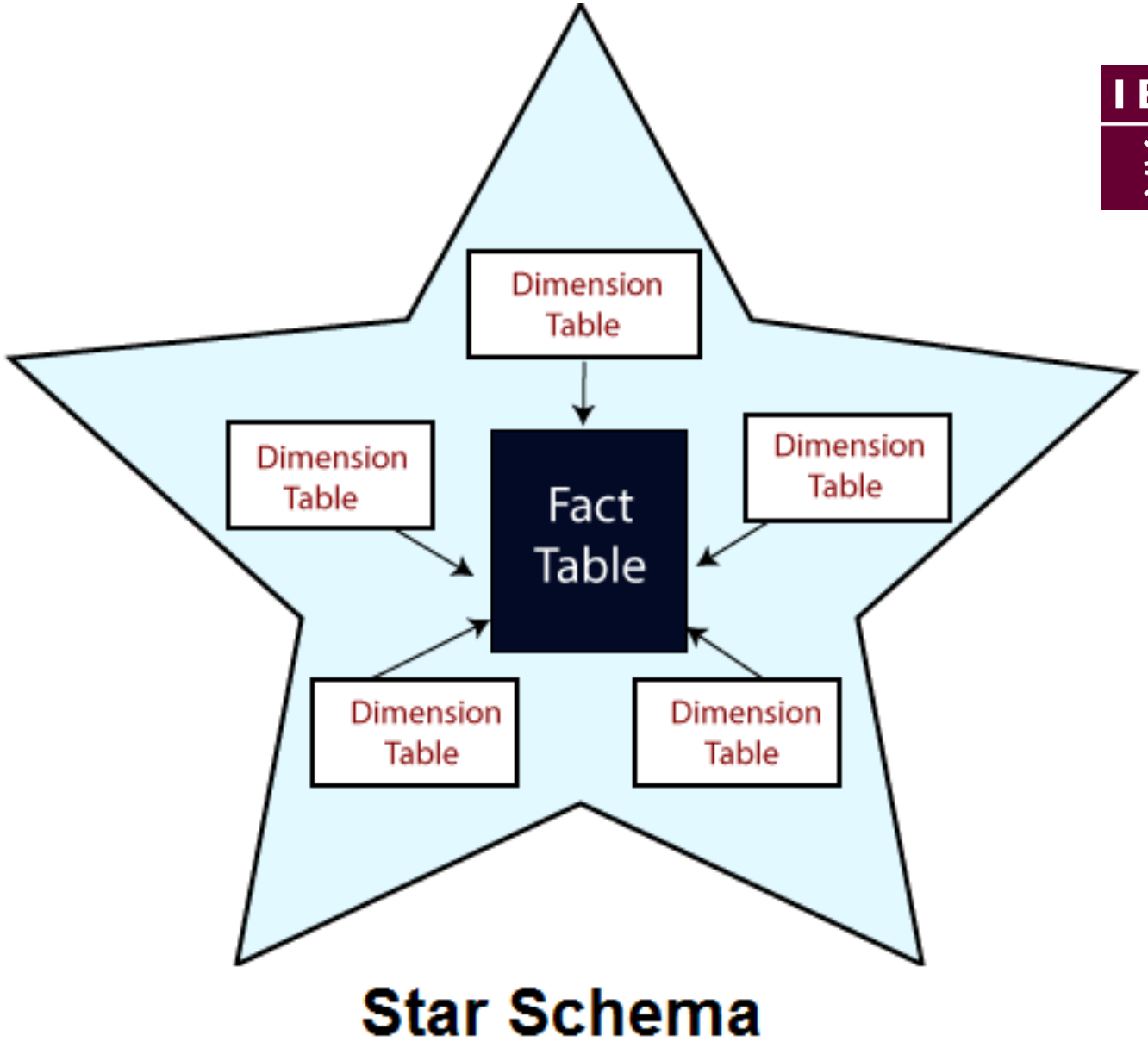
Database Spaghetti

- Normalization decomposes the structure into greater tables.
- Random changes in database leads to further problems.
- Growing business leads to exponential growth of tables.

Problems:

- Multi-table joins → long, time-consuming, complex
- In the olden days, hardware availability was an issue: small size yet too expensive to acquire.

Introducing Star Schema



Star Schema

Star Schema →*Analyzing Facts across Dimensions*

- The **fact table** stores two types of information: numeric values and dimension attribute values. Using a sales database as an example:
- **Numeric Value cells (facts)**
  - Are unique to each row or data point and do not correlate or relate to data stored in other rows.
  - These might be facts about a transaction, such as an orderID, total amount, net profit, order quantity or a **business measure**.
- **The Dimension Attribute Values**
  - Store the **foreign key value** for a row in a related dimensional table.
  - Many rows in the fact table will reference this type of information. So, for example, it might store the sales employee ID, a date value, a product ID or a branch office ID.

Star Schema →*Analyzing Facts across Dimensions*

- **Dimension tables** store supporting information to the fact table.
- The dimension tables contain the textual context associated with a business process measurement event.
- They describe the "who, what, where, when, how, and why" associated with the event.
- Each star schema database has at least one dimension table, but will often have many. Each dimension table will relate to a column in the fact table with a dimension value, and will store additional information about that value.

Examples

- The employee dimension table may use the employee ID as a key value and can contain information such as the employee's name, gender, address or phone number.
- A product dimension table may store information such as the product name, manufacture cost, color or first date on market.

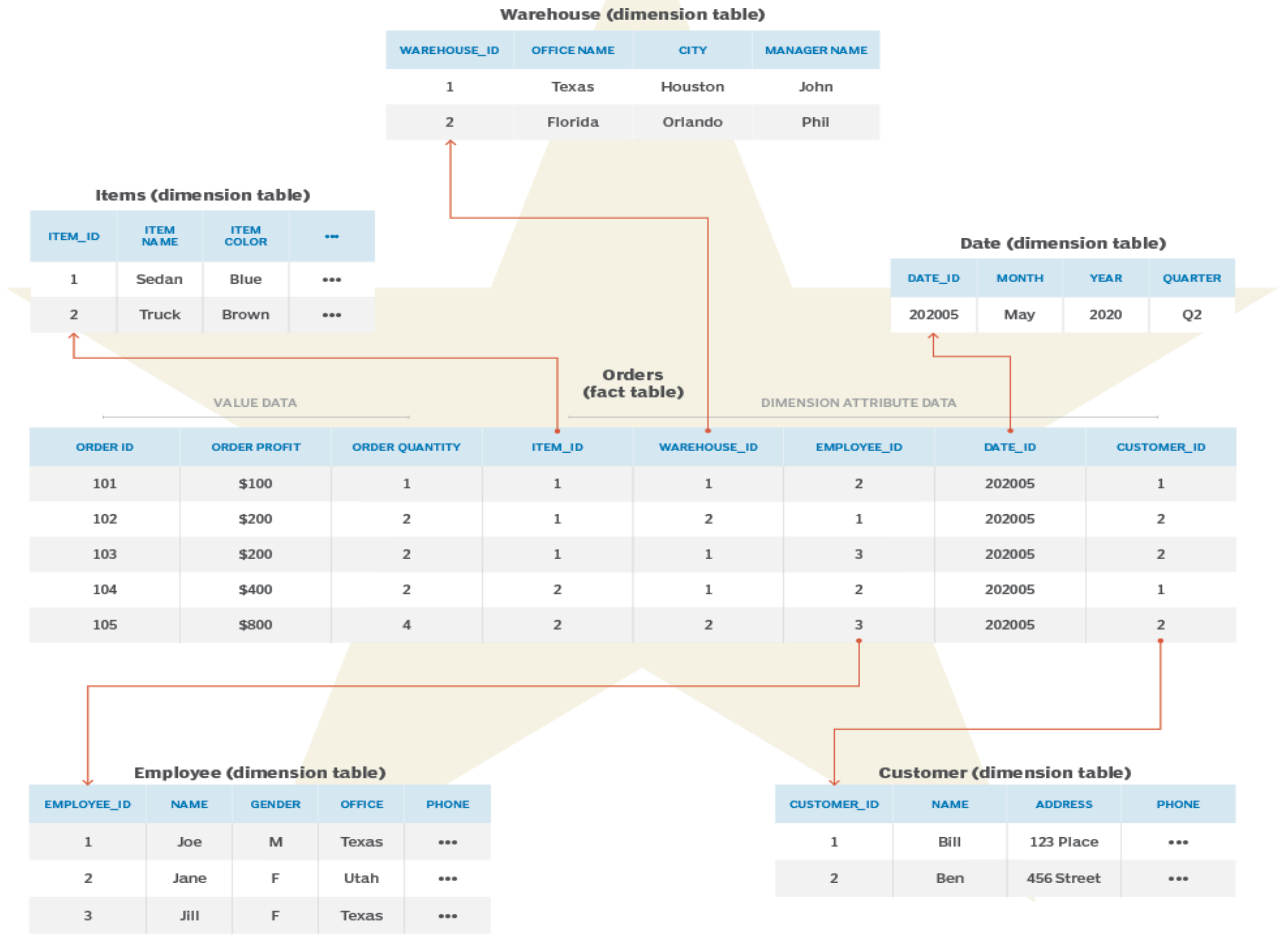
Construct an Orders ERD

- **Required tables**
  - Orders
  - Employees
  - Customer
  - Item
  - Warehouse

- **Relations:**
  - One order is placed by a single customer and completed by a single employee.
  - Employees can be assigned many orders.
  - Customers can place multiple orders.
  - A single order may have multiple items and items can be ordered in multiple orders.
  - The order is fulfilled by a single warehouse

ERD to Star Schema

Star schema


Warehouse (dimension table)
WAREHOUSE_ID | OFFICE NAME | CITY | MANAGER NAME
1 | Texas | Houston | John
2 | Florida | Orlando | Phil
Items (dimension table)
ITEM_ID | ITEM NAME | ITEM COLOR | •••
1 | Sedan | Blue | •••
2 | Truck | Brown | •••
Date (dimension table)
DATE_ID | MONTH | YEAR | QUARTER
202005 | May | 2020 | Q2
Orders (fact table)
VALUE DATA
DIMENSION ATTRIBUTE DATA
ORDER ID | ORDER PROFIT | ORDER QUANTITY | ITEM_ID | WAREHOUSE_ID | EMPLOYEE_ID | DATE_ID | CUSTOMER_ID
101 | $100 | 1 | 1 | 1 | 2 | 202005 | 1
102 | $200 | 2 | 1 | 2 | 1 | 202005 | 2
103 | $200 | 2 | 1 | 1 | 3 | 202005 | 2
104 | $400 | 2 | 2 | 1 | 2 | 202005 | 1
105 | $800 | 4 | 2 | 2 | 3 | 202005 | 2
Employee (dimension table)
EMPLOYEE_ID | NAME | GENDER | OFFICE | PHONE
1 | Joe | M | Texas | •••
2 | Jane | F | Utah | •••
3 | Jill | F | Texas | •••
Customer (dimension table)
CUSTOMER_ID | NAME | ADDRESS | PHONE
1 | Bill | 123 Place | •••
2 | Ben | 456 Street | •••
©2021 TECHTARGET. ALL RIGHTS RESERVED TechTarget