Data warehouse (IS 422)

Lecture 3

Main Data warehouse types

Dr. Wael Abbas

2023 - 2024

Data warehousing

- **ADWH** is a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which aids the strategic use of data. **(Inmon Bill)**
- Whereas data warehouse is a repository of data, **data warehousing** is literally the entire process.
- **Data warehousing** is a discipline that results in applications that provide decision support capability, allows ready access to business information, and creates business insight.

Data warehousing main types

- The three main types of data warehouses are :
  - Data marts (DMs)
  - operational data stores (ODS)
  - enterprise data warehouses (EDW)

Data Mart

- Whereas a data warehouse combines databases across an entire enterprise, a data mart (DM) is usually smaller and focuses on a particular subject or department.
- A DM is a subset of a data warehouse, typically consisting of a single subject area (e.g., marketing, operations).
- A DM can be either **dependent** or **independent.**

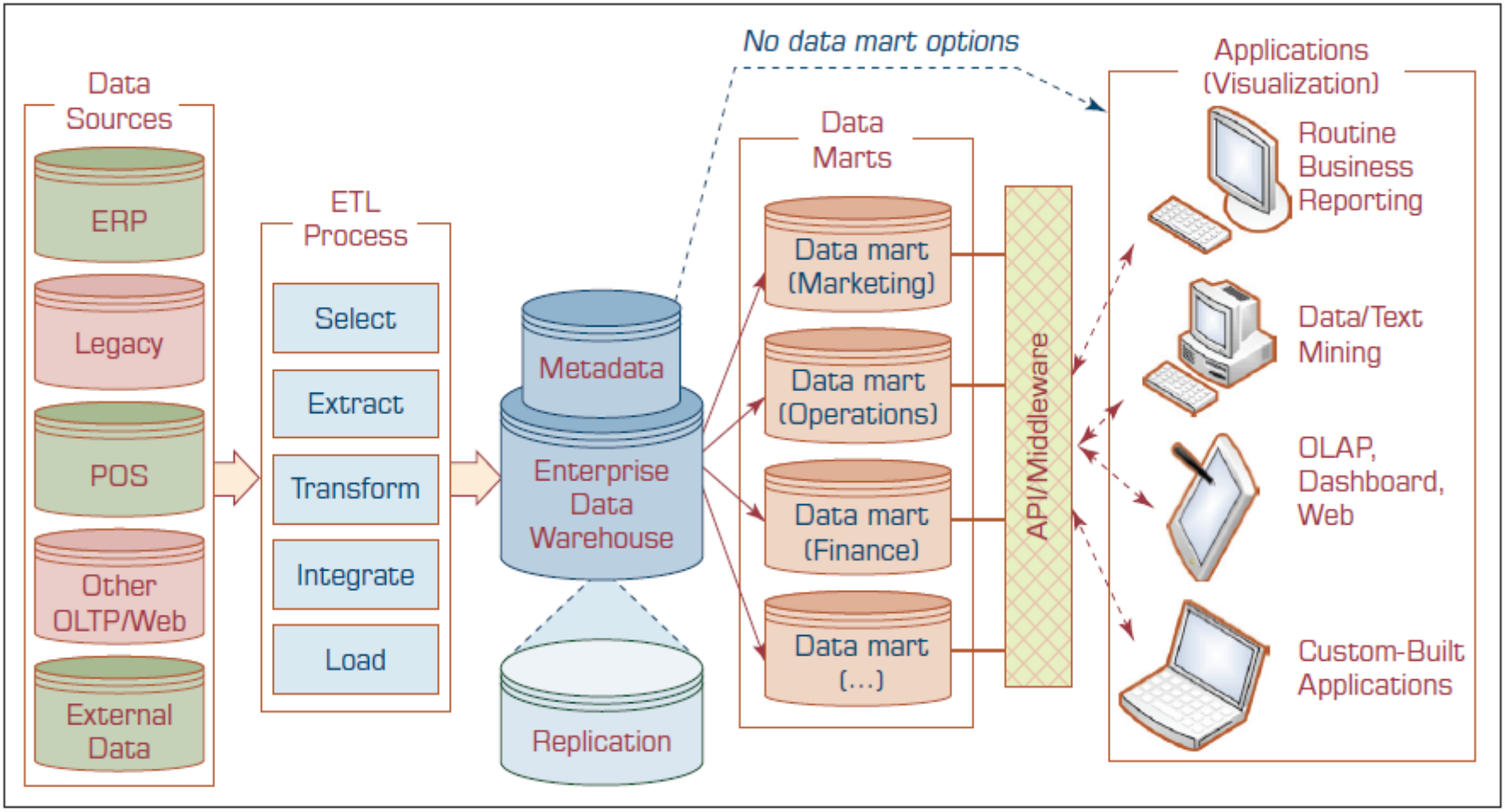
Data Mart

- **A dependent data mart** is a subset that is created directly from the data warehouse. It has the advantages of using a consistent data model and providing quality data.
- Dependent DMs support the concept of a single enterprise-wide data model, but the data warehouse must be constructed first.
- A dependent DM ensures that the end user is viewing the same version of the data that is accessed by all other data warehouse users. The high cost of data warehouses limits their use to large companies.

Data Mart

- Many firms use a lower-cost, scaled-down version of a data warehouse referred to as an **independent** DM subset.
- **independent data mart** is a small warehouse designed for a strategic business unit or a department, but its source is not an **EDW**.

Data warehouse



operational data stores (ODS)

- **Operational data store** (ODS) provides a fairly recent form of customer information file.
- This type of database is often used as an interim **staging** area for a data warehouse.
- Unlike the static contents of a data warehouse, the contents of an ODS are updated throughout the course of business operations.
- An ODS is used for short-term decisions involving mission-critical applications rather than for the medium- and long-term decisions associated with an **EDW**.

operational data stores (ODS)

- An ODS is similar to short-term memory in that it stores only very recent information. In comparison, a data warehouse is like long-term memory because it stores permanent information.
- An ODS consolidates data from multiple source systems and provides a near–real-time, integrated view of volatile, current data.
- The exchange, transfer, and load (**ETL**) processes for an ODS are identical to those for a data warehouse.
- ODS is based on change data capture(CDC).This approach used to determine the data change and apply action based on this change.

Enterprise data warehouses (EDW)

- An **enterprise data warehouse (EDW)** is a large-scale data warehouse that is used across the enterprise for decision support.
- The large-scale nature of an EDW provides integration of data from many sources into a standard format for effective BI and decision support applications.
- EDWs are used to provide data for many types of decision support systems (DSS), including customer relationship management (CRM), supply chain management (SCM), business performance management (BPM), business activity monitoring, product life cycle management, revenue management, and sometimes even knowledge management systems.

Dimensional modeling

- Like enterprise relationship (ER) modeling, **dimensional modeling** is a logical design technique**.**
- Dimensional modeling is much better suited for business intelligence (BI) applications and data warehousing (DW).
- It depicts business processes throughout an enterprise and organizes that data and its structure in a logical way.
- The purpose of dimensional modeling is to enable BI reporting, query, and analysis.
- The key concepts in dimensional modeling are **facts**, **dimensions**, and **attributes**. There are different types of facts, depending on whether they can be added together.

Dimensional modeling

- Dimensions can have different hierarchies, and have attributes that define the who, what, where, and why of the dimensional model. The grain, or level of **granularity**, is another key concept with dimensional modeling, as it determines the level of detail.
- Facts, dimensions, and attributes can be organized in several ways, called **schemas**. The choice of schema depends on variables such as the **type** of reporting that the model needs to facilitate and the type of BI tool being used.
- Building a dimensional model includes additional puzzle pieces such as calendar and time dimensions, and more complicated pieces such as degenerative dimensions and consolidated fact tables.

Dimensional modeling

- **Facts**
  - A **fact** is a measurement of a business activity, such as a business event or transaction, and is generally numeric. Examples of facts are sales, expenses, and inventory levels. Numeric measurements may include counts, dollar amounts, percentages, or ratios.
  - A **fact** is a collection of related data consisting of **measures**.
  - In a data warehouse, facts are implemented in the core tables in which all of the numeric data is stored.
  - Facts can be **aggregated** or **derived**. For example, you can sum up the total revenue or calculate the profitability of a set of sales transactions.

Dimensional modeling

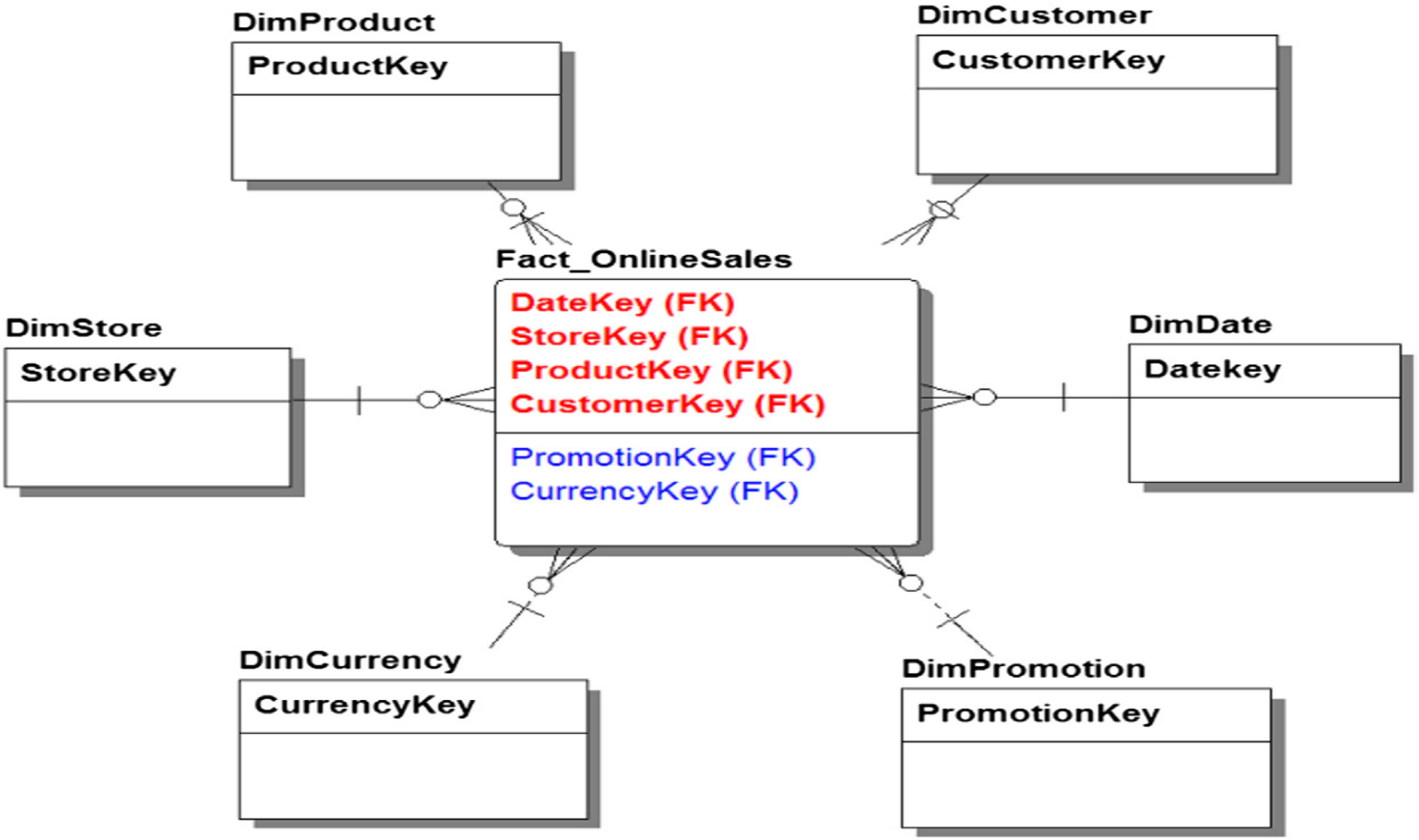
- **Facts**
  - A **fact** is a measurement of Facts provide the measurements of how well or how poorly the business is performing.
  - A fact is also referred to as organizational performance measure.
  - Fact contains **redundancy**.
  - **Ninety-percent** of the data in a dimensional model is typically located in the fact tables.
  - The key dimensional modeling design concerns when working with the data in fact tables are how to minimize and standardize it and make it consistent.

Dimensional modeling

- **Facts**
  - Fact tables are composed of two types of columns: **keys and measures.**
  - The first, the key column, consists of a group **of foreign keys (FK)** that point to the primary keys of dimensional tables that are associated with this fact table to enable business analysis.
  - The relationships between fact tables and the dimensions are one-to-many

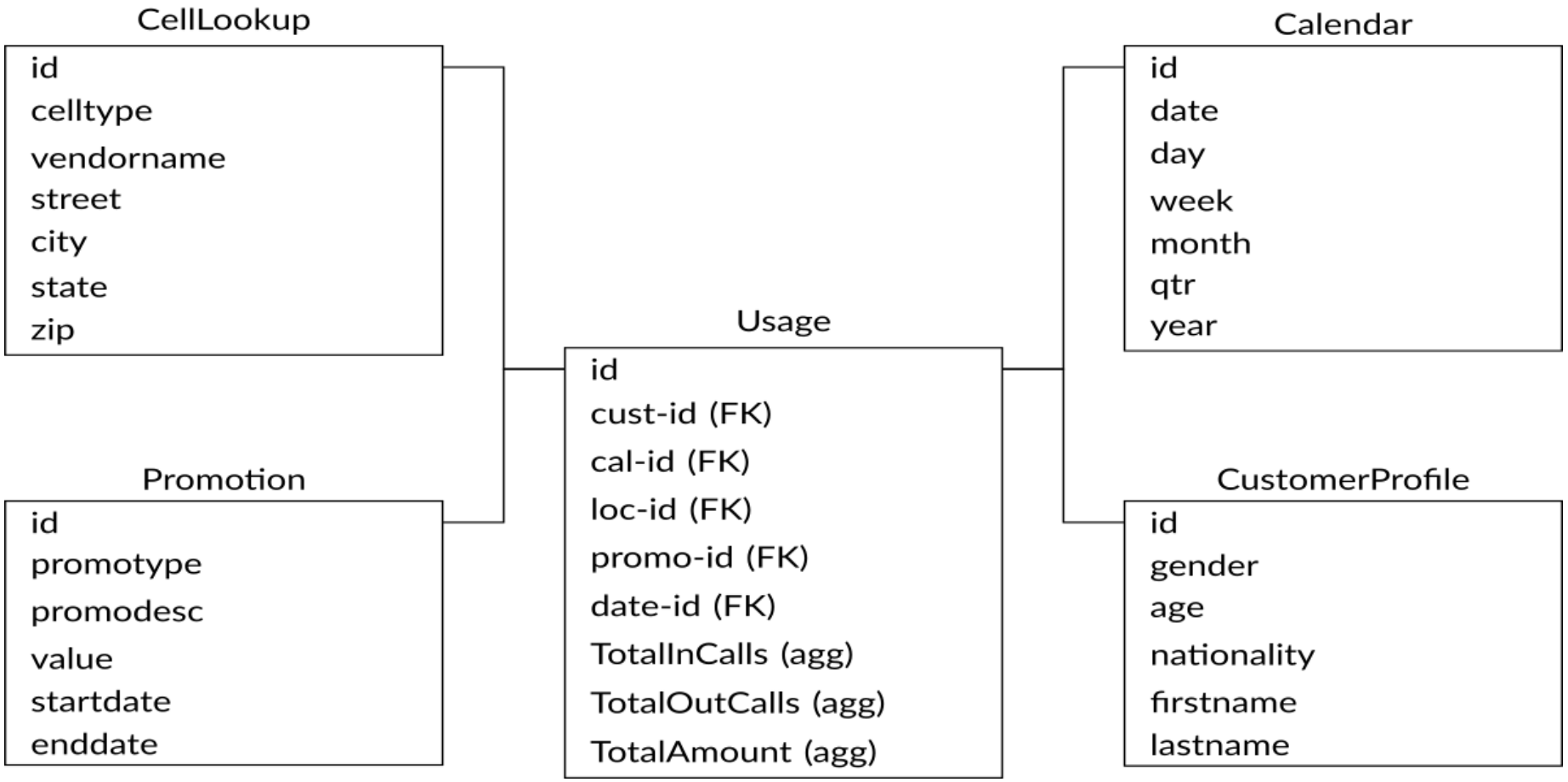
Dimensional modeling

- **Facts**



Dimensional modeling

- **Facts**



Dimensional modeling

- **Facts**
  - The second type of column in a fact table is the actual **measures** of the business activity such as the **sales revenue and order quantity**.
  - Every measurement has a grain, which is the level of detail in the measurement of an event such as a unit of measure, currency used, or ending daily balance of an account.
  - For example SalesQuantity, SalesAmount, ReturnAmount, ReturnQuantity, DiscountAmount, DiscountQuantity, and TotalCost that apply to a customer for a product purchased at a specific time.
  - All of these measures are related to the business event (the sale) that the fact represents and they have a level of granularity related to that event.

Dimensional modeling

- **Facts**

.

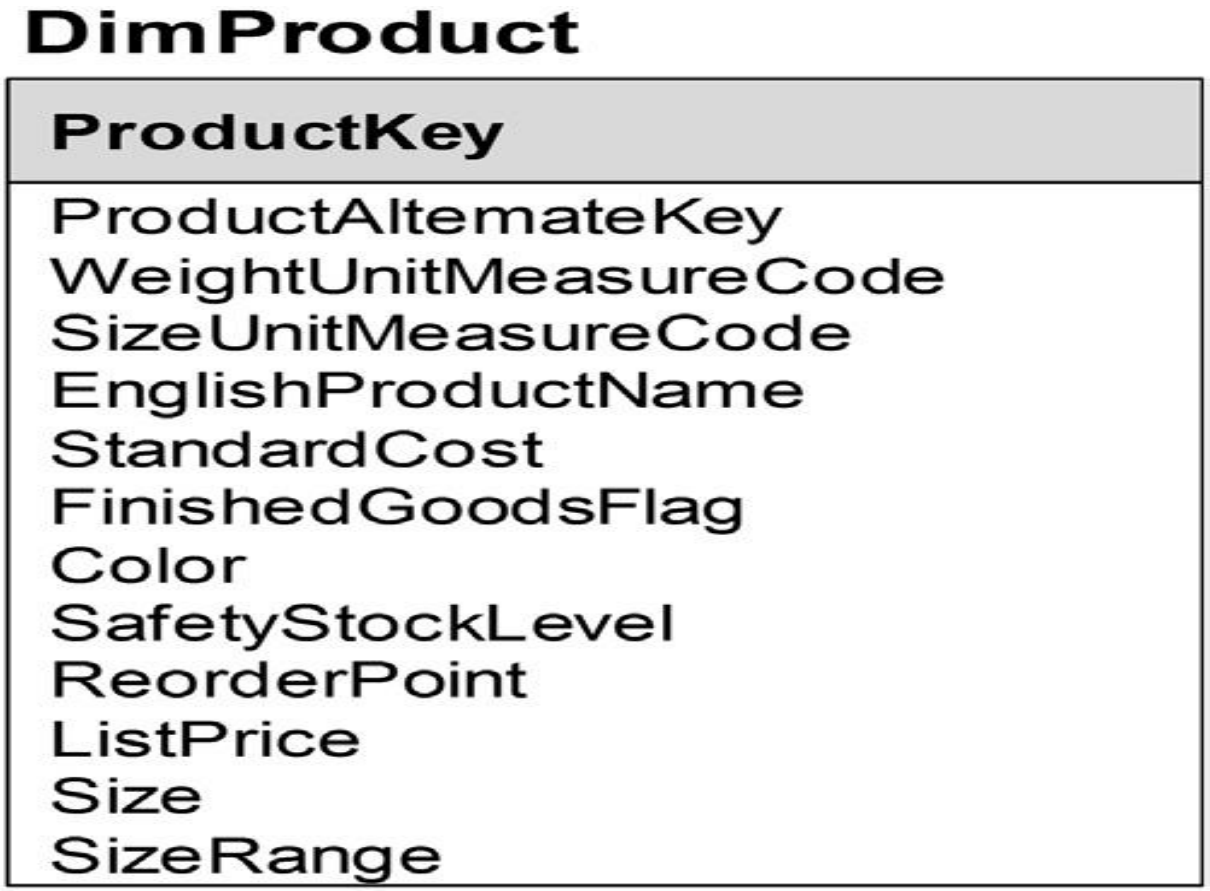
Measures in fact tables

Dimensional modeling

- **Dimensions**
  - A dimension is an entity that establishes the business context for the measures (facts) used by an enterprise.
  - Dimensions define the who, what, where, and why of the dimensional model, and group similar attributes into a category or subject area. Examples of dimensions are product, geography, customers, employees, and time.
  - Whereas facts are **numeric**, dimensions are **descriptive** in nature (although some of those descriptions, such as a product list price, may be numeric).

Dimensional modeling

- **Dimensions**



Schema Types

- **Schema types are :**
  1. **Star schema**
  2. **Snowflake schema**

You choose which schema to use when building the dimensional model by considering these questions:

1. What kind of analysis are you trying to perform on that data and how complex is it?
2. What are the analytical requirements and restrictions?
3. How consistent is the data you want to query and analyze?
4. What BI tool do you plan to use? Although different tools may appear to show the same type of data, results, and graphs, they can be very different under the covers, and rely on a specific schema for the best results.

Note : most slides in this file produced from : Sharda, Ramesh, Dursun Delen, and Efraim Turban. *Business intelligence, analytics, and data science: a managerial perspective*. pearson,

Schema Types

Star schema

- The star schema (sometimes referenced as star join schema) is the most commonly used and the simplest style of dimensional modeling.
- A star schema contains a central **fact table** surrounded by and connected to **several dimension** tables.
- The fact table contains a large number of rows that correspond to observed facts and external links (i.e., foreign keys).
- A fact table contains the descriptive attributes needed to perform decision analysis and query reporting, and foreign keys are used to link to dimension tables.

Schema Types

Star schema

- The decision analysis attributes consist of performance measures, operational metrics, aggregated measures (e.g., sales volumes, customer retention rates, profit margins, production costs, scrap rate), and all the other metrics needed to analyze the organization's performance.
- In other words, the **fact** table primarily addresses what the data warehouse supports for **decision analysis.**
- Surrounding the central fact tables (and linked via foreign keys) are **dimension tables**.

Schema Types

Star schema

- The dimension tables contain classification and aggregation information about the central fact rows.
- Dimension tables contain attributes that describe the data contained within the fact table; they address how data will be analyzed and summarized.
- Dimension tables have a **one-to-many** relationship with rows in the central **fact table.**
- In querying, the dimensions are used to **slice and dice** the numerical values in the fact table to address the requirements of an ad hoc information need.

Schema Types

Star schema

Star Schema Main Characteristics

1. **Simplicity**: It is the simplest type of DWH schemas.
2. **Query effectiveness**: Because of simplicity, It needs less join to query the data (It is optimized to query large dataset).
3. **Data Redundancy and Large Table Size**: Due to de-normalization, it has a data redundancy, and the table size is huge.
4. **Most** used and **widely** supported.

Note : This slide from : https://gability.com/en/courses/big-data-in-depth/02-dwh/03-architecture/10-data-modeling/04-schema-types/

Schema Types

Star schema

Star Schema additional Characteristics

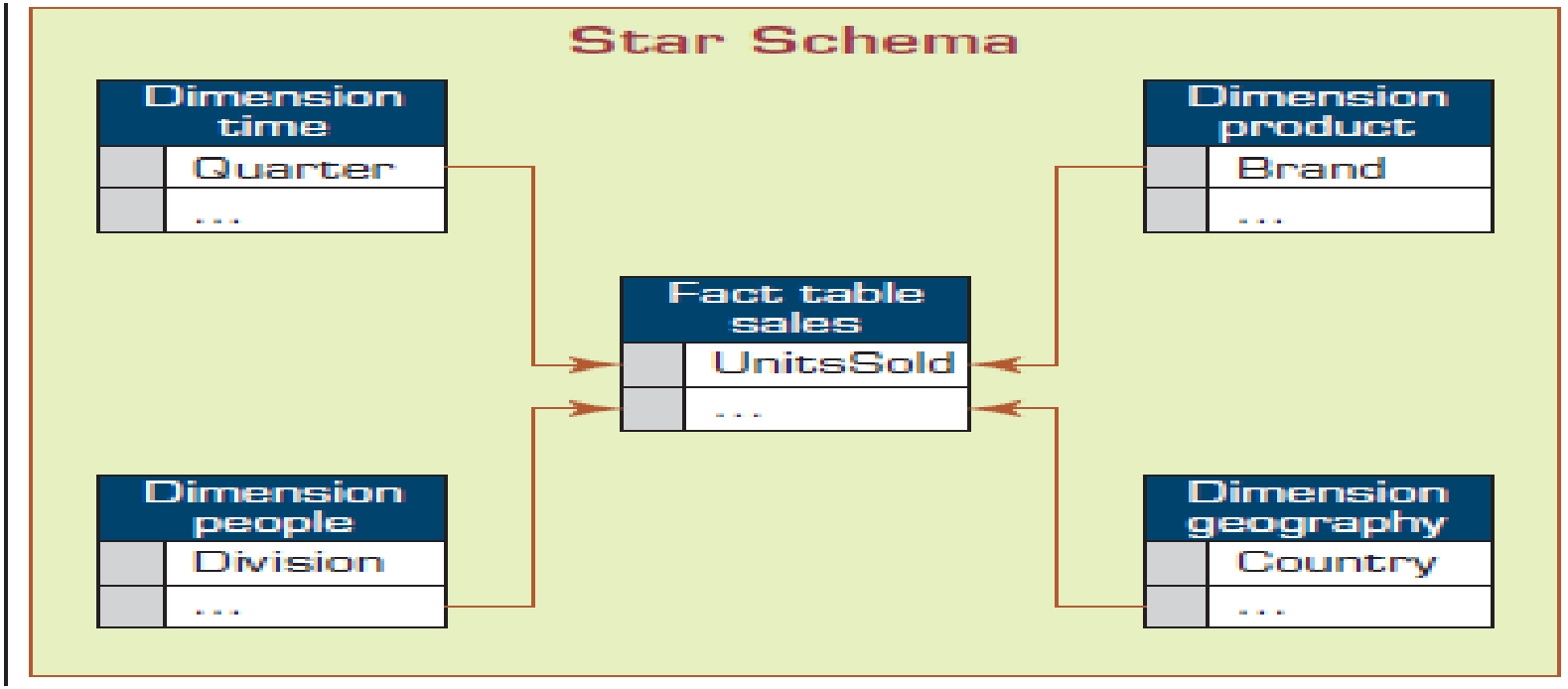
1. Dimensions represented by one one-dimension table.
2. The dimension table are not joined to each other
3. The fact table would contain key and measure.
4. Data integrity is not enforced due to the de-normalized structure.

Note : This slide from : https://gability.com/en/courses/big-data-in-depth/02-dwh/03-architecture/10-data-modeling/04-schema-types/

Schema Types

Star schema

- The star schema is designed to provide fast query-response time, simplicity, and ease of maintenance for read-only database structure.



Schema Types

Snowflake schema

- Closely related to the star schema, the snowflake schema is represented by centralized fact tables (usually only one), which are connected to multiple dimensions.
- It takes the star schema, with the facts surrounded by denormalized dimensions, one step further by normalizing the hierarchies within a **particular dimension**.
- Each level in the dimensional hierarchy becomes its own dimensional table with parent keys created to link the hierarchical structure together.
- The fact table stores the foreign key to the lowest level of the dimensional hierarchy.

Schema Types

Snowflake schema

- Dimensions are normalized into multiple related tables, whereas the star schema's dimensions are denormalized, with each dimension being represented by a single table.

Snowflake Schema Characteristics

1. **Extension**: Snowflake is an extension of the Star Schema.
2. **Normalized**: Dimension tables are normalized; this means every dimension may expand into additional tables.

Schema Types

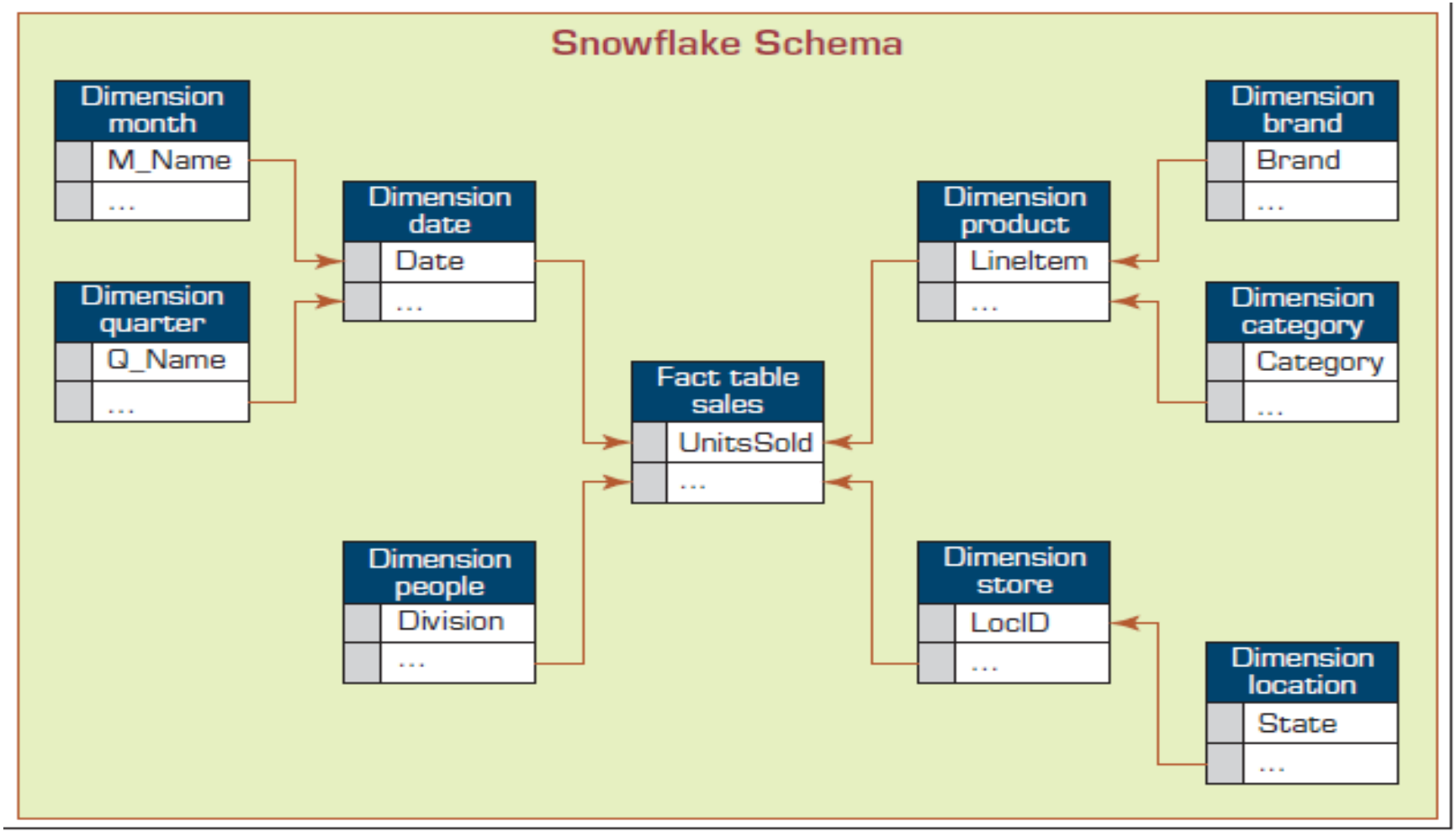
Snowflake schema .

Snowflake Schema Characteristics

3. **Disk Space Efficiency**: Due to its normalization methodology, it uses less desk space, which enhances the query as we scan less data size.
4. **Complicated**: Due to the normalization query needs to join more table in some cases to get the data which reduces the performance.

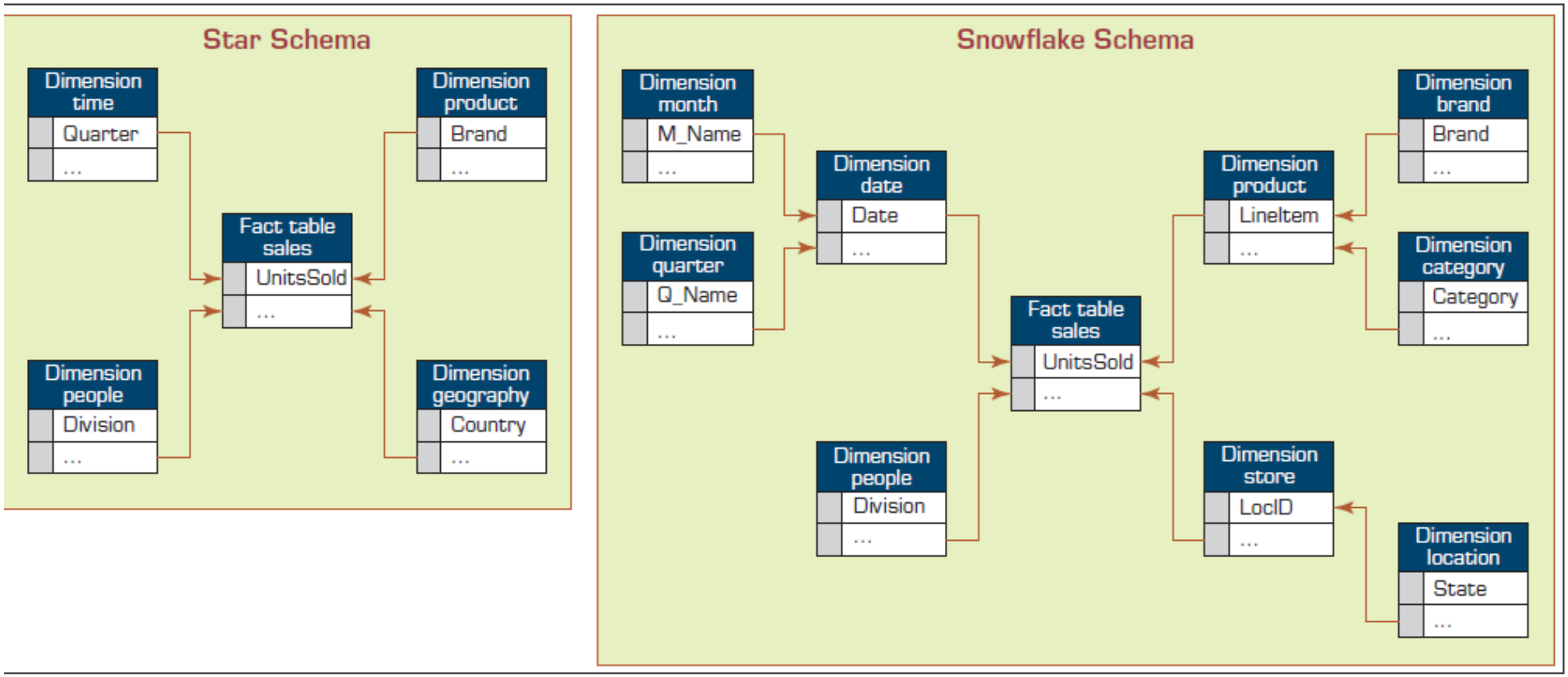
Schema Types

Snowflake schema .



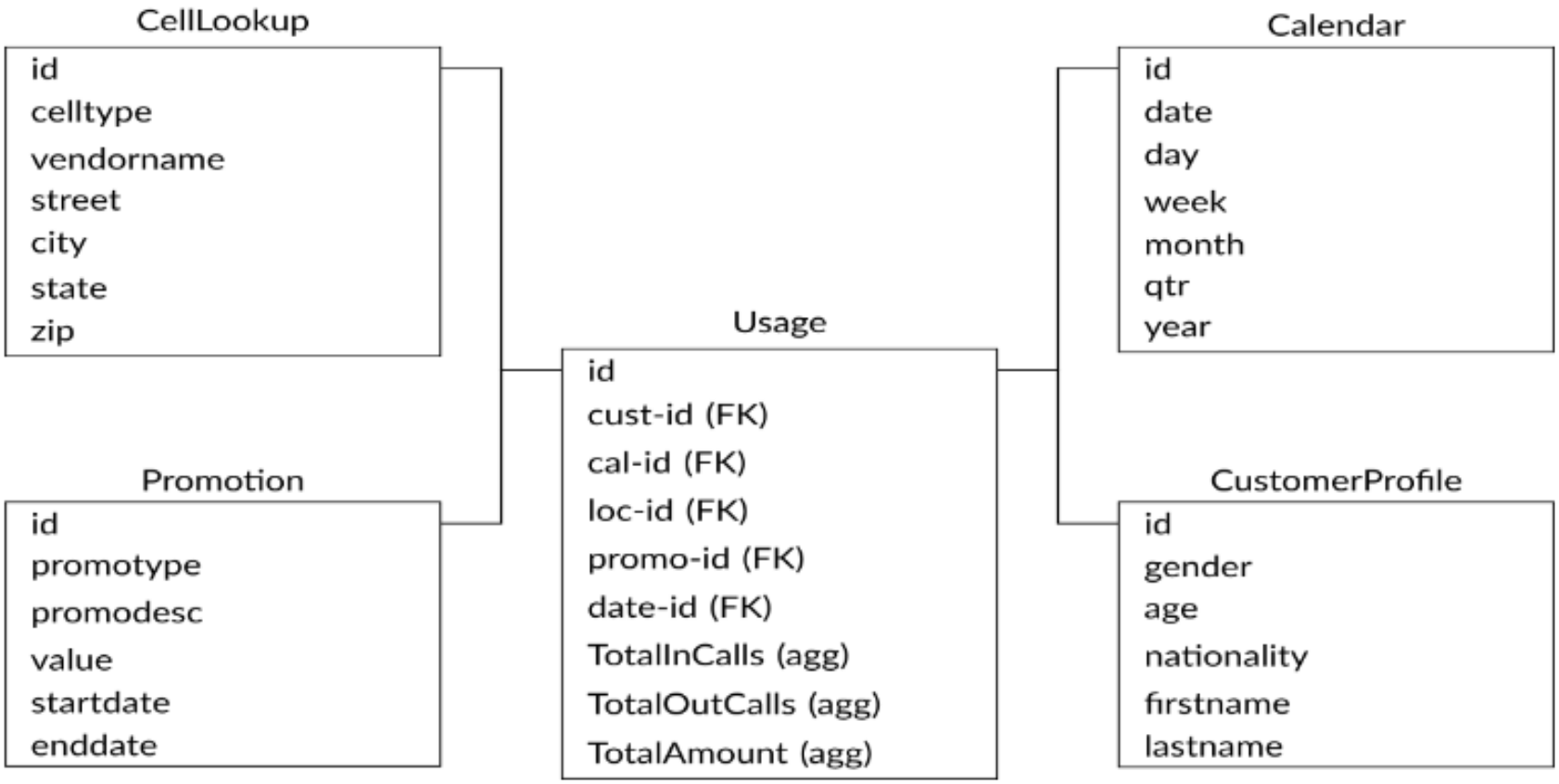
Schema Types

Star vs snowflake example



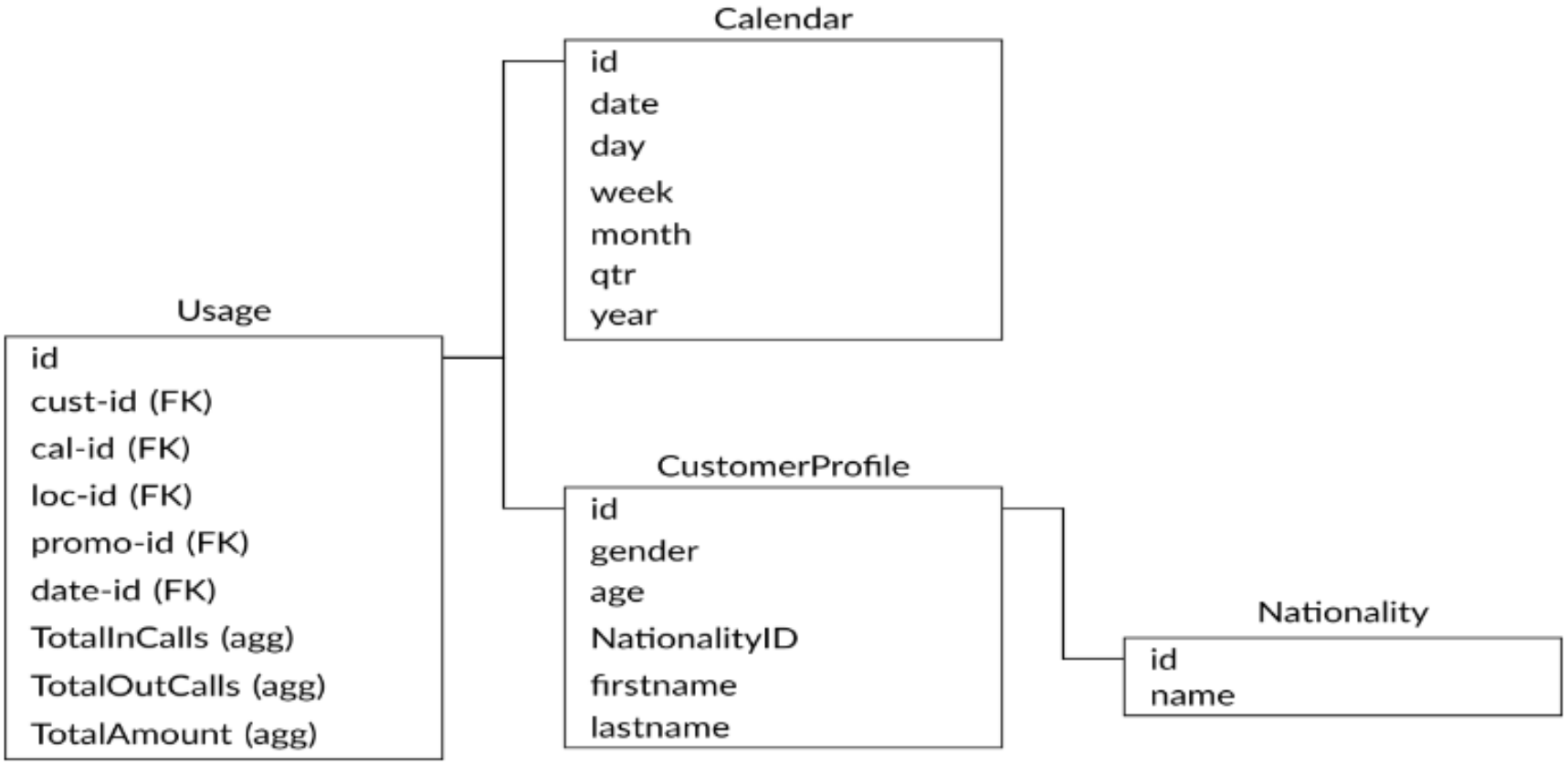
Schema Types

Star



Schema Types

Snowflake



Schema Types

Star	Snowflake
Dimension represented by one-table	Dimension tables are expanded into multi-tables
Fact table surrounded by dimension tables	Fact table surrounded by Hierarchy of dimension tables
Less join	Requires many joins
Simple Design	Very Complex Design
De-normalized Data structure	Normalized Data Structure
High level of Data redundancy	Very low-level data redundancy
Maintenance is difficult	Maintenance is easier
Good for datamarts with simple relationships (1:1 or 1:many)	Good for core to simplify (many: many)

Schema Types

Multi-fact star models

- Examples of dimensional models, whether star, snowflake, or multidimensional, usually depict them with a single fact table surrounded by dimensions. Although this is a popular illustration and a terrific approach that simplifies explanations, **the reality is that data in an enterprise requires multiple facts**.
- Examples of facts are sales, expenses, inventory, and the many other business events associated with running an enterprise.
- For example , shows two facts: store sales and store inventory (*Tbl_Fact_Store_Sales* and *Tbl_Fact_ Store_Inventory*).

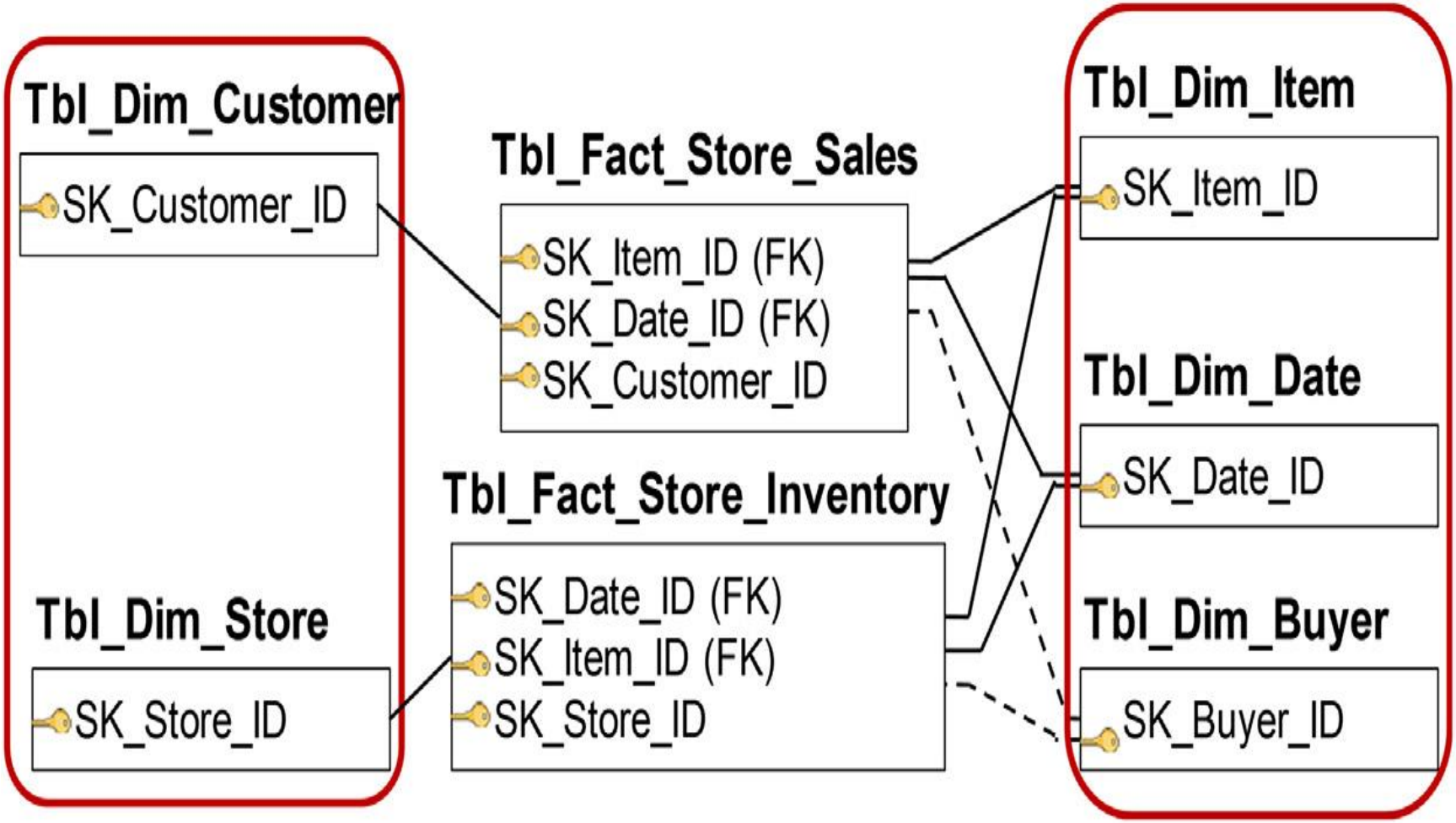
Schema Types

Multi-fact star models

- On the right side are three dimensions that both facts share: the item dimension, the date dimension, and the buyer dimension. The shared dimensions are referred to as **conformed dimensions**.
- On the left side of the example are the customer and store dimensions that are not shared across the two facts. The *Tbl_Dim_Customer* is related to *Tbl_Fact_Store_Sales*, whereas the *Tbl_Dim_Store* is related to *Tbl_Fact_Store_Inventory*

Schema Types

Multi-fact star models



ENTITY RELATIONSHIP vs DIMENSIONAL MODELING

- **ER** modeling, sometimes referred to as normalized modeling, is the standard for transactional systems (also called operational or online transactional processing (**OLTP**) systems).
- Dimensional modeling is the best practice for BI and OLAP systems. The reason for this stems from how the two systems process and manipulate data.
- The data in operational systems(OLTP) is being constantly updated. The main job of these workhorses is transaction throughput. They have to maintain and update a large number of records, so they have to be able to handle high amounts of consistent throughput with large volumes.
- Compare that with the BI and reporting systems, which are generally not updating data. Rather, they are copying, extracting, transforming, and loading from operational systems. They are not doing realtime updates like operational systems.

ENTITY RELATIONSHIP vs DIMENSIONAL MODELING

- The focus is not on transactional throughput once the data is loaded. The focus is on query performance, and gathering and aggregating large sets of data. That's a critical operation. Transactional systems update a small numbe of data records quite often, whereas BI systems gather and aggregate large sets of data records, but not so often—sometimes over large spans of time.
- The data from the BI system then get used for reports.
- The operational system and its accompanying normalized models have minimal **redundancy** to support the high transaction throughput.
- in BI reporting, there is a lot of **redundancy**. After all, the warehouse and data marts and whatever else you store on the reporting side are all copies of data that originally was in the operational systems. The BI system depends heavily on indexes and partitioning because it needs much more storage space for reporting.

ENTITY RELATIONSHIP vs DIMENSIONAL MODELING

Operational Systems	BI and Analytics
Normalized models are standard for OLTP	Dimensional models are standard for BI and OLAP
Highly volatile	Generally not updated
Transaction throughput (updating and maintaining numerous records) is critical	Query performance (gathering and aggregating large sets of records) is critical
Characteristics supporting use of normalized models:	**Characteristics supporting use of dimensional models:**
Minimal redundancy (normalization)	Increased redundancy (denormalization)
Limited index use	Increased index use
Efficient use of storage space	Increased storage space
Eliminate inconsistent data	Consolidate inconsistent data
Few maintenance concerns	Increased maintenance issues