Yarmouk University
Faculty of Information Technology and Computer Sciences

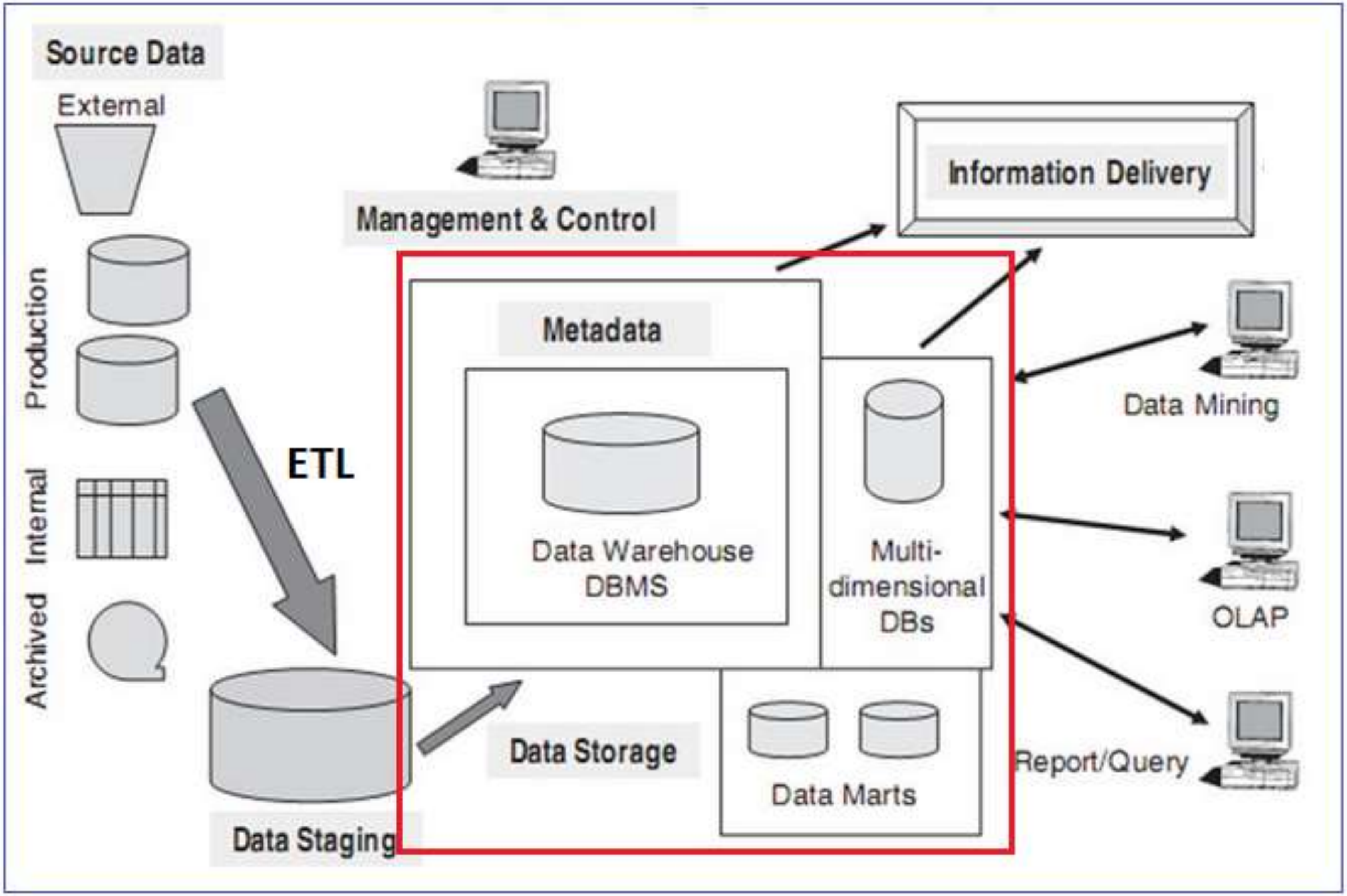
CIS 367: Data Warehousing

Topic 2: Dimensional Modeling

Dr. Rafat Hammad

Acknowledgements: Most of these slides have been prepared based on various online tutorials and presentations, with respect to their authors and adopted for our course. Additional slides have been added from the mentioned references in the syllabus

COMPONENTS OF A DATA WAREHOUSE



TOPIC 2 : OUTLINE

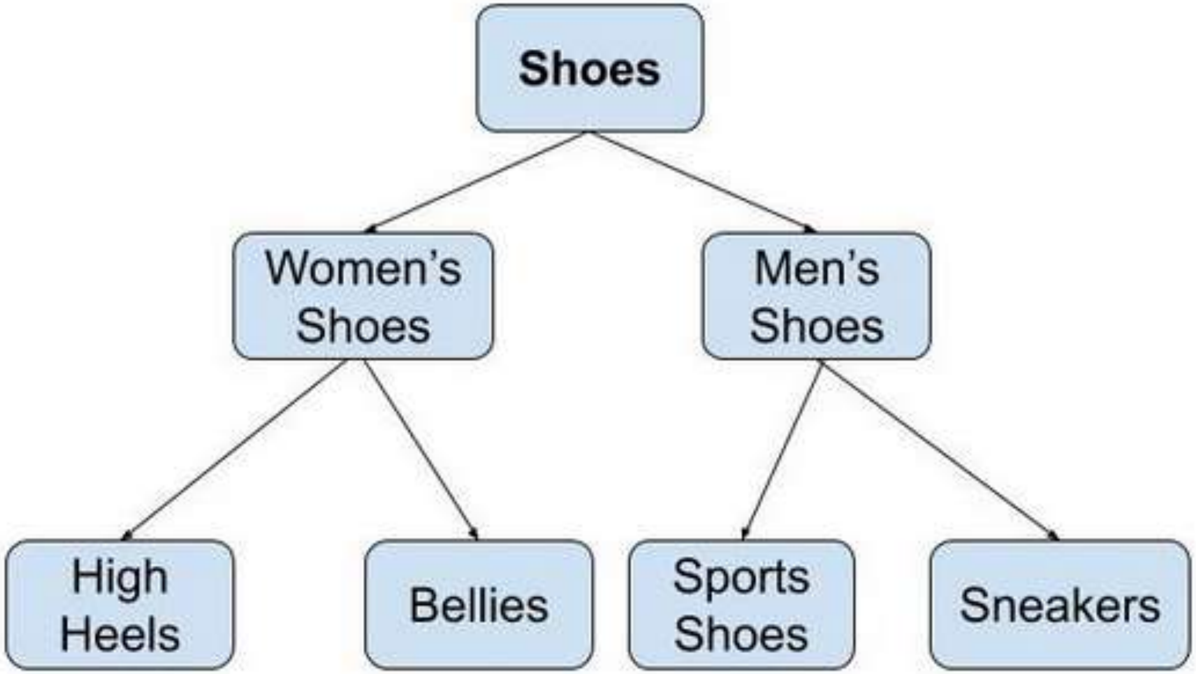
- Database Models
- Steps of Dimensional Modelling
- Schemas of Dimensional Modelling

DATABASE MODELS

- A Database model is a set of concepts to describe the structure of a database and defines how data will be stored, accessed and updated in a database management system.
- While the **Relational Model** is the most widely used database model, there are other models too:
  - Hierarchical Model
  - Network Model
  - Entity-relationship Model
  - Relational Model
  - **Dimensional Model**

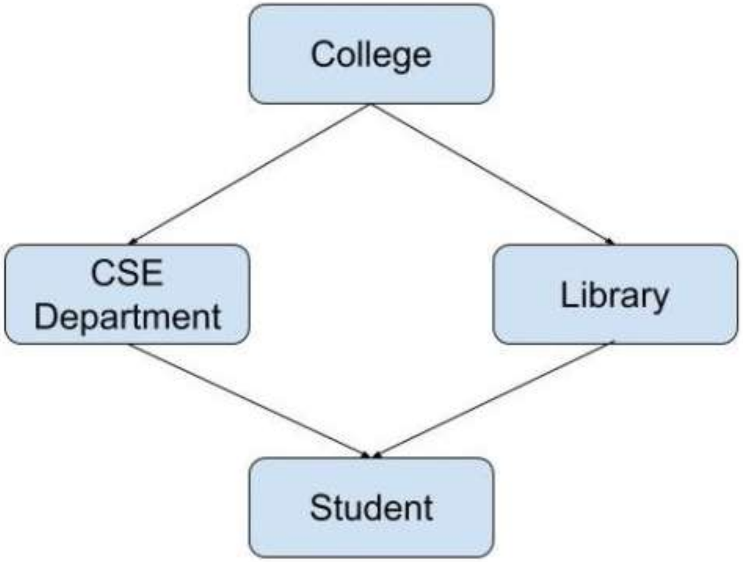
HIERARCHICAL MODEL

- Hierarchical Model was the first DBMS model. This model organizes the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node. .



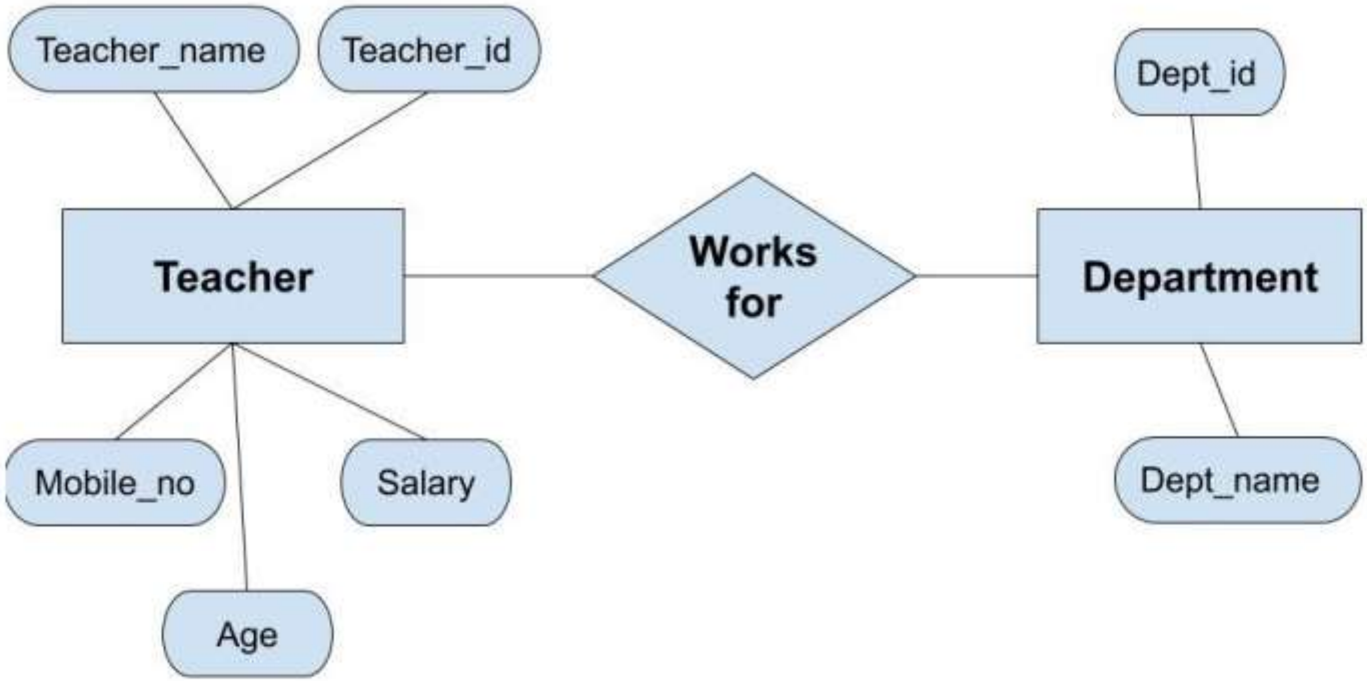
Network Model

- This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have more than one parent.
- This model has the ability to manage one-to-one relationships as well as many-to-many relationships.



Entity-relationship (ER) Model

- In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes. Different entities are related using relationships.



RELATIONAL MODEL

- Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called relations in the relational model.

EMPLOYEE

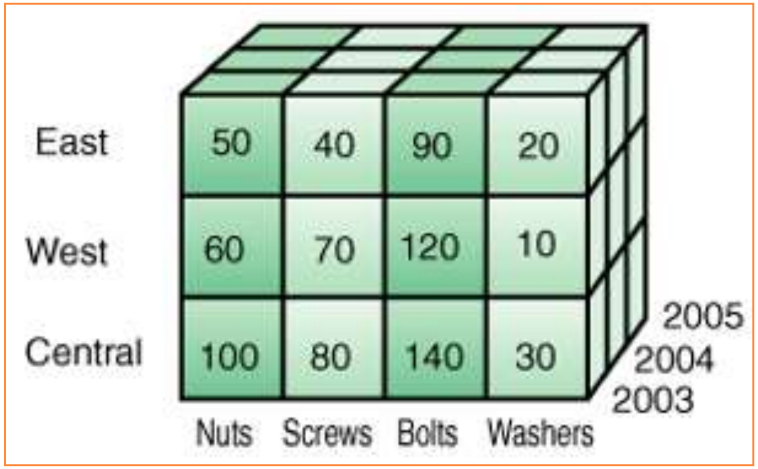
EMP_ID	EMP_NAME	ADDRESS	DEPT_ID
100	Joseph	Clinton Town	10
101	Rose	Fraser Town	20
102	Mathew	Lakeside Village	10
103	Stewart	Troy	30
104	William	Holland	30

DEPARTMENT

DEPT_ID	DEPT_NAME
10	Accounting
20	Quality
30	Design

Dimensional Model

- This is a variation of the relational model designed to facilitate improved analytical processing. While the relational model is optimized for online transaction processing (OLTP), this model is designed for online analytical processing (OLAP).
- Each cell in a dimensional database contains data about the dimensions tracked by the database. Visually, it's like a collection of cubes, rather than two-dimensional tables.



Relational vs. Dimensional Data Model

Relational Data Modeling	Dimensional Data Modeling
Data is stored in RDBMS	Data is stored in RDBMS or Multidimensional databases
Tables are units of storage	Cubes are units of storage
Data is normalized and used for OLTP. Optimized for OLTP processing	Data is de-normalized and used in data warehouse and data mart. Optimized for OLAP
Several tables and chains of relationships among them	Few tables and fact tables are connected to dimensional tables
Volatile(several updates) and time variant	Non volatile and time invariant
SQL is used to manipulate data	MDX is used to manipulate data
Detailed level of transactional data	Summary of bulky transactional data(Aggregates and Measures) used in business decisions
Normal Reports	User friendly, interactive, drag and drop multidimensional OLAP Reports

Benefits of Dimensional Modeling

- Standardization of dimensions allows easy reporting across areas of the business.
- The dimensional model is very understandable by the business. This model is based on business terms, so that the business knows what each fact, dimension, or attribute means.
- Dimensional modeling creates a schema which is optimized for high performance.
- Dimensional models can comfortably accommodate change. Dimension tables can have more columns added to them without affecting existing business intelligence applications using these tables.

Elements of Dimensional Data Model

- These are the main elements of dimensional data model:
  1) Facts
  2) Dimensions
  3) Attributes
  4) Fact Tables
  5) Dimension Tables

1) Facts

- **Fact** is a data, usually numeric and additive, that can be examined and analyzed (e.g., sum, count, average, minimum, maximum).
- Examples include: *sales*, *cost*, and *profit*.
- Fact is also known by the following names:
  - Measure
  - Measurement
  - Metric
  - Fact

2) Dimensions

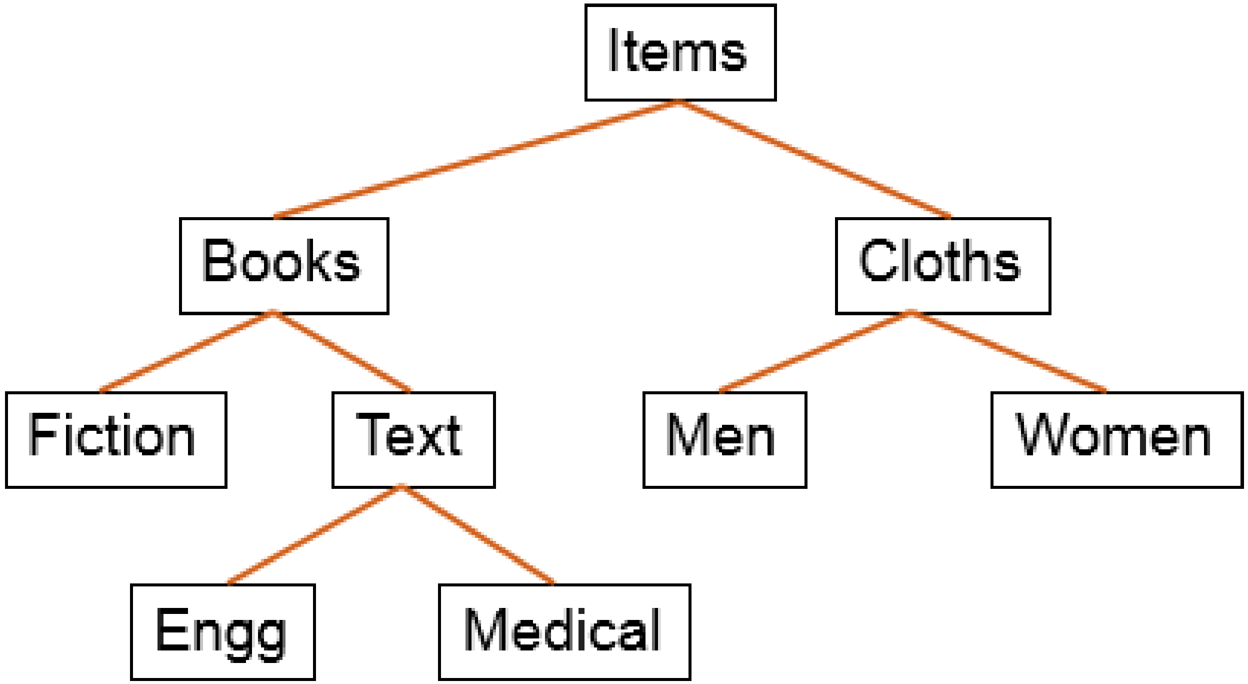
- **Dimensions** provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact.
- In ***Sales*** business process, for the fact quarterly sales number, dimensions would be
  - Who – Customer Names
  - Where – Location
  - What – Product Name
- In other words, a dimension is a window to view information in the facts.

Dimensions have Hierarchies

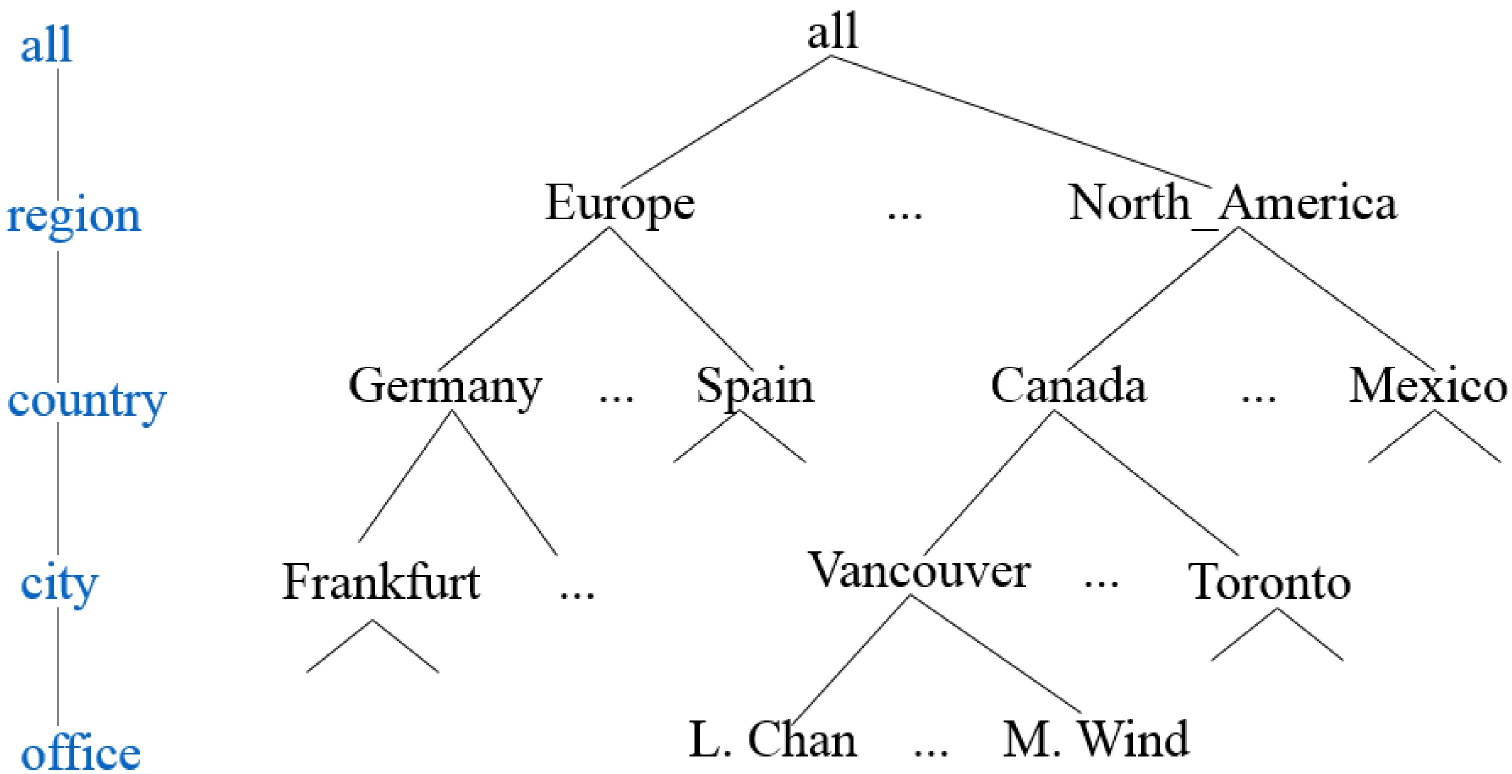
- You can put attributes into a hierarchical structure to assist user analysis
- One of the most common functions in Business Intelligence (BI) is to “drill down” to a more detailed level
- For example, Time hierarchy might be to go from Year to Quarter to Month to Day
- Another Time hierarchy might go from Year to Month to Week to Day to Hour

Dimensions have Hierarchies (cont.)

- Analysts tend to look at the data through dimension at a particular "level" in the hierarchy



DIMENSIONS HAVE HIERARCHIES (CONT.)



3) Attributes

- The Attributes are the various characteristics of the dimension.
- In the Location dimension, the attributes can be
  - State
  - Country
  - Zip code etc.
- Attributes are used to search, filter, or classify facts.
- Dimension Tables contain Attributes

4) Fact Table

- A fact table is a primary table in a dimensional model.
- A Fact Table contains
  - Measurements/facts
  - Foreign key to dimension table
- The primary key of a fact table is usually a composite key that is made up of all of its foreign keys.
- Large number of records.
- Only a few attributes.

5) Dimension Table

- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Each dimension table has a simple (non-composite) primary key that corresponds exactly to one of the components of the composite key in the fact table.
- Large number of attributes
- Textual attributes
- Attributes not directly related.
- Multiple hierarchies
- Ability to drill down/roll up
- Less number of records

Dimension Table – Surrogate Keys

- All natural keys are replaced with **surrogate keys**. Means that every join between fact and dimension tables is based on surrogate keys, not natural keys
- Values for the surrogate keys are typically simple auto-increment integer values
- Surrogate keys allows the data in the warehouse to have some independence from the data used and produced by the data warehouse systems.
- Surrogate key values have no meaning or purpose except to give each dimension a new column that serves as a primary key within the dimensional model instead of the operational key.
- Surrogate keys are simple and short, it speed-up the join performance.

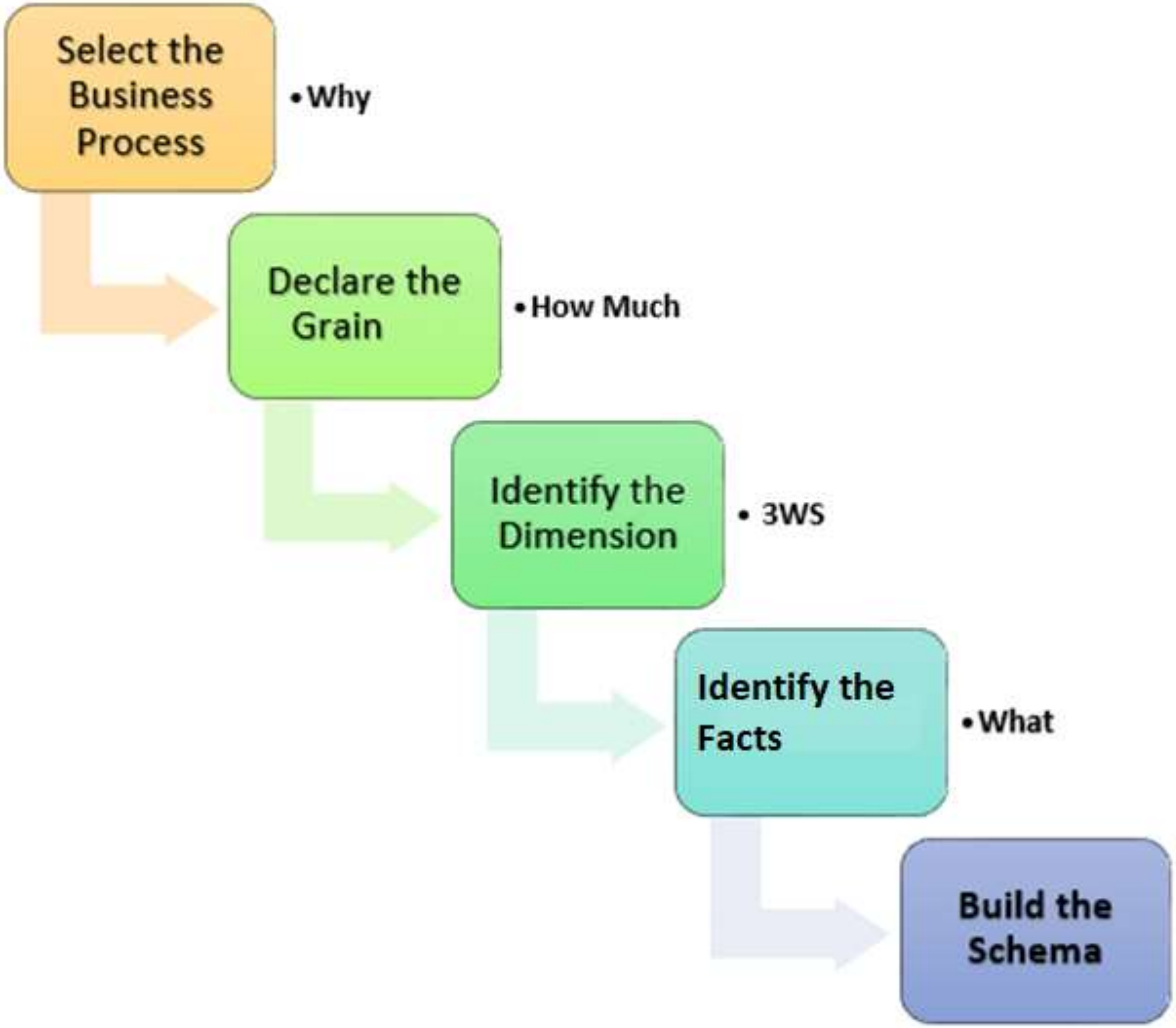
Topic 2 : outline

- Data Modelling
- Steps of Dimensional Modelling
- Schemas of Dimensional Modelling

Steps of Dimensional Modelling

- The accuracy in creating your Dimensional modeling determines the success of your data warehouse implementation. Here are the steps to create Dimension Model:
  1) Identify Business Process
  2) Identify Grain (level of detail)
  3) Identifying and Conforming the Dimensions
  4) Identify Facts
  5) Build Schema
- The model should describe the Why, How much, When/Where/Who and What of your business process

Steps of Dimensional Modelling (cont.)



Step 1) Identify the business process

- Identifying the actual business process (a subject) a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization.
- The selection of the business process also depends on the quality of data available for that process.
- It is the most important step of the Data Modelling process, and a failure here would have cascading and irreparable defects.

STEP 2) IDENTIFY THE GRAIN

- The Grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.
- **Example:** The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
  - So, the grain is "product sale information by location by the day."

Step 2) Identify the grain (cont.)

- During this stage, you answer questions like:
  - Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for data warehouse
  - Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives
  - How do the above two choices affect the database size?

Step 3) Identifying the Dimensions

- Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.
- **Example:** The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
  - *Dimensions*: Product, Location and Time
  - *Attributes*: For Product: Product key (Foreign Key), Name, Type, Specifications
  - *Hierarchies*: For Location: Country, State, City, Street Address, Name

Identifying the Dimensions - Example

Dimension Tables

Product

Model Name
Model Year
Package Styling
Product Line
Product Category
Exterior Color
Interior Color
First Year

Time

Year
Quarter …………..

Payment Method

Finance Type
Term ……………....

Customer Demo-graphics

Age
Gender …………..

Dealer

Dealer Name
City …………..

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Step 4) Identify the Fact

- This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.
- **Example:** CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
  - The fact here is Sum of Sales by product by location by time

IDENTIFY THE FACT - EXAMPLE

Automaker Sales

Fact Table

Actual Sale Price
MSRP
Options Price
Full Price
Dealer Add-ons
Dealer Credits
Dealer Invoice
Down Payment
Proceeds Finance

Dimensions

Time	Product	Payment Method	Customer Demo-graphics	Dealer	
Year	Model Name	Finance Type	Age	Dealer Name	
Quarter	Model Year	Term (Months)	Gender	City	
Month	Package Styling	Interest Rate	Income Range	State	
Date	Product Line	Agent	Marital Status	Single Brand Flag	
Day of Week	Product Category		House-hold Size	Date First Operation	
Day of Month	Exterior Color		Vehicles Owned		
Season	Interior Color		Home Value		
Holiday Flag	First Year		Own or Rent		

Facts: Actual Sale Price, MSRP, Options Price, Full Price, Dealer Add-ons, Dealer Credits, Dealer Invoice, Down Payment, Proceeds, Finance

Step 5) Build Schema

- In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are three popular schemas:
  - Star Schema
  - Snow Flake Schema
  - Fact Constellation Schema (Galaxy Schema)

TOPIC 2 : OUTLINE

- Data Modelling
- What is Dimensional Modelling?
- Schemas of Dimensional Modelling

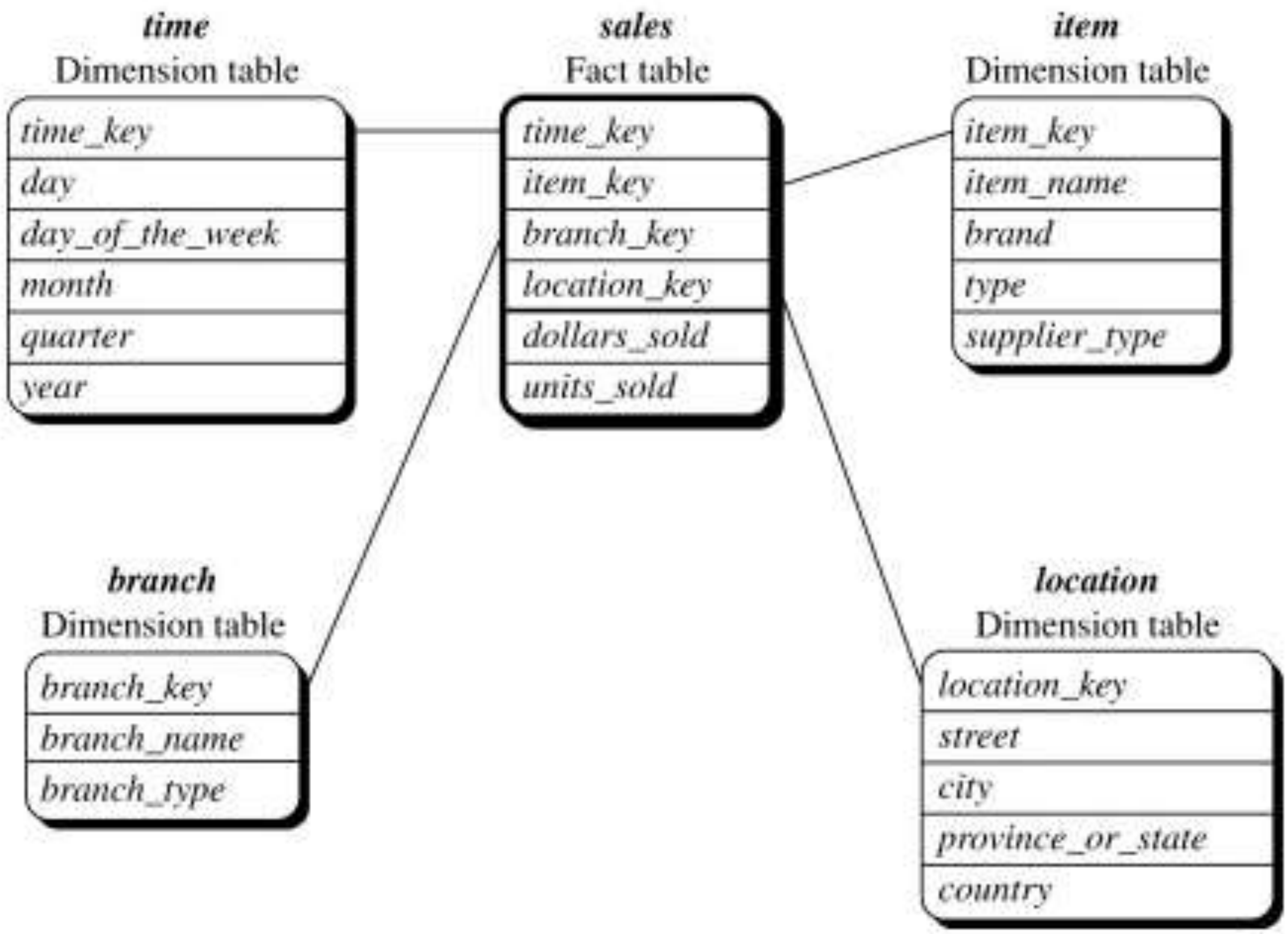
DIMENSIONAL SCHEMAS

- A schema is a collection of database objects, including tables, views, indexes, and synonyms.
- Multidimensional schema is especially designed to model data warehouse systems. The schemas are designed to address the unique needs of very large databases designed for the analytical purpose (OLAP).
- There is a variety of ways of arranging schema objects in the schema models designed for data warehousing.
- There are three popular schemas:
  - Star Schema
  - Snow Flake Schema
  - Fact Constellation Schema (Galaxy Schema)

Star Schema

- The star schema is the simplest and mostly used type of Data Warehouse schema.
- It is known as star schema as its structure resembles (يشبه) a star.
- In the Star schema, the center of the star can have one fact tables and numbers of associated dimension tables.
- It is optimized for querying large data sets.
- Every dimension in a star schema is represented with the only one-dimension table.

Star Schema (cont.)



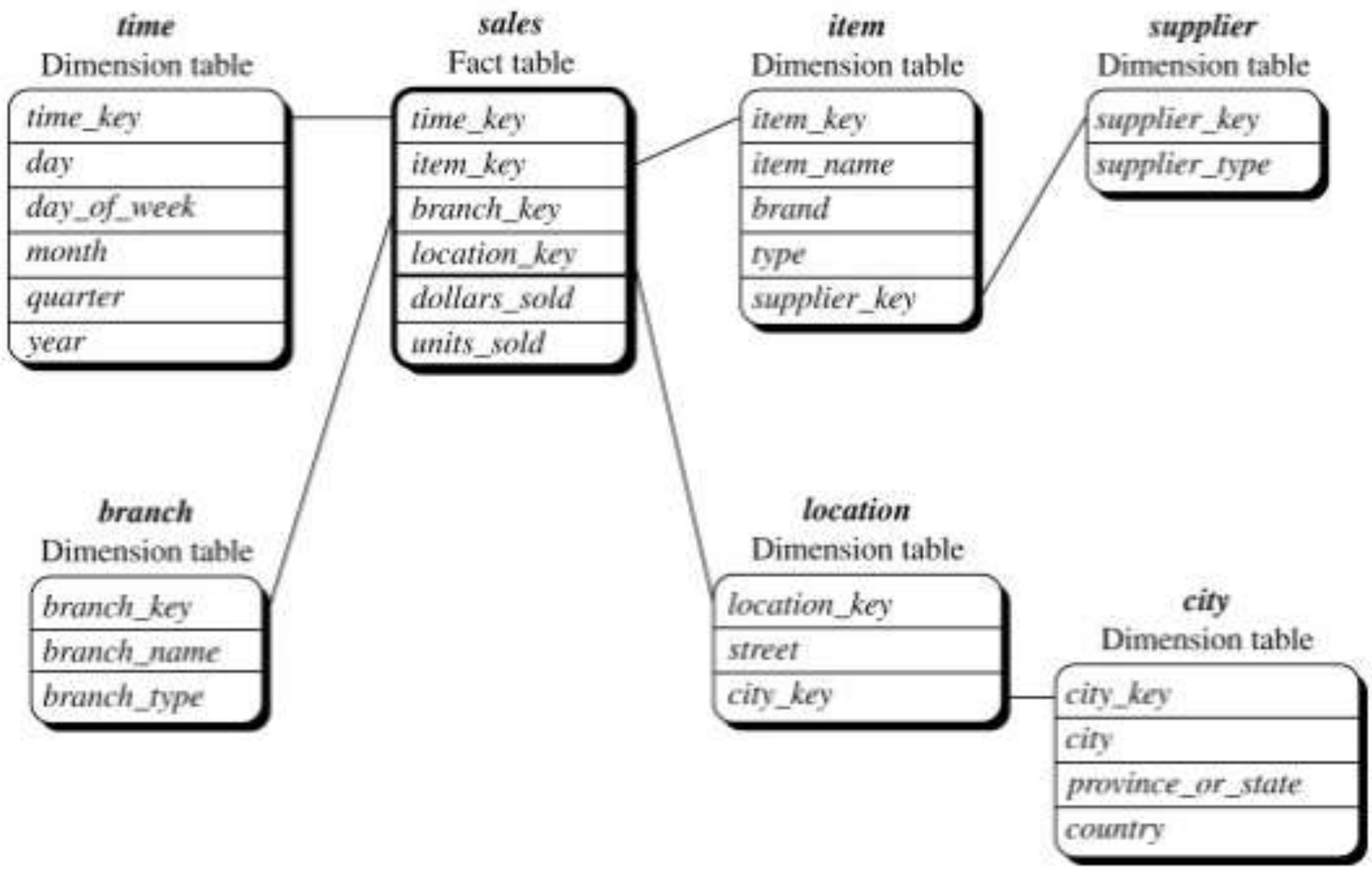
Characteristics of Star Schema

- The dimension table is joined to the fact table using a foreign key
- The dimension table are not joined to each other
- It has one-to-many relationship between each dimension table and the fact table
- Fact table would contain key and measure
- The Star schema is easy to understand.
- The dimension tables are not normalized.
- The schema is widely supported by BI Tools

Snowflake Schema

- A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions.
- It is called snowflake because its diagram resembles a Snowflake(ندفة الثلج).
- The dimension tables are normalized which splits data into additional tables.

SNOWFLAKE SCHEMA (CONT.)



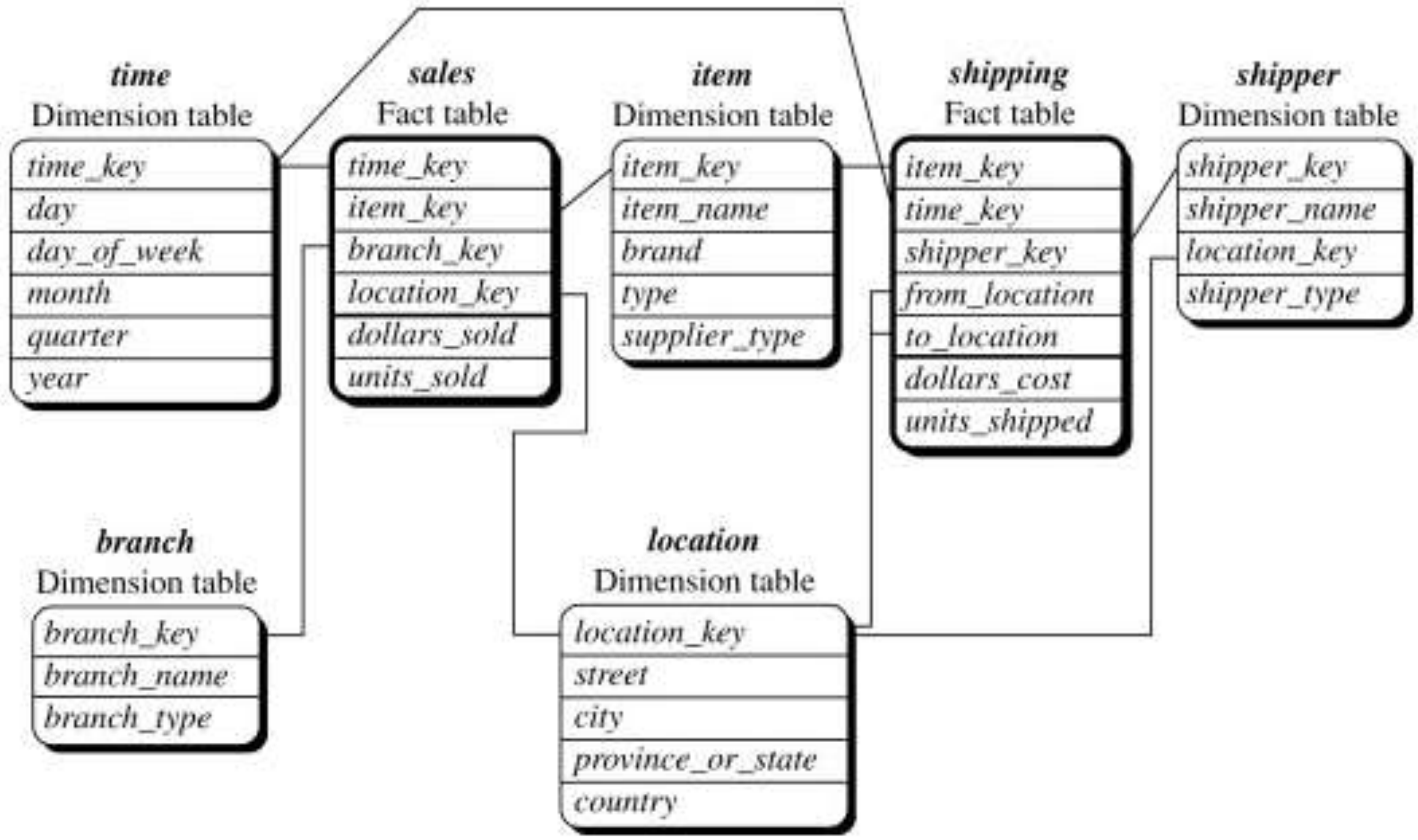
Snowflake vs. Star Schemas

	Snowflake Schema	Star Schema
DimTable Normalization:	3 Normal Form	2 Normal Denormalized Form
Joins:	Higher number of Joins	Fewer Joins
Ease of Use:	More complex queries and hence less easy to understand	Less complex queries and easy to understand
Query Performance:	More foreign keys-and hence more query execution time	Less no. of foreign keys and hence lesser query execution time
Ease of maintenance/change:	No redundancy and hence more easy to maintain and change	Has redundant data and hence less easy to maintain/change
Type of Datawarehouse:	Good to use for small datawarehouses/datamarts	Good for large datawarehouses
Dimension table:	It may have more than one dimension table for each dimension	Contains only single dimension table for each dimension

Galaxy schema

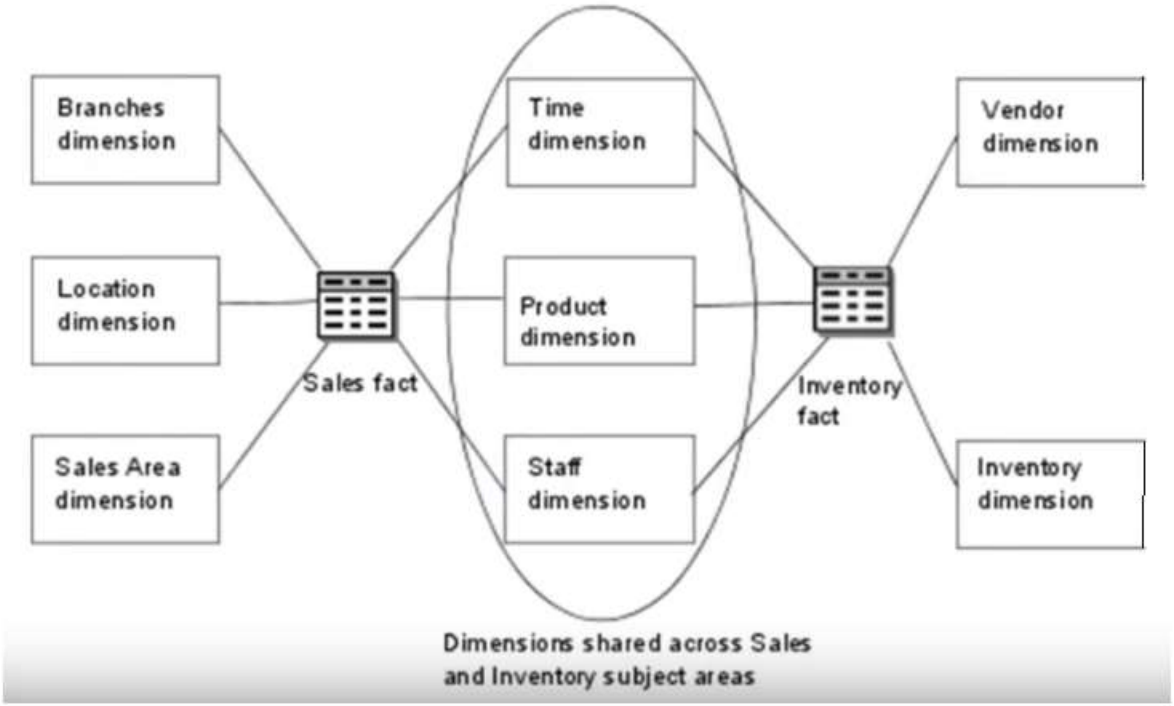
- A galaxy schema is also known as Fact Constellation Schema. In this schema, multiple fact tables share the same dimension tables. The arrangement of fact tables and dimension tables looks like a collection of stars in the Galaxy schema model.
- The schema is viewed as a collection of stars hence the name Galaxy Schema.
- The shared dimensions in this model are known as Conformed dimensions.
- This type of schema is used for sophisticated requirements and for aggregated fact tables that are more complex to be supported by the Star schema (or) SnowFlake schema.

Galaxy Schema (cont.)



Conformed Dimensions

A conformed dimension is a set of data attributes that have been physically referenced in multiple database tables using the same key value to refer to the same structure, attributes, domain values, definitions and concepts.

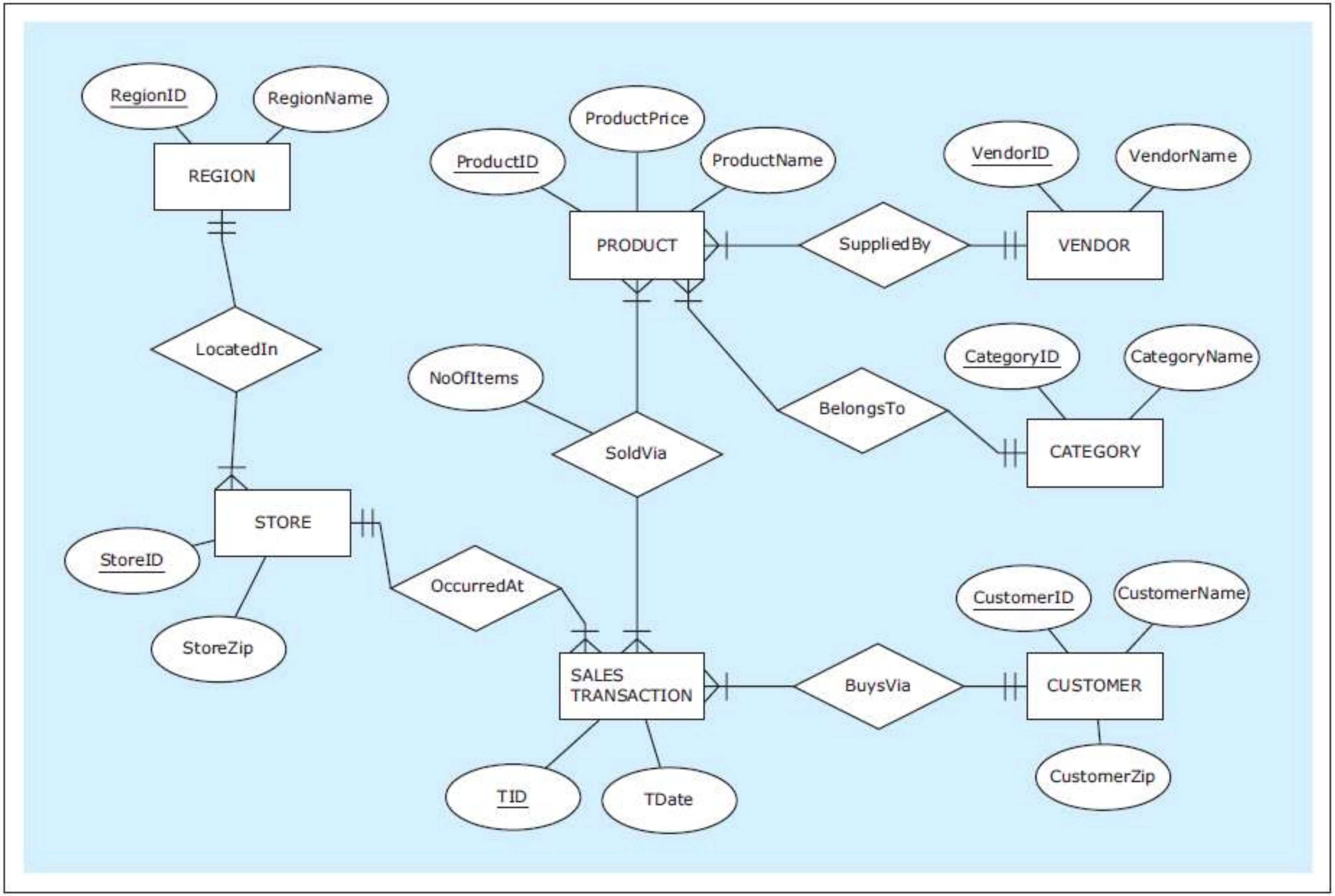


Conformed Dimensions (Cont.)

- Common dimensions and attributes should be standardized across data marts
- Create master copy of each common dimension table
- Three types of "conformed" dimensions:
  - Dimension table identical to master copy
  - Dimension table has subset of rows from the master copy
    - Can improve performance when many dimension rows are not relevant to a particular process
  - Dimension table has subset of attributes from master copy
    - Allows for roll-up dimensions at different grains

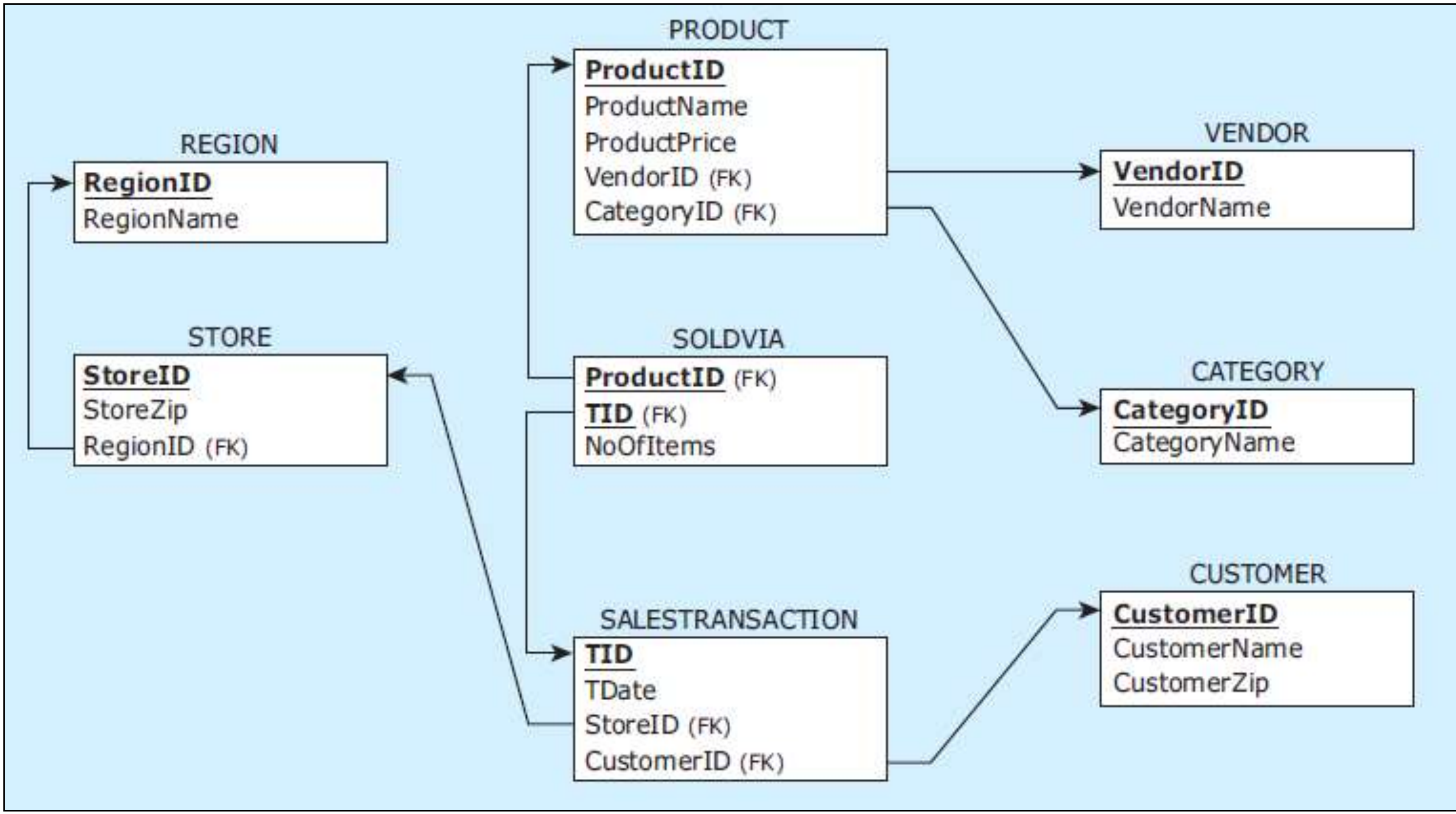
EXAMPLE

ER Diagram - Retail Company Sales Department Database



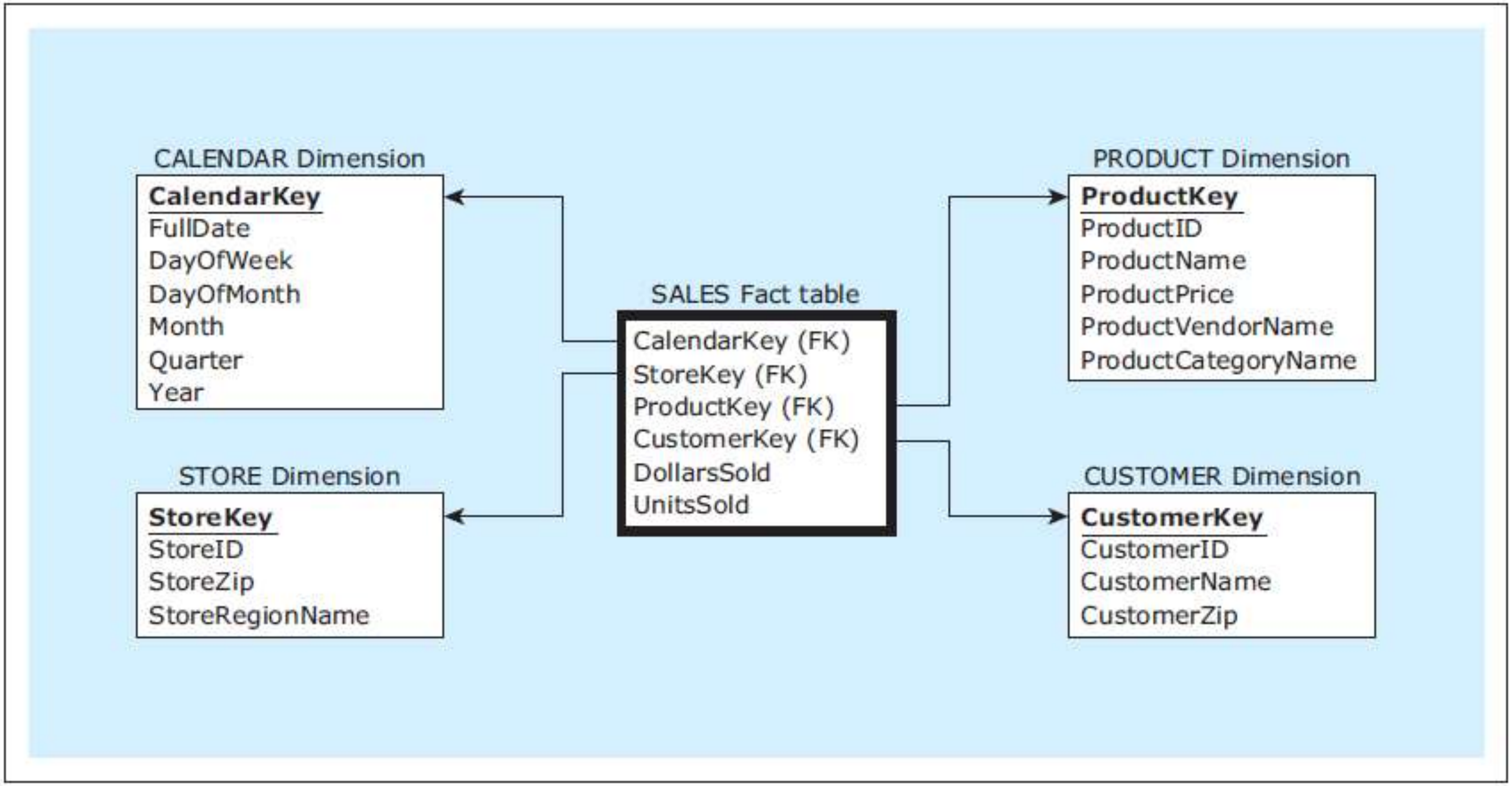
EXAMPLE (CONT.)

Relational Schema - Retail Company Sales Department Database



EXAMPLE (CONT.)

Star Model - Retail Company dimensional model for the subject sales



EXAMPLE (CONT.)

Star Model - Retail Company dimensional model for the subject sales, populated with the data from the operational data source

CALENDAR Dimension

CalendarKey	FullDate	DayOf Week	DayOf Month	Month	Qtr	Year
1	1/1/2013	Tuesday	1	January	Q1	2013
2	1/2/2013	Wednesday	2	January	Q1	2013

PRODUCT Dimension

ProductKey	ProductID	Product Name	Product Price	Product Vendor Name	Product Category Name
1	1X1	Zzz Bag	$100	Pacifica Gear	Camping
2	2X2	Easy Boot	$70	Mountain King	Footwear
3	3X3	Cosy Sock	$15	Mountain King	Footwear
4	4X4	Dura Boot	$90	Pacifica Gear	Footwear
5	5X5	Tiny Tent	$150	Mountain King	Camping
6	6X6	Biggy Tent	$250	Mountain King	Camping

STORE Dimension

StoreKey	StoreID	StoreZip	StoreRegionName
1	S1	60600	Chicagoland
2	S2	60605	Chicagoland
3	S3	35400	Tristate

CUSTOMER Dimension

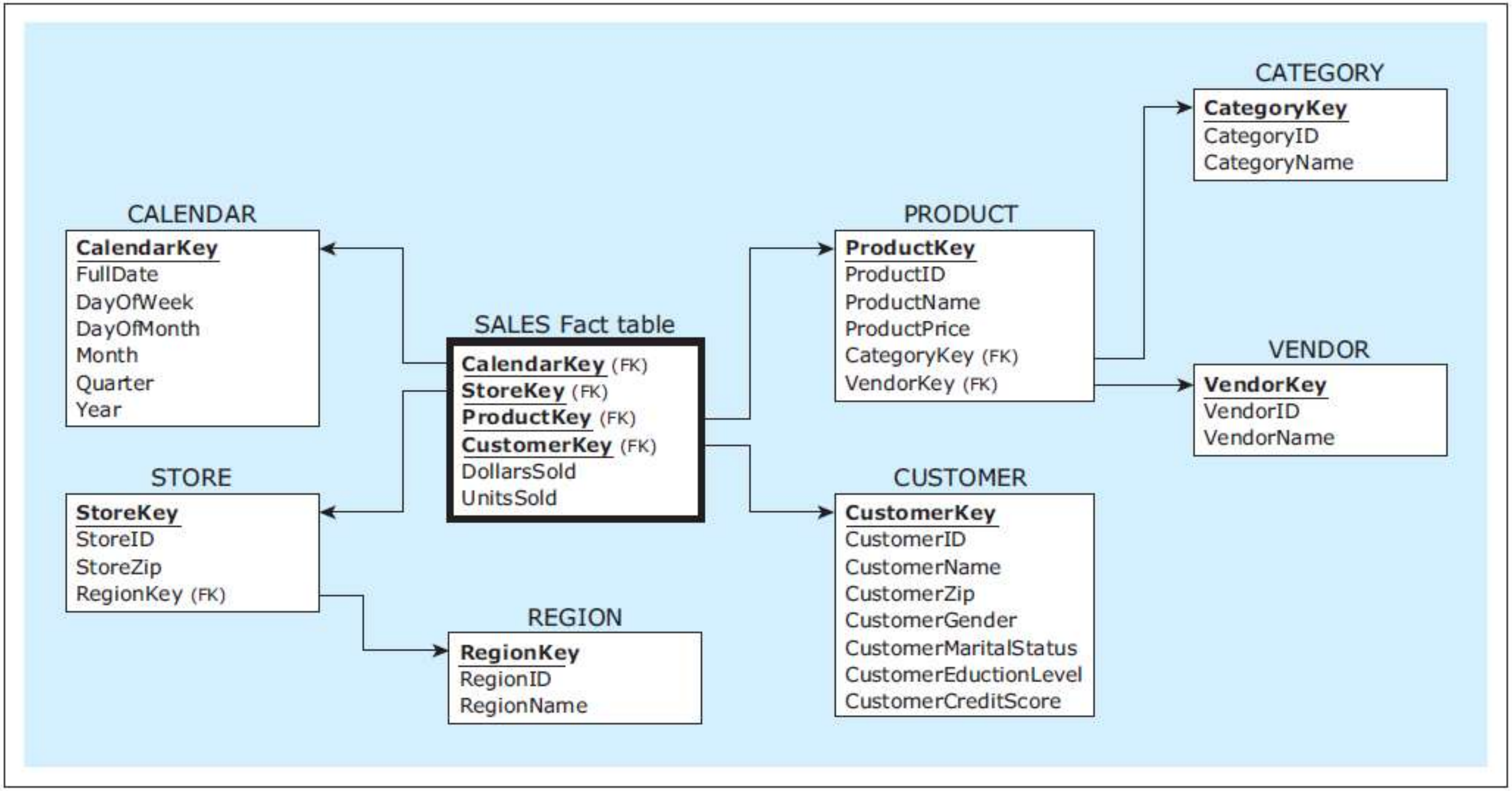
CustomerKey	CustomerID	CustomerName	CustomerZip
1	1-2-333	Tina	60137
2	2-3-444	Tony	60611
3	3-4-555	Pam	35401

SALES Fact table

CalendarKey	StoreKey	ProductKey	CustomerKey	DollarsSold	UnitsSold
1	1	1	1	$100	1
1	2	2	2	$70	1
2	3	3	1	$75	5
2	3	1	1	$100	1
2	3	4	3	$90	1
2	3	2	3	$140	2
2	3	4	2	$360	4
2	3	5	2	$300	2
2	3	6	2	$250	1

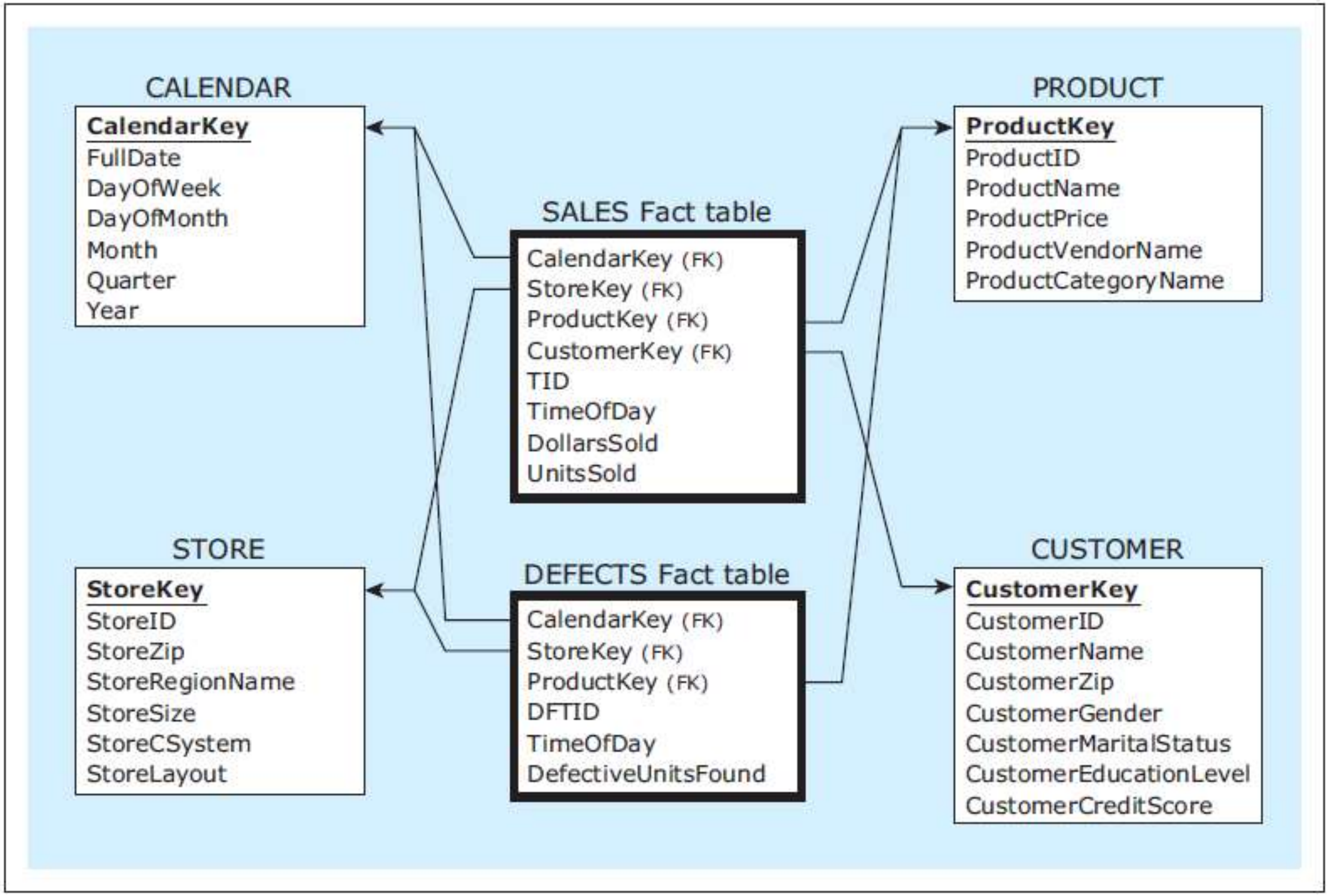
EXAMPLE (CONT.)

Snowflake Model - Retail Company dimensional model for the subject sales



EXAMPLE (CONT.)

Constellation Model - Retail Company dimensional model for the subjects sales and defects



EXAMPLE (CONT.)

Constellation Model : Retail Company dimensional model for the subjects sales and defects , populated with the data from the four sources

CALENDAR Dimension

CalendarKey	FullDate	DayOf Week	DayOf Month	Month	Qtr	Year
1	1/1/2013	Tuesday	1	January	Q1	2013
2	1/2/2013	Wednesday	2	January	Q1	2013

PRODUCT Dimension

ProductKey	ProductID	Product Name	Product Price	Product Vendor Name	Product Category Name
1	1X1	Zzz Bag	$100	Pacifica Gear	Camping
2	2X2	Easy Boot	$70	Mountain King	Footwear
3	3X3	Cosy Sock	$15	Mountain King	Footwear
4	4X4	Dura Boot	$90	Pacifica Gear	Footwear
5	5X5	Tiny Tent	$150	Mountain King	Camping
6	6X6	Biggy Tent	$250	Mountain King	Camping

STORE Dimension

StoreKey	StoreID	StoreZip	StoreRegion Name	Store Size (m^2)	Store CSystem	Store Layout
1	S1	60600	Chicagoland	51000	Cashiers	Modern
2	S2	60605	Chicagoland	35000	Self Service	Traditional
3	S3	35400	Tristate	55000	Mixed	Traditional

CUSTOMER Dimension

CustomerKey	CustomerID	Customer Name	Customer Zip	Customer Gender	Customer MaritalStatus	Customer EducationLevel	Customer CreditScore
1	1-2-333	Tina	60137	Female	Single	College	700
2	2-3-444	Tony	60611	Male	Single	High School	650
3	3-4-555	Pam	35401	Female	Married	College	623

SALES Fact table

CalendarKey	StoreKey	ProductKey	CustomerKey	TID	TimeOfDay	DollarsSold	UnitsSold
1	1	1	1	T111	8:23:59 AM	$100	1
1	2	2	2	T222	8:24:30 AM	$70	1
2	3	3	1	T333	8:15:08 AM	$75	5
2	3	1	1	T333	8:15:08 AM	$100	1
2	3	4	3	T444	8:20:33 AM	$90	1
2	3	2	3	T444	8:20:33 AM	$140	2
2	3	4	2	T555	8:30:00 AM	$360	4
2	3	5	2	T555	8:30:00 AM	$300	2
2	3	6	2	T555	8:30:00 AM	$250	1

DEFECTS Fact table

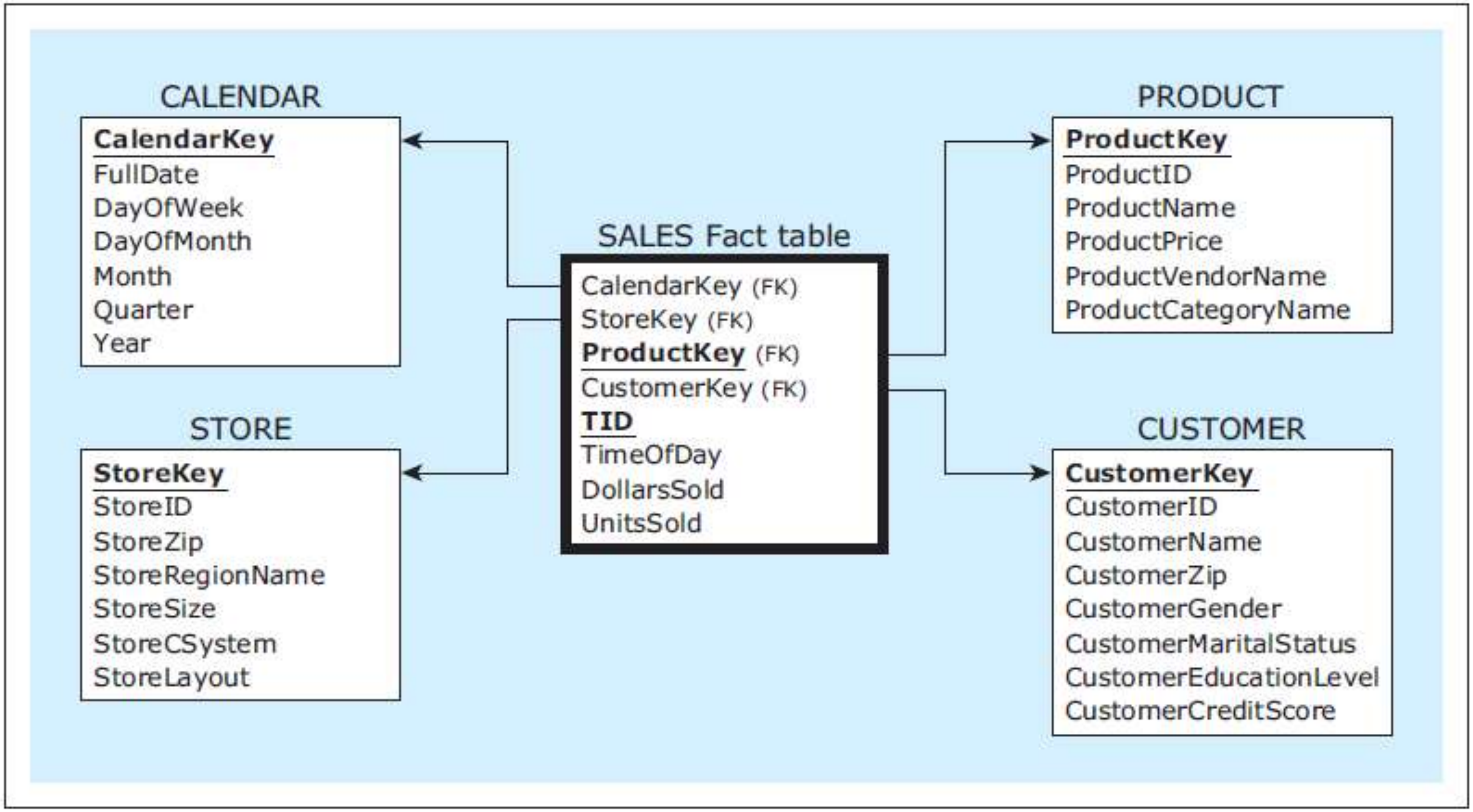
CalendarKey	StoreKey	ProductKey	DFTID	TimeOfDay	DefectiveUnitsFound
1	1	1	DFT101	8:00:00 AM	1
1	2	2	DFT202	8:30:00 AM	2
2	3	3	DFT303	8:45:00 AM	6

DETAILED VERSUS AGGREGATED FACT TABLES

- Fact tables in a dimensional model can contain either detailed data or aggregated data
- In **detailed fact tables** each record refers to a single fact
- In **aggregated fact tables** each record summarizes multiple facts

Detailed Fact Table Example

Retail Company dimensional model for the subject *sales*



DETAILED FACT TABLE EXAMPLE (CONT.)

CALENDAR Dimension

CalendarKey	FullDate	DayOf Week	DayOf Month	Month	Qtr	Year
1	1/1/2013	Tuesday	1	January	Q1	2013
2	1/2/2013	Wednesday	2	January	Q1	2013

PRODUCT Dimension

ProductKey	ProductID	Product Name	Product Price	Product Vendor Name	Product Category Name
1	1X1	Zzz Bag	$100	Pacifica Gear	Camping
2	2X2	Easy Boot	$70	Mountain King	Footwear
3	3X3	Cosy Sock	$15	Mountain King	Footwear
4	4X4	Dura Boot	$90	Pacifica Gear	Footwear
5	5X5	Tiny Tent	$150	Mountain King	Camping
6	6X6	Biggy Tent	$250	Mountain King	Camping

STORE Dimension

StoreKey	StoreID	StoreZip	StoreRegion Name	Store Size (m^2)	Store CSystem	Store Layout
1	S1	60600	Chicagoland	51000	Cashiers	Modern
2	S2	60605	Chicagoland	35000	Self Service	Traditional
3	S3	35400	Tristate	55000	Mixed	Traditional

CUSTOMER Dimension

CustomerKey	CustomerID	Customer Name	Customer Zip	Customer Gender	Customer MaritalStatus	Customer EducationLevel	Customer CreditScore
1	1-2-333	Tina	60137	Female	Single	College	700
2	2-3-444	Tony	60611	Male	Single	High School	650
3	3-4-555	Pam	35401	Female	Married	College	623

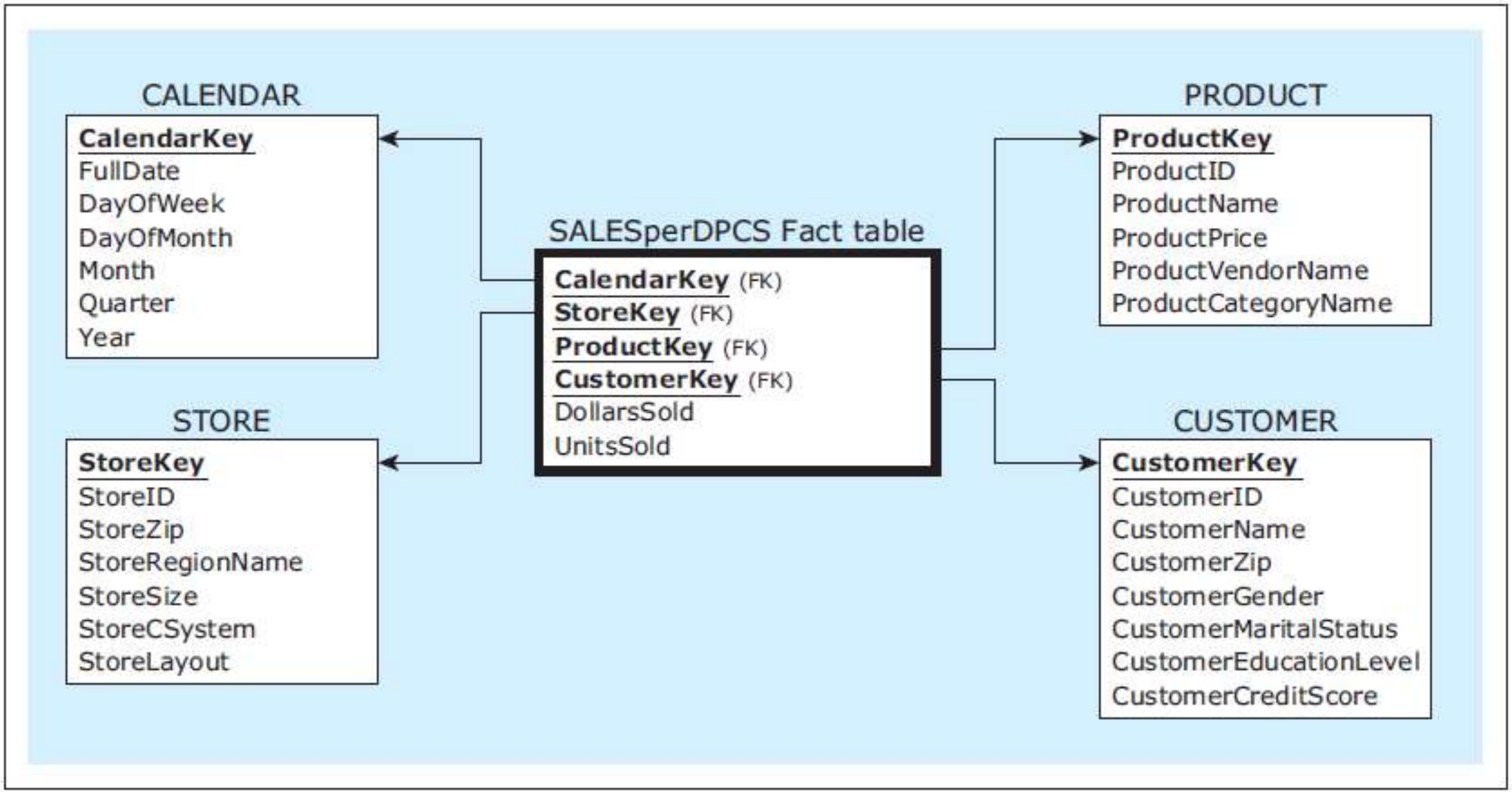
SALES Fact table

CalendarKey	StoreKey	ProductKey	CustomerKey	TID	TimeOfDay	DollarsSold	UnitsSold
1	1	1	1	T111	8:23:59 AM	$100	1
1	2	2	2	T222	8:24:30 AM	$70	1
2	3	3	1	T333	8:15:08 AM	$75	5
2	3	1	1	T333	8:15:08 AM	$100	1
2	3	4	3	T444	8:20:33 AM	$90	1
2	3	2	3	T444	8:20:33 AM	$140	2
2	3	4	2	T555	8:30:00 AM	$360	4
2	3	5	2	T555	8:30:00 AM	$300	2
2	3	6	2	T555	8:30:00 AM	$250	1
2	3	5	2	T666	9:30:00 AM	$300	2
2	3	6	2	T666	9:30:00 AM	$250	1

Retail Company dimensional model for the subject ***sales***, populated with the additional data records

Aggregated Fact Table Example

Retail Company dimensional model with an aggregated fact table Sales per *day*, *product*, *customer*, and *store*



AGGREGATED FACT TABLE EXAMPLE (CONT.)

Retail Company dimensional model for the subject sales with an aggregated fact table

Sales per
 day,
 product,
 customer,
 store,
populated with the data

CALENDAR Dimension

CalendarKey	FullDate	DayOf Week	DayOf Month	Month	Qtr	Year
1	1/1/2013	Tuesday	1	January	Q1	2013
2	1/2/2013	Wednesday	2	January	Q1	2013

PRODUCT Dimension

ProductKey	ProductID	Product Name	Product Price	Product Vendor Name	Product Category Name
1	1X1	Zzz Bag	$100	Pacifica Gear	Camping
2	2X2	Easy Boot	$70	Mountain King	Footwear
3	3X3	Cosy Sock	$15	Mountain King	Footwear
4	4X4	Dura Boot	$90	Pacifica Gear	Footwear
5	5X5	Tiny Tent	$150	Mountain King	Camping
6	6X6	Biggy Tent	$250	Mountain King	Camping

STORE Dimension

StoreKey	StoreID	StoreZip	StoreRegion Name	Store Size (m^2)	Store CSystem	Store Layout
1	S1	60600	Chicagoland	51000	Cashiers	Modern
2	S2	60605	Chicagoland	35000	Self Service	Traditional
3	S3	35400	Tristate	55000	Mixed	Traditional

CUSTOMER Dimension

CustomerKey	CustomerID	Customer Name	Customer Zip	Customer Gender	Customer MaritalStatus	Customer EducationLevel	Customer CreditScore
1	1-2-333	Tina	60137	Female	Single	College	700
2	2-3-444	Tony	60611	Male	Single	High School	650
3	3-4-555	Pam	35401	Female	Married	College	623

SALESPerDPCS Fact table

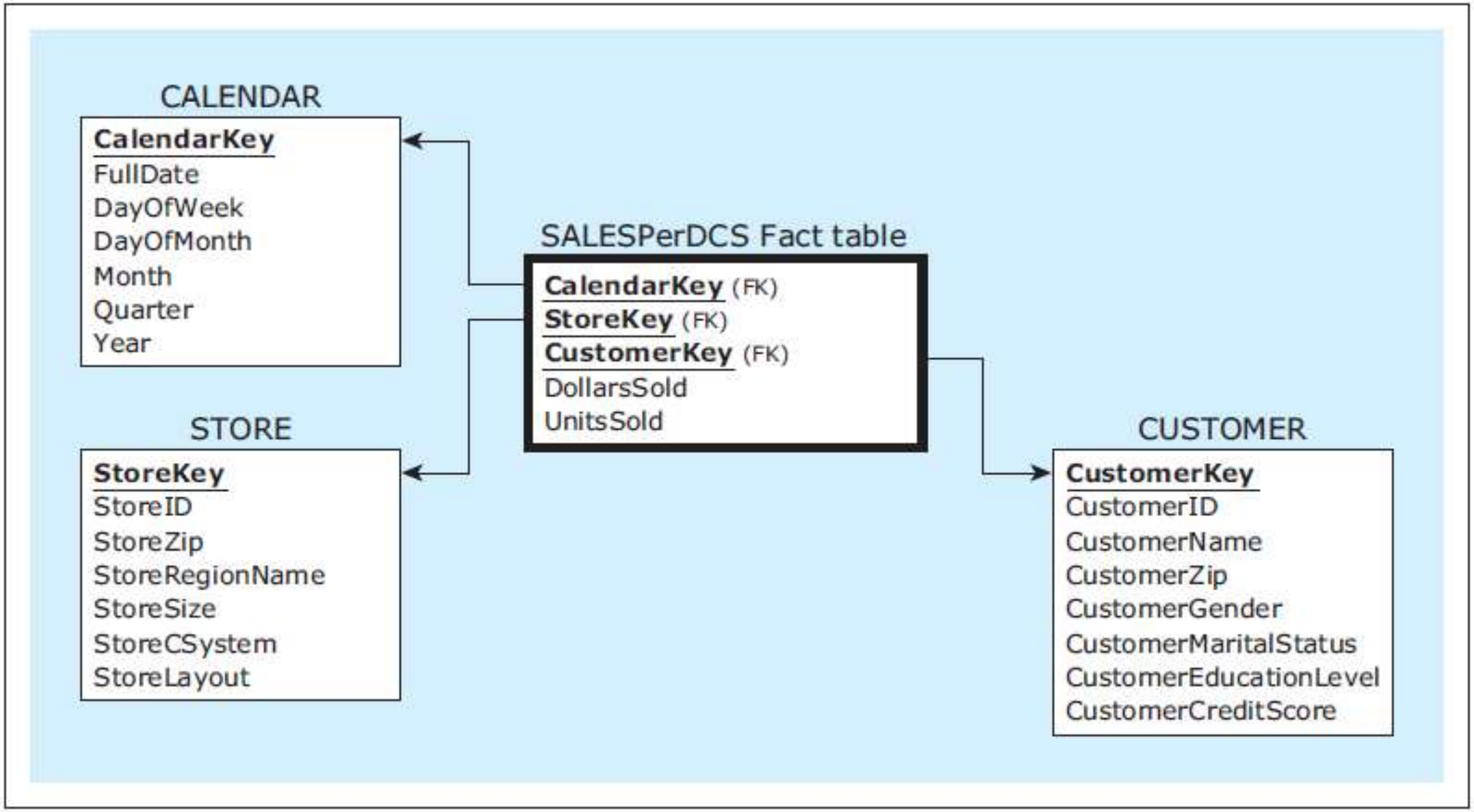
CalendarKey	StoreKey	ProductKey	CustomerKey	DollarsSold	UnitsSold
1	1	1	1	$100	1
1	2	2	2	$70	1
2	3	3	1	$75	5
2	3	1	1	$100	1
2	3	4	3	$90	1
2	3	2	3	$140	2
2	3	4	2	$360	4
2	3	5	2	$600	4
2	3	6	2	$500	2

Amounts from 8th and 10th records in SALES fact table in Figure 8.23 combined (added)

Amounts from 9th and 11th records in SALES fact table in Figure 8.23 combined (added)

AGGREGATED FACT TABLE EXAMPLE (CONT.)

Retail Company star schema with an aggregated fact table Sales per *day*, *customer*, and *store*



Aggregated Fact Table Example (cont.)

Retail Company dimensional model for the subject sales with an aggregated fact table

Sales per *day*, *customer*, *store*, populated with the data

CALENDAR Dimension

CalendarKey	FullDate	DayOf Week	DayOf Month	Month	Qtr	Year
1	1/1/2013	Tuesday	1	January	Q1	2013
2	1/2/2013	Wednesday	2	January	Q1	2013

STORE Dimension

StoreKey	StoreID	StoreZip	StoreRegion Name	Store Size (m^2)	Store CSystem	Store Layout
1	S1	60600	Chicagoland	51000	Cashiers	Modern
2	S2	60605	Chicagoland	35000	Self Service	Traditional
3	S3	35400	Tristate	55000	Mixed	Traditional

CUSTOMER Dimension

CustomerKey	CustomerID	Customer Name	Customer Zip	Customer Gender	Customer MaritalStatus	Customer EducationLevel	Customer CreditScore
1	1-2-333	Tina	60137	Female	Single	College	700
2	2-3-444	Tony	60611	Male	Single	High School	650
3	3-4-555	Pam	35401	Female	Married	College	623

SALESPerDCS Fact table

CalendarKey	StoreKey	CustomerKey	DollarsSold	UnitsSold
1	1	1	$100	1
1	2	2	$70	1
2	3	1	$175	6
2	3	3	$230	3
2	3	2	$1,460	10

Amounts from 3rd and 4th records in SALES fact table in Figure 8.23 combined (added)

Amounts from 5th and 6th records in SALES fact table in Figure 8.23 combined (added)

Amounts from 7th through 11th records in SALES fact table in Figure 8.23 combined (added)

Granularity of the fact tables

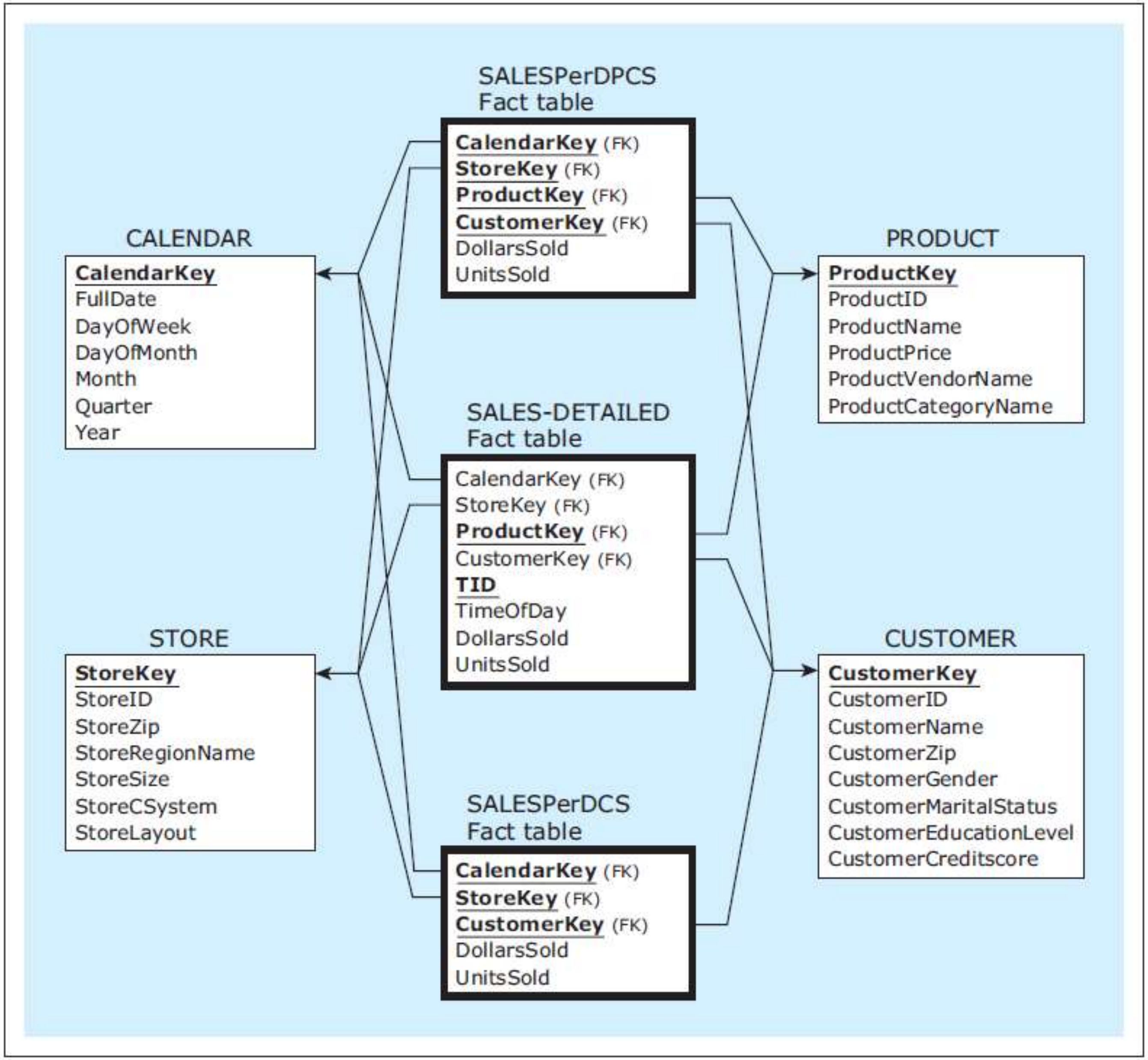
- Granularity describes what is depicted by one row in the fact table
- Detailed fact tables have *fine level* of granularity because each record represents a single fact
- Aggregated fact tables have a *coarser level* of granularity than detailed fact tables as records in aggregated fact tables always represent summarizations of multiple facts

Granularity of the fact tables

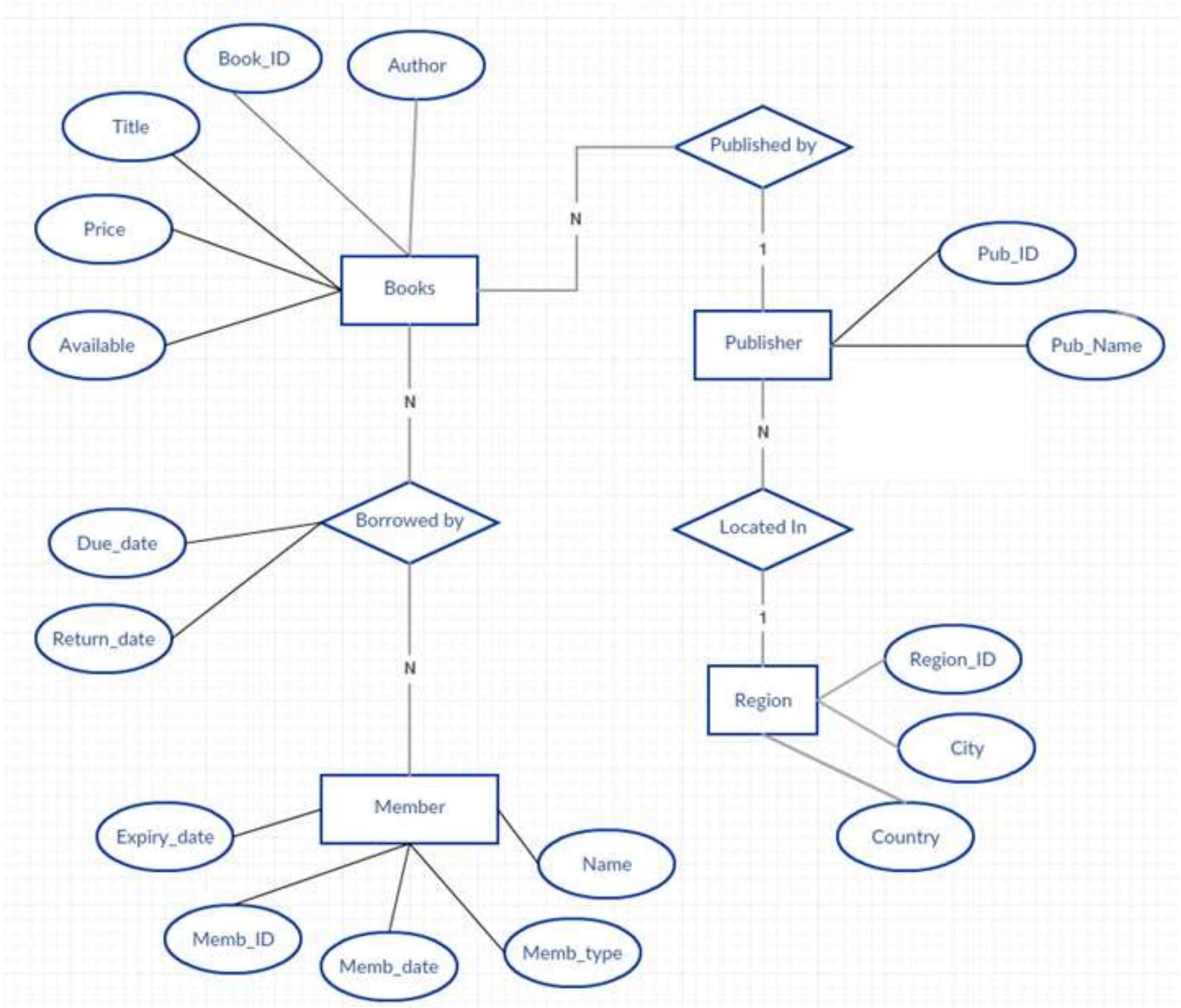
- Due to their compactness, coarser granularity aggregated fact tables are quicker to query than detailed fact tables
- Coarser granularity tables are limited in terms of what information can be retrieved from them
- One way to take advantage of the query performance improvement provided by aggregated fact tables, while retaining the power of analysis of detailed fact tables, is to have both types of tables coexisting within the same dimensional model, i.e. in the same constellation

Example

A constellation of detailed and aggregated facts - Example



Exercise



Book_ID
Author
Title
Price
Available
Books
Published by
N
1
Pub_ID
Publisher
Pub_Name
N
N
Borrowed by
Due_date
Return_date
N
Located In
1
Region_ID
Region
City
Country
Member
Expiry_date
Name
Memb_ID
Memb_date
Memb_type

Exercise (cont.)

- Use the below ER diagram to answer the following questions:
  A. Design a logical relational database schema for transactional database (OLTP)
  B. Design a logical relational database schema for data warehouse database (OLAP) using "*Star Schema*".
  C. Design a logical relational database schema for data warehouse database (OLAP) using "*Snowflake Schema*".

The End