CSC12107 – Information Systems for Business Intelligence

Chapter 3
DATA MODELING

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Learning Objectives

- After complete this chapter, student can:
  - Define the meaning of the facts and dimensional attributes
  - Define the data hierarchies , slowly changing dimension
  - Design a Star or Snowflake data model diagram from analytical business requirements and OLTP system

Main topic

- Designing the Dimensional Data Store
  - Star schema
  - Snowflake schema
- Designing the Normalize Data Store

Introduction

Building a data warehouse system to make sure that the system will help users achieve the business objectives:

1. gather functional and nonfunctional requirements
2. find out business issues and challenges
3. identify the business areas that would benefit from a data warehouse system
4. dive into each of these areas and learn about the business operations within that area
5. define what the data warehouse system does
6. find out whether the data that we need is available in the source systems and whether it is accessible

Understanding Business Operations

- understand the business operations, we will define the *functional requirements* ➔ define what the data warehouse system does:
  - the questions or issues that a data warehouse system will be able to answer
  - the data that will be stored in the data warehouse
  - the analysis the users will be performing

- *Nonfunctional requirements* guide and constrain the architecture
  - security, availability, and performance…

Understanding Business Operations

- An **event** is an activity that happens repeatedly every few seconds or minutes. Or it could be every few hours or days.
  - Ex:
    - in the purchasing area, we have a document called a purchase order.
    - a customer buys a record from the online store
- A **level** is a quantitative measurement of an object at a particular point in time
- **Roles** are the who, whom, and what involved in the event.

Understanding Business Operations

Amadeus Entertainment Data Warehouse System Functional Requirements

- **EX1:** The business users need to be able to analyze "product sales" (that's when a High customer is buying a product rather than subscribing to a package) over time by geographic area, by customer demographic, by stores and sales territory, and by product hierarchy. The users also need to know the revenues, the costs, and the margin in U.S. dollars. Local currencies will be converted to U.S. dollars.

- **EX2:** The business users will be able to analyze "supplier performance," which is the Low weighted average of the totals spent, costs, value of returns, value of rejects, title and format availability, stock outages, lead time, and promptness

Dimensional modeling (DM)

- DM is a logical design technique. The purpose of DM is to enable BI reporting, query, and analysis

- The key concepts in DM are **facts**, **dimensions**, and **attributes**

- Facts, dimensions, and attributes can be organized in several ways, called schemas:
  - Star schema
  - Snowflake schema
  - Multidimensional/galaxy schema

- 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
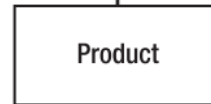
A DIMENSIONAL MODEL

when those
products were sold

Date

Product Sales

Customer

who bought
the products

sales_date_key
customer_key
product_key
store_key
- - - - - - - - - -
order_id
line_number
quantity
unit_price
unit_cost
sales_value
sales_cost
margin

what products
were sold

Product

Store

Where those products
were bought

- **Fact table**: tbl_Fact_Sales
- **Measure**: Quantity, sales_value, sales_cost, unit_price, unit_cost, margin
- **Fact table keys**: sales_date_key, customer_key, product_key, store_key
- **Dimension**: Date, Customer, Product, Store

Dimension

- 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

- Ex: product, geography, customers, employees, time….

- Key purpose of dimensions:
  - use their attributes to filter and analyze data based on performance measures

Dimension keys

DimProduct

ProductKey
ProductAlternateKey
WeightUnitMeasureCode
SizeUnitMeasureCode
EnglishProductName
StandardCost
FinishedGoodsFlag
Color
SafetyStockLevel
ReorderPoint
ListPrice
Size
SizeRange

DimGeography

GeographyKey
PostalCode
City
StateProvinceCode
StateProvinceName
CountryRegionCode

- A key concept in constructing dimensions is that each row of a dimension table is unique

- **Surrogate keys**

  - Is often generated by the database system

  - is an integer whose value is meaningless

  - to provide an identifier that is consistent and unique across source systems and time, and independent of business systems

  - is a great data type to index and join in a relational model

Dimension keys

- Best practice:
  - maintain the source system's primary key as an alternate key in the dimension (also called source system's natural key)
- Ex: Dim_Customers
  - CustomerNK is the natural key in the dimension and
  primary key in the source system
  - SOR_NK:

Dim_Customers	
PK	**CustomerSK**
	SOR_NK
U1	**CustomerNK**
	Title
	FirstName
	MiddleName
	LastName

Dimension hierachy

- Group things together in ways that an enterprise would measure itself

- These hierarchies represent many-to-one relationships

- EX:
  - DimProduct: Product → Product subcategory → Product category
  - DimeDate: Day → month → quarter → Year
 - DimGeography: City → State → Country

Dimension hierachy

Users drill/roll up and down dimensional hierarchies to obtain more or less detail about the business

Dim_Time
- Year
- Quater
- Month
- Day

Dim_Store
- All
- Region
- Country
- City

Dim_Product
- Product Category
- Product SubCategory
- Product

Year	Region	Sales
1996	Asia	1,000
	Europe	50,000
	America	20,000
1997	Asia	1,500
	.....	

Dimension hierachy

Users drill/roll up and down dimensional hierarchies to obtain more or less detail about the business

Dim_Time

Year

Quater

Month

Day

Cửa hàng

All

Region

Country

City

**Sản phẩm**

Product Category

Product SubCategory

Product
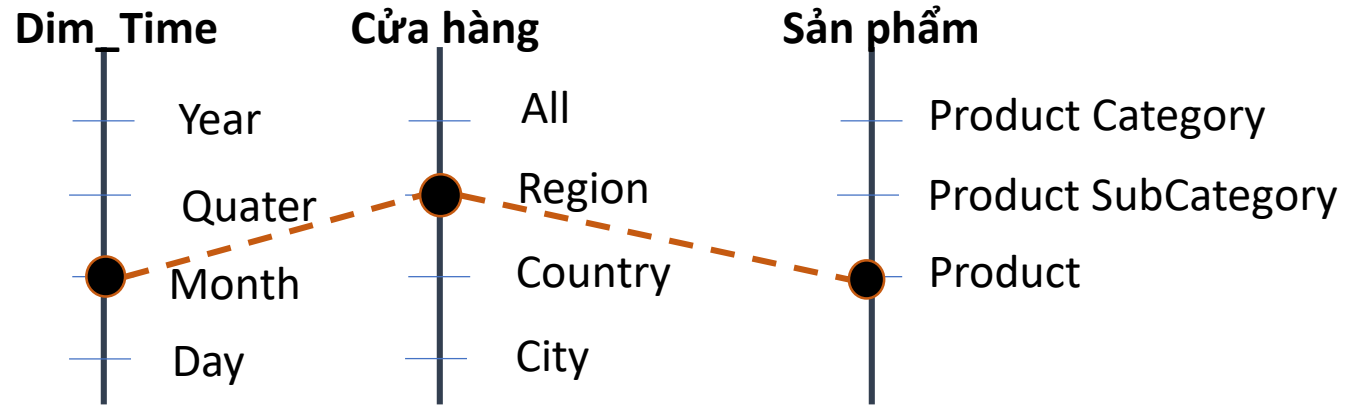
| Year	Quarter	Vùng miền	Doanh thu
1996	Q1	Asia	200
	Q2	Asia	200
	Q3	Asia	250
1997	Q4	Asia	350
	Q1	Europe	10,000
-----	------	------	------

Date dimension

- Important

- Several levels in the dimension table → recorded all the time levels needed for analysis and in accordance with the content

- Example:
  - Analysis financial data of an enterprise: day, week, month, quarter, year, holiday…

Modeling the calendar

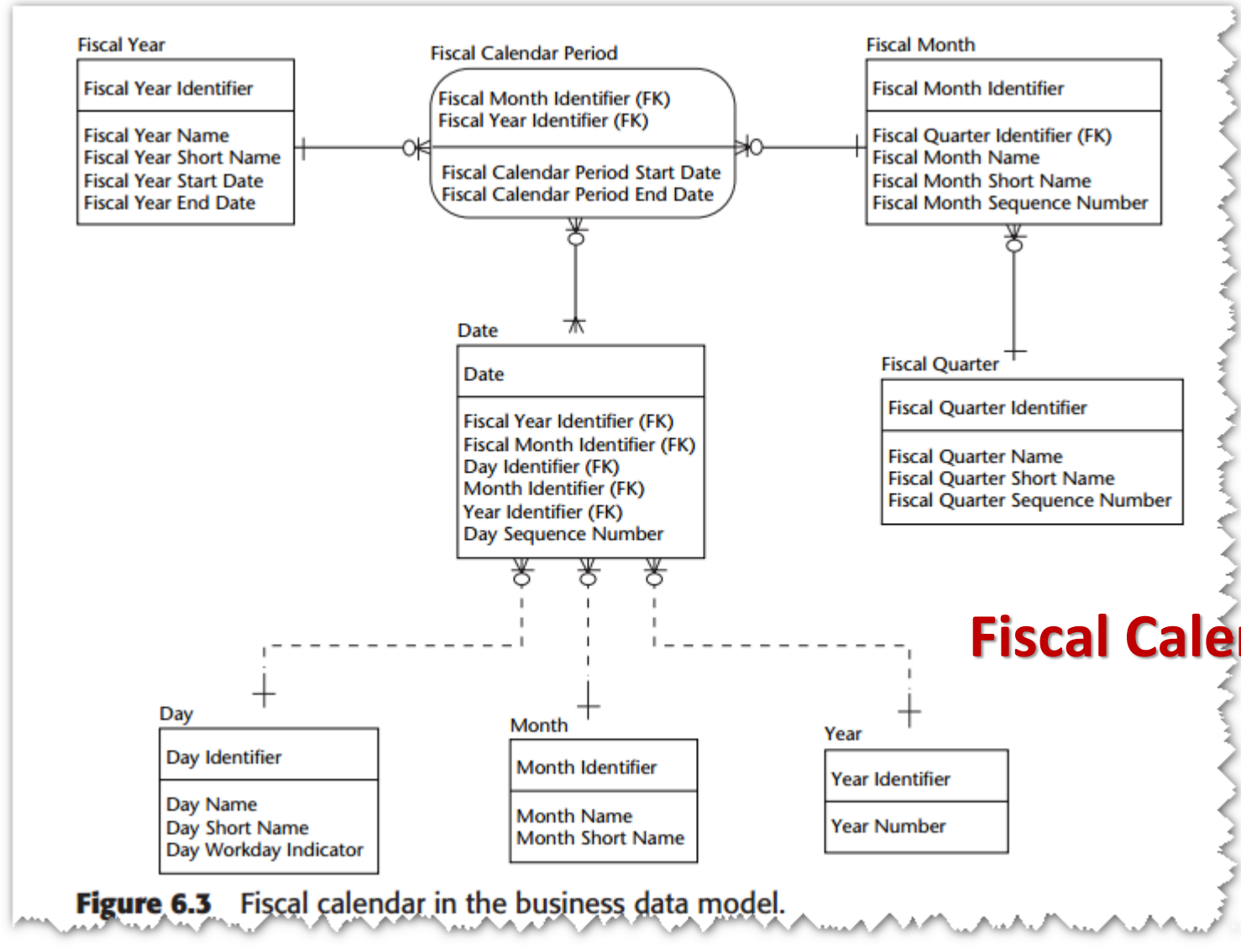**Figure 6.1** Calendar in the business model.
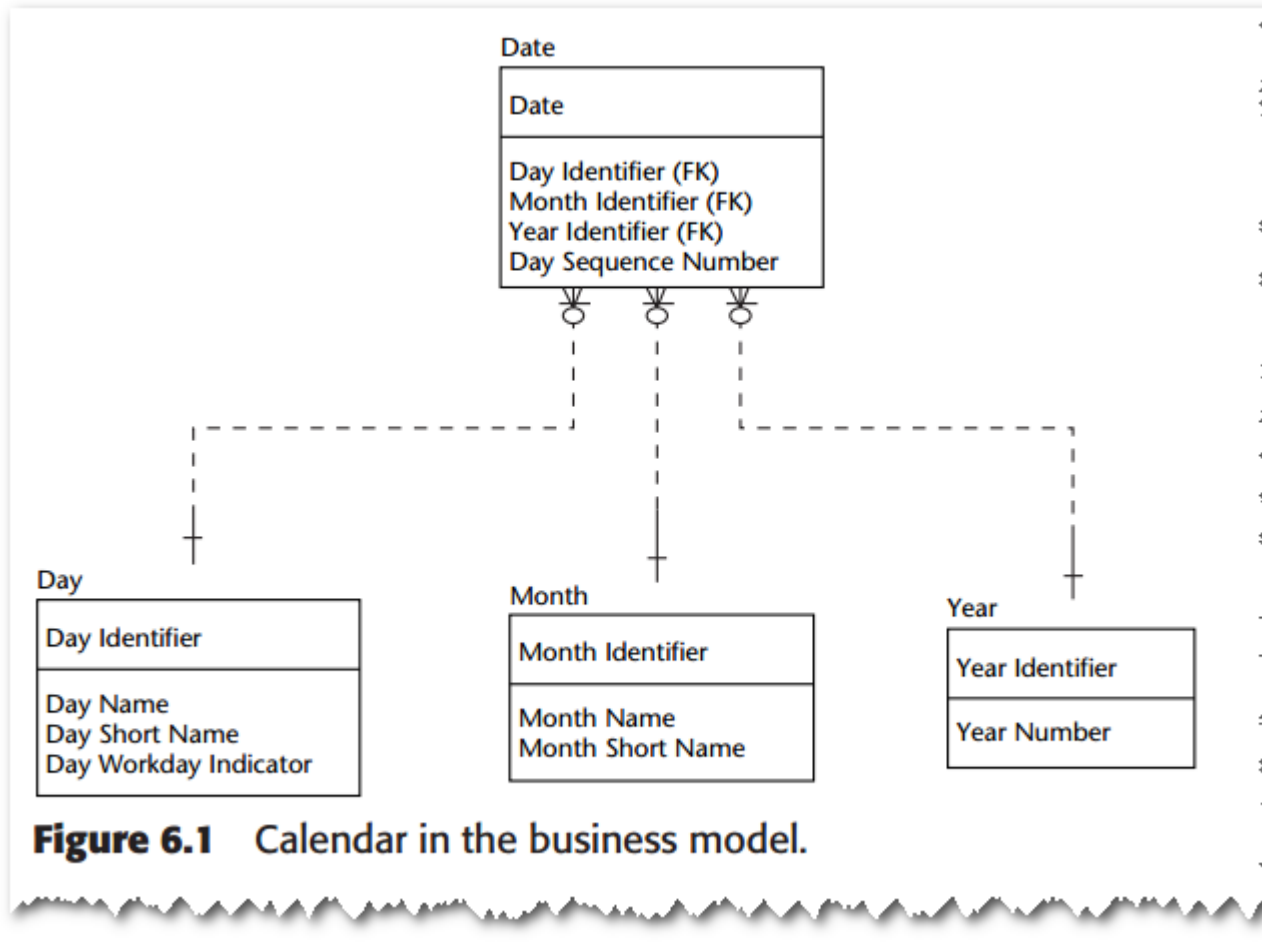
Gregorian calendar

Modeling the calendar

Figure 6.2 Fiscal calendar.
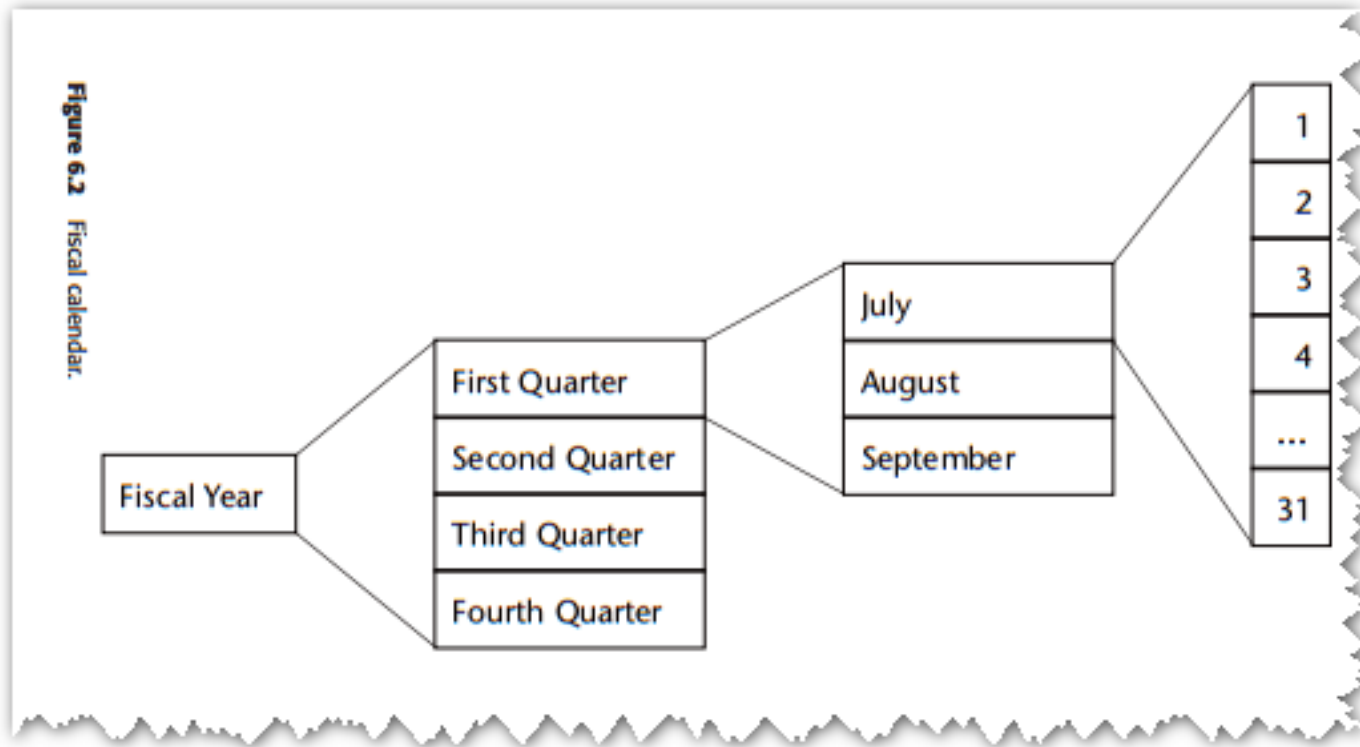
Fiscal Calendar

Modeling the calendar

Figure 6.3 Fiscal calendar in the business data model.

Fiscal Calendar

Slowly Changing Dimension (SCD)

- SCD: is a technique used to store the **historical value** of dimension attributes

- Three types of SCD:
  - overwrite the old values with the new ones (SCD type 1)
  - Preserve the old value:
    - store the old values as rows (SCD type 2)
    - store them as columns (SCD type 3)

Slowly Changing Dimension (SCD)

- Store 7 was in region 1, but it is now in region 2

key	store	region	status
1	7	1	expired
2	7	2	active

what if store 7 moves region again, to region 3, for example?

Storing Historical Information As a Row

key	store	current_region	old_region	effective_date
1	7	2	1	11/18/2007

Storing Historical Information As a Column

Rapidly changing dimension (RCD)

- Ex: a customer dimension with ten attributes.
  - Attributes 1 to 9 change once a year.
  - Attribute 10 changes every day
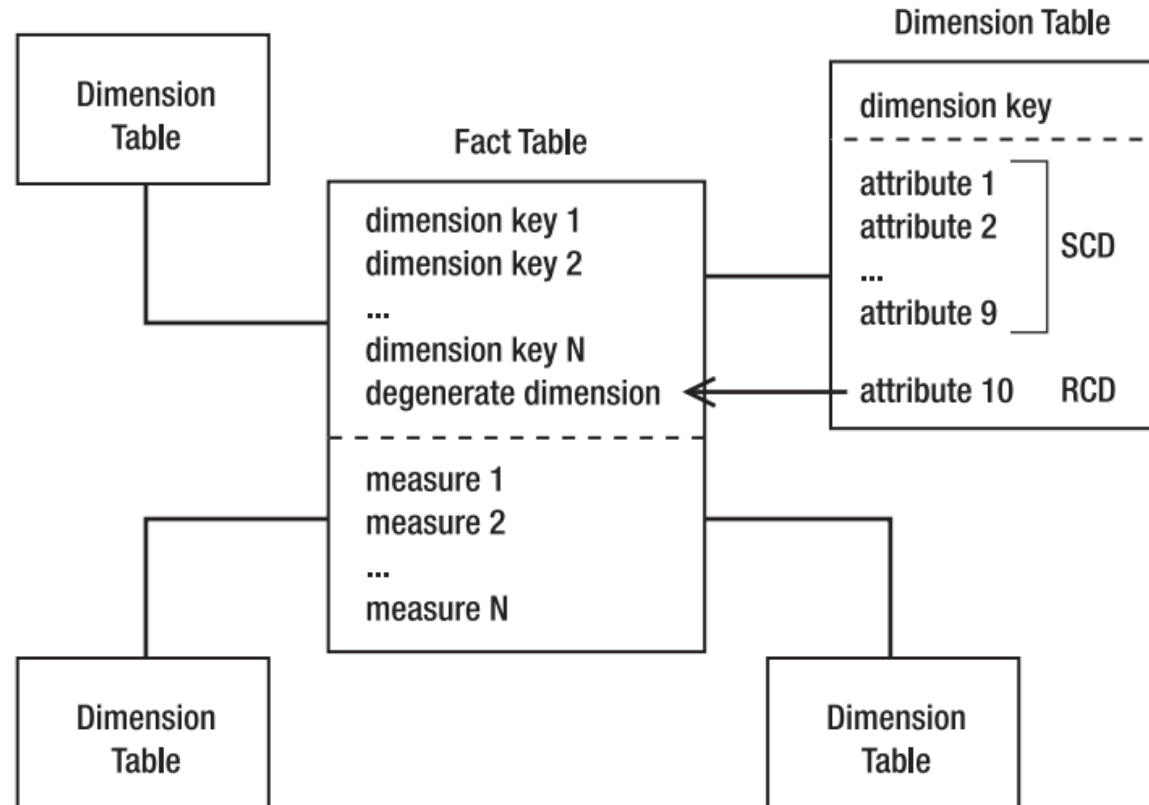
  → **How do we store the historical information?**

Degenerate dimension

- Context that is insignificant for analysis such as invoice no, ticket no, …in an transaction
- Has an informative role, is stored in a fact table

Rapidly changing dimension (RCD)

Fact

- **A fact** is a **measurement** of a business activity, such as a business event or transaction, and is generally numeric
  - Fact: sales
  - Numeric measurements: counts, dollar amounts, percentages, or ratios
- Facts can be aggregated or derived
  - Ex: sum up the total revenue or calculate the profitability of a set of sales transactions
- Facts provide the measurements of how well or how poorly the business is performing
- Fact tables are normalized and contain little redundancy

FACT TABLE

- Fact tables are composed of two types of columns:
  - Keys
    - a group of foreign keys (FK) that point to the primary keys of dimensional tables
  - Measures
    - actual measures of the business activity
    - Every measurement has a grain, which is the level of detail in the measurement of an event

FACT TABLE KEYS

- **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 primary key of a fact table is typically a multipart key consisting of the combination of foreign keys that can uniquely identify the fact table row.
  - primary key is a surrogate key.
  - primary key with degenerative dimensions

FACT TABLE KEYS

- The relationships between fact tables and the dimensions are **one-to-many**

- If combining a subset of foreign keys creates uniqueness, then this multipart key will become the primary key
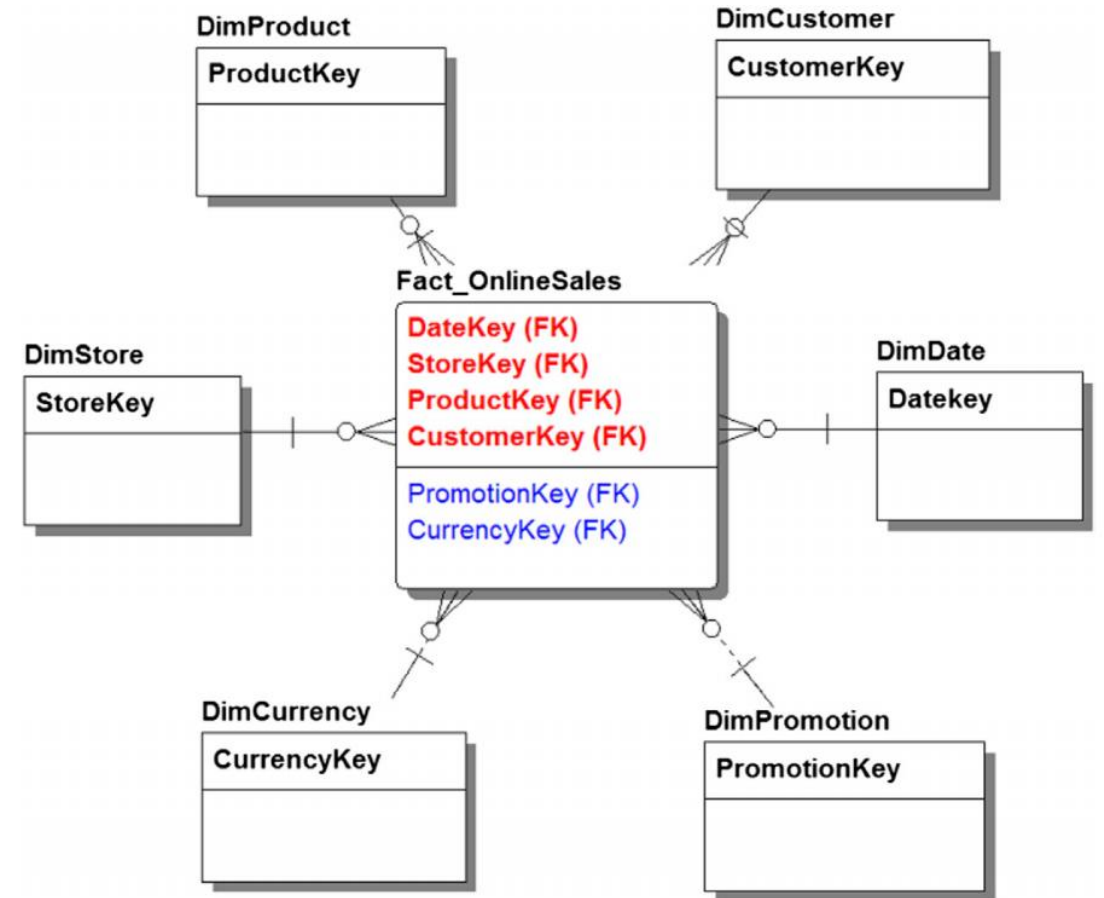
Fact keys:
(DateKey, StoreKey, ProductKey, CustomerKey)

Fact table—primary key is a surrogate key

- if you cannot identify unique rows with any of the methods discussed so far, create a primary key based on a **surrogate key:**
  - **is often generated by the database system using an IDENTITYdata type**
  - **is an integer whose value is meaningless**
- OnlineSalesKey: is the surrogate key that was created as the PK
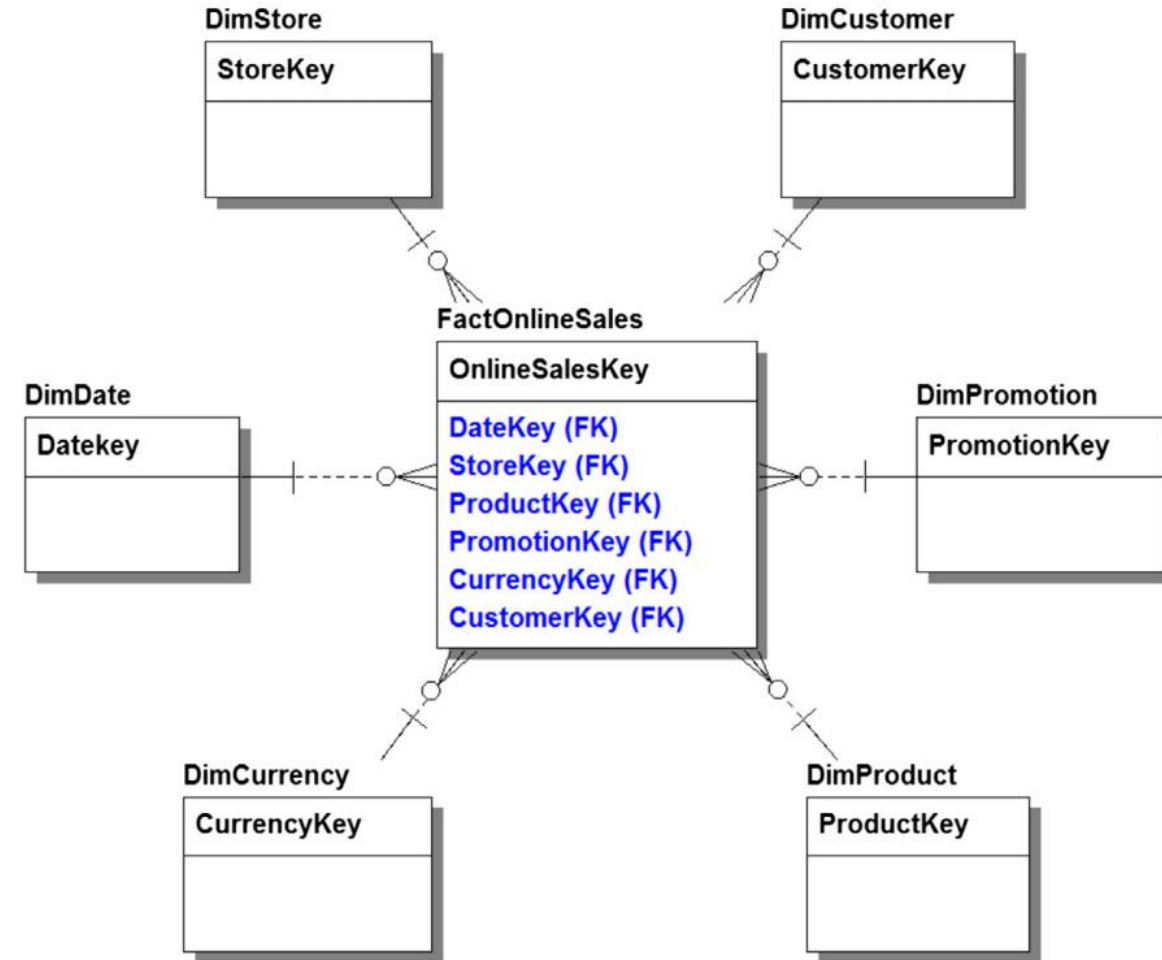
FACT TABLE MEASURES

- Every measurement has a grain, which is the level of detail in the measurement of an event
- Granularity is determined by its data source, indicates the level of detail stored in a table. Example:
  - Each Time record represents a day
  - Each Product record represents a product
  - Each Organization record represents one branch
  - ➡ then the grain of a sales Fact table with these dimensions is sales per product per day per branch
- Granularity at **too high** a level severely limits the ability of users to obtain additional detail
- Granularity at **too low** a level results in an exponential increase in the size requirements of the DW

FACT TABLE MEASURES

- **Type of fact measure**:
  - Additive Facts
    - can be summed across any of the dimensions associated with the fact table.
  - Semi-additive Facts
    - can be summed across some dimensions, but not all
  - Non-additive Facts
    - can't be added across any dimensions

FACT MEASURES - Additive Facts

- is the easiest to define and manage, can be added across all dimensions
  - Ex: the quantity of items (number of books)

Date		FactSales		Product	
DateKey	→	DateKey	←	**ProductKey**	
Year		ProductKey		ProName	
Quarter		SalePrices		Category	
Month				SubCategory	
day					
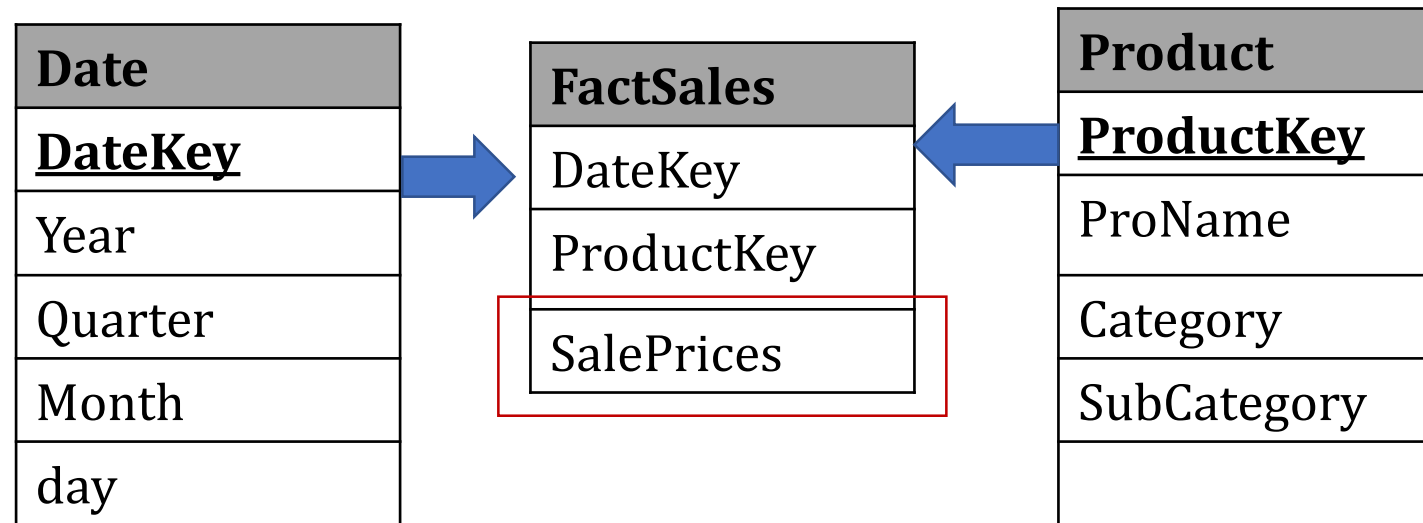
FACT MEASURES - SemiAdditive Facts

- Can be added across some dimensions but not others
- Ex: bank account balances, the number of students attending a class, or inventory levels

Date
DateKey
Year
Quarter
Month
day

FactAccounts
AccountKey
DateKey
TypeKey
CustomerKey
Balance

Customer
CusKey
CusName

AccType
TypeKey
TypeName
AccountKey
CusKey
DateKey

http://sqlserver-qa.net/2015/06/25/different-types-of-facts-and-fact-tables-in-data-warehouse-design/
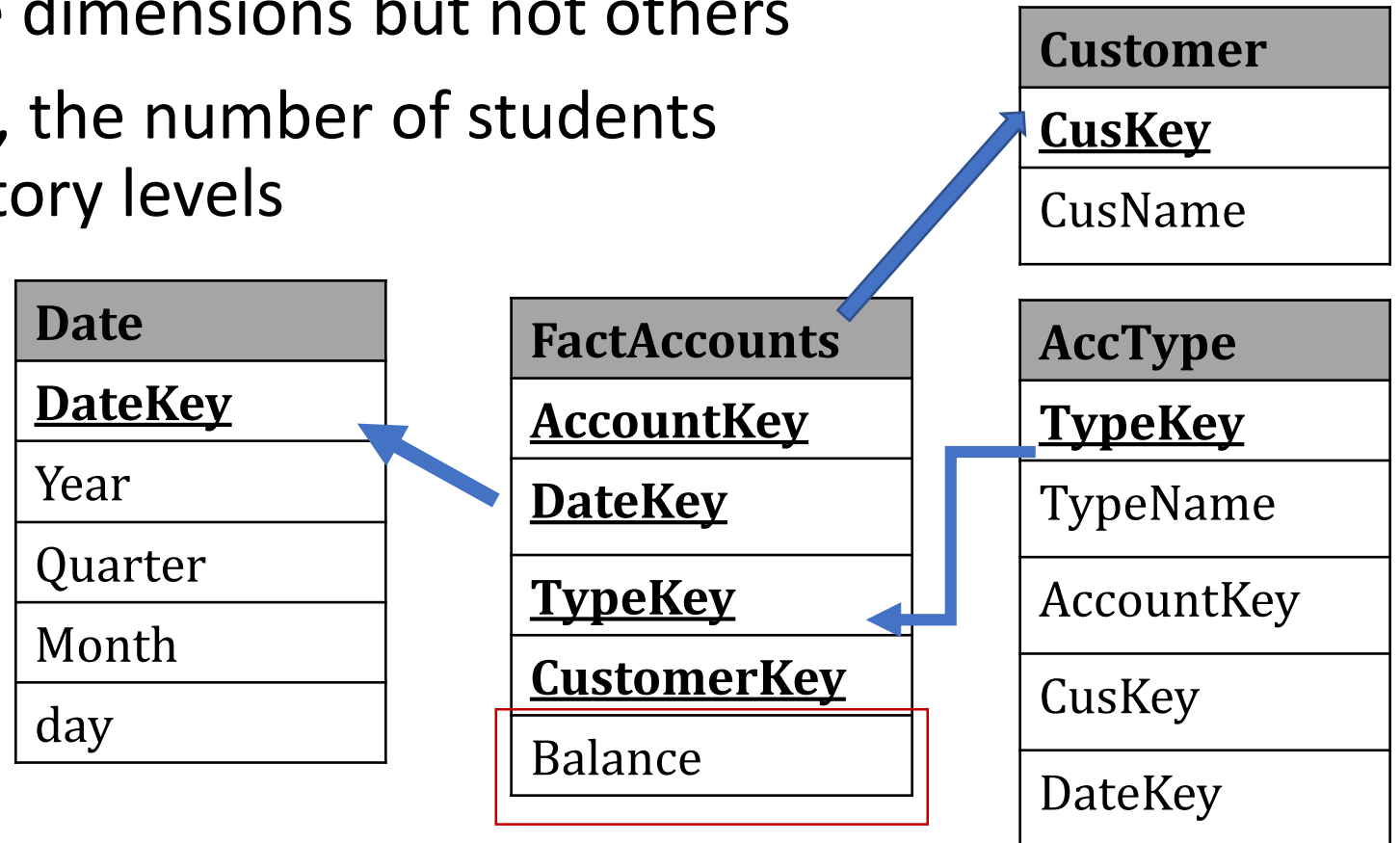
FACT MEASURES – NonAdditive Facts

- Can't be added across any dimensions
- Ex: ratios, and temperatures;
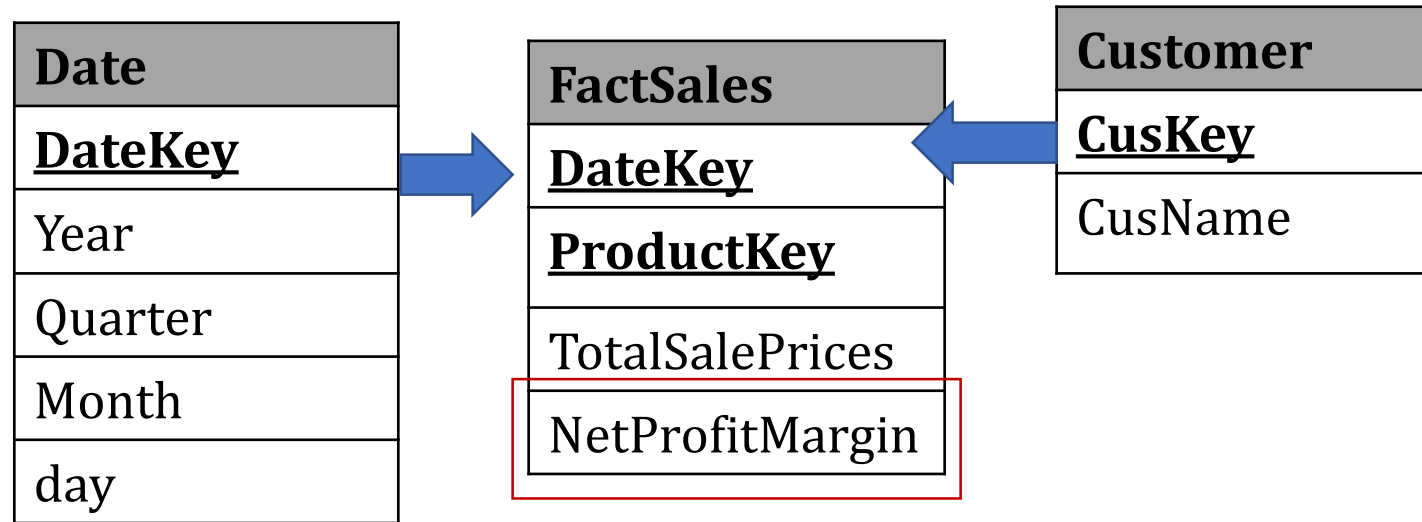
Date
DateKey
Year
Quarter
Month
day

FactSales
DateKey
ProductKey
TotalSalePrices
NetProfitMargin

Customer
CusKey
CusName

http://sqlserver-qa.net/2015/06/25/different-types-of-facts-and-fact-tables-in-data-warehouse-design/

SCHEMA

- which schema should use when building the dimensional model?
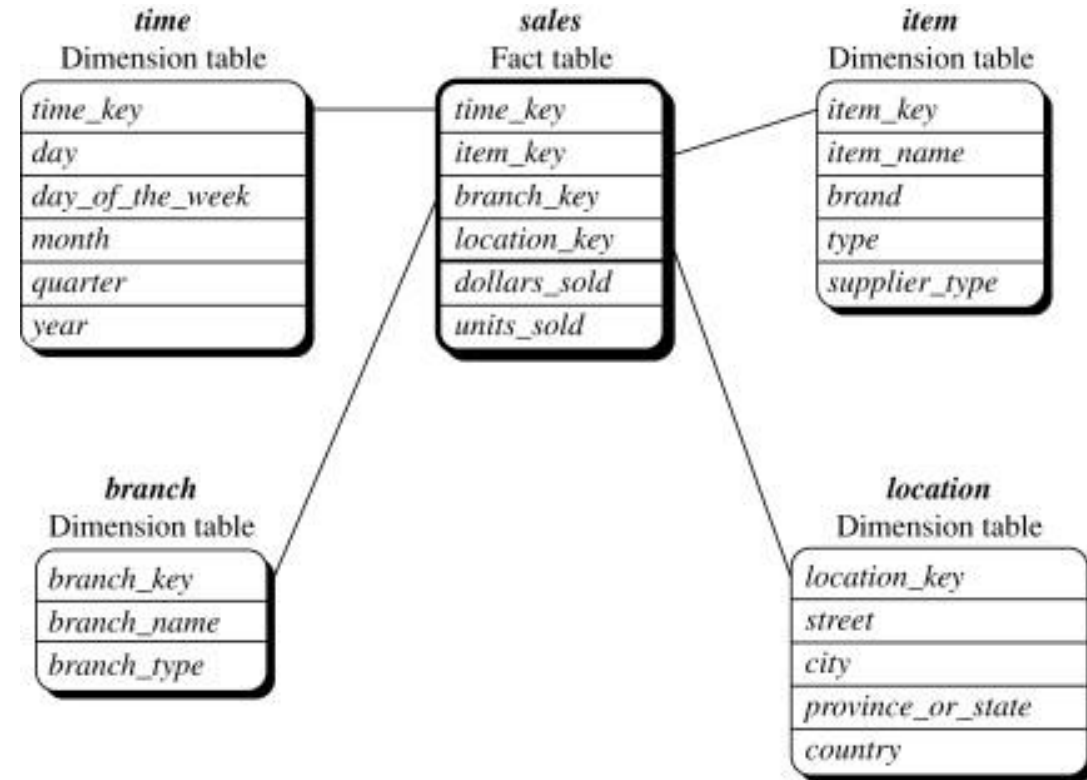 - What kind of analysis are you trying to perform on that data and how complex is it?
  - What are the analytical requirements and restrictions?
  - How consistent is the data you want to query and analyze?
  - What BI tool do you plan to use?

Star Schema

- is the most common schema for dimensional models
- It is a fact table surrounded by multiple **denormalized** dimension tables
- a dimension does not have a subtable (a subdimension)
- easier for the ETL processes to load the data into DDS

time
Dimension table

| time_key |
| day |
| day_of_the_week |
| month |
| quarter |
| year |

sales
Fact table

| time_key |
| item_key |
| branch_key |
| location_key |
| dollars_sold |
| units_sold |

item
Dimension table

| item_key |
| item_name |
| brand |
| type |
| supplier_type |

branch
Dimension table

| branch_key |
| branch_name |
| branch_type |

location
Dimension table

| location_key |
| street |
| city |
| province_or_state |
| country |

Snowflake schema

- a dimension can have a subdimension → is to minimize redundant data
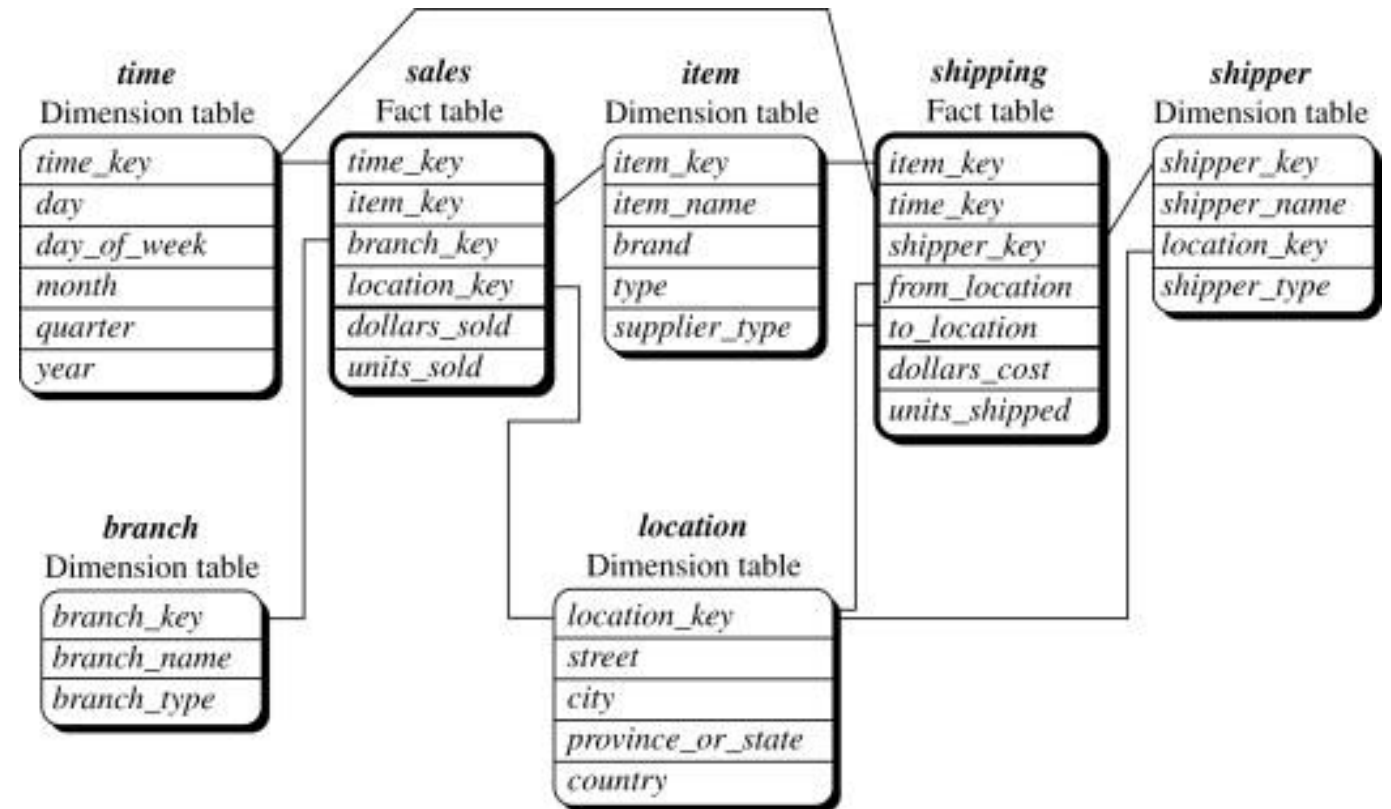- The fact table stores the foreign key to the lowest level of the dimensional hierarchy.

Galaxy schema

time
Dimension table
| time_key |
| day |
| day_of_week |
| month |
| quarter |
| year |

sales
Fact table
| time_key |
| item_key |
| branch_key |
| location_key |
| dollars_sold |
| units_sold |

item
Dimension table
| item_key |
| item_name |
| brand |
| type |
| supplier_type |

shipping
Fact table
| item_key |
| time_key |
| shipper_key |
| from_location |
| to_location |
| dollars_cost |
| units_shipped |

shipper
Dimension table
| shipper_key |
| shipper_name |
| location_key |
| shipper_type |

branch
Dimension table
| branch_key |
| branch_name |
| branch_type |

location
Dimension table
| location_key |
| street |
| city |
| province_or_state |
| country |

- This schema specifies two fact tables, *sales* and *shipping*
  - allows dimension tables to be shared between fact tables.
  - For example, the dimensions tables for *time, item*, and *location* are shared between the *sales* and *shipping* fact tables
  - A fact table may be in relationship with some (not all) dimension table

Exercises

- steps for data warehouse dimensional modeling example:
  - Step 1: Chose Business Objective
    - how many paracetamol and diclofenac tablets sold from single MedPlus store every day
  - Step 2: Identify Granularity
    - MedPlus shop sells 1,000 paracetamol tablets on a specific day then granularity is daily and 10,000 on specific month then granularity would be monthly
  - Step 3: Identify Dimension and its attributes
    - Shop", "Medicine", and "Day"
  - Step 4: Identify Fact
    - number of tablets sold is a measure