

# Dimensional Modeling

**CS 537- Big Data Analytics**

**Dr. Faisal Kamiran**

## Dimensional Modeling (DM)

- Introduced by Ralph Kimball in 1996  
(The word “Kimball” is synonymous with dimensional modeling.)
- Includes a set of methods and techniques to optimize data storage in a Data Warehouse
- Optimizes the database for faster retrieval
- Dimensional Models divide data into **measurements (facts)** and their **descriptive contexts (dimensions)**

## Dimensional Modeling VS Relational Modeling

**Dimensional Models** are used in data warehousing systems to answer business questions. They are designed to read, summarize and analyze numeric data.

**Relational Models** are used in transaction systems where many transactions are executed. They are optimized for addition, updating and deletion of data in these systems.

Relational models are not useful for analysis type of queries

Most relational models have a similar looking design. There are hundreds of tables connected by an even larger number of join paths. The result is overwhelming, and, from a business user perspective, unusable. No human being or computer software can analyze and run queries on it in a meaningful amount of time.

## Collaboration in Dimensional Modeling

- Design should always be done in collaboration with business experts
- Dimensional model should be developed via interactive workshops between the data modeler and subject matter experts
- Important: **Collaboration** is critical

The workshops provide another opportunity to flesh out the requirements with the business.

**Dimensional models should not be designed in isolation by folks who don't fully understand the business and their needs**

## Dimensional Modeling Process

Four key decisions made during the design of a dimensional model:

1. Select the business process
2. Declare the grain
3. Identify the dimensions
4. Identify the facts

The answers to these questions are determined by considering the needs of the business along with the realities of the underlying source data during the collaborative modeling sessions.

Following the business process, grain, dimension, and fact declarations, the design team determines the table and column names, sample domain values, and business rules. Business data governance representatives must participate in this detailed design activity to ensure business buy-in.

## Gathering Business Requirements

- Data modeler needs to understand the **needs of the business** as well as their underlying **data**
- Requirements are uncovered via sessions with business representatives
- Includes understanding DM objectives, business issues, decision-making processes and required analytic needs
- The quality of the available data is also identified at this stage

## Grain

- The Grain describes the level of detail for the business problem/solution.
- It involves identifying the lowest level of information for each table

### **Example**

*"A manager wants to find the sales of different products on a daily basis."*

Here, the grain is product sales by **day**

Grain is level of detail

## Facts and Dimensions

### **Facts**

- Measurements that result from a business process event
- Typically numeric

### **Dimensions**

- The “who, what, where, when, why, and how” context surrounding a business process event.

Facts are very specific, well-defined numeric attributes



## Facts and Dimensions

### **Example**

What is the average annual faculty salary of CS department?

## Facts and Dimensions

### Example

What is the **average annual faculty** salary of CS department?

Measurement

## Facts and Dimensions

### Example

What is the average annual faculty salary of CS department?

Dimensional Context

## Facts and Dimensions

- Work together to create an organized data model
- While fact and dimension are not created differently in the DDL, they are conceptual and extremely important for organization.
  - Fact and dimension tables are most often just explicit tables in a DB.

## Fact Tables

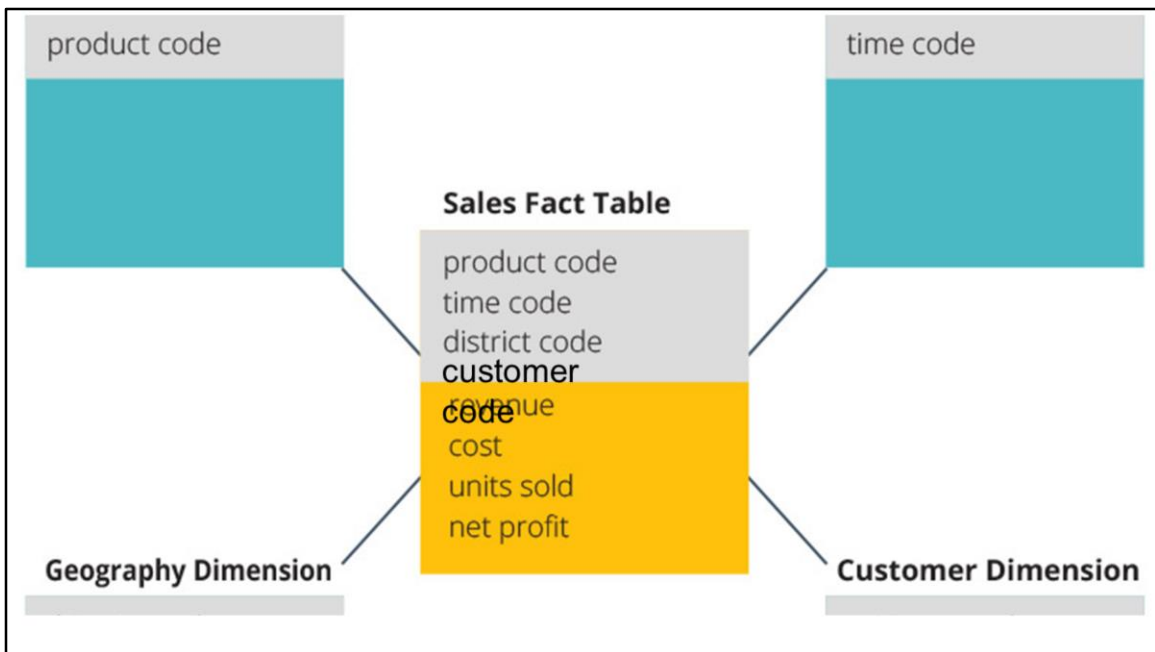
Fact tables consist of the measurements, metrics or facts of a business process.

- Fact tables are made up of facts (events that have actually happened).
- Fact tables can be aggregations of data and aren't meant to be updated at place.
- Fact tables normally have integers or numbers.
- Fact tables also typically have quantitative data. The quantity sold, the price per item, total price, and so on.

## Dimension

A structure that categorizes facts and measures in order to enable users to answer business questions. Dimensions are people, products, place and time.

- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes



Example of a dimensional model. The measurements include numeric attributes in a sales transaction, while the dimensions include the product in the transaction, time of the transaction, location of the store and customer who made the transaction

Model measurements as fact tables with multiple foreign keys referring to the contextual entities.

Fact tables always create a characteristic many-to-many relationship among the dimensions. Many customers buy many products in many stores at many times.

## Fact or Dimension Dilemma

- **Fact tables**

- Record business events, like an order, a phone call, a book review
- Fact tables columns record events recorded in quantifiable metrics like quantity of an item, duration of a call, a book rating.

- **Dimension tables**

- Record the context of the business events, e.g., who, what, where, why, etc.
- Dimension tables columns contain attributes like the store at which an item is purchased, or the customer who made the call, etc.



## Facts are numeric and **additive**

Additivity is crucial because data warehouse applications **almost never retrieve a single fact table record**; rather, they **fetch back hundreds**, thousands, or even millions of these records at a time, and almost the only useful thing to do with so many fact records is to add them up.

## Facts (Additivity)

- OLAP queries involve retrieving many fact table rows and aggregating them e.g.
  - *“Total university tuition fess collected in 2019”*

Because most fact tables are huge, with millions or even billions of rows, you almost never fetch a single record into your *queried* answer set.

Rather, you fetch a very large number of records, which you compress into digestible form by adding, counting, averaging, or taking the min or max. But for practical purposes, the most common choice, by far, is **adding**.

So it is imperative to store facts which are additive

## Facts (Additivity)

- OLAP queries involve retrieving many fact table rows and aggregating them e.g.
  - *"Total university tuition fess collected 2019"*
  - Tuition Payment measure is additive so it can be aggregated in the result

Tuition_Payment_Fact		
<u>Tuition_Payment</u>	<u>Student_Key</u>	<u>Date_Key</u>
\$7,000.00	732017235	88085255
\$6,500.00	481011832	88085255
\$7,000.00	881838281	82324174
\$7,000.00	298191999	13216661
...	...	...

Example of tuition payment fact. Stores the paid tuition fess by students along with the the student and date dimensions

## Facts (Additivity)

A data warehousing fact can be:

- Additive
  - An additive fact can be added under all circumstances e.g. sales amount
- Non-additive
  - Cannot be added
- Semi-additive
  - They can be added along some dimensions but not with others

## Facts (Additivity)

Example of a non-additive fact (GPA)

Student Credit Hours Tracking				
LastName	FirstName	Year...	Fall 2020 GPA	
Jackson	Sally	FR	3.3	
Thompson	Richard	SO	3.2	
Williams	Greta	FR	2.8	
Young	Ted	FR	4.0	
				13.3

Makes no sense to add the GPA of some individual students.

However, one can say that the average student GPA is a useful value and we can find that instead. However, please note that **additivity** relates solely to the ability to **add**

## Facts (Additivity)

Typical non-additive facts

- Ratios
- Percentages
- Calculated averages (e.g., GPA)

## Facts (Additivity)

Typical non-additive facts

- Ratios
- Percentages
- Calculated averages (e.g., GPA)

With non-additive facts

- Store underlying components in fact tables
- Calculate **aggregate** averages from the totals of these underlying components at report time

Store underlying components in fact tables e.g., for GPA, store credit hours and grade points instead of storing the ratio/average itself. GPA can then be calculated at report time

## Facts (Additivity)

### Semi-additive facts

- Can be added sometimes (along some dimensions)
- But other times, they cannot be added (along the other dimensions)

Customer_Key	Time_Key	Balance
618	201512141824	1500
618	201512141830	1400
700	201512141824	3000
700	201512141830	2800
701	201512141824	10000
701	201512141826	9800

Balance Fact: Fact table storing the account balance of customers at times.

Explain the table

Customer 618 had balance 1500. He withdrew some amount and the balance dropped to 1400. Similarly for other customers ...



## Facts (Additivity)

### Semi-additive facts

What is the total balance at time 201512141824?

1500 + 3000 + 10000

Balance_Fact		
Customer_Key	Time_Key	Balance
618	201512141824	1500
618	201512141830	1400
700	201512141824	3000
700	201512141830	2800
701	201512141824	10000
701	201512141826	9800

Can add balance along the customer dimension


## Facts (Additivity)

### Semi-additive facts

Cannot add along the  
time dimension

What is the total balance of customer 618?

Balance\_Fact

1500  1400

Customer_Key	Time_Key	Balance
618	201512141824	1500
618	201512141830	1400
700	201512141824	3000
700	201512141830	2800
701	201512141824	10000
701	201512141826	9800

We cannot say that customer 618's account balance is 1500+ 1400. This makes no sense

## Facts (Additivity)

### Semi-additive facts

However, we can perform other operations along  
the time dimension

Balance\_Fact

Customer_Key	Time_Key	Balance
618	201512141824	1500
618	201512141830	1400
700	201512141824	3000
700	201512141830	2800
701	201512141824	10000
701	201512141826	9800

$(1500 + 1400) / 2$

Customer  
618's  
average  
account  
balance is  
1450

Recommended approach is to always remember to divide by the number of time slots for such semi-additive facts

## Primary Key

- A unique identifier for each row in a database table
- **Natural Key**
  - Transferred from the source system to the DWH
  - Has **contextual or business meaning**
  - E.g., *PersonName*
- **Surrogate Key**
  - Generated artificially
  - Does not have any business meaning
  - Generated while transferring data to the DWH
  - Usually sequentially assigned integers

<https://www.sisense.com/blog/when-and-how-to-use-surrogate-keys/>

<https://www.kimballgroup.com/2009/05/the-10-essential-rules-of-dimensional-modeling/>

## Primary Key in Dimension Tables

- In dimension tables, use **surrogate key as the primary key**
  - Primary keys in dimension table are used as foreign keys in the fact table

Surrogate keys used because Natural keys do not stand the test of time. Symbols which might have been business meaning could become meaningless, or bear a different meaning in the future. Also, surrogate keys are smaller in size and allow faster indexing

In the future, when merging data from other systems, it might be possible that there is a possibility of conflicts in natural key

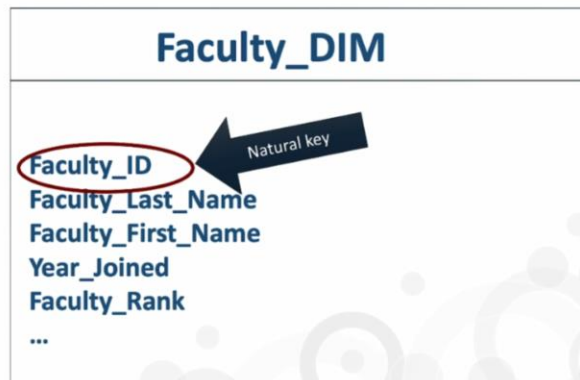
## Primary Key in Dimension Tables

Faculty_DIM	
Faculty_ID	
Faculty_Last_Name	
Faculty_First_Name	
Year_Joined	
Faculty_Rank	
...	

Consider a dimension table

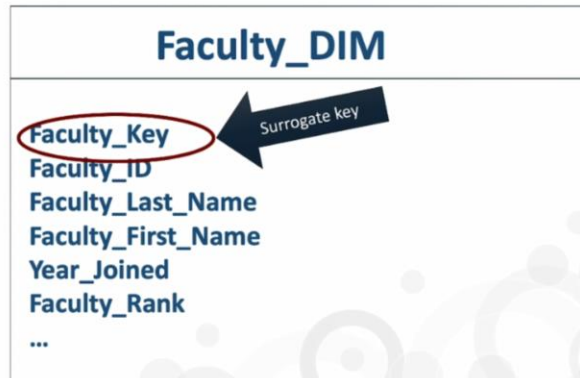
Note: Good practice to write \_DIM in table name to indicate that this is a dimension table

## Primary Key in Dimension Tables



Faculty\_ID is an identification number associated with each faculty member. This attribute is a part of the business process.

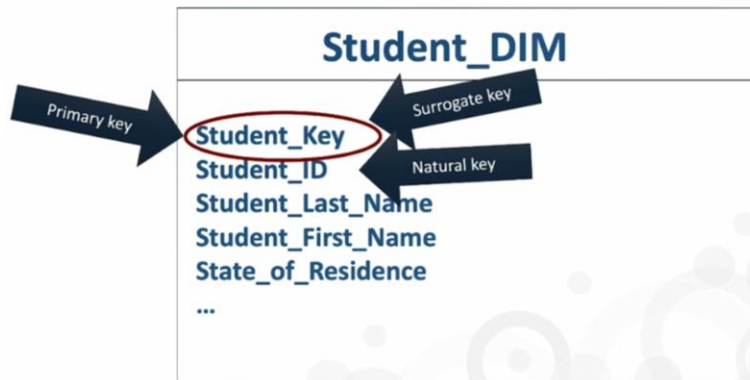
## Primary Key in Dimension Tables



However, according to the guidelines, we must use a surrogate key as the primary key. Therefore, a new attribute (typically integer) will be generated during the data modeling process and used as the primary key.



## Primary Key in Dimension Tables

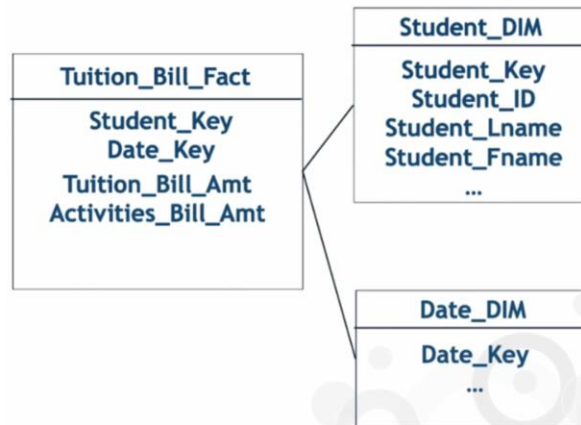


The surrogate key is used as the primary key

## Primary Key in Fact Tables

- The **combination of all foreign keys** relating back to dimension tables
  - Even if a fact table has a natural key

## Primary Key in Fact Tables



Consider the above schema representing the tuition fess of a student.  
The tuition and activities bill amounts are the facts and the student id and date are dimensions

## Primary Key in Fact Tables

