INFORMATION TECHNOLOGY UNIVERSITY

DATA SCIENCE LAB

Dimensional Modeling

CS 537- Big Data Analytics

Dr. Faisal Kamiran

Dimensional Modeling (DM)

- Introduced by Ralph Kimball in 1996
  (The word “Kimball” is now considered 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.

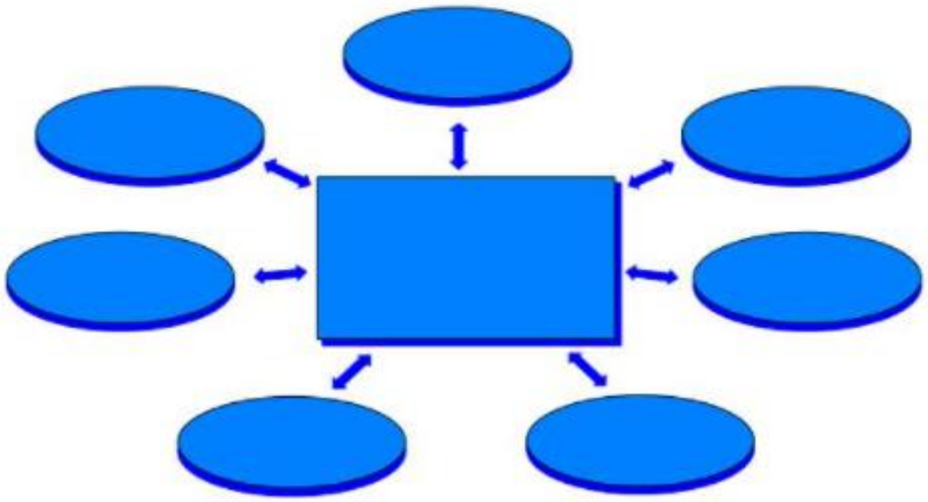
Collaboration in Dimensional Modeling

- 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

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



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**

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 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 (Fact)

Facts and Dimensions

Example

What is the average annual faculty salary of CS department?

Dimensional Information

Fact Tables

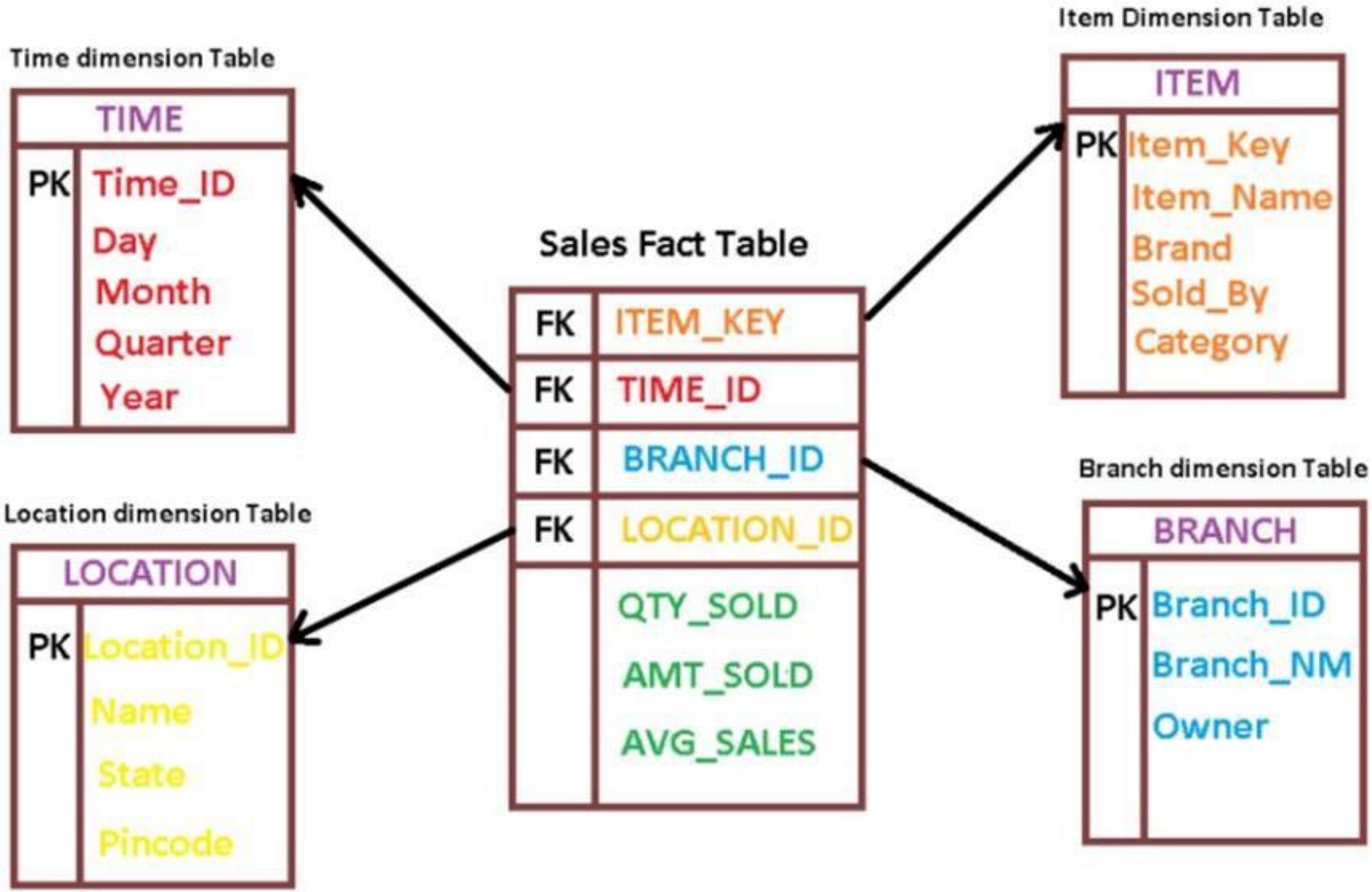
Fact tables consist of the measurements, metrics or facts of a business process.

- 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 denormalized 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


Time dimension Table
TIME
PK
Time_ID
Day
Month
Quarter
Year
Location dimension Table
LOCATION
PK
Location_ID
Name
State
Pincode
Sales Fact Table
FK
ITEM_KEY
FK
TIME_ID
FK
BRANCH_ID
FK
LOCATION_ID
QTY_SOLD
AMT_SOLD
AVG_SALES
Item Dimension Table
ITEM
PK
Item_Key
Item_Name
Brand
Sold_By
Category
Branch dimension Table
BRANCH
PK
Branch_ID
Branch_NM
Owner

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 (Aggregations)

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 (Additive)

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

Tuition_Payment_Fact		
Tuition_Payment	Student_Key	Date_Key
$7,000.00	732017235	88085255
$6,500.00	481011832	88085255
$7,000.00	881838281	82324174
$7,000.00	298191999	13216661
...	...	...

Facts (Non-Additive)

Typical non-additive facts

- Ratios
- Percentages
- Calculated averages

With non-additive facts

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

Facts (Non-Additive)

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

Facts (Additivity)

Semi-additive facts

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

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

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

Facts (Additivity)

Cannot add along the time dimension

Semi-additive facts

What is the total balance of customer 618?

~~1500 + 1400~~

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

Facts (Additivity)

Semi-additive facts

However, we can perform other operations along the time dimension

(1500 + 1400) / 2

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

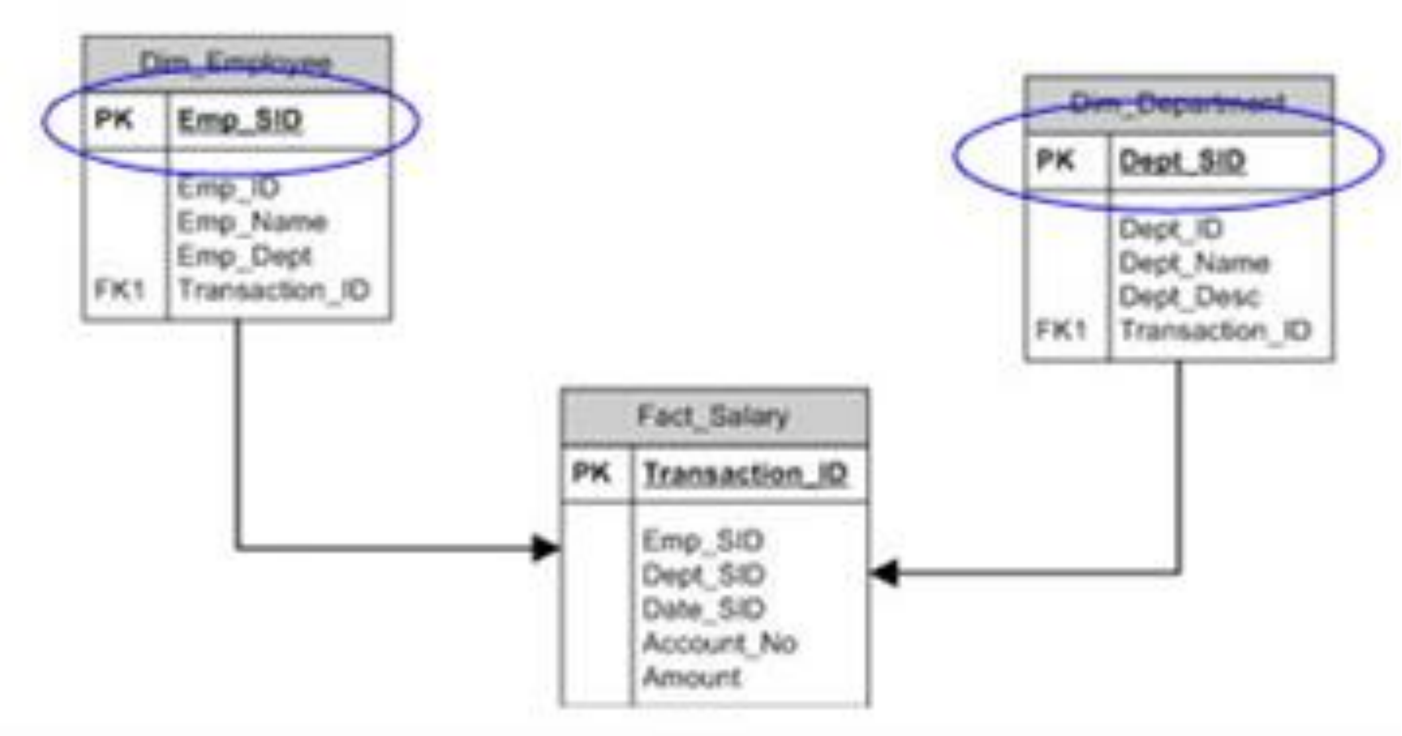
Customer 618’s average account balance is 1450

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

Primary Key in Dimension Tables

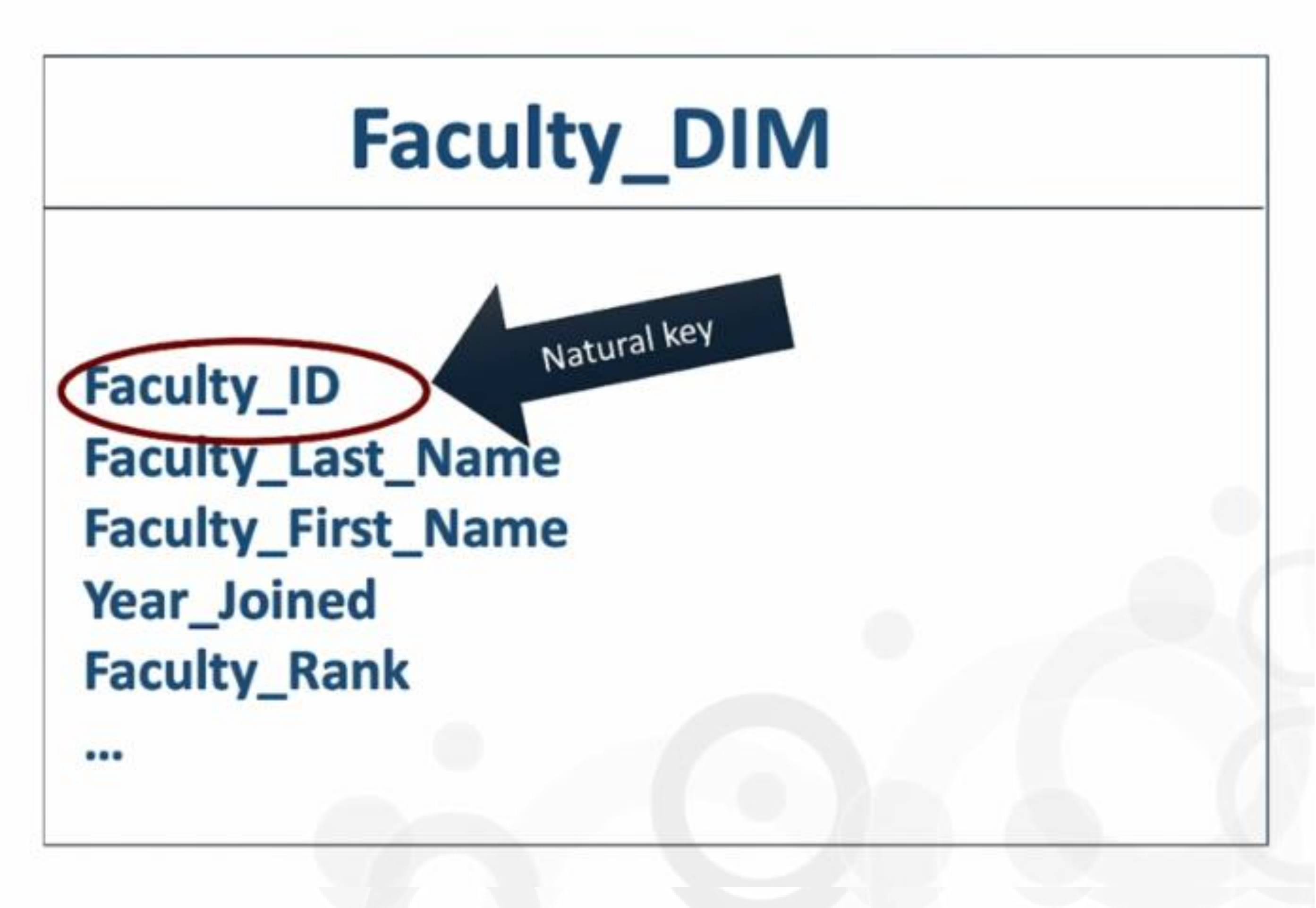
- In dimension tables, use **surrogate key as the primary key**
  - Primary keys in dimension table are used as foreign keys in the fact table



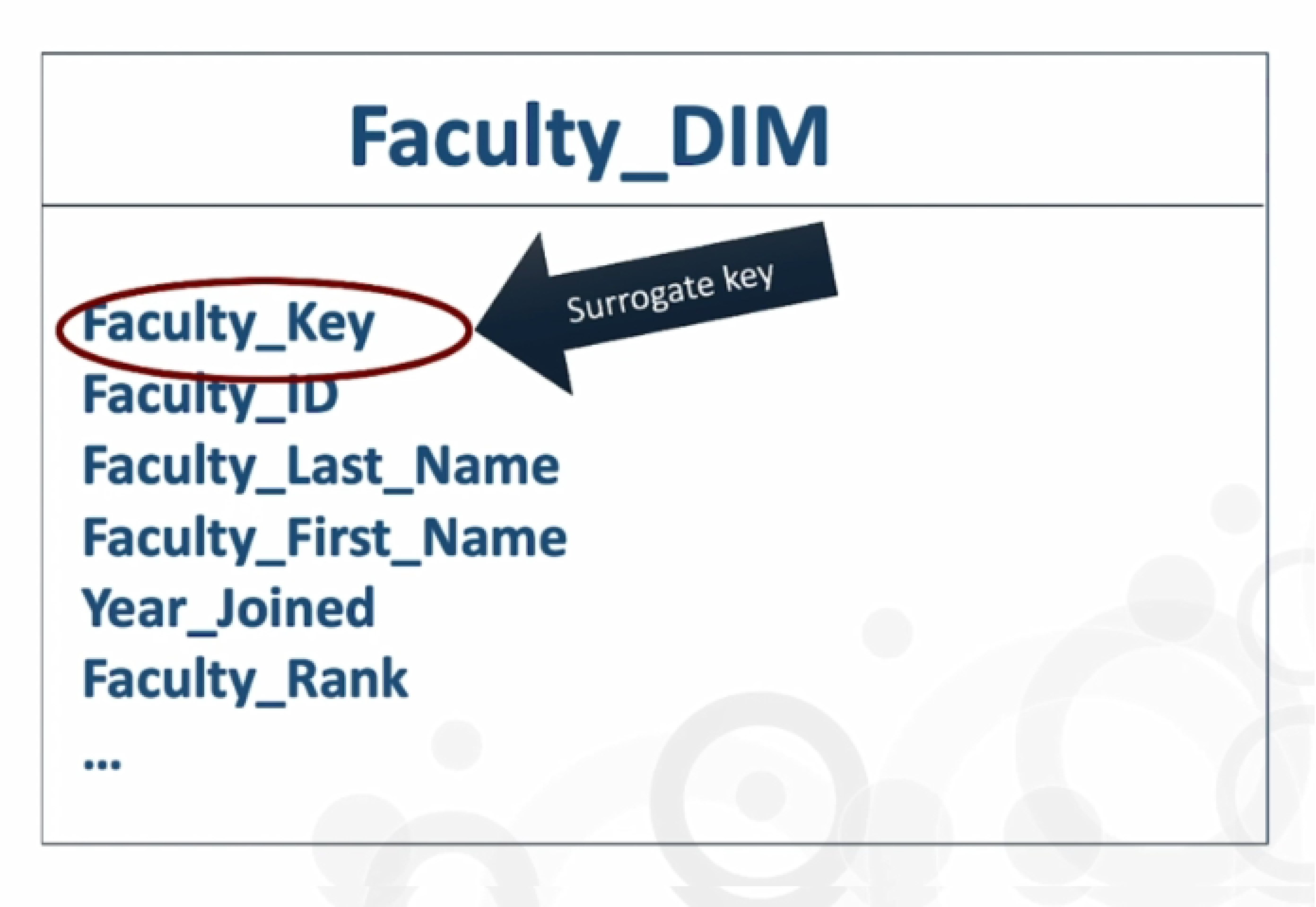
Primary Key in Dimension Tables

Faculty_DIM
Faculty_ID Faculty_Last_Name Faculty_First_Name Year_Joined Faculty_Rank ...

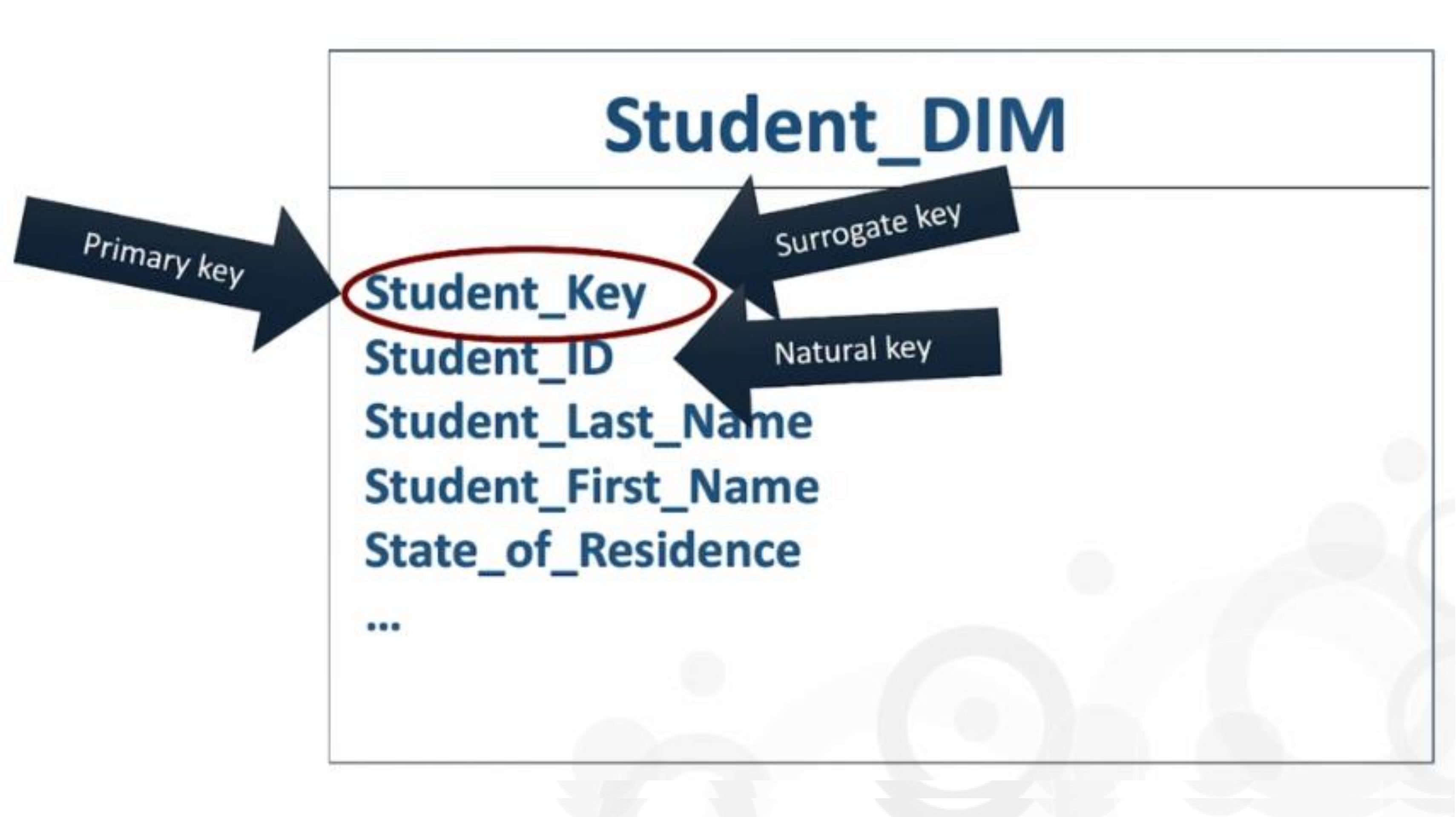
Primary Key in Dimension Tables



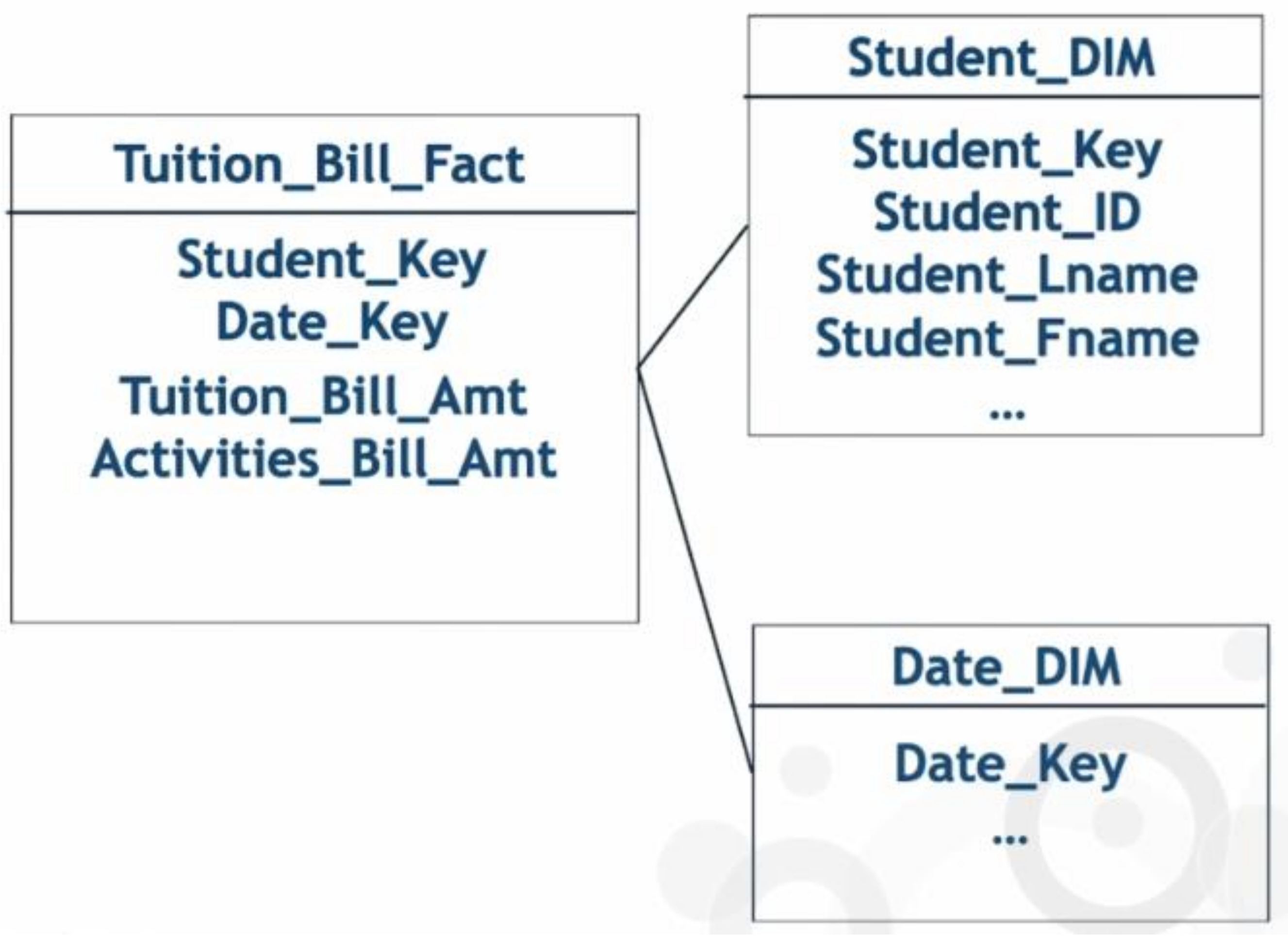
Primary Key in Dimension Tables



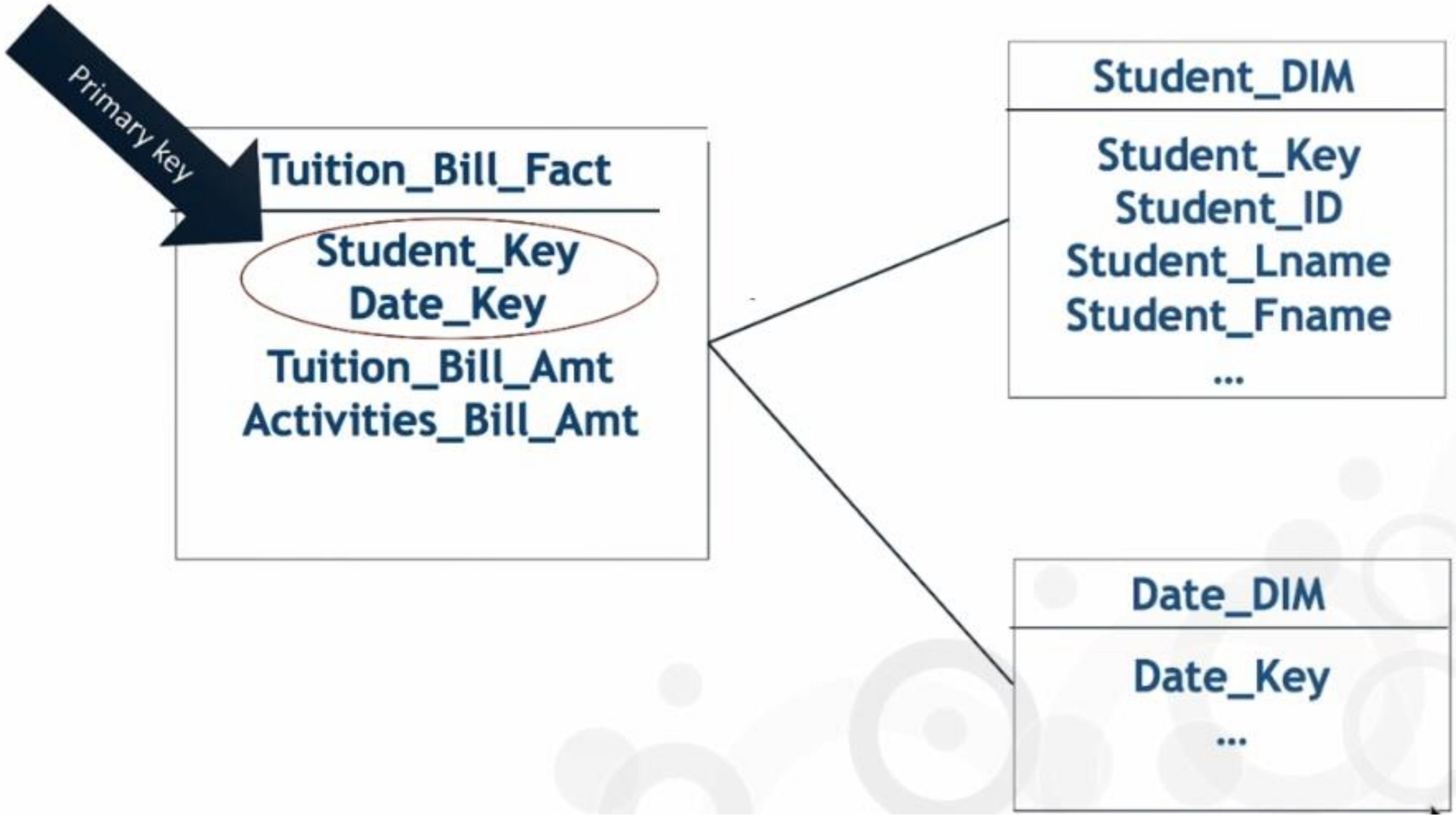
Primary Key in Dimension Tables



Primary Key in Fact Tables

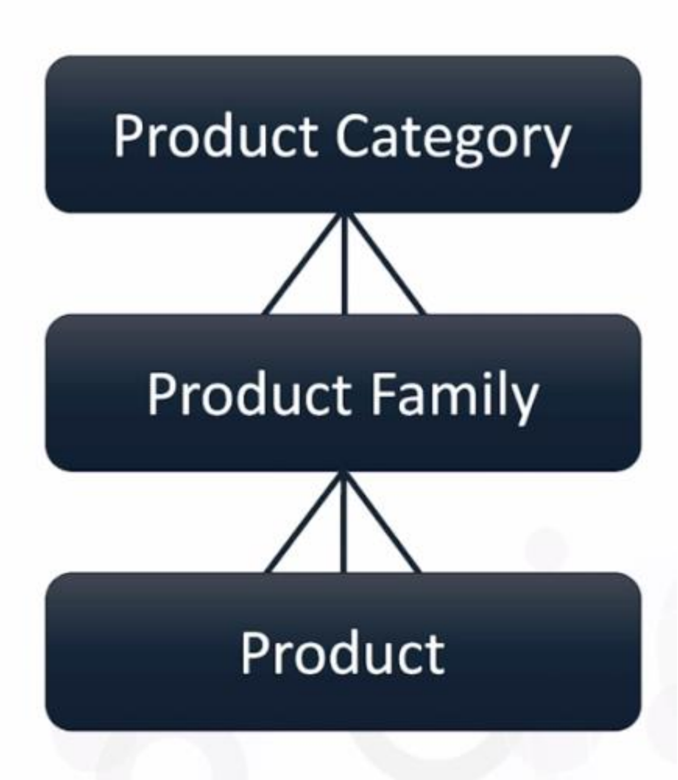


Primary Key in Fact Tables



Dimension Types

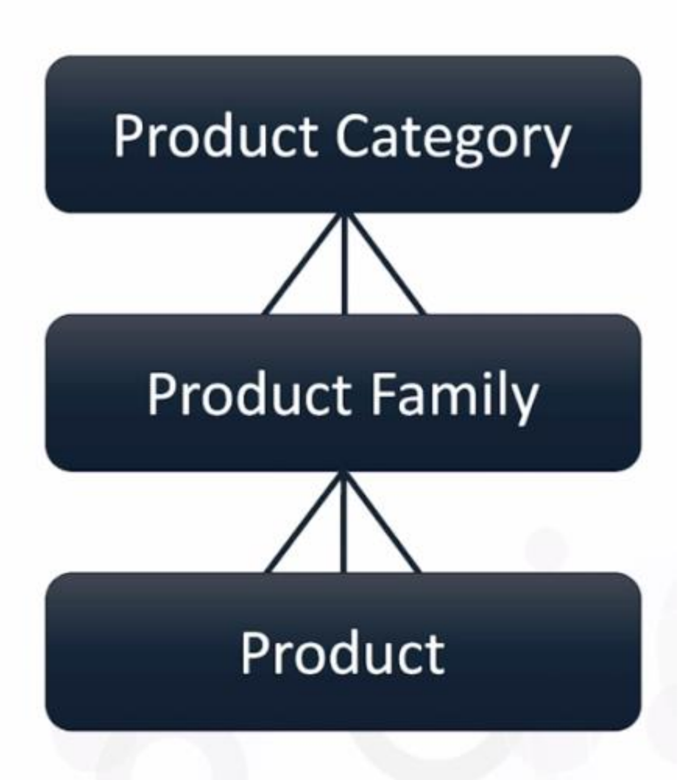
- Dimensions can consist of multiple hierarchies



Dimension

- Dimensions can consist of multiple hierarchies

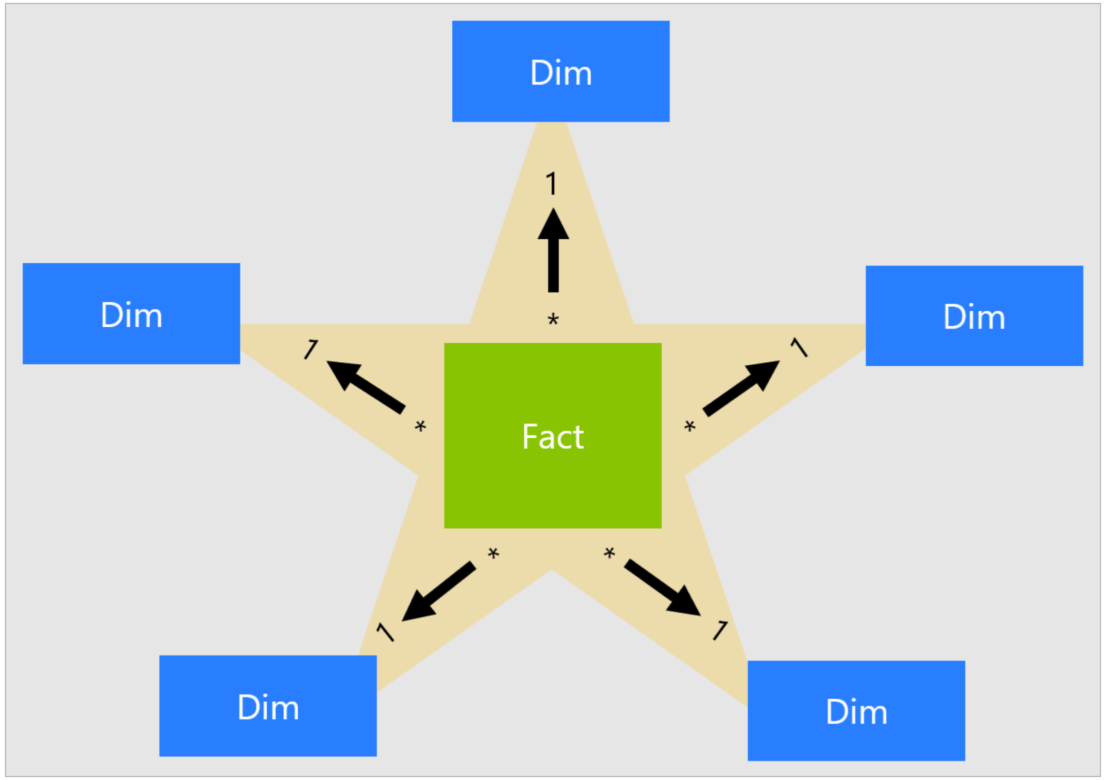
The product dimension will refer to the entire set of these objects



Implementing Different Schemas

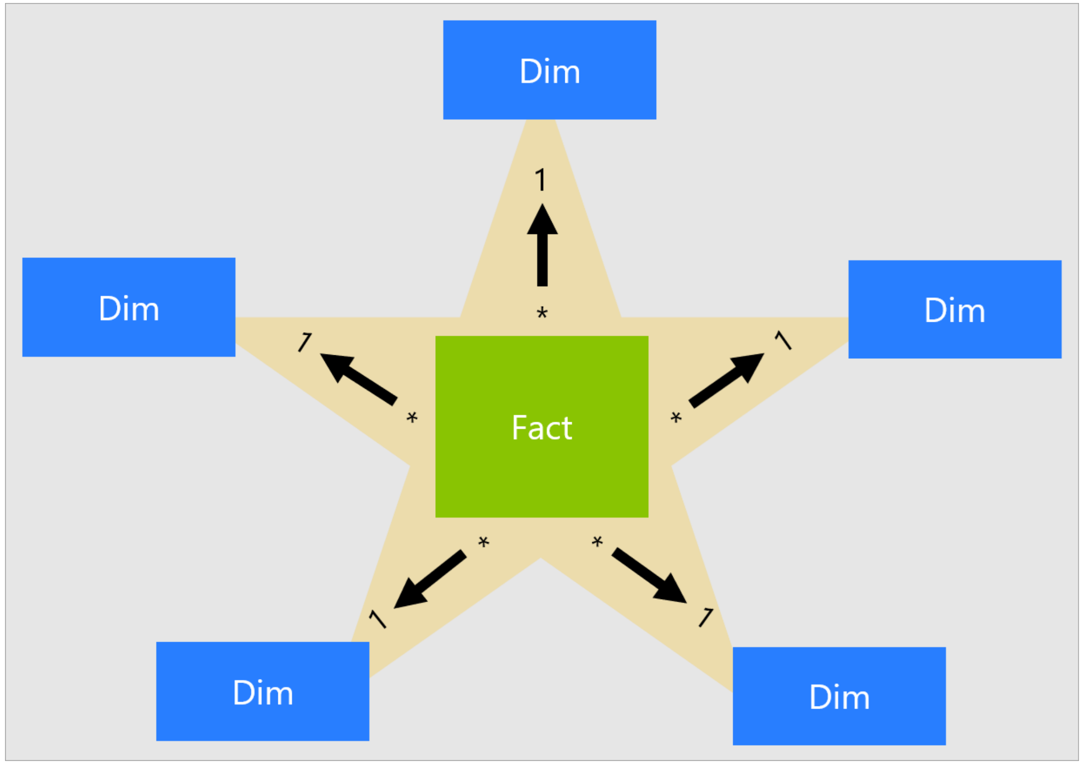
Two of the most popular (because of their simplicity) data mart schemas for data warehouses are:

- Star Schema
- Snowflake Schema



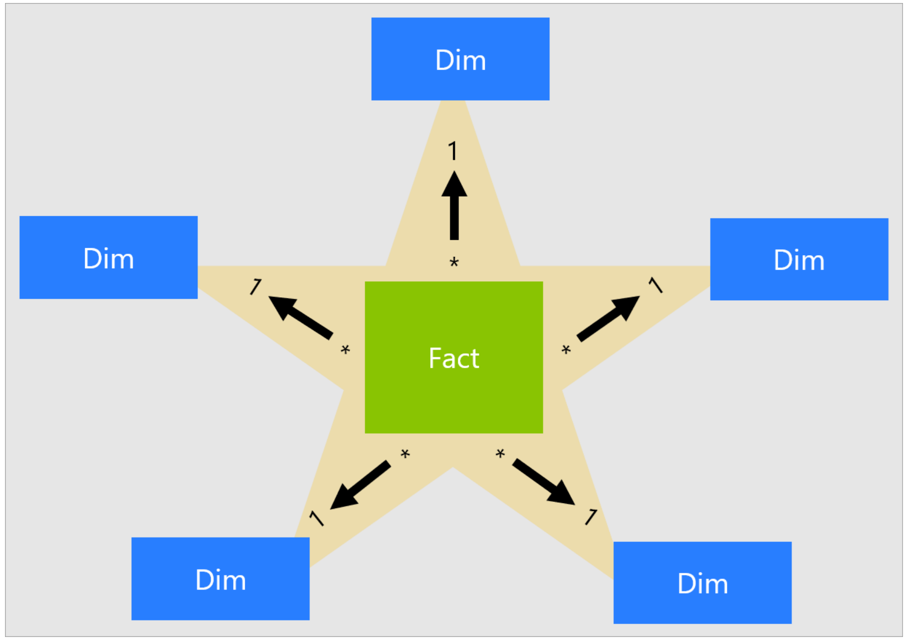
Star Schema

- Star Schema is the simplest style of data mart schema.
- The star schema consists of one fact table referencing any number of dimension tables.

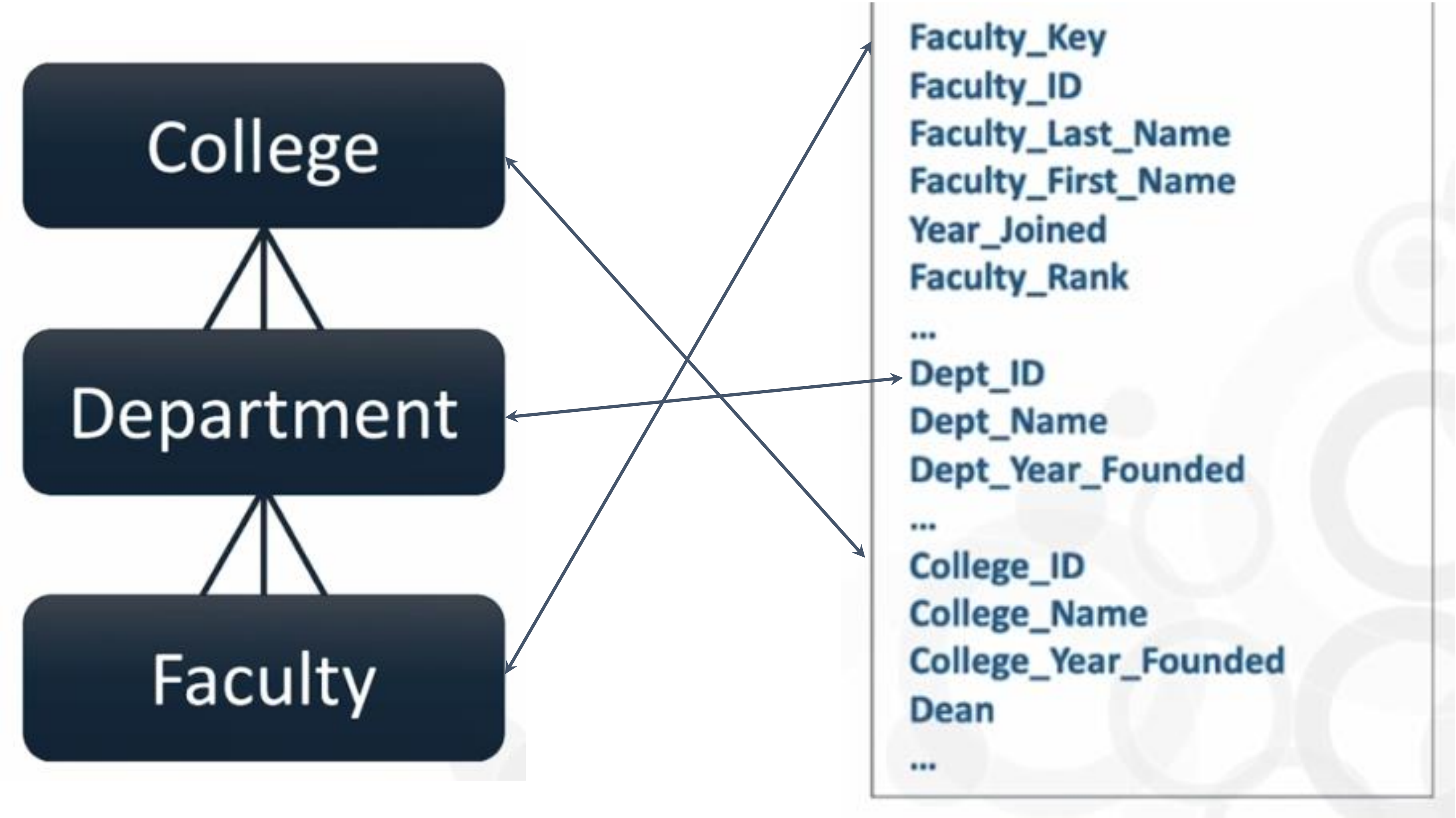


Why "star" schema?

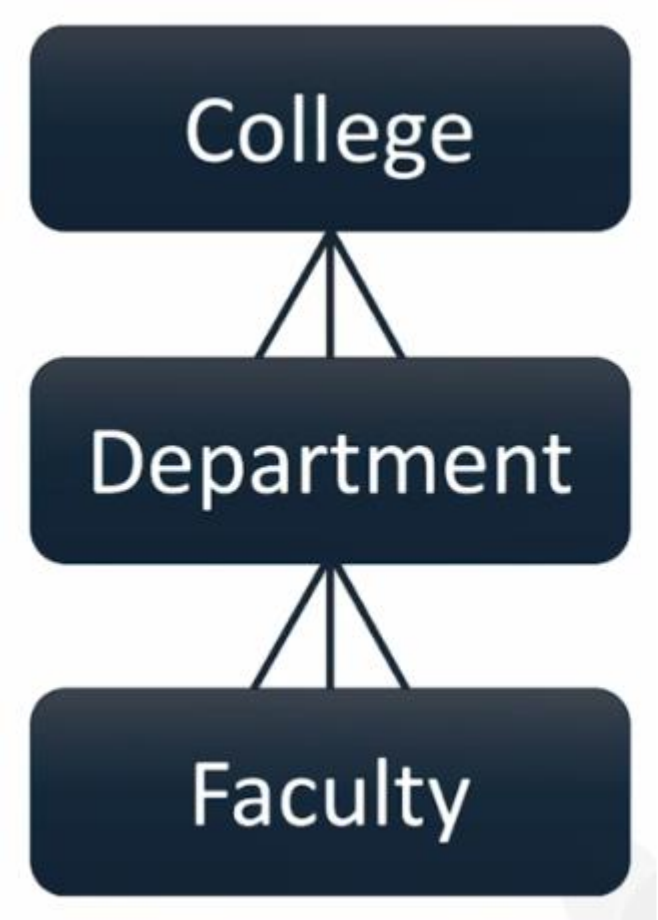
- Gets its name from the physical model resembling a star shape
- A fact table is at its center
- Dimension table surrounds the fact table representing the star's points.



Star Schema


College
Department
Faculty
Faculty_Key
Faculty_ID
Faculty_Last_Name
Faculty_First_Name
Year_Joined
Faculty_Rank
...
Dept_ID
Dept_Name
Dept_Year_Founded
...
College_ID
College_Name
College_Year_Founded
Dean
...

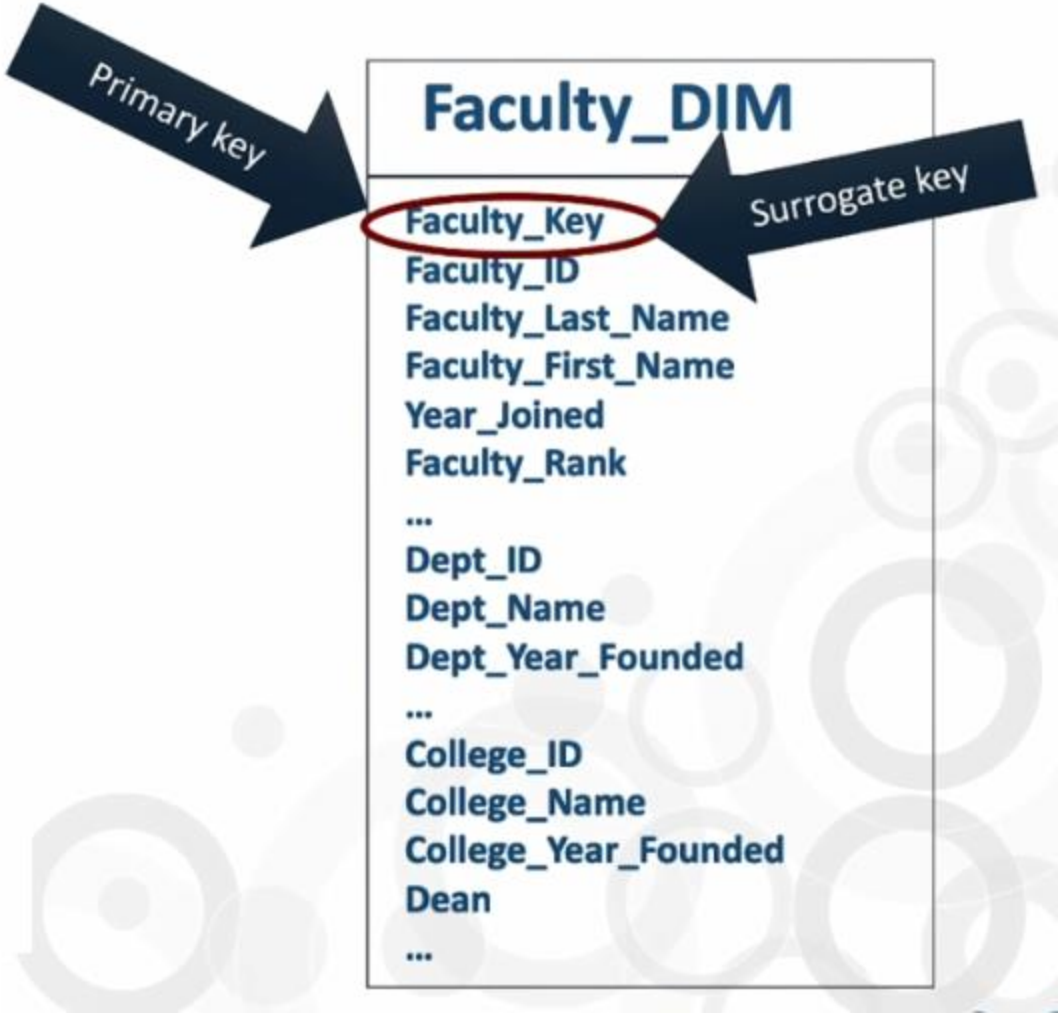
Star Schema

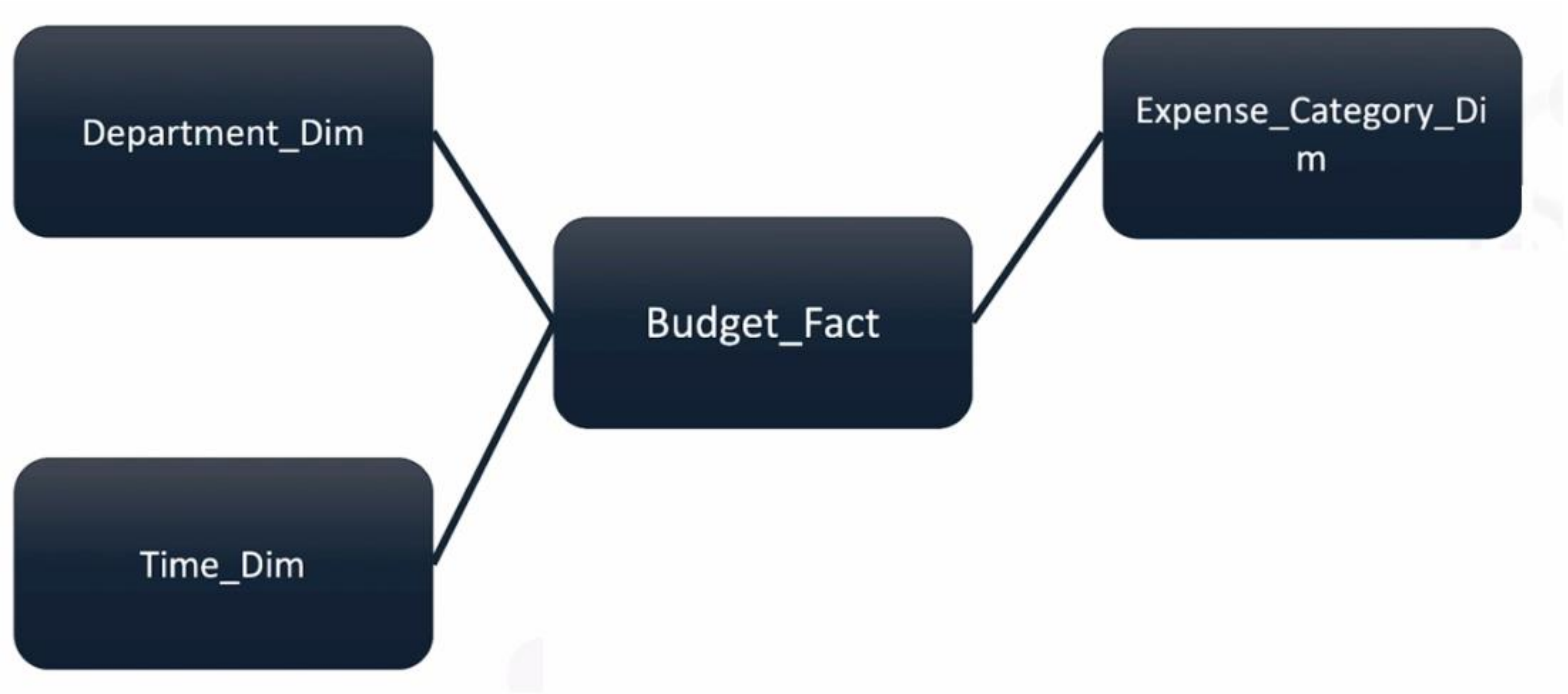


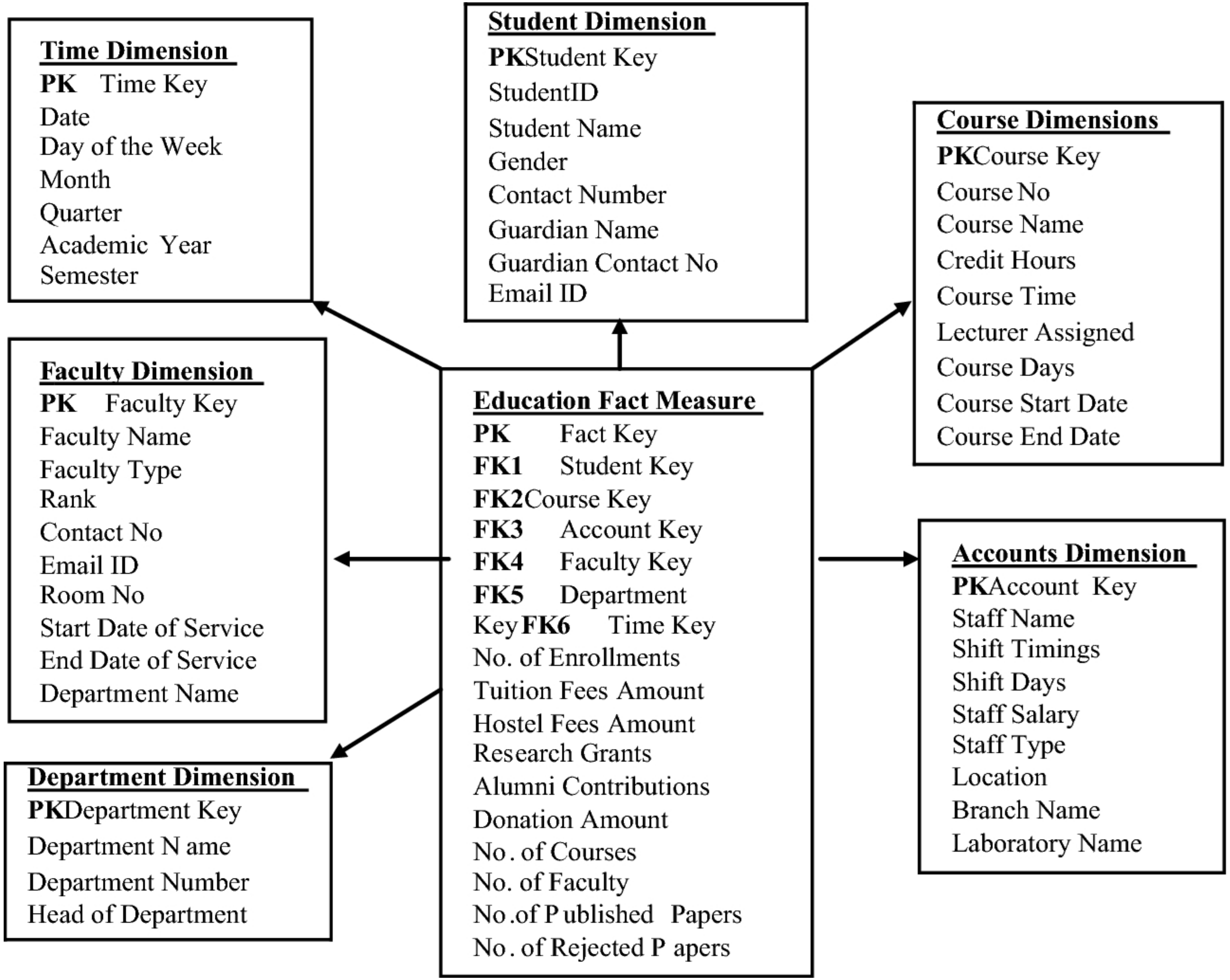
Faculty_Key
Faculty_ID
Faculty_Last_Name
Faculty_First_Name
Year_Joined
Faculty_Rank
...
Dept_ID
Dept_Name
Dept_Year_Founded
...
College_ID
College_Name
College_Year_Founded
Dean
...

Star Schema


College
Department
Faculty


Primary key
Faculty_DIM
Surrogate key
Faculty_Key
Faculty_ID
Faculty_Last_Name
Faculty_First_Name
Year_Joined
Faculty_Rank
...
Dept_ID
Dept_Name
Dept_Year_Founded
...
College_ID
College_Name
College_Year_Founded
Dean
...


Department_Dim
Expense_Category_Di
m
Budget_Fact
Time_Dim


Time Dimension
PK Time Key
Date
Day of the Week
Month
Quarter
Academic Year
Semester
Student Dimension
PKStudent Key
StudentID
Student Name
Gender
Contact Number
Guardian Name
Guardian Contact No
Email ID
Course Dimensions
PKCourse Key
Course No
Course Name
Credit Hours
Course Time
Lecturer Assigned
Course Days
Course Start Date
Course End Date
Faculty Dimension
PK Faculty Key
Faculty Name
Faculty Type
Rank
Contact No
Email ID
Room No
Start Date of Service
End Date of Service
Department Name
Education Fact Measure
PK Fact Key
FK1 Student Key
FK2Course Key
FK3 Account Key
FK4 Faculty Key
FK5 Department
Key FK6 Time Key
No. of Enrollments
Tuition Fees Amount
Hostel Fees Amount
Research Grants
Alumni Contributions
Donation Amount
No. of Courses
No. of Faculty
No.of Published Papers
No. of Rejected Papers
Accounts Dimension
PKAccount Key
Staff Name
Shift Timings
Shift Days
Staff Salary
Staff Type
Location
Branch Name
Laboratory Name
Department Dimension
PKDepartment Key
Department Name
Department Number
Head of Department

Benefits

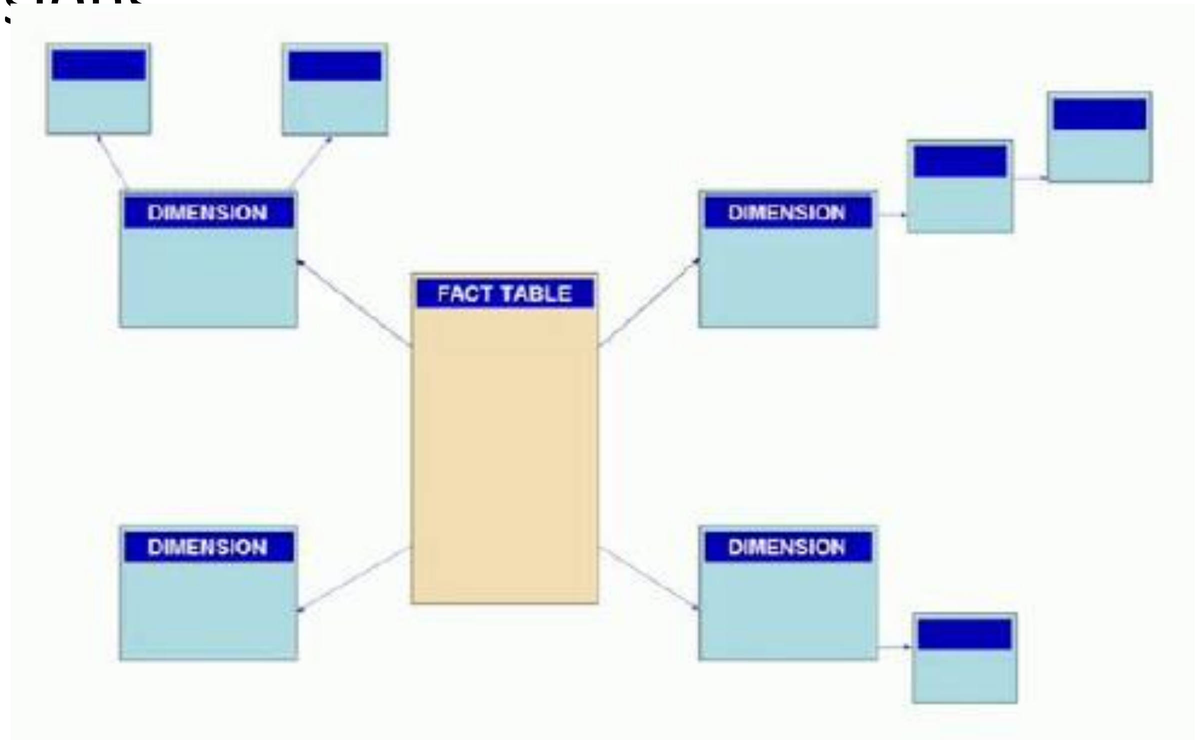
- Denormalized
- Simplifies queries
- Fast aggregations

Drawbacks

- Issues that come with denormalization
- Data Integrity
- Decrease query flexibility

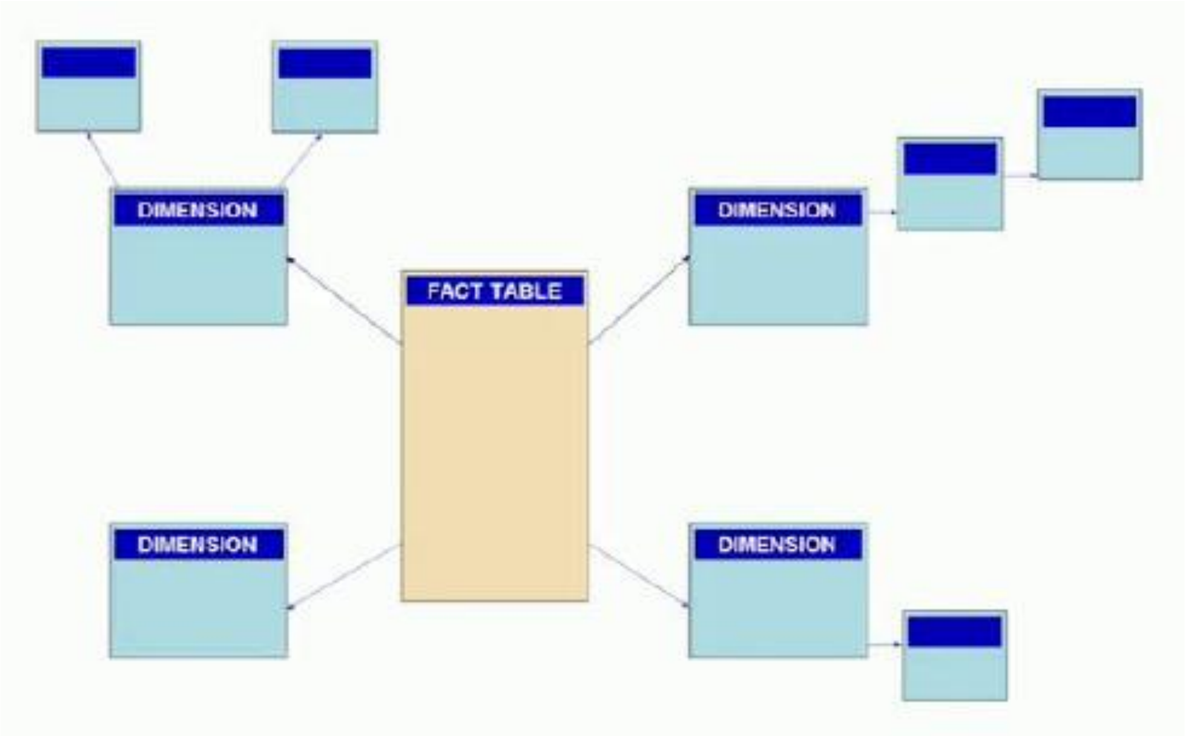
Snowflake Schema

Logical arrangement of tables in a multidimensional database represented by centralized fact tables which are connected to multiple dimensions

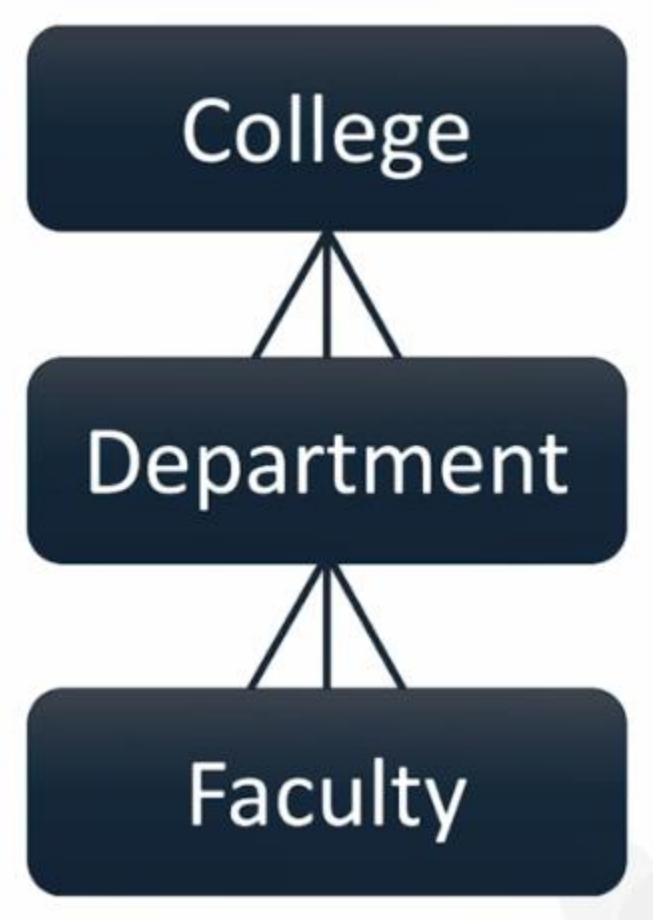


Why "snowflake" schema?

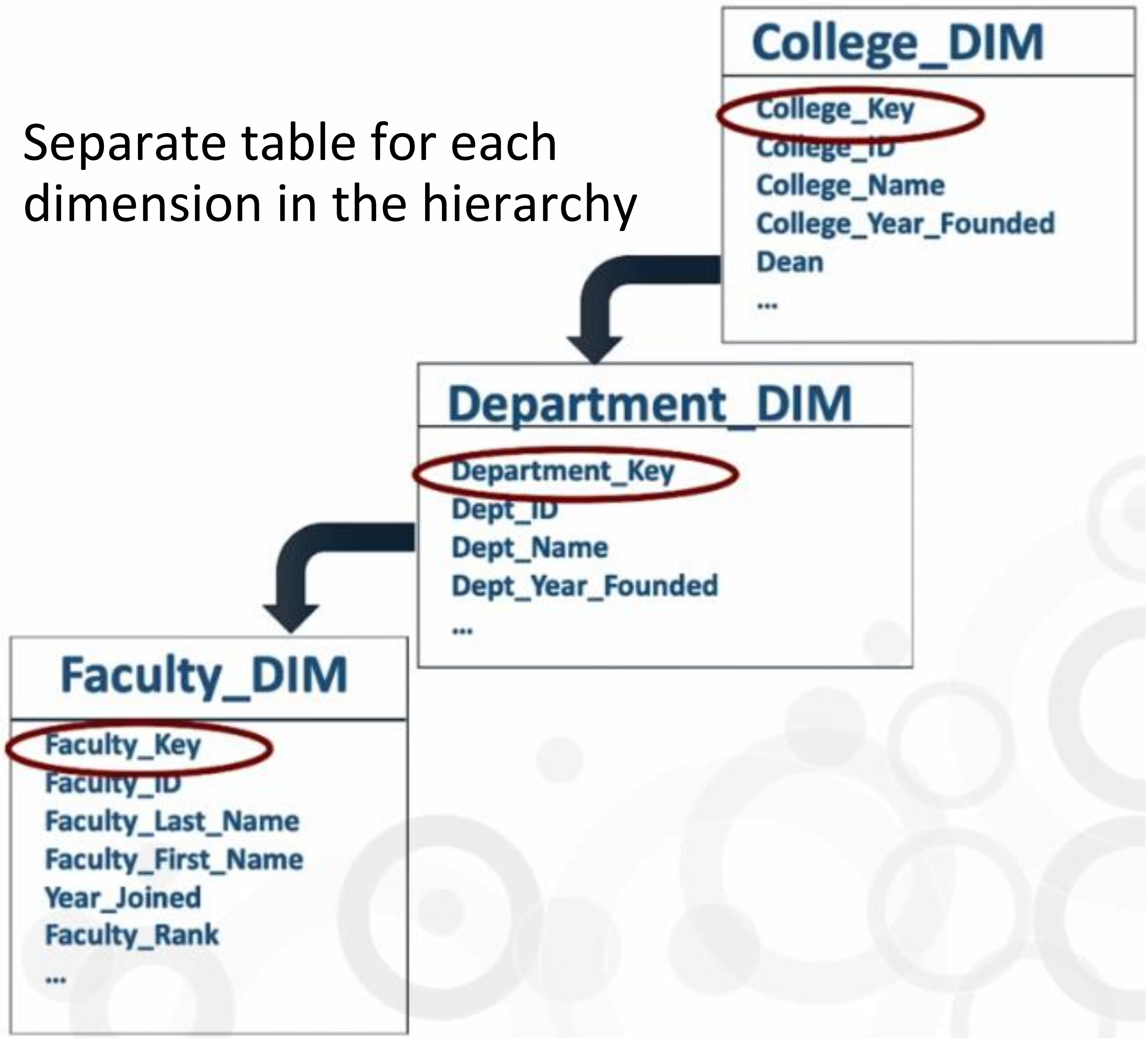
"A complex snowflake shape emerges when the dimensions of a snowflake schema are elaborated, having multiple levels of relationships, child tables having multiple parents."



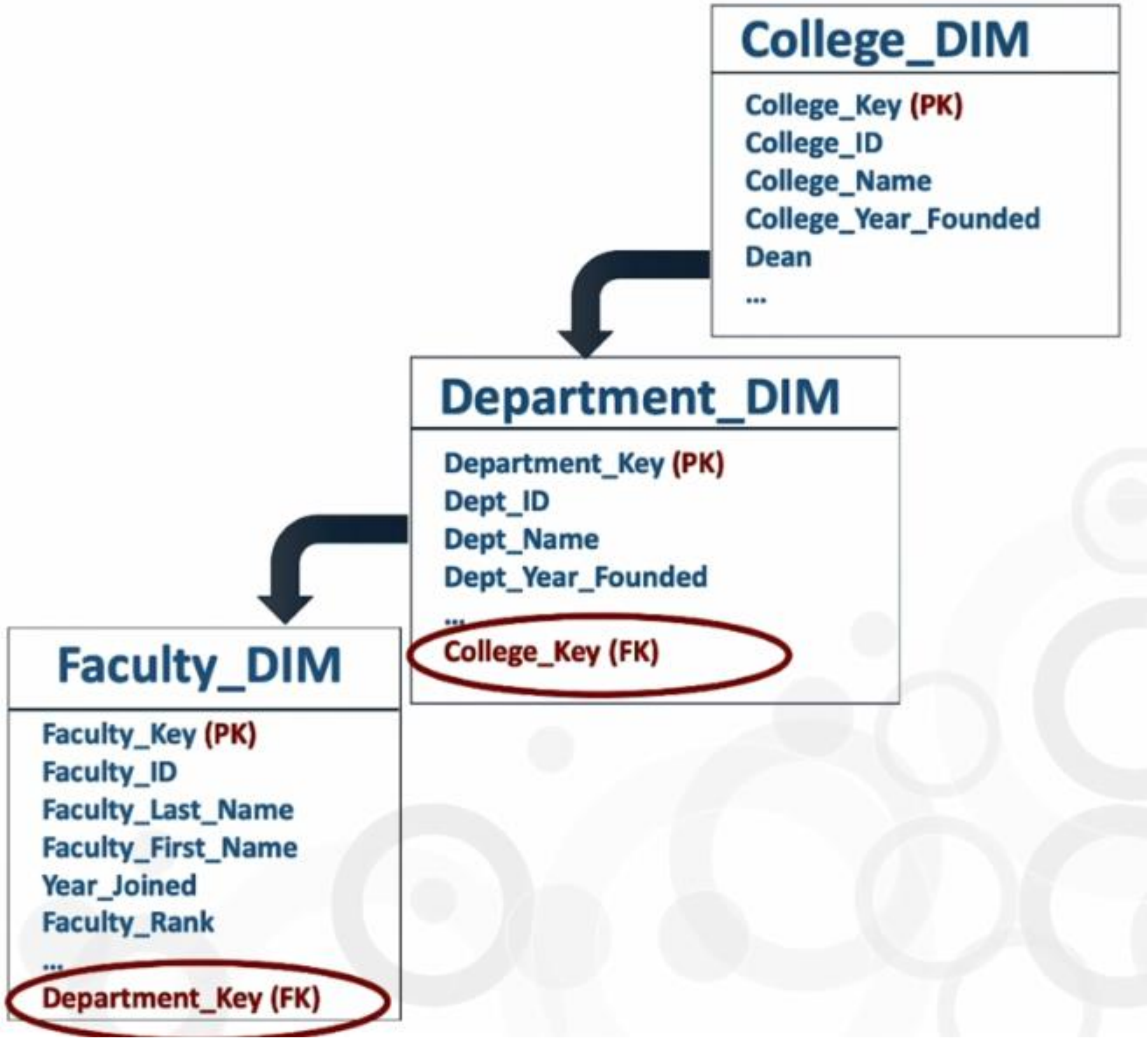
Snowflake schema



Separate table for each dimension in the hierarchy



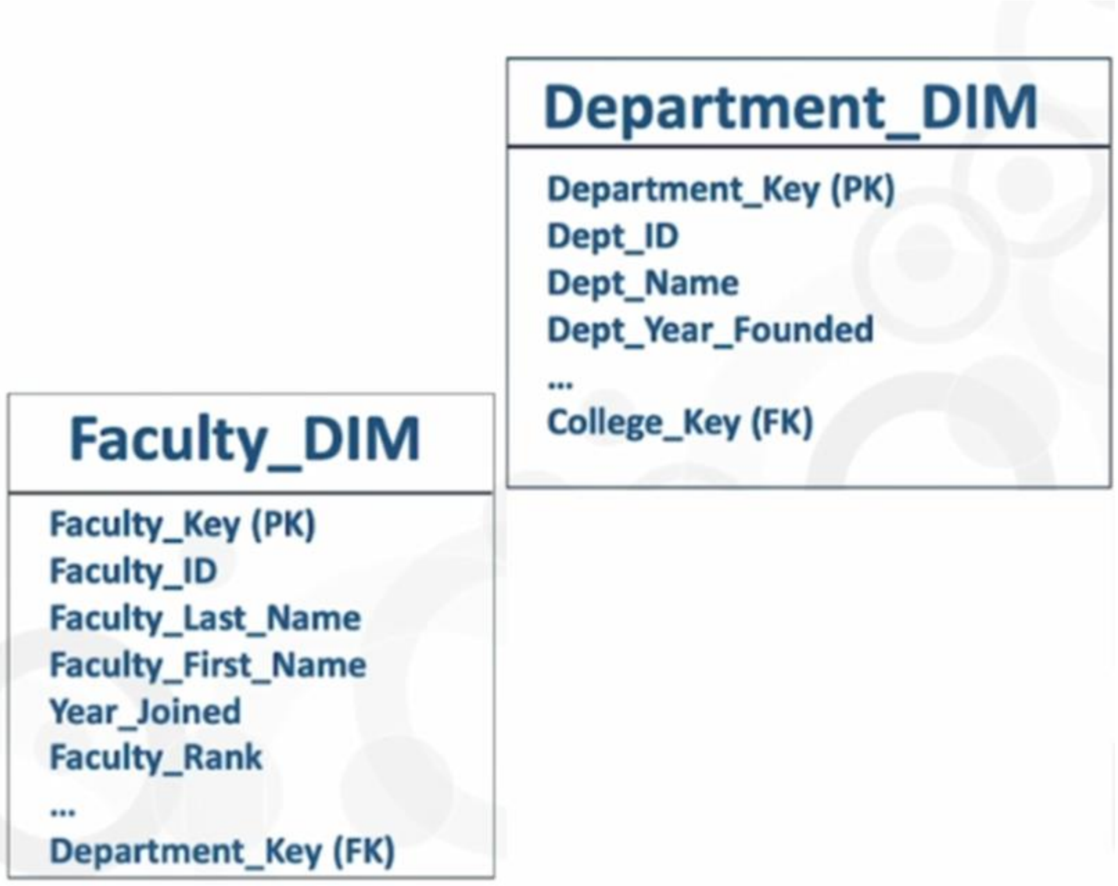
Snowflake schema



Snowflake Schema PK-FK Rules

Every **non-terminal** dimension has:

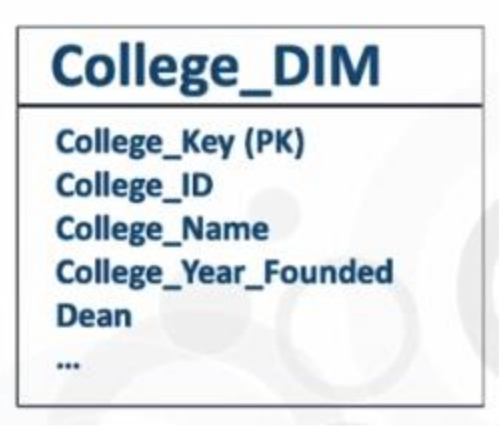
- Primary/surrogate key
- The next-highest level's primary/surrogate key as a foreign key

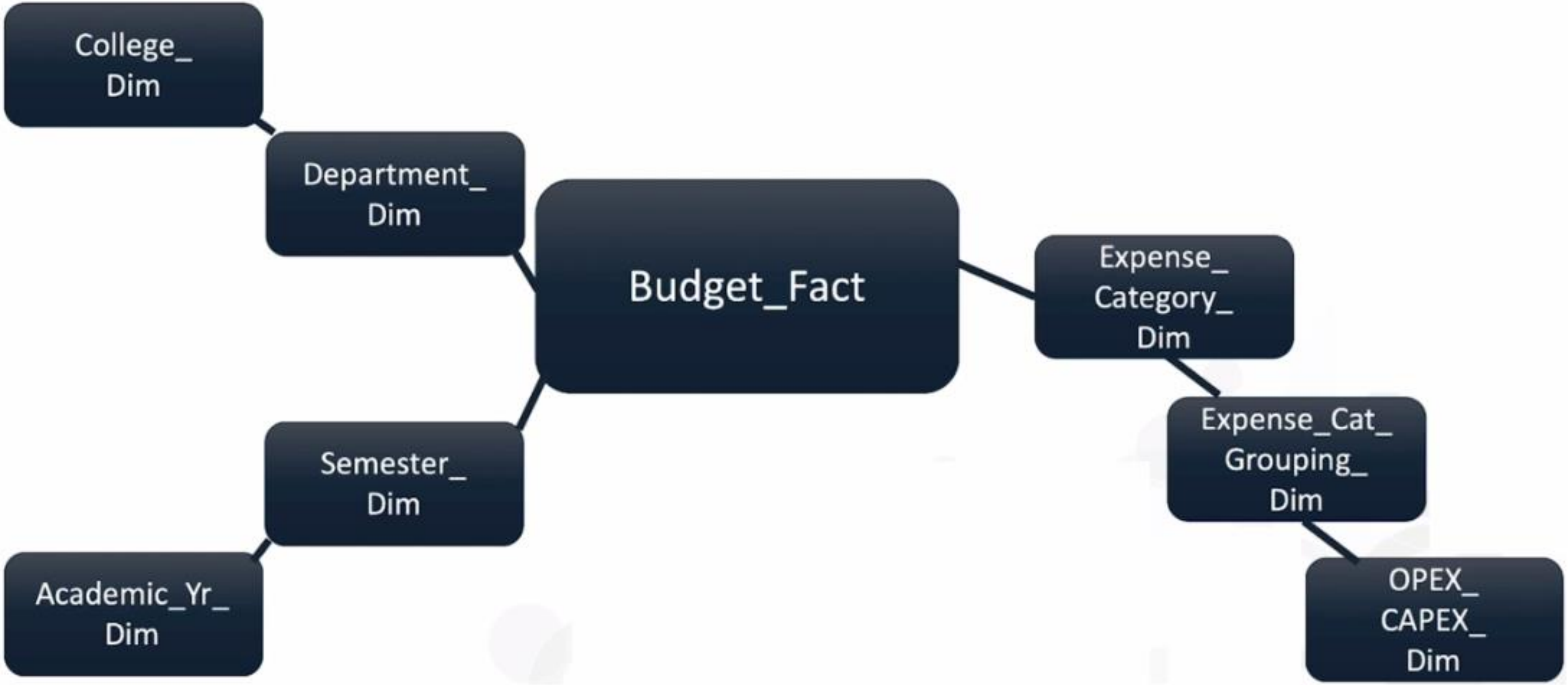


Snowflake Schema PK-FK Rules

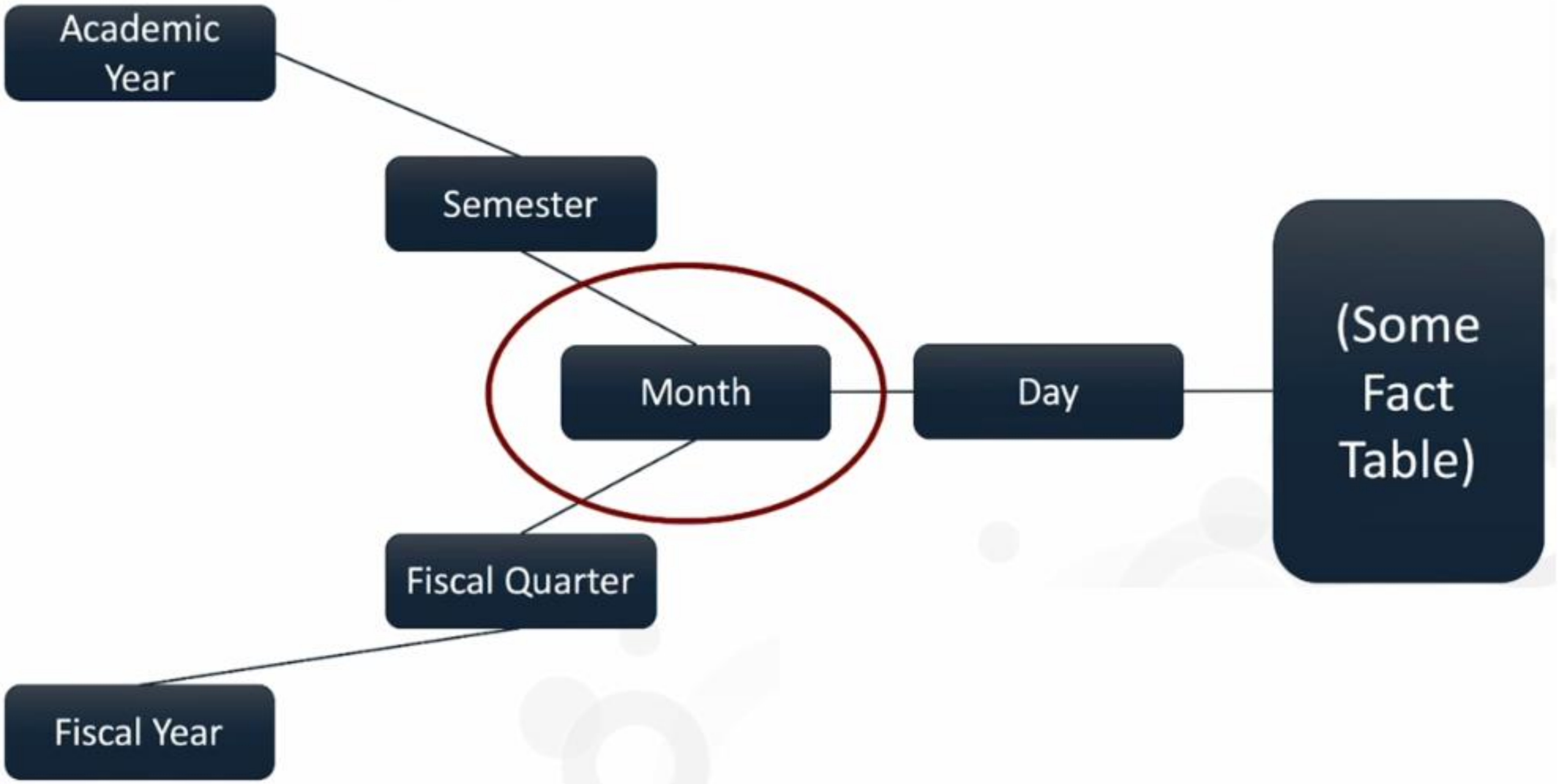
Every **terminal** dimension has:

- Primary/surrogate key
- No hierarchy-based foreign keys (because no higher level)

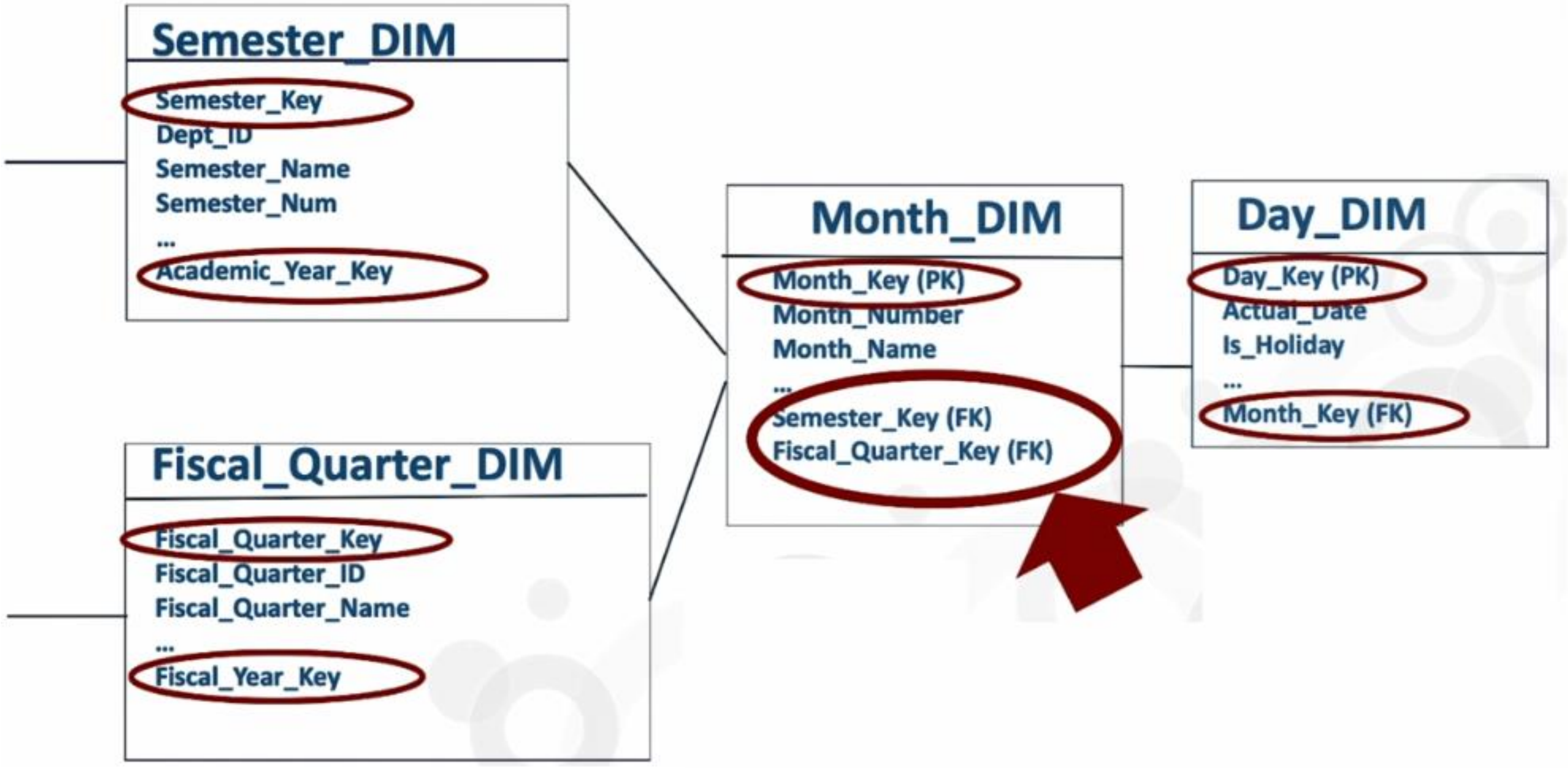


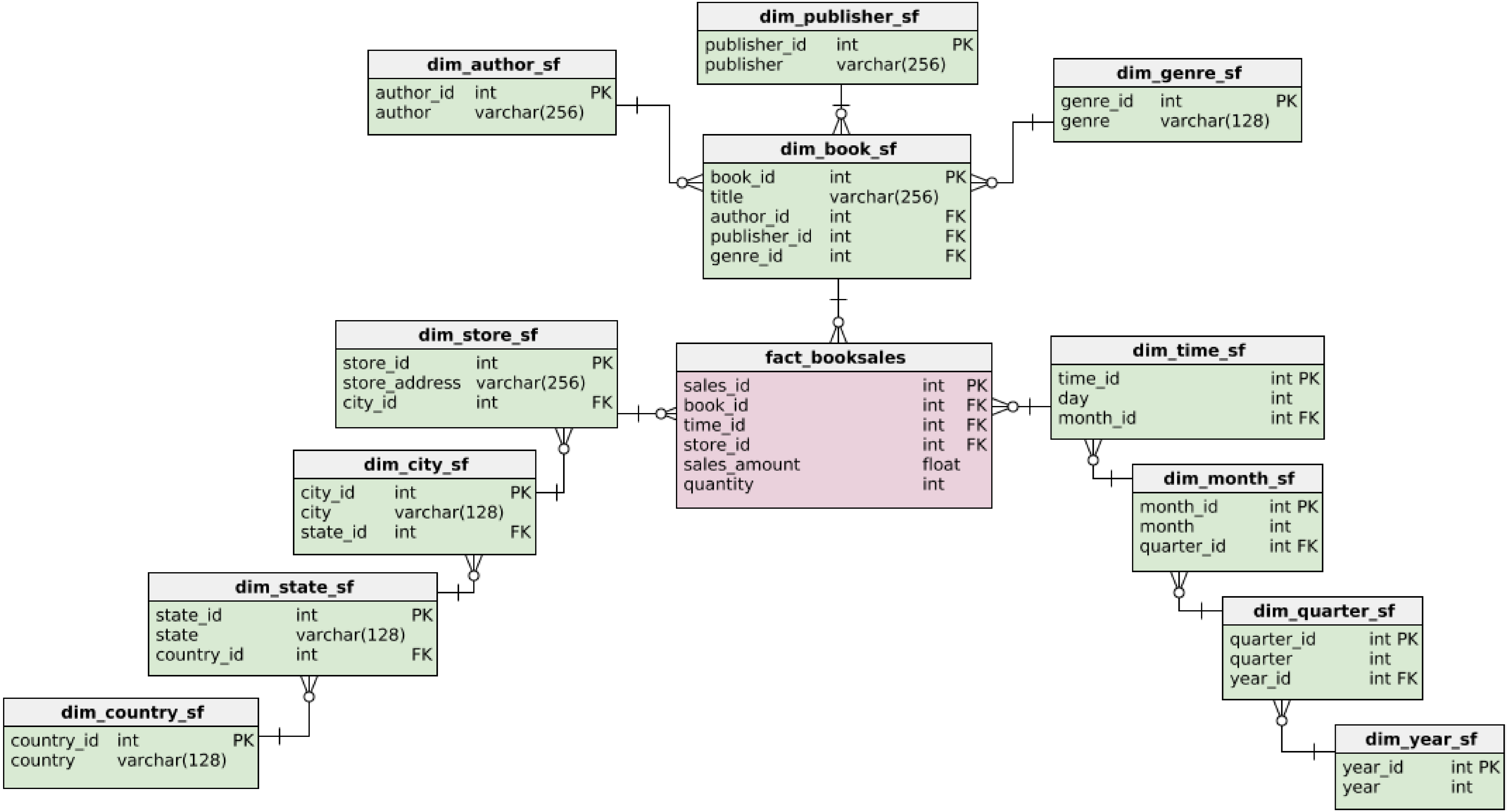

College_
Dim
Department_
Dim
Budget_Fact
Expense_
Category_
Dim
Semester_
Dim
Expense_Cat_
Grouping_
Dim
Academic_Yr_
Dim
OPEX_
CAPEX_
Dim

Snowflake hierarchy with branching



Snowflake hierarchy with branching





dim_publisher_sf
publisher_id int PK
publisher varchar(256)
dim_author_sf
author_id int PK
author varchar(256)
dim_genre_sf
genre_id int PK
genre varchar(128)
dim_book_sf
book_id int PK
title varchar(256)
author_id int FK
publisher_id int FK
genre_id int FK
dim_store_sf
store_id int PK
store_address varchar(256)
city_id int FK
fact_booksales
sales_id int PK
book_id int FK
time_id int FK
store_id int FK
sales_amount float
quantity int
dim_time_sf
time_id int PK
day int
month_id int FK
dim_city_sf
city_id int PK
city varchar(128)
state_id int FK
dim_month_sf
month_id int PK
month int
quarter_id int FK
dim_state_sf
state_id int PK
state varchar(128)
country_id int FK
dim_quarter_sf
quarter_id int PK
quarter int
year_id int FK
dim_country_sf
country_id int PK
country varchar(128)
dim_year_sf
year_id int PK
year int

Snowflake vs Star

Star Schema	Snowflake Schema
All dimensions along a given hierarchy in one dimension table	Each dimension/dimensional level in its own table
One level away from fact table along each hierarchy	One or more levels away from fact table along each hierarchy
With one fact table usually resembles a star	With one fact table usually resembles a snowflake

Snowflake vs Star

Star Schema	Snowflake Schema
Overall fewer database joins for drilling up/down	Overall more database joins for drilling up/down
Database primary->foreign key relationships straightforward	Database primary->foreign key relationships more complex
Typically more database storage needed for dimensional data	Typically less database storage needed for dimensional data
Denormalized dimensional table data	Denormalization is less than in star schema

Thanks