INFORMATION TECHNOLOGY UNIVERSITY

DATA SCIENCE LAB

Dimensional Modeling

CS 537- Big Data Analytics

Dr. Faisal Kamiran

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

Dimension

- Dimensions can consist of multiple hierarchies

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

1 Hierarchy

3 levels of dimensions

We will use this to explain the difference between star and snowflake schemas

Implementing Different Schemas

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

- Star Schema
- Snowflake Schema

These schemas differ on the basis of how they represent the different hierarchies in dimensional tables

Star Schema

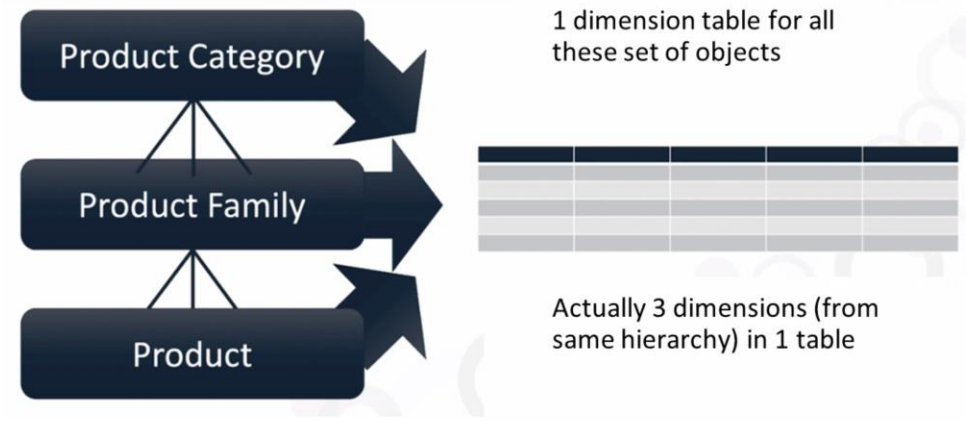
Star Schema is the simplest style of data mart schema.

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



This highlights the main difference with snowflake schema

Star Schema

Faculty_DIM

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
...

Another example, depicting the use of primary keys as well
Here, only one table will be used to store all the hierarchy levels.

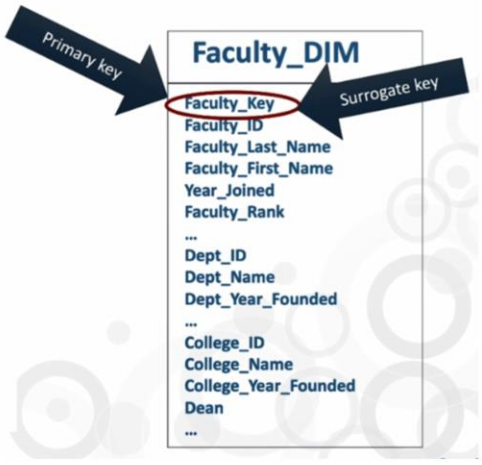
Star Schema

Faculty_DIM

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
...

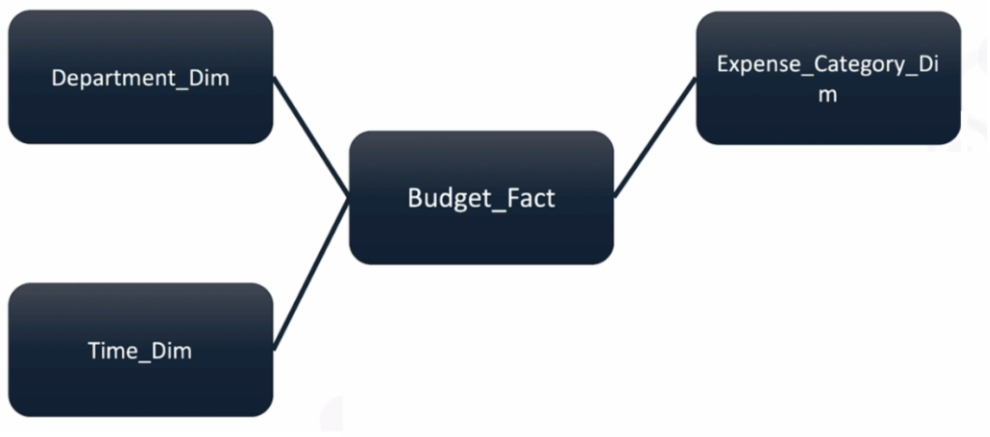
Highlighted natural keys. Not recommended to choose them as the primary key

Star Schema



DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

As previously discussed, create a surrogate key and use that as the primary key


Department_Dim
Expense_Category_Dim
Budget_Fact
Time_Dim

Benefits

- Denormalized
- Simplifies queries
- Fast aggregations

Drawbacks

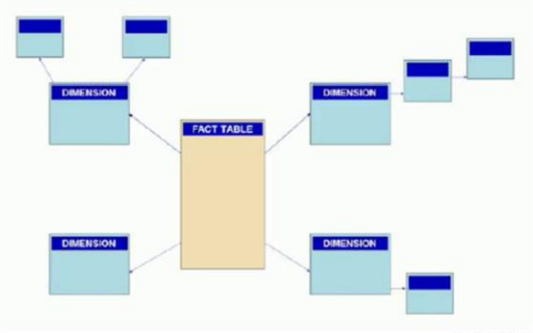
- Issues that come with denormalization
- Data Integrity
- Decrease query flexibility

Snowflake Schema

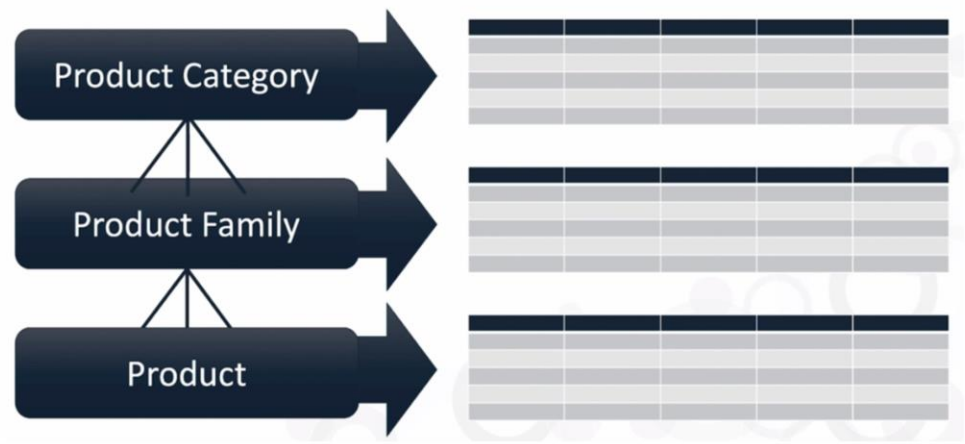
Logical arrangement of tables in a multidimensional database represented by centralized fact tables which are connected to multiple dimensions.

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."

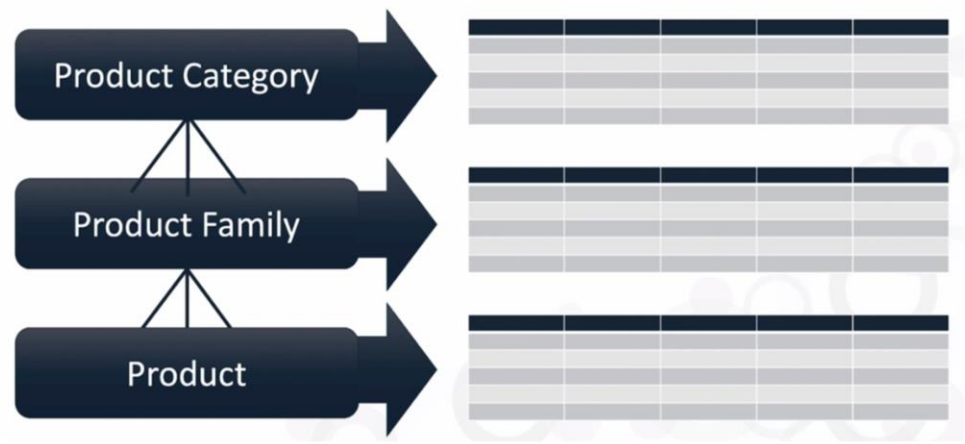


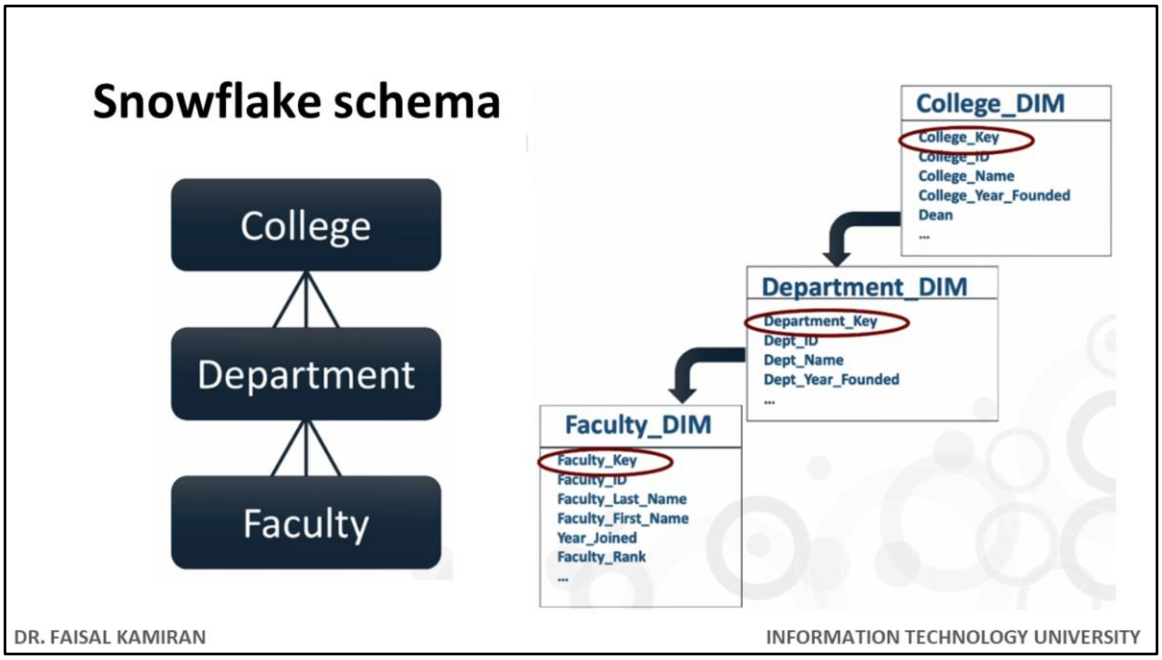
Snowflake schema



Snowflake schema

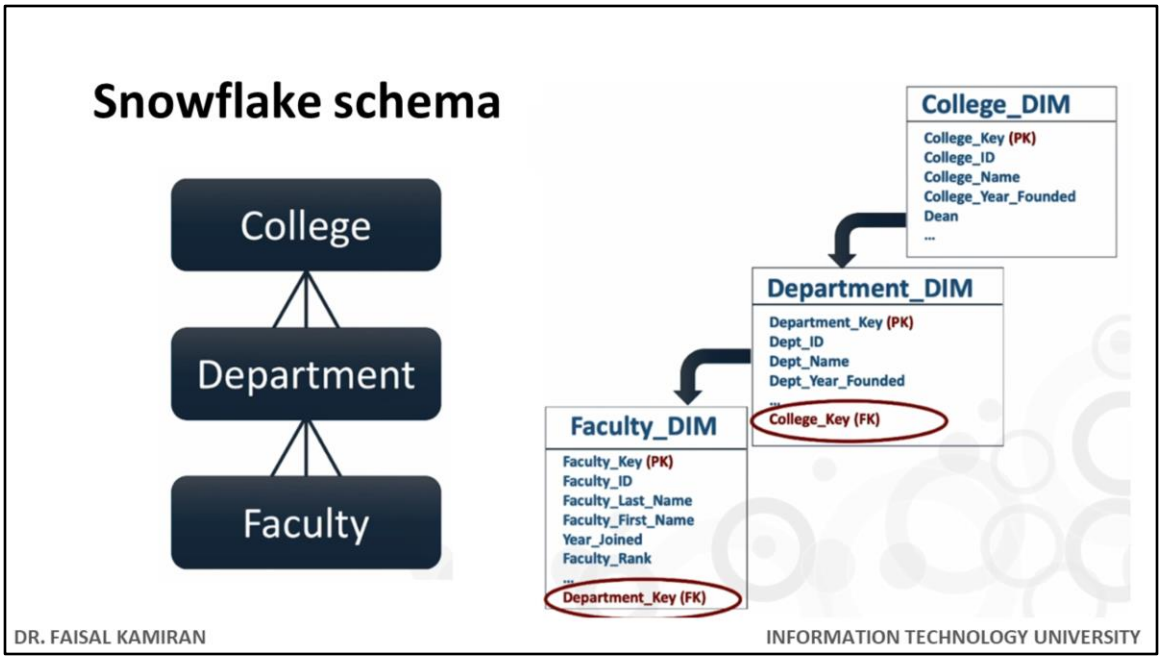
Separate table for each dimension in the hierarchy



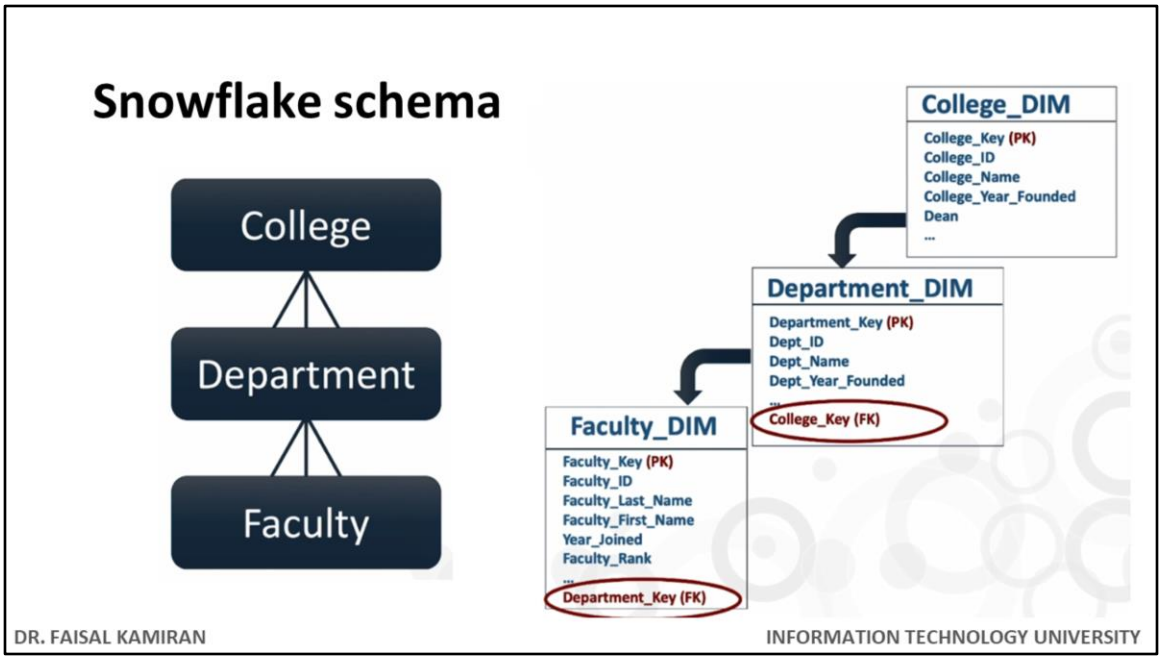


In snowflake schema, instead of storing the dimension in a single denormalized table, multiple partially normalized tables are created, which store different levels of hierarchy.

Foreign keys link the tables together. Here, department key is used as foreign key in the faculty_dim and college_key in department_dim.



In snowflake schema, instead of storing the dimension in a single denormalized table, multiple partially normalized tables are created, which store different levels of hierarchy.



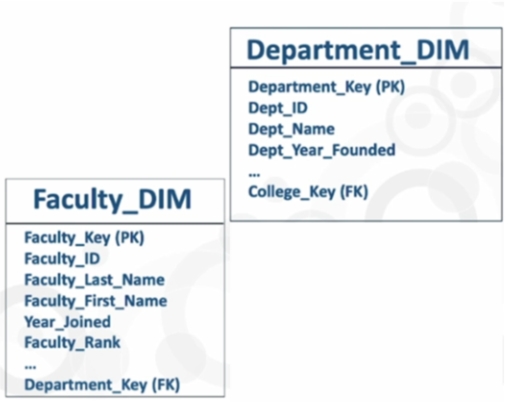
Foreign keys link the tables together. Here, department key is used as foreign key in the faculty_dim and college_key in department_dim.

No foreign keys in terminal tables (College_DIM in this example)

Snowflake Schema PK-FK Rules

Every **non-terminal** dimension has:

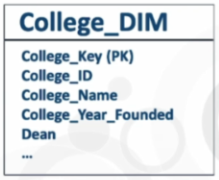
- Primary/surrogate key
- The next-highest level's primary/surrogate key as a foreign key

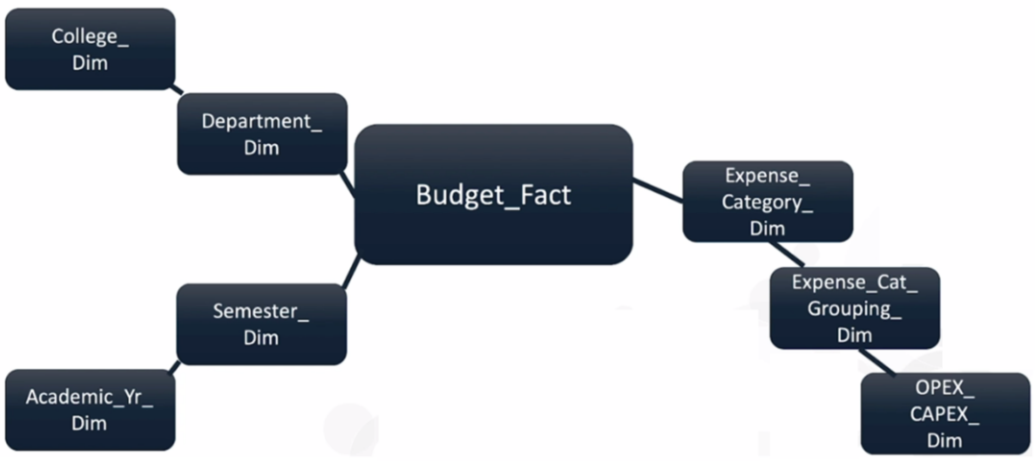


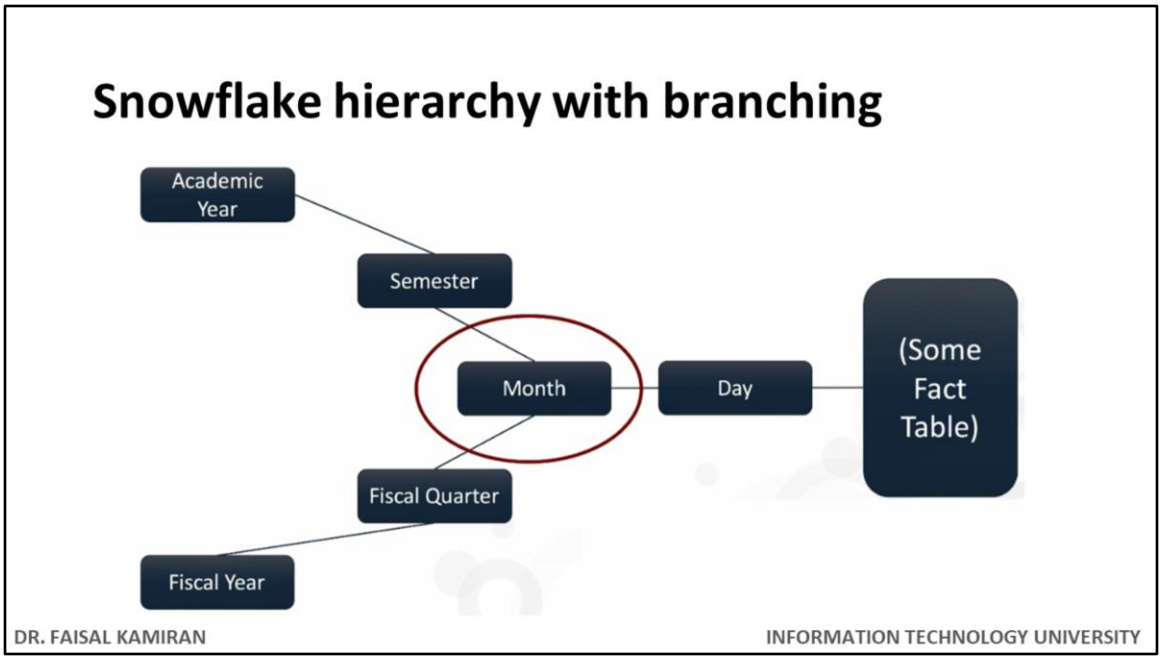
Snowflake Schema PK-FK Rules

Every **terminal** dimension has:

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

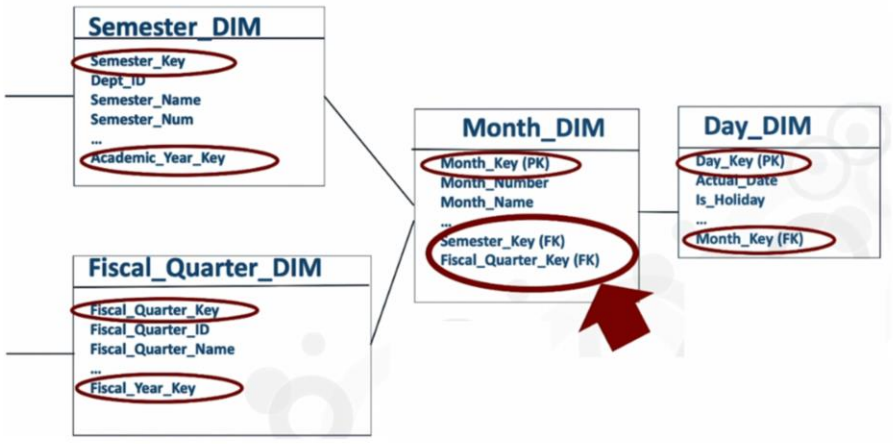



College_ Dim
Department_ Dim
Budget_Fact
Expense_ Category_ Dim
Expense_Cat_ Grouping_ Dim
OPEX_ CAPEX_ Dim
Semester_ Dim
Academic_Yr_ Dim



Here, we have a fact table which also has some form of a time dimension.
Months can fall in an academic calendar and also in a fiscal calendar. So we have a split here

Snowflake hierarchy with branching



Represent this in data with two foreign keys in the month dimension

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
Only 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

Types of Fact Tables

- Transaction fact tables
- Periodic snapshot fact tables
- Accumulating snapshot fact tables
- Factless fact tables

Types of Fact Tables

Fact Table Type	Usage
Transaction	Record facts (measurements) from transactions
Periodic snapshot	Track a given measurement at regular intervals
Accumulating snapshoot	Track the progress of business processes through formally defined stages
Factless	Record Occurrence of a transaction that has no measurements

DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

A transaction fact table is one in which we will store facts or measurements, that occur in our source systems, at an appropriate level of detail

A periodic snapshot fact table is one in which we will track specific things, which we want to measure at regular intervals. Kind of like periodic readings of a thing

Accumulating snapshot related to the periodic snapshot in a way that it stores a snapshot at a point in time but it is used to track a well-defined business process

The factless fact table (oddly named) is used to just record the occurrence of a thing which has no facts/measurements

Transaction Fact Table

- Each row in a transaction fact table corresponds to an event in space and time
- Grain is the individual transaction
- Literally: A table where we store our facts from transactions
- Each event is stored in the fact table only once
- Heart of dimensional models

Heart of dimensional models: Most of the dimensional models are of this type

Transaction Fact Table

Example

"Students are required to pay tuition fees for the university"

We want to track:

- Which students paid the fees
- The amount paid by each student
- Date of payment

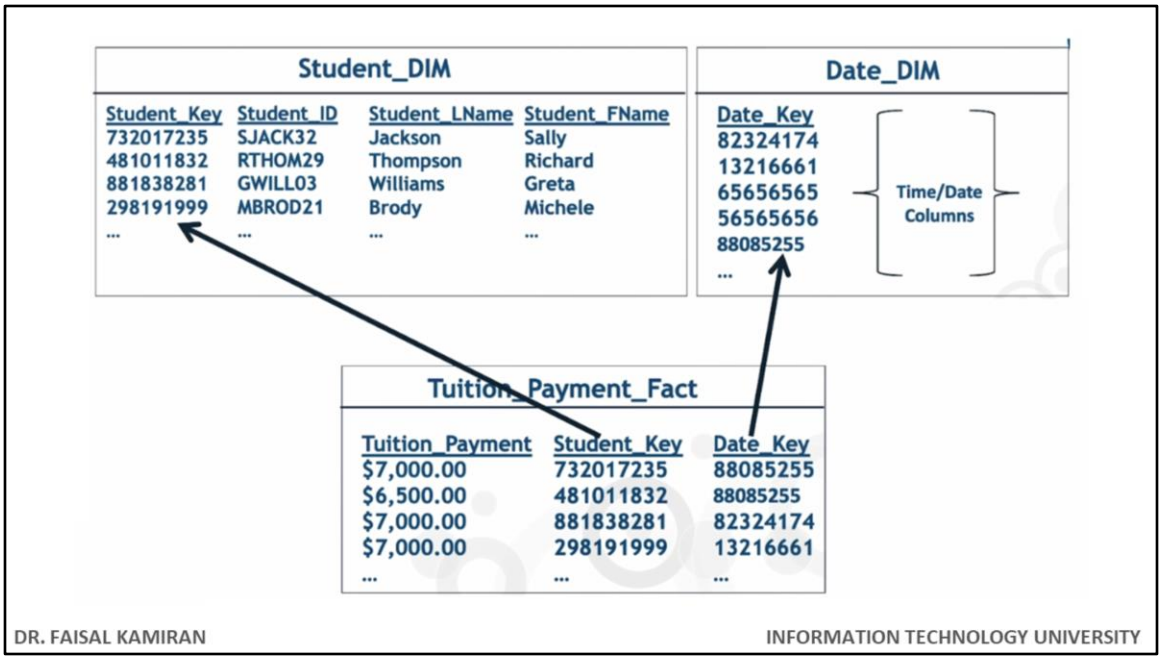
Transaction Fact Table

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
...	...	...

DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

Each individual tuition payment transaction is stored


Student_DIM
Student_Key Student_ID Student_LName Student_FName
732017235 SJACK32 Jackson Sally
481011832 RTHOM29 Thompson Richard
881838281 GWILL03 Williams Greta
298191999 MBROD21 Brody Michele
... ...
Date_DIM
Date_Key
82324174
13216661
65656565
56565656
88085255
...
Time/Date Columns
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
... ...

Periodic Snapshot Fact Table

- Record aggregated measurements of transactions over a period
- Period may be a day, week or month etc.
- Grain is the period
- Transactions are recorded regularly at each cycle of the defined period

A row in a *periodic snapshot fact table* summarizes many measurement events occurring over a standard period, such as a day, a week, or a month.

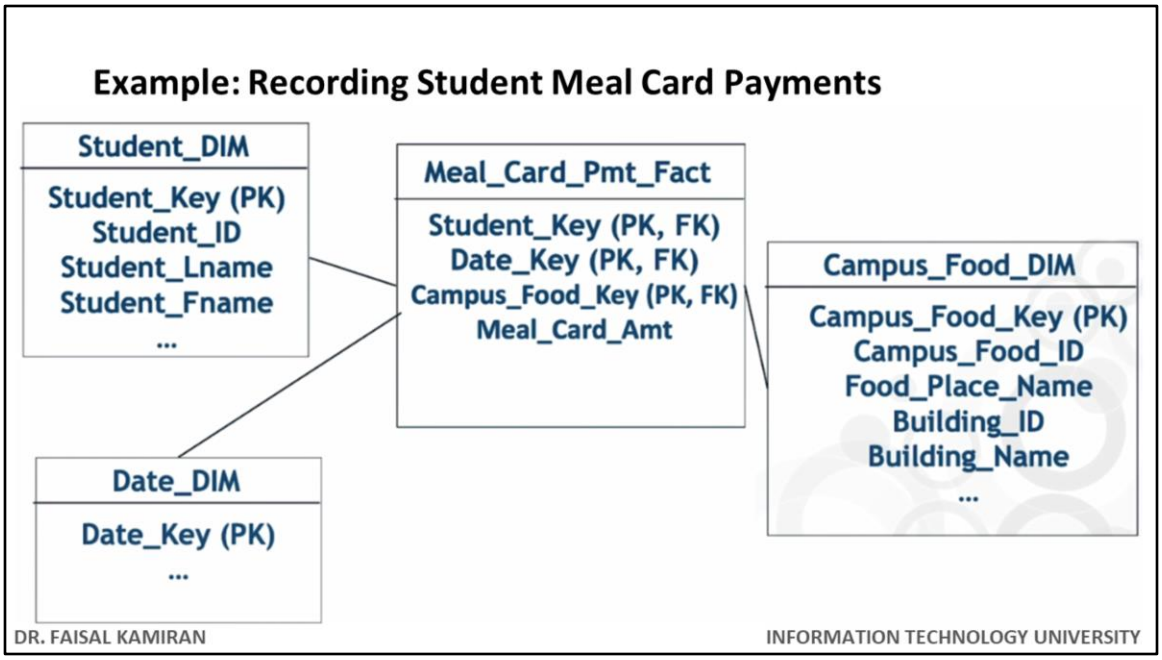
Periodic Snapshot Fact Table

Example

"Each student is assigned a meal card by their universities, which can be used to purchase meals from the university cafeteria."

We want to track the transactions made with these meal cards:

- The student who made the transaction
- Date of transaction
- Campus (The university has multiple campuses)
- Amount paid in the transaction



This dimensional model stores the meal transactions made by students with their university assigned meal cards

Above is a transaction grained fact table representing this. Each time a student makes a payment with his meal card, it will record who the student was, what date the transaction occurred and at what campus did the transaction occur. E.g. if a student spent 500 Rs., then a new row will be created in the fact table

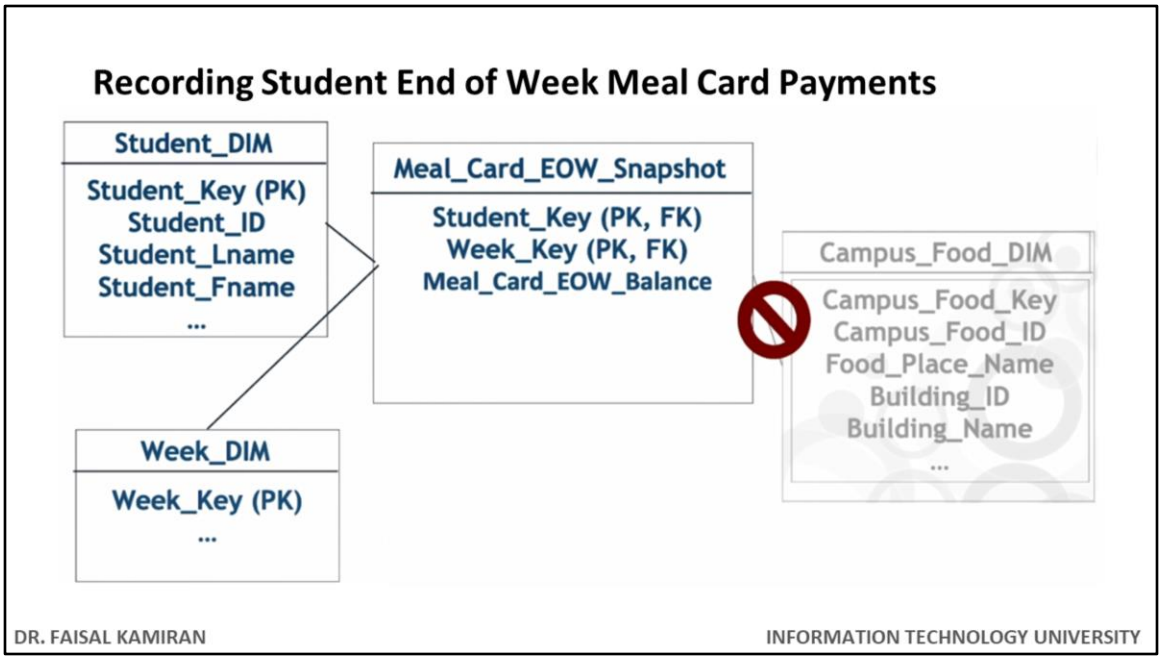
Periodic Snapshot Fact Table

Objective

Track and analyze **end-of-week** meal account balances throughout the semester

This can be done with the regular transaction grained fact table, but the analytic queries will be complex and compute intensive

We will create a periodic snapshot fact table, which will summarize the transactions occurring each week



This fact table will store the end of week meal balance of by a given student at the end of a given week.

The campus_food dimension is not needed because for this analysis, we only need to analyze the balance for students, regardless of campus location

Periodic Snapshot Fact Table

For certain types of business questions, periodic snapshot fact tables are more easily structured and provide more direct access

Accumulating Snapshot Fact Table

- Used to track the progress of a business process through formally defined stages
- Measure elapsed time spent in each phase
- Include both the completed and in-progress phases
- Can also track other measures (in addition to time) as process proceeds
- Introduces concept of relationships from a fact table back to a single dimension table

DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

A row in an *accumulating snapshot fact table* summarizes the measurement events occurring at predictable steps between the beginning and the end of a process. Pipeline or workflow processes, such as order fulfillment or claim processing, that have a defined start point, standard intermediate steps, and defined end point can be modeled with this type of fact table.

Include both the completed and in-progress phases: So that we can track at a particular row that at which processes have completed

Accumulating Snapshot Fact Table

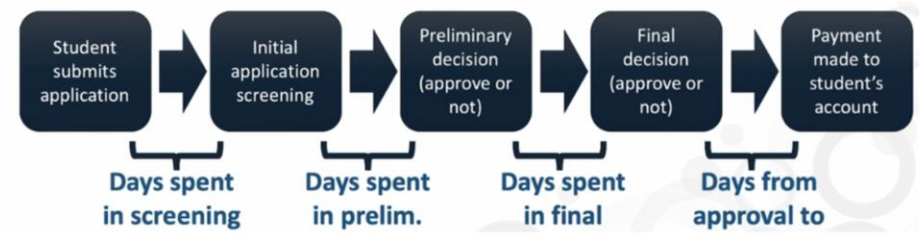
Example

"Student financial aid application process"

The application process has multiple stages

We want to track the time (number of days) spent on each stage

Example: Student Financial Aid Application Process



DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

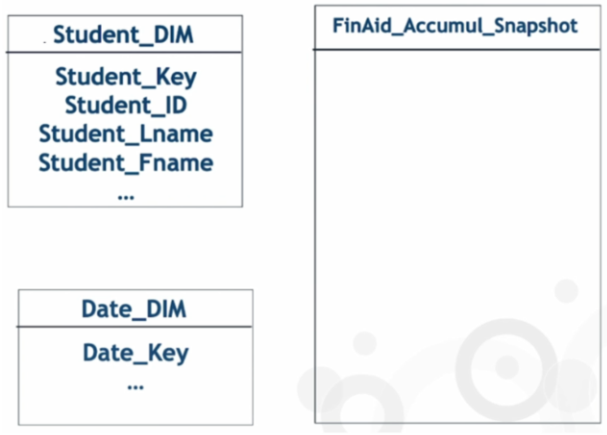
We will be recording the number of days spent in each phase

Example: Student Financial Aid Application Process

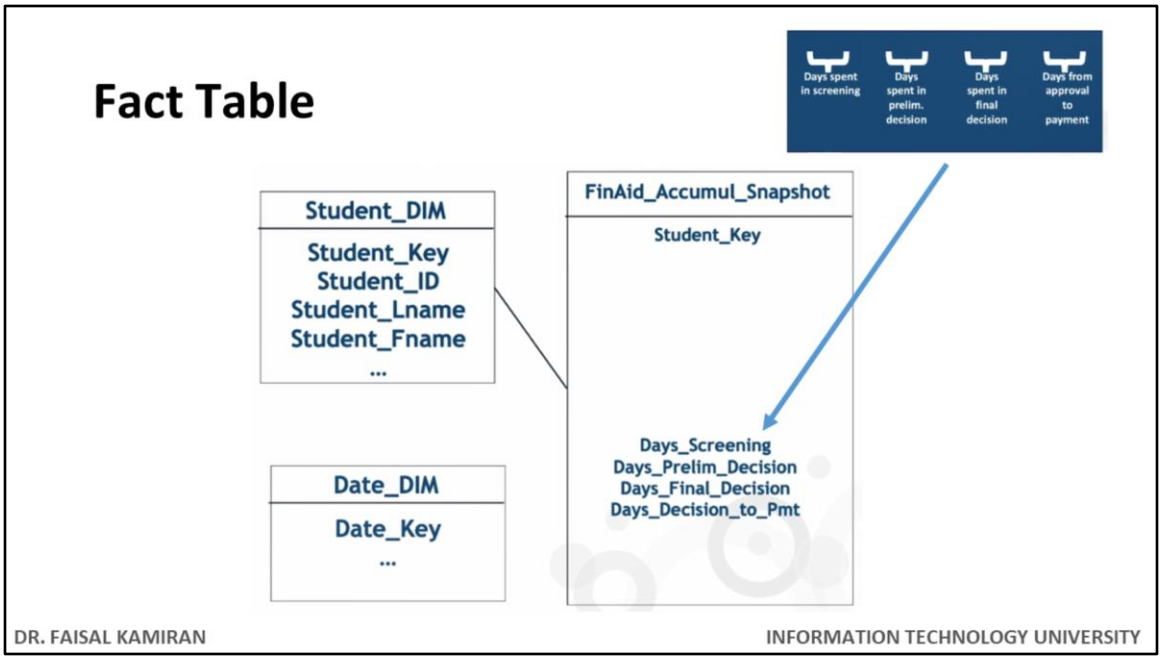
Dimensions for this process

- Student
- Time (Specific day a process begins)

Example: Fact and dimension tables

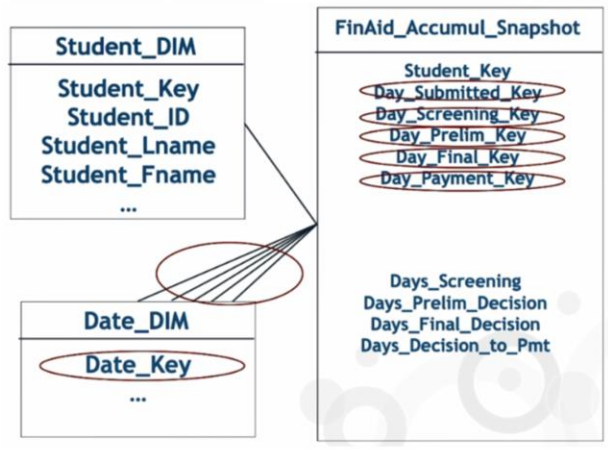


Dimensions for the student who applied for the aid, date
Accumulating snapshot fact table recording the progress of each phase



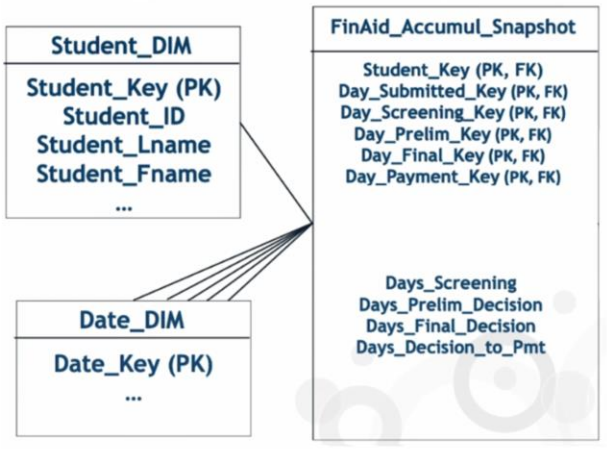
Facts table will contain the actual measurements, which are the days spent at each phase

Multiple links to data dimension



Multiple links to the date dimension for each phase. Notice that the date foreign keys in the fact table are named differently

Accumulating Snapshot Fact Table



As usual, the primary key of the fact table will be a composite primary key of all the dimension foreign keys

Factless Fact Table

Record the *occurrence* of events which have no numerical results associated with them

Factless Fact Table

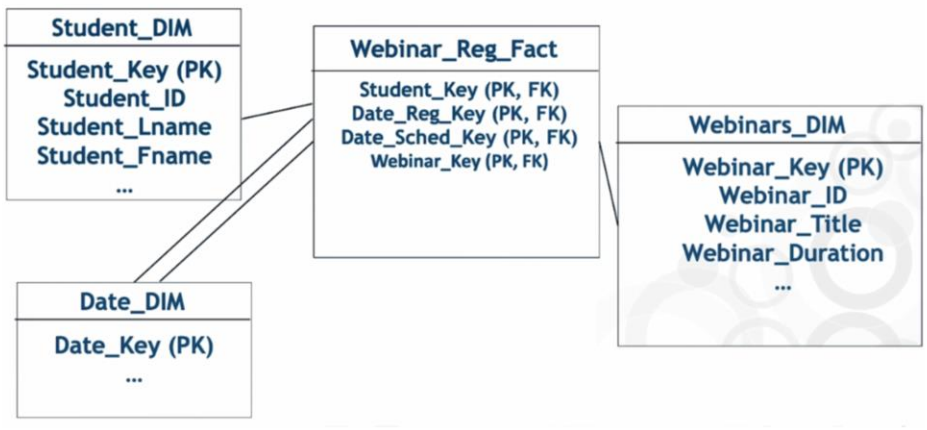
Example

"Students can register for online webinar throughout the semester"

We want to track:

- Which students register
- Which webinar they register for
- Date of registration
- Scheduled date of webinar

Factless Fact Table



DR. FAISAL KAMIRAN INFORMATION TECHNOLOGY UNIVERSITY

No facts in the fact table. With this dimensional model, we can answer questions like

how many students registered for webinars in semester 2?

Essential Rules of Dimensional Modeling

(Recommended by Kimball)

- Load detailed atomic data into dimensional structures
- Structure dimensional models around business processes.
- Ensure that every fact table has an associated date dimension table.
- Ensure that all facts in a single fact table are at the same grain or level of detail.

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

Essential Rules of Dimensional Modeling

(Recommended by Kimball)

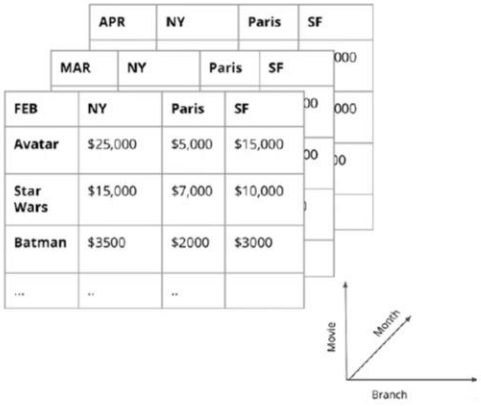
- Resolve many-to-many relationships in fact tables.
- Resolve many-to-one relationships in dimension tables.
- Make certain that dimension tables use a surrogate key.
- Create conformed dimensions to integrate data across the enterprise.
- Continuously balance requirements and realities to deliver a DW solution that is accepted by business users and that supports their decision-making.

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

OLAP Cubes

OLAP Cubes

- An OLAP cube is an aggregation of a fact metric on a number of dimensions
- E.g. Movie, Branch, Month
- Easy to communicate to business users
- Common OLAP operations include: Rollup, drill-down, slice, & dice



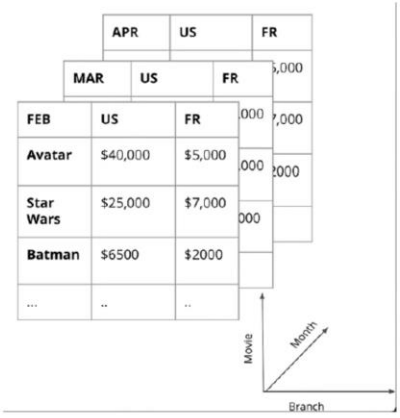
APR	NY	Paris	SF

MAR	NY	Paris	SF

FEB	NY	Paris	SF
Avatar	$25,000	$5,000	$15,000
Star Wars	$15,000	$7,000	$10,000
Batman	$3500	$2000	$3000
...	..	..	

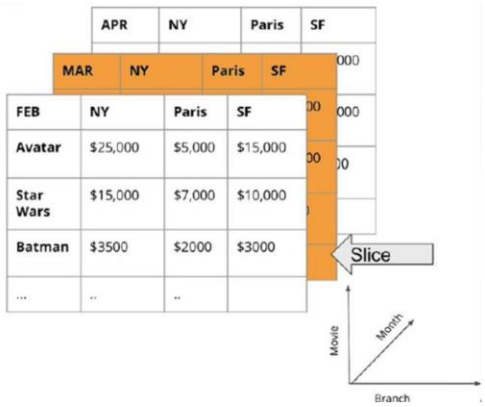
OLAP Cubes Operations: Roll-up & Drill Down

- **Roll-up:** Sum up the sales of each city by Country: e.g. US, France (less columns in branch dimension)
- **Drill-Down:** Decompose the sales of each city into smaller districts (more columns in branch dimension)
- The OLAP cubes should store the finest grain of data (atomic data), in case we need to drill-down to the lowest level, e.g. Country -City - District - Street etc..



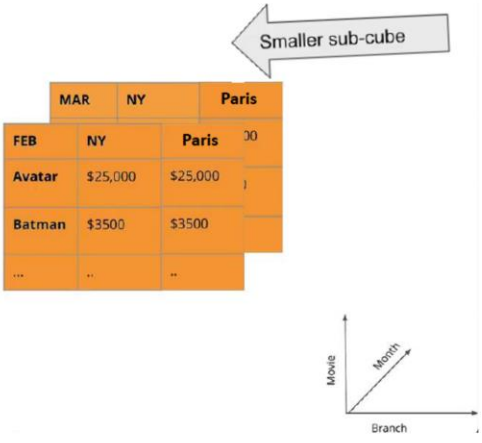
OLAP Cubes Operations: Slice

- Reducing N dimensions to N-1 dimensions by restricting one dimension to a single value
- E.g. month='MAR'



OLAP Cubes Operations: Dice

- Same dimensions but computing a sub-cube by restricting some of the values of the dimensions
- E.g. month in ['FEB', 'MAR'], movie in ['Avatar', 'Batman'] and branch in ['NY', 'Paris']



OLAP Cubes query optimization

- Business users will typically want to slice, dice, rollup and drill-down all the time
- Each such combination will potentially go through all the facts table (suboptimal)

OLAP Cubes query optimization

- The **"GROUP by CUBE (movie , branch, month)"** will make **one pass through the facts** table and will aggregate all possible combinations of groupings, of lengths 0, 1, 2 and 3 e.g:
  - Total revenue
  - Revenue by movie
  - Revenue by branch
  - Revenue by month
  - Revenue by movie, branch
  - Revenue by branch, month
  - Revenue by movie, month
  - Revenue by movie, branch, month
- Saving/Materializing the output of the CUBE operation and using it is usually enough to answer all forthcoming aggregations from business users without having to process the whole facts table again

The Last Mile: Delivering the analytics to users

Data is available...

- In an understandable & performant dimensional model
- With Conformed Dimensions or separate Data Marts
- For users to report and visualize
  - By interacting directly with the model
  - Or in most cases, through a Bl application

The Last Mile: Delivering the analytics to users

OLAP cubes is a very convenient way for slicing, dicing and drilling down

How do we serve these OLAP cubes?

OLAP Cubes technology

Approach 1: **Pre-aggregate** the OLAP Cubes and save them on a special-purpose non-relational database **(MOLAP)**

Approach 2: Compute OLAP Cubes **on the fly** from the existing relational databases where the dimensional model resides **(ROLAP)**

References & Further Reading

- Kimball, The Data Warehouse Toolkit [LINK]
- Inmon, Building the Data Warehouse [LINK]
- Rainardi, Building a Data Warehouse [LINK]