What is Dimensional Model?

A dimensional model is a data structure technique optimized for Data warehousing tools. The concept of Dimensional Modelling was developed by Ralph Kimball and is comprised of "fact" and "dimension" tables.

A Dimensional model is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.

These dimensional and relational models have their unique way of data storage that has specific advantages.

For instance, in the relational mode, normalization and ER models reduce redundancy in data. On the contrary, dimensional model arranges data in such a way that it is easier to retrieve information and generate reports.

Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

Elements of Dimensional Data Model

Fact

Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number

Dimension

Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the 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.

Attributes

The Attributes are the various characteristics of the dimension.

In the Location dimension, the attributes can be

- State
- Country
- Zipcode etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

Fact Table

A fact table is a primary table in a dimensional model.

A Fact Table contains

1. Measurements/facts
2. Foreign key to dimension table

Dimension table

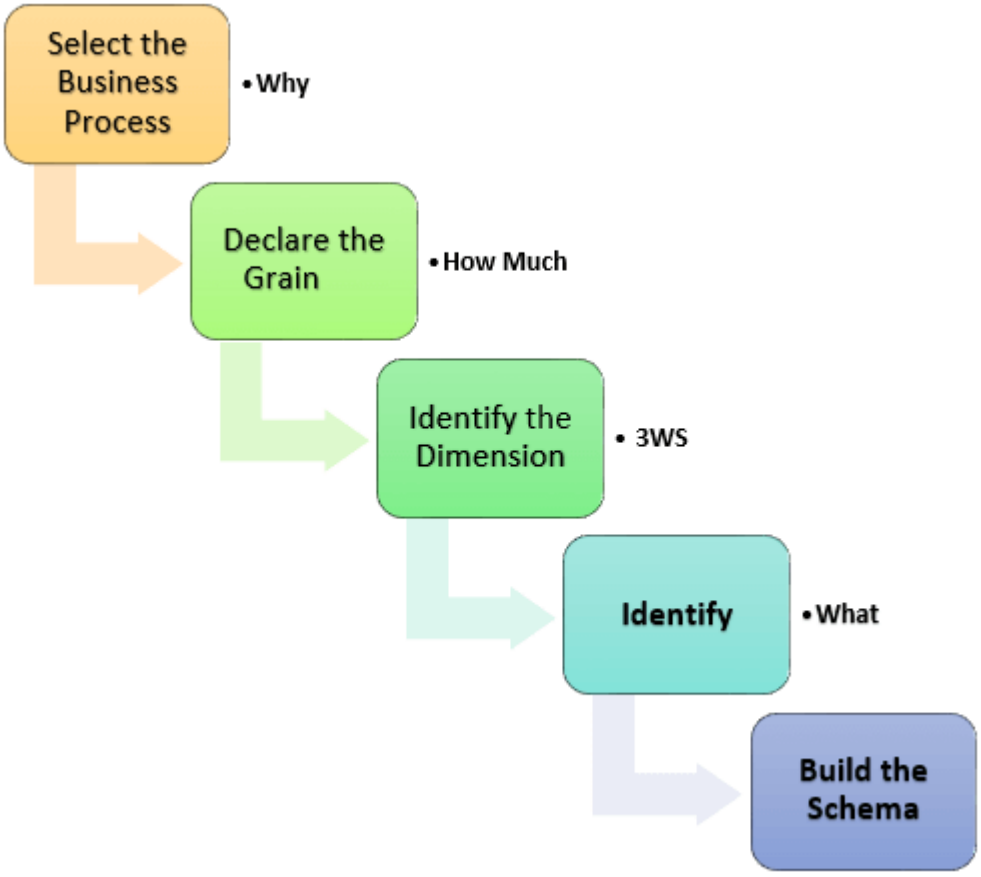
- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes
- No set limit set for given for number of dimensions
- The dimension can also contain one or more hierarchical relationships

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. Identify Dimensions
4. Identify Facts
5. Build Star

The model should describe the Why, How much, When/Where/Who and What of your business process



Step 1) Identify the business process

Identifying the actual business process 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.

To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

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.

During this stage, you answer questions like

1. 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

2. 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
3. How do the above two choices affect the database size?

Example of Grain:

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 3) Identify 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 of Dimensions:

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

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 of Facts:

The 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.

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 two popular schemas

1. **Star Schema**

The star schema architecture is easy to design. It is called a star schema because diagram resembles a star, with points radiating from a centre. The centre of the star consists of the fact table, and the points of the star are dimension tables.

The fact tables in a star schema which is third normal form whereas dimensional tables are de-normalized.

2. **Snowflake Schema**

The snowflake schema is an extension of the star schema. In a snowflake schema, each dimension are normalized and connected to more dimension tables.

Rules for Dimensional Modelling

- Load atomic data into dimensional structures.
- Build dimensional models around business processes.
- Need to 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.
- It's essential to store report labels and filter domain values in dimension tables
- Need to ensure that dimension tables use a surrogate key
- Continuously balance requirements and realities to deliver business solution to support their decision-making

Benefits of dimensional modeling

- Standardization of dimensions allows easy reporting across areas of the business.
- Dimension tables store the history of the dimensional information.
- It allows to introduced entirely new dimension without major disruptions to the fact table.
- Dimensional also to store data in such a fashion that it is easier to retrieve the information from the data once the data is stored in the database.
- Compared to the normalized model dimensional table are easier to understand.
- Information is grouped into clear and simple business categories.
- 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 attributes means.
- Dimensional models are deformalized and optimized for fast data querying. Many relational database platforms recognize this model and optimize query execution plans to aid in performance.
- Dimensional modeling creates a schema which is optimized for high performance. It means fewer joins and helps with minimized data redundancy.
- The dimensional model also helps to boost query performance. It is more de-normalized therefore it is optimized for querying.
- Dimensional models can comfortably accommodate change. Dimension tables can have more columns added to them without affecting existing business intelligence applications using these tables.

Summary:

- A dimensional model is a data structure technique optimized for Data warehousing tools.
- Facts are the measurements/metrics or facts from your business process.
- Dimension provides the context surrounding a business process event.
- The Attributes are the various characteristics of the dimension.
- A fact table is a primary table in a dimensional model.
- A dimension table contains dimensions of a fact.
- There are three types of facts 1. Additive 2. Non-additive 3. Semi- additive.
- Types of Dimensions are Conformed, Outrigger, Shrunken, Role-playing, Dimension to Dimension Table, Junk, Degenerate, Swappable and Step Dimensions.
- Five steps of Dimensional modeling are 1. Identify Business Process 2. Identify Grain (level of detail) 3. Identify Dimensions 4. Identify Facts 5. Build Star
- In Dimensional modeling, there is need to ensure that every fact table has an associated date dimension table.

Difference between ER Modeling and Dimensional Modeling

The entity-relationship model is a method used to represent the logical flow of entities/objects graphically that in turn create a database. It has both logical and physical model. And it is good for reporting and point queries.

Dimensional model is a method in which the data is stored in two types of tables namely facts table and dimension table. It has only physical model. It is good for ad hoc query analysis.

Dimensional modeling is a form of modeling of data that is more flexible for the perspective of user. The ER modeling is for databases that are OLTP databases which uses normalized data using 1st or 2nd or 3rd normal forms.

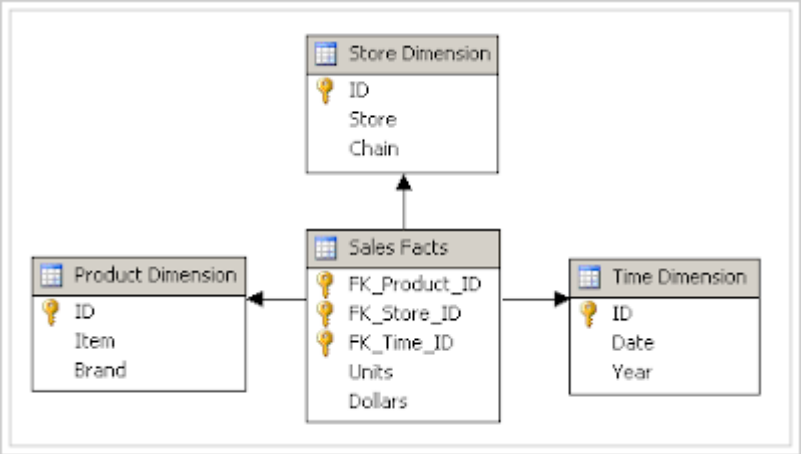
Dimensional Modeling is used in data warehouse that uses 3rd normal form. It contains de-normalized data.

Data Warehouse Dimensional Modelling (Types of Schemas)

There are four types of schemas are available in data warehouse. Out of which the star schema is mostly used in the data warehouse designs. The second mostly used data warehouse schema is snow flake schema. We will see about these schemas in detail.

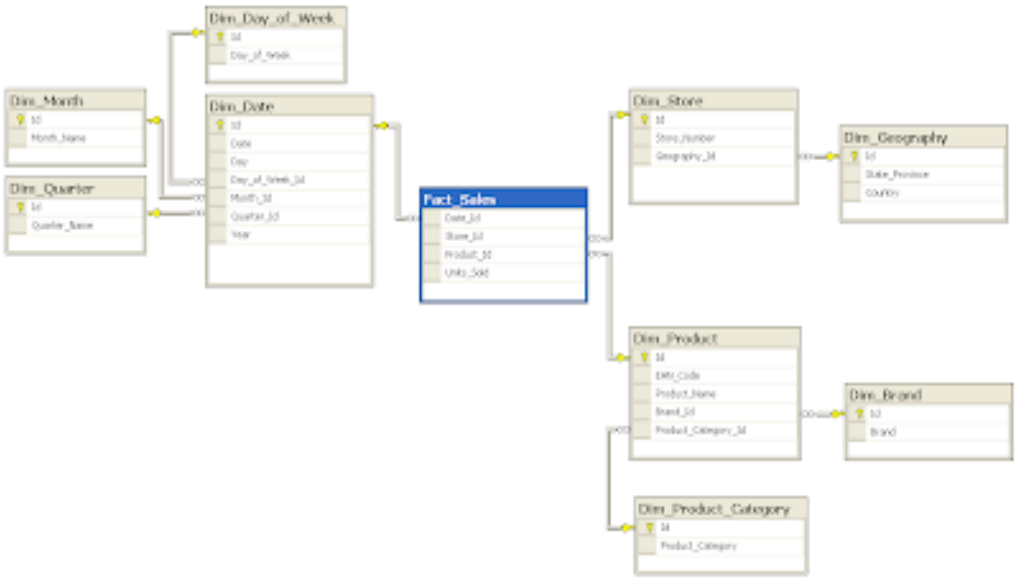
Star Schema

A star schema is the one in which a central fact table is surrounded by de-normalized dimensional tables. A star schema can be simple or complex. A simple star schema consists of one fact table where as a complex star schema have more than one fact table.



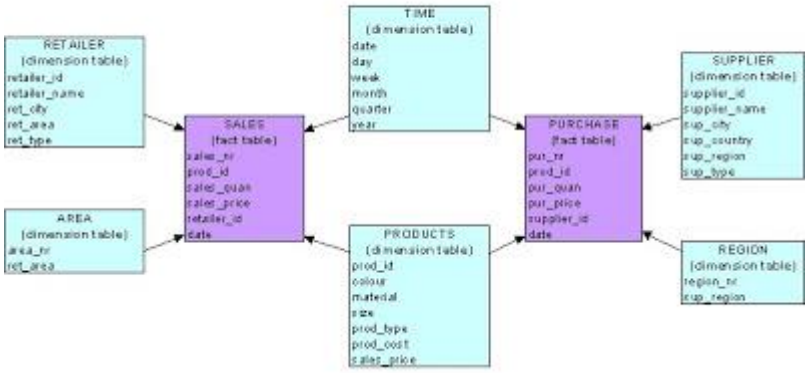
Snow Flake Schema

A snow flake schema is an enhancement of star schema by adding additional dimensions. Snow flake schema is useful when there are low cardinality attributes in the dimensions.



Galaxy Schema

Galaxy schema contains many fact tables with some common dimensions (conformed dimensions). This schema is a combination of many data marts.

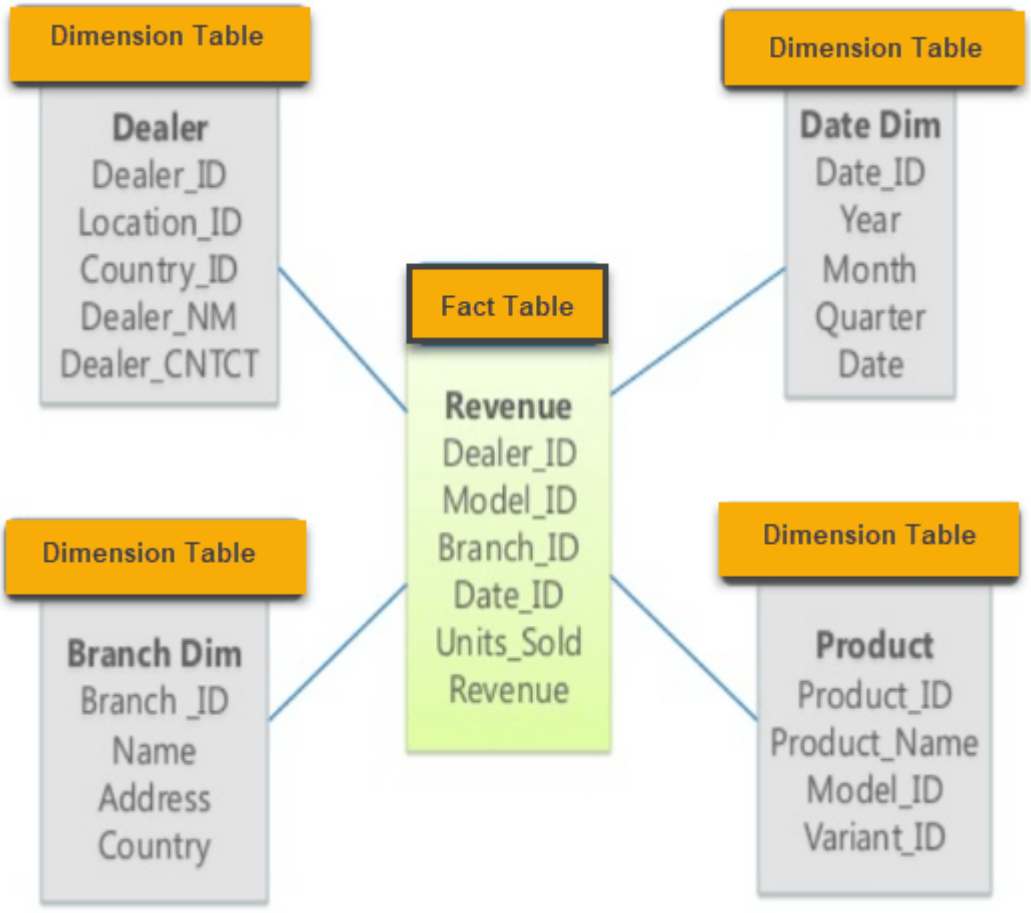


Fact Constellation Schema

The dimensions in this schema are segregated into independent dimensions based on the levels of hierarchy. For example, if geography has five levels of hierarchy like territory, region, country, state and city; constellation schema would have five dimensions instead of one.

What is a Star Schema?

The star schema is the simplest type of Data Warehouse schema. It is known as star schema as its structure resembles a star. In the Star schema, the center of the star can have one fact tables and numbers of associated dimension tables. It is also known as Star Join Schema and is optimized for querying large data sets.



For example, as you can see in the above-given image that fact table is at the center which contains keys to every dimension table like Deal_ID, Model ID, Date_ID, Product_ID, Branch_ID & other attributes like Units sold and revenue.

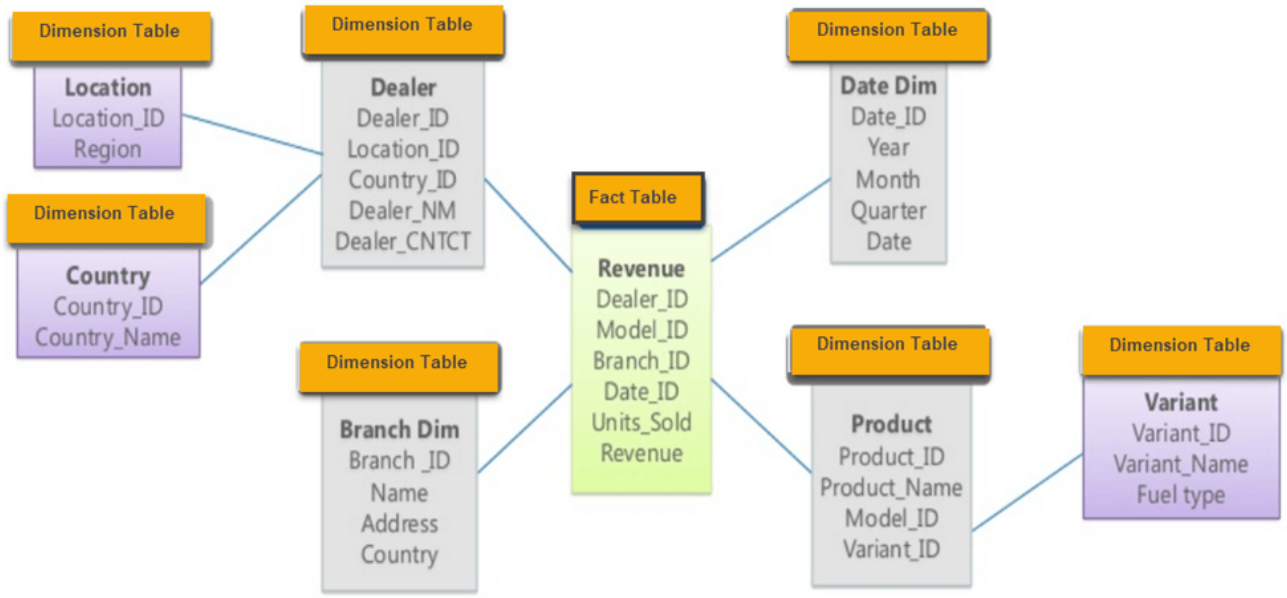
Characteristics of Star Schema:

- Every dimension in a star schema is represented with the only one-dimension table.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a foreign key
- The dimension table are not joined to each other
- Fact table would contain key and measure
- The Star schema is easy to understand and provides optimal disk usage.
- The dimension tables are **not normalized**. For instance, in the above figure, Country_ID does not have Country lookup table as an OLTP design would have.
- The schema is widely supported by BI Tools

What is a 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.

The dimension tables are **normalized** which splits data into additional tables. In the following example, Country is further normalized into an individual table.

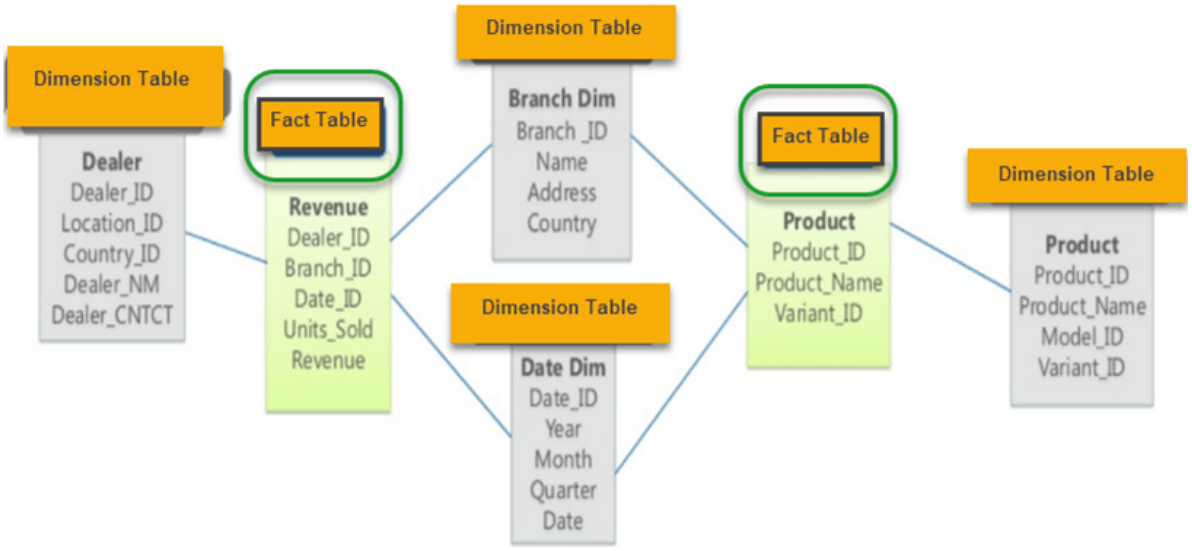


Characteristics of Snowflake Schema:

- The main benefit of the snowflake schema it uses smaller disk space.
- Easier to implement a dimension is added to the Schema
- Due to multiple tables query performance is reduced
- The primary challenge that you will face while using the snowflake Schema is that you need to perform more maintenance efforts because of the more lookup tables.

What is a Galaxy schema?

A Galaxy Schema contains two fact tables that shares dimension tables. It is also called **Fact Constellation Schema**. The schema is viewed as a collection of stars hence the name Galaxy Schema.



As you can see in above figure, there are two facts table

1. Revenue
2. Product.

In Galaxy schema shares dimensions are called Conformed Dimensions.

Characteristics of Galaxy Schema:

- The dimensions in this schema are separated into separate dimensions based on the various levels of hierarchy.
- For example, if geography has four levels of hierarchy like region, country, state, and city then Galaxy schema should have four dimensions.
- Moreover, it is possible to build this type of schema by splitting the one-star schema into more Star schemes.
- The dimensions are large in this schema which is needed to build based on the levels of hierarchy.
- This schema is helpful for aggregating fact tables for better understanding.

Star Vs Snowflake Schema: Key Differences

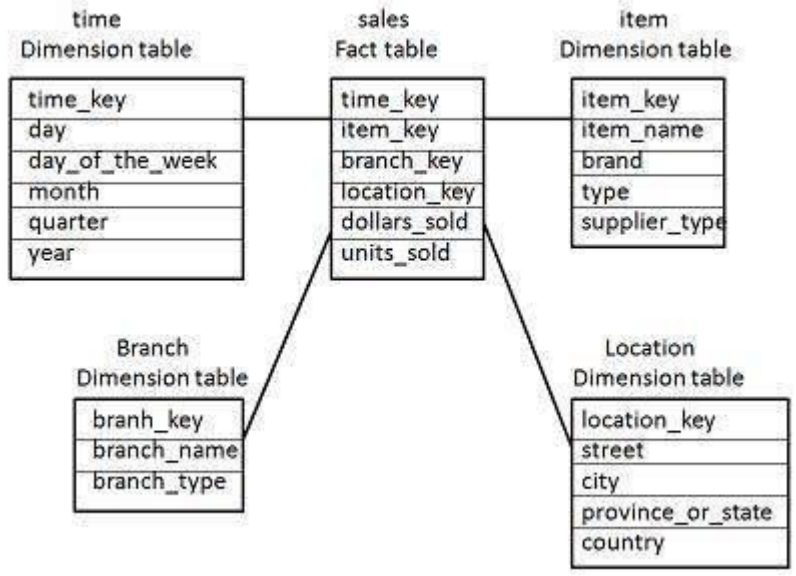
Star Schema	Snow Flake Schema
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
De-normalized Data structure and query also run faster.	Normalized Data Structure.
High level of Data redundancy	Very low-level data redundancy
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions.	The Snow Flake Schema is represented by centralized fact table which unlikely connected with multiple dimensions.

Data Warehousing - Schemas

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

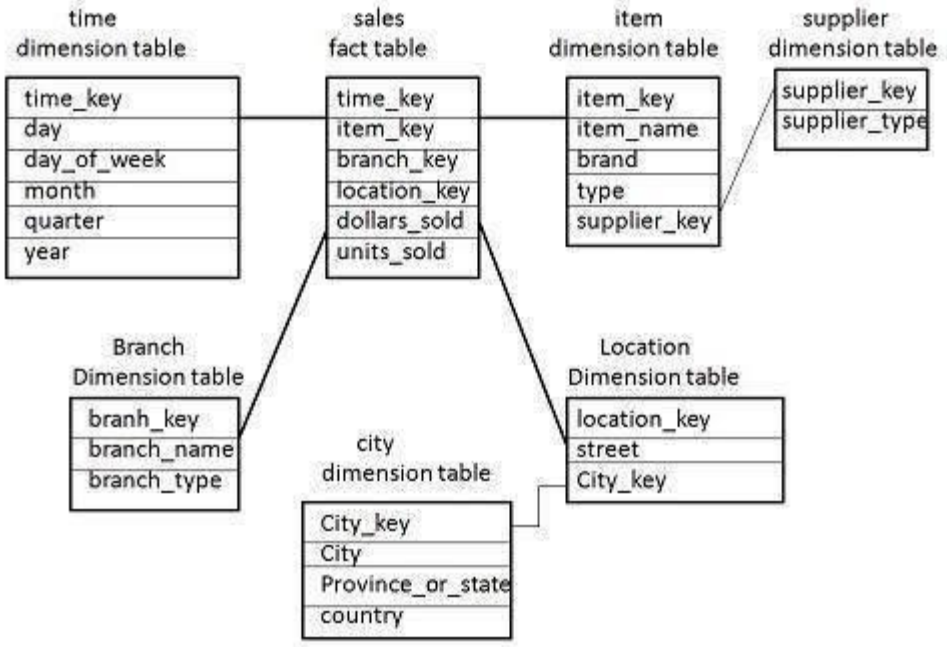


- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note − Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

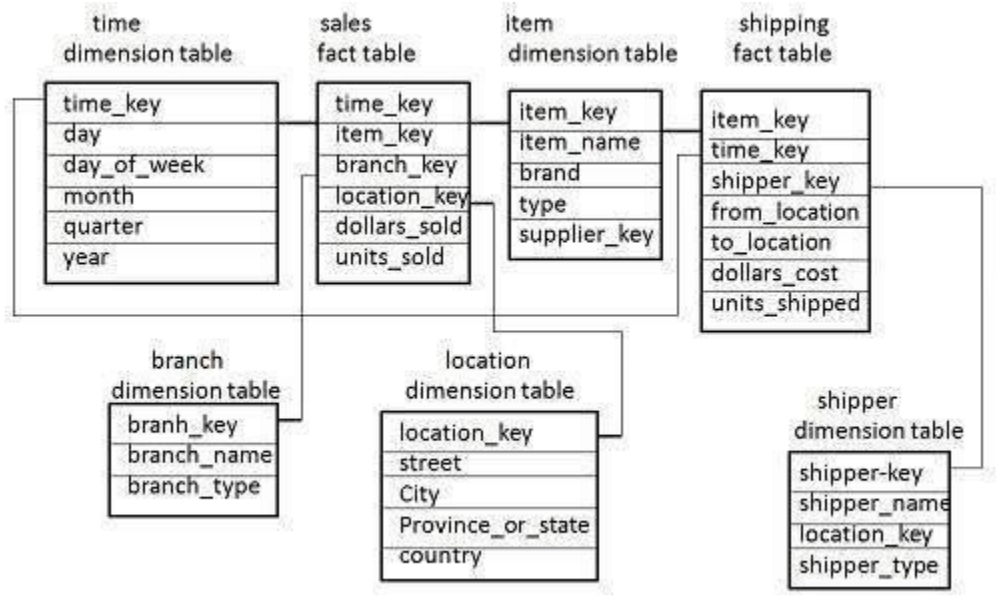


- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note − Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.
- The sales fact table is same as that in the star schema.



- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

```
define cube < cube_name > [ < dimension-list > }: < measure_list >
```

Syntax for Dimension Definition

```
define dimension < dimension_name > as ( < attribute_or_dimension_list > )
```

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows −

```
define cube sales star [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)
```

```
define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city, province or state, country)
```

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows −

```
define cube sales snowflake [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier (supplier key,
supplier type))
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city (city key, city, province or
state, country))
```

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows −

```
define cube sales [time, item, branch, location]:

dollars sold = sum(sales in dollars), units sold = count(*)

define dimension time as (time key, day, day of week, month, quarter, year)
define dimension item as (item key, item name, brand, type, supplier type)
define dimension branch as (branch key, branch name, branch type)
define dimension location as (location key, street, city, province or state,country)
define cube shipping [time, item, shipper, from location, to location]:

dollars cost = sum(cost in dollars), units shipped = count(*)

define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper key, shipper name, location as location in cube
sales, shipper type)
define dimension from location as location in cube sales
define dimension to location as location in cube sales
```

Data Warehousing – OLAP

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This chapter cover the types of OLAP, operations on OLAP, difference between OLAP, and statistical databases and OLTP.

Types of OLAP Servers

We have four types of OLAP servers −

Relational OLAP (ROLAP)

Multidimensional OLAP (MOLAP)

Hybrid OLAP (HOLAP)

Specialized SQL Servers

Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

ROLAP includes the following −

- Implementation of aggregation navigation logic.
- Optimization for each DBMS back end.
- Additional tools and services.

Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allow storing the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

Specialized SQL Servers

Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

OLAP Operations

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations −

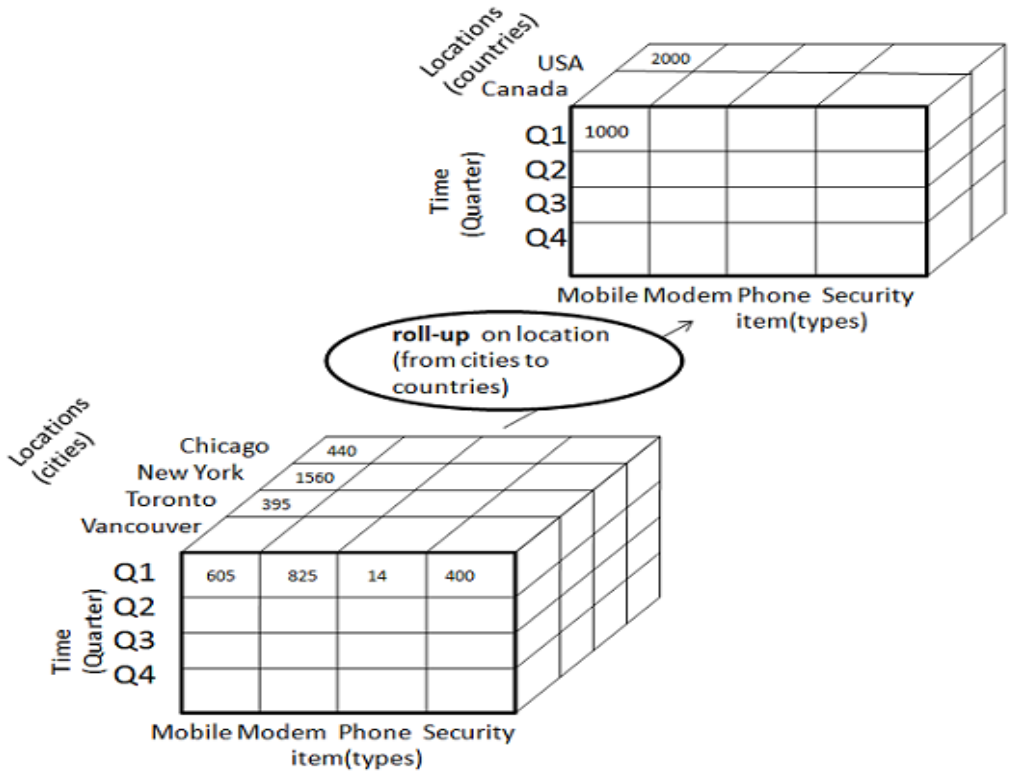
- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up

Roll-up performs aggregation on a data cube in any of the following ways −

- By climbing up a concept hierarchy for a dimension
- By dimension reduction

The following diagram illustrates how roll-up works.



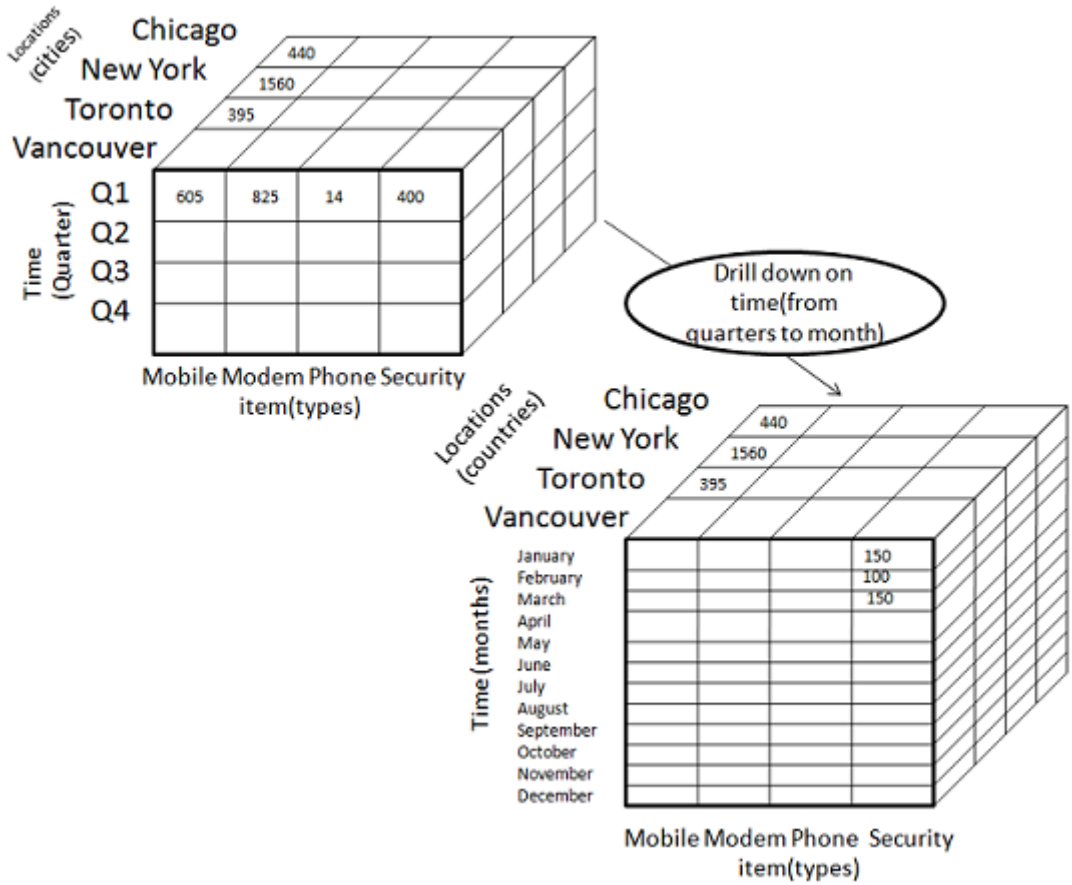
Roll-up is performed by climbing up a concept hierarchy for the dimension location.
Initially the concept hierarchy was "street < city < province < country".
On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
The data is grouped into cities rather than countries.
When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways −

- By stepping down a concept hierarchy for a dimension
- By introducing a new dimension.

The following diagram illustrates how drill-down works –

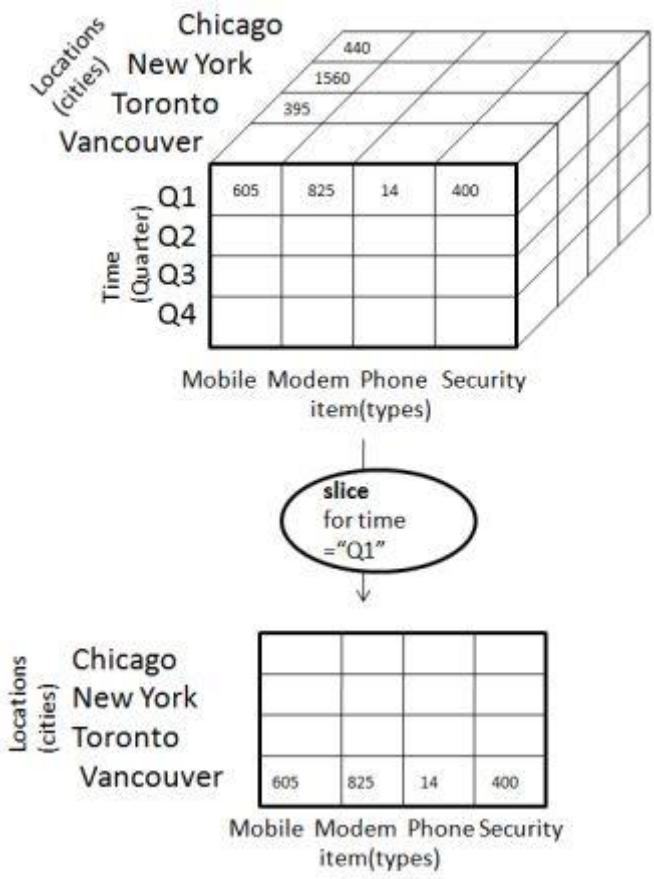


Drill-down is performed by stepping down a concept hierarchy for the dimension time.

Initially the concept hierarchy was "day < month < quarter < year."
On drilling down, the time dimension is descended from the level of quarter to the level of month.
When drill-down is performed, one or more dimensions from the data cube are added.
It navigates the data from less detailed data to highly detailed data.

Slice

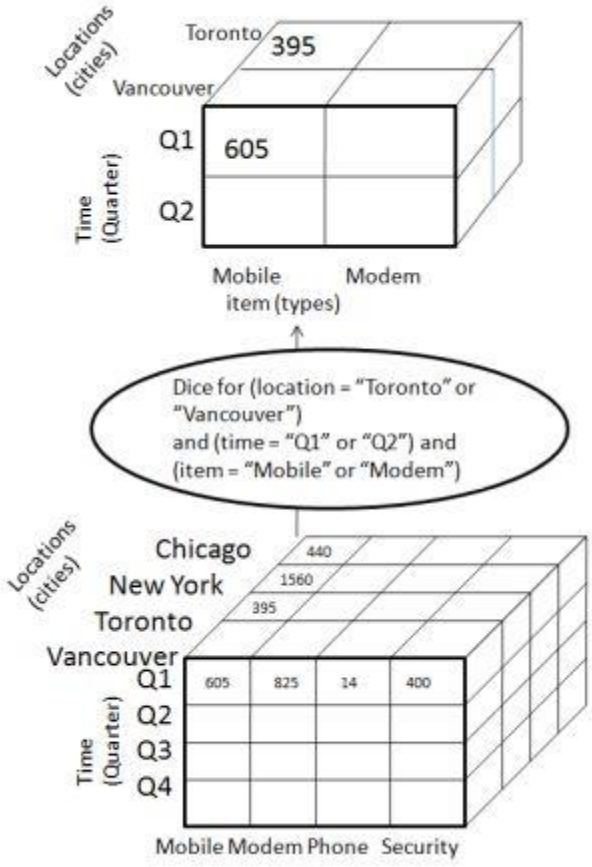
The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



- Here Slice is performed for the dimension "time" using the criterion time = "Q1".
- It will form a new sub-cube by selecting one or more dimensions.

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

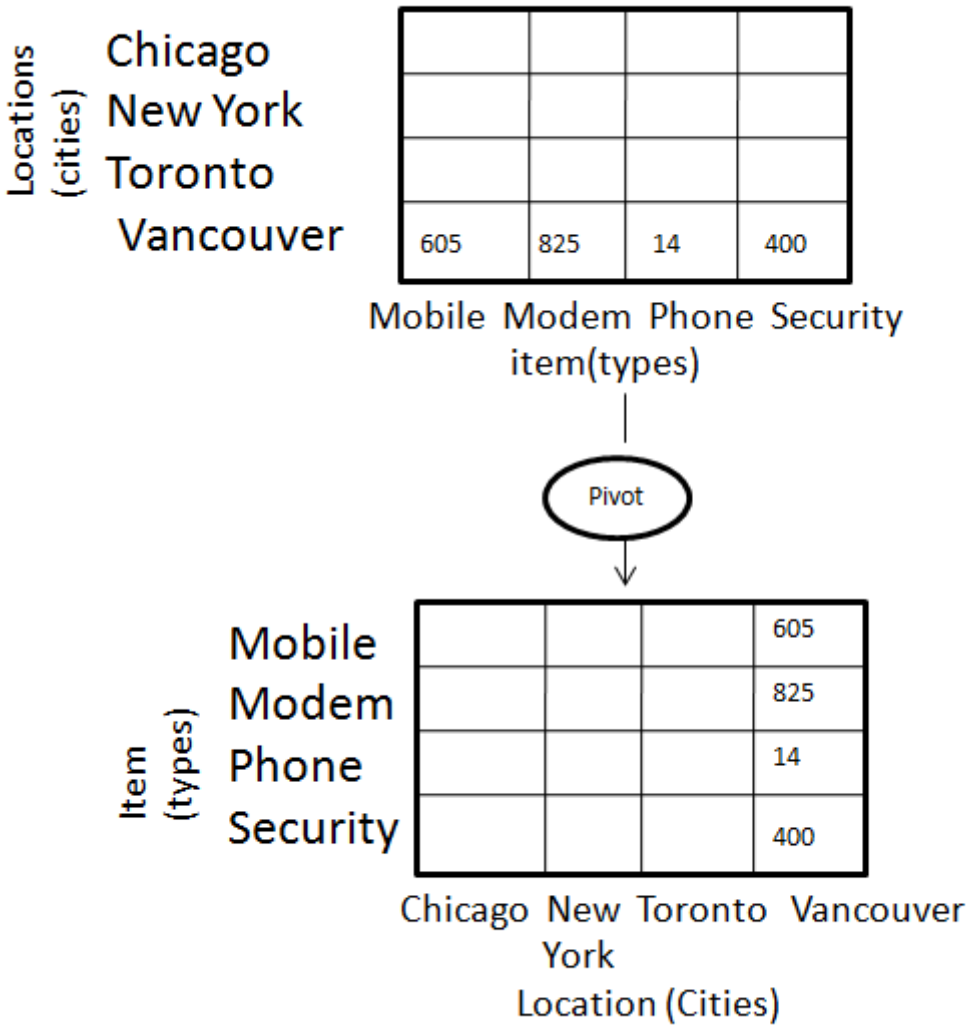


The dice operation on the cube based on the following selection criteria involves three dimensions.

- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



OLAP vs OLTP

Sr.No.	Data Warehouse (OLAP)	Operational Database (OLTP)
1	Involves historical processing of information.	Involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	Useful in analyzing the business.	Useful in running the business.
4	It focuses on Information out.	It focuses on Data in.

5	Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.	Based on Entity Relationship Model.
6	Contains historical data.	Contains current data.
7	Provides summarized and consolidated data.	Provides primitive and highly detailed data.
8	Provides summarized and multidimensional view of data.	Provides detailed and flat relational view of data.
9	Number or users is in hundreds.	Number of users is in thousands.
10	Number of records accessed is in millions.	Number of records accessed is in tens.
11	Database size is from 100 GB to 1 TB	Database size is from 100 MB to 1 GB.
12	Highly flexible.	Provides high performance.