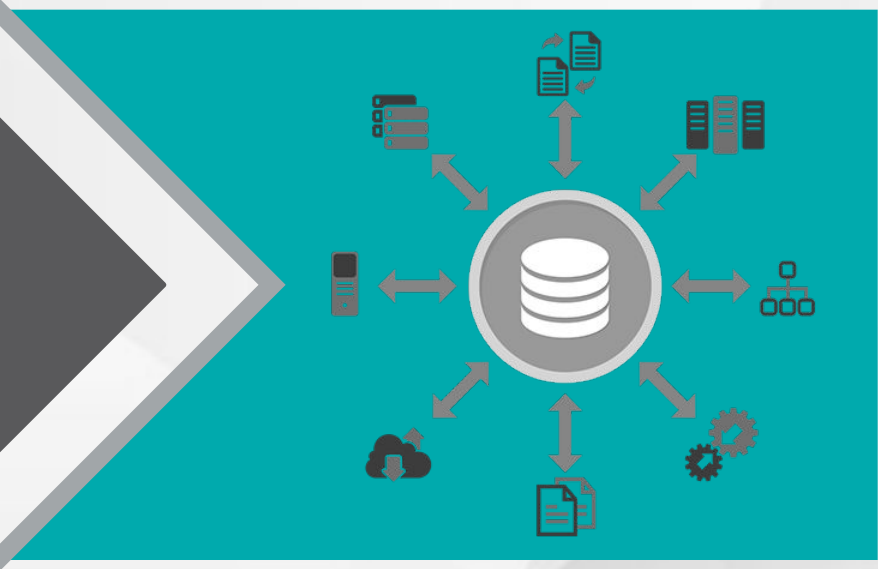
203105453 – Data Mining & Business Intelligence

Unit-3

Data Warehousing and Online Analytical Processing



Prof. Prashant V. Sahatiya

8155812895

prashant.sahatiya270187@paruluniversity.ac.in

Parul® University

Outline

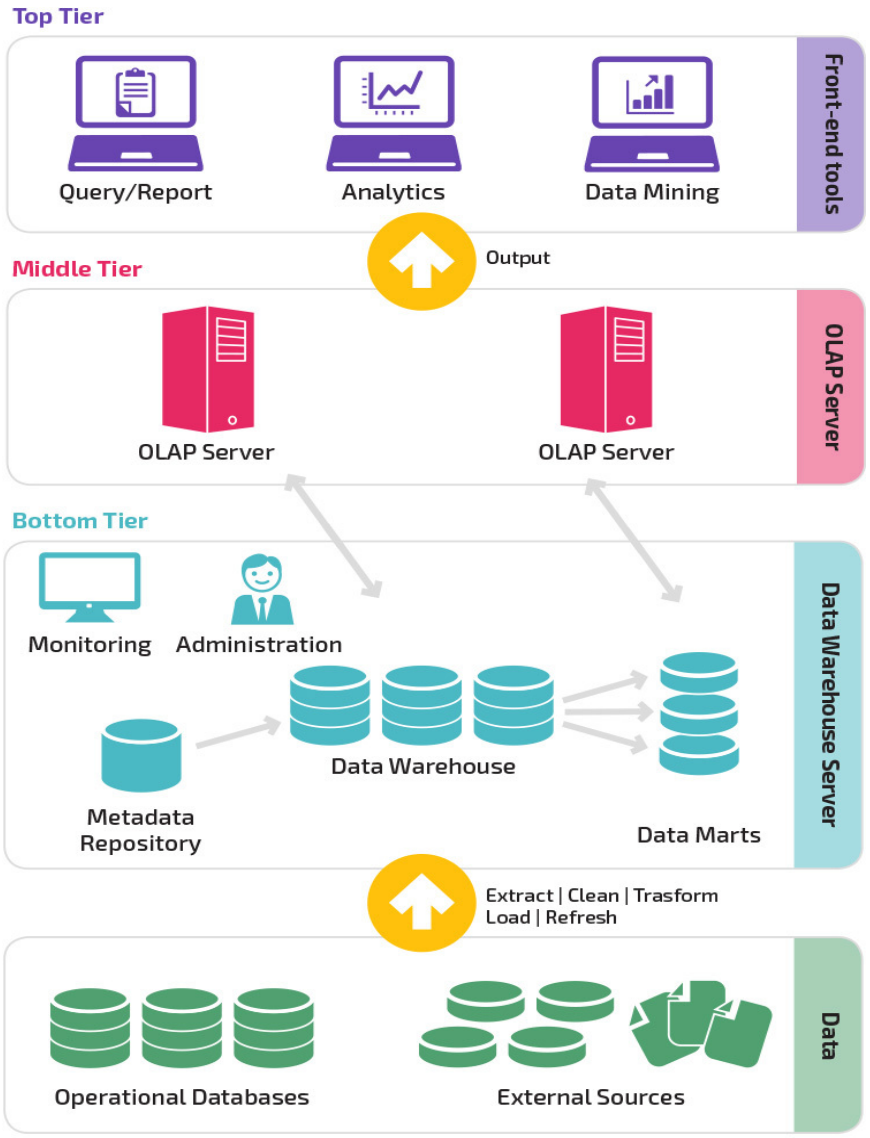
- Data Warehouse Architecture
- OLTP v/s OLAP
- Data Warehouse Schema Architecture
- OLAP Operations
- OLAP Servers

Data Warehouse Architecture

Top Tier

Middle Tier

Bottom Tier



Data Warehouse Architecture

- **Bottom tier:**
  - The **bottom tier** is a warehouse **database server** that is almost always a relational database system.
  - **Back-end tools and utilities are used to feed data** into the bottom tier from operational databases or other external sources.
  - These tools and utilities **perform data extraction, cleaning, and transformation** , as well as load and refresh functions to update the data warehouse.
  - The data are extracted using application program interfaces known as **gateways**.
  - A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.
  - Examples of gateways include ODBC (Open Database Connection) and OLEDB (Open Linking and Embedding for Databases) by Microsoft and JDBC (Java Database Connection).
  - This tier also **contains a metadata repository** , which stores information about the data warehouse and its contents.

Data Warehouse Architecture

- **Middle tier:**
  - The middle tier is an OLAP (Online Analytical Processing Server) that is typically implemented using either
    - A **relational OLAP (ROLAP**) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations or,
    - A **multidimensional OLAP (MOLAP)** model, that is, a special-purpose server that directly implements multidimensional data and operations.
- **Top tier:**
  - The top tier is a front-end client layer, which contains **query and reporting tools, analysis tools, and/or data mining tools**.

OLAP (**O**n-**L**ine **A**nalytical **P**rocessing)

- OLAP is characterized by relatively **low volume of transactions**.
- Queries are often **very complex and involve aggregations**.
- For OLAP systems a **response time is an effectiveness measure**.
- OLAP applications are widely used by Data Mining techniques.
- In OLAP database there is **aggregated, historical data, stored in multi-dimensional** schemas (usually star schema).

OLTP (**O**n-**L**ine **T**ransaction **P**rocessing)

- It is characterized by a **large number of short on-line transactions** (INSERT, UPDATE, DELETE).
- The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second.
- In OLTP database, **there is detailed and current data**, and schema used to store transactional databases is the entity model (usually 3NF).

OLTP v/s OLAP (Understanding)

OLTP	OLAP
Many Short Transactions (Queries + Updates)	Long Transactions (Complex Queries)
Examples · Update account balance · Enroll in course · Add book to shopping cart	**Examples** · Report total sales for each department in each month · Identify top-selling books · Count classes with fewer than 10 students
Queries touch small amount of data (one record or few records)	Queries touch large amount of data
Updates are frequent	Updates are infrequent

OLTP v/s OLAP

Functionality	OLTP	OLAP
Characteristic	Operational processing informational processing	Transaction Analysis
Orientation	Transaction	Analysis
User	Clerk, DBA, database professional	Knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, **application-oriented**	Star/snowflake, **subject-oriented**
Data	Current; guaranteed up-to-date	Historical; accuracy maintained over time
Summarization	Primitive, highly detailed	Summarized, consolidated
View	Detailed, flat relational	Summarized, multidimensional
Unit of work	Short, simple transaction	Complex query
Access	Read/write	Mostly read

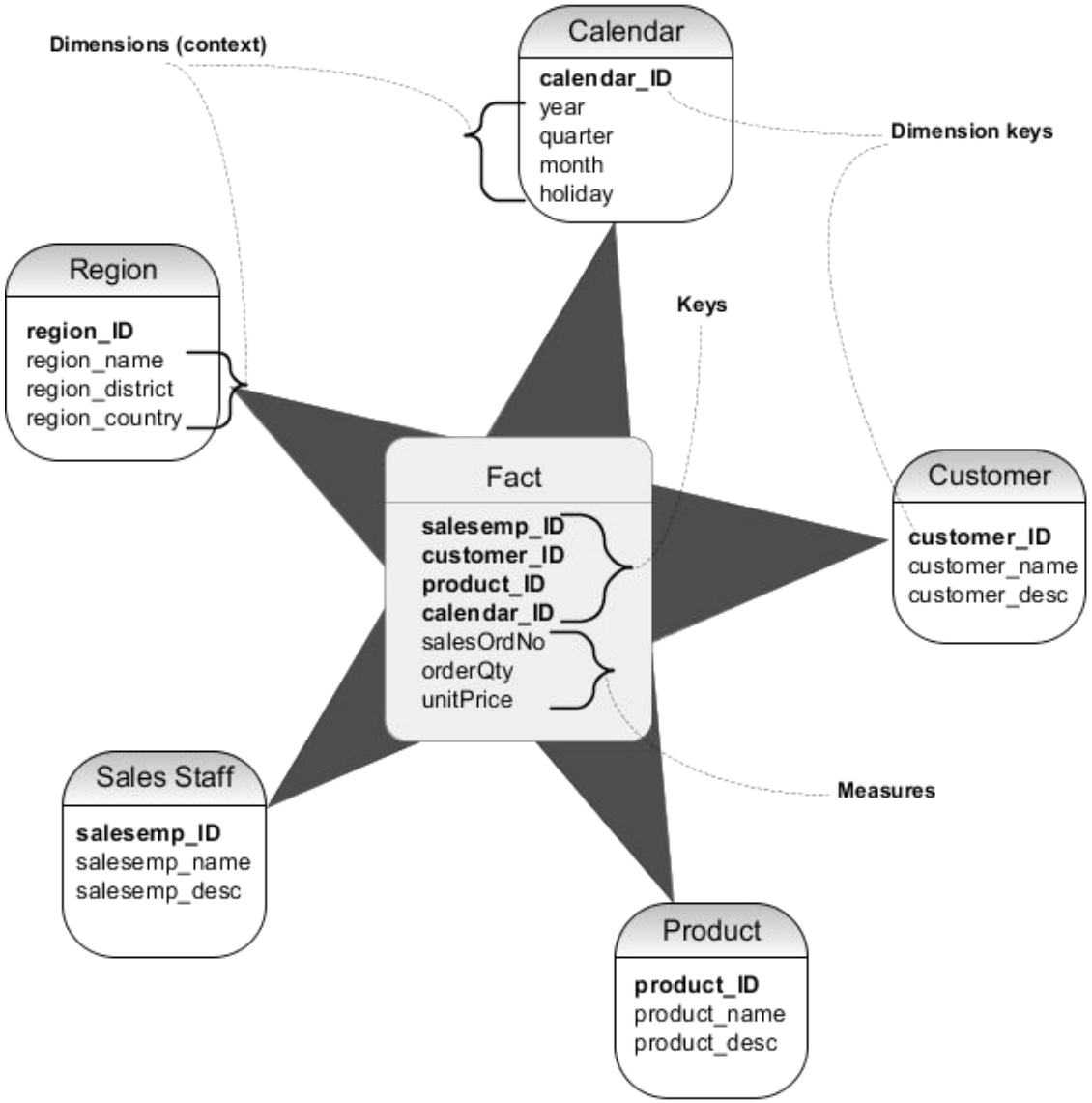
Data Warehouse Schema Architecture

- Data Warehouse environment usually transforms the relational data model into some special architectures.
- There are many schema models designed for data warehousing but the most commonly used are:
  - **Star Schema**
  - **Snowflake Schema**
  - **Fact constellation(Group of star, Collection of fact tables) Schema**
- The determination of which schema model should be used for a data warehouse based upon the analysis of project requirements, accessible tools and project team preferences.

Star Schema

- The star schema architecture is the **simplest data warehouse schema.**
- It is called a star schema because the diagram resembles a **star**, with points radiating from a center.
- The center of the star consists of **fact table** and the **points of the star are the dimension tables**.
- Usually the fact tables in a star schema are in third normal form (3NF) whereas dimensional tables are de-normalized.
- Despite the fact that the star schema is the simplest architecture, it is **most commonly used nowadays** and is recommended by Oracle.

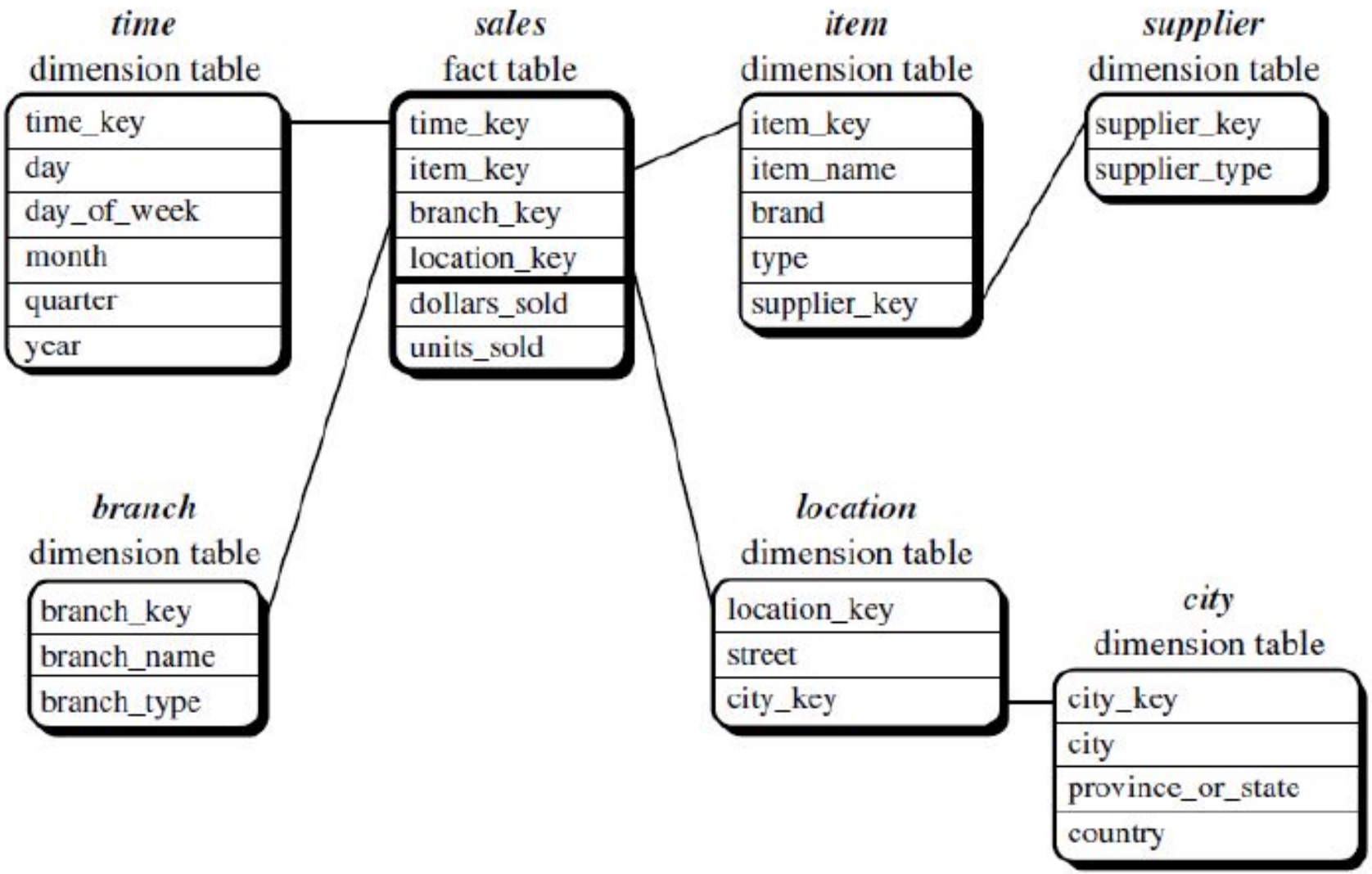
Star Schema - Example



Snowflake Schema

- The snowflake schema architecture is a **more complex variation of the star schema** used in a data warehouse, because the tables which describe the dimensions are normalized.
- This table is easy to maintain and saves storage space.
- However, this saving of space is negligible in comparison to the typical size of the fact table.
- Furthermore, the snowflake structure can reduce the effectiveness of browsing, since **more joins** will be needed to execute a query.
- Hence, although the **snowflake schema reduces redundancy**, it is not as popular as the star schema in data warehouse design.

Snowflake Schema - Example



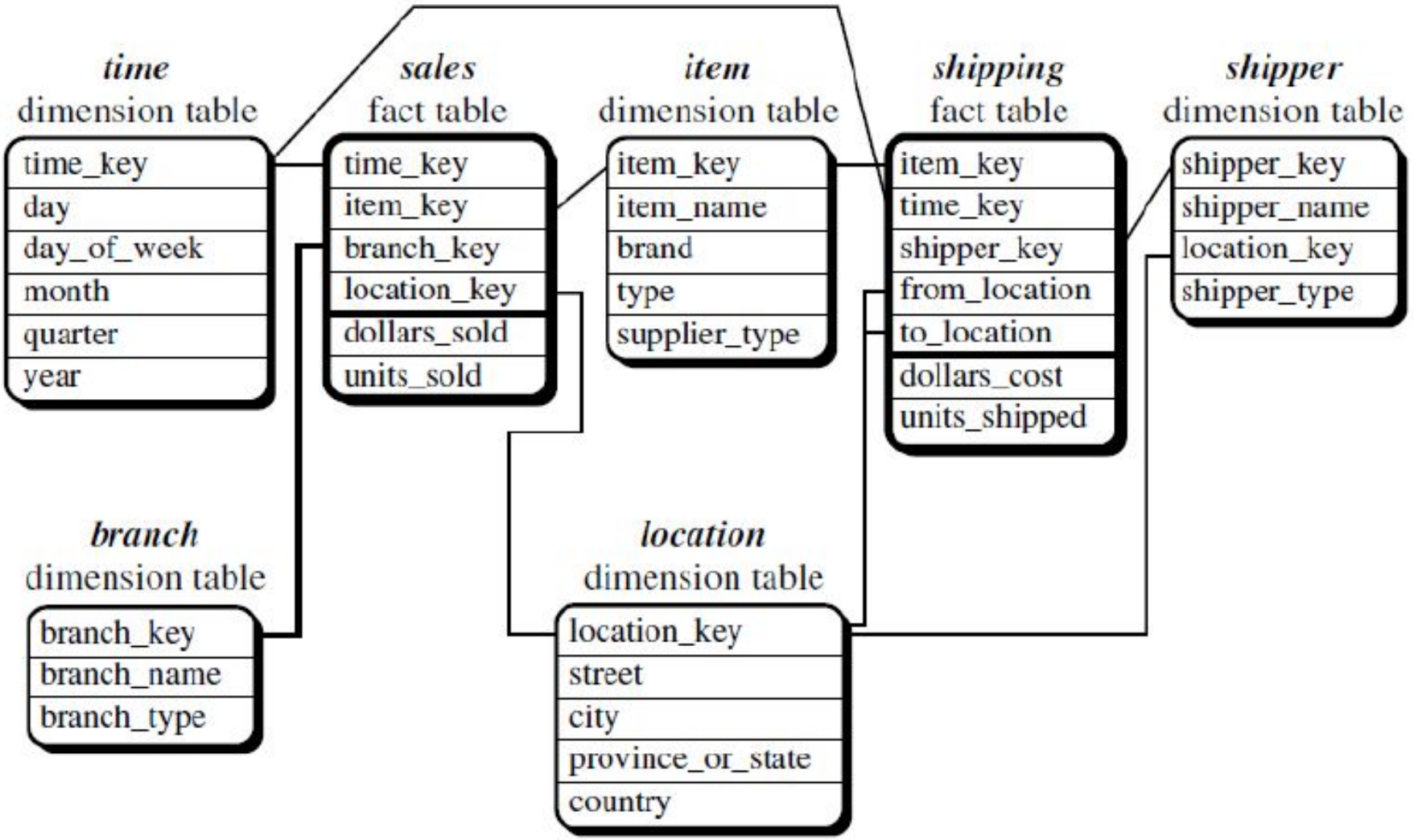
Snowflake Schema - Example

- DMQL(Data Mining Query Language) code for Snowflake Schema can be written 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

- Sophisticated applications may require **multiple fact tables** to share dimension tables.
- This kind of schema can be viewed as a **collection of stars**, and hence is called a **galaxy schema** or a **fact constellation**.
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimensions tables for ***time, item***, and ***location*** are shared between both the **sales and shipping** fact tables.
- The main shortcoming of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected.

Fact Constellation Schema



Fact Constellation Schema

- DMQL code for Fact Constellation schema can be written 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

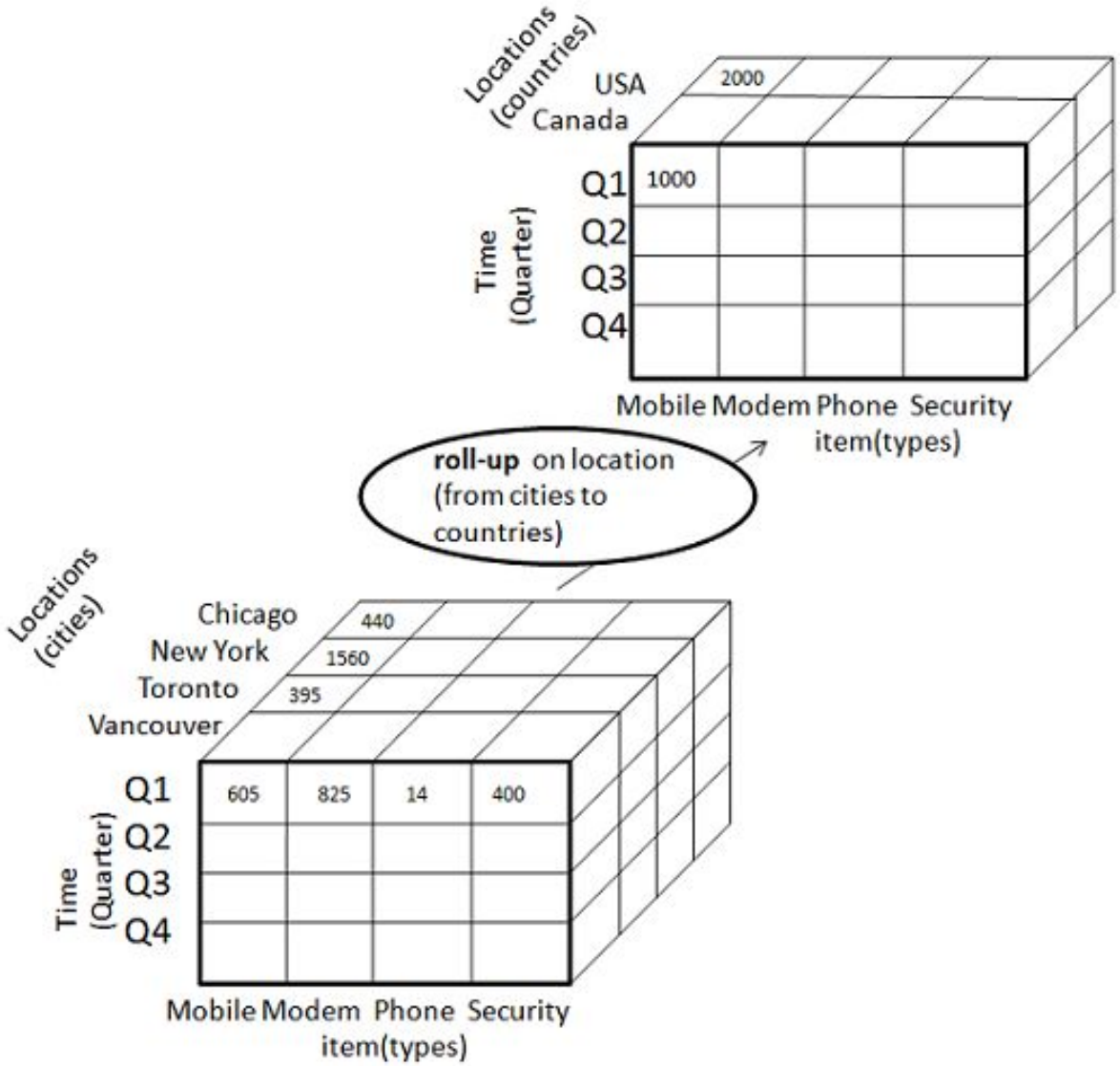
OLAP Operations

- Roll up
- Drill Down
- Slice
- Dice
- Pivot (Rotate)

Roll up – OLAP Operation

- The roll-up operation (also called drill-up or aggregation operation) **performs aggregation on a data cube** by following ways:
  - By climbing up a concept hierarchy for a dimension
  - By dimension reduction
- 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.

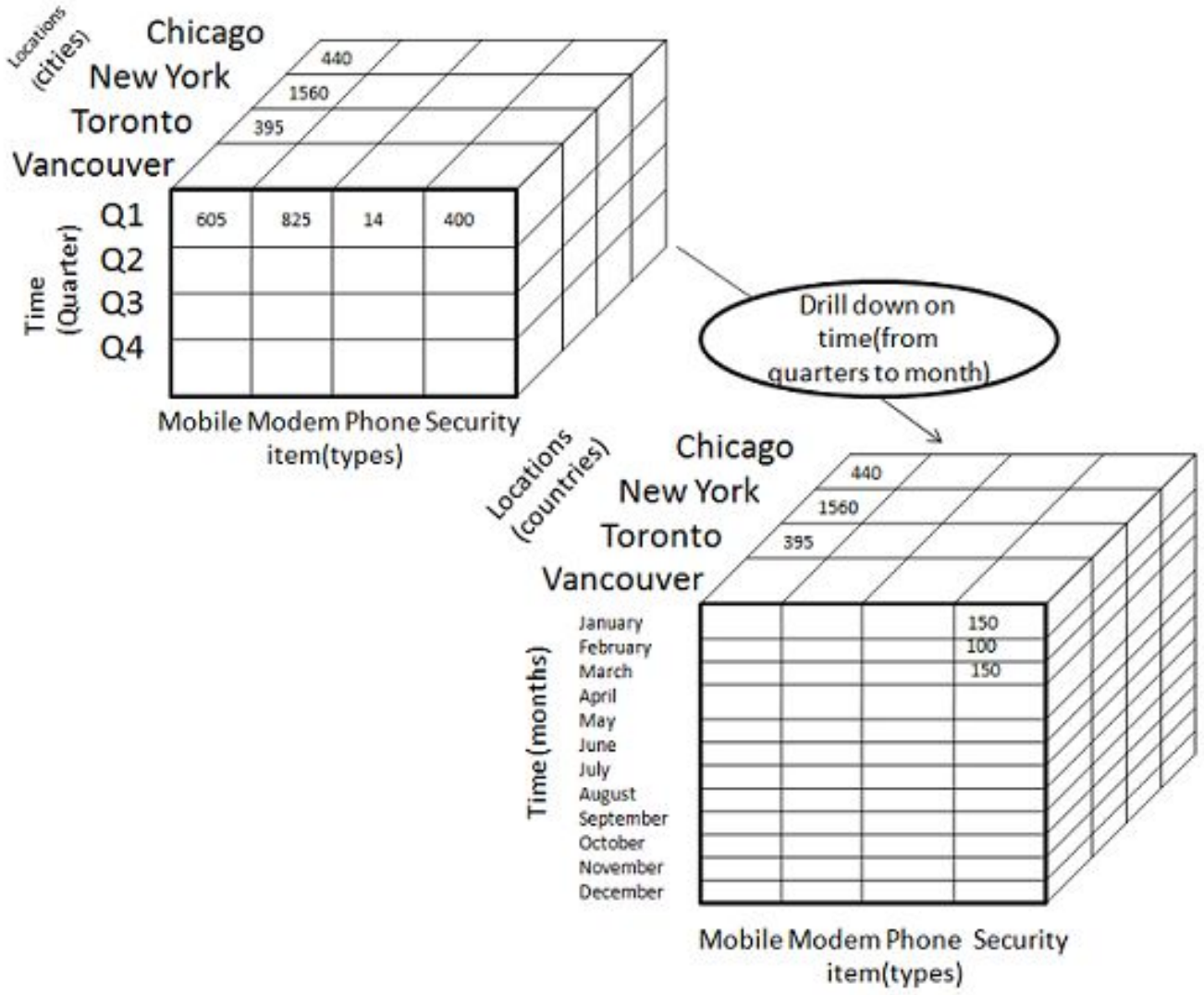
Roll up – OLAP Operation



Drill Down – OLAP Operation

- 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
- 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.

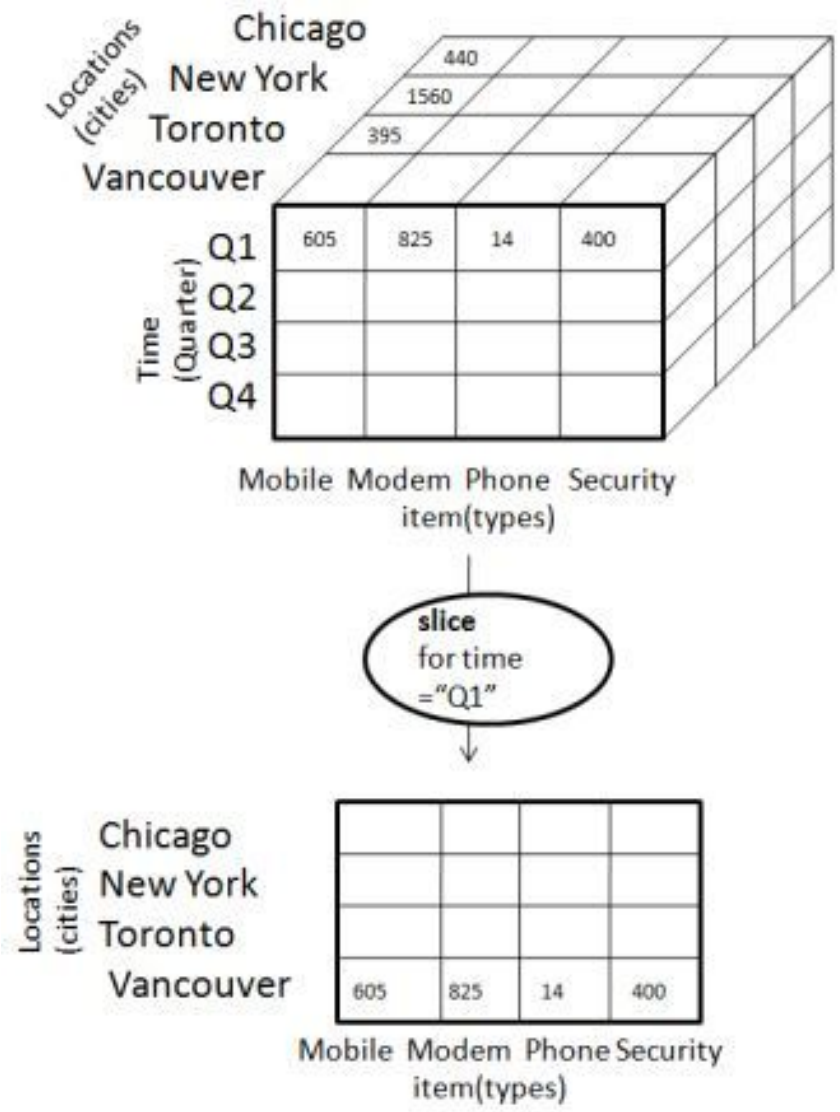
Drill Down – OLAP Operation



Slice – OLAP Operation

- The slice operation **selects one particular dimension from a given cube and provides a new sub cube**.
- Here Slice is performed for the dimension "time" using the criterion time = "Q1“, time = "Q2“, time = "Q3“ etc.
- It will form a new sub-cube by selecting one or more dimensions.

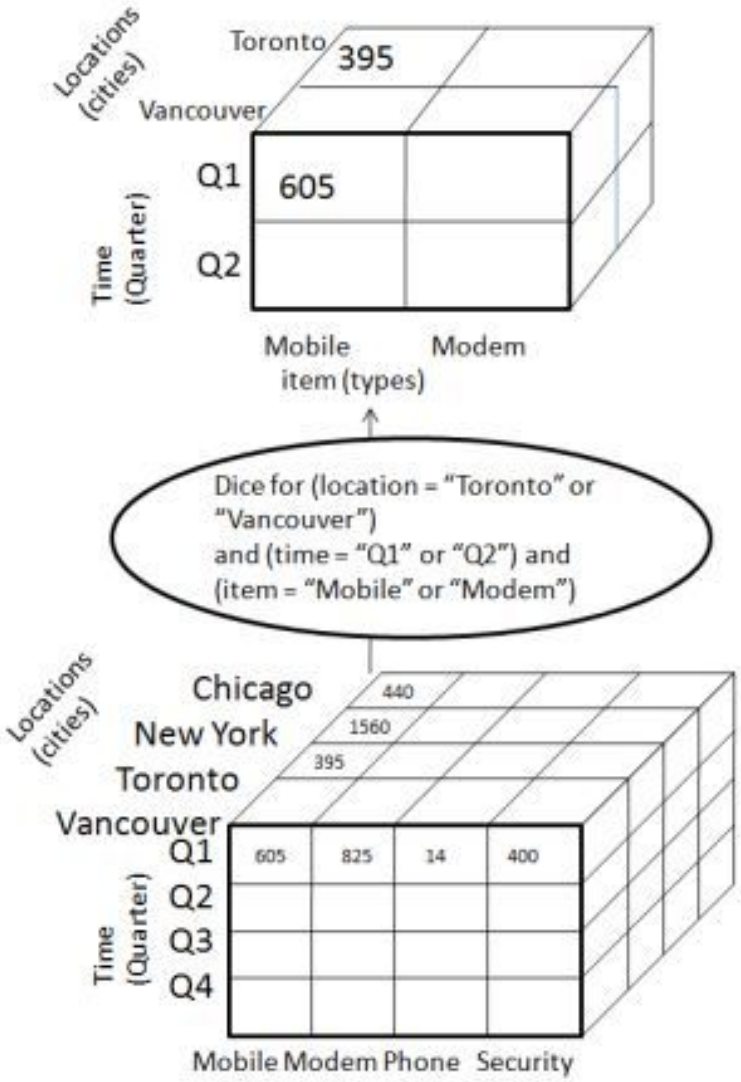
Slice – OLAP Operation



Dice – OLAP Operation

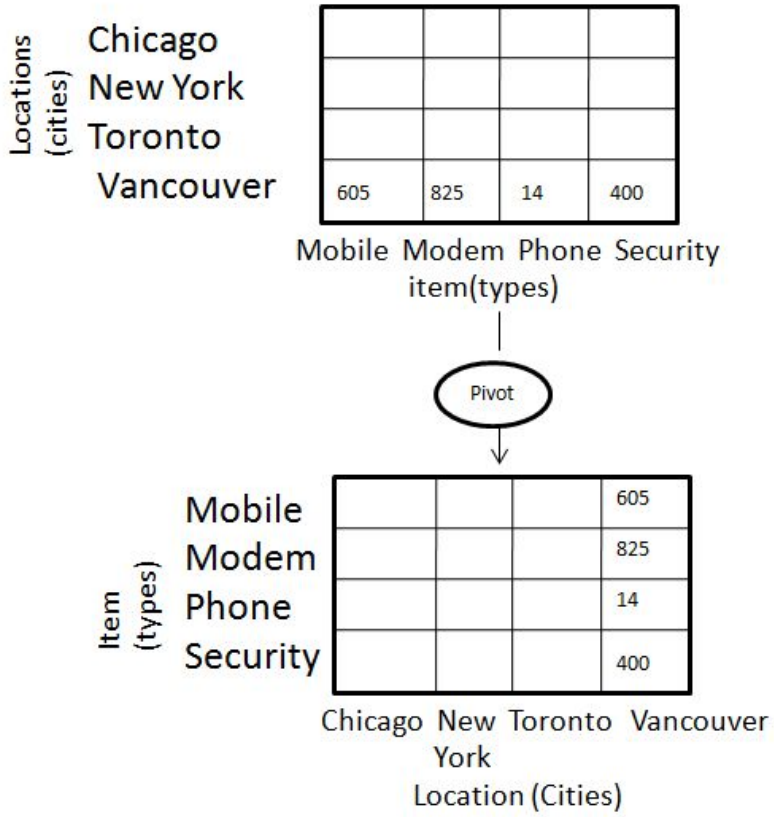
- Dice **selects two or more dimensions** from a given cube and provides a new sub cube.
- The dice operation on the cube based on the following selection criteria involves three dimensions.
  - (location = "Toronto" or "Vancouver")
  - (time = "Q1" or "Q2")
  - (item =" Mobile" or "Modem")

Dice – OLAP Operation



Pivot – OLAP Operation

- The pivot operation is also known as **rotation**.
- It rotates the data axes in view in order to provide an alternative presentation of data.



OLAP Servers

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)

Relational OLAP (ROLAP)

- Relational On-Line Analytical Processing (ROLAP) work mainly for the data that resides in a **relational database**, where the base data and dimension tables are **stored as relational tables**.
- ROLAP servers are placed between the relational back-end server and client front-end tools.
- ROLAP servers use RDBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.
  - **Advantages of ROLAP**
    - ROLAP can handle large amounts of data.
    - Can be used with data warehouse and OLTP systems.
  - **Disadvantages of ROLAP**
    - Limited by SQL functionalities.
    - Hard to maintain aggregate tables.

Multidimensional OLAP (MOLAP)

- Multidimensional On-Line Analytical Processing (MOLAP) support **multidimensional views of data** through array-based multidimensional storage engines.
- With multidimensional data stores, the storage utilization may be low if the data set is sparse.
  - **Advantages of MOLAP**
    - Optimal for slice and dice operations.
    - Performs better than ROLAP when data is dense(heavy).
    - Can perform complex calculations.
  - **Disadvantages of MOLAP**
    - Difficult to change dimension without re-aggregation.
    - MOLAP can handle limited amount of data.

Hybrid OLAP (HOLAP)

- Hybrid On-Line Analytical Processing (HOLAP) is a **combination of ROLAP and MOLAP**.
- HOLAP provide greater scalability of ROLAP and the faster computation of MOLAP.
  - **Advantages of HOLAP**
    - HOLAP provide advantages of both MOLAP and ROLAP.
    - Provide fast access at all levels of aggregation.
  - **Disadvantages of HOLAP**
    - HOLAP **architecture is very complex** because it support both MOLAP and ROLAP servers.

Thank you!