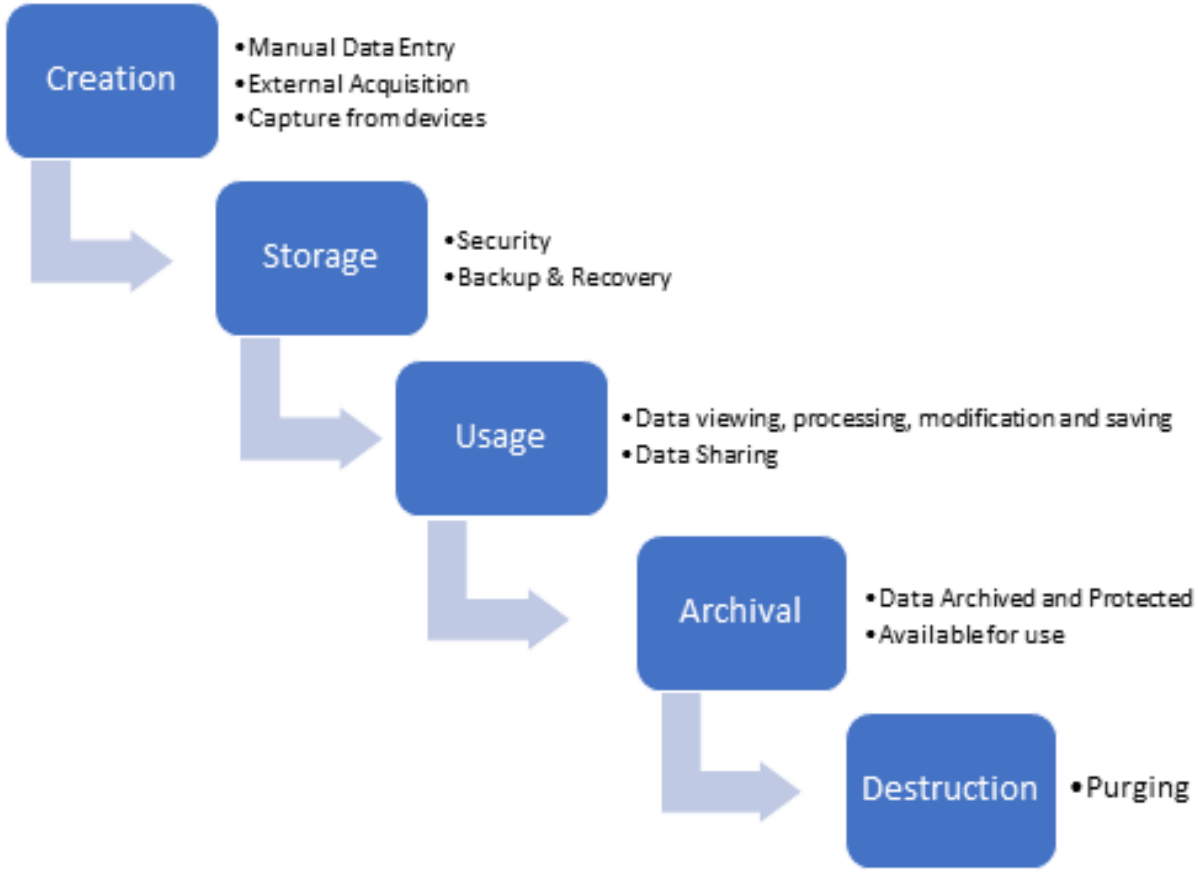
Unit 1
Introduction to Data Warehousing

Prepared By

Arjun Singh Saud, Asst. Prof. CDCSIT

Life Cycle of Data

- The data lifecycle represents all of the stages of data throughout its life from its creation for a study to its distribution and destruction.



Life Cycle of Data

- **Data Collection/Creation**: In this phase, data comes into an organization, usually through data entry, acquisition from an external source or signal reception, such as transmitted sensor data.
- **Data Storage**: Once data has been created within the organization, it needs to be stored and protected, with the appropriate level of security applied. A robust backup and recovery process should also be implemented to ensure retention of data during the lifecycle.

Life Cycle of Data

- **Data Usage:** During the usage phase of the data lifecycle, data is used to support activities in the organization. Data can be viewed, processed, modified and saved. Data may also be made available to share with others outside the organization.
- **Data Archival:** Data Archival is the process of removing data from active production environment and keeping copy of data so that it can be used again in an active production environment in future, if needed.

Life Cycle of Data

- **Data Destruction**: Data destruction or purging is the removal of every copy of a data item from an organization. It is typically done from an archive storage location. If we want to save all data forever, it's not feasible. Storage cost and compliance issues create pressure to destroy data no longer need.

Types of Data

- Data flows into a data warehouse from the transactional system and other relational databases. Data may be:
  - *Structured data*
  - *Semi-structured data*
  - *Unstructured data.*

Types of Data

Structured Data

- Data that is the easiest to search and organize, because it is usually contained in rows and columns and its elements can be mapped into fixed pre-defined fields, is known as structured data. Structured data follows a relational data model.
- Relation Databases and SQL is suitable for managing structured data.

Types of Data

Unstructured Data

- Data that cannot be contained in a row-column database is called unstructured data and doesn't have an associated data model. The lack of structure made unstructured data more difficult to search, manage and analyze, which is why companies have widely discarded unstructured data, until the recent proliferation of AI and machine learning algorithms made it easier to process.
- Examples of unstructured data include photos, video and audio files, text files, social media content, open-ended survey responses etc. Instead of relational databases, unstructured data is usually stored in NoSQL databases and data warehouses.

Types of Data

Semi-structured Data

- The type of data defined as semi-structured data has some defining or consistent characteristics but doesn't conform to a structure as rigid as is expected with a relational database.
- There are some organizational properties such as semantic tags to make it easier to organize, but there's still variability in the data.
- Email messages are a good example. While the actual content is unstructured, it does contain structured data such as name and email address of sender and recipient, time sent, etc. XML is suitable for managing semi-structured data.

Data Warehouse

- A data warehouse is a repository of information collected from multiple sources that stores historical data and provides support for decision-makers for data modeling and analysis.
- The data warehouse is the core of the Business Intelligence system which is built for data analysis and reporting.
- Data warehousing is the process of building data warehouse. It requires ETL operations and requires periodic data refreshing.
- ETL is a process that **extracts** the data from different source systems, then **transforms** the data and finally **loads** the data into the Data Warehouse system. Full form of ETL is Extract, Transform and Load.

Data Warehouse

- Key features of data warehouse are: *Subject Oriented, Integrated, Time Variant, and Non-Volatile.*
- **Subject Oriented:** A data warehouse target on the modeling and analysis of data for decision-makers. Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations. This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

Data Warehouse

- **Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactional database system, where only the most recent data is kept. For example, a transactional database system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer since last 5 years.

Data Warehouse

- **Integrated:** A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records etc. It requires performing data cleaning and transformation during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.
- **Non-volatile–** The data in a warehouse is non-volatile. This ensures that your previous data is not lost as new data is updated rather new version of the data is inserted. This separates data warehouse from operational databases which are subject to frequent changes.

Operational Database vs Data Warehouse

- The major task of database systems is to perform on-line transaction and query processing. These systems are called on-line transaction processing (OLTP) systems. They cover most of the day-to-day operations of an organization, such as purchasing, inventory, manufacturing, banking, payroll, registration, accounting etc. On the other hand Data Warehouse systems serve knowledge workers in the role of data analysis and decision making. These systems are known as on-line analytical processing (OLAP) systems.
- Key differences between operational databases and data warehouse is provided in next slide.

Operational Database vs Data Warehouse

Operational Database	Data Warehouse
• Databases use Online Transactional Processing (OLTP) to delete, insert, replace, and update large numbers of short online transactions quickly.	• Data warehouses use Online Analytical Processing (OLAP) to analyze massive volumes of data rapidly. This process gives analysts the power to look at your data from different points of view.
• Databases stores current data only	• Data warehouses stores historical data
• Databases are optimized for performing write(add, delete, modify) operations.	• Optimized for efficiently reading/ retrieving large data sets and for aggregating data.
• The data in databases are normalized to reduce or eliminate data redundancy	• The data in data warehouses are denormalized so that data can be accessed faster.

Operational Database vs Data Warehouse

Operational Database	Data Warehouse
• Databases serves data workers that are in the role of processing day-to-day data.	• Data Warehouses systems serve knowledge workers in the role of data analysis and decision making.
• Databases usually adopts an entity-relationship (ER) data model and an application-oriented database design.	• Data warehouses typically adopts star or snowflake model and a subject oriented database design.
• Database transactions usually are executed in an ACID (Atomic, Consistent, Isolated, and Durable) compliant manner.	• Since data warehouses focus on reading, rather than modifying, historical data from many different sources, ACID compliance is less strictly enforced.

Multidimensional Data Model

- Data warehouses and OLAP tools are based on a multidimensional data model. This is the data model that views data in the form of a *data cube*.
- A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by *dimensions* and *facts*.
- Dimensions are the entities with respect to which an organization wants to keep records. For example, an organization may create a *sales* data warehouse in order to keep records of the store's sales with respect to the dimensions *time*, *item*, *branch*, and *location*.

Multidimensional Data Model

- Each dimension may have a table associated with it, called a *dimension table*. This table further describes the dimensions. For example, a dimension table for *item* may contain the attributes *item name, brand,* and *type*.
- A multidimensional data model is typically organized around a central theme, like *sales*. This theme is represented by a fact table.
- Facts are numerical measures. These are the quantities by which we want to analyze relationships between dimensions. Examples of facts for a sales data warehouse include *sales_amount, units_sold*.

Multidimensional Data Model

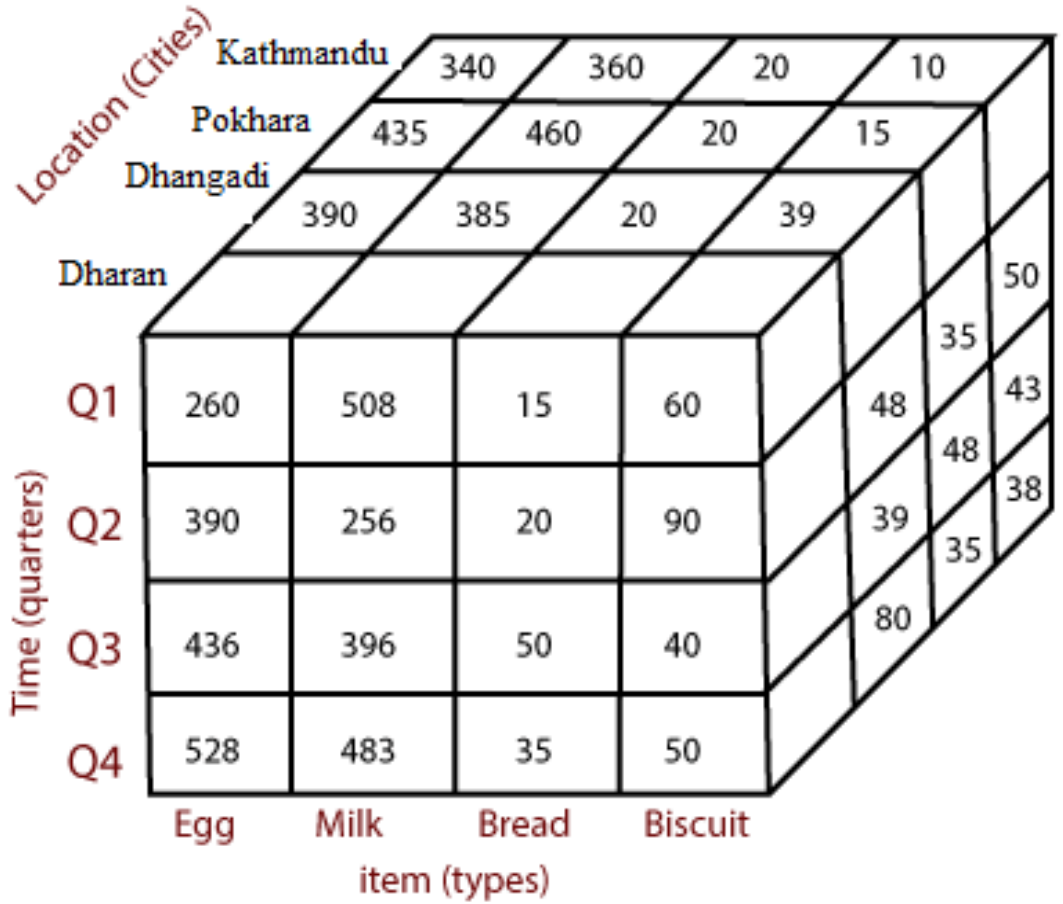
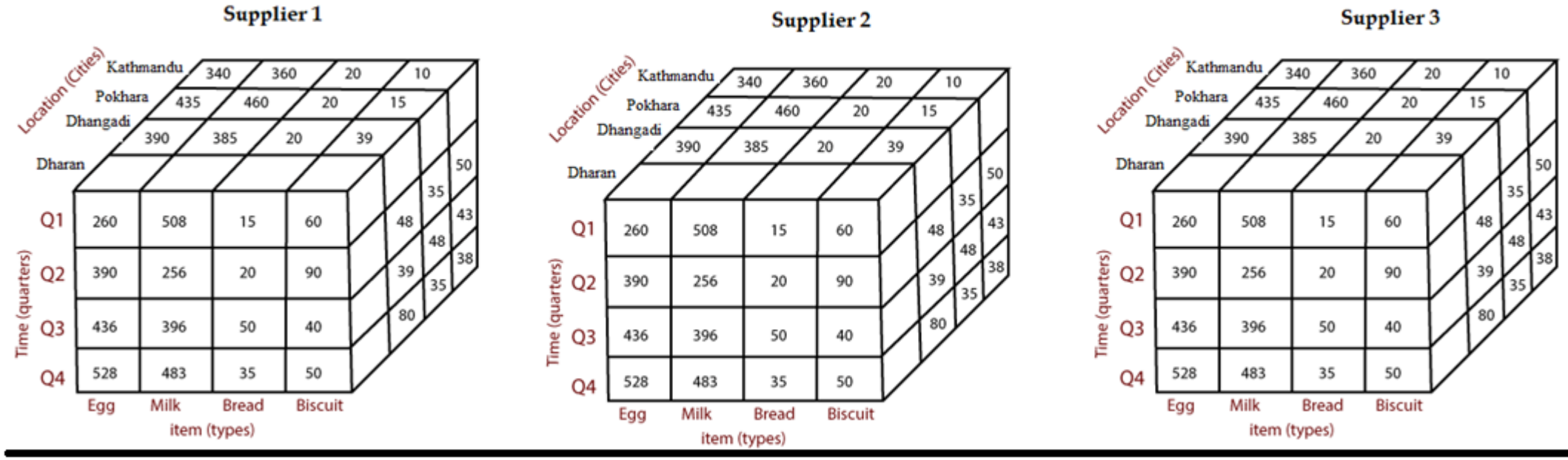


Figure: A 3-D representation of the data according to the dimensions time, item, and location.

Multidimensional Data Model

- Suppose that we would now like to view our sales data with an additional fourth dimension, such as *supplier*. Viewing things in 4-D becomes tricky. However, we can think of a 4-D cube as being a series of 3-D cubes, as shown in Figure below.



Multidimensional Data Model

- If we continue in this way, we may display any *n*-dimensional data as a series of (*n*-1) dimensional cubes.
- The data cube is a metaphor for multidimensional data storage. The actual physical storage of such data may differ from its logical representation.
- The important thing to remember is that data cubes are *n*-dimensional and do not confine data to 3-D.

Conceptual Modeling of Data Warehouse

- A conceptual data model recognizes the highest-level relationships between the different entities.
- The goal of conceptual data warehouse modeling is to develop a schema for logical representation of data stored in data warehouse.
- Schema is a logical description of the entire data warehouse. It includes the name and description of records and aggregates.
- We use *Star schema, Snowflake schema, and Fact-Constellation schema* for conceptual modeling of data warehouse.

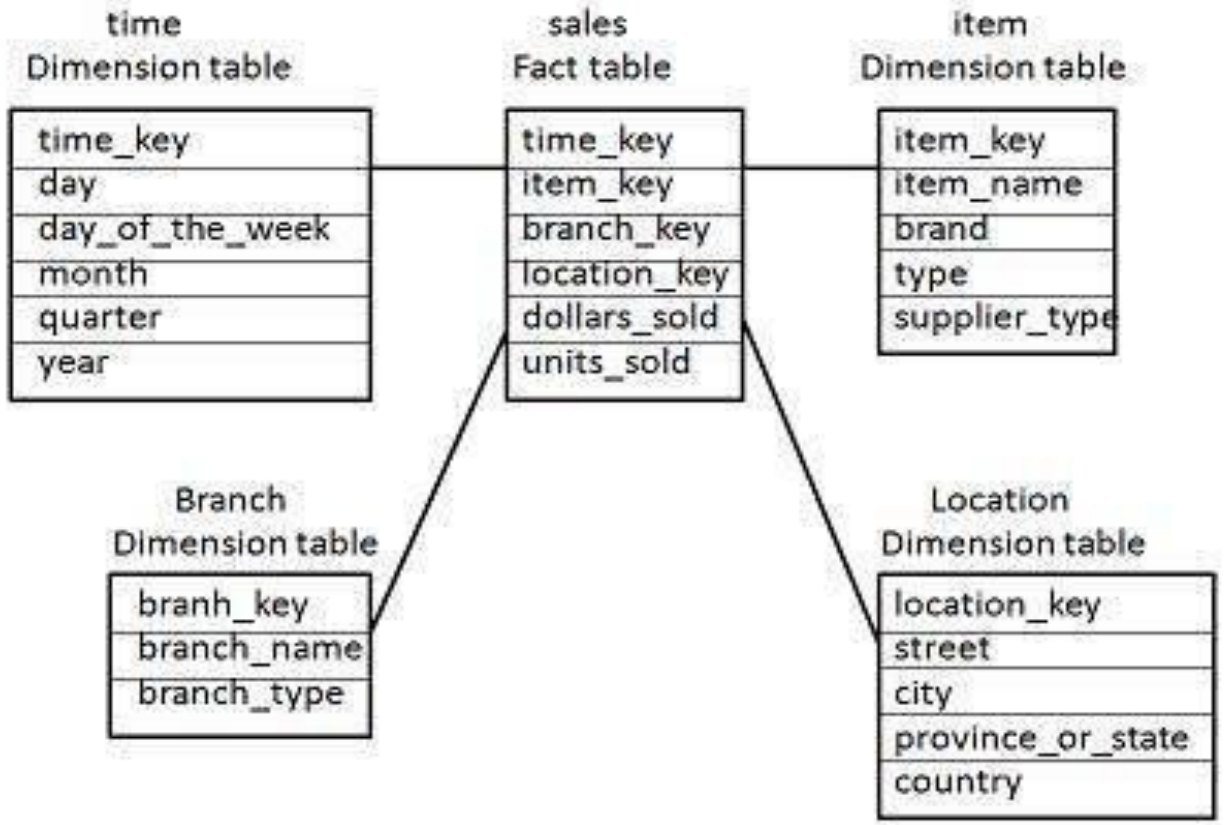
Conceptual Modeling of Data Warehouse

Star Schema

- It is the data warehouse schema that contains two types of tables: *Fact Table and Dimension Tables*. Fact Table lies at the center point and dimension tables are connected with fact table such that star shape is formed.
  - **Fact Tables:** A fact table typically has two types of columns: foreign keys to dimension tables and measures that contain numeric facts. Those facts contain aggregates of data at specified level.
  - **Dimension Tables:** Dimension tables usually have a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data.

Conceptual Modeling of Data Warehouse

Star Schema



Conceptual Modeling of Data Warehouse

Star Schema

- Since star schema contains **de-normalized** dimension tables, it leads to **simpler queries** due to lesser number of join operations and it also leads to **better system performance**. On the other hand, it is difficult to **maintain integrity** of data and **data redundancy** is also high in star schema due to de-normalized tables. It is the widely used data warehouse schema and is also recommended by oracle

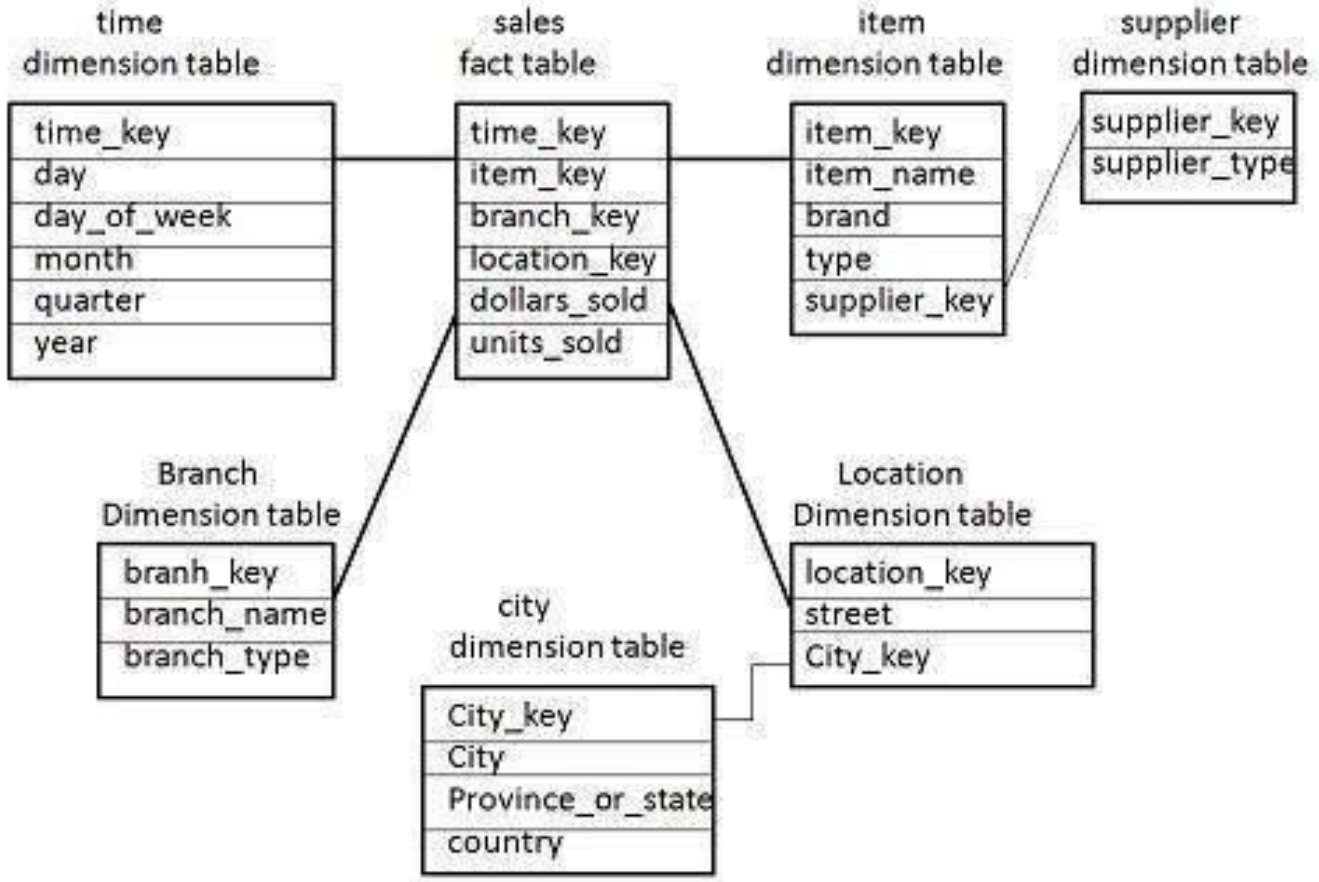
Conceptual Modeling of Data Warehouse

Snowflake Schema

- The snowflake schema is a variant of the star schema model, where some dimension tables are **normalized,** thereby further splitting the data into additional tables.
- The resulting schema graph forms a shape similar to a snowflake. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.

Conceptual Modeling of Data Warehouse

Snowflake Schema



Conceptual Modeling of Data Warehouse

Snowflake Schema

- Due to normalization table is easy to **maintain integrity** and **saves storage space**. However, this saving of space is negligible in comparison to the typical magnitude of the fact table. Furthermore, the snowflake structure can reduce the effectiveness of browsing, since **more joins will be needed to execute a query**. Consequently, the **system performance** may be adversely impacted. Hence, although the snowflake schema reduces **redundancy**, it is not as popular as the star schema in data warehouse design.

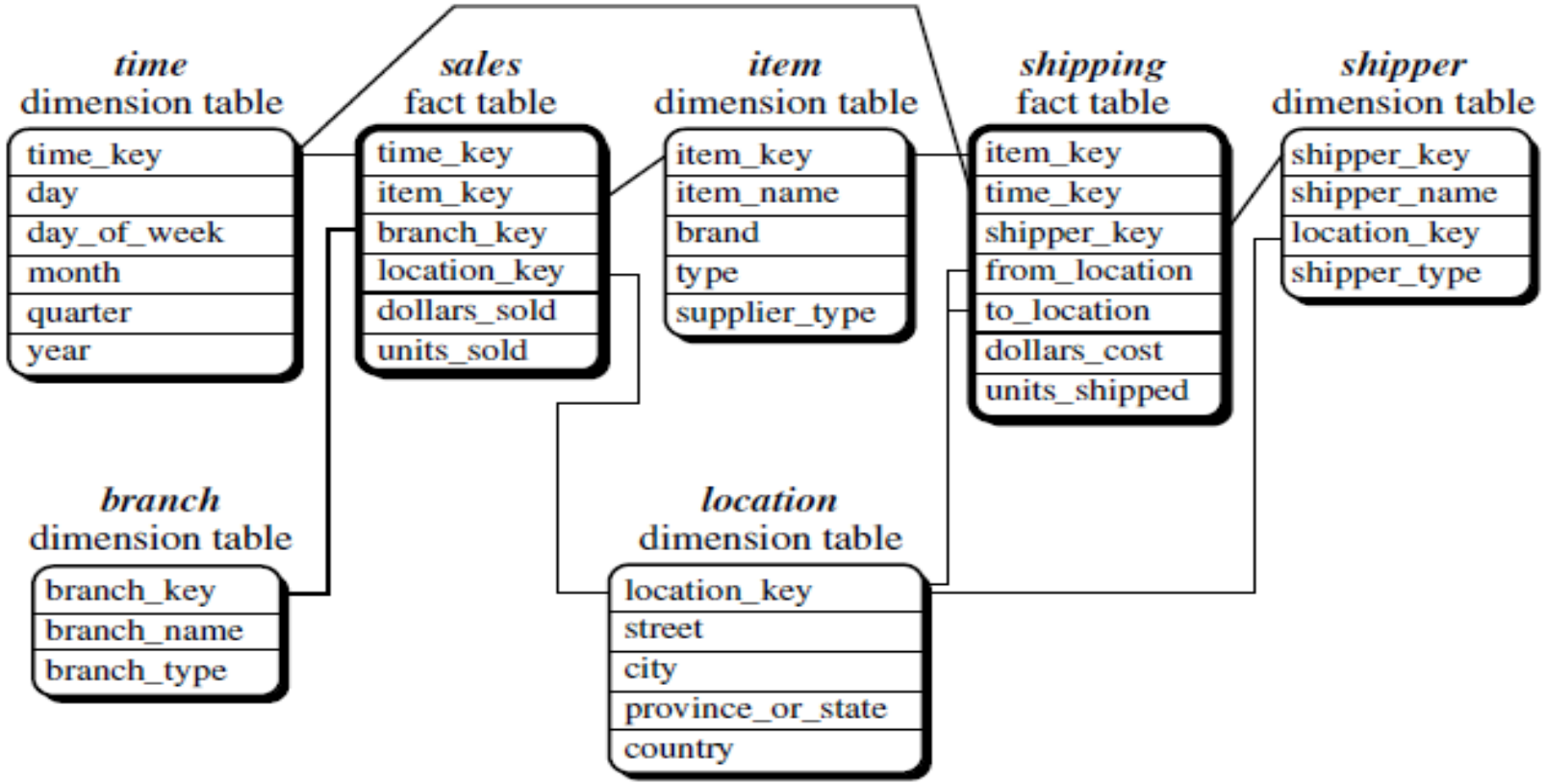
Conceptual Modeling of Data Warehouse

Fact-Constellation Schema

- This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation schema. **A fact constellation schema allows dimension tables to be shared between fact tables**.
- For example, following schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of the star schema. The *shipping* table has five dimensions, or keys: *item key, time key, shipper key, from location*, and *to location*, and two measures: *dollars cost* and *units shipped*.

Conceptual Modeling of Data Warehouse

Fact-Constellation Schema



OLAP Operations

- Online Analytical Processing (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.
- Here is the list of OLAP operations: *Roll-up, Drill-down, Slice, Dice, Pivot (rotate).*

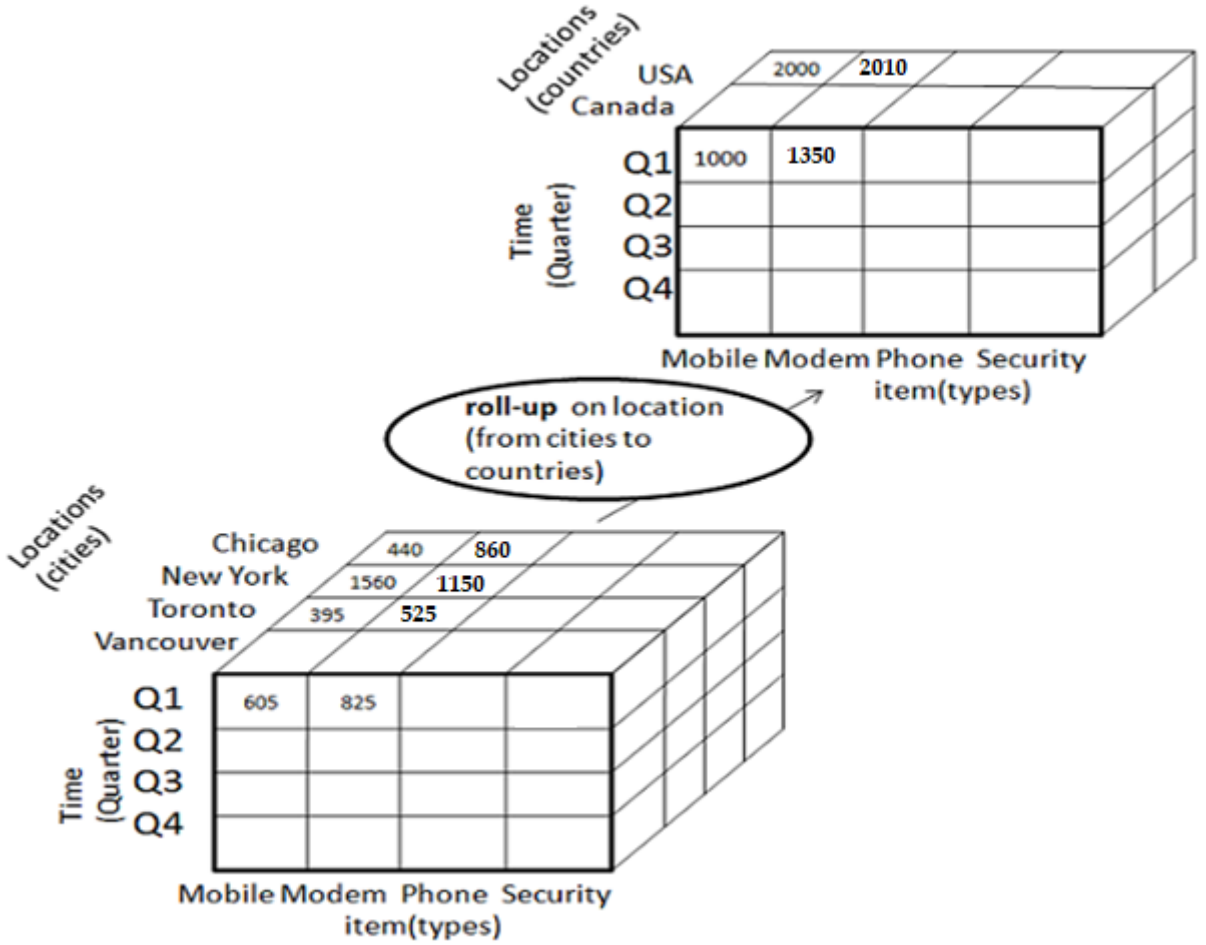
OLAP Operations

Roll-up

- Roll-up is also known as "consolidation" or "aggregation." The Roll-up operation can be performed in 2 ways
  - Reducing dimensions
  - Climbing up concept hierarchy
- Concept hierarchy is a system of grouping things based on their order or level.

OLAP Operations

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.



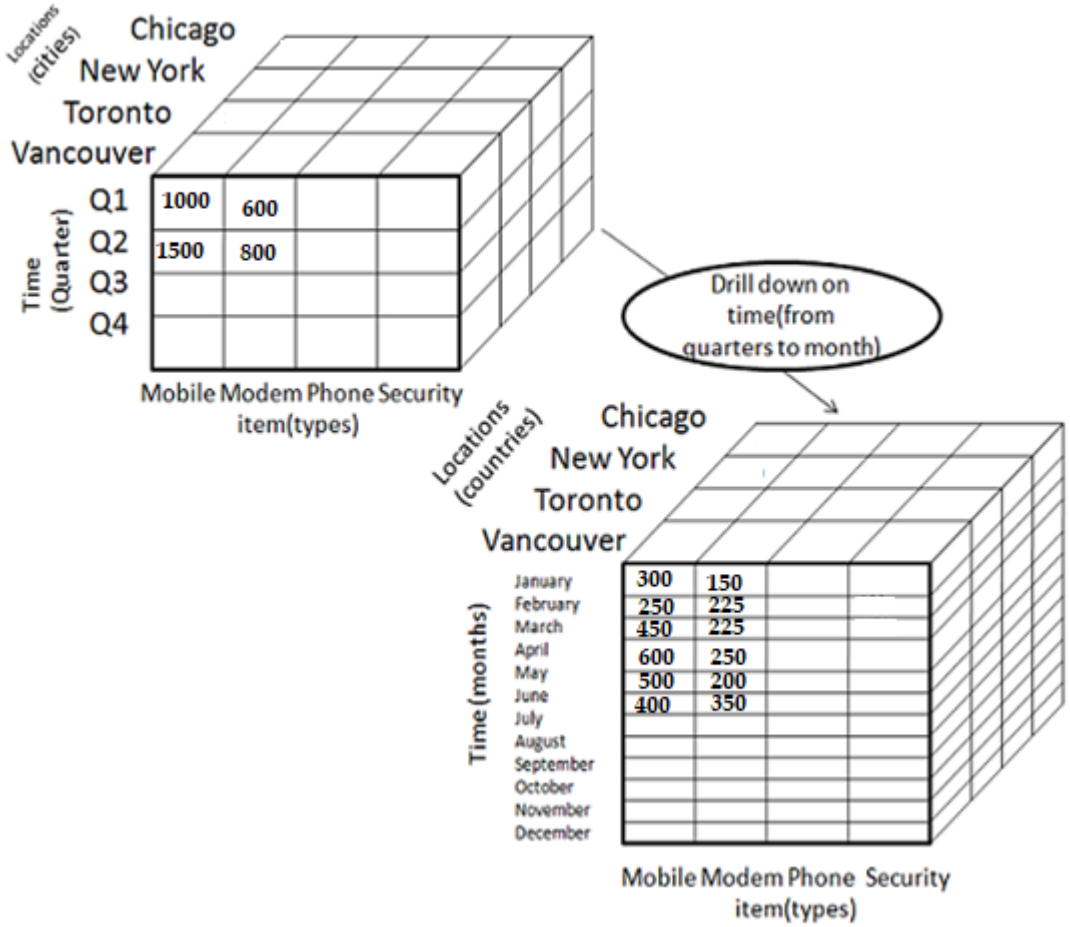
OLAP Operations

Drill-down

- Drill-down is the reverse operation of roll-up. This operation is performed when we want to view details of aggregate data. It is performed by either of the following ways:
  - By stepping down a concept hierarchy for a dimension or
  - By introducing a new dimension.

OLAP Operations

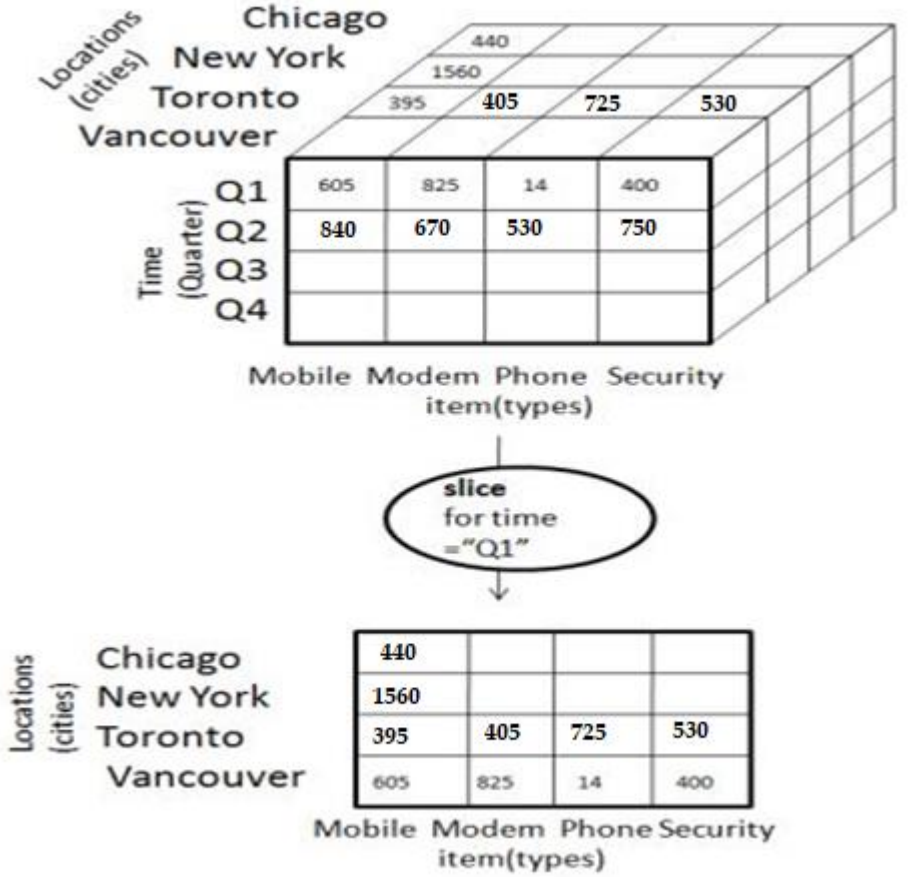
Drill-down is performed by stepping down a concept hierarchy for the dimension time. Initially the concept hierarchy was "day < month < quarter < year." On drilling down, the time dimension is descended from the level of quarter to the level of month.



OLAP Operations

Slice

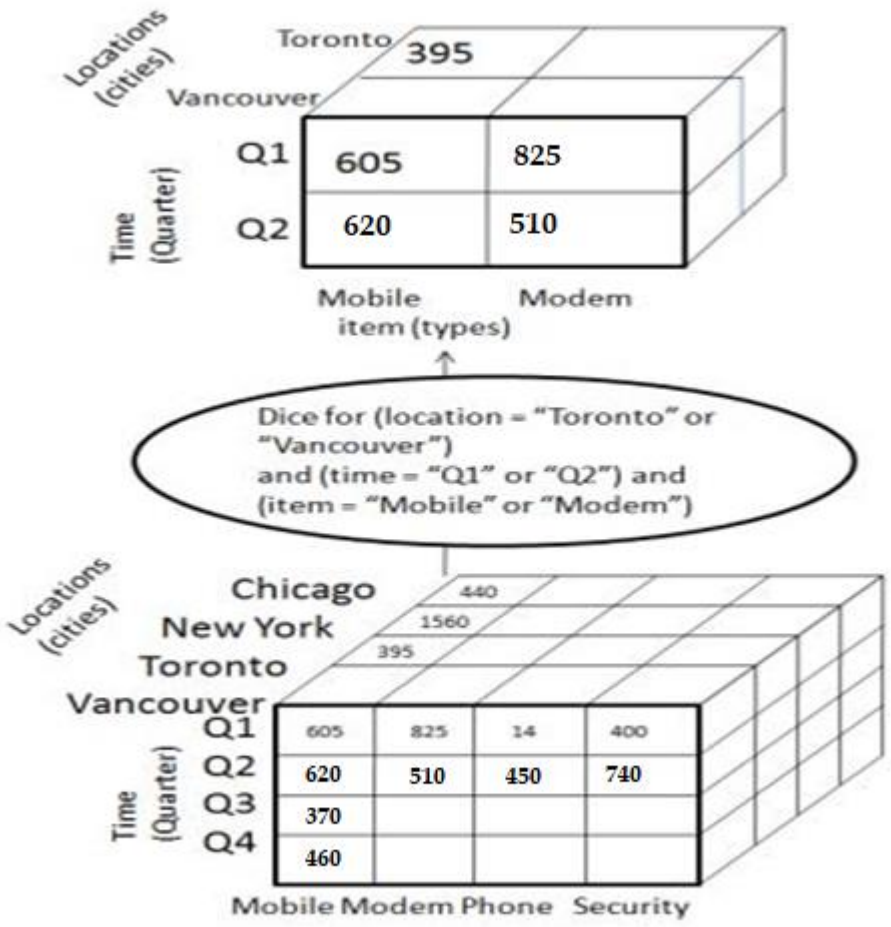
- The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.
- Here Slice is performed for the dimension "time" using the criterion time = "Q1". It will form a new sub-cube by selecting any one.



OLAP Operations

Dice

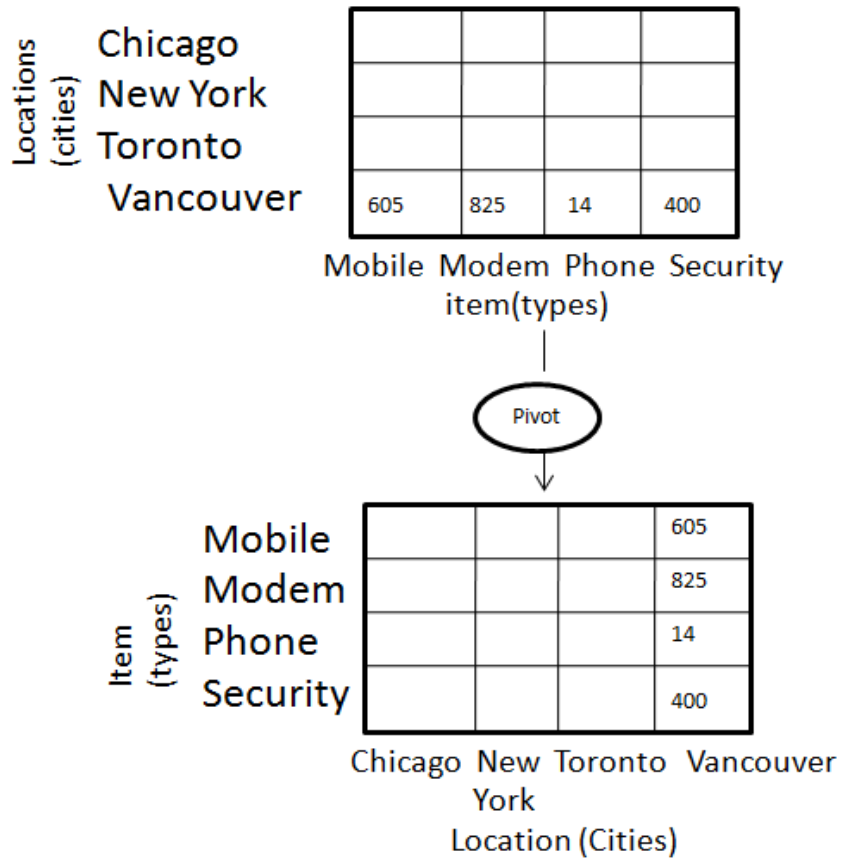
- Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.
- The dice operation on the cube based on the following selection criteria involves three dimensions: (location = "Toronto" or "Vancouver"), (time = "Q1"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



OLAP Operations

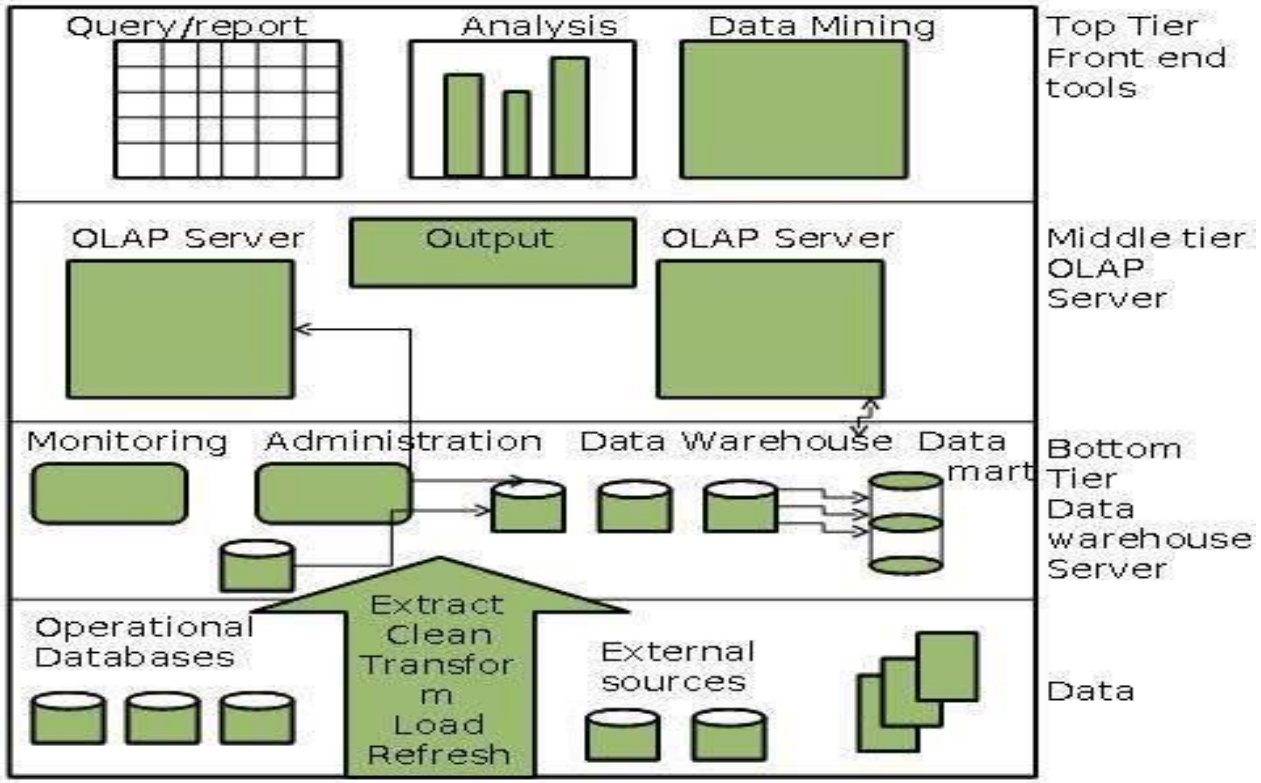
Pivot

- The pivot operation is also known as rotation. It rotates the data axes in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.
- Normally pivoting is performed after slicing.



Data Warehouse Architecture

- Generally a data warehouses adopts three-tier architecture: Bottom, Middle, and Top Tier.



Data Warehouse Architecture

Bottom Tier

- The bottom tier of the architecture is the data warehouse or database server.
- Back end tools and utilities are used to feed data into the bottom tier from operational databases and other data sources.
- These back end tools and utilities perform the Extract, Clean, Load, and Refresh functions.

Data Warehouse Architecture

Middle Tier

- In the middle tier, we have the OLAP Server that can be implemented in either of the following ways: *ROLAP and MOLAP*.
- By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
- By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.

Data Warehouse Architecture

Top-Tier

- This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.
- These tools are helpful in generating trend analysis, prediction, and so on.

Data Warehouse Architecture

Top-Tier

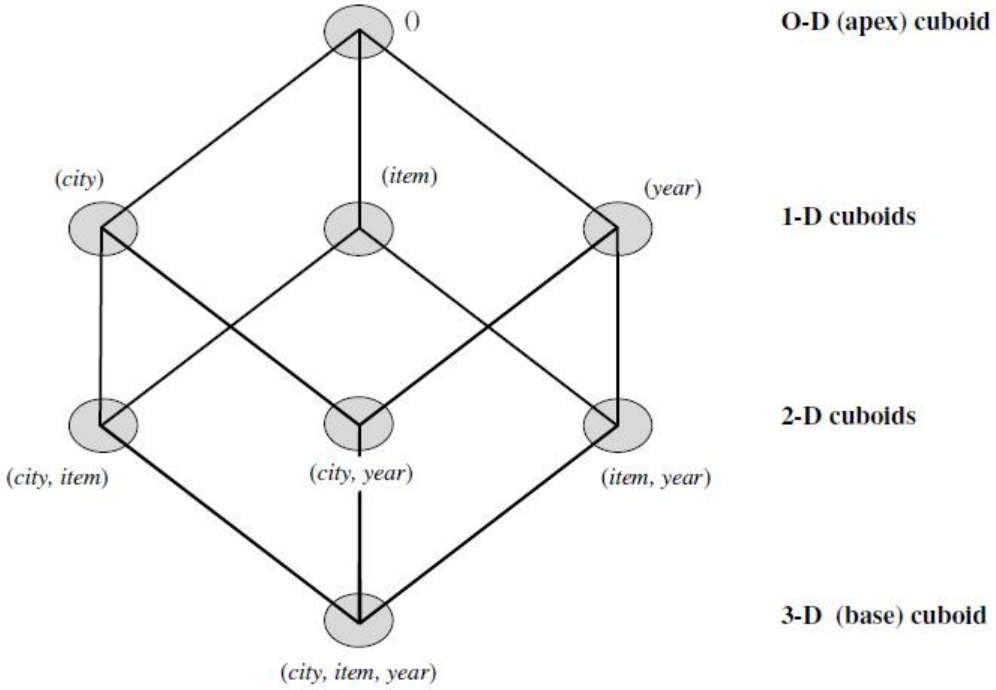
- This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.
- These tools are helpful in generating trend analysis, prediction, and so on.

Data Warehouse Implementation

- At the core of multidimensional data analysis is the efficient computation of aggregations across many sets of dimensions.
- In SQL terms, these aggregations are referred to as group-by's. Each group-by can be represented by a *cuboid*, where the set of group-by's forms a lattice of cuboids defining a data cube.
- One approach to cube computation extends SQL so as to include a **compute cube operator**.
- The compute cube operator computes aggregates over all subsets of the dimensions specified in the operation. This can require excessive storage space, especially for large numbers of dimensions.

Data Warehouse Implementation

- Suppose that we would like to create a data cube for *Electronics* sales that contains three dimension (*city, item, and year)* and *amount_sold as measure*. The total number of cuboids that can be computed for this data cube is $2^3 = 8$ as shown in lattice of cuboids.



Data Warehouse Implementation

- A statement such as: ***compute cube sales_cube*** would explicitly instruct the system to compute the sales aggregate cuboids for all of the eight subsets of the set {*city, item, year*}, including the empty subset.
- A major challenge related to pre-computation of cuboids is that the required storage space may explode if all of the cuboids in a data cube are pre-computed, especially when the cube has many dimensions.

Data Warehouse Implementation

- The storage requirements are even more excessive when many of the dimensions have associated concept hierarchies, each with multiple levels. This problem is referred to as the ***curse of dimensionality***.
- For an *n*-dimensional data cube with dimension characterized by concept-hierarchy, the total number of cuboids that can be generated is given by following formula.

$$Total\ number\ of\ cuboids = \prod_{i=1}^{n}(L_i + 1),$$

Where Li is the number of levels associated with dimension i

Data Warehouse Implementation

Example

Suppose that we have 10 dimensional data. What will be total number of cuboids generated? If we consider each dimension has 4 levels, what will be the number of cuboids generated?

Solution

Here n=10 Li=4 for i=1,2…….10

Without considering concept hierarchy,

Total number of cuboids=2^{10}=1024

After considering concept hierarchy

Total number of cuboids= 5×5×5×5×5×5×5×5×5×5=5^{10}=9765625

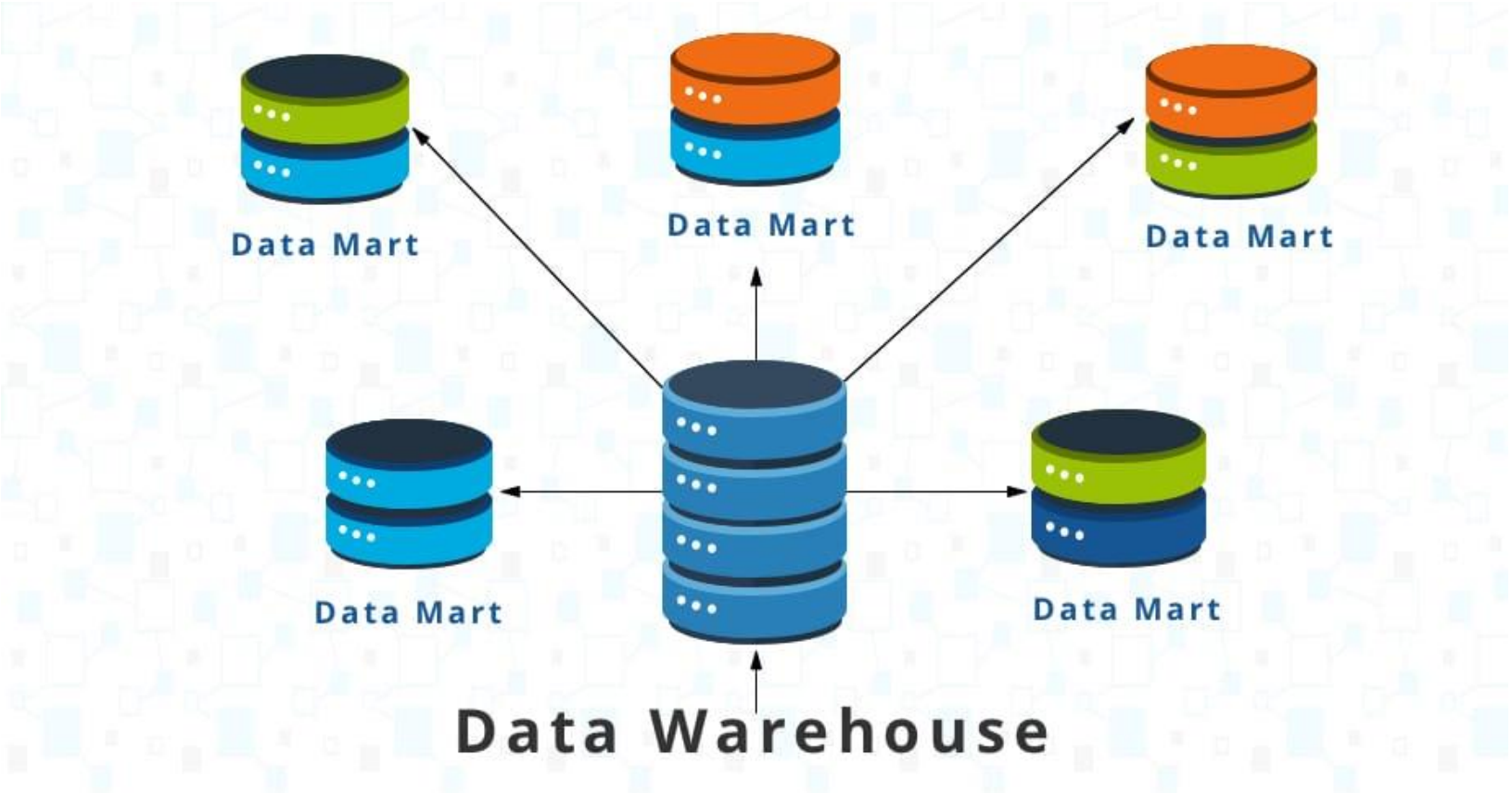
Data Mart

- A data mart is a subset of a data warehouse focused on a particular line of business, department, or subject area.
- Data marts make specific data available to a defined group of users, which allows those users to quickly access critical insights without wasting time searching through an entire data warehouse.
- For example, many companies may have a data mart that aligns with a specific department in the business, such as finance, sales, or marketing.

Data Mart

- Data warehouses are built to serve as the central store of data for the entire business, whereas a data mart fulfills the request of a specific division or business function.
- Because a data warehouse contains data for the entire company, it is best practice to have strictly control who can access it. Additionally, querying the data you need in a data warehouse is an incredibly difficult task for the business.
- Thus, the primary purpose of a data mart is to isolate—or partition—a smaller set of data from a whole to provide easier data access for the end consumers.

Data Mart


Data Mart
Data Mart
Data Mart
Data Mart
Data Mart
Data Warehouse

Components of Data Warehouse

- A typical data warehouse has four main components: a central database, ETL tools, metadata, and access tools.

Central Database

- The central data warehouse database is the cornerstone of the data warehousing environment. This database is traditionally implemented on the RDBMS technology.
- However, RDBMS products are optimized for transactional database processing. Certain data warehouse attributes, such as very large database size, need of computing aggregates, and drill-downs have become drivers for different technological approaches to the data warehouse database.

Components of Data Warehouse

ETL Tools

- A significant portion of the implementation effort is spent extracting data from operational systems and putting it in a format suitable for informational applications that run off the data warehouse. ETL (extract, transform, load) tools are used for this purpose.

Components of Data Warehouse

Meta data

- Meta data is data about data that describes the data warehouse. It is used for building, maintaining, managing and using the data warehouse. Technical meta data contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management. Business meta data contains information that gives users an easy-to-understand perspective of the information stored in the data warehouse.

Components of Data Warehouse

Meta data

- Meta data is data about data that describes the data warehouse. It is used for building, maintaining, managing and using the data warehouse. Technical meta data contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management. Business meta data contains information that gives users an easy-to-understand perspective of the information stored in the data warehouse.

Components of Data Warehouse

Data Warehouse Access Tools

- Access tools allow users to interact with the data in your data warehouse. Examples of access tools include: query and reporting tools, application development tools, data mining tools, and OLAP tools.

Need for Data Warehousing

Data warehouse is needed due to following reasons:

- To integrate data from multiple sources in one repository.
- To enable business users to view summarized data from different angle.
- To store historical data from past
- To help managers to make better decisions
- To reduce time needed for analysis and reporting

Trends in Data Warehousing

- **Multiple Data Types**: Different types of data needs to be integrated into data warehouse systems. These data types include structured numeric data, structured text, images, video, audio, spatial data, etc. Data warehouses need to provide facilities of searching all types of data stored.
- **Visualization Types**: Data warehouses must support various types of charts and interactive visualization system. Beside, modern applications needs 3D representations of visualization and mechanisms of summary at different levels.

Trends in Data Warehousing

- **Parallel Processing**: Analysts need to analyze large volume of data stored in data warehouse and need to produce results fast. Uniprocessor systems may not be sufficient in many cases therefore data warehouse systems need to support parallel processing. It can be achieved either by using parallel processor or by using parallel query processing technique.
- **Query Tools**: Data warehouses systems need to provide query tools to users so that users can specify task, provide feedback, and seek more explanation from the system . Such tools must be use friendly so that non-technical usesr can also use these tools easily.

Trends in Data Warehousing

- **Data Fusion**: It is the technology dealing with merging of data from disparate sources. It has wider scope and includes real-time merging of data from instruments and monitoring systems.
- **Software Agents**: Software agent is program that is executed in certain environment autonomously and is capable of making decisions based on data obtained from the environment and from other agents. Such agents needs to be integrated into data warehouse systems to provide alerts about predefined business conditions to users.