
# University of Technology
# الجامعة التكنولوجية

## Computer Science Department
## قسم علوم الحاسوب

# Data Warehouse
# مخازن البيانات

Asst. Prof. Dr. Khalil I. Ghathwan
أ.م.د. خليل ابراهيم غثوان


cs.uotechnology.edu.iq

# The Contents

- Data Warehousing Definitions and Concepts.
- Data Warehousing Process Overview.
- Data Warehousing Architectures.
- Data Integration and the Extraction, Transformation, and Load (ETL) Process.
- Data Warehouse Development.
- Real-Time Data Warehousing.
- Data Warehouse Administration and Security Issues.

# Chapter One

# Data Warehousing Definitions and Concepts.

## 1 Introduction

The term "Data Warehouse" was first coined by Bill Inmon in 1990. According to Inmon, a data warehouse is a subject oriented, integrated, time-variant, and non-volatile collection of data.

This data helps analysts to take informed decisions in an organization. An operational database undergoes frequent changes on a daily basis on account of the transactions that take place. Suppose a business executive wants to analyze previous feedback on any data such as a product, a supplier, or any consumer data, then the executive will have no data available to analyze because the previous data has been updated due to transactions.

A data warehouses provides us generalized and consolidated data in multidimensional view. Along with generalized and consolidated view of data, a data warehouses also provides us **Online Analytical Processing OLAP** tools. These tools help us in interactive and effective analysis of data in a multidimensional space. This analysis results in data generalization and data mining. Data mining functions such as association, clustering, classification, prediction can be integrated with OLAP operations to enhance the interactive mining of knowledge at multiple level of abstraction. That's why data warehouse has now become an important platform for data analysis and online analytical processing.

## 1.1 Understanding a Data Warehouse

1. A data warehouse is a database, which is kept separate from the organization's operational database.

2. There is no frequent updating done in a data warehouse.
3. It possesses consolidated historical data, which helps the organization to analyze its business.
4. A data warehouse helps executives to organize, understand, and use their data to take strategic decisions.
5. Data warehouse systems help in the integration of diversity of application systems.
6. A data warehouse system helps in consolidated historical data analysis.

## 1.2 Why a Data Warehouse is Separated from Operational Databases

A data warehouses is kept separate from operational databases due to the following reasons:

1. An operational database is constructed for well-known tasks and workloads such as searching particular records, indexing, etc. In contract, data warehouse queries are often complex and they present a general form of data.

2. Operational databases support concurrent processing of multiple transactions. Concurrency control and recovery mechanisms are required for operational databases to ensure robustness and consistency of the database.

3. An operational database query allows to read and modify operations, while an OLAP query needs only read only access of stored data.

4. An operational database maintains current data. On the other hand, a data warehouse maintains historical data.

## 1.3 Data Warehouse Features

The key features of a data warehouse are discussed below:

1. **Subject Oriented** - A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. These subjects can be product, customers, suppliers, sales, revenue, etc. A data warehouse does not focus on the ongoing operations, rather it focuses on modelling and analysis of data for decision making.

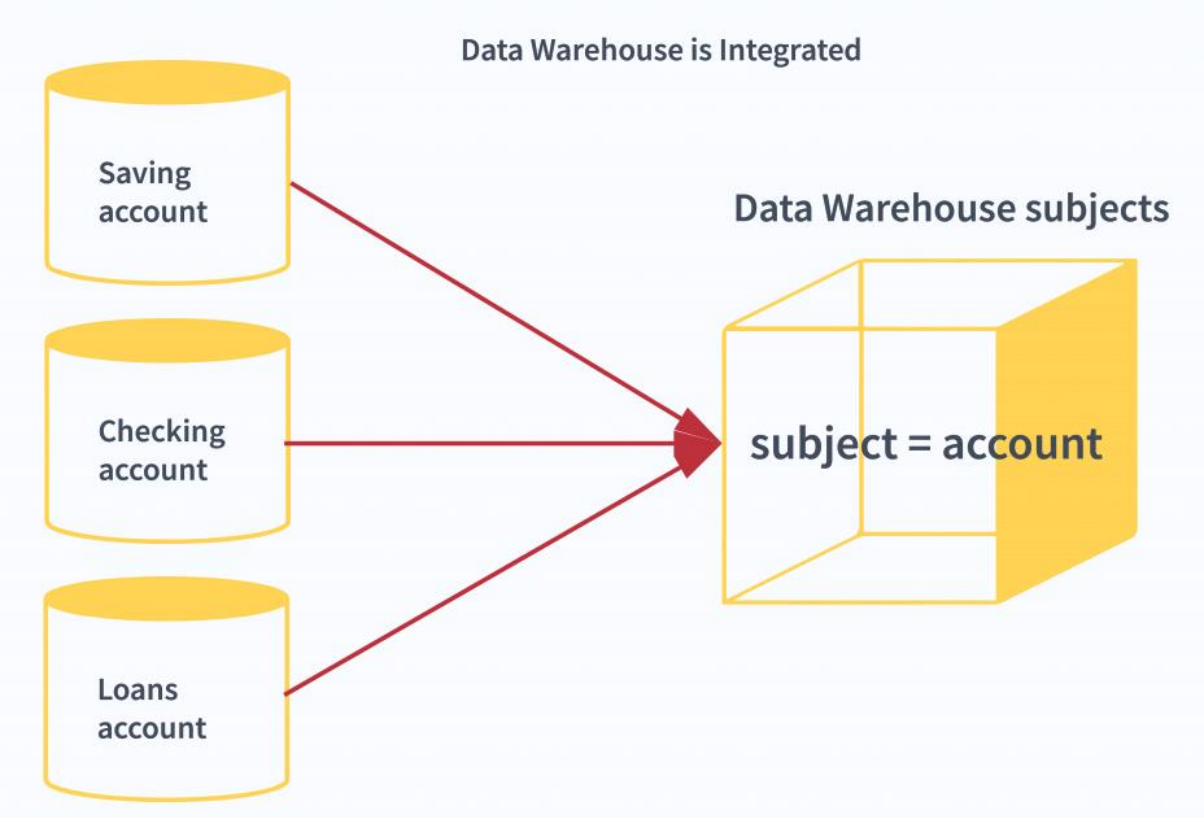
**Subject-oriented**

The data collected relate to a particular subject (e.g. sales) and not actions (e.g. servicing requests)

2. **Integrated** - A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. This integration enhances the effective analysis of data.

**Integrated**

Data has been standarized regardless of how it is stored in the source systems

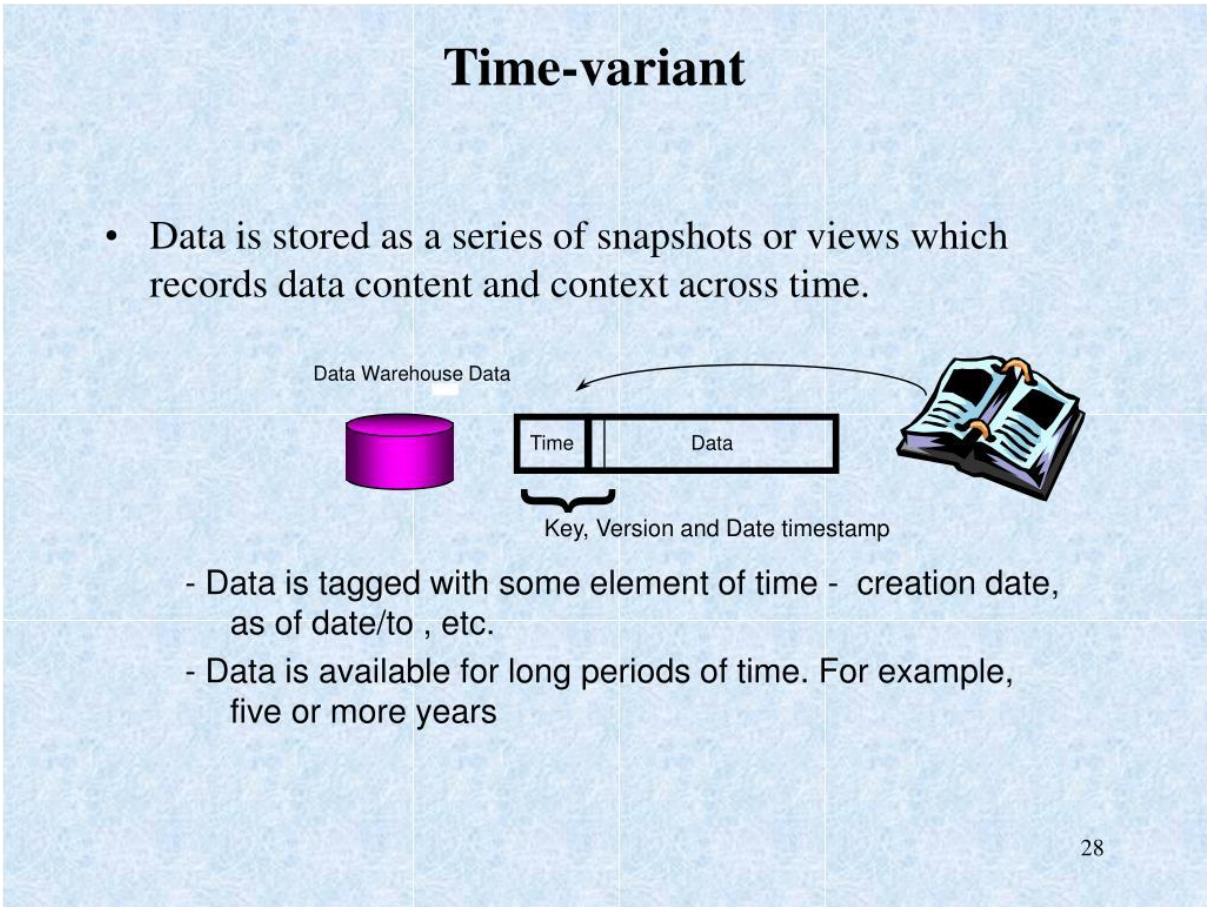



For example, gender field might be represented differently from one transactional system to another. i.e. X/Y in human resources system or 0/1 in the salary system, while the data warehouse has one consistent format like M/F as illustrated in figure

| OLTP systems | | DW system |
|---|---|---|
| Gender in Human Resources (HR) system: X/Y | Gender in Salary System: 0/1 | Gender in data Warehouse (DW) system: M/F |

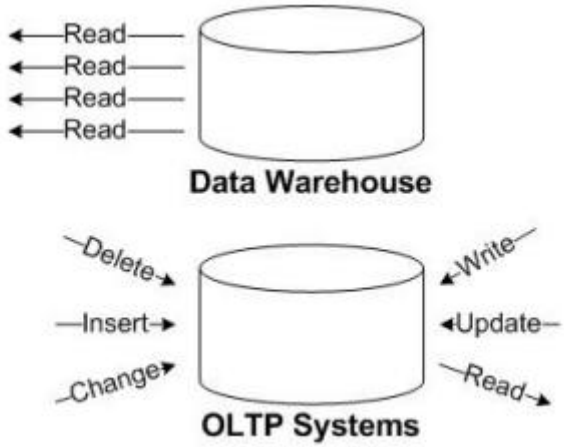
3. **Time Variant** - The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical point of view. Operational systems store current up-to-date data for fast and atomic transactions. Old data transferred to archives since it affects performance, while the data warehouse contains a large amount of historical

information that helps the analysts to discover trends in business. Every entry in the data warehouse is time-stamped through timedimension. Noteworthy, data in OLTP systems maintained for few months or less than a year, while DW data kept for over three years (3 - 10).




4. **Non-volatile** - Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

   Note: A data warehouse does not require transaction processing, recovery, and concurrency controls, because it is physically stored and separate from the operational database.




The final crucial characteristic of the data warehouse is Non-Volatile, data inside operational systems are updated (Insert, Delete, or Read and Write) also known as “CRUD”. While, data warehouse data are considered static or not up-to-date but only for Read. DW refreshed from time to time in a batch mode. This property enables DW to be heavily optimized for query processing; figure illustrates a simple comparison of this feature between operational and informational systems.

## 1.4 Data Warehouse Applications

As discussed before, a data warehouse helps business executives to organize, analyze, and use their data for decision making. A data warehouse serves as a sole part of a plan-execute-assess "closed-loop" feedback system for the enterprise management. Data warehouses are widely used in the following fields:

- Financial services
- Banking services
- Consumer goods
- Retail sectors
- Controlled manufacturing

## 1.5 Types of Data Warehouse

Information processing, analytical processing, and data mining are the three types of data warehouse applications that are discussed below:

1. **Information Processing** - A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.

2. **Analytical Processing** - A data warehouse supports analytical processing of the information stored in it. The data can be analyzed by means of basic OLAP operations, including sliceand-dice, drill down, drill up, and pivoting.

3. **Data Mining** - Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.

## 1.6 Differences between Operational Database Systems and Data Warehouses

The major task of online operational database systems is to perform online transaction and query processing. These systems are called online transaction processing (OLTP) systems. They cover most of the day-to-day operations of an organization such as purchasing, inventory, banking, payroll, registration, and accounting. Data warehouse systems, on the other hand, serve users or knowledge workers in the role of data analysis and decision making. These systems are known as online analytical processing (OLAP) systems. The major distinguishing features of OLTP and OLAP are summarized as follows:

1. **Users and system orientation**: An OLTP system is application-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is subject-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.

2. **Data contents**: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making, while OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, these features make the data easier to use for informed decision making.

3. **Database design**: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

4. **View**: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores.

5. **Access patterns**: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However, accesses to OLAP systems are mostly read-only operations (because most data warehouses store historic rather than up-to-date information), although many could be complex queries. Other features that distinguish between OLTP and OLAP systems include database size, frequency of operations, and performance metrics. These are summarized in Table.

| Sr.No. | Data Warehouse OLAP | Operational Database OLTP |
|---|---|---|
| **1** | It involves historical processing of information. | It 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** | It is used to analyze the business. | It is used to run the business. |
| **4** | It focuses on Information out. | It focuses on Data in. |
| **5** | It is based on Star Schema, Snowflake Schema, and Fact Constellation Schema. | It is based on Entity Relationship Model. |
| **6** | It focuses on Information out. | It is application oriented. |
| **7** | It contains historical data. | It contains current data. |
| **8** | It provides summarized and consolidated data. | It provides primitive and highly detailed data. |
| **9** | It provides summarized and multidimensional view of data. | It provides detailed and flat relational view of data. |
| **10** | The number of users is in hundreds. | The number of users is in thousands. |
| **11** | The number of records accessed is in millions. | The number of records accessed is in tens. |
| **12** | The database size is from 100GB to 100 TB. | The database size is from 100 MB to 100 GB. |
| **13** | These are highly flexible. | It provides high performance. |

# Chapter Two

# Data Warehousing Process Overview

## 2.1 A Multidimensional Data Model

Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a data cube. In this section, you will learn how data cubes model n-dimensional data. Various multidimensional models are shown: star schema, snowflake schema, and fact constellation.

### 2.1.1 Data Cube

A data cube helps us represent data in multiple dimensions. It is defined by dimensions and facts. The dimensions are the entities with respect to which an enterprise preserves the records.

### 2.1.2 Illustration of Data Cube

Suppose a company wants to keep track of sales records with the help of sales data warehouse with respect to time, item, branch, and location. These dimensions allow to keep track of monthly sales and at which branch the items were sold. There is a table associated with each dimension. This table is known as dimension table.

Suppose a company, AllElectronics 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.

These dimensions allow the store to keep track of things like monthly sales of items and the branches and locations at which the items were sold. Each dimension may have a table associated with it, called a dimension table.

For example, a dimension table for item may contain the attributes item name, brand, and type. Dimension tables can be specified by users or experts, or automatically generated and adjusted based on data distributions.

A multidimensional data model is typically organized around a central theme, such as sales. This theme is represented by a fact table. Facts are numeric measures. Think of them as the quantities by which we want to analyze relationships between dimensions.

Examples of facts for a sales data warehouse include dollars sold (sales amount in dollars) and units sold (number of units sold). The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.

We usually think of cubes as 3-D geometric structures, in data warehousing the data cube is n-dimensional. To gain a better understanding of data cubes and the multidimensional data model, let's start by looking at a simple 2-D data cube that is, in fact, a table or spreadsheet for sales data from AllElectronics. In particular, we will look at the AllElectronics sales data for items sold per quarter in the city of Vancouver.

These data are shown in Table. In this 2-D representation, the sales for Vancouver are shown with respect to the time dimension (organized in quarters) and the item dimension (organized according to the types of items sold). The fact or measure displayed is dollars sold (in thousands).

| | ***location*** = *"Vancouver"* | | | |
|---|---|---|---|---|
| | ***item*** *(type)* | | | |
| ***time*** *(quarter)* | *home entertainment* | *computer* | *phone* | *security* |
| Q1 | 605 | 825 | 14 | 400 |
| Q2 | 680 | 952 | 31 | 512 |
| Q3 | 812 | 1023 | 30 | 501 |
| Q4 | 927 | 1038 | 38 | 580 |

*Note:* The sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

Now, suppose that we would like to view the sales data with a third dimension. For instance, suppose we would like to view the data according to time and item, as well as location, for the cities Chicago, New York, Toronto, and Vancouver. These 3-D data are shown in Table 3. The 3-D data in the table are represented as a series of 2-D tables. Conceptually, we may also represent the same data in the form of a 3-D data cube, as in Figure Table . 3-D View of Sales Data for AllElectronics According to time, item, and location.

| | ***location*** = "Chicago" | | | | ***location*** = "New York" | | | | ***location*** = "Toronto" | | | | ***location*** = "Vancouver" | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ***item*** | | | | ***item*** | | | | ***item*** | | | | ***item*** | | | |
| | *home* | | | | *home* | | | | *home* | | | | *home* | | | |
| ***time*** | *ent.* | *comp.* | *phone* | *sec.* | *ent.* | *comp.* | *phone* | *sec.* | *ent.* | *comp.* | *phone* | *sec.* | *ent.* | *comp.* | *phone* | *sec.* |
| Q1 | 854 | 882 | 89 | 623 | 1087 | 968 | 38 | 872 | 818 | 746 | 43 | 591 | 605 | 825 | 14 | 400 |
| Q2 | 943 | 890 | 64 | 698 | 1130 | 1024 | 41 | 925 | 894 | 769 | 52 | 682 | 680 | 952 | 31 | 512 |
| Q3 | 1032 | 924 | 59 | 789 | 1034 | 1048 | 45 | 1002 | 940 | 795 | 58 | 728 | 812 | 1023 | 30 | 501 |
| Q4 | 1129 | 992 | 63 | 870 | 1142 | 1091 | 54 | 984 | 978 | 864 | 59 | 784 | 927 | 1038 | 38 | 580 |

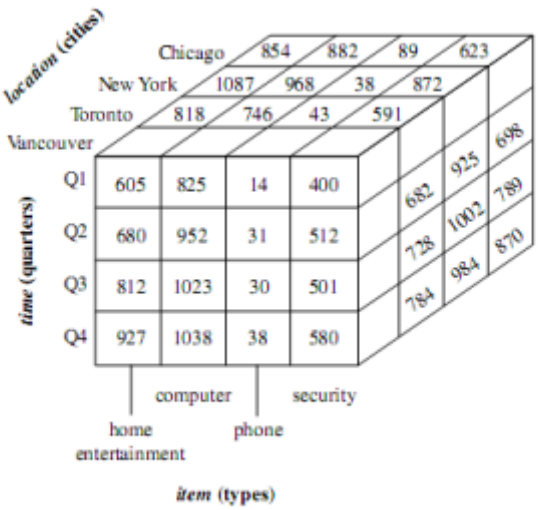



Figure . A 3-D data cube representation of the data in Table 3, according to time, item, and location.

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 .

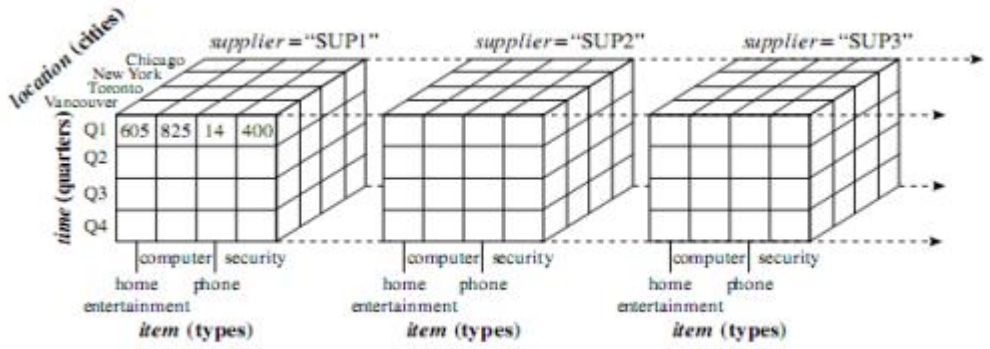
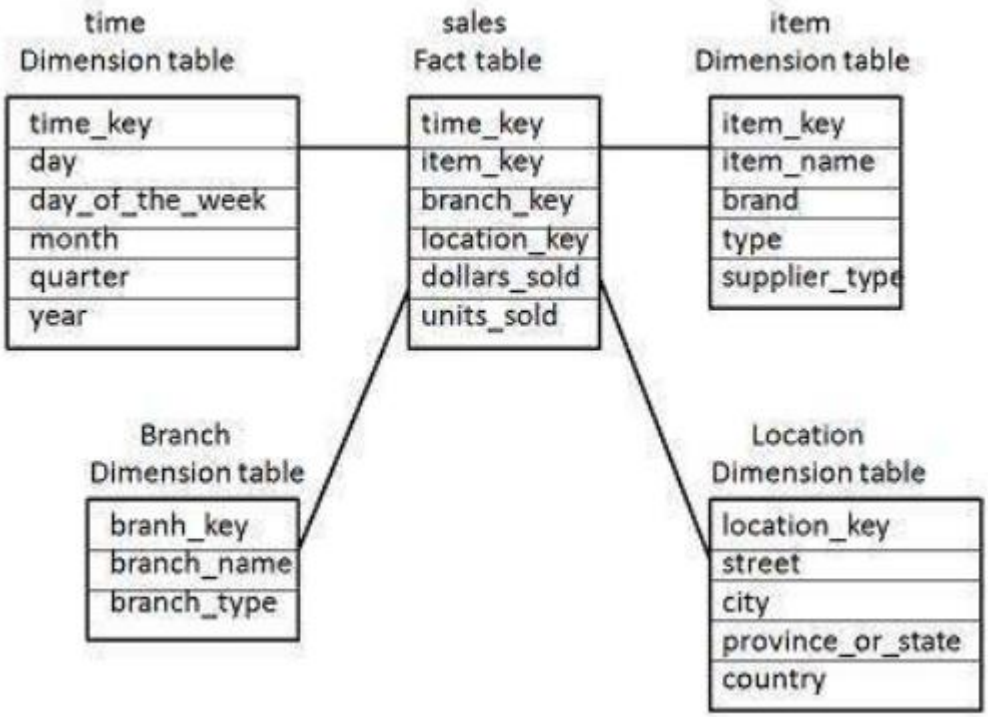



Figure. A 4-D data cube representation of sales data, according to time, item, location, and supplier. For improved readability, only some of the cube values are shown.

## 2.2 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.

1. **Star schema**: The most common modeling paradigm is the star schema, in which the data warehouse contains
   - 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.

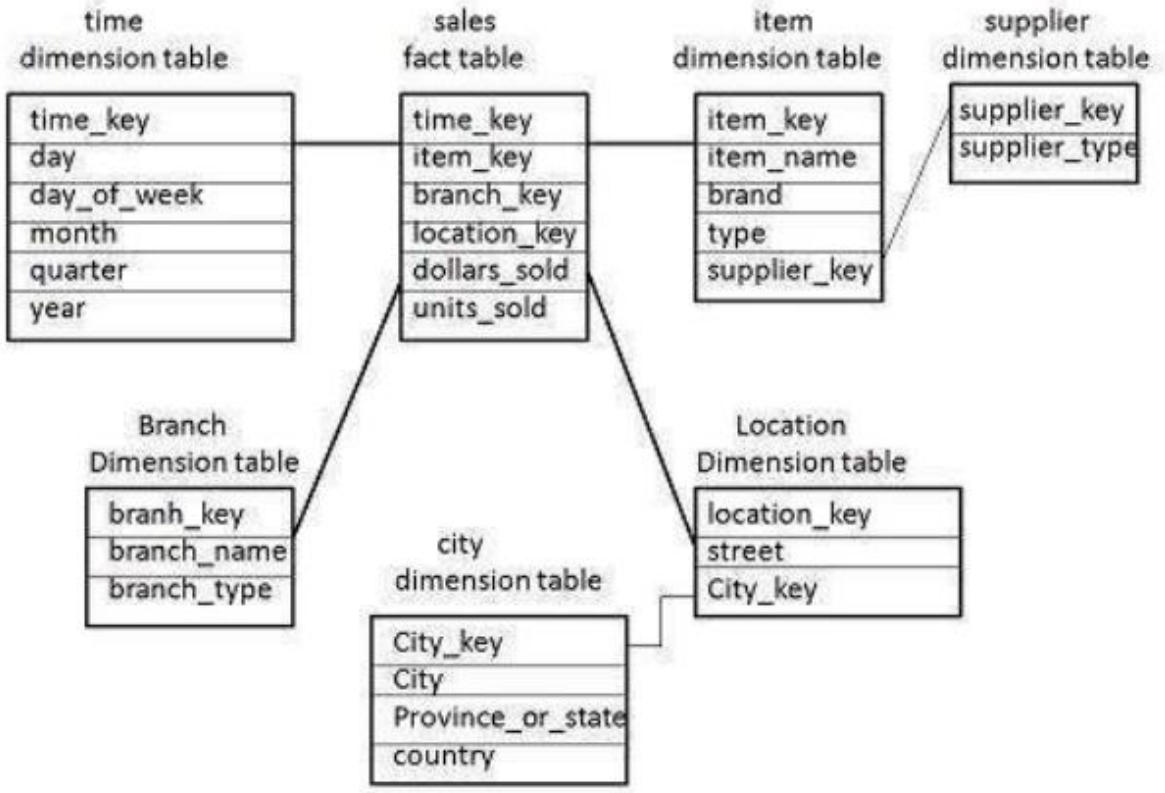
For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy.

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.

## 2. Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.

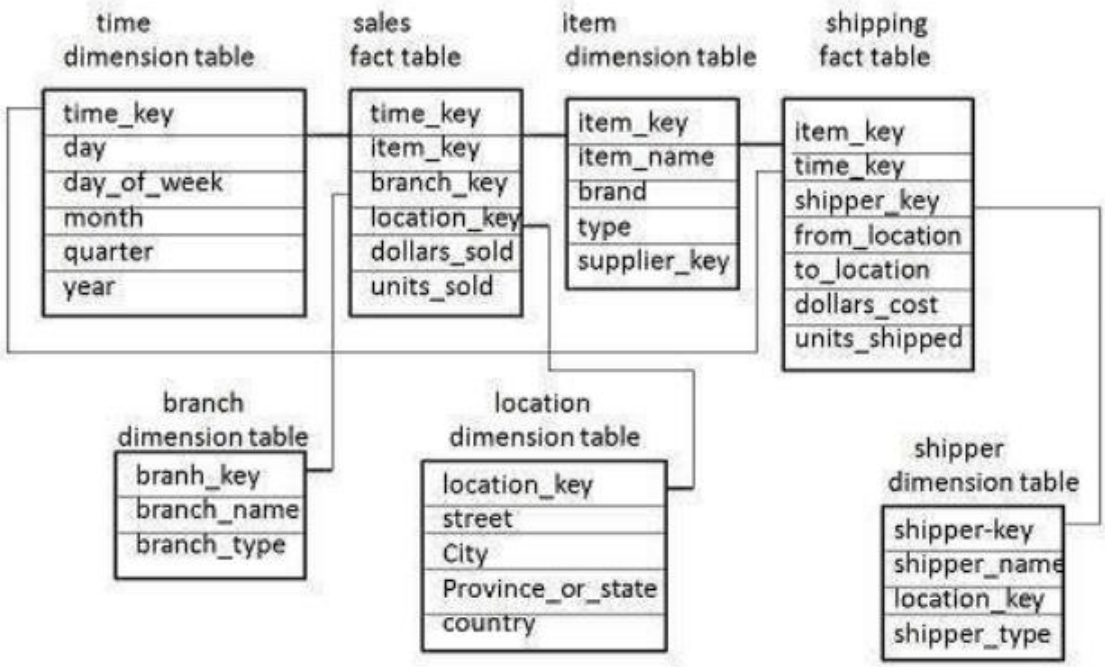
- Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.
- Now the item dimension table contains the attributes item_key, item_name, type, brand, 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.

### 3. 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.