Unit 9: Data Warehousing

A data warehouse is a database designed to enable business intelligence activities, it exists to help users understand and enhance their organization's performance. It is designed for query and analysis rather than for transaction processing, and usually contains historical data derived from transaction data, but can include data from other sources. Data warehouses separate analysis workload from transaction workload and enable an organization to consolidate data from several sources. This helps in:

- Maintaining historical records
- Analyzing the data to gain a better understanding of the business and to improve the business

To achieve the goal of enhanced business intelligence, the data warehouse works with data collected from multiple sources. The source data may come from internally developed systems, purchased applications, third-party data and other sources. It may involve transactions, production, marketing, human resources and more.

Data warehouses provide on-line analytical processing (OLAP) [refer unit 2 for OLAP] tools for the interactive analysis of multidimensional data, which facilitates effective data generalization and data mining. Many other data mining functions, such as association [refer unit 4 for detail], classification [refer unit 3 for detail], prediction, and clustering [refer unit 5 for detail], can be integrated with OLAP operations to enhance interactive mining of knowledge at multiple levels of abstraction. Hence, the data warehouse has become an increasingly important platform for data analysis and on-line analytical processing and will provide an effective platform for data mining. Therefore, data warehousing and OLAP form an essential step in the knowledge discovery process.

"**A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process**" **Subject-oriented:** A data warehouse is organized around major subjects, such as customer, supplier, product, and sales. Rather than concentrating on the day-to-day operations and transaction processing of an organization, a data warehouse focuses on the modeling and analysis of data for decision makers. Hence, data warehouses typically provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Integrated: A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

Time-variant: Data are stored to provide information from a historical perspective (e.g., the past 5–10 years).

Nonvolatile: A data warehouse is always a physically separate store of data transformed from the application data found in the operational environment. Due to this separation, a data warehouse does not require transaction processing, recovery, and concurrency control mechanisms. It usually requires only two operations in data accessing: initial loading of data and access of data.

Differences between Operational Database Systems and Data Warehouses

In most of the organization, there occur large databases in operation for normal daily transactions called **operational database**. A data warehouse is a large database built from the operational database.

	Data Warehouse	Database
Purpose	Analysis, Decision making	Day to day use
Support For	OLAP(on-line analytical processing)	OLTP(on-line transaction processing)
Data model	Multi-dimentional	Rational
Age of data	Current & time series	Current & real time
Data modification	Read/access only	Insert, update, delete
Type of data	Static	Dynamic
Amount of data per transaction	Larger	Smaller

A data mart serves the same role as a data warehouse, but it is intentionally limited in scope. It may serve one particular department or line of business. The advantage of a data mart versus a data warehouse is that it can be created much faster due to its limited coverage.

Operational data stores (ODS) exist to support daily operations. The ODS data is cleaned and validated, but it is not historically deep, it may be just the data for the current day. The ODS may also be used as a source to load the data warehouse.

ETL (Extract, Transform, Load)

A data warehouse usually stores many years of data to support historical analysis. The data in a data warehouse is typically loaded through an extraction, transformation, and loading (ETL) process from multiple data sources. Modern data warehouses are moving toward an extract, load, and transformation (ELT) architecture in which all or most data transformation is performed on the database that hosts the data warehouse.

[Note: Refer Unit 1(Architecture of data warehouse) for more details]

Data Warehouse Processes, Managers and their functions

Load manager performs the operations required to extract and load the data into the database. The size and complexity of a load manager varies between specific solutions from one data warehouse to another.

Load Manager

The load manager does perform the following functions:

- Extract data from the source system.
- Fast load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.

Extract Data from Source

The data is extracted from the operational databases or the external information providers. Gateways are the application programs that are used to extract data. It is supported by underlying DBMS and allows the client program to generate SQL to be executed at a server. Open Database Connection (ODBC) and Java Database Connection (JDBC) are examples of gateway.

Simple Transformations

While loading, it may be required to perform simple transformations. After completing simple transformations, we can do complex checks. Suppose we are loading the sales transaction, we need to perform the following checks:

- Strip out all the columns that are not required within the warehouse.
- Convert all the values to required data types.

Warehouse Manager

The warehouse manager is responsible for the warehouse management process. It consists of a third-party system software, C programs, and shell scripts. The size and complexity of a warehouse manager varies between specific solutions.

Functions of Warehouse Manager

A warehouse manager performs the following functions:

- Analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates the existing aggregations.
- Generates normalizations.
- Transforms and merges the source data of the temporary store into the published data warehouse.
- Backs up the data in the data warehouse.
- Archives the data that has reached the end of its captured life.

Query Manager

The query manager is responsible for directing the queries to suitable tables. By directing the queries to appropriate tables, it speeds up the query request and response process. In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.

Functions of Query Manager

- It presents the data to the user in a form they understand.
- It schedules the execution of the queries posted by the end-user.
- It stores query profiles to allow the warehouse manager to determine which indexes and aggregations are appropriate.

Data Warehouses Design

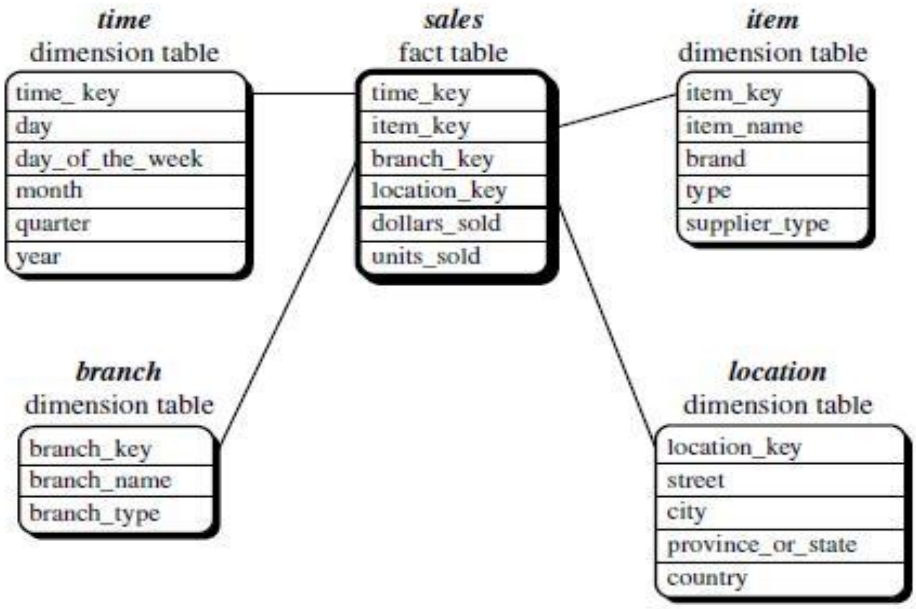
The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing.

A data warehouse requires a concise, subject-oriented schema that facilitates on-line data analysis. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a **star schema, a snowflake schema, or a fact constellation schema**.

Star schema: The most common modeling paradigm is the star schema, in which the data warehouse contains (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and (2) a set of smaller attendant tables (dimension tables), one for each dimension.

Example: A star schema for AllElectronics sales is shown in Figure below. Sales are considered along four dimensions, namely: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold. To minimize the size of the fact table, dimension identifiers (such as time key and item key) are system-generated identifiers.

Notice that in the star schema, each dimension is represented by only one table, and each table contains 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 introduce some redundancy. For example, "Vancouver" and "Victoria" are both cities in the Canadian province of British Columbia. Entries for such cities in the location dimension table will create redundancy among the attributes province or state and country, that is, (..., Vancouver, British Columbia, Canada) and (..., Victoria, British Columbia, Canada).Moreover, the attributes within a dimension table may form either a hierarchy (total order) or a lattice (partial order).

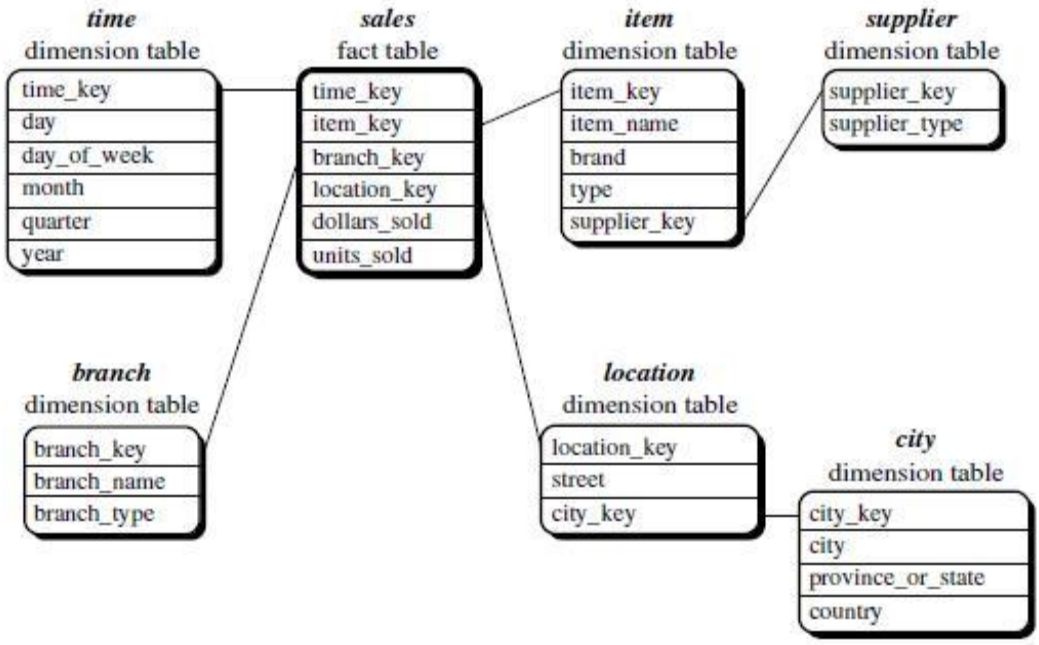


Snowflake schema: The snowflake schema is a variant of the star schema model, where

some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

Example: A snowflake schema for AllElectronics sales is given in Figure below

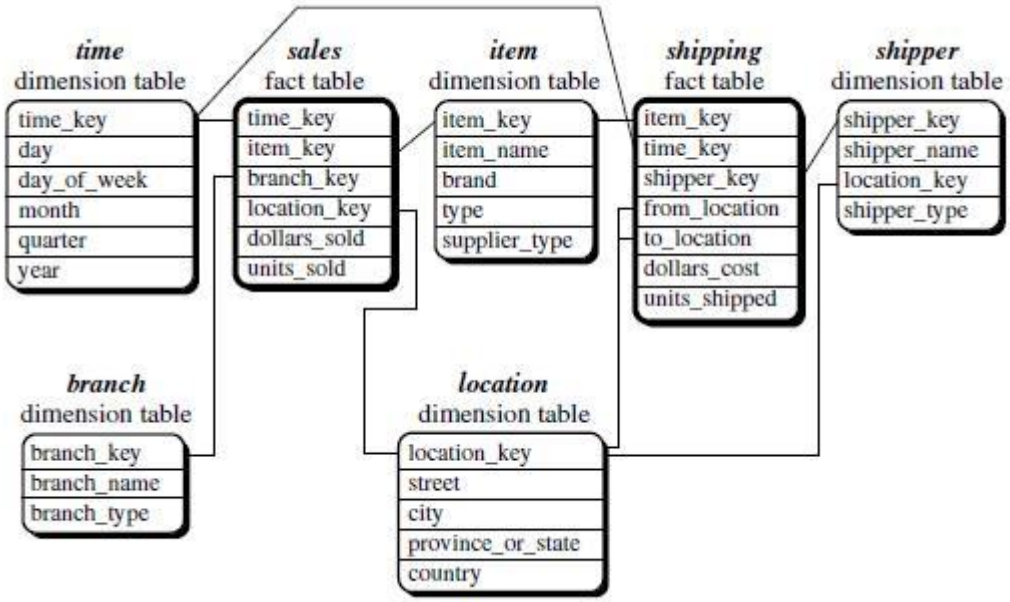
Here, the sales fact table is identical to that of the star schema in Figure above. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item key, item name, brand, type, and supplier key, where supplier key is linked to the supplier dimension table, containing supplier key and supplier type information. Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city key in the new location table links to the city dimension.



Note: The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies. Such a table is easy to maintain 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.

Fact constellation: 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.

Example: A fact constellation schema is shown in Figure below. This 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. 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.



GUIDELINES FOR DATA WAREHOUSE IMPLEMENTATION Implementation steps

1. Requirements analysis and capacity planning: In the first step in data warehousing involves defining enterprise needs, defining architecture, carrying out capacity planning and selecting the hardware and software tools. This step will involve consulting senior management as well as the various stakeholders.

2. Hardware integration: Once the hardware and software have been selected, they need to be put together by integrating the servers, the storage devices and the client software tools.

3. Modeling: Modeling is a major step that involves designing the warehouse schema and views. This may involve using a modeling tool if the data warehouse is complex.

4. Physical modeling: For the data warehouse to perform efficiently, physical modeling is required. This involves designing the physical data warehouse organization, data placement, data partitioning, deciding on access methods and indexing.

5. Sources: The data for the data warehouse is likely to come from a number of data sources. This step involves identifying and connecting the sources using gateways, ODBC drives or other wrappers.

6. ETL: The data from the source systems will need to go through an ETL process. The step of designing and implementing the ETL process may involve identifying a suitable ETL tool vendor and purchasing and implementing the tool. This may include customizing the tool to suit the needs of the enterprise.

7. Populate the data warehouse: Once the ETL tools have been agreed upon, testing the tools will be required. Once everything is working satisfactorily, the ETL tools may be used in populating the warehouse given the schema and view definitions.

8. User applications: For the data warehouse to be useful there must be end-user applications. This step involves designing and implementing applications required by the end users.

9. Roll-out the warehouse and applications: Once the data warehouse has been populated and the end-user applications tested, the warehouse system and the applications may be rolled out for the user community to use.

Implementation Guidelines

1. Build incrementally: Data warehouses must be built incrementally. Generally it is recommended that a data mart may first be built with one particular project in mind and once it is implemented a number of other sections of the enterprise may also wish to implement similar systems. An enterprise data warehouse can then be implemented in an iterative manner allowing all data marts to extract information from the data warehouse. Data warehouse modeling itself is an iterative methodology as users become familiar with the technology and are then able to understand and express their requirements more clearly.

2. Need a champion: A data warehouse project must have a champion who is willing to carry out considerable research into expected costs and benefits of the project. Data warehousing projects require inputs from many units in an enterprise and therefore need to be driven by someone who is capable of interaction with people in the enterprise and can actively persuade colleagues. Without the cooperation of other units, the data model for the

warehouse and the data required to populate the warehouse may be more complicated than they need to be. Studies have shown that having a champion can help adoption and success of data warehousing projects.

3. Senior management support: A data warehouse project must be fully supported by the senior management. Given the resource intensive nature of such projects and the time they can take to implement, a warehouse project calls for a sustained commitment from senior management. This can sometimes be difficult since it may be hard to quantify the benefits of data warehouse technology and the managers may consider it a cost without any explicit return on investment. Data warehousing project studies show that top management support is essential for the success of a data warehousing project.

4. Ensure quality: Only data that has been cleaned and is of a quality that is understood by the organization should be loaded in the data warehouse. The data quality in the source systems is not always high and often little effort is made to improve data quality in the source systems. Improved data quality, when recognized by senior managers and stakeholders, is likely to lead to improved Support for a data warehouse project.

5. Corporate strategy: A data warehouse project must fit with corporate strategy and business objectives. The objectives of the project must be clearly defined before the start of the project. Given the importance of senior management support for a data warehousing project, the fitness of the project with the corporate strategy is essential.

6. Business plan: The financial costs (hardware, software, and HR), expected benefits and a project plan (including an ETL plan) for a data warehouse project must be clearly outlined and understood by all stakeholders. Without such understanding, rumors about expenditure and benefits can become the only source of information, undermining the project.

7. Training: A data warehouse project must not overlook data warehouse training requirements. For a data warehouse project to be successful, the users must be trained to use the warehouse and to understand its capabilities. Training of users and professional development of the project team may also be required since data warehousing is a complex task and the skills of the project team are critical to the success of the project.

8. Adaptability: The project should build in adaptability so that changes may be made to the data warehouse if and when required. Like any system, a data warehouse will need to change, as needs of an enterprise change. Furthermore, once the data warehouse is operational, new

applications using the data warehouse are almost certain to be proposed. The system should be able to support such new applications.

9. Joint management: The project must be managed by both IT and business professionals in the enterprise. To ensure good communication with the stakeholders and that the project is focused on assisting the enterprise's business, business professionals must be involved in the project along with technical professionals.