DATA WEAR HOUSE

Data wear house

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources.

DATA EAR HOUSE AND DATA BASE REALATION

In addition to a relational database, a data warehouse environment includes an extraction, transportation a transformation, and an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users

.Reasons for building Data warehouse

1. Improving integration
2. Speeding up response times
3. . Faster and more flexible reporting
4. Recording changes to build history
5. Increasing data quality
6. Unburdening the IT department
7. . Increasing recognisability
8. . Increasing findability

Granularity

Granularity refers to the level of detail or summarization of the units of data in the data warehouse. The more detail there is, the lower the level of granularity. The less detail there is, the higher the level of granularity. Granularity is the major design issue in the data warehouse environment because it profoundly affects the volume of data that resides in the data warehouse and the type of query that can be answered. The volume of data in a warehouse is traded off against the level of detail of a query. In almost all cases, data comes into the data warehouse at too high a level of granularity. This means that the developer must spend a lot of resources breaking the data apart. Occasionally, though, data enters the warehouse at too low a level of granularity. An example of data at too low a level of granularity is the Web log data generated by the Web-based e-business environment. Web log click stream data must be edited, filtered, and summarized before its granularity is fit for the data warehouse environment

The Benefits of Granularity

- Reusability.

- Ability to reconcile data

. • Flexibility.

- It contains a history of activities and events across the corporation

Differences between Operational Database Systems and Data Warehouses

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.

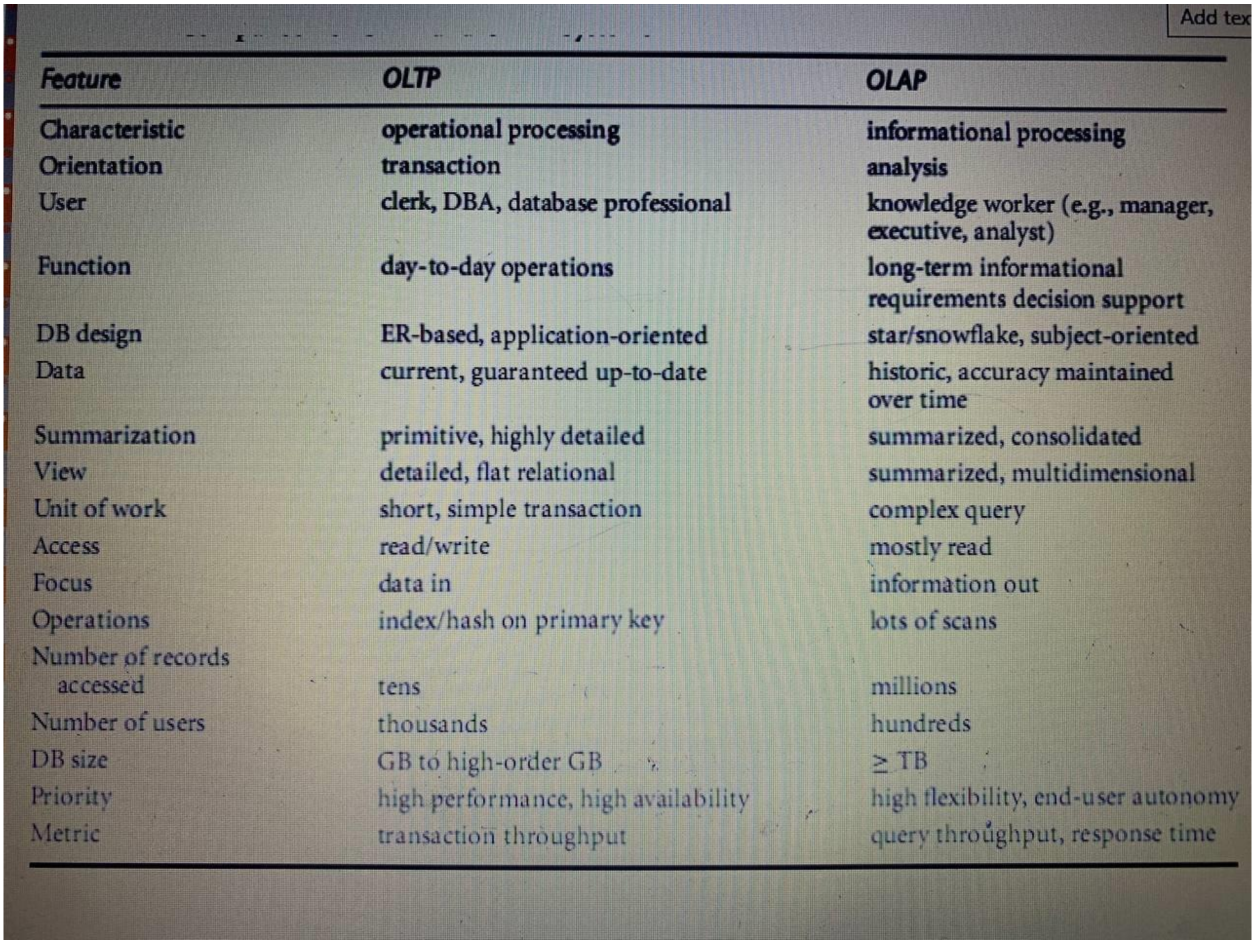
3. Database design: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

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.

Comparison of OLTP and OLAP Systems

فقط خمس نقاط



Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
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	historic, 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
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	GB to high-order GB	≥ TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

A Multidimensional Data Mode

l 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.

Data Cube: From Tables and Spreadsheets to Data Cubes

"What is a data cube?" A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts. In general terms, dimensions are entities with respect to which an organization wants to keep records. For example, 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.

Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Data Models

1. 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. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.
2. 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 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 space savings 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.
3.

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.

figure 11الرسم صفحة 22 مطلوب يجي كال
roll up/drill down/slice and dice/rotateايعازات ال
ذني مطلوبات كتطبيق على الرسم اذا جابة بل امتحان

Data Warehouse Components
Building data warehouse involves grouping a number of essential components like building a school or a hospital can be seen as building blocks consists of class rooms, offices, etc. but with a specific arrangements that makes the best interest. Same way, data warehouses consist of a number of blocks and the structure of these components called architecture. Building blocks consist of source data, data staging, data storage, and information delivery. 1. Source Data Component

• Production Data: This sort of data comes from different operational systems and based on the information needed by data warehouse requirements, choosing segments of data from these systems. These data reside on multiplatform hardware systems with different formats supported by different operating systems.

• Internal Data: Inside organization users may keep their private spreadsheets, customer profiles, and documents. This data may be a part of the data warehouse.

• Archived Data: Transactional databases are designated to do current businesses; some of the data are archived in files. Many different archiving methods do exist. The first, storing data on archival databases that may be still in use on-line. Whilst, the second storing older data to flat files on disk storage. And finally, archiving data in cartridges kept off-line (Long period data).

• External Data: Executives depend mostly on information from external sources such as statistics relating to their business by external agencies may be used.

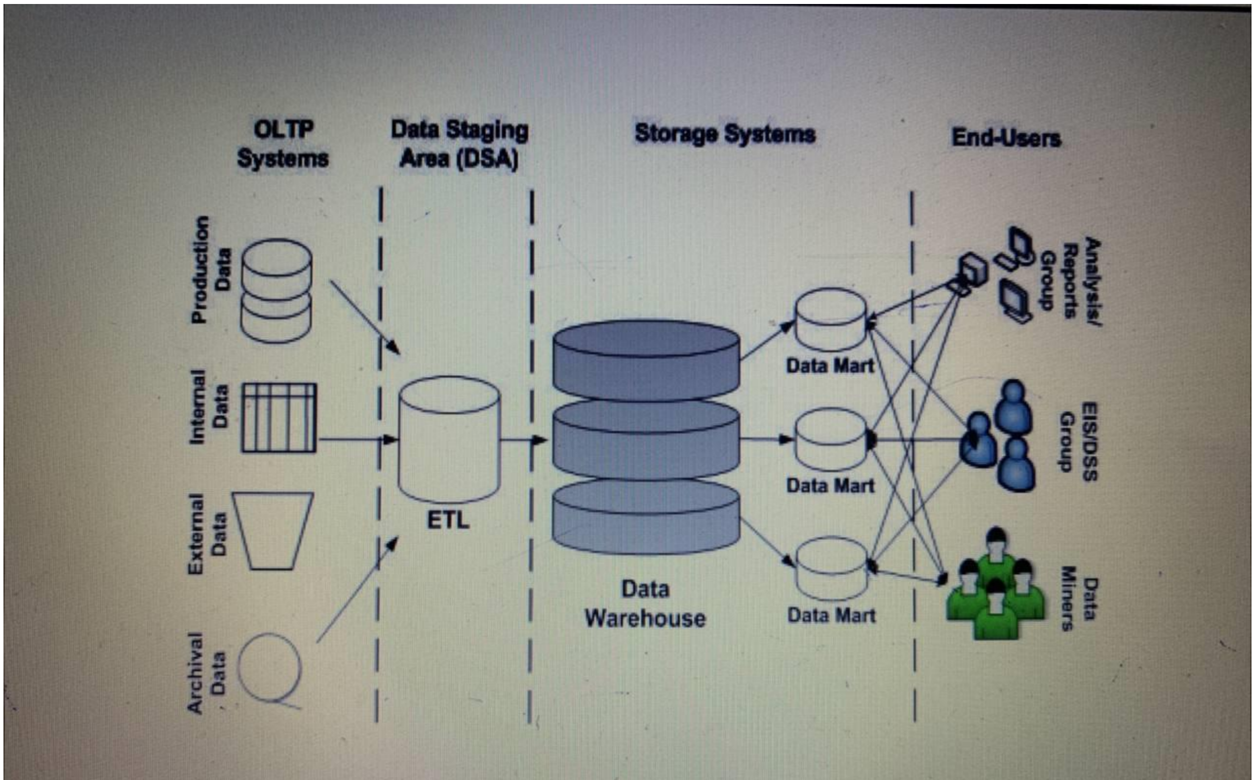


Figure 13. Data Warehouse Components.

- Meta Data For Example:

In order to store data, over the years, many application designers in each branch have made their individual decisions as to how an application and database should be built. So source systems will be different in naming conventions, variable measurements, encoding structures, and physical attributes of data. Consider a bank that has got several branches in several countries, has millions of customers and the lines of business of the enterprise are savings, and loans. The following example explains how the data is integrated from source systems to target systems.

Data Mart A data warehouse is a cohesive data model that defines the central data repository for an organization. A data mart is a data repository for a specific user group. It contains summarized data that the user group can easily understand, process, and apply. A data mart cannot stand alone; it requires a data warehouse. Because each data warehousing effort is unique, your company's data warehousing environment may differ slightly from what we are about to introduce. Each data mart is a collection of tables organized according to the particular requirements of a user or group of users. Retrieving a collection of different kinds of data from a “normalized” warehouse can be complex and time-consuming. Hence the need to rearrange the data so they can be retrieved more easily. The notion of a “mart” suggests that it is organized for theultimate consumers with the potato chips, and video tapes all next to each other. This organization does not have to follow any particular inherent rules or structures. Indeed, it may not even make sense.

And however the marts are organized initially, the requirements are almost certain to change once the user has seen the implications of the request.

This means that the creation of data marts requires:

- Understanding of the business involved.

- Responsiveness to the user’s stated objectives.

- Sufficient facility with database modeling and design to produce new tables quickly.

- Tools to convert models into data marts quickly.

Steps for the Design and Construction of Data Warehouse:

- Top-down view allows the selection of the relevant information necessary for the data warehouse based on current and future needs

- Data source view exposes the information being captured, stored, and managed by operational systems (E/R models, CASE, etc). This information may be documented at various levels of detail.

- Data warehouse view ♣ Includes fact tables and dimension tables, pre-calculated totals and counts. ♣ Source information, date, and time provide historical context.

- Business query view is the data perspective in the data warehouse from the end􀂊user’s viewpoint.

Methodologies used to construct large systems (Data Warehouse):

♣ Waterfall: structured and systematic analysis at each step. ♣

Spiral: rapid generation of increasingly functional systems, short turnaround time, quick turnaround.

Typical data warehouse design process consist the following steps:

♣ Choose a business process to model, e.g., orders, invoices, shipments, sales, etc.

♣ Choose the grain (atomic level of data) of the business process.

♣ Choose the dimensions that will apply to each fact table record.

♣ Choose the measure that will populate each fact table record.