# Data warehousing

A data warehouse is a centralized repository of integrated data from one or more disparate sources. Data warehouses store current and historical data and are used for reporting and analysis of the data.

To move data into a data warehouse, data is periodically extracted from various sources that contain important business information. As the data is moved, it can be formatted, cleaned, validated, summarized, and reorganized. Alternatively, the data can be stored in the lowest level of detail, with aggregated views provided in the warehouse for reporting. In either case, the data warehouse becomes a permanent data store for reporting, analysis, and business intelligence (BI).

## When to use this solution

Choose a data warehouse when you need to turn massive amounts of data from operational systems into a format that is easy to understand. Data warehouses don't need to follow the same terse data structure you may be using in your OLTP databases. You can use column names that make sense to business users and analysts, restructure the schema to simplify relationships, and consolidate several tables into one. These steps help guide users who need to create reports and analyze the data in BI systems, without the help of a database administrator (DBA) or data developer.

Consider using a data warehouse when you need to keep historical data separate from the source transaction systems for performance reasons. Data warehouses make it easy to access historical data from multiple locations, by providing a centralized location using common formats, keys, and data models.

Because data warehouses are optimized for read access, generating reports is faster than using the source transaction system for reporting.

Other benefits include:

* The data warehouse can store historical data from multiple sources, representing a single source of truth.
* You can improve data quality by cleaning up data as it is imported into the data warehouse.
* Reporting tools don't compete with the transactional systems for query processing cycles. A data warehouse allows the transactional system to focus on handling writes, while the data warehouse satisfies the majority of read requests.
* A data warehouse can consolidate data from different software.
* Data mining tools can find hidden patterns in the data using automatic methodologies.
* Data warehouses make it easier to provide secure access to authorized users, while restricting access to others. Business users don't need access to the source data, removing a potential attack vector.
* Data warehouses make it easier to create business intelligence solutions, such as [OLAP cubes](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing).

## Challenges

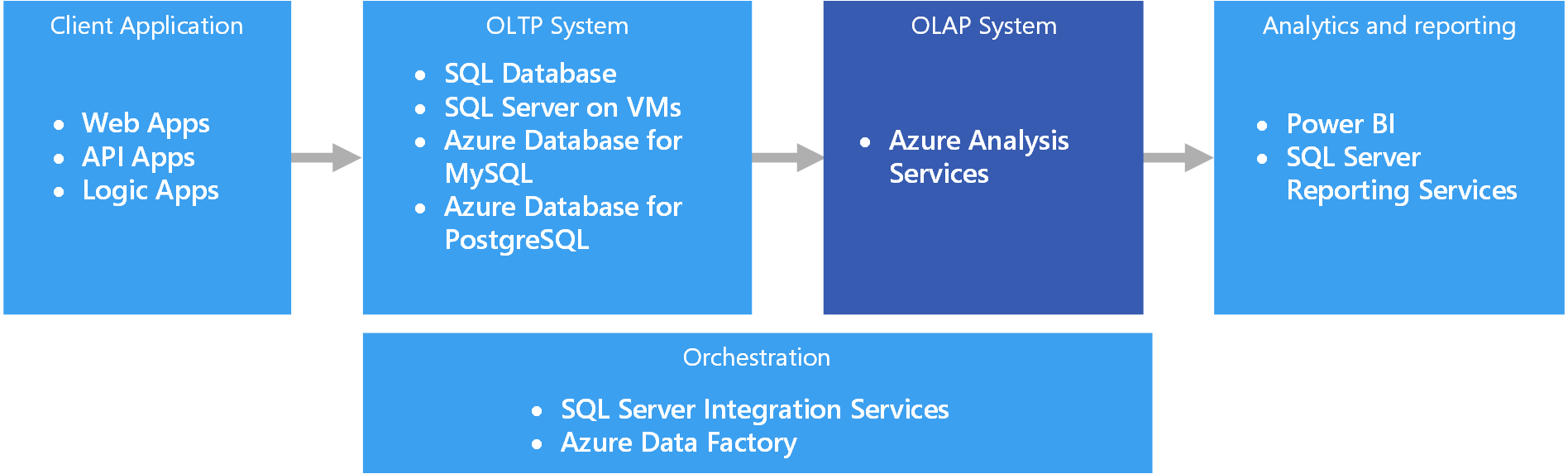
Properly configuring a data warehouse to fit the needs of your business can bring some of the following challenges:

* Committing the time required to properly model your business concepts. Data warehouses are information driven. You must standardize business-related terms and common formats, such as currency and dates. You also need to restructure the schema in a way that makes sense to business users but still ensures accuracy of data aggregates and relationships.
* Planning and setting up your data orchestration. Consider how to copy data from the source transactional system to the data warehouse, and when to move historical data from operational data stores into the warehouse.
* Maintaining or improving data quality by cleaning the data as it is imported into the warehouse.

# Online analytical processing (OLAP)

**Online analytical processing (OLAP)** is a technology that organizes large business databases and supports complex analysis. It can be used to perform complex analytical queries without negatively affecting transactional systems.

The databases that a business uses to store all its transactions and records are called **online transaction processing (OLTP)** databases. These databases usually have records that are entered one at a time. Often they contain a great deal of information that is valuable to the organization. The databases that are used for OLTP, however, were not designed for analysis. Therefore, retrieving answers from these databases is costly in terms of time and effort. OLAP systems were designed to help extract this business intelligence information from the data in a highly performant way. This is because OLAP databases are optimized for heavy read, low write workloads.



# Online transaction processing (OLTP)

The management of transactional data using computer systems is referred to as online transaction processing (OLTP). OLTP systems record business interactions as they occur in the day-to-day operation of the organization, and support querying of this data to make inferences.

## Transactional data

Transactional data is information that tracks the interactions related to an organization's activities. These interactions are typically business transactions, such as payments received from customers, payments made to suppliers, products moving through inventory, orders taken, or services delivered. Transactional events, which represent the transactions themselves, typically contain a time dimension, some numerical values, and references to other data.

Transactions typically need to be atomic and consistent. Atomicity means that an entire transaction always succeeds or fails as one unit of work, and is never left in a half-completed state. If a transaction cannot be completed, the database system must roll back any steps that were already done as part of that transaction. In a traditional RDBMS, this rollback happens automatically if a transaction cannot be completed. Consistency means that transactions always leave the data in a valid state. (These are very informal descriptions of atomicity and consistency. There are more formal definitions of these properties, such as ACID.)

Transactional databases can support strong consistency for transactions using various locking strategies, such as pessimistic locking, to ensure that all data is strongly consistent within the context of the enterprise, for all users and processes.

The most common deployment architecture that uses transactional data is the data store tier in a 3-tier architecture. A 3-tier architecture typically consists of a presentation tier, business logic tier, and data store tier. A related deployment architecture is the N-tier architecture, which may have multiple middle-tiers handling business logic.

## Typical traits of transactional data

Transactional data tends to have the following traits:

| **TYPICAL TRAITS OF TRANSACTIONAL DATA** | |
| --- | --- |
| **Requirement** | **Description** |
| Normalization | Highly normalized |
| Schema | Schema on write, strongly enforced |
| Consistency | Strong consistency, ACID guarantees |
| Integrity | High integrity |
| Uses transactions | Yes |
| Locking strategy | Pessimistic or optimistic |
| Updateable | Yes |
| Appendable | Yes |
| Workload | Heavy writes, moderate reads |
| Indexing | Primary and secondary indexes |
| Datum size | Small to medium sized |
| Model | Relational |
| Data shape | Tabular |
| Query flexibility | Highly flexible |
| Scale | Small (MBs) to Large (a few TBs) |

## When to use this solution

Choose OLTP when you need to efficiently process and store business transactions and immediately make them available to client applications in a consistent way. Use this architecture when any tangible delay in processing would have a negative impact on the day-to-day operations of the business.

OLTP systems are designed to efficiently process and store transactions, as well as query transactional data. The goal of efficiently processing and storing individual transactions by an OLTP system is partly accomplished by data normalization — that is, breaking the data up into smaller chunks that are less redundant. This supports efficiency because it enables the OLTP system to process large numbers of transactions independently, and avoids extra processing needed to maintain data integrity in the presence of redundant data.

## Challenges

Implementing and using an OLTP system can create a few challenges:

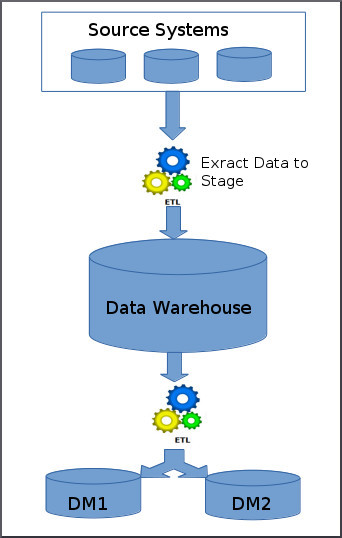
* OLTP systems are not always good for handling aggregates over large amounts of data, although there are exceptions, such as a well-planned SQL Server-based solution. Analytics against the data, that rely on aggregate calculations over millions of individual transactions, are very resource intensive for an OLTP system. They can be slow to execute and can cause a slow-down by blocking other transactions in the database.
* When conducting analytics and reporting on data that is highly normalized, the queries tend to be complex, because most queries need to de-normalize the data by using joins. Also, naming conventions for database objects in OLTP systems tend to be terse and succinct. The increased normalization coupled with terse naming conventions makes OLTP systems difficult for business users to query, without the help of a DBA or data developer.
* Storing the history of transactions indefinitely and storing too much data in any one table can lead to slow query performance, depending on the number of transactions stored. The common solution is to maintain a relevant window of time (such as the current fiscal year) in the OLTP system and offload historical data to other systems, such as a data mart or data warehouse.

**Data Warehouse design approaches** are very important aspect of building data warehouse. Selection of right data warehouse design could save lot of time and project cost.

There are two different Data Warehouse Design Approaches normally followed when designing a Data Warehouse solution and based on the requirements of your project you can choose which one suits your particular scenario. These methodologies are a result of research from Bill Inmon and Ralph Kimball.

#### **Bill Inmon – Top-down Data Warehouse Design Approach**

“Bill Inmon” is sometimes also referred to as the “father of data warehousing”; his design methodology is based on a top-down approach. In the top-down approach, the data warehouse is designed first and then data mart are built on top of data warehouse.



The above image depicts how the top-down approach works.

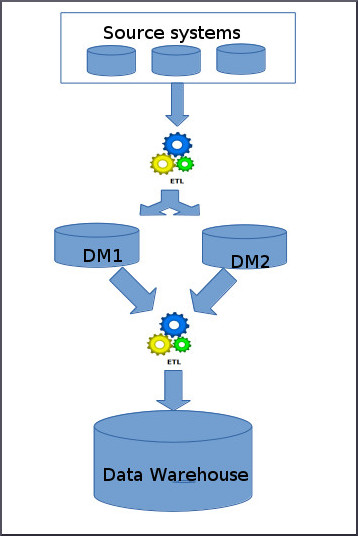
Below are the steps that are involved in top-down approach:

* Data is extracted from the various source systems. The extracts are loaded and validated in the stage area. Validation is required to make sure the extracted data is accurate and correct. You can use the ETL tools or approach to extract and push to the data warehouse.
* Data is extracted from the data warehouse in regular basis in stage area. At this step, you will apply various aggregation, summarization techniques on extracted data and loaded back to the data warehouse.
* Once the aggregation and summarization is completed, various data marts extract that data and apply the some more transformation to make the data structure as defined by the data marts.

#### **Ralph Kimball – Bottom-up Data Warehouse Design Approach**

Ralph Kimball is a renowned author on the subject of data warehousing. His data warehouse design approach is called dimensional modelling or the Kimball methodology. This methodology follows the bottom-up approach.

As per this method, data marts are first created to provide the reporting and analytics capability for specific business process, later with these data marts enterprise data warehouse is created.



The above image depicts how the bottom-up approach works.

Basically, Kimball model reverses the Inmon model i.e. Data marts are directly loaded with the data from the source systems and then ETL process is used to load in to Data Warehouse. The above image depicts how the top-down approach works.

Below are the steps that are involved in bottom-up approach:

* The data flow in the bottom up approach starts from extraction of data from various source system into the stage area where it is processed and loaded into the data marts that are handling specific business process.
* After data marts are refreshed the current data is once again extracted in stage area and transformations are applied to create data into the data mart structure. The data is the extracted from Data Mart to the staging area is aggregated, summarized and so on loaded into EDW and then made available for the end user for analysis and enables critical business decisions.
* **What is data mart?**

A data mart is a subset of an organizational data store, usually oriented to a specific purpose or major data subject that may be distributed to support business needs. Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization.

Data marts are often derived from subsets of data in a data warehouse, though in the bottom-up data warehouse design methodology the data warehouse is created from the union of organizational data marts.

**What is Dimensional Modeling in Data Warehouse?**

**Dimensional Modeling (DM)** is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional modeling is to optimize the database for faster retrieval of data. The concept of Dimensional Modelling was developed by Ralph Kimball and consists of “fact” and “dimension” tables.

A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.

These dimensional and relational models have their unique way of data storage that has specific advantages.

For instance, in the relational mode, normalization and ER models reduce redundancy in data. On the contrary, dimensional model in data warehouse arranges data in such a way that it is easier to retrieve information and generate reports.

## Elements of Dimensional Data Model

### Fact

Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number

### Dimension

Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be

* Who – Customer Names
* Where – Location
* What – Product Name

In other words, a dimension is a window to view information in the facts.

### Attributes

The Attributes are the various characteristics of the dimension in dimensional data modeling.

In the Location dimension, the attributes can be

* State
* Country
* Zipcode etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

### Fact Table

A fact table is a primary table in dimension modelling.

A Fact Table contains

1. Measurements/facts
2. Foreign key to dimension table

### Dimension Table

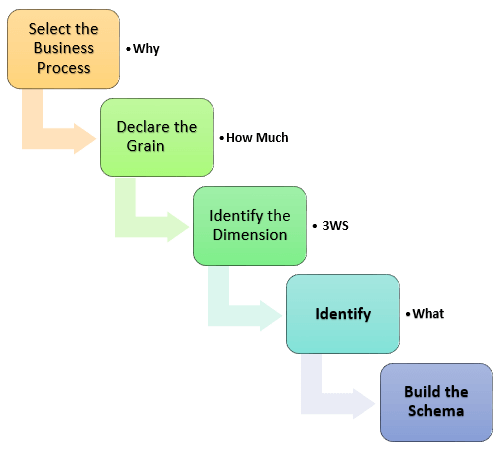
* A dimension table contains dimensions of a fact.
* They are joined to fact table via a foreign key.
* Dimension tables are de-normalized tables.
* The Dimension Attributes are the various columns in a dimension table
* Dimensions offers descriptive characteristics of the facts with the help of their attributes
* No set limit set for given for number of dimensions
* The dimension can also contain one or more hierarchical relationships

**Steps of Dimensional Modelling**

The accuracy in creating your Dimensional modeling determines the success of your data warehouse implementation. Here are the steps to create Dimension Model

1. Identify Business Process
2. Identify Grain (level of detail)
3. Identify Dimensions
4. Identify Facts
5. Build Star

The model should describe the Why, How much, When/Where/Who and What of your business process



### Step 1) Identify the Business Process

Identifying the actual business process a datarehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization. The selection of the Business process also depends on the quality of data available for that process. It is the most important step of the Data Modelling process, and a failure here would have cascading and irreparable defects.

To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

### Step 2) Identify the Grain

The Grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.

During this stage, you answer questions like

1. Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for Datawarehouse
2. Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives
3. How do the above two choices affect the database size?

**Example of Grain:**

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

So, the grain is "product sale information by location by the day."

### Step 3) Identify the Dimensions

Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.

**Example of Dimensions:**

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

Dimensions: Product, Location and Time

Attributes: For Product: Product key (Foreign Key), Name, Type, Specifications

Hierarchies: For Location: Country, State, City, Street Address, Name

### Step 4) Identify the Fact

This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.

**Example of Facts:**

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

The fact here is Sum of Sales by product by location by time.

### Step 5) Build Schema

In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

1. **Star Schema**

The star schema architecture is easy to design. It is called a star schema because diagram resembles a star, with points radiating from a center. The center of the star consists of the fact table, and the points of the star is dimension tables.

The fact tables in a star schema which is third normal form whereas dimensional tables are de-normalized.

1. **Snowflake Schema**

The snowflake schema is an extension of the star schema. In a snowflake schema, each dimension are normalized and connected to more dimension tables.

## What is Multi-Dimensional Data Model in Data Warehouse?

**Multidimensional data model** in data warehouse is a model which represents data in the form of data cubes. It allows to model and view the data in multiple dimensions and it is defined by dimensions and facts. Multidimensional data model is generally categorized around a central theme and represented by a fact table.

**Dimension Surrogate Keys**

 dimension table is designed with one column serving as a unique primary key. This primary key cannot be the operational system’s natural key because there will be multiple dimension rows for that natural key when changes are tracked over time. In addition, natural keys for a dimension may be created by more than one source system, and these natural keys may be incompatible or poorly administered.  The DW/BI system needs to claim control of the primary keys of all dimensions; rather than using explicit natural keys or natural keys with appended dates, you should create anonymous integer primary keys for every dimension. These *dimension surrogate keys*are simple integers, assigned in sequence, starting with the value 1, every time a new key is needed.

**Degenerate Dimensions**

Sometimes a dimension is deﬁned that has no content except for its primary key. For example, when an invoice has multiple line items, the line item fact rows inherit all the descriptive dimension foreign keys of the invoice, and the invoice is left with no unique content. But the invoice number remains a valid dimension key for fact tables at the line item level. This *degenerate dimension*is placed in the fact table with the explicit acknowledgment that there is no associated dimension table.

**Role-Playing Dimensions**

A single physical dimension can be referenced multiple times in a fact table, with each reference linking to a logically distinct role for the dimension. For instance, a fact table can have several dates, each of which is represented by a foreign key to the date dimension.  It is essential that each foreign key refers to a separate view of the date dimension so that the references are independent. These separate dimension views (with unique attribute column names) are called *roles*.

**Junk Dimensions**

Transactional business processes typically produce a number of miscellaneous, low-cardinality ﬂags and indicators. Rather than making separate dimensions for each ﬂag and attribute, you can create a single *junk dimension*combining them together. This dimension, frequently labeled as a *transaction proﬁle dimension*in a schema, does not need to be the Cartesian product of all the attributes’ possible values, but should only contain the combination of values that actually occur in the source data.

**Data reconciliation** (DR) is a term typically used to describe a verification phase during a **data** migration where the target **data** is compared against original source **data** to ensure that the migration architecture has transferred the **data** correctly.

**Data Cleansing** – It is the procedure to rectify the corrupted, inaccurate, incomplete, and irrelevant parts of data in data sets, data records or database. The whole process refers to identifying the incomplete data either by removing it or replacing it.

To make data accuracy 100% correct, most businesses like accounting, retail, manufacturing, healthcare etc. use such techniques.

**Data scrubbing** – As I mentioned above the whole process is same under the data scrubbing process. Hence, it is also called data cleansing. In the data scrubbing process, data that is incomplete, incorrect, improperly structured or formatted or unorganized- requires minor changes or to be removed from the data sets.

Industries like banking, insurance, retail, and transportation use tools related to data scrubbing to maintain a systematical process of data flaws.

**Data aggregation** – In this process any form of data and information gathered is represented in a summarized form. This process is implemented with the aim to achieve some specific business objective or to conduct human analysis.

Businesses that deal with bulk data use such techniques, which are also counted under business intelligence solutions. Data aggregation is a kind of tool that finds relevant search query data and presents that data in summarized format that involves only meaningful information.

## What is active data warehousing?

The transactional data captured and reposited in the Active Data Warehouse. This repository can be utilized in finding trends and patterns that can be used in future decision making.

## What is active data warehousing?

An Active data warehouse aims to capture data continuously and deliver real time data. They provide a single integrated view of a customer across multiple business lines. It is associated with Business Intelligence Systems.

# **Difference between Operational Database and Data Warehouse**

The Operational Database is the source of information for the data warehouse. It includes detailed information used to run the day to day operations of the business. The data frequently changes as updates are made and reflect the current value of the last transactions.

Operational Database Management Systems also called as OLTP (Online Transactions Processing Databases), are used to manage dynamic data in real-time.

Data Warehouse Systems serve users or knowledge workers in the purpose of data analysis and decision-making. Such systems can organize and present information in specific formats to accommodate the diverse needs of various users. These systems are called as Online-Analytical Processing (OLAP) Systems.

Data Warehouse and the OLTP database are both relational databases. However, the goals of both these databases are different.

|  |  |
| --- | --- |
| **Operational Database** | **Data Warehouse** |
| Operational systems are designed to support high-volume transaction processing. | Data warehousing systems are typically designed to  support high-volume analytical processing (i.e., OLAP). |
| Operational systems are usually concerned with current data. | Data warehousing systems are usually concerned with historical data. |
| Data within operational systems are mainly updated regularly according to need. | Non-volatile, new data may be added regularly. Once Added rarely changed. |
| It is designed for real-time business dealing and processes. | It is designed for analysis of business measures by subject area, categories, and attributes. |
| It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table. | It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table. |
| It is optimized for validation of incoming information during transactions, uses validation data tables. | Loaded with consistent, valid information, requires no real-time validation. |
| It supports thousands of concurrent clients. | It supports a few concurrent clients relative to OLTP. |
| Operational systems are widely process-oriented. | Data warehousing systems are widely subject-oriented |
| Operational systems are usually optimized to perform fast inserts and updates of associatively small volumes of data. | Data warehousing systems are usually optimized to perform fast retrievals of relatively high volumes of data. |
| Data In | Data Out |
| Less Number of data accessed. | Large Number of data accessed. |
| Relational databases are created for on-line transactional Processing (OLTP) | Data Warehouse designed for on-line Analytical Processing (OLAP) |

## Difference between OLTP and OLAP

## OLTP System

OLTP System handle with operational data. Operational data are those data contained in the operation of a particular system. Example, ATM transactions and Bank transactions, etc.

## OLAP System

OLAP handle with Historical Data or Archival Data. Historical data are those data that are achieved over a long period. For example, if we collect the last 10 years information about flight reservation, the data can give us much meaningful data such as the trends in the reservation. This may provide useful information like peak time of travel, what kind of people are traveling in various classes (Economy/Business) etc.

The major difference between an OLTP and OLAP system is the amount of data analyzed in a single transaction. Whereas an OLTP manage many concurrent customers and queries touching only an individual record or limited groups of files at a time. An OLAP system must have the capability to operate on millions of files to answer a single query.

|  |  |  |
| --- | --- | --- |
| **Feature** | **OLTP** | **OLAP** |
| Characteristic | It is a system which is used to manage operational Data. | It is a system which is used to manage informational Data. |
| Users | Clerks, clients, and information technology professionals. | Knowledge workers, including managers, executives, and analysts. |
| System orientation | OLTP system is a customer-oriented, transaction, and query processing are done by clerks, clients, and information technology professionals. | OLAP system is market-oriented, knowledge workers including managers, do data analysts executive and analysts. |
| Data contents | OLTP system manages current data that too detailed and are used for decision making. | OLAP system manages a large amount of historical data, provides facilitates for summarization and aggregation, and stores and manages data at different levels of granularity. This information makes the data more comfortable to use in informed decision making. |
| Database Size | 100 MB-GB | 100 GB-TB |
| Database design | OLTP system usually uses an entity-relationship (ER) data model and application-oriented database design. | OLAP system typically uses either a star or snowflake model and subject-oriented database design. |
| View | OLTP system focuses primarily on the current data within an enterprise or department, without referring to historical information or data in different organizations. | OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with data that originates from various organizations, integrating information from many data stores. |
| Volume of data | Not very large | Because of their large volume, OLAP data are stored on multiple storage media. |
| Access patterns | The access patterns of an OLTP system subsist mainly of short, atomic transactions. Such a system requires concurrency control and recovery techniques. | Accesses to OLAP systems are mostly read-only methods because of these data warehouses stores historical data. |
| Access mode | Read/write | Mostly write |
| Insert and Updates | Short and fast inserts and updates proposed by end-users. | Periodic long-running batch jobs refresh the data. |
| Number of records accessed | Tens | Millions |
| Normalization | Fully Normalized | Partially Normalized |
| Processing Speed | Very Fast | It depends on the amount of files contained, batch data refresh, and complex query may take many hours, and query speed can be upgraded by creating indexes. |

**Load and Index**  
  
Load and index meant for adding new data and updating index to the warehouse. The required data are loaded in data warehouse And then it processed such as modifying, updating & deleting the unnecessary or redundant data. It is a process to load the data in the data warehouse and to create the necessary indexes.

Some Characteristics of Data Warehouse Data

Transient data:Data in which changes to existing records are written over previous records, thus destroying the previous data content.

Periodic data:Data that are never physically altered or deleted once they have been added to the store

SCD

<https://dwgeek.com/different-extraction-methods-data-warehouse.html/#:~:text=Extraction%20is%20the%20first%20step,in%20the%20data%20warehouse%20environment>.