Data Warehouse Modeling

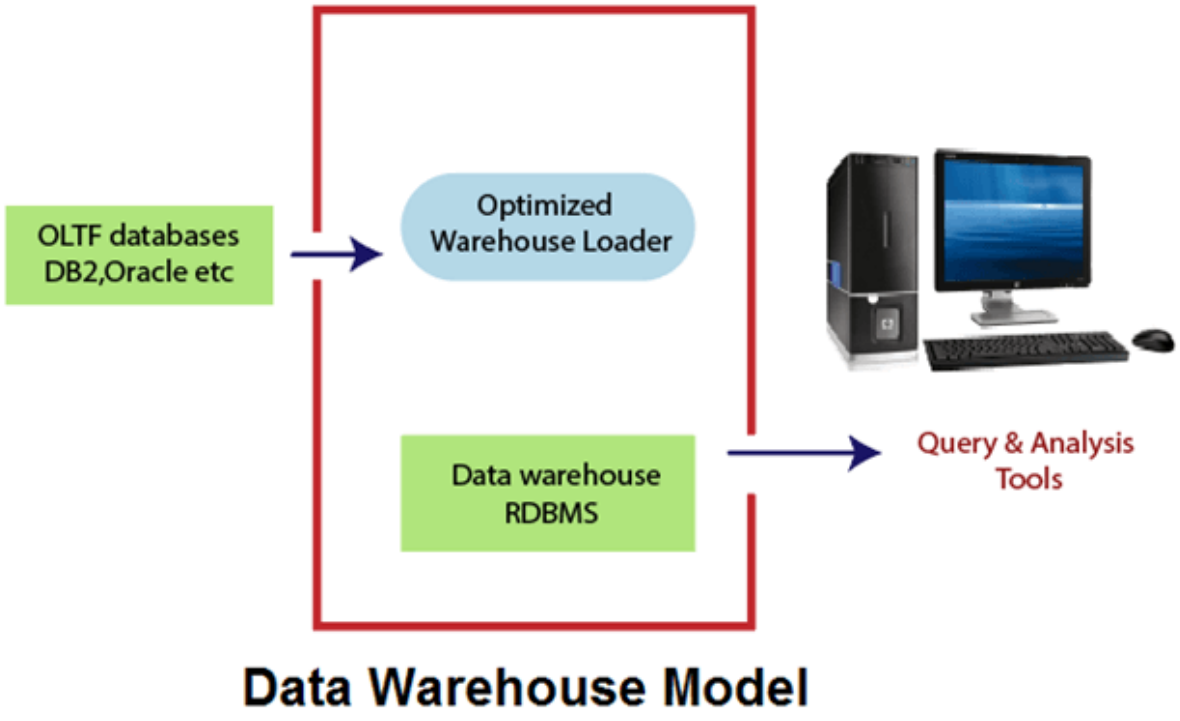
Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse. The goal of data warehouse modeling is to develop a schema describing the reality, or at least a part of the fact, which the data warehouse is needed to support.

Data warehouse modeling is an essential stage of building a data warehouse for two main reasons. Firstly, through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease. Secondly, a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.

Data modeling in data warehouses is different from data modeling in operational database systems. The primary function of data warehouses is to support DSS processes. Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.

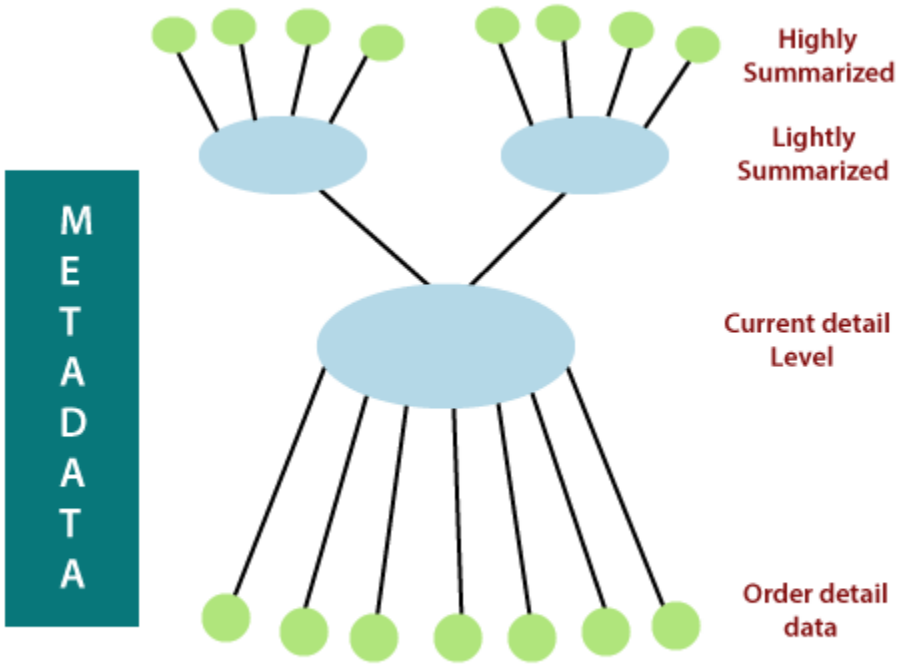
In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data. Moreover, data warehouses are designed for the customer with general information knowledge about the enterprise, whereas operational database systems are more oriented toward use by software specialists for creating distinct applications.

Data Warehouse model is illustrated in the given diagram.



Data Warehouse Model

The data within the specific warehouse itself has a particular architecture with the emphasis on various levels of summarization, as shown in figure:



The Structure of data inside the data warehouse

The current detail record is central in importance as it:

- Reflects the most current happenings, which are commonly the most stimulating.
- It is numerous as it is saved at the lowest method of the Granularity.
- It is always (almost) saved on disk storage, which is fast to access but expensive and difficult to manage.

Older detail data is stored in some form of mass storage, and it is infrequently accessed and kept at a level detail consistent with current detailed data.

Lightly summarized data is data extract from the low level of detail found at the current, detailed level and usually is stored on disk storage. When building the data warehouse have to remember what unit of time is summarization done over and also the components or what attributes the summarized data will contain.

Highly summarized data is compact and directly available and can even be found outside the warehouse.

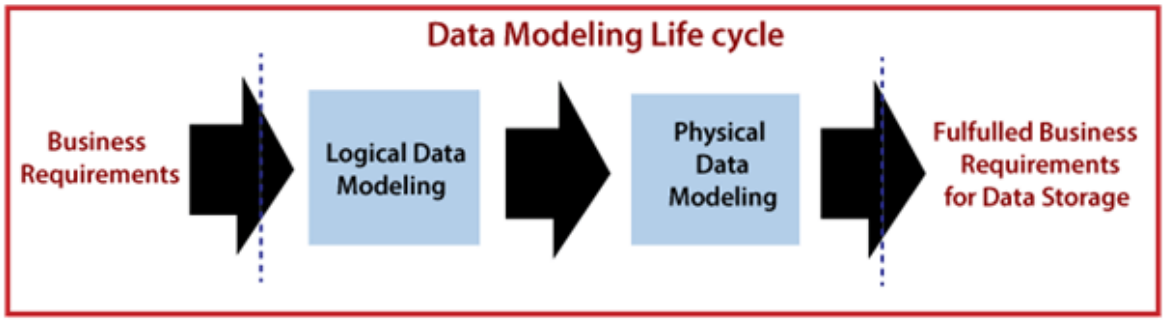
Metadata is the final element of the data warehouses and is really of various dimensions in which it is not the same as file drawn from the operational data, but it is used as:-

- A directory to help the DSS investigator locate the items of the data warehouse.
- A guide to the mapping of record as the data is changed from the operational data to the data warehouse environment.
- A guide to the method used for summarization between the current, accurate data and the lightly summarized information and the highly summarized data, etc.

Data Modeling Life Cycle

In this section, we define a data modeling life cycle. It is a straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems. The result is a logical and physical data model for an enterprise data warehouse.

The objective of the data modeling life cycle is primarily the creation of a storage area for business information. That area comes from the logical and physical data modeling stages, as shown in Figure:



A generic data modeling life cycle

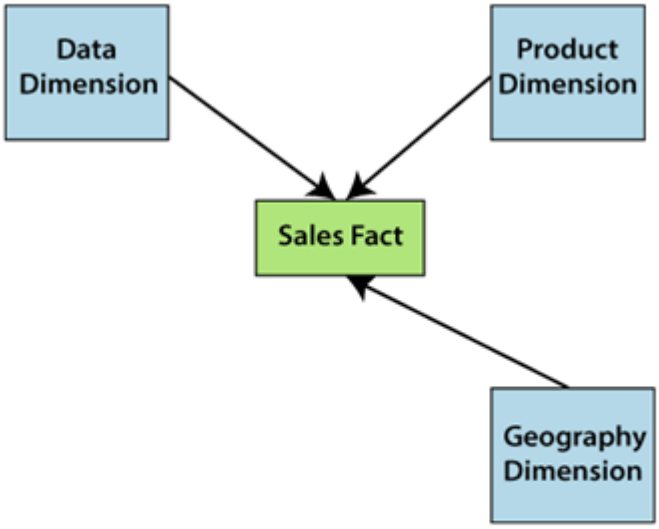
Conceptual Data Model

A conceptual data model recognizes the highest-level relationships between the different entities.

Characteristics of the conceptual data model

- It contains the essential entities and the relationships among them.
- No attribute is specified.
- No primary key is specified.

We can see that the only data shown via the conceptual data model is the entities that define the data and the relationships between those entities. No other data, as shown through the conceptual data model.



Example of Conceptual Data Model

Logical Data Model

A logical data model defines the information in as much structure as possible, without observing how they will be physically achieved in the database. The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model.

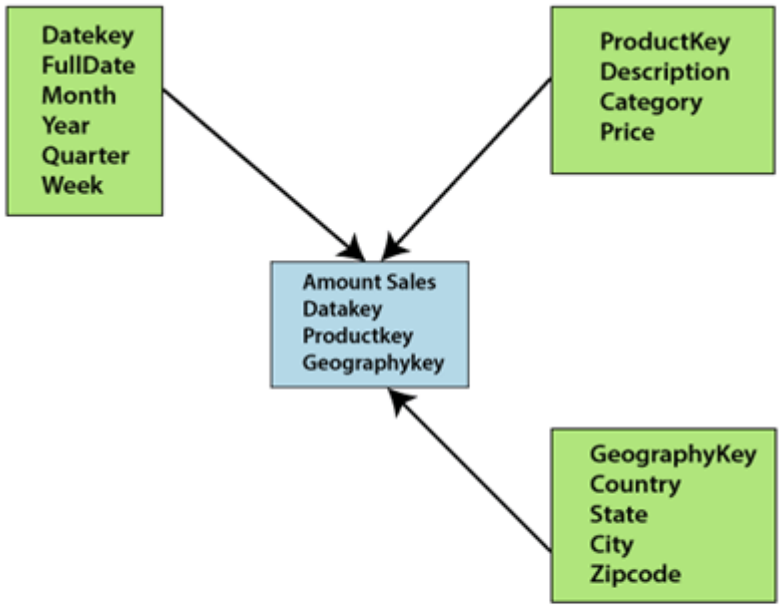
Features of a logical data model

- It involves all entities and relationships among them.

- All attributes for each entity are specified.
- The primary key for each entity is stated.
- Referential Integrity is specified (FK Relation).

The phase for designing the logical data model which are as follows:

- Specify primary keys for all entities.
- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed



Example of Logical Data Model

Physical Data Model

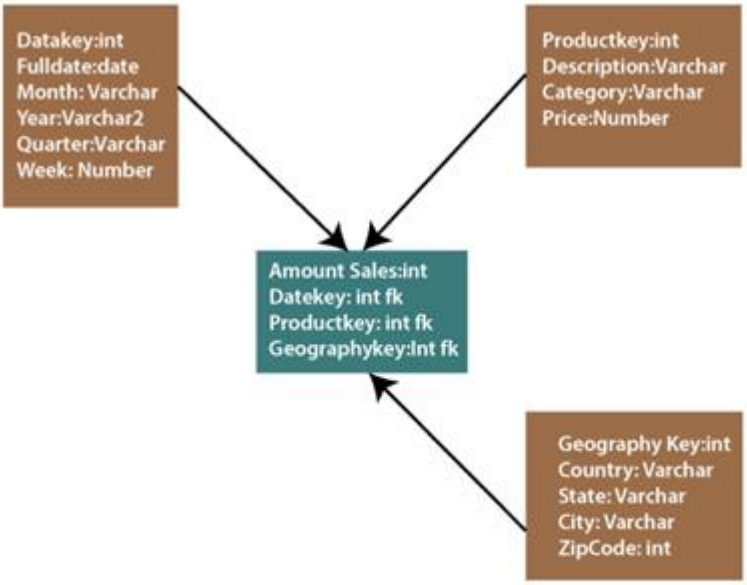
Physical data model describes how the model will be presented in the database. A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables. The purpose of physical data modeling is the mapping of the logical data model to the physical structures of the RDBMS system hosting the data warehouse. This contains defining physical RDBMS structures, such as tables and data types to use when storing the information. It may also include the definition of new data structures for enhancing query performance.

Characteristics of a physical data model

- Specification all tables and columns.
- Foreign keys are used to recognize relationships between tables.

The steps for physical data model design which are as follows:

- Convert entities to tables.
- Convert relationships to foreign keys.
- Convert attributes to columns.

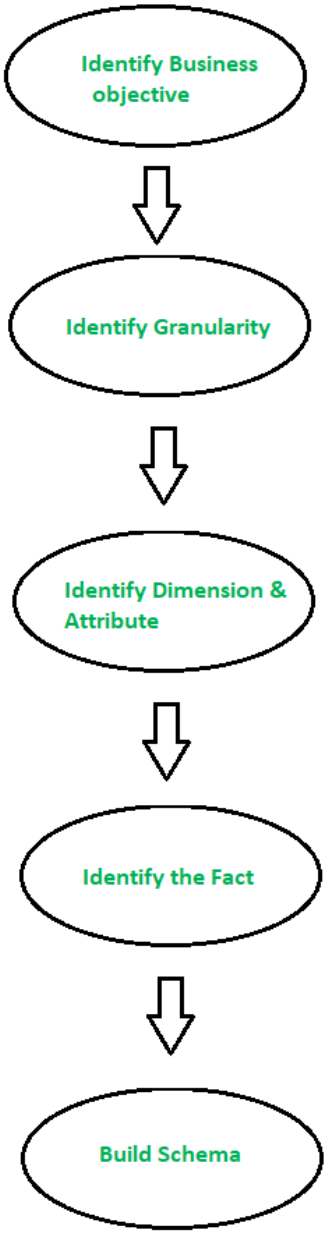


Example of Physical Data Model

CREATING DIMENSIONAL MODELLING

The concept of Dimensional Modelling was developed by Ralph Kimball which is comprised of *facts and dimension* tables. Since the main goal of this modelling is to improve the data retrieval so it is optimized for *SELECT OPERATION*. The advantage of using this model is that we can store data in such a way that it is easier to store and retrieve the data once stored in a data warehouse. Dimensional model is the data model used by many OLAP systems.

Steps to Create Dimensional Data Modelling:



- **Step-1: Identifying the business objective –** The first step is to identify the business objective. Sales, HR, Marketing, etc. are some examples as per the need of the organization. Since it is the most important step of Data Modelling the selection of business objective also depends on the quality of data available for that process.
- **Step-2: Identifying Granularity –** Granularity is the lowest level of information stored in the table. The level of detail for business problem and its solution is described by Grain.
- **Step-3: Identifying Dimensions and its Attributes –** Dimensions are objects or things. Dimensions categorize and describe data warehouse facts and measures in a way that supports meaningful answers to business questions. A data warehouse organizes descriptive attributes as columns in dimension tables. For Example, the data dimension may contain data like a year, month and weekday.
- **Step-4: Identifying the Fact –** The measurable data is held by the fact table. Most of the fact table rows are numerical values like price or cost per unit, etc.
- **Step-5: Building of Schema –** We implement the Dimension Model in this step. A schema is a database structure. There are two popular schemes: Star Schema and Snowflake Schema.

Benefits of Dimensional Modeling

Dimensional modeling is still the most commonly used data modeling technique for designing enterprise data warehouses because of the benefits it yields. These include:

Faster Retrieval of Data

Dimensional data modeling merges the tables in the model itself, which enables users to retrieve data faster from different data sources by running join queries. The denormalized schema of a dimensional model data warehouse, as opposed to normalized one in snowflake schema, is optimized to run ad hoc queries. As a result, it greatly complements the business intelligence (BI) goals of an organization.

Better Understanding of Business Processes

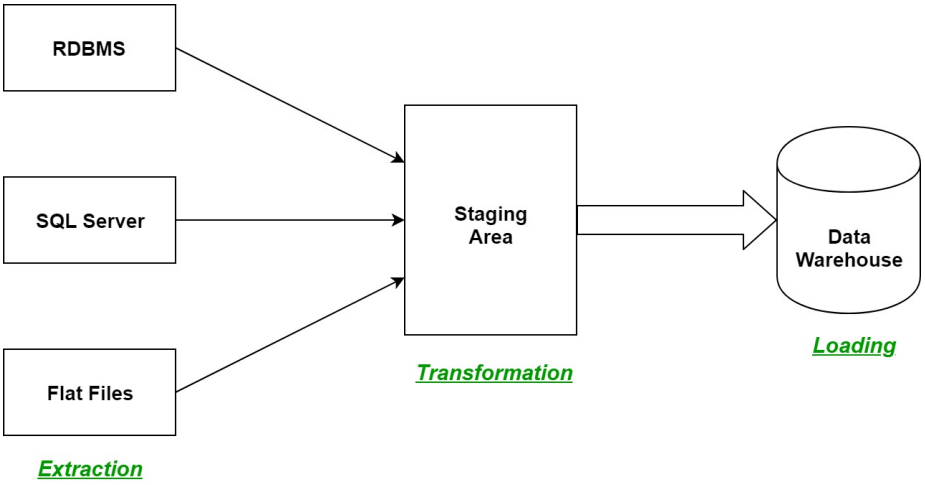
The principles of dimensional modeling are based on fact and dimension tables. We will cover what facts and dimensions are in the subsequent sections. This categorization of data into facts and dimensions, and the entity-relationship structure of a dimensional model, present complex business processes in an easy-to-understand manner to analysts.

Flexible to Change

Dimensional modeling framework makes the data warehousing process extensible. The design can be easily modified to incorporate any new business requirements or make any adjustments to the central repository. New entities can be added in the model or layout of the existing ones can be changed to reflect modified business processes.

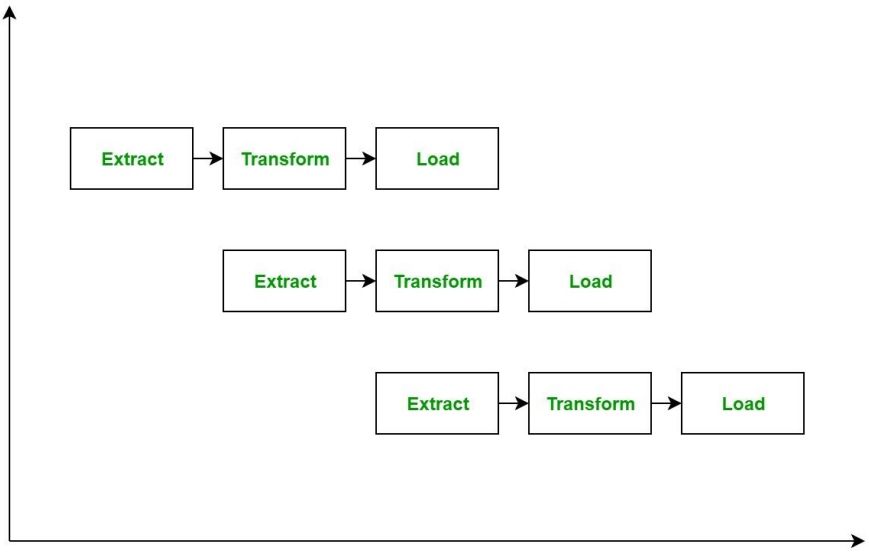
ETL:

ETL is a process in Data Warehousing and it stands for **Extract**, **Transform** and **Load**. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system.



1. **Extraction:**
   The first step of the ETL process is extraction. In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the staging area. It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also. Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.
2. **Transformation:**
   The second step of the ETL process is transformation. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format. It may involve following processes/tasks:
   - Filtering – loading only certain attributes into the data warehouse.
   - Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States, and America into USA, etc.
   - Joining – joining multiple attributes into one.
   - Splitting – splitting a single attribute into multiple attributes.
   - Sorting – sorting tuples on the basis of some attribute (generally key-attribute).
3. **Loading:**
   The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse. Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals. The rate and period of loading solely depends on the requirements and varies from system to system.

ETL process can also use the pipelining concept i.e. as soon as some data is extracted, it can transformed and during that period some new data can be extracted. And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed. The block diagram of the pipelining of ETL process is shown below:



ETL Tools: Most commonly used ETL tools are **Hevo**, Sybase, Oracle Warehouse builder, CloverETL, and MarkLogic.

Data Warehouses: Most commonly used Data Warehouses are **Snowflake**, Redshift, BigQuery, and Firebolt.

Need of ETL

There are many reasons the need for ETL is arising:

• ETL helps the companies to analyze their business data for making critical business decisions.

• Data warehouse provides a shared data repository.

• ETL provides a method of moving data from various sources into a data warehouse.

• As the data sources change, the data warehouse will automatically update.

• ETL helps to migrate the data into a data warehouse.

Extract, transform, load (ETL) refers to the process of moving data from a source system into a data warehouse. Before loading into the warehouse, the data is transformed from a raw state into the format required by the enterprise data warehouse.

The 5 steps of the ETL process are:

extract, clean, transform, load, and analyze.

Extract: Retrieves raw data from an unstructured data pool and migrates it into a temporary, staging data repository

Clean: Cleans data extracted from an unstructured data pool, ensuring the quality of the data prior to transformation.

Transform: Structures and converts the data to match the correct target source Load: Loads the structured data into a data warehouse so it can be properly analyzed and used

Analyze: Big data analysis is processed within the warehouse, enabling the business to gain insight from the correctly configured data.

Each step is performed sequentially. However, the exact nature of each step – which format is required for the target database – depends on the enterprise's specific needs and requirements.

Extraction can involve copying data to tables quickly to minimize the time spent querying the source system. In the transformation step, the data is most usually stored in one set of staging tables as part of the process. Finally, a secondary transformation step might place data in tables that are copies of the warehouse tables, which eases loading.

Each ETL stage requires interaction by data engineers and developers to deal with the capacity limitations of traditional data warehouses.

ETL has been the standard for data warehousing and analytics within enterprises for some time. But as we progress further into 2021, we must view ETL not just as its own microcosm of data readiness processes within an enterprise, but also in the context of an enterprise-wide integration and enhanced business outcomes.

What is Metadata

Metadata is simply defined as data about data.

The data that is used to represent other data is known as metadata.

we can say that metadata is the summarized data that leads us to detailed data. In terms of data warehouse, we can define metadata as follows.

- Metadata is the road-map to a data warehouse.
- Metadata in a data warehouse defines the warehouse objects.
- Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

Metadata is data about the data or documentation about the information which is required by the users. In data warehousing, metadata is one of the essential aspects.

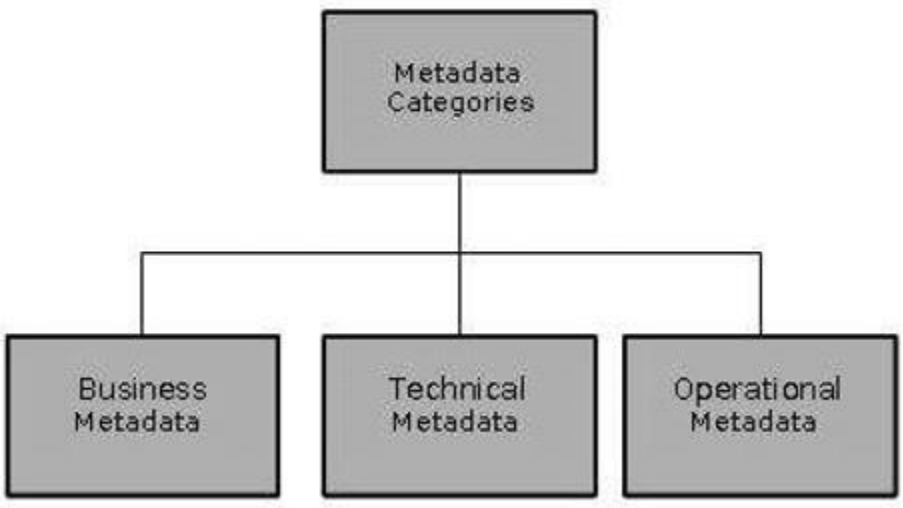
Metadata includes the following:

1. The location and descriptions of warehouse systems and components.
2. Names, definitions, structures, and content of data-warehouse and end-users views.
3. Identification of authoritative data sources.
4. Integration and transformation rules used to populate data.
5. Integration and transformation rules used to deliver information to end-user analytical tools.
6. Subscription information for information delivery to analysis subscribers.
7. Metrics used to analyze warehouses usage and performance.
8. Security authorizations, access control list, etc.

Categories of Metadata

Metadata can be broadly categorized into three categories −

- **Business Metadata** − It has the data ownership information, business definition, and changing policies.
- **Technical Metadata** − It includes database system names, table and column names and sizes, data types and allowed values. Technical metadata also includes structural information such as primary and foreign key attributes and indices.
- **Operational Metadata** − It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.



metadata in a data warehouse fall into three major parts:

- Operational Metadata
- Extraction and Transformation Metadata
- End-User Metadata

Operational Metadata

As we know, data for the data warehouse comes from various operational systems of the enterprise. These source systems include different data structures. The data elements selected for the data warehouse have various fields lengths and data types.

In selecting information from the source systems for the data warehouses, we divide records, combine factor of documents from different source files, and deal with multiple coding schemes and field lengths. When we deliver information to the end-users, we must be able to tie that back to the source data sets. Operational metadata contains all of this information about the operational data sources.

Extraction and Transformation Metadata

Extraction and transformation metadata include data about the removal of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction. Also, this category of metadata contains information about all the data transformation that takes place in the data staging area.

End-User Metadata

The end-user metadata is the navigational map of the data warehouses. It enables the end-users to find data from the data warehouses. The end-user metadata allows the end-users to use their business terminology and look for the information in those ways in which they usually think of the business.

Comparisons of OLAP vs OLTP :

Sr. No.	*Category*	*OLAP (Online analytical processing)*	*OLTP (Online transaction processing)*
1.	**Definition**	It is well-known as an online database query management system.	It is well-known as an online database modifying system.
2.	**Data source**	Consists of historical data from various Databases. In other words, different OLTP databases are used as data sources for OLAP.	Consists of only of operational current data. In other words, the original data source is OLTP and its transactions.
3.	**Method used**	It makes use of a data warehouse.	It makes use of a standard database management system (DBMS).
4.	**Application**	It is subject-oriented. Used for Data Mining, Analytics, Decisions making, etc.	It is application-oriented. Used for business tasks.
5.	**Normalized**	In an OLAP database, tables are not normalized.	In an OLTP database, tables are normalized (3NF).
6.	**Usage of data**	The data is used in planning, problem-solving, and decision-making.	The data is used to perform day-to-day fundamental operations.
7.	**Task**	It reveals a snapshot of present business tasks.	It provides a multi-dimensional view of different business tasks.

8.	**Purpose**	It serves the purpose to extract information for analysis and decision-making.	It serves the purpose to Insert, Update, and Delete information from the database.
9.	**Volume of data**	A large amount of data is stored typically in TB, PB	The size of the data is relatively small as the historical data is archived. For ex MB, GB
10.	**Queries**	Relatively slow as the amount of data involved is large. Queries may take hours.	Very Fast as the queries operate on 5% of the data.
11.	**Update**	The OLAP database is not often updated. As a result, data integrity is unaffected.	The data integrity constraint must be maintained in an OLTP database.
12.	**Backup and Recovery**	It only need backup from time to time as compared to OLTP.	Backup and recovery process is maintained rigorously
13.	**Processing time**	The processing of complex queries can take a lengthy time.	It is comparatively fast in processing because of simple and straightforward queries.
14.	**Types of users**	This data is generally managed by CEO, MD, GM.	This data is managed by clerks, managers.
15.	**Operations**	Only read and rarely write operation.	Both read and write operations.
16.	**Updates**	With lengthy, scheduled batch operations, data is refreshed on a regular basis.	The user initiates data updates, which are brief and quick.
17.	**Nature of audience**	Process that is focused on the customer.	Process that is focused on the market.
18.	**Database Design**	Design with a focus on the subject.	Design that is focused on the application.
19.	**Productivity**	Improves the efficiency of business analysts.	Enhances the user's productivity.

OLAP OPERATIONS:

https://www.javatpoint.com/olap-operations