What is a Data Warehouse?

Think of yourself as a data analyst working for a company that has the following three departments: Marketing, Sales and Finance. Now, let’s assume that each department maintains a separate database.

This could lead to a situation wherein each department has its own version of the facts. For a question such as 'What is the total revenue of the last quarter?', every department might have a different answer. This is because each department draws information from a different database.

This is where a data warehouse can prove to be useful. It can help with creating a single version of the truth and the facts. A data warehouse would thus be the central repository of data of the entire enterprise.

Now, in the upcoming video, you will learn what a data warehouse is and how it is useful for companies in carrying out data analytics. You will also learn about **OLAP**, which stands for **Online Analytical Processing systems**. OLAP is used to extract business-relevant information and perform analysis on the data stored in data warehouses. In the next video, you will understand the data warehouse.

So, a data warehouse is a collection of data. It has the following properties:

- **Subject-oriented:** A data warehouse should contain information about a few well-defined subjects rather than the enterprise.
- **Integrated:** A data warehouse is an integrated repository of data. It contains information from various systems within an organisation.
- **Non-volatile:** The data values in a database cannot be changed without a valid reason.
- **Time-variant:** A data warehouse contains historical data for analysis.

In the next segment, you will learn about the structure of the data warehouse.

Data Warehousing

A data warehouse holds data from multiple sources in a single repository or location. For a typical organisation, what are the benefits of a data warehouse? More than one option may be correct.

□ It helps integrate data from multiple source systems, enabling a central view across the enterprise.

□ Data persistence. A data warehouse stores current as well as historical data, which may not be the case with locally stored files in MS Excel, etc.

□ It enables analysts to create holistic business reports easily using information from multiple departments.

□ It speeds up the analysis and the processing of data as compared with having isolated databases.

Ans:

All of the above answers are correct.

Structure of a Data Warehouse

In the previous segment, you learnt about the basic concepts of data warehousing. Now, one of the primary methods of designing a data warehouse is **dimensional modelling**.

The two key elements of dimensional modelling include **facts** and **dimensions**, which are basically the different types of variables that are used to design a data warehouse. They are arranged in a specific manner, known as a **schema diagram**. So, in the following video, you will learn more about facts and dimensions.

So, essentially, facts are the numerical data in a data warehouse and dimensions are the metadata (that is, data explaining some other data) attached to the fact variables. Both facts and dimensions are equally important for generating actionable insights from a data set.

So, you have learnt about the structure of the data warehouse. In the next segment, you will get the idea about the star schema but before that let's solve some topic related MCQs.

Question 1/2

Facts and Dimensions

Consider a bank that has thousands of ATMs across India. In every transaction, the following variables are recorded:

1. Withdrawal amount
2. Account balance after withdrawal
3. Transaction charge amount
4. Customer ID
5. ATM ID
6. Date of withdrawal

Which of the variables above are fact variables (select all that apply)?

□ Withdrawal amount

□ Account balance after withdrawal

□ Transaction charge amount

□ Customer ID

□ ATM ID

□ Date of withdrawal

Ans:

1st three answers are correct.

Dimensional Modelling

What are the benefits of having dimension variables apart from facts? More than one option may be correct.

☑ It helps with performing various types of analyses, such as sector-wise, country-wise or funding-type-based analyses.

□ It helps in summarising the fact variables by calculating their sum, average, range, etc.

☑ It helps with extracting specific, useful information, such as the total investment made in the Automobile sector in India between 2014 and 2015.

Ans:

1st and 3rd answers are correct.

Star Schema

In the previous segment, you learnt about facts and dimensions, which are the two key elements of dimension modelling. Now, a typical problem might involve multiple databases with many different variables, but we may not be interested in all of them. Hence, only some facts and dimensions are combined in a specific manner to build the structure of a data warehouse, called a schema diagram. A schema is an outline of the entire data warehouse. It shows how different data sets are connected and how the different attributes of each data set are used for the data warehouse

In the upcoming video, you will see an example of how an e-commerce company can design a data warehouse.

So, in the video, you learnt about **star schema**. Although there are other types of schemas available for a data warehouse, the star schema is the most widely used. You can visit the link provided under Additional Reference to learn more about the different types of schemas.

Additional Reference:

Click Data Warehouse Dimensional Modelling (Types of Schemas) to learn more about the different types of schemas.

https://www.folkstalk.com/2010/01/data-warehouse-dimensional-modelling.html

Star Schema

In a star schema, what is the name of the table that contains the metadata that is needed for analysing numerical/quantitative data?

○ Fact table

● Dimension table

○ All the tables of the schema

○ None of the above

Ans:

Dimension table answers is correct.

OLAP vs OLTP

Now that you have developed a fair understanding of databases, you might be wondering how a data warehouse and a database differ from each other.

So, in this segment, you will learn exactly which features of a data warehouse differentiate it from a regular database. In the upcoming video, our expert will talk about the differences between a transactional database and a data warehouse.

So, in the video, you learnt about the differences between a transactional database (i.e., **OLTP, or Online Transactional Processing**) and a data warehouse (which is often referred to as **OLAP, or Online Analytical Processing**). Notice that the major difference between OLAP and OLTP is apparent in the names themselves: OLTP is used for day-to-day transactions, whereas OLAP is used for analytical purposes.

Earlier, you were introduced to the terms **dimensional modelling** and **star schema.** They are essential for creating the structure of a data warehouse. These techniques involve finding out the variables on which analysis can be performed and then combining them with the metadata to derive meaningful insights. You will learn more about them in the next session.

Additional Resources:

1. You can go Data Warehousing Concepts to learn more about a data warehouse and its characteristics.
2. You can go OLTP vs. OLAP to learn more about the differences between OLAP and OLTP systems.

Question 1/3

OLAP vs OLTP

Which of the following statement(s) is/are true? More than one option may be correct.

□ In OLAP systems, data is taken from a single data source.

□ Compared with OLTP, OLAP systems use complex queries to perform tasks, because in OLAP, we perform analysis on various tables and it needs complex queries.

□ OLAP systems use ETL.

□ Schemas are an essential part of creating OLAP systems.

Ans: last 3 answers are Correct.

Question 2/3

OLAP vs OLTP

List the types of databases that are used to perform each of the following operations:

1. Making a purchase
2. Withdrawing cash from an ATM
3. Finding the average sales of a local store
4. Finding profits by region for a specific product of a food MNC

● 1 - Transactional, 2 - Transactional, 3 - Transactional and data warehouse, 4 - Data warehouse

○ 1 - Data warehouse, 2 - Transactional, 3 - Data warehouse, 4 - Data warehouse

○ 1 - Transactional, 2 - Transactional, 3 - Transactional and data warehouse, 4 - Transactional

○ 1 - Data warehouse, 2 - Transactional, 3 - Transactional and data warehouse, 4 - Data warehouse

Ans: 1st answer is Correct.

Question 3/3

OLAP vs OLTP

Choose the correct statement from the following options given below:

○ Transactional databases and data warehouse databases use a similar front-end interface.

○ Dimension modelling is used to create a transactional database.

○ Data is stored differently in a transactional database and a data warehouse database.

○ A star schema is used for creating a transactional database.

Ans:

3rd answer is Correct.

In this way, you have understood about OLTP and OLAP system. In the next segment, you will learn about the SETL (Select, Extract, Transform and Load).

SETL

So far in this module, you have learnt about the different types of databases that are available for solving different problems. Now, what is the next step in the process? As you can see, we now need to get the data into the schema to perform relevant operations on it. You might be wondering how that is done. In the upcoming video, Prof. Chandrashekhar R will give you the answer to this question.

So, in the video, you were introduced to the SETL process, as follows:

SETL: Select, Extract, Transform and Load.

- Select: Identification of the data that you want to analyse
- Extract: Connecting to the particular data source and pulling out the data
- Transform: Modifying the extracted data to standardise it
- Load: Pushing the data into the data warehouse

This process includes the typical operations that are involved in selecting the required data, extracting data from multiple sources, operating on the data so that data from multiple sources is compatible, and loading this data into a data warehouse for analytical purposes.

In this way, you have understood the SETL. In the upcoming segments, you will understand the various constraints.

Entity Constraints

You have already learnt about the SETL process, which is used to ingest data into a schema and perform operations on it. But can we add just any value that we want to a schema or are there any constraints to maintain the sanctity of the database schema?

Put yourself in the place of a data analyst at Uber. A database at Uber has several tables, which record several details, such as details on the rider, the driver, the vehicle used and transaction details. Each such table has several related attributes, which describe it in detail. Consider the 'Rider' table. You go through the values entered in a column that contains the fares for each ride. It would be right for you to expect that the fares can have a maximum of four digits – a fare more than even ?5,000 would raise concerns. This is definitely an outlier and needs to be replaced with the right value. Ensuring that you implement the right constraints for the right attributes in a table can help you avoid such fallacies.

So, as you learnt in the video, constraints are the rules that are used in MySQL to restrict the values that can be stored in the columns of a database. This ensures data integrity, which is nothing but the accuracy and consistency of the data stored in the database. Let's continue our discussion on the relational model in the next video.

In the next video, you will learn about the relational data model.

So, as you learnt in the video, entity constraints are of the following different types:

- **Unique:** This constraint is used for columns that need unique values. For example, 'employee ids' should be unique in an 'employees' table.
- **Null:** This constraint is used to determine the columns that can have null values. For example, an employee may not need to specify their location, which means the 'location' column can have null values in an 'employees' table.
- **Primary Key:** This constraint is used to determine the column that uniquely identifies a table. For example, 'employee ids' uniquely identify every employee. Two employees may have the same name or the same salary, but not the same employee id.

Note that there may be a situation wherein, say, a company wants to store the records of its employees over multiple years. In this case, 'employee id' may not be unique, since an employee will have multiple rows storing details of multiple years. Also, you will need a new column named 'year' in the table.

In this case, a combination of an employee id and a year, i.e., a variable EmpID-Year, can act as a **composite primary key**. To see how this is done in MySQL, you can refer to this Stack Overflow answer.

Question 1/2

Use of Entity Constraints

Entity constraints are used to:

○ Improve the quality of data the entered for a specific property.

○ Control who is allowed access to the data.

○ Ensure that duplicate records are not entered into the table.

○ Prevent users from changing the values stored in the table.

Ans: 3rd answer is Correct.

Question 2/2

Primary Keys

A primary key constraint always enforces both the UNIQUE and NOT NULL constraints.

○ True

○ False

Ans: 1st answer is Correct.

Referential Constraints

In the previous segment, you learnt about the first type of constraints, entity constraints, which pertain to the values in a single table. The second type of constraints is called referential constraints. These are used to restrict the values that are taken by a column in one table based on the values that exist in another table.

Consider the tables given below.

Customers

CustomerID	FirstName	LastName	Age
1	John	Wick	52
2	Sheldon	Cooper	41
3	Charlie	Harper	45

Orders

OrderID	OrderNumber	CustomerID
1	12345	2
2	31343	2
3	12466	3
4	41234	1

CustomerID is a foreign key in the 'Orders' table because it is used to reference the 'Customers' table. You can easily determine which customer placed a particular order and find out all the details of that customer by looking up the corresponding customer id in the 'Customers' table. Watch the upcoming videos to understand how foreign keys make it easier to design and query databases.

So, as you learnt in the video, a referential constraint is a rule between two tables. According to this rule, the value that appears as a foreign key in a table is valid only if it also appears as a primary key in the table to which it refers.

To sum up, a given table has only one primary key but it can have multiple foreign keys. Before you assign a column as a foreign key, you need to ensure that the primary key column of the table that it refers to is present and it does not have null or duplicate values.

Question 1/2

An Example of a Referential Constraint

Which of the following scenarios would require using a referential constraint?

○ All phone numbers should contain the area code.

○ Certain fields are mandatory to fill (such as phone number) before the record is accepted in a database.

○ Information on a customer must be known before anything can be sold to that customer, so that you can refer to other fields to get the information of the customers.

○ When entering the card number for payment, the user must input a 16-digit number.

Ans: 3rd answer is correct

Feedback:

This would constitute a foreign key constraint, which is a referential constraint. This is because customer details are required before a particular product can be associated with the customer who placed the order for that product.

Question 2/2

Primary Key vs Foreign Key

Mention two differences between a primary key and a foreign key.

Ans:

1. For a given table there is only one primary key , but there can be a number of foreign key in a suingle table

2. Primary key is unique but foreign key is not unique

Suggested Answer

1. A primary key identifies each row in a table uniquely, whereas a foreign key uniquely identifies a row in the same table or in another table.

2. Only one primary key is allowed in a table, whereas it can have multiple foreign keys.

Semantic Constraints

Suppose you work at Tata Motors and are asked to analyse the data on the prices of different car models. The values can range from ?2 lakhs to around ?12 lakhs if you are looking to sell the cars to middle-class households. Now, imagine you come across a car model that costs ?1 crore. You would need to get rid of this value as it would negatively affect your analysis and generate misleading insights. How can you ensure such values (also known as outliers) are not present in your data? Let's find out in the upcoming video.

Note: At [02:33], the professor misspeaks that NOT NULL is a semantic constraint. This is, however, not true. You have already learnt that it is an entity constraint.