What is Database Normalization? Why we need normalization?

1. **Normalization** is the process of structuring and handling the relationship between data to minimize redundancy in the relational table and avoid the unnecessary anomalies properties from the database like insertion, update and delete.
2. It helps to divide large database tables into smaller tables and make a relationship between them. It can remove the redundant data and ease to add, manipulate or delete table fields.
3. The main reason for normalizing the relations is removing these anomalies.
4. Failure to eliminate anomalies leads to data redundancy and can cause data integrity and other problems as the database grows.
5. Normalization consists of a series of guidelines that helps to guide you in creating a good database structure

Differentiate between DBMS and RDBMS

DBMS	RDBMS
DBMS stores data as file.	RDBMS stores data in tabular form.
Data elements need to access individually.	Multiple data elements can be accessed at the same time.
No relationship between data.	Data is stored in the form of tables which are related to each other.
Normalization is not present.	Normalization is present.
DBMS does not support distributed database.	RDBMS supports distributed database.

What is Data Model ? Explain Its Types .

Data Model gives us an idea that how the final system will look like after its complete implementation. It defines the data elements and the relationships between the data elements. Data Models are used to show how data is stored, connected, accessed and updated in the database management system

1 Entity-Relationship Model

Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand. It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model.

2 Relational Model

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called *relations* in the relational model.

What is Entity ? Differentiate between strong and weak entity ?

In a database management system (DBMS), an entity is a piece of data that is stored in the database. An entity can be a person, place, thing, or even an event. There are two types of entities in DBMS: strong and weak.

Strong Entity	**Weak Entity**
A strong entity in DBMS is an independent table that doesn't rely on any other tables for its existence	A weak entity type is a dependent table that relies on another table for its existence; it has no meaningful attributes of its own except for the foreign key from the parent table.
A primary key uniquely identifies each row in the table, and foreign keys are used to relate this table to other tables. Null values are not allowed in the primary key column.	For a weak entity type to exist, it must have some relationship with another (parent) table; otherwise, it wouldn't appear in the database!
A simple example of a strong entity type would be "customer" in a customer relational database table. The customerID attribute would be the primary key (and it can't have duplicate values or be NULL), and other information about the customer would be stored in separate attributes such as "firstname," "last name," etc.	A typical example of a weak entity type is an "order." It has no meaning by itself–it must be placed by some customer so it has a foreign key relation to the "customer" table–but it has several attributes of its own such as orderID, productID, etc.

Difference between SQL and PL-SQL.

1	Definition	SQL, is Structural Query Language for database.	PL/SQL is a programming language using SQL for a database.
2	Variables	SQL has no variables.	PL/SQL has variables, data types etc.
3	Control Structures	SQL has no FOR loop, if control and similar structures.	PL/SQL has FOR loop, while loop, if controls and other similar structures.
4	Operations	SQL can execute a single operation at a time.	PL/SQL can perform multiple operation at a time.
5	Language Type	SQL is a declarative language.	PL/SQL is a procedural language.
6	Embedded	SQL can be embedded in a PL/SQL block.	PL/SQL can also be embedded in SQL code.
6	Interaction	SQL directly interacts with database server.	PL/SQL does not directly interacts with database server.
7	Orientation	SQL is data oriented language.	PL/SQL is application oriented language.
8	Objective	SQL is used to write queries, create and execute DDL and DML statments.	PL/SQL is used to write program blocks, functions, procedures, triggers and packages.

What is DML ? Discuss different types of DML statements with example.

DML is an abbreviation of **Data Manipulation Language**.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

Select : SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

Syntax:
SELECT column_Name_1, column_Name_2, ….., column_Name_N **FROM** Name_of_table;

Insert : INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.

Syntax:

INSERT INTO TABLE_NAME (column_Name1 , column_Name2 , column_Name3 , ... col umn_NameN) **VALUES** (value_1, value_2, value_3, ... value_N) ;

Update : UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.

Syntax:

UPDATE Table_name **SET** [column_name1= value_1, ….., column_nameN = value_N] **WH ERE** CONDITION;

Delete: DELETE is a DML command which allows SQL users to remove single or multiple existing records from the database tables.This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

Syntax: **DELETE FROM** Table_Name **WHERE** condition;