

What is database?

A database is a logically coherent collection of data with some inherent meaning, representing some aspect of real world and which is designed, built and populated with data for a specific purpose.

What is DBMS?

It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

What is a Database system?

The database and DBMS software together is called as Database system.

What are the advantages of DBMS?

1. Redundancy is controlled.
2. Unauthorised access is restricted.
3. Providing multiple user interfaces.
4. Enforcing integrity constraints.
5. Providing backup and recovery.

Disadvantage of DBMS?

- Too huge
- Complexity
- Costly
- Database failure

What is the disadvantage in File Processing System?

1. Data redundancy and inconsistency.
2. Difficult in accessing data.
3. Data isolation.
4. Data integrity.
5. Concurrent access is not possible.
6. Security Problems.

What is RDBMS?

RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

What is the difference between DBMS and RDBMS?

- **DBMS (Database Management System):** A system that allows users to create, store, modify, and delete data. It does not require a relational structure for data organization. Examples: Microsoft Access, XML databases.
- **RDBMS (Relational Database Management System):** A subset of DBMS that stores data in a structured format, using tables (relations), and supports relational operations like joins. It enforces data integrity through keys and supports SQL for querying. Examples: MySQL, Oracle, SQL Server.

Describe the three levels of data abstraction?

There are three levels of abstraction:

1. Physical level: The lowest level of abstraction describes how data are stored.
2. Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
3. View level: The highest level of abstraction describes only part of entire database.

What are the different types of DBMS?

The four types of DBMS are:

- **Hierarchical DBMS:** Data is organized in a tree-like structure with parent-child relationships. Example: IBM's IMS.
- **Network DBMS:** Data is represented as a graph with many-to-many relationships. Example: Integrated Data Store (IDS).
- **Relational DBMS (RDBMS):** Data is organized in tables (relations) and managed through SQL. Example: MySQL, PostgreSQL.
- **Object-Oriented DBMS:** Data is stored as objects, like in object-oriented programming. Example: ObjectDB.

What are the primary components of a DBMS?

The primary components of a DBMS are:

- **Database Engine:** Responsible for storing, retrieving, and managing data.
- **Database Schema:** The structure that defines the organization of the database.
- **Query Processor:** Interprets and executes SQL queries.
- **Transaction Manager:** Ensures the ACID properties of transactions.
- **Storage Manager:** Manages the physical storage of data.

Explain the concept of a schema in DBMS.

A schema in DBMS is the structure that defines the organization of data in a database. It includes tables, views, relationships, and other elements. A schema defines the tables and their columns, along with the constraints, keys, and relationships.

What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

What is an Entity?

It is a 'thing' in the real world with an independent existence.

What is an Entity set?

It is a collection of all entities of particular entity type in the database.

What is Weak Entity set?

An entity set may not have sufficient attributes to form a primary key, and its primary key compromises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

What is an attribute?

It is a particular property, which describes the entity.

What is a Relation Schema and a Relation?

A relation Schema denoted by $R(A_1, A_2, \dots, A_n)$ is made up of the relation's name R and the list of attributes A_i that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples $(t_1, t_2, t_3, \dots, t_n)$. Each tuple is an ordered list of n -values $t = (v_1, v_2, \dots, v_n)$.

What is degree of a Relation?

It is the number of attributes of its relation schema.

What is Relationship?

It is an association among two or more entities.

What is Relationship set?

The collection (or set) of similar relationships.

What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set of entity types.

What are the different types of relationships in the DBMS?

A Relationship in DBMS depicts an association between the tables.

Different types of relationships are:

- **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables.
- **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one, or none records in the related table.
- **Many-to-Many:** This states that both the tables can be related to many other tables.

What is degree of Relationship type?

It is the number of entity type participating.

What are super, primary, candidate, and foreign keys?

A **super key** is a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent. No two rows can have the same value of super key attributes.

A **Candidate key** is a minimal super key, i.e., no proper subset of Candidate key attributes can be a super key.

A **Primary Key** is one of the candidate keys. One of the candidate keys is selected as most important and becomes the primary key. There cannot be more than one primary key in a table.

A **Foreign key** is a field (or collection of fields) in one table that uniquely identifies a row of another table.

What is the difference between primary key and unique constraints?

The primary key cannot have NULL value, the unique constraints can have NULL values.

There is only one primary key in a table, but there can be multiple unique key constraints.

What is a composite primary key?

A composite primary key is a primary key made up of two or more columns. Together, these columns must form a unique combination for each row in the table. It's used when a single column isn't sufficient to uniquely identify a record.

What is a query?

A query with respect to DBMS relates to user commands that are used to interact with a data base.

What is SQL?

- SQL is a standard language for storing, manipulating and retrieving data in databases.
- SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures.

Different types of SQL tools?

MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems.

How do you communicate with an RDBMS?

we communicate with an RDBMS using Structured Query Language (SQL).

What types of SQL commands (or SQL subsets) do you know? Give some examples of common SQL commands of each type.

- **Data Definition Language (DDL)** – to define and modify the structure of a database.
DDL: CREATE, ALTER TABLE, DROP, TRUNCATE, and ADD COLUMN
- **Data Manipulation Language (DML)** – to access, manipulate, and modify data in a database.
DML: UPDATE, DELETE, and INSERT
- **Data Control Language (DCL)** – to control user access to the data in the database and give or revoke privileges to a specific user or a group of users.
DCL: GRANT and REVOKE
- **Transaction Control Language (TCL)** – to control transactions in a database.
TCL: COMMIT, SET TRANSACTION, ROLLBACK, and SAVEPOINT
- **Data Query Language (DQL)** – to perform queries on the data in a database to retrieve the necessary information from it.
DQL: – SELECT

How do constraints improve database integrity?

Constraints enforce rules that the data must follow, preventing invalid or inconsistent data from being entered:

- **NOT NULL:** Ensures that a column cannot contain NULL values.
- **UNIQUE:** Ensures that all values in a column are distinct.
- **PRIMARY KEY:** Combines NOT NULL and UNIQUE, guaranteeing that each row is uniquely identifiable.
- **FOREIGN KEY:** Ensures referential integrity by requiring values in one table to match primary key values in another.
- **CHECK:** Validates that values meet specific criteria (e.g., CHECK (Salary > 0)).
By automatically enforcing these rules, constraints maintain data reliability and consistency.

What is referential integrity in DBMS?

Referential Integrity ensures that relationships between tables are maintained correctly. It requires that the foreign key in one table must match a primary key or a unique key in another table (or be NULL). This ensures that data consistency is maintained, and there are no orphan records in the database.

Example: In the Orders table, if the CustomerID is a foreign key, it should match a valid CustomerID in the Customers table or be NULL.

What SQL constraints do you know?

- **DEFAULT** – provides a default value for a column.
- **UNIQUE** – allows only unique values.
- **NOT NULL** – allows only non-null values.
- **PRIMARY KEY** – allows only unique and strictly non-null values (NOT NULL and UNIQUE).
- **FOREIGN KEY** – provides shared keys between two or more tables.

What is a SQL operator? What types of SQL operators do you know?

A reserved character, a combination of characters, or a keyword used in SQL queries to perform a specific operation. SQL operators are commonly used with the WHERE clause to set a condition (or conditions) for filtering the data.

- Arithmetic (+, -, *, /, etc.)
- Comparison (>, <, =, >=, etc.)
- Compound (+, -, *, /, etc.)
- Logical (AND, OR, NOT, BETWEEN, etc.)
- String (% , _ , + , ^ , etc.)
- Set (UNION, UNION ALL, INTERSECT, and MINUS (or EXCEPT))

What is a clause in SQL?

A condition imposed on a SQL query to filter the data to obtain the desired result. Some examples are WHERE, LIMIT, HAVING, LIKE, AND, OR, ORDER BY, etc.

What are some common statements used with the SELECT query?

The most common ones are FROM, GROUP BY, JOIN, WHERE, ORDER BY, LIMIT, and HAVING.

What is the order of appearance of the common statements in the SELECT query?

SELECT → FROM → JOIN → ON → WHERE → GROUP BY → HAVING → ORDER BY

In which order does the interpreter execute the common statements in the SELECT query?

The SQL order of execution:

FROM → JOIN → ON → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

What is the DISTINCT statement? Or how to prevent duplicate records when making a query?

This statement is used with the SELECT statement to filter out duplicates and return only unique values from a column of a table.

What Aggregate functions? What aggregate functions do you know?

Aggregate functions – work on multiple, usually grouped records for the provided columns of a table, and return a single value (usually by group).

- AVG() – returns the average value
- SUM() – returns the sum of values
- MIN() – returns the minimum value
- MAX() – returns the maximum value
- COUNT() – returns the number of rows, including those with null values
- FIRST() – returns the first value from a column
- LAST() – returns the last value from a column

What set operators do you know?

- UNION – returns the records obtained by at least one of two queries (excluding duplicates)
- UNION ALL – returns the records obtained by at least one of two queries (including duplicates)
- INTERSECT – returns the records obtained by both queries
- EXCEPT (called MINUS in MySQL and Oracle) – returns only the records obtained by the first query but not the second one

What is the difference between UNION and UNION ALL in SQL?

- **UNION:** Combines the result of two queries and removes duplicate rows.
- **UNION ALL:** Combines the result of two queries but does not remove duplicates, thus it is faster than UNION.

What is the difference between the DELETE and TRUNCATE statements?

DELETE is a reversible DML (Data Manipulation Language) command used to delete one or more rows from a table based on the conditions specified in the WHERE clause. Instead, TRUNCATE is an irreversible DDL (Data Definition Language) command used to delete all rows from a table. DELETE works slower than TRUNCATE. Also, we can't use the TRUNCATE statement for a table containing a foreign key.

What is a cascade delete in DBMS?

A cascade delete is a type of referential action that automatically deletes records in related tables when a record in the primary table is deleted. This operation ensures that no orphan records (i.e., records in child tables that reference a non-existing parent record) remain in the database.

What is the difference between the DROP and TRUNCATE statements?

DROP deletes a table from the database completely, including the table structure and all the associated constraints, relationships with other tables, and access privileges. TRUNCATE deletes all rows from a table

without affecting the table structure and constraints. DROP works slower than TRUNCATE. Both are irreversible DDL (Data Definition Language) commands.

What is the difference between the HAVING and WHERE statements?

The HAVING works on aggregated data after they are grouped, while the WHERE checks each row individually. If both statements are present in a query, they appear in the following order: WHERE – GROUP BY – HAVING. The SQL engine interprets them also in the same order.

What is the purpose of the SQL ORDER BY clause?

The ORDER BY clause sorts the result set of a query in either **ascending** (default) or **descending order**, based on one or more columns. This helps present the data in a more meaningful or readable sequence.

What is the importance of the COMMIT and ROLLBACK operations?

- **COMMIT:** Saves all changes made during the current transaction to the database permanently.
- **ROLLBACK:** Reverses all changes made during the current transaction, restoring the database to its previous state.

Both operations ensure the ACID properties of transactions: Atomicity and Durability.

What is a subquery in SQL? Provide an example.

A subquery in SQL is a query embedded within another query. It is used to retrieve data that will be used in the outer query. Subqueries can be used in SELECT, INSERT, UPDATE, or DELETE statements.

There are two types of subqueries:

- **Single-row subquery:** Returns a single value.
- **Multiple-row subquery:** Returns multiple values.

Example of a subquery: To find the names of students who have a higher age than the average age:

```
SELECT Name FROM Student
WHERE Age > (SELECT AVG(Age) FROM Student);
```

What is a correlated subquery?

A correlated subquery is a subquery that references columns from the outer query. It is re-executed for each row processed by the outer query. This makes it more dynamic, but potentially less efficient.

Example:

```
SELECT Name,
       (SELECT COUNT (*)
        FROM Orders
        WHERE Orders.CustomerID = Customers.CustomerID) AS OrderCount
FROM Customers;
```

What is Join?

An SQL Join is used to combine data from two or more tables, based on a common field between them.

What are the different types of joins in SQL?

In SQL, the following join types are used to retrieve data from multiple tables:

- **CROSS JOIN:** Merges each row from the first table with every row from the second table, creating a Cartesian product.
- **INNER JOIN:** Fetches rows that have corresponding matches in both tables. Categories into Equi-join and conditional/theta join.
- **NATURAL JOIN:** creates a join on the base of the common columns in the tables. To perform natural join there must be one common attribute (Column) between two tables.
- **LEFT JOIN (LEFT OUTER JOIN):** Fetches all rows from the left table and the matched rows from the right table. When no match is found, the columns from the right table are filled with NULL values.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Fetches all rows from the right table and the matched rows from the left table. NULL values are returned for the left table columns if there's no match.
- **FULL JOIN (FULL OUTER JOIN):** Retrieves all rows when there is a match in either of the tables, with NULL values where there is no match.
- **SELF JOIN:** Joins a table to itself, which is useful for comparing rows within the same table.

What is the difference between having and where clause?

HAVING is used to specify a condition for a group or an aggregate function used in a select statement. The WHERE clause selects before grouping. The HAVING clause selects rows after grouping. Unlike the HAVING clause, the WHERE clause cannot contain aggregate functions.

What is relational algebra?

It is a procedural language, which takes relation as input and generate relation as output.

Name the different operations in relational algebra

Ans: Unary relational operations:

- **Project operation(π)** : This operator is used for selecting columns of a table to them displayed in the result
- **Select operation(σ)** : This operator is used to select some rows from the table which satisfy particular selection condition given in selection operation.
- **Rename operation(ρ)** : This operator is used to rename the column that have same name.

Set theory operations:

- **Union operation(\cup)** : This operator effectively appends the result of first query to the second query
- **Difference operation($-$)** : This operation finds the records that are present in one relation but not in other and display them in the result.
- **Intersection operation(\cap)** : This operator is used to display common values from both the tables in the result

Binary operations:

- **Join operations(\Join)** : This operation is used to retrieve data from multiple tables.
- **Cartesian operations(\times)** : This operation is used to perform relational product or cartesian product of two tables specified in the query
- **Division operations(\div)** : This operator is used to divide values of one table with the values of another table and display their result in the resultant table.

Are the resulting relations of PRODUCT and JOIN operation the same? No

PRODUCT: Concatenation of every row in one relation with every row in another.

JOIN: Concatenation of rows from one relation and related rows from another.

What is decomposition and its type?

The process of breaking up or dividing a single relation into two or more sub-relations is called as decomposition of a relation. Two types are Lossless decomposition and Lossy decomposition.

What is Lossless join property?

It guarantees that the **spurious tuple** generation does not occur with respect to relation schemas after decomposition.

What is dependency preserving decomposition?

In this technique, the original relation is decomposed into smaller relations in such a way that the resulting relations preserve the functional dependencies of the original relation.

What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- (1) Minimizing redundancy,
- (2) Minimizing insertion, deletion and update anomalies.

What is Functional Dependency?

A Functional dependency is denoted by $X \rightarrow Y$ between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if $t1[X] = t2[X]$ then they have $t1[Y] = t2[Y]$. This means the value of X component of a tuple uniquely determines the value of component Y.

What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency $X \rightarrow Y$ is fully functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD $X \rightarrow A$ either of the following is true

1. X is a Super-key of R.
2. A is a prime attribute of R.

In other words, if every non-prime attribute is non-transitively dependent on primary key.

What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD $X \rightarrow A$, X must be a candidate key.

What is 4NF?

A relation schema R is in 4NF if it is in BCNF and there are no multivalued dependencies.

What is 5NF or Project-Join Normal Form (PJNF)?

A relation schema R is in 5NF if it is in 4NF and won't have lossless decomposition into smaller tables.

What is a view in SQL?

A view is a virtual table based on the result-set of an SQL statement.

What is a view, and why use it?

- A view is defined as a database object that allows us to create a virtual table in the database whose contents are defined by a query or taken from one or more tables.
- Views take very little space, simplify complex queries, limit access to the data for security reasons, enable data independence, and summarize data from multiple tables.

What is a transaction? What are ACID properties?

A **transaction** in DBMS is a sequence of one or more SQL operations executed as a single unit of work. A transaction ensures data integrity, consistency, and isolation, and it guarantees that the database reaches a valid state, regardless of errors or system failures.

Properties of a transaction (ACID Properties):

- **Atomicity:** All operations within the transaction are completed successfully, or none are applied (i.e., the transaction is atomic).
- **Consistency:** The transaction brings the database from one valid state to another valid state.
- **Isolation:** The operations of one transaction are isolated from others; intermediate results are not visible to other transactions.
- **Durability:** Once a transaction is committed, its effects are permanent, even in the event of a system crash.

What is a cursor in SQL?

A cursor is a database object used to retrieve, manipulate, and traverse through rows in a result set one row at a time. Cursors are helpful when performing operations that must be processed sequentially rather than in a set-based manner.

What are cursors give different types of cursors?

PL/SQL uses cursors for all database information accesses statements. The language supports the use two types of cursors:

- 1.) Implicit
- 2.) Explicit

What is a trigger in SQL?

A trigger is a set of SQL statements that automatically execute in response to certain events on a table, such as INSERT, UPDATE, or DELETE. Triggers help maintain data consistency, enforce business rules, and implement complex integrity constraints.

What is a stored procedure?

A stored procedure is a precompiled set of SQL statements stored in the **database**. It can take input parameters, perform logic and queries, and return output values or result sets. Stored procedures improve **performance** and **maintainability** by centralizing business logic.

What is the difference between a trigger and a stored procedure?

Trigger:

- A trigger is an automatic action executed by the DBMS when a specific event occurs on a table, such as an INSERT, UPDATE, or DELETE.
- It cannot be invoked manually and is tied to a specific event.

Stored Procedure:

- A stored procedure is a precompiled set of SQL statements that can be executed explicitly by a user or application.
- It is invoked manually, and it can accept input parameters and return output.

What is the role of the Database Administrator (DBA)?

A **Database Administrator (DBA)** is responsible for managing and overseeing the entire database environment. Their key responsibilities include:

- **Database Design:** Structuring the database for optimal storage and performance.
- **Backup and Recovery:** Ensuring regular backups and providing recovery solutions in case of failures.
- **Performance Tuning:** Monitoring and optimizing the database's performance.
- **Security Management:** Managing user access, privileges, and enforcing security policies.
- **Data Integrity:** Ensuring data consistency and integrity through constraints and checks.
- **Upgrades and Patches:** Keeping the database software up-to-date with patches and upgrades.
- **Troubleshooting:** Identifying and resolving database-related issues.