

DATABASE:

A database is a data structure that stores organized information. Databases contain multiple tables, which may each include several different fields.

For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table.

The purpose of a database is to help our business stay organized and keep information easily accessible, so that you can use it.

Where do we use database?

Databases are used just about everywhere including **banks**, retail, websites and warehouses. **Banks** use databases to keep track of customer accounts, balances and deposits. Retail stores can use databases to store prices, customer information, sales information and quantity on hand.

What is SQL and why it is used?

Structured Query Language (SQL).

It is used to manage and organize data in all sorts of systems in which various data relationships exist. SQL is a valuable programming language with strong career prospects.

QUERY

A **query** is a way of requesting information from the **database**.

SUBQUERY

A subquery is a `SELECT` statement nested in another statement.

SELECT

SELECT clause is the list of columns or SQL expressions that must be returned by the query.

ORDER BY CLAUSE

The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns.

DISTINCT KEYWORD

The `distinct` keyword allows you to remove the duplicate rows in the result set.

WHERE CLAUSE

The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fulfill a specified condition.

BETWEEN OPERATOR

The `BETWEEN` operator is a logical operator that tests whether a value is in range of values.

IN OPERATOR

The IN operator determines whether a value matches any value in a list or a subquery

AND, OR AND NOT OPERATOR

The AND and OR operators are used to filter records based on more than one condition:

- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

GROUP BY CLAUSE

The GROUP BY statement groups rows that have the same values into summary rows.

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

NESTED GROUP FUNCTION

which means that one group function can enclose an expression that is itself a group operation on another expression or column.

HAVING CLAUSE

The **HAVING** clause specifies a search condition for a group.
You often use the **HAVING** clause with the **GROUP BY** clause

INNER/EQUI JOIN CLAUSE

In relational databases, data is often distributed in many related tables. A table is associated with another table using [foreign keys](#).

To [query data](#) from multiple tables, you use **INNER JOIN** clause. The **INNER JOIN** clause combines columns from correlated tables.
Suppose you have two tables: A and B.

A has a1, a2, and f columns. B has b1, b2, and f column. The A table links to the B table using a foreign key column named f.

NON-EQUI JOIN CLAUSE

A **Join** that has at least one **comparison in the join** condition that is not an "equals" **comparison** is called a **Nonequi-Join**.

SELF-JOIN

The self-join is a special kind of joins that allow you to join a table to itself

The self-join compares values of the same or different columns in the same table. Only one table is involved in the self-join.

Key is also helpful for finding unique record or row from the table.

- **Primary Key** - is a column or group of columns in a table that uniquely identify every row in that table.
- **Foreign Key** - is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.
- **Candidate Key** - is a set of attributes that uniquely identify tuples in a table.

Candidate Key is a super key with no repeated attributes.

Candidate key is also a unique key to identify a record uniquely in a table but a table can have multiple candidate keys.

- **Composite Key** - A primary key having two or more attributes is called composite key.

A table can have only one primary key, which may consist of single or multiple fields. multiple fields are used as a primary key, they are called a composite key.

An example can be – Here our composite key is OrderID and ProductID – {OrderID, ProductID}

Normalization

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. Normalization divides the larger table into the smaller table and links them using relationship.

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Second Normal Form (2NF)

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Partial Dependency, where an attribute in a table depends on only a part of the primary key and not on the whole key.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

Transitive Dependency. When a non-prime attribute depends on other non-prime attributes rather than depending upon the prime attributes or primary key

Transitive Dependency: When an indirect relationship causes functional dependency it is called Transitive Dependency.

If $P \rightarrow Q$ and $Q \rightarrow R$ is true, then $P \rightarrow R$ is a transitive dependency.

To achieve 3NF, eliminate the Transitive Dependency.

Partial Dependency: Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key.

The 2nd Normal Form (2NF) eliminates the Partial Dependency.

Functional Dependency : A functional dependency in a database enforces a set of constraints between attributes.

This occurs when one attribute in a relation uniquely determines another attribute. This can be written $A \rightarrow B$ which means "B is functionally dependent upon A." This is also called a database dependency.