SQL

What is Database

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

Database Management System

Database management system is a software which is used to manage the database. For example: MySQL, Oracle, etc are a very popular commercial database which is used in different applications.

DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.

It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

SQL

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

DBMS AND RDBMS

DBMS stores data in the form of files. There may not be relation between the data. RDBMS (Relational DBMS) stores data in the form of tables and the tables can be related using a common column or key. This is the main difference.

SQL Commands

SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

Types of SQL Commands

There are 4 types: DDL, DML, DCL, TCL.

DDL – Data Definition Language

Statements used to create, alter, drop database objects.
CREATE/ALTER/DROP/TRUNCATE TABLE
Note: All the commands of DDL are auto-committed that means it permanently saves all the changes in the database.

DML – Data Manipulation Language

Statements used to retrieve, insert, remove and manage data within DB objects.
SELECT/INSERT/UPDATE/DELETE/MERGE
Note: The commands of DML are not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

DCL – Data Control Language

Statements used to control the access of data stored in database and provide data security.
GRANT/REVOKE/EXECUTE AS/REVERT

TCL – Transaction Control Language

Statements used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.
COMMIT/ROLLBACK/SAVEPOINT

DDL COMMANDS

CREATE is used to create a new table in the database.

Syntax: CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

Example: Create Employee table with name, email and date of birth.
> create table Employee(name varchar2(20), email varchar2(100), dob date);

To see the structure of this table:
> desc Employee;

To store the data into this table:
> insert into Employee values('Srinu', 'srinu123@gmail.com', '10-JUN-1990');

ALTER is used to alter (change) the structure of the table. This change could be either to modify an existing column or to add a new column.

Syntax: ALTER TABLE TABLE_NAME ADD(COLUMN DEFINITION);
ALTER TABLE TABLE_NAME MODIFY(COLUMN DEFINITION);
Example: to add a new column 'address':
> alter table Employee add(address varchar2(100));
To modify the name column as name varchar2(15):
>alter table Employee modify(name varchar2(15));

Note: Since a new column 'address' is added, let us store the value into it.
> update Employee set address = 'HNO-33, Ameerpet, Hyderabad' where name= 'Srinu';

TRUNCATE is used to delete all the rows from the table. But table structure remains.

Syntax: TRUNCATE TABLE table_name;
Example:
> truncate table Employee;

DROP is used to delete both the structure and records stored in the table.

Syntax: DROP TABLE;
Example: To delete the Employee table:
> drop table Employee;

DML COMMANDS

INSERT is used to insert data into the row of a table.

Syntax:
INSERT INTO TABLE_NAME (col1, col2, col3,.... colN)
VALUES (value1, value2, value3, ... valueN);
Syntax:
INSERT INTO TABLE_NAME VALUES (value1, value2, value3, ... valueN);

Example: Let us insert data into Employee table
> create table Employee(name varchar2(20), dob date, sal float);
> insert into Employee(name, dob) values('Vishnu', '22-JAN-1985');
> insert into Employee values('Laxmi', '19-APR-1977', 34500.75);

UPDATE is used to update or modify the value of a column in the table.

Syntax:
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]
Example: to update the salary of Vishnu as 25000.00:
> update Employee set sal=25000.00 where name='Vishnu';
Example: to update the existing dob of Laxmi:
> update Employee set dob='29-APR-1977' where name='Laxmi';

DELETE is used to remove one or more rows from a table.

Syntax:
DELETE FROM table_name [WHERE condition];

Example: To delete rows of 'Laxmi':
> delete from Employee where name = 'Laxmi';

SELECT is used to select the columns / rows based on the condition described by WHERE clause.
Note: SELECT is also called DQL (Data Query Language) command.

Syntax:

SELECT expressions
FROM TABLE
WHERE conditions;

Examples: To select all rows from Employee table:
First insert: insert into Employee values('Anil', '10-OCT-1980', 34500.75);
> select * from Employee;
To select only names from table:
> select name from Employee;
To select only names and dobs where salary is more than 25000:
> select name, dob from Employee where sal > 25000;

DCL COMMANDS

GRANT is used to give user access privileges to a table.

Example:
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;

REVOKE is used to take back permissions from the user.

Example:
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;

TCL COMMANDS

COMMIT command is used to save all the transactions to the database.

Syntax:
COMMIT;
Example:
> delete from Employee where name = 'Vishnu';
> commit;

ROLLBACK command is used to undo transactions that have not already been saved to the database.

Syntax:
ROLLBACK;
Example:
> insert into Employee values('Amrut', '5-MAY-1970', 45000.00);
> select * from Employee;
> rollback;
> select * from Employee;

SAVEPOINT is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

SAVEPOINT SAVEPOINT_NAME;

SQL OPERATORS

SQL Arithmetic Operators

Let's assume 'variable a' and 'variable b'. Here, 'a' contains 20 and 'b' contains 10.

Operator	Description	Example
+	It adds the value of both operands.	a+b will give 30
-	It is used to subtract the right-hand operand from the left-hand operand.	a-b will give 10
*	It is used to multiply the value of both operands.	a*b will give 200
/	It is used to divide the left-hand operand by the right-hand operand.	a/b will give 2
%	It is used to divide the left-hand operand by the right-hand operand and returns reminder.	a%b will give 0

SQL Comparison Operators

Let's assume 'variable a' and 'variable b'. Here, 'a' contains 20 and 'b' contains 10.

Operator	Description	Example
=	It checks if two operands values are equal or not, if the values are queal then condition becomes true.	(a=b) is not true
!=	It checks if two operands values are equal or not, if values are not equal, then condition becomes true.	(a!=b) is true
<>	It checks if two operands values are equal or not, if values are not equal then condition becomes true.	(a<>b) is true
>	It checks if the left operand value is greater than right operand value, if yes then condition becomes true.	(a>b) is true
<	It checks if the left operand value is less than right operand value, if yes then condition becomes true.	(a<b) is not true
>=	It checks if the left operand value is greater than or equal to the right operand value, if yes then condition becomes true.	(a>=b) is true

<=	It checks if the left operand value is less than or equal to the right operand value, if yes then condition becomes true.	(a<=b) is not true
!<	It checks if the left operand value is not less than the right operand value, if yes then condition becomes true.	(a!<b) is true
!>	It checks if the left operand value is not greater than the right operand value, if yes then condition becomes true.	(a!>b) is not true

SQL Logical Operators

This is the list of logical operator used in SQL:

Operator	Description
ALL	It compares a value to all values in another value set.
AND	It allows the existence of multiple conditions in an SQL statement.
ANY	It compares the values in the list according to the condition.
BETWEEN	It is used to search for values that are within a set of values.
IN	It compares a value to that specified list value.
NOT	It reverses the meaning of any logical operator.
OR	It combines multiple conditions in SQL statements.
EXISTS	It is used to search for the presence of a row in a specified table.
LIKE	It compares a value to similar values using wildcard operator.