**What is SQL:-** SQL is a standard language for accessing and manipulating databases. SQL stands for Structured Query Language Used to interact with database.

**Use of SQL:-**Mainly used for CRUD purpose

C-Create R-Read U-Update D-Delete

**SQL Application:-**Retrieve create or manipulation.

Banking application database , E-Commerse website

**What is database:-**Database is a system that allow user to Store and organize zata

**Why use database:-**

**Types of database:-**

1.Relational database.:-store data in the form of table

2.Non relational database.:-Store data into key value pair

**use of EXCEL:-**

-Small amount of data. -Untrained persons.

-One time analysis. -Quick chart or graph.

**Use of Database:-**

-Large amount of data . -Store real time data from websites/Apps.

-Easy and safe access. -Easily combine with different dataset.

-Data integrity. -Automate steps and can re-use.

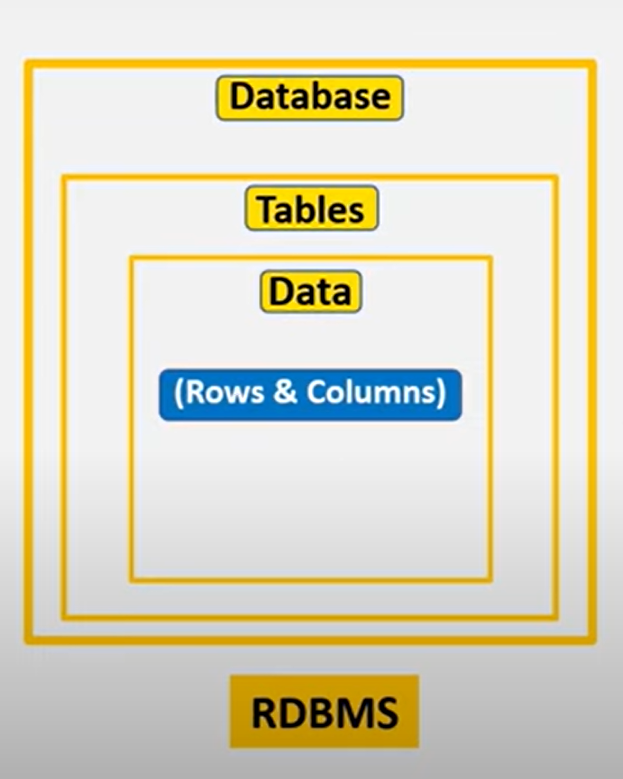
-Data search Capabilities.

**Different IDE-**

-MYSQL -SQLite -PostgreSQL -ORACLE Pl/sql -SQLServer

-snowflake -AMzon redshift

**SQL Structure:-**



**Data types:-**

Data type of column defines what value the column can store in table.

Defined while creating database.

Data types mainly classified into 3 main categories.

-**String**:-char, varchar etc.

-**Numeric**:-int, float, Boolean etc.

-**Date and time**:-date,datetime etc.

Commonly used datatypes in SQL:-

**int:-**Used for integer value.

**float:-**Used to specify a decimal point number.

**bool:-**used to specify Boolean values true or false.

**char:-**fixed length string that can contain numbers,letters and special characters.

**varchar:-**variable length string that can contain numbers,letters and special character.

**date:-**date format YYYY-MM-DD .

**datetime:-**date and time combination format-YYYY-MM-DD hh:mm:ss.

**Primary Key(PK):-**

In SQL, a primary key is a column or a set of columns that uniquely identifies each record in a table.

Table can have only one primary key which should be unique and not null.

**Foreign key:-**Used to link two or more tables together

Table can have any number of foreingn key.

It can contain duplicate and null value.

**First table:-**

CREATE TABLE empTable (

eid int not null auto\_increment primary key,

ename varchar(30)

);

**Second table:-**

CREATE TABLE department(

did int primary key,

dname varchar(30),

empid int not null,

foreign key(empid)references emptable(eid)

);

**Constraint:-**Constraints are used to specify rules for data in a table.

-This insures accuracy and reliability of the data in the table.

-Constraints are specified when the table is created with the create table statement Or after the table creation alter table statement.

**Ex:-**Create table table\_name(

Col1 dattype **constraint,**

Col2 dattype **constraint,**

…

);

Commonly used constraints:-

**Not null :-**Ensures that column can not have null value.

**Unique:-**Ensure all values in column is different.

**Primary Key:-**combination of not null and unique.

**Foreign Key:-**used to link multiple tables together.

**Check:-** Ensures the value in the column satisfy specific condition.

**Default:-**Set default value to column if no value is specified.

**Create Index:-**used to create and retrieve data from the database very quick.

**SQL QUERIES**

1. **SHOW DATABASE;**

This query is used to show all the databases .

1. **CREATE DATABASE;**

**CREATE DATABASE Database\_name;**

This query is used to create database.

1. **USE DATABASE;**

**USE Database\_name;**

For using particular database this query is used.

1. **SHOW TABLES;**

Showing all the tables inside the database this query is used.

**DDL command**- DDL stands for Data Definition Language, and it's a subset of SQL (Structured Query Language) used to define and manage the structure of database objects. DDL commands are used to create, modify, and delete database objects such as tables, indexes, views, and schemas.

**[Create,Alter,Truncate,Drop,Rename]**

1. **CREATE TABLE :-** This query is used to create table in to the database.

**Syntax-**

CREATE TABLE table\_name (

column1 datatype constraints,

column2 datatype constraints,

...

);

**EX-**

CREATE TABLE student(

Id int primary key,

Name varchar(20),

Marks double

);

1. **Rename:-** Used to rename an existing database object. Example (renaming a table)

**Syntax:-** ALTER TABLE old\_table\_name RENAME TO new\_table\_name;

**EX:-**ALTER TABLE employeedetail RENAME TO empdetails;

1. **ALTER TABLE:-** Used to modify the structure of an existing table. You can add, modify, or drop columns, constraints, and more.

**1.Add new Column.**

**Syntax:**-ALTER TABLE TABLE\_NAME ADD COLUMN\_NAME DATATYPE;

**Ex:-**ALTER TABLE employees ADD COLUMN email VARCHAR(100);

**2. Modify an existing column(ex. change datatype)**

**Syntax:-** ALTER TABLE Table\_Name MODIFY Column\_Name new\_datatype;

**EX:-**ALTER TABLE sudent MODIFY marks double;

**3.Drop a Column**

**Syntax:-**ALTER TABLE Table\_Name DROP Column\_Name;

**EX:-**ALTER TABLE student DROP attendance;

**4.Alter View**

**Synatx:-** ALTER VIEW sales\_view AS SELECT column1, column2 FROM new\_table WHERE condition;

**EX:-**

1. **Truncate table:-**Delete all table and Structure remains same.

**Syntax:-**TRUNCATE TABLE TABLE\_NAME;

**Ex:-**TRUNCATE TABLE student;

1. **DROP TABLE:-** Used to delete an existing table along with its data and structure. **[DATABASE,VIEW,INDEX,PROCEDURE,FUNCTION]**

**Syntax:-**DROP TABLE TABLE\_NAME;

**EX:-** DROP TABLE employees;

1. **CREATE INDEX:-**Create an index on a table.

**Syntax:-**CREATE INDEX index\_name ON table\_name (column\_name);

**EX:-**CREATE INDEX idx\_department ON Employees (Department);

**DML (Data Manipulation Language) command[Insert , Update, Delete]**

1. **Update Table:-**The UPDATE statement modifies existing records in a table.

**Syntax: -**UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;

**EX:-** UPDATE employees SET salary = 55000 WHERE first\_name = 'John’;

1. **INSERT:-**Used to insert record in table.

**Syntax:-**INSERT INTO TABLE\_NAME(COL1,COL2,…)VALUES(VAL1,VAL2,…);

**EX:-**INSERT INTO EMPLOYEE(EMP\_ID, EMP\_NAME, EMP\_AGE) VALUES (1,’John’,30);

1. **DELETE:-**The DELETE statement removes records(row) from a table. And structure remains same.

**Syntax:-** DELETE FROM table\_name WHERE condition;

**1. Delete only single row :-**

**Ex:-**DELETE FROM employees WHERE last\_name = 'Doe';

**2. Delete multiple rows:-**

**EX:-**DELETE FROM STUDENT WHERE NUM>5 && NUM<11;

3**. Delete all rows** :-structure is remaining (data removed):-

**EX:-** DELETE FROM class;

1. **Select:-**Retrive data from one or more table.

**Synatx:-**Select \* from table\_Name where condition;

**EX:-**Select \* from student where marks>80;

**DCL (data control language)command[Grant, Revoke]**

1. **Grant:-**Grant specific privileges to users or role.

**Syntax:-** GRANT SELECT, INSERT ON employees TO user1;

**EX:-**

1. **Revoke:-**Revokes previously granted privileges.

**Synatx:-** REVOKE INSERT ON employees FROM user1;

**EX:-**

**TCL (Transaction control language)commands:-[Commit , Rollback , SavePoint , Set Transaction]**

1. **COMMIT:-** save changes made during transaction.

**Syntax:-** COMMIT;

**EX:-**

1. **ROLLBACK:-** Discards changes made during a transaction.

**Syntax:-** ROLLBACK;

**EX:-**

1. **SAVEPOINT:**-Sets a savepoint within a transaction.

**Syntax:-** SAVEPOINT savepoint\_name;

**EX:-**

1. **SET TRANSACTION:-**properties for the current transaction.

**Syntax:-** SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

**EX:-**

**DQL (Data query language)Commands:-[Select, Comment]**

1. **Select:-**Retrive data from table based on specific criteria.

**Syntax:-**Select \* from student;

1. **Comment:-** Used to add comments to database objects for documentation purposes.Example (adding a comment to a table):

**Syntax:**- COMMENT ON TABLE employees IS 'This table stores employee information.';

**Where Clause in SQL:-**used to filter records It is used to extract only those records that fulfill a specified condition

-To define some conditions.

**Syntax:-**SELECT col1, col2 FROM table\_name WHERE conditions;

**EX:-** SELECT \* FROM STUDENT WHERE MARKS > 80;

Using Operators in WHERE

Arithmetic operators : +, -, \*, /, %

Comparison Operators : = (equal to), <> (not equal to), > , >= , <=

Logical Operators : AND, OR , NOT, IN, BETWEEN, ALL, LIKE, ANY

Bitwise Operators : & (Bitwise AND), | (Bitwise OR)

**Order By Clause in SQL**

To sort in ascending (ASC) or descending order (DESC)

**Syntax:-**Select Column\_Name From Table\_Name Order By Column\_Name Desc;

**EX:-** SELECT \* FROM EMPLOYEE ORDER BY Emp\_Salary, Emp\_Name ASC;

**EX:-**SELECT \* FROM EMPLOYEE ORDER BY Emp\_Salary, Emp\_Name DESC;

**Distinct:-**Select distinct values from column (different).

Syntax:-Select Distinct Column from Table\_name;

EX:-Select Distinct department from empdetails;

Select Count(Distinct department) as count from empdetails;

**LIKE OPERATOR:-**

**LIMIT clause:-** It is used to restrict the number of rows returned by a query, which can be useful when dealing with large datasets or when you only need a subset of the results.

**Syntax:-** SELECT column1, column2, ... FROM table\_name LIMIT number\_of\_rows;

**EX:-**

**Joins:-**

**Nested Queries:-**

**Operators in sql:-**

SQL reserved words and characters are called operators, which are used with a WHERE Clause in SQL query;

**Arithmetic Operators:-**

These operators perform arithmetic operations on numeric data types.

* + (Addition)
* -(Subtraction)
* \*(Multiplication)
* / (Division)
* % (Modulo - Returns the remainder of a division)

**Comparison Operators:-**

These operators compare values and return a Boolean result (TRUE or FALSE).

* = (Equal to):- **where col=condition;**
* != or <> (Not equal to) :- **where col<>condition;**
* < (Less than) :- **where col<condition;**
* >(Greater than) :- **where col>condition;**
* <= (Less than or equal to) :- **where col<=condition;**
* >= (Greater than or equal to) :- **where col=>condition;**

**Logical Operators:-**

These operators are used to combine multiple conditions in a WHERE clause.

* AND (Logical AND)
* OR (Logical OR)
* NOT (Logical NOT)

**Wildcard Operators:-**

Wildcards are used with the LIKE operator to match patterns in strings.

* % (Matches zero or more characters)
* \_ (Matches a single character)

**IN Operator:-** Checks whether a value matches any value in a list.

* IN

**BETWEEN Operator:-**Checks whether a value is within a range.

* BETWEEN:-

1. select \* from empdetails where salary between 12000 and 20000;
2. select \* from empdetails where salary between 20000 and 30000;

**IS NULL Operator:-**Checks if a value is NULL.

* IS NULL

**IS NOT NULL Operator:-** Checks if a value is not NULL.

* IS NOT NULL

**Assignment Operator:**-Assigns a value to a variable.

* :=

**Concatenation Operator:-**Concatenates strings.

* || or CONCAT

**Import CSV files**

**Functions in sql:-**

In SQL, functions are predefined operations that operate on one or more values, input parameters, or rows and return a single value. These functions can be used in SQL queries to perform various tasks like calculations, string manipulation, date and time operations, and more.

Types of function:-

1. System defined functions:-this are built in functions-

EX:-rand(),round(),upper(),lower(),count(),sum(),max() etc.

1. User-Definde Function:-Once you define a function that can call it in the way as the built-in functions.

**String functions in sql:-**Mostly used String function

Used to perform operation on input string and return and output string.

1. **LOWER() and UPPER():**-These functions convert a string to lowercase or uppercase, respectively.

**Syntax:-**

Select **Upper**(Coumn\_Name) from Table\_Name;

Select **Lower**(Column\_Name) from Table\_Name;

SELECT UPPER('world') AS UpperCase; -- Outputs: WORLD

SELECT LOWER('Hello') AS LowerCase; -- Outputs: hello

1. **LENGTH():** Returns the length of a string (number of characters).

1.SELECT **LENGTH**('Hello') AS StringLength; -- Outputs: 5

1. **SUBSTRING() or SUBSTR():** Extracts a substring from a string based on a specified starting position and length.

1.SELECT **SUBSTRING**('Hello, world', 1, 5) AS SubstringResult; -- Outputs: Hello

2.SELECT **SUBSTR**('Hello, world', 8) AS SubstringResult; -- Outputs: world

1. **CONCAT() OR ||:** Concatenates two or more strings together.

1.SELECT **CONCAT(**'Hello', ' ', 'world'**)** AS ConcatenatedString; -- Outputs: Hello world

**Oracle also uses || for string concatenation, similar to PostgreSQL.**

SELECT column1 || ' ' || column2 AS concatenated\_string

FROM your\_table;

1. **TRIM():** Removes leading and trailing spaces or specified characters from a string.

1.SELECT **TRIM(**'x' FROM 'xxHelloxx'**)** AS TrimmedString; -- Outputs: Hello

1. **REPLACE():** Replaces occurrences of a substring within a string with another substring.

1.SELECT **REPLACE(**'Hello world', 'world', 'SQL'**)** AS ReplacedString; -- Outputs: Hello, SQL

1. **CHARINDEX() or POSITION():** Returns the starting position of a substring within a string.

1.SELECT **CHARINDEX(**'world', 'Hello, world'**)** AS Position; -- Outputs: 7

2.SELECT **POSITION(**'world' IN 'Hello, world'**)** AS Position; -- Outputs: 7

1. **LEFT() and RIGHT():** Extracts a specified number of characters from the left or right side of a string.

1.SELECT LEFT('Hello, world', 5) AS LeftString; -- Outputs: Hello

2.SELECT RIGHT('Hello, world', 5) AS RightString; -- Outputs: world

Select id,name from empdetails where salary>10000;

Count:-Syntax:-select count(role) from empdetails where role=’dev’;

Select sum(salary) from empdetails;

Select \* from empdetails where name like’R%’;

Select salary from empdetails where order by salary;

Select \* from empdetails where order by salary desc;

Select distinct department from empdetails;

MIN()-return smallest within selected columns.

MAX()-

COUNT()-

SUM()-

AVG()-