Video-7

Topics to cover:

- WHERE Clause
- UPDATE Command
- DELETE Command
- Use of Auto Increment
- Use of LIMIT & OFFSET

WHERE Clause

The WHERE clause in SQL is used to filter records that meet specific conditions. It's commonly used with SELECT, UPDATE, DELETE, and other statements to target only the rows you want.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

 # Example: SELECT with WHERE

```
SELECT * FROM employee
WHERE age = 25;
```

 # Example: SELECT with WHERE

```
SELECT * FROM employee
WHERE hiring_date = "2021-08-10";
```

Creating employee table to understand different Commands

```
Create table employee (
        Id int,
        Name varchar(50),
        Age int,
        Hiring_date date,
        Salary int
);

insert into employee values(1,"Ankit", 24, "2021-08-10", 10000);
insert into employee values(2,"Rahul", 25, "2021-08-10", 20000);
insert into employee values(3,"Sunny", 22, "2021-08-11", 11000);
insert into employee values(4,"Amit", 25, "2021-08-11", 12000);
insert into employee values(5,"Mohit", 26, "2021-08-12", 50000);

select * from employee;

desc employee;
```

UPDATE Command

The UPDATE command in SQL is used to modify existing records in a table.

Syntax:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
```

Syntax with where clause:

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

UPDATE Command

 # Updates will be made for all rows:

```
UPDATE employee
SET age = 20;

SET SQL_SAFE_UPDATES = 0;

select * from employee;
```

 # Update the salary of employee after giving 20% increment

```
UPDATE employee
SET salary = salary * 1.2;
```

 # Update command for multiple columns

```
UPDATE employee
SET salary = 1.2, age=24;
```

UPDATE Command

 # Update a single column

```
UPDATE employee
SET salary = 30000
WHERE id = 1;
```

 # Update multiple columns

```
UPDATE employee
SET salary = 80000, age = 30
WHERE id = 3;
```

DELETE Command

The DELETE command in SQL is used to remove one or more rows from a table based on a condition.

Syntax:

```
DELETE FROM table_name
WHERE condition;
```

 # Delete a specific record

```
DELETE FROM employee
WHERE id = 1;
```

 # Delete multiple records

```
DELETE FROM employee
WHERE Hiring_date = "2021-08-11";
```

 # Delete all rows (you can also use truncate here as truncate is much faster than delete)

```
DELETE FROM employee;
```

Auto increment

The AUTO_INCREMENT attribute in SQL is used to automatically generate a unique value for a column whenever a new row is inserted. It's commonly applied to primary key columns like id.

Syntax:

```
CREATE TABLE employees (
  id INT AUTO_INCREMENT,
  name VARCHAR(100),
  PRIMARY KEY (id)
);
```

 # Each time you insert a new row, employee_id will automatically increment from the last value.

```
INSERT INTO employees (name)
VALUES ('Ankit'), ('Mohit');
```

Auto increment

```
insert into employees (id, name)
values(5,"Amit");

insert into employees (name)
values("Nikhil");

Select * from employees;
```

Use of LIMIT & OFFSET

The LIMIT and OFFSET clauses in SQL are used to control how many rows are returned and where to start retrieving them.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
LIMIT number_of_rows OFFSET start_position;
```

 # Get the first 3 rows:

```
SELECT * FROM employee
LIMIT 3;
```

 # Skip the first 2 rows and return the next 3

```
SELECT * FROM employee
LIMIT 3 OFFSET 2;
```

 # Alternative Syntax

```
SELECT * FROM employee
LIMIT 2, 3;
```