








# SQL Video 2

|  |   |
|--|---|
|  Created by       |   |
|  Created time     | @January 6, 2023 12:16 PM   |
|  Last edited by   |   |
|  Last edited time | @January 11, 2023 1:08 AM   |
|  Status          | New   |
|  Tags           | SQL   |
|  URL            | <a href="https://www.youtube.com/watch?v=wTXem0vg0Ak&amp;list=PLtgiThe4j67rAoPmnCQmcgLS4ilc5ungg&amp;index=3">https://www.youtube.com/watch?v=wTXem0vg0Ak&amp;list=PLtgiThe4j67rAoPmnCQmcgLS4ilc5ungg&amp;index=3</a> |

## CRUD Operations

Create , Read, Update and Delete (CRUD)

Create - Insert Statements

Read - Select statements

Update - Update statements

Delete - Delete Statements

## Creation of Table and Insert Statements

*Scenario - You are a Bangalore-based startup and you are looking to hire employees **preferably from Bangalore** and maintain a record of the employees in the table*

This command is called a **Data Definition Language (DDL)** command as we are defining a structure to store the data.

```
CREATE TABLE employee(  
  firstname varchar(20),  
  middlename varchar(20),  
  lastname varchar(20),  
  age int,  
  salary int,  
  location varchar(20)  
)
```

To see all records in a table

```
SELECT * FROM employee;
```

## Inserting Data into Tables

```
INSERT INTO employee (firstname, middlename, lastname, age, salary, location)  
VALUES ('Kapil', 'Kumar', 'Sharma', 28, 10000, 'Bangalore');  
  
//ANOTHER WAY WITHOUT MENTIONING COLUMNS  
  
INSERT INTO employee  
VALUES ('Kapil', 'Kumar', 'Sharma', 28, 10000, 'Bangalore');
```

## Why using the Column List is useful?

Consider a scenario where the person doesn't have a middlename. So that field will be blank while inserting into the row. So if we don't specify the columns and corresponding values an error will be thrown as the number of values and number of columns by default don't match.

```
mysql> INSERT INTO employee  
-> VALUES ('Kapil','Sharma',28,10000,'Bangalore');  
ERROR 1136 (21S01): Column count doesn't match value count at row 1
```

So mentioning column list is important as it maps values to the column and we can enter only the required values.

```
mysql> INSERT INTO employee (firstname,lastname, age, salary, location)  
-> VALUES ('Kapil','Sharma',28,10000,'Bangalore');  
Query OK, 1 row affected (0.00 sec)
```

The field for which the data was not inserted, it is field with **NULL**.

NULL is not defined data. It is not zero or any other value.

```
mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| firstname | middlename | lastname | age | salary | location |
+-----+-----+-----+-----+-----+-----+
| Kapil      | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Satish     | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil      | NULL       | Sharma   | 28  | 10000  | Bangalore |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## Putting Single Quotes(') in Values

- Use **Double Quotes("")** to enclose values.

This makes single quotes a part of the value to be entered

```
mysql> INSERT INTO employee
      -> VALUES ("Kapil's", 'Kumar', 'Sharma', 28, 10000, 'Bangalore');

Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| firstname | middlename | lastname | age | salary | location |
+-----+-----+-----+-----+-----+-----+
| Kapil     | Kumar     | Sharma   | 28  | 10000  | Bangalore |
| Satish    | Kumar     | Sharma   | 28  | 10000  | Bangalore |
| Kapil     | NULL      | Sharma   | 28  | 10000  | Bangalore |
| Kapil's   | Kumar     | Sharma   | 28  | 10000  | Bangalore |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

- Another Way

We can use Single Quotes only but still enter it as a value using **Escape Character (\)**.

```
mysql> INSERT INTO employee
-> VALUES ('Kapil\'s','Kumar','Sharma',28,10000,'Bangalore')
;
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| firstname | middlename | lastname | age | salary | location |
+-----+-----+-----+-----+-----+-----+
| Kapil      | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Satish     | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil      | NULL       | Sharma   | 28  | 10000  | Bangalore |
| Kapil's    | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil's    | Kumar      | Sharma   | 28  | 10000  | Bangalore |
+-----+-----+-----+-----+-----+-----+
+
```

## Entering Multiple Values (Bulk Insert)

Use a comma and within brackets enter the values respective of the columns.

```
mysql> INSERT INTO employee (firstname, middlename, lastname, age, salary, location) VALUES ('Kapil', 'Kumar', 'Sharma', 28, 10000, 'Bangalore'), ('Satish', 'Kumar', 'Sinha', 30, 20000, 'Bangalore');
Query OK, 2 rows affected (0.02 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> select * from employee;
```

| firstname | middlename | lastname | age | salary | location  |
|-----------|------------|----------|-----|--------|-----------|
| Kapil     | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Satish    | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil     | NULL       | Sharma   | 28  | 10000  | Bangalore |
| Kapil's   | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil's   | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Kapil     | Kumar      | Sharma   | 28  | 10000  | Bangalore |
| Satish    | Kumar      | Sinha    | 30  | 20000  | Bangalore |

## NULL Restriction on Columns

```
CREATE TABLE employee(
  firstname varchar(20) NOT NULL,
  middlename varchar(20),
  lastname varchar(20) NOT NULL,
  age int NOT NULL,
  salary int NOT NULL,
```

```
location varchar(20) NOT NULL
);
```

```
mysql> CREATE TABLE employee(
->  firstname varchar(20) NOT NULL,
->  middlename varchar(20),
->  lastname varchar(20) NOT NULL,
->  age int NOT NULL,
->  salary int NOT NULL,
->  location varchar(20) NOT NULL
-> );
```

Query OK, 0 rows affected (0.04 sec)

```
mysql> desc employee;
```

| Field      | Type        | Null | Key | Default | Extra |
|------------|-------------|------|-----|---------|-------|
| firstname  | varchar(20) | NO   |     | NULL    |       |
| middlename | varchar(20) | YES  |     | NULL    |       |
| lastname   | varchar(20) | NO   |     | NULL    |       |
| age        | int         | NO   |     | NULL    |       |
| salary     | int         | NO   |     | NULL    |       |
| location   | varchar(20) | NO   |     | NULL    |       |

6 rows in set (0.00 sec)

We restrict the column and set it to **NOT NULL** and Null values cannot be inserted.

```
mysql> INSERT INTO employee(middlename,lastname,age,salary,location) VALUES('kumar','sharma',28,20000,'Bangalore');
ERROR 1364 (HY000): Field 'firstname' doesn't have a default value
```

## Default Values



*As discussed, we are hiring employees mostly from Bangalore. So why enter location everytime as most employees are from Bangalore. We can do something like this where by default the location will be Bangalore, and in case some other location comes, we mention it explicitly.*

```
CREATE TABLE employee(  
  firstname varchar(20) NOT NULL,  
  middlename varchar(20),  
  lastname varchar(20) NOT NULL,  
  age int NOT NULL,  
  salary int NOT NULL,  
  location varchar(20) DEFAULT 'Bangalore'  
);
```

```
mysql> desc employee;
```

| Field      | Type        | Null | Key | Default   | Extra |
|------------|-------------|------|-----|-----------|-------|
| firstname  | varchar(20) | NO   |     | NULL      |       |
| middlename | varchar(20) | YES  |     | NULL      |       |
| lastname   | varchar(20) | NO   |     | NULL      |       |
| age        | int         | NO   |     | NULL      |       |
| salary     | int         | NO   |     | NULL      |       |
| location   | varchar(20) | YES  |     | Bangalore |       |

6 rows in set (0.01 sec)

Both the scenarios are covered

- First command we do not mention location and Bangalore is taken as the value.

- Second command we explicitly mention **Hyderabad** and that value is entered instead of the DEFAULT value.

```
mysql> INSERT INTO employee (firstname,lastname, age, salary) VA
LUES ('Kapil','Sharma',28,10000);
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| firstname | middlename | lastname | age | salary | location |
+-----+-----+-----+-----+-----+-----+
| Kapil     | NULL      | Sharma   | 28  | 10000  | Bangalore |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO employee (firstname,lastname, age, salary,loc
ation) VALUES ('Anuj','Sharma',28,10000,'Hyderabad');
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| firstname | middlename | lastname | age | salary | location |
+-----+-----+-----+-----+-----+-----+
| Kapil     | NULL      | Sharma   | 28  | 10000  | Bangalore |
| Anuj      | NULL      | Sharma   | 28  | 10000  | Hyderabad |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Combining DEFAULT and NOT NULL

Currently the **location** column allows Null values as well. So we can manually enter null value.

```
mysql> INSERT INTO employee (firstname,lastname, age, salary,location) VALUES ('Kapil','Sharma',28,10000,null);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from employee;
```

| firstname | middlename | lastname | age | salary | location  |
|-----------|------------|----------|-----|--------|-----------|
| Kapil     | NULL       | Sharma   | 28  | 10000  | Bangalore |
| Anuj      | NULL       | Sharma   | 28  | 10000  | Hyderabad |
| Kapil     | NULL       | Sharma   | 28  | 10000  | NULL      |

3 rows in set (0.00 sec)

To restrict Null values, we have to define in the table structure as NOT NULL

```
CREATE TABLE employee(
  firstname varchar(20) NOT NULL,
  middlename varchar(20),
  lastname varchar(20) NOT NULL,
  age int NOT NULL,
  salary int NOT NULL,
  location varchar(20) NOT NULL DEFAULT 'Bangalore'
);
```

```
mysql> CREATE TABLE employee(
-> firstname varchar(20) NOT NULL,
-> middlename varchar(20),
-> lastname varchar(20) NOT NULL,
-> age int NOT NULL,
-> salary int NOT NULL,
-> location varchar(20) NOT NULL DEFAULT 'Bangalore'
-> );
```

Query OK, 0 rows affected (0.04 sec)

```
mysql> desc employee;
```

| Field      | Type        | Null | Key | Default   | Extra |
|------------|-------------|------|-----|-----------|-------|
| firstname  | varchar(20) | NO   |     | NULL      |       |
| middlename | varchar(20) | YES  |     | NULL      |       |
| lastname   | varchar(20) | NO   |     | NULL      |       |
| age        | int         | NO   |     | NULL      |       |
| salary     | int         | NO   |     | NULL      |       |
| location   | varchar(20) | NO   |     | Bangalore |       |

6 rows in set (0.01 sec)

```
mysql> INSERT INTO employee (firstname,lastname, age, salary,location) VALUES ('Kapil','Sharma',28,10000,null);
ERROR 1048 (23000): Column 'location' cannot be null
```