SQL Video 3

⊙ Created by	
⊙ Created time	@January 12, 2023 11:26 AM
⊙ Last edited by	
⊙ Last edited time	@January 12, 2023 12:36 PM
⊙ Status	New
⊙ Tags	SQL
⊙ URL	

> *There can be 2 employees or names which are* the *same. So in that scenario how do we separate or uniquely identify them?*

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

Primary Key

It uniquely identifies each record in the table.

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

```
location varchar(20)
);
```

```
mysql> select * from employee;
+------+-----------+------------+----------+------+--------+----------+
| id   | firstname | middlename | lastname | age  | salary | location |
+------+-----------+------------+----------+------+--------+----------+
|    1 | Rajesh    | NULL       | Sharma   |   28 |  10000 | NULL     |
|    1 | Rajesh    | NULL       | Sharma   |   28 |  10000 | NULL     |
+------+-----------+------------+----------+------+--------+----------+
2 rows in set (0.00 sec)
```

So it allows us to enter same id values for different records. So by default the field is not set to Primary Key.

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

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| id         | int         | NO   | PRI | NULL    |       |
| firstname  | varchar(20) | YES  |     | NULL    |       |
| middlename | varchar(20) | YES  |     | NULL    |       |
| lastname   | varchar(20) | YES  |     | NULL    |       |
| age        | int         | YES  |     | NULL    |       |
| salary     | int         | YES  |     | NULL    |       |
| location   | varchar(20) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Now if we try to enter duplicate values

```
mysql> insert into employee (id,firstname,lastname,age, salary) VALUES (1
, 'Rajesh','Sharma',28,10000);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee (id,firstname,lastname,age, salary) VALUES (1
, 'Rajesh','Sharma',28,10000);
ERROR 1062 (23000): Duplicate entry '1' for key 'employee.PRIMARY'
```

If we try to enter NULL Values

```
mysql> insert into employee (id,firstname,lastname,age, salary) VALUES (N
ULL, 'Rajesh','Sharma',28,10000);
ERROR 1048 (23000): Column 'id' cannot be null
mysql>
```

So Primary Key doesn't allow

- NULL Values

- Repeated/ Duplicate Values

> 📌 SO WHAT IS A GOOD CANDIDATE FOR A PRIMARY KEY?

Something that can be used to uniquely identify records such as Customer ID, Employee ID, Registration Number etc.

The values that cannot be duplicated are Primary Keys

Another Way of defining a Primary Key

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

- Sometimes based on 2 columns we can have primary keys.

- This indicates that the combination of these two fields cannot be duplicate

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

AUTO INCREMENT

So when we are entering the record for employee, suppose we have 10 employees and the id of last employee will be 10. Now when we enter the 11th employee record, we need to remember the last entered value of id and it becomes difficult.

So we use **AUTO_INCREMENT** that helps incrementing the id value by 1 each time we enter the record. So do not need to manually mention id each time.

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

```
+------------+-------------+------+-----+-----------+----------------+
| Field      | Type        | Null | Key | Default   | Extra          |
+------------+-------------+------+-----+-----------+----------------+
| id         | int         | NO   | PRI | NULL      | auto_increment |
| firstname  | varchar(20) | YES  |     | NULL      |                |
| middlename | varchar(20) | YES  |     | NULL      |                |
| lastname   | varchar(20) | YES  |     | NULL      |                |
| age        | int         | YES  |     | NULL      |                |
| salary     | int         | YES  |     | NULL      |                |
| location   | varchar(20) | NO   |     | Bangalore |                |
+------------+-------------+------+-----+-----------+----------------+
7 rows in set (0.01 sec)
```

We don't mention the id values each time we enter the records but it is auto incremented

```
mysql> insert into employee (firstname,lastname,age, salary) VALUES('Raje
sh','Sharma',28,10000);
Query OK, 1 row affected (0.01 sec)

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

mysql> insert into employee (firstname,lastname,age, salary) VALUES('Kapi
l','Sharma',28,10000);
Query OK, 1 row affected (0.02 sec)

mysql> select * from employee;
+----+-----------+------------+----------+------+--------+-----------+
| id | firstname | middlename | lastname | age  | salary | location  |
+----+-----------+------------+----------+------+--------+-----------+
|  1 | Rajesh    | NULL       | Sharma   |   28 |  10000 | Bangalore |
|  2 | Kapil     | NULL       | Sharma   |   28 |  10000 | Bangalore |
+----+-----------+------------+----------+------+--------+-----------+
2 rows in set (0.00 sec)
```

UNIQUE KEY

- It ensures that the values in a particular Field are Unique.

- It makes sure the values are not duplicated

- Unique keys can be on more than one column

Primary Key v/s Unique Key

- We can have only 1 Primary Key

- Primary Key Cannot Hold Null Values

- Primary Key is used when we need to uniquely identify each record.

- Unique Keys can hold Null Values

- The Number of Null values a Unique Key can hold varies from database to database.

- In MySQL a unique key can hold any number of Null Values

- In some other famous DB's unique key can hold only one NULL

📌 We can have Multiple Unique Keys but only one Primary Key

```
mysql> insert into employee values('Kapil','Sharma',28);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values('Kapil','Sharma',28);
ERROR 1062 (23000): Duplicate entry 'Kapil-Sharma' for key 'employee.PRIM
ARY'
mysql> insert into employee values('Kapil','Sinha',28);
Query OK, 1 row affected (0.01 sec)
```

When we merge two columns as Primary Key, we concatenate and treat them together as a key. So the same combination cannot be entered.

```
mysql> CREATE TABLE employee(
    -> id INT UNIQUE KEY,
    -> firstname varchar(20),
    -> lastname varchar(20),
    -> age INT NOT NULL
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> desc employee;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int         | YES  | UNI | NULL    |       |
| firstname | varchar(20) | YES  |     | NULL    |       |
| lastname  | varchar(20) | YES  |     | NULL    |       |
| age       | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

Hence Unique Key Allows Null values

```
mysql> insert into employee values(null, 'Kapil','Sharma',20);
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+------+-----------+----------+-----+
| id   | firstname | lastname | age |
+------+-----------+----------+-----+
|    1 | Kapil     | Sharma   |  20 |
| NULL | Kapil     | Sharma   |  20 |
+------+-----------+----------+-----+
2 rows in set (0.00 sec)
```

Duplicate entries are restricted as id column has unique key constraint

```
mysql> insert into employee values(1, 'Kapil','Sharma',20);
Query OK, 1 row affected (0.01 sec)

mysql> insert into employee values(1, 'Kapil','Sharma',20);
ERROR 1062 (23000): Duplicate entry '1' for key 'employee.id'
```