CREATE TABLE actor (

actor\_id String(32767),

first\_name String(32767),

last\_name String(32767),

last\_update String(32767)

);

* actor is the table name
* actor\_id is a column name whose content is expected to be in text format (String) and not exceed 32767 characters.
* first\_name is a column name whose content is expected to be in text format (String) and not exceed 32767 characters.
* last\_name is a column name whose content is expected to be in text format (String) and not exceed 32767 characters.
* last\_update is a column name whose content is expected to be in text format (String) and not exceed 32767 characters.

Once I execute the query, the new table will be visible along with the other tables, so I can query from it anytime.

CREATE TABLE Trainer (

trainer\_id varchar(10),

first\_name varchar(20),

last\_name varchar(10),

last\_update varchar(10)

);

ALTER TABLE trainer ADD nationality varchar(20);

Insert into trainer values(“101”,”Preetti”,”singh”,”21 may”,”Indian”);

ALTER TABLE trainer RENAME COLUMN last\_name TO middle\_name;

ALTER TABLE trainer DROP COLUMN last\_update;

ALTER TABLE trainer ALTER COLUMN last\_update TIMESTAMP;

Conditions:

CREATE DATABASE company;

USE company;

CREATE TABLE employee (

Id INT PRIMARY KEY,

Name VARCHAR(45) NOT NULL,

Department VARCHAR(45) NOT NULL,

Salary FLOAT NOT NULL,

Gender VARCHAR(45) NOT NULL,

Age INT NOT NULL,

City VARCHAR(45) NOT NULL

);

Insert values to Employee table

**True Conditions in MySQL**

A typical WHERE condition looks like, **SELECT \* FROM employee WHERE id = 1006;** In this query, first the FROM clause will create an intermediate result set that contains all the data rows from the employee table. Then the system will filter the records using the WHERE clause with a condition id = 1006. It compares each row’s Id column value to the constant value 1006. System returns the data rows where the conditions are true. Everything happens all at once. The WHERE condition can be TRUE for one or more than one data row.

That is the WHERE condition provided in the SQL statement is true for this only one data row. Similarly, If we execute, **SELECT \* FROM employee WHERE age = 28;**

**SELECT \* FROM employee WHERE salary = 50000;**

**FALSE Conditions in MySQL**

As we already discussed, if a condition is not true it means the condition is FALSE. We can use false conditions to filter out the data that we don’t need. A simple example of a WHERE clause with a false condition is,

**SELECT \* FROM employee WHERE NOT id = 1006;**Here, you can see we get all the data rows except id=1006

The **NOT keyword** followed by the condition inverts the truthfulness of the condition. We can also write,

**SELECT \* FROM employee WHERE id <> 1006;** Here “less than greater than” is a special symbol that represents “**not equal to**”.

**SELECT \* FROM employee WHERE NOT department = ‘IT’;**

**SELECT \* FROM employee WHERE city <> ‘London’;**

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1001, 'John Doe', 'IT', 35000, 'Male', 25, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1002, 'Mary Smith', 'HR', 45000, 'Female', 27, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1003, 'James Brown', 'Finance', 50000, 'Male', 28, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1004, 'Mike Walker', 'Finance', 50000, 'Male', 28, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1005, 'Linda Jones', 'HR', 75000, 'Female', 26, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1006, 'Anurag Mohanty', 'IT', 35000, 'Male', 25, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1007, 'Priyanla Dewangan', 'HR', 45000, 'Female', 27, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1008, 'Sambit Mohanty', 'IT', 50000, 'Male', 28, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1009, 'Pranaya Kumar', 'IT', 50000, 'Male', 28, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1010, 'Hina Sharma', 'HR', 75000, 'Female', 26, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1011, 'Pramod Panda', 'IT', 45000, 'Male', 27, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1012, 'Preety Tiwary', 'HR', 55000, 'Female', 28, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1013, 'Santosh Dube', 'IT', 52000, 'Male', 28, 'Mumbai');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1014, 'Sara Talour', 'HR', 85000, 'Female', 26, 'London');

INSERT INTO employee (Id, `Name`, Department, Salary, Gender, Age, City) VALUES (1015, 'Pamela Kar', 'Finance', 70000, 'Female', 26, 'London');