

**Lab Report**

**on**

**Database Management System(SQL)**

**Submitted To:**

**Ms. Tania Khatun**

**Lecturer (Senior Scale), Department of CSE**

Daffodil International University

**Submitted By:**

**Md. Neamoth Ullah**

ID No: 162-15-8202

Sec: PC-B

BSC Program

Department of Computer Science & Engineering

Daffodil International University

**Insert Table**

The CREATE TABLE statement is used to create a new table in a database.

***Code:***

CREATE TABLE Customer (

EmployeeID int AUTO\_INCREMENT PRIMARY KEY,

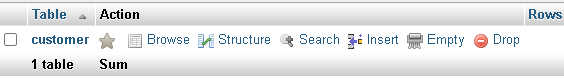
Name varchar(255),

Address varchar(255),

Salary INT(20)

);

***Output:***



**Insert Data**

The INSERT INTO statement is used to insert new records in a table

***Code:***

INSERT INTO customer (Name, Address, Salary)

VALUES

("Neamoth", "Savar", 2000),

("Keya", "Baipal", 1500),

("Nahid", "Narshingdi", 2000),

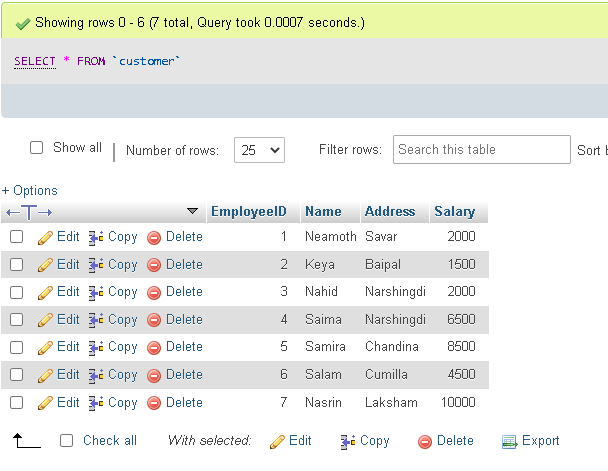
("Saima", "Narshingdi", 6500),

("Samira", "Chandina", 8500),

("Salam", "Cumilla", 4500),

("Nasrin", "Laksham", 10000);

***Output:***



**Find Data**

The SELECT statement is used to select data from a database. And where statement is used for condition. By using These statement we can pull out any data from Database.

**Question:** Find the Customer details where salary 6500

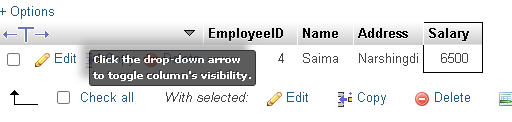
***Code:***

SELECT \*

FROM customer

Where Salary=6500;

***Output:***



**Aggregate Function**

There are five aggregate functions:

1. Max()
2. Min()
3. Count()
4. Sum()
5. Avg()

**Min Function:**

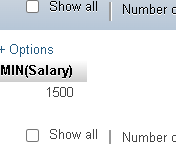
The MIN() function returns the smallest value of the selected column.

***Code:***

SELECT MIN(Salary)

FROM Customer;

***Output:***



**Max Function:**

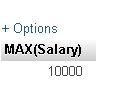
The MAX() function returns the highest value of the selected column.

***Code:***

SELECT MAX(Salary)

FROM employee;

***Output:***



**Count Function:**

The COUNT() function returns the number of rows that matches a specified criterion.

***Code:***

SELECT Count(EmployeeID)

FROM employee;

***Output:***



**Sum Function:**

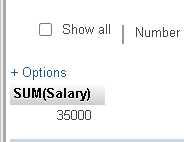
The SUM() function returns the total sum of a numeric column

***Code:***

SELECT SUM(Salary)

FROM employee;

***Output:***



**AVG Function:**

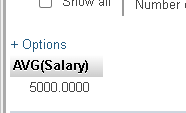
The AVG() function returns the average value of a numeric column.

***Code:***

SELECT AVG(Salary)

FROM employee;

***Output:***



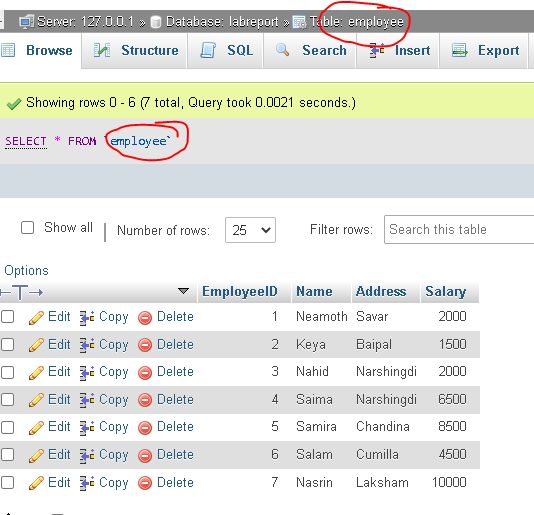
**Alter Table**

The Alter TABLE statement is used to rename an existing table.

***Code:***

ALTER TABLE customer RENAME Employee

***Output:***



**Update**

The UPDATE statement is used to modify the existing records in a table.

***Code:***

UPDATE employee

SET Address="Cumilla"

WHERE EmployeeID=4;

***Output:***



**Truncate**

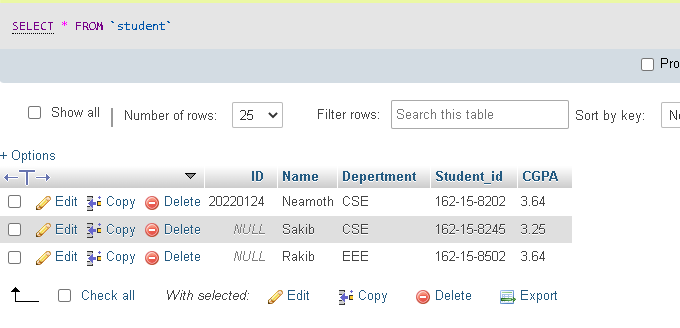
The TRUNCATE TABLE command deletes the data inside a table, but not the table itself .

***Code:***

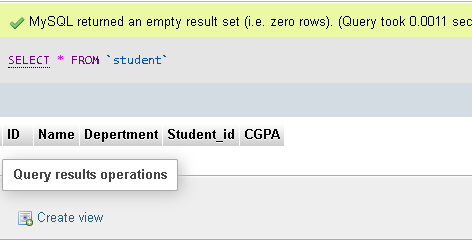
TRUNCATE TABLE Student;

***Output:***

**Before:**

****

**After:**

****

**Order By**

The ORDER BY keyword is used to sort the result-set in ascending or descending order. The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

***Code:***

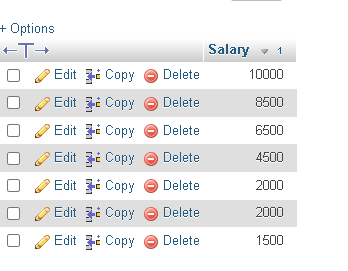
Only showing the DESC Order in this code:

SELECT Salary

FROM employee

ORDER BY Salary DESC;

***Output:***



**Between**

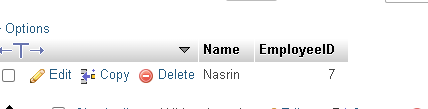
***Code:***

SELECT Name,EmployeeID

FROM employee

WHERE Salary BETWEEN 9000 AND 10000;

***Output:***



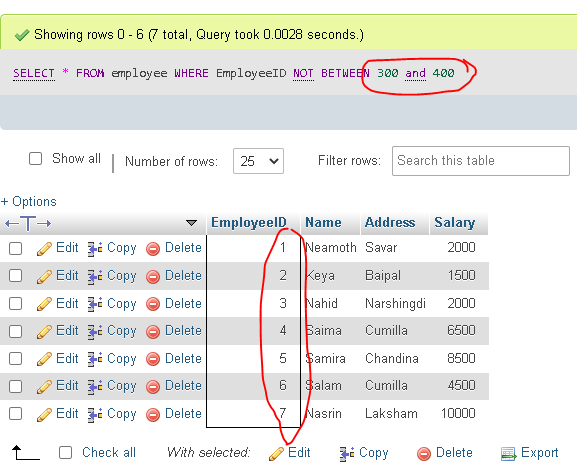
**Not Between**

***Code:***

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM employee

WHERE EmployeeID [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) BETWEEN 300 [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) 400

***Output:***



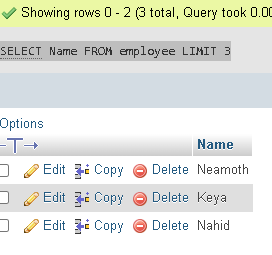
**Limit**

***Code:***

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Name FROM employee

LIMIT 3

***Output:***



**Group By**

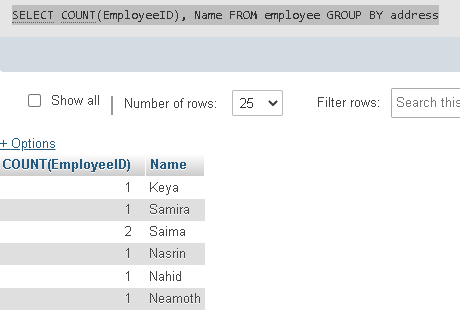
***Code:***

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(EmployeeID), Name

FROM employee

GROUP BY address

***Output:***



**Having**

***Code:***

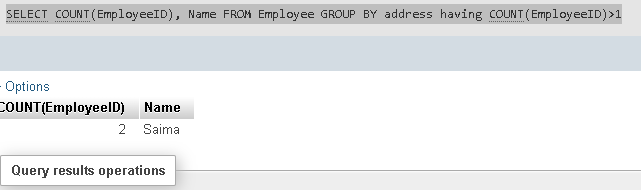
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(EmployeeID), Name

FROM Employee

GROUP BY address

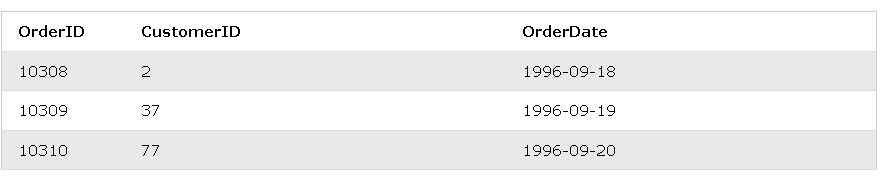
having [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(EmployeeID)>1

***Output:***

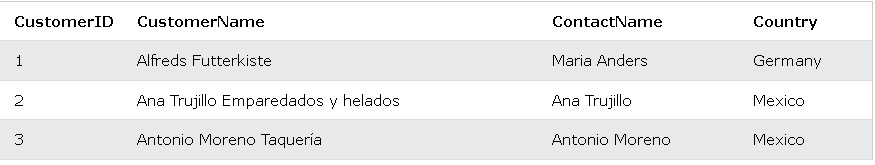


**Inner Join**

**Order Table:**

****

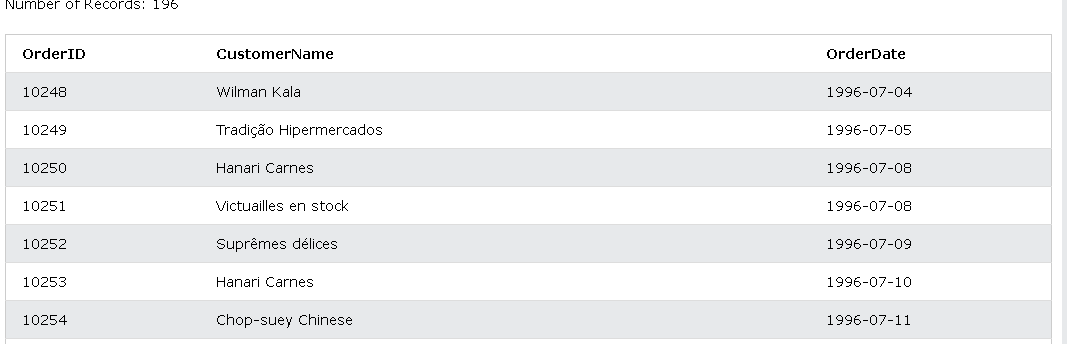
**Customer Table:**

****

***Code:***

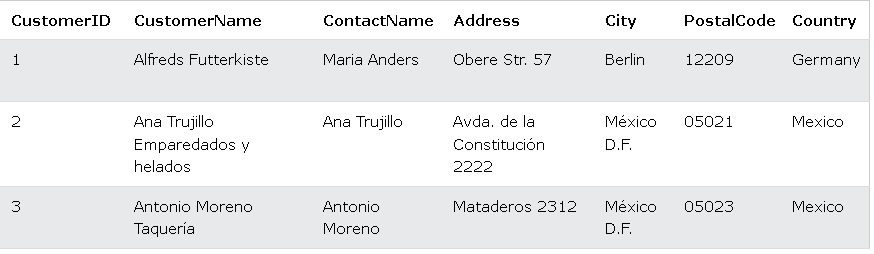
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

***Output:***

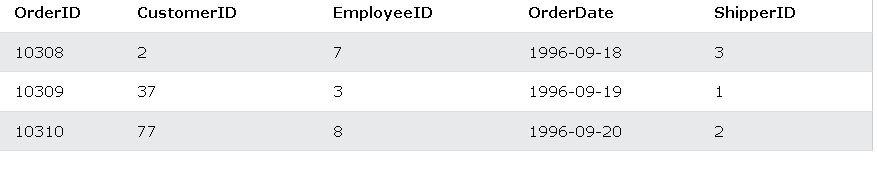


**Left Join**

**Customer Table:**

****

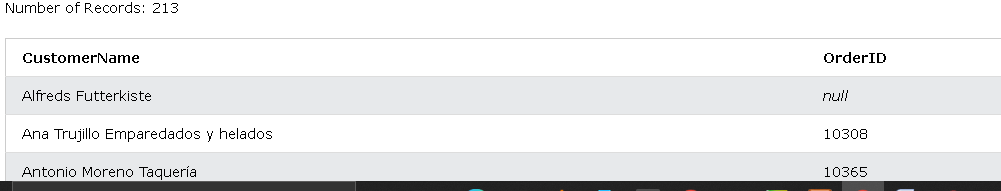
**Order Table:**

****

***Code:***

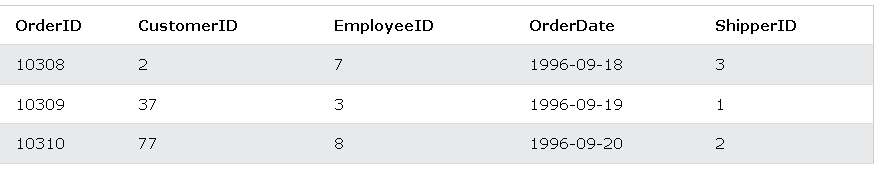
SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

***Output:***

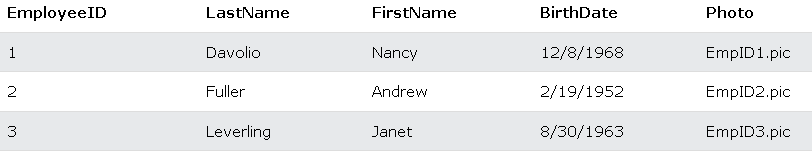


**Right Join**

**Order Table:**

****

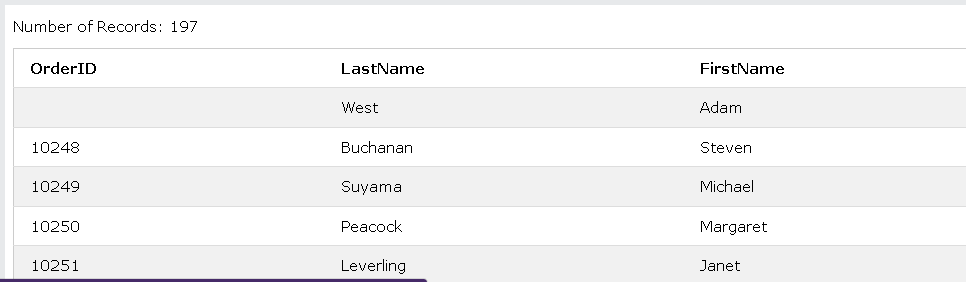
**Employee Table:**

****

***Code:***

SELECT Orders.OrderID, Employees.LastName, Employees.FirstName  
FROM Orders  
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID  
ORDER BY Orders.OrderID;

***Output:***



**Full Outer Join**

***Code:***

SELECT column\_name(s)  
FROM table1  
FULL OUTER JOIN table2ON table1.column\_name = table2.column\_nameWHERE condition;

***Output:***



**Self Join**

***Code:***

SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City  
ORDER BY A.City;

***Output:***

