**Experiment No: 2**

**Experiment Title: Usage of CREATE, INSERT, and all versions of SELECT commands in MYSQL.**

**Aim: Creation of Table, Insert, Usage of SELECT commands**

**(1) SQL Command: CREATE Database1;**

Command name: **CREATE TABLE**

Command Description: **The CREATE TABLE statement is used to create a new table in a database.**

Command Syntax:

**CREATE TABLE table\_name (**

**column1 datatype,**

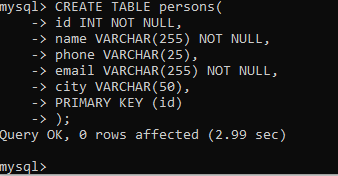
**column2 datatype,**

**column3 datatype,**

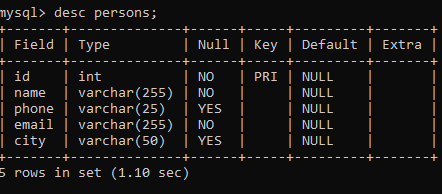
**....**

**);**

Query executed Screenshot:



Outcome: **This will create a table “person” with the above-mentioned column names with specific datatypes.**



**(2)SQL Command:**

**INSERT INTO persons(id,name,phone,email,city)**

**VALUES (1,'priyansh','9876543210', 'priyanshsen19@gmail.com','bhopal');**

Command name: **INSERT INTO**

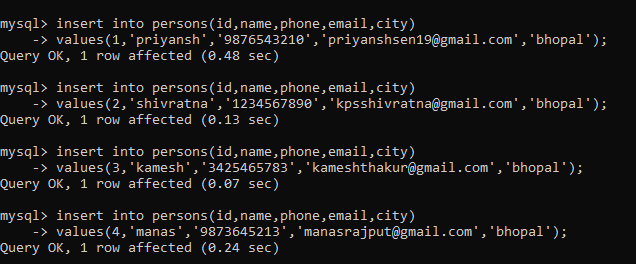
Command Description: **The INSERT INTO statement is used to insert new records in a table.**

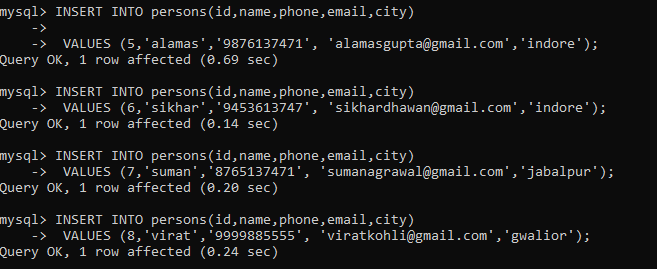
Command Syntax:

**INSERT INTO TABLE\_NAME (column1, column2, column3,...columnN)**

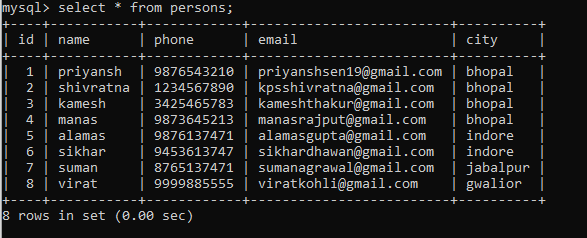
**VALUES (value1, value2, value3,...valueN);**

Query executed Screenshot:





Outcome: T**he query will insert the above details in the table persons one by one.**



**(3)SQL Command: SELECT name, phone from persons;**

Command name: **SELECT**

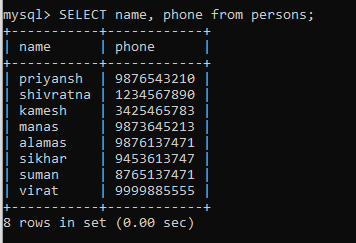
Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

Command Syntax:

**SELECT column1, column2....columnN**

**FROM table\_name;**

Query executed Screenshot:



Outcome: **As we can see from the above table the query will display the selected column names of your choice from the table persons.**

**(4)SQL Command: SELECT \* from persons;**

Command name: **SELECT**

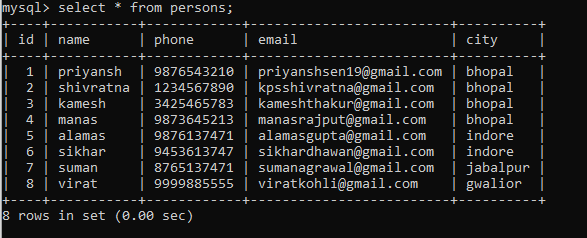
Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

Command Syntax:

**SELECT \***

**FROM table\_name;**

Query executed Screenshot:



Outcome: **As you can see the query will display all the data from the table persons.**

**(5)SQL Command: SELECT \* FROM persons WHERE city='bhopal';**

Command name: **SELECT**

Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

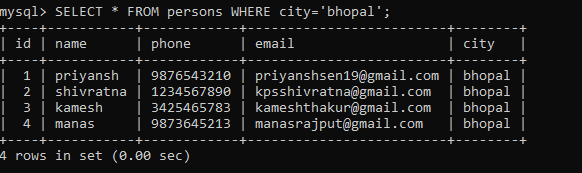
Command Syntax:

**SELECT column1, column2....columnN**

**FROM table\_name**

**WHERE CONDITION;**

Query executed Screenshot:



Outcome: **As you can see the query will display all the data from the table persons where city is Bhopal.**

**(6)SQL Command: SELECT id,name,phone FROM persons WHERE phone LIKE '9%' ORDER BY id DESC ;**

Command name: **SELECT**

Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

Command Syntax:

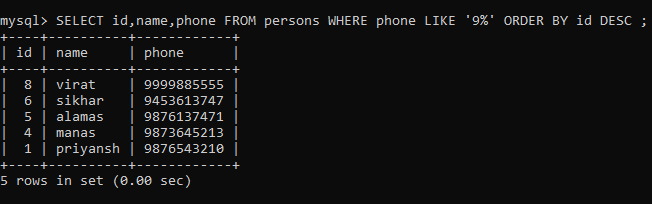
**SELECT column1, column2....columnN**

**FROM table\_name**

**WHERE CONDITION**

**ORDER BY column\_name {ASC|DESC};**

Query executed Screenshot:



Outcome: **As you can see the query will display all the data from the table persons where phone number starts with ‘9’ where the table is order by id in descending order.**

**(7)SQL Command: SELECT city, COUNT(\*) FROM persons WHERE city='bhopal' GROUP BY city;**

Command name: **SELECT**

Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

Command Syntax:

**SELECT column\_name**

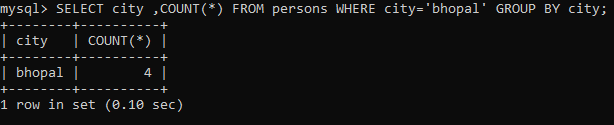
**FROM table\_name**

**WHERE CONDITION**

**GROUP BY column\_name;**

**ORDER BY column\_name;**

Query executed Screenshot:



Outcome: **As you can see the query will display the count data of city column from the table persons where id is city is bhopal where the table is group by city .**

**(8)SQL Command: SELECT COUNT(city) FROM persons WHERE city='bhopal' GROUP BY city having COUNT(\*)>3;**

Command name: **SELECT**

Command Description: **The SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.**

Command Syntax:

**SELECT SUM(column\_name)**

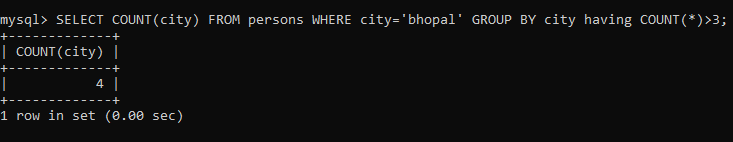
**FROM table\_name**

**WHERE CONDITION**

**GROUP BY column\_name**

**HAVING (arithmetic function condition)**

Query executed Screenshot:



Outcome: **As you can see the query will display the count data of city column from the table persons where id is city is bhopal where the table is group by city and having count of all data greater than 3.**

**(9)SQL Command: SELECT employees.\*,persons.\* FROM employees INNER JOIN persons ON employees.empid=persons.id;**

Command name: **INNER JOIN**

Command Description: **The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.**

Command Syntax:

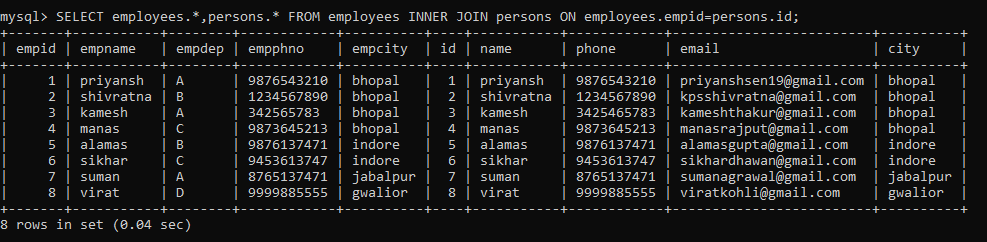
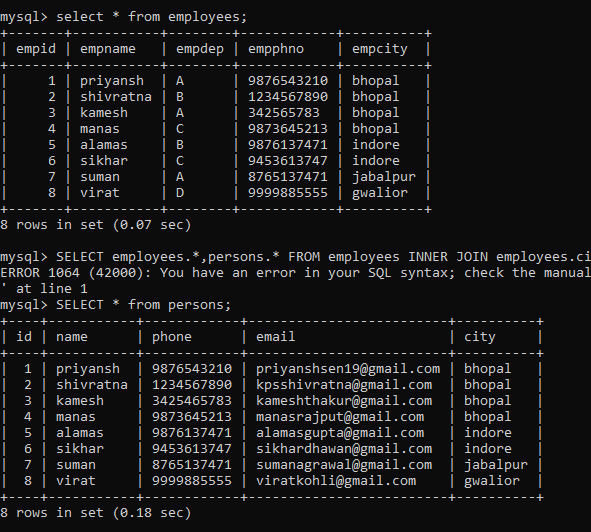
**SELECT table1.column1, table1.column2, table2.column1, ....**

**FROM table1**

**INNER JOIN table2**

**ON table1.matching\_column = table2.matching\_column;**

Query executed Screenshot:



Outcome: **As we can see the query will display all the data from persons and employees as the inner join of the employees table on persons where employees city and persons city is equal.**

**(10)SQL Command: SELECT persons.name,employees.empid FROM persons LEFT JOIN employees ON employees.empname=persons.name ORDER BY persons.name;**

Command name: **LEFT JOIN**

Command Description: **This join returns all the rows of the table on the left side of the join and matching rows for the table on the right side of the join. The rows for which there is no matching row on the right side, the result-set will contain NULL. LEFT JOIN is also known as LEFT OUTER JOIN.**

Command Syntax:

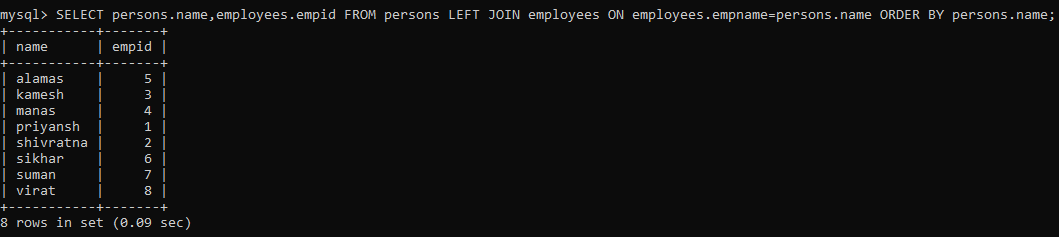
**SELECT table1.column1,table1.column2,table2.column1,....**

**FROM table1**

**LEFT JOIN table2**

**ON table1.matching\_column = table2.matching\_column;**

Query executed Screenshot:



Outcome:**As we can see the query will display person names and employees ids as the left join of the employees table on persons where employees city and persons city is equal.**

**(11)SQL Command: SELECT persons.name,employees.empphno FROM persons RIGHT JOIN employees ON employees.empphno=persons.phone ORDER BY persons.name;**

Command name: **RIGHT JOIN**

Command Description: **RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. The rows for which there is no matching row on the left side, the result-set will contain NULL. RIGHT JOIN is also known as RIGHT OUTER JOIN.**

Command Syntax:

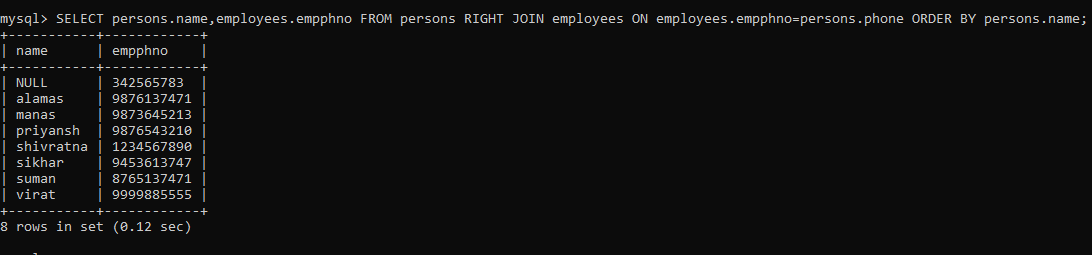
**SELECT table1.column1,table1.column2,table2.column1,....**

**FROM table1**

**RIGHT JOIN table2**

**ON table1.matching\_column = table2.matching\_column;**

Query executed Screenshot:



Outcome:**As we can see the query will display person names and employees phone numbers as the right join of the employees table on persons where employees phno and persons phone is equal.**

**(12)SQL Command: SELECT persons.name,employees.empphno FROM persons RIGHT JOIN employees ON employees.empphno=persons.phone ORDER BY persons.name;**

**SELECT city FROM persons UNION SELECT empcity FROM employees ORDER BY city;**

Command name: **UNION**

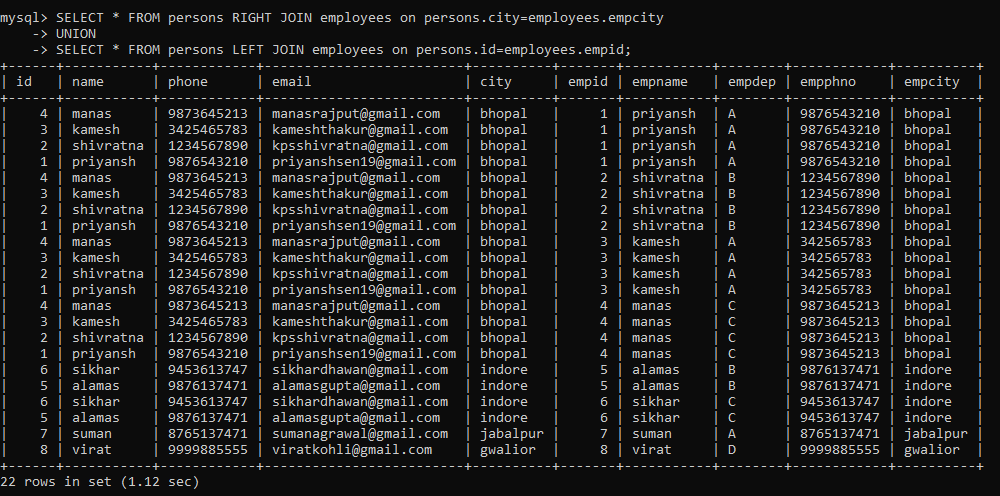
Command Description: **The UNION operator is used to combine the result-set of two or more SELECT statements.**

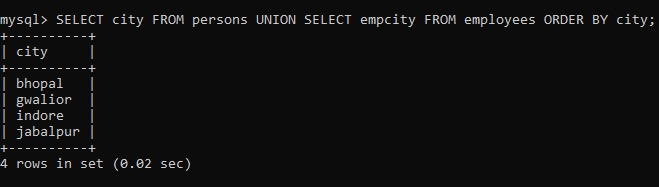
Command Syntax:

**SELECT COLUMN\_NAME(S) FROM TABLE1**

**UNION SELECT COLUMN\_NAME(S) FROM TABLE2;**

Query executed Screenshot:





Outcome:**As we can see the query will display person all data as the right join of the employees table on persons where employees city and persons city is equal union left join of the employees id and person id are equal .**

**(13)SQL Command: SELECT name,city,phone FROM persons WHERE phone IS NOT NULL;**

Command name: **IS/IS NOT NULL**

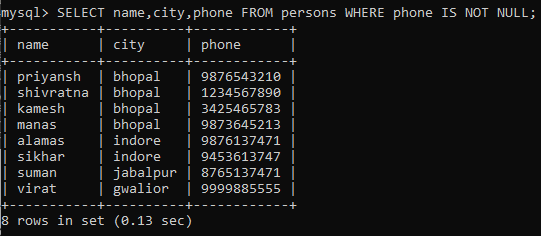
Command Description: **SQL allows queries that check whether an attribute value is NULL. Rather than using = or to compare an attribute value to NULL, SQL uses IS and IS NOT. This is because SQL considers each NULL value as being distinct from every other NULL value, so equality comparison is not appropriate..**

Command Syntax:

**SELECT COLUMN\_NAME(S) FROM TABLE1**

**UNION SELECT COLUMN\_NAME(S) FROM TABLE2;**

Query executed Screenshot:



Outcome:**As we can see the query will display person name,city and phone from persons where phone is not null which is no where hence we can see all the data .**

**(14)SQL Command: SELECT city,COUNT(city) FROM persons GROUP BY city;**

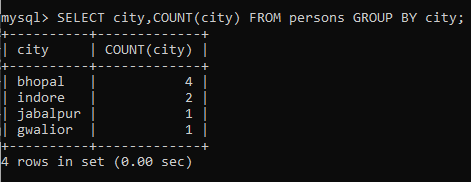
Command name: **COUNT()**

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

Command Syntax:

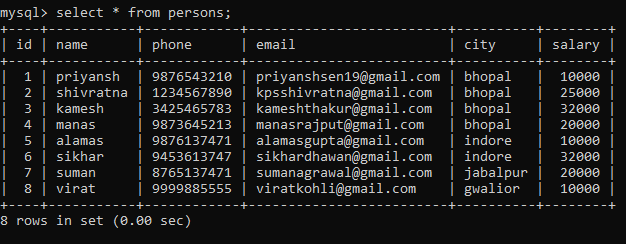
**SELECT COUNT(COLUMN\_NAME) FROM TABLE\_NAME WHERE CONDITION;**

Query executed Screenshot:



Outcome:**As we can see the query will display city and count of no. of cities grouped by city column from persons .**

**NOTE: Altering persons table for executing the rest of the commands**

****

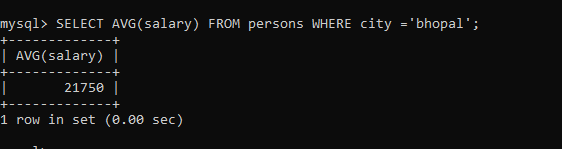
**(15)SQL Command: SELECT AVG(salary) FROM persons WHERE city ='bhopal';**

Command name: **AVG()**

Command Description: **The AVG() function returns the average value of a numeric column.**

Command Syntax: **SELECT AVG(COLUMN\_NAME) FROM TABLE\_NAME WHERE CONDITION;**

Query executed Screenshot:

****

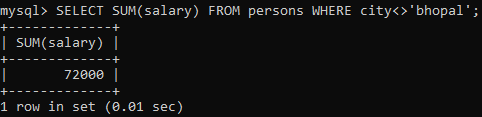
**(16)SQL Command: SELECT SUM(salary) FROM persons WHERE city<>'bhopal';**

Command name: **SUM()**

Command Description: **The SUM() function returns the total sum of a numeric column.**

Command Syntax: **SELECT SUM(COLUMN\_NAME) FROM TABLE\_NAME WHERE CONDITION;**

Query executed Screenshot:



**(17)SQL Command: SELECT MAX(salary) FROM persons;**

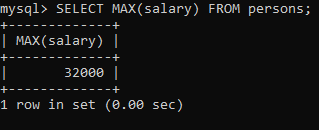
Command name: **MAX()**

Command Description: **The MAX() function returns the largest value of the selected column.**

Command Syntax: **SELECT MAX(COLUMN\_NAME)**

**FROM TABLE\_NAME WHERE CONDITION;**

Query executed Screenshot:



**(18)SQL Command: SELECT MIN(salary) FROM persons;**

Command name: **MIN()**

Command Description: **The MIn() function returns the smallest value of the selected column.**

Command Syntax: **SELECT MIN(COLUMN\_NAME)**

**FROM TABLE\_NAME WHERE CONDITION;**

Query executed Screenshot:

