# EXPERIMENT – 01

**Aim:** Introduction to Structure Query Language (SQL)

**Theory:** SQL is a standard language for accessing and manipulating. databases. It stands for structured Query Language SOL an lets you access and manipulate databases ANSI (American National Standard Institute) Standard.

**What can SOL do ?**

SQL can execute requires against a database. SQL can retrieve data from a database. SQL can insert records in a database. SQL can update, delete record from a database can create new databases. It can create a new tables in a database. It can create stored procedures in a databases. SQL can create views in database. It can set permission on tables, procedures and views.

Although, SQL is an ANSI standard, there are different versions of the SQL language. However, to be compliant with an ANSI standard they all support at least the major commands such as SELECT, UPDATE, INSERT, WHERE in the similar manner. Most of the SQL database program also have their own proprietary extensions in addition to SQL standard.

**What is RDBMS?**

A relational database is a type of database that stores and provides access to data points that are related to one another. Relational database are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

**What is Database Tables?**

A database most often contain one or more tables each table is identified by a name example “customer” or “employee” table contains records with data.

**What is SQL statement?**

Most of the actions you need to perform on the database done with SQL statement following SQL statement select all the records in the “customer” table.

```
Select * from customer;
```

SQL is not case a sensitive, Select is same as select.

**Semicolon after the SQL statements?**

Semicolon at the end of each SQL statement. Semicolon is a standard way to separate each SQL statement in a database system that allow more than one SQL statement to be

executed in the same all to server. But MS SQL server execute query with or without the semicolon.

Most Important SQL commands:

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

**Conclusion:** We have successfully understand the introduction to Structure Query Language(SQL).

# Experiment – 02

**Aim:** To study Basic SQL commands (create database, create table, use , drop, insert) and execute the following queries using these commands:

- Create a database named ‘ Employee’.
- Use the database ‘Employee’ and create a table ‘Emp’ with attributes ‘ename’,’ecity’,’salary’,’enumber’,’eaddress’,’depttname’.
- Create another table ‘Company’ with attributes ‘cname’, ccity’,’empnumber’ in the database ‘Employee’.

**Theory:**

1. **CREATE TABLE:** The create database statement is used to create a new SQL database.
   **SYNTAX:** create table tablename(colname1 datatype, colname2 datatype)
   **EXAMPLE**: create table student (sname varchar(20), rollno. int, marks int)

2. **CREATE DATABASE:** This command is used to create a new database.
   **SYNTAX**: create database databaseName
   **EXAMPLE**: create database college.

3. **USE:** This command is used to work on a specified database
   **SYNTAX**: use database
   **EXAMPLE**: use college.

4. **INSERT INTO:** This command is used to insert data in the form of records or rows into a specified table.
   **SYNTAX**: insert into tableName_value(‘expression1’, expression2’)
   **EXAMPLE**: insert into student value (‘German’, 01)

5. **DROP TABLE:** This command is used to remove the table from the database completely and delete all its data.
   **SYNTAX**: drop table tableName
   **EXAMPLE**: drop table student
6. **DROP DATABASE:** This command is used to remove the whole database completely and delete all its data.
   **SYNTAX**: drop database databaseName
   **EXAMPLE**: database college.

**NOTE:** You can not drop a database that is currently in use.

**OUTPUT:**

```
mysql> create database Employee_AAAI;
Query OK, 1 row affected (0.10 sec)
```

```
mysql> create database Employee_AAAI;
Query OK, 1 row affected (0.10 sec)

mysql> use Employee_AAAI;
Database changed
mysql> create table Emp (ename varchar(40), ecity varchar(40),salary float, enumber int, eaddress varchar(80),deptname varchar(30));
Query OK, 0 rows affected (1.47 sec)

mysql> insert into Emp values( 'Anubha','Delhi', 90000.0 , 10, 'Karol Bagh','IT');
Query OK, 1 row affected (0.17 sec)

mysql> insert into Emp values( 'Anmol' , 'Delhi', 79000.0, 11 , 'Janakpuri', 'HR');
Query OK, 1 row affected (0.11 sec)

mysql> insert into Emp values( 'Inderpreet' , 'Banglore', 95000.0, 12 , 'Subhash Nagar', 'IT');
Query OK, 1 row affected (0.12 sec)

mysql> insert into Emp values( 'Arpit' , 'Chandigarh', 86000.0, 13 , 'Uttam Nagar', 'HR');
Query OK, 1 row affected (0.11 sec)

mysql> insert into Emp values( 'Siddharth' , 'Delhi', 56000.0, 15 , 'Patel Nagar', 'Sales');
Query OK, 1 row affected (0.79 sec)

mysql> Select * from Emp;
+-----------+-----------+--------+---------+---------------+----------+
| ename     | ecity     | salary | enumber | eaddress      | deptname |
+-----------+-----------+--------+---------+---------------+----------+
| Anubha    | Delhi     |  90000 |      10 | Karol Bagh    | IT       |
| Anmol     | Delhi     |  79000 |      11 | Janakpuri     | HR       |
| Inderpreet | Banglore  |  95000 |      12 | Subhash Nagar | IT       |
| Arpit     | Chandigarh |  86000 |      13 | Uttam Nagar   | HR       |
| Siddharth | Delhi     |  56000 |      15 | Patel Nagar   | Sales    |
+-----------+-----------+--------+---------+---------------+----------+
5 rows in set (0.04 sec)
```

```
mysql> create table company( cname varchar(20),ccity varchar(30), empnumber int, numofemp int);
Query OK, 0 rows affected (1.18 sec)

mysql> insert into company values('TCS' , 'Banglore' , 11, 200);
Query OK, 1 row affected (12.34 sec)

mysql> insert into company values('HCL','Delhi' , 15, 170);
Query OK, 1 row affected (1.46 sec)

mysql> insert into company values('TCS','Delhi' , 12, 130);
Query OK, 1 row affected (26.50 sec)

mysql>  insert into company values('OLA','Chennai' , 10, 250);
Query OK, 1 row affected (0.08 sec)

mysql>  insert into company values('Amazon','Mumbai' , 13, 220);
Query OK, 1 row affected (0.14 sec)

mysql> select * from company;
+--------+----------+-----------+----------+
| cname  | ccity    | empnumber | numofemp |
+--------+----------+-----------+----------+
| TCS    | Banglore |        11 |      200 |
| HCL    | Delhi    |        15 |      170 |
| TCS    | Delhi    |        12 |      130 |
| OLA    | Chennai  |        10 |      250 |
| Amazon | Mumbai   |        13 |      220 |
+--------+----------+-----------+----------+
5 rows in set (0.00 sec)
```

# Experiment – 03

**Aim:** To study the viewing commands (select , update) and execute the following queries using these commands:

- Find the names of all employees who live in Delhi.
- Increase the salary of all employees by Rs. 5,000.
- Find the company names where the number of employees is greater than 10,000.
- Change the Company City to Gurgaon where the Company name is ‘TCS’.

**Theory:**

1. SELECT: This command is view some or all the specific details from a table specified.
   SYNTAX: SELECT * from tableName;
   Example: select * form student

2. UPDATE: This command is used to update the specific contents of the table. It is used to change and alter the table values.
   SYNTAX: UPDATE tableName set conName1 = expression1 where condition.
   Example: UPDATE student set marks = marks + 2 where rollNo = 5.

**QUERIES:**

1. Select ename from emp where ecity = ‘Delhi’;

2. Update emp set salary = salary + 5000;

3. Select ename from company where num of emp > 10000;

4. Update company set ecity = ‘Gurgaon’ where ecity = ‘Delhi’;

**OUTPUT:**

```
mysql> use Employee_AAAI;
Database changed
mysql> select ename from Emp
    -> where ecity = 'Delhi';
+-----------+
| ename     |
+-----------+
| Anubha    |
| Anmol     |
| Siddharth |
+-----------+
3 rows in set (1.40 sec)
```

```
mysql> update emp
    -> set salary = salary + 5000;
Query OK, 5 rows affected (1.45 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from emp;
+------------+------------+--------+---------+---------------+----------+
| ename      | ecity      | salary | enumber | eaddress      | deptname |
+------------+------------+--------+---------+---------------+----------+
| Anubha     | Delhi      | 100000 |      10 | Karol Bagh    | IT       |
| Anmol      | Delhi      |  89000 |      11 | Janakpuri     | HR       |
| Inderpreet | Banglore   | 105000 |      12 | Subhash Nagar | IT       |
| Arpit      | Chandigarh |  96000 |      13 | Uttam Nagar   | HR       |
| Siddharth  | Delhi      |  66000 |      15 | Patel Nagar   | Sales    |
+------------+------------+--------+---------+---------------+----------+
5 rows in set (0.00 sec)
```

```
mysql> select cname from company
    -> where numofemp > 10000;
+--------+
| cname  |
+--------+
| OLA    |
| Amazon |
+--------+
2 rows in set (0.07 sec)
```

# Experiment – 04

**Aim:** To study the commands to modify the structure of table (alter, delete) and execute the following queries using these commands:

- Add an attribute named ‘ Designation’ to the table ‘Emp’.
- Modify the table ‘Emp’, Change the datatype of ‘salary’ attribute to float.
- Drop the attribute ‘depttname’ from the table ‘emp’.
- Delete the entries from the table ‘ Company’ where the number of employees are less than 500.

**Theory:**

1. **ALTER**: This command is used to alter the contents of the table.
   SYNTAX: Alter table tableName add new colname datatype;
   Example: alter table student add age int;

2. **DELETE**: This command is used to delete the database from the given table.
   SYNTAX: Delete from tableName where condition
   Example: delete from student where rollNo = 1

**QUERIES:**

1. Alter table emp add designation varchar (30);

2. Alter table emp modify salary float;

3. Alter table emp drop column depttname’;

4. delete from company where num of emp < 500;

**OUTPUT:**

```
mysql> alter table emp
    -> add designation varchar(30);
Query OK, 0 rows affected (4.17 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from emp;
+------------+------------+--------+---------+---------------+----------+-------------+
| ename      | ecity      | salary | enumber | eaddress      | deptname | designation |
+------------+------------+--------+---------+---------------+----------+-------------+
| Anubha     | Delhi      | 100000 |      10 | Karol Bagh    | IT       | NULL        |
| Anmol      | Delhi      |  89000 |      11 | Janakpuri     | HR       | NULL        |
| Inderpreet | Banglore   | 105000 |      12 | Subhash Nagar | IT       | NULL        |
| Arpit      | Chandigarh |  96000 |      13 | Uttam Nagar   | HR       | NULL        |
| Siddharth  | Delhi      |  66000 |      15 | Patel Nagar   | Sales    | NULL        |
+------------+------------+--------+---------+---------------+----------+-------------+
5 rows in set (3.40 sec)

mysql> update emp
    -> set Designation = 'Project Head'
    -> where ename ='Anubha';
Query OK, 1 row affected (1.79 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update emp
    -> set Designation = 'Manager'
    -> where ename ='Arpit';
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update emp
    -> set Designation = 'IT Head'
    -> where ename = 'Inderpreet' ;
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp
    -> set Designation = 'Sr. HR Executive'
    -> where ename = 'Anmol';
Query OK, 1 row affected (0.70 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update emp
    -> set Designation = 'Programmer'
    -> where ename = 'Siddharth';
Query OK, 1 row affected (4.48 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from emp;
+------------+------------+--------+---------+---------------+----------+------------------+
| ename      | ecity      | salary | enumber | eaddress      | deptname | designation      |
+------------+------------+--------+---------+---------------+----------+------------------+
| Anubha     | Delhi      | 100000 |      10 | Karol Bagh    | IT       | Project Head     |
| Anmol      | Delhi      |  89000 |      11 | Janakpuri     | HR       | Sr. HR Executive |
| Inderpreet | Banglore   | 105000 |      12 | Subhash Nagar | IT       | IT Head          |
| Arpit      | Chandigarh |  96000 |      13 | Uttam Nagar   | HR       | Manager          |
| Siddharth  | Delhi      |  66000 |      15 | Patel Nagar   | Sales    | Programmer       |
+------------+------------+--------+---------+---------------+----------+------------------+
5 rows in set (0.00 sec)
```

```
mysql> alter table emp
    -> modify salary float;
Query OK, 0 rows affected (0.73 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> describe emp;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ename       | varchar(40) | YES  |     | NULL    |       |
| ecity       | varchar(40) | YES  |     | NULL    |       |
| salary      | float       | YES  |     | NULL    |       |
| enumber     | int         | YES  |     | NULL    |       |
| eaddress    | varchar(80) | YES  |     | NULL    |       |
| deptname    | varchar(30) | YES  |     | NULL    |       |
| designation | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
7 rows in set (3.19 sec)
```

```
mysql> alter table emp
    -> drop column deptname;
Query OK, 0 rows affected (7.81 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from emp;
+------------+------------+--------+---------+---------------+------------------+
| ename      | ecity      | salary | enumber | eaddress      | designation      |
+------------+------------+--------+---------+---------------+------------------+
| Anubha     | Delhi      | 100000 |      10 | Karol Bagh    | Project Head     |
| Anmol      | Delhi      |  89000 |      11 | Janakpuri     | Sr. HR Executive |
| Inderpreet | Banglore   | 105000 |      12 | Subhash Nagar | IT Head          |
| Arpit      | Chandigarh |  96000 |      13 | Uttam Nagar   | Manager          |
| Siddharth  | Delhi      |  66000 |      15 | Patel Nagar   | Programmer       |
+------------+------------+--------+---------+---------------+------------------+
5 rows in set (0.00 sec)
```

```
mysql> select * from company;
+--------+----------+-----------+----------+
| cname  | ccity    | empnumber | numofemp |
+--------+----------+-----------+----------+
| TCS    | Banglore |        11 |    10000 |
| HCL    | Gurgaon  |        15 |      250 |
| TCS    | Gurgaon  |        12 |     9930 |
| OLA    | Chennai  |        10 |    10050 |
| Amazon | Mumbai   |        13 |    10020 |
+--------+----------+-----------+----------+
5 rows in set (0.00 sec)

mysql> delete from company
    -> where numofemp < 500;
Query OK, 1 row affected (2.11 sec)

mysql> select * from company;
+--------+----------+-----------+----------+
| cname  | ccity    | empnumber | numofemp |
+--------+----------+-----------+----------+
| TCS    | Banglore |        11 |    10000 |
| TCS    | Gurgaon  |        12 |     9930 |
| OLA    | Chennai  |        10 |    10050 |
| Amazon | Mumbai   |        13 |    10020 |
+--------+----------+-----------+----------+
4 rows in set (0.00 sec)
```

# Experiment – 05

**Aim:** To study the commands that involve compound conditions (and, or, in , not in, between , not between , like , not like) and execute the following queries using these commands:

- Find the names of all employees who live in ' Gurgaon' and whose salary is between Rs. 20,000 and Rs. 30,000.
- Find the names of all employees whose names begin with either letter 'A' or 'B'.
- Find the company names where the company city is 'Delhi' and the number or employees is not between 5000 and 10,000.
- Find the names of all companies that do not end with letter 'A'.

**Commands:**

1. **AND / OR:** The AND and OR operators are used to filter records based on more than One are Condition
    - The AND operator displays a record if all the conditions is separated by AND are TRUE
      **SYNTAX**: select * from table_name where <condition!> AND <condition 2> ;
      **EXAMPLE**: select * from student where sname = 'XYZ' and roll = 10;

    - The OR operator displays a record if any of the conditions is separated by OR is TRUE. **SYNTAX:**- Select * from table_name where <condition1> OR <condition 2>;
      **EXAMPLE:** select * where from student where sname = "ABC" OR roll = 1;

2. **IN/NOT IN:** The IN operator allows to specify multiple in a WHERE clause. It is a shorthand for multiple OR conditions. NOT IN is its complement.
   **SYNTAX:** select * from table_name where colname IN/NOT IN ('value1' , 'value2');
   **EXAMPLE:** select * where from student where sname in ("ABC", "XYZ"), select * from student where role not in (1,3);

3. **BETWEEN NOT BETWEEN:** The BETWEEN operator selects the values within a given range. The values can be numbers, text or dates. The BETWEEN is inclusive: begin and end value is included. The NOT BETWEEN operator select values outside the range.
   **SYNTAX:** select * from table_name where colname BETWEEN/ NOT BETWEEN <lower limit > and < upper limit>;
   **EXAMPLE:** select * from student where roll between 11 and 20;

4. **LIKE/ NOT LIKE:** The Like operator is used in a where clause to search for a specified pattern in a column. To get the record that does not match the like pattern we use NOT LIKE.
   There are two wild cards often used in conjunction with the like operator:
   1. The percentage represent zero, one or multiple character
   2. The underscore represents one and single character.
   3. values outside the range.
      **SYNTAX:** select * from table_name where colname LIKE NOT LIKE <pattern>
      **EXAMPLE:** select * from student where sname LIKE 'A%'; (Names starting with A)

**QUERIES:**

1. Select ename from emp where is ecity to “Gurgaon” and salary between 20000 and 30000;
2. Select ename from emp where ename like ‘A%’ or ename like ‘B%’.
3. Select ename form company where city = “Delhi” and numOfEmp not between 5000 and 10000;
4. Select ename from company where ename not like “%A”;

**OUTPUTS:**

```
mysql> select * from emp;
+------------+------------+--------+---------+---------------+------------------+
| ename      | ecity      | salary | enumber | eaddress      | designation      |
+------------+------------+--------+---------+---------------+------------------+
| Anubha     | Gurgaon    |  40000 |      10 | Karol Bagh    | Project Head     |
| Anmol      | Gurgaon    |  29000 |      11 | Janakpuri     | Sr. HR Executive |
| Inderpreet | Banglore   |  45000 |      12 | Subhash Nagar | IT Head          |
| Arpit      | Chandigarh |  36000 |      13 | Uttam Nagar   | Manager          |
| Siddharth  | Gurgaon    |  22000 |      15 | Patel Nagar   | Programmer       |
+------------+------------+--------+---------+---------------+------------------+
5 rows in set (0.00 sec)

mysql> select ename from emp
    -> where ecity = 'Gurgaon' and salary between 20000 and 30000;
+-----------+
| ename     |
+-----------+
| Anmol     |
| Siddharth |
+-----------+
2 rows in set (0.00 sec)
```

```
mysql> select ename from emp
    -> where ename like 'A%' or ename like 'B%';
+--------+
| ename  |
+--------+
| Anubha |
| Anmol  |
| Arpit  |
+--------+
3 rows in set (0.00 sec)
```

```
mysql> select * from company;
+---------+----------+-----------+-----------+
| cname   | ccity    | empnumber | numofemp  |
+---------+----------+-----------+-----------+
| TCS     | Banglore |        11 |     10500 |
| TCS     | Delhi    |        12 |     15000 |
| OLA     | Chennai  |        10 |     10050 |
| Amazon  | Mumbai   |        13 |     10020 |
+---------+----------+-----------+-----------+
4 rows in set (0.32 sec)

mysql> select cname from company
    -> where ccity = 'Delhi' and numofemp not between 5000 and 10000;
+--------+
| cname  |
+--------+
| TCS    |
+--------+
1 row in set (0.00 sec)
```

```
mysql> select cname from company
    -> where cname not like '%A';
+--------+
| cname  |
+--------+
| TCS    |
| TCS    |
| Amazon |
+--------+
3 rows in set (0.00 sec)
```

# Experiment - 06

**Aim:** To study the aggregate functions (sum, count, max, min, average) and execute the following queries using these commands:

- Find the sum and average of salaries of all employees in computer science department.
- Find the number of all employees who live in Delhi.
- Find the maximum and the minimum salary in the HR department.

**Commands:**

1. **SUM() :** The sum() function returns the total sum of a numeric column.
   **SYNTAX:** select sum (marks) from student where roll between 1 and 10;
   **EXAMPLE:** select sum (marks) from student where roll between 1 and 10;

2. **COUNT():** The count() function returns the number of rows that matches a specified criteria.
   **SYNTAX:** select count (col_name) from table_name where <condition>
   **EXAMPLE:** select count (roll) from student;

3. **MAX():** The max() function returns the largest value of the selected column
   **SYNTAX:** select max (col_name) from table_name where <condition>;
   **EXAMPLE:** select max (marks) from student where roll between 1 and 10;

4. **MIN():** The min() function returns the smallest value of the selected column
   **SYNTAX:** select min (col_name) from table_name where <condition>;
   **EXAMPLE:** select min (marks) from student

5. **AVERAGE():** the AVG() function returns the average value of the numeric column
   SYNTAX: select avg (col_name) from table_name where <condition>
   EXAMPLE: select avg (marks) form student where roll between 1 and 20;

**QUERIES:**

1. Select sum (salary), avg (salary) from emp where designation = ‘Computer Science’ ;

2. Select count (enumber) from emp where ecity = ‘Delhi’ ;

3. Select max (salary), min(salary) from emp where designation = ‘HR’ ;

**OUTPUTS:**

```
mysql> select * from emp;
+------------+------------+--------+---------+--------------+------------------+
| ename      | ecity      | salary | enumber | eaddress     | designation      |
+------------+------------+--------+---------+--------------+------------------+
| Anubha     | Gurgaon    |  40000 |      10 | Karol Bagh   | Project Head     |
| Anmol      | Gurgaon    |  29000 |      11 | Janakpuri    | Sr. HR Executive |
| Inderpreet | Banglore   |  45000 |      12 | Subhash Nagar | IT Head          |
| Arpit      | Chandigarh |  36000 |      13 | Uttam Nagar  | Manager          |
| Siddharth  | Gurgaon    |  22000 |      15 | Patel Nagar  | Programmer       |
| Rishabh    | Delhi      |  65000 |      16 | Shadipur     | Computer Science |
| Pankaj     | Delhi      |  55000 |      17 | Karol Bagh   | HR               |
| Tushar     | Banglore   |  62000 |      18 | Patel Nagar  | Computer Science |
+------------+------------+--------+---------+--------------+------------------+
8 rows in set (0.00 sec)

mysql> select sum(salary) , avg(salary) from emp
    -> where designation = 'Computer Science';
+-------------+-------------+
| sum(salary) | avg(salary) |
+-------------+-------------+
|      127000 |       63500 |
+-------------+-------------+
1 row in set (0.28 sec)
```

```
mysql> select count(enumber) from emp
    -> where ecity = 'Delhi';
+----------------+
| count(enumber) |
+----------------+
|              3 |
+----------------+
1 row in set (0.04 sec)
```

```
mysql> select max(salary), min(salary) from emp
    -> where designation = 'HR';
+-------------+-------------+
| max(salary) | min(salary) |
+-------------+-------------+
|       75664 |       55000 |
+-------------+-------------+
1 row in set (4.57 sec)
```

# Experiment – 07

**Aim:** To study the grouping commands (group by, order by) and execute the following queries using these commands:

- List all employee names in descending order.
- Find number of employees in each department where number of employees is greater than 5.
- List all the department names where average salary of a department is Rs.10,000.

## Commands:

1. **GROUP BY():** The group by statement groups rows that have the same value into summary rows. The group by statement is often used with aggregate functions like count, maximum, minimum, etc to group the result set by one or more columns. The having clause is used because the where key word can not be used with the aggregate functions.
   **SYNTAX:** select col1, col2 from table_name where <condition> -> before grouping group by call name having <condition> -> after grouping
   **EXAMPLE:** select marks, count (roll) from student group by marks;
   select marks, count(roll) from student where marks = 90 group by marks;

2. **ORDER BY():** This keyword is used to sort the result set in ascending or descending order the order. The order by keyword sorts the result in ascending order by default and to sort the records in descending order DESC keyword is used
   **SYNTAX:** select col1, col2 from table_name order by calname ASC/ DESC ;
   **EXAMPLE:** select sname from student order by roll DESC;

## QUERIES:

1. Select ename from emp order by ename_desc;

2. Select designation, count (enumber) from emp group by designation having count (enumber) > 3;

3. Select designation from emp group by designation having avg (salary) = 69000;

**OUTPUTS:**

```
mysql> select ename from emp
    -> order by ename desc;
+------------+
| ename      |
+------------+
| Tushar     |
| Siddharth  |
| Rishabh    |
| Rajesh     |
| Pankaj     |
| Nitin      |
| Kris       |
| Inderpreet |
| Geeta      |
| Arpit      |
| Anubha     |
| Anmol      |
+------------+
12 rows in set (0.03 sec)
```

```
mysql> select designation, count(enumber) from emp
    -> group by designation
    -> having count(enumber) > 2;
+-----------------+---------------+
| designation     | count(enumber) |
+-----------------+---------------+
| Computer Science |             3 |
| HR              |             3 |
+-----------------+---------------+
2 rows in set (0.08 sec)
```

```
mysql> select designation from emp
    -> group by designation
    -> having avg(salary) = 69000;
+------------------+
| designation      |
+------------------+
| Computer Science |
+------------------+
1 row in set (0.00 sec)
```

# Experiment – 08

**Aim:** To study the commands involving data constraints and execute the following queries using these commands:

- Alter table 'Emp' and make 'enumber' as the primary key.
- Alter table 'Company' and add the foreign key constraint.
- Add a check constraint in the table 'Emp' such that salary has the value between 0 and Rs.1,00,000.
- Alter table 'Company' and add unique constraint to column cname.
- Add a default constraint to column ccity of table company with the value 'Delhi'.

**Commands:**

1. **PRIMARY KEY CONSTRAINT:** The PRIMARY KEY CONSTRAINT uniquely identify each record in a table primary keys must contain unique values and cannot contain NULL values.

   **SYNTAX:** after table table_name add constraints consname primary key (colname);
   **EXAMPLE:** alter table student add constraint c1 primary key (roll);

2. **FOREIGN KEY CONSTRAINT:** The FOREIGN KEY CONSTRAINT specifies that the key can only contain the values that are in the referred primary key.

   **SYNTAX:** alter table table_name1 add constraints consname foreign key (colname) references table_name (primary key of table2) ;
   **EXAMPLE:** alter table orders add constraints c2 foreign by (personID) references persons (personID) ;

3. **UNIQUE KEY CONSTRAINT:** The UNIQUE KEY CONSTRAINT prevents two records from having identical values in a column.

   **SYNTAX:** alter table table_name add constraints constraints_name unique (colname) ;
   **EXAMPLE:** alter table student add constraints c3 uniques (adm no.)

4. **DEFAULT CONSTRAINT:** DEFAULT CONSTRAINT is used to fill column with a default and the fixed value.

   **SYNTAX:** alter table table_name alter column set default value ;

   **EXAMPLE:** alter table student alter table age set default 20;

**NOTE:** To view constraints we use " show to create table table_name; "

**QUERIES:**

1. Alter table emp modify enumber int not NULL ;
   Alter table emp add constraints G1 primary key (enumber);

2. Alter table company add constraints G2 foreign key (empNumber) references emp (enumber)

3. Alter table company add constraints G4 unique (ename);

4. Alter table company alter company ecity et default ‘Delhi’ ;

**OUTPUTS:**

```
mysql> alter table Emp
    -> add constraint g1 primary key(enumber);
Query OK, 0 rows affected (1.80 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc emp;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| ename       | varchar(40) | YES  |     | NULL    |       |
| ecity       | varchar(40) | YES  |     | NULL    |       |
| salary      | float       | YES  |     | NULL    |       |
| enumber     | int         | NO   | PRI | NULL    |       |
| eaddress    | varchar(80) | YES  |     | NULL    |       |
| designation | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.05 sec)
```

```
mysql> alter table company
    -> add constraint g2 foreign key(empnumber)
    -> references emp(enumber);
Query OK, 4 rows affected (2.51 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> desc company;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| cname     | varchar(20) | YES  |     | NULL    |       |
| ccity     | varchar(30) | YES  |     | NULL    |       |
| empnumber | int         | YES  | MUL | NULL    |       |
| numofemp  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (1.58 sec)
```

```
mysql> alter table company
    -> add constraint g3 unique(cname);
Query OK, 0 rows affected (1.54 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc company;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| cname     | varchar(20) | YES  | UNI | NULL    |       |
| ccity     | varchar(30) | YES  |     | NULL    |       |
| empnumber | int         | YES  | MUL | NULL    |       |
| numofemp  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (2.02 sec)
```

```
mysql> alter table company
    -> alter column ccity set default 'Delhi';
Query OK, 0 rows affected (0.61 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc company;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| cname     | varchar(20) | YES  | UNI | NULL    |       |
| ccity     | varchar(30) | YES  |     | Delhi   |       |
| empnumber | int         | YES  | MUL | NULL    |       |
| numofemp  | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

# Experiment- 09

**Aim: -** To study the commands for aliasing and renaming and execute the following queries using these commands.

- **rename the name of database to employee**
- **rename the name of table 'emp' to 'empl'**
- **change the attribute from ' ename' to 'empname'.**

**Aliasing:** To rename a column or a table temp, by using another name or an alias, the change in temp and the actual table name does not change in database.

**Syntax:**

rename old_db_name to new_db_name;

or

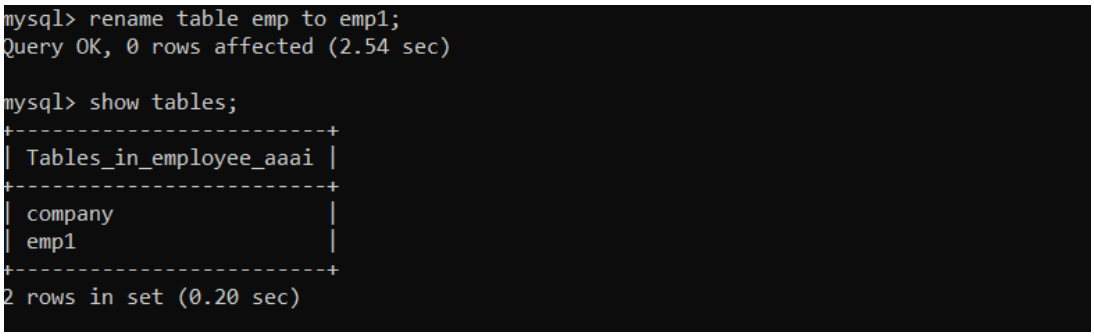
ALTER DATABASE old_name MODIFY TO +new_name;

EXEC sp_rename db 'old_name', 'new_name'

**Queries:**

1. Rename table emp to emp1;
2. Show tables;
3. Alter table emp1
   Change ename empname varchar(40);
4. Desc emp1;

**Output:**



```
mysql> alter table emp1
    -> change ename empname varchar(40);
Query OK, 0 rows affected (0.55 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc emp1;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| empname     | varchar(40) | YES  |     | NULL    |       |
| ecity       | varchar(40) | YES  |     | NULL    |       |
| salary      | float       | YES  |     | NULL    |       |
| enumber     | int         | NO   | PRI | NULL    |       |
| eaddress    | varchar(80) | YES  |     | NULL    |       |
| designation | varchar(30) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

# Experiment- 10

**Aim: -** To study the commands for joins (cross join, inner join, outer join) and execute the following queries using these commands.

- **retrieve the complete record of an employee and its company from both the table using joins. list all the employee working in the company tcs".**

**Join:** A JOIN clause is used to combine rows from two or more tables, based on or related column between them

**Cross Join:** the SQL CROSS JOIN Produces a result set which is the no. rows in the first table multiplied by the no of rows in the second table if no WHERE clause is used along with CROSS JOIN> this kind of result is called Cartesian product.

IF where clause is used with CROSS JOIN, it functions like an INNER JOIN.

**Syntax:**

select* from CROSS JOIN.

**Queries:**

1. Select enumber from emp1
   Where ecity = ‘Delhi’
   Union
   Select empnumber from company
   Where ccity = ‘Gurgaon’;
2. Select enumber from emp1
   Where ecity = ‘Delhi’ and enumber not in (
   Select empnumber from company
   Where ccity = ‘Gurgaon’);

**Output:**

```
mysql> select enumber from emp1
    -> where ecity ='Delhi'
    -> union
    -> select empnumber from company
    -> where ccity = 'Gurgaon';
+---------+
| enumber |
+---------+
|      16 |
|      19 |
|      21 |
+---------+
3 rows in set (0.00 sec)
```

```
mysql> select enumber from emp1
    -> where ecity ='Delhi' and enumber not in (
    -> select empnumber from company
    -> where ccity = 'Gurgaon');
+---------+
| enumber |
+---------+
|      19 |
+---------+
1 row in set (0.00 sec)
```

# Experiment- 11

**Aim: -** To study the various set operation and execute the following queries using these commands

- **List the number of all employees who lives in Delhi and whose company is in Gurugram or if both are true**
- **List the number of employees who live in Delhi but whose company is not in Gurugram.**

**Set Operation:**

**Union:** The union operation is used to combine results of 2 or more select operation. condition for union is that all the number of rows and columns must be same in both table on which select is applied

**Syntax: -**

```
select column_name from table1;
UNION;
select column_name from table2;
```

**Union all:** Union all operation is equal to the union operation. it returns the set without removing duplication and sorting the data.

**Syntax: -**

```
select column_name from table1;
union all
select column_name from table2;
```

**Intersect:** It is also used to combine 2 table but it returns the common rows from both tables. The number of columns must be the same.

**Syntax: -**

```
select column_name from table 1;
INTERSECT
select column_name from table 2;
```

**Minus:** It contains the result of two select statement minus operator s used to display the rows which are present in the first query but absent in the second query.

**Syntax: -**

```
select column_name from table1;

MINUS

select column_name from table2;
```

**Queries:**

1. Select emp1.*, company.*
   From emp1
   Cross join company;
2. Select emp1.empname
   From emp1
   Inner join company
   On emp1.enumber = company.empnumber
   Where cname = ‘TCS’;
3. Select emp1.empname
   From emp1,company
   Where emp1.enumber = company.empnumber and company.cname = ‘TCS’;

**Output:**

```
mysql> select emp1.* , company.*
    -> from emp1 cross join company;
```

| empname | ecity | salary | enumber | eaddress | designation | cname | ccity | empnumber | numofemp |
|---|---|---|---|---|---|---|---|---|---|
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | Mindtree | Delhi | 15 | 6550 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | Wipro | Gurgaon | 21 | 9850 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | Infosys | Mumbai | 19 | 8050 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | Microsoft | Gurgaon | 16 | 5600 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | Amazon | Mumbai | 13 | 10020 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | OLA | Chennai | 10 | 10050 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | TCS | Delhi | 12 | 15000 |
| Anubha | Gurgaon | 40000 | 10 | Karol Bagh | Project Head | HCL | Banglore | 11 | 10500 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Mindtree | Delhi | 15 | 6550 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Infosys | Mumbai | 19 | 8050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Amazon | Mumbai | 13 | 10020 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | OLA | Chennai | 10 | 10050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | TCS | Delhi | 12 | 15000 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | HCL | Banglore | 11 | 10500 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Mindtree | Delhi | 15 | 6550 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Wipro | Gurgaon | 21 | 9850 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Infosys | Mumbai | 19 | 8050 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Microsoft | Gurgaon | 16 | 5600 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Amazon | Mumbai | 13 | 10020 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | OLA | Chennai | 10 | 10050 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | TCS | Delhi | 12 | 15000 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | HCL | Banglore | 11 | 10500 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Mindtree | Delhi | 15 | 6550 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Wipro | Gurgaon | 21 | 9850 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Infosys | Mumbai | 19 | 8050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Microsoft | Gurgaon | 16 | 5600 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Amazon | Mumbai | 13 | 10020 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | OLA | Chennai | 10 | 10050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | TCS | Delhi | 12 | 15000 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | HCL | Banglore | 11 | 10500 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Mindtree | Delhi | 15 | 6550 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Infosys | Mumbai | 19 | 8050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Amazon | Mumbai | 13 | 10020 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | OLA | Chennai | 10 | 10050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | TCS | Delhi | 12 | 15000 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | HCL | Banglore | 11 | 10500 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | OLA | Chennai | 10 | 10050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | TCS | Delhi | 12 | 15000 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | HCL | Banglore | 11 | 10500 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | OLA | Chennai | 10 | 10050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | TCS | Delhi | 12 | 15000 |

```
mysql> select emp1.empname from emp1
    -> inner join company on
    -> emp1.enumber = company.empnumber
    -> where cname = 'TCS';
+------------+
| empname    |
+------------+
| Inderpreet |
+------------+
1 row in set (0.00 sec)

mysql> select emp1.empname
    -> from emp1,company
    -> where emp1.enumber = company.empnumber and company.cname = 'TCS';
+------------+
| empname    |
+------------+
| Inderpreet |
+------------+
1 row in set (0.00 sec)
```

# Experiment- 12

**Aim: -**To study the various scalex function and string function and execute the following quires using the above commands.

**Theory: -**

**Scalex function:** A SQL scalar function is a user defined function written in SQL and it returns a single value each time it is invoked.

1) uppercase function: UCASE (): it converts the value of a field ti uppercase

   **Syntax:**
   SELECT UCASE (column_name) from table_name;

2) lowercase function: LCASE (): it converts the value of a fields to lowercase

   **Syntax:**
   SELECT LCASE ( column_name) from table_name

**String function:** These functions are used to perform an operation on input sorting and return an output string

1) LOWER (): this function converts the uppercase string into lower case

   **Syntax:**
   select LOWER (column_ name) FROM table_name;

2) REVERSE (): returns the string with the order of the characters reversed

   **Syntax:**
   SELECT REVERSE (column_name) FROM table_name;

3) SUBSTRING (): return parts of a character binary text or images expression in SQL server.

   **Syntax:**
   select SUBSTRING (expression, start, length) from table_name.

**Commands:**

1) SELECT REVERSE (ename) from emp;

2) SELECT UCASE (ename) FROM company;

3) SELECT CONCAT (ename, ecity) from emp;

**QUERIES:**

1. Select reverse(empname)
   From emp1;
2. Select upper(ccity) from company;
3. Select concat(empname,ecity)
   From emp1;

**Output:**

```
mysql> select reverse(empname)
    -> from emp1;
+------------------+
| reverse(empname) |
+------------------+
| ahbunA           |
| lomnA            |
| teerprednI       |
| tiprA            |
| htrahddiS        |
| hbahsiR          |
| sirK             |
| rahsuT           |
| hsejaR           |
| ateeG            |
| jaknaP           |
| nitiN            |
+------------------+
12 rows in set (0.03 sec)
```

```
mysql> select upper(ccity) from company;
+--------------+
| upper(ccity) |
+--------------+
| BANGLORE     |
| DELHI        |
| CHENNAI      |
| MUMBAI       |
| GURGAON      |
| MUMBAI       |
| GURGAON      |
| DELHI        |
+--------------+
8 rows in set (0.09 sec)
```

```
mysql> select concat(empname,ecity)
    -> from emp1;
+-----------------------+
| concat(empname,ecity) |
+-----------------------+
| AnubhaGurgaon         |
| AnmolGurgaon          |
| InderpreetBanglore    |
| ArpitChandigarh       |
| SiddharthGurgaon      |
| RishabhDelhi          |
| KrisChandigarh        |
| TusharBanglore        |
| RajeshDelhi           |
| GeetaMumbai           |
| PankajDelhi           |
| NitinBanglore         |
+-----------------------+
12 rows in set (0.03 sec)
```

# EXPERIMENT-13

**Aim:** to study the commands for view and execute the following queries using these commands.

- **create a view having ename and ecity.**
- **in the above view change, the ecity to 'Delhi' where ename is 'john'.**
- **create a view having attributes from both the tables.**
- **update the above view and increase the salary of all employees of IT department by rs.1000.**

**Theory: -**

**View:** view in SQL is a virtual table based on the result-set of an SQL statement.

**Syntax:**

CREATE VIEW view_name AS

SELECT column, column2....

FROM table_name

WHERE condition;

**Example:**

CREATE VIEW [Brazil customers]AS

SELECT customerName, contactName

FROM customers

WHERE country='Brazil

**SQL updating a view:** a view can be updated with the CREATE OR REPLACE VIEW command.

**Syntax:**

CREATE OR REPLACE VIEW Name AS

SELECT column1, column 2....

FROM table-name

WHERE condition;

**Example:**

CREATE OR REPLACE VIEW [Brazil customer]AS

SELECT customerName, ContactName, contactName, city

FROM customers

WHERE counrty ="brazil;

**SQL Dropping a view**: A view is deleted with a DROP view command

**Syntax:**

DROP VIEW view_name;

**QUERIES:**

1. Create view v1 as (select empname, ecity from emp1);
2. Select * from v1;
3. Update v1
   Set ecity = ‘Delhi’
   Where empname = ‘John’;
4. Select * from v1;
5. Create view v2 as (select emp1.*, company.* from emp1,company);
6. Select * from v2;
7. Update v2
   Set salary = salary + 1000
   Where Designation = ‘IT Head’;
8. Select * from v2;

**Output:**

```
mysql> Create view v1 as (select empname, ecity from emp1);
Query OK, 0 rows affected (0.18 sec)

mysql> Select * from v1;
+------------+------------+
| empname    | ecity      |
+------------+------------+
| Anubha     | Gurgaon    |
| Anmol      | Gurgaon    |
| Inderpreet | Banglore   |
| Arpit      | Chandigarh |
| Siddharth  | Gurgaon    |
| Rishabh    | Delhi      |
| Kris       | Chandigarh |
| Tushar     | Banglore   |
| Rajesh     | Delhi      |
| Geeta      | Mumbai     |
| Pankaj     | Delhi      |
| Nitin      | Banglore   |
+------------+------------+
12 rows in set (0.09 sec)
```

```
mysql> update v1
    -> set ecity = 'Delhi'
    -> where empname = 'Anubha';
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from v1;
+------------+------------+
| empname    | ecity      |
+------------+------------+
| Anubha     | Delhi      |
| Anmol      | Gurgaon    |
| Inderpreet | Banglore   |
| Arpit      | Chandigarh |
| Siddharth  | Gurgaon    |
| Rishabh    | Delhi      |
| Kris       | Chandigarh |
| Tushar     | Banglore   |
| Rajesh     | Delhi      |
| Geeta      | Mumbai     |
| Pankaj     | Delhi      |
| Nitin      | Banglore   |
+------------+------------+
12 rows in set (0.00 sec)
```

```
mysql> create view v2 as (select emp1.*,company.* from emp1,company);
Query OK, 0 rows affected (3.00 sec)

mysql> select * from v2;
```

| empname | ecity | salary | enumber | eaddress | designation | cname | ccity | empnumber | numofemp |
|---|---|---|---|---|---|---|---|---|---|
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Mindtree | Delhi | 15 | 6550 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Wipro | Gurgaon | 21 | 9850 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Infosys | Mumbai | 19 | 8050 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Microsoft | Gurgaon | 16 | 5600 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Amazon | Mumbai | 13 | 10020 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | OLA | Chennai | 10 | 10050 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | TCS | Delhi | 12 | 15000 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | HCL | Banglore | 11 | 10500 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Mindtree | Delhi | 15 | 6550 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Infosys | Mumbai | 19 | 8050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Amazon | Mumbai | 13 | 10020 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | OLA | Chennai | 10 | 10050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | TCS | Delhi | 12 | 15000 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | HCL | Banglore | 11 | 10500 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Mindtree | Delhi | 15 | 6550 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Wipro | Gurgaon | 21 | 9850 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Infosys | Mumbai | 19 | 8050 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Microsoft | Gurgaon | 16 | 5600 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | Amazon | Mumbai | 13 | 10020 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | OLA | Chennai | 10 | 10050 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | TCS | Delhi | 12 | 15000 |
| Inderpreet | Banglore | 45000 | 12 | Subhash Nagar | IT Head | HCL | Banglore | 11 | 10500 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Mindtree | Delhi | 15 | 6550 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Wipro | Gurgaon | 21 | 9850 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Infosys | Mumbai | 19 | 8050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Microsoft | Gurgaon | 16 | 5600 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Amazon | Mumbai | 13 | 10020 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | OLA | Chennai | 10 | 10050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | TCS | Delhi | 12 | 15000 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | HCL | Banglore | 11 | 10500 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Mindtree | Delhi | 15 | 6550 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Infosys | Mumbai | 19 | 8050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Amazon | Mumbai | 13 | 10020 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | OLA | Chennai | 10 | 10050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | TCS | Delhi | 12 | 15000 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | HCL | Banglore | 11 | 10500 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | OLA | Chennai | 10 | 10050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | TCS | Delhi | 12 | 15000 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | HCL | Banglore | 11 | 10500 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Microsoft | Gurgaon | 16 | 5600 |

| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Amazon | Mumbai | 13 | 10020 |
|---|---|---|---|---|---|---|---|---|---|
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | OLA | Chennai | 10 | 10050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | TCS | Delhi | 12 | 15000 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | HCL | Banglore | 11 | 10500 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | OLA | Chennai | 10 | 10050 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | TCS | Delhi | 12 | 15000 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | HCL | Banglore | 11 | 10500 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Mindtree | Delhi | 15 | 6550 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Wipro | Gurgaon | 21 | 9850 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Infosys | Mumbai | 19 | 8050 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Microsoft | Gurgaon | 16 | 5600 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Amazon | Mumbai | 13 | 10020 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | OLA | Chennai | 10 | 10050 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | TCS | Delhi | 12 | 15000 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | HCL | Banglore | 11 | 10500 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Mindtree | Delhi | 15 | 6550 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Wipro | Gurgaon | 21 | 9850 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Infosys | Mumbai | 19 | 8050 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Microsoft | Gurgaon | 16 | 5600 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Amazon | Mumbai | 13 | 10020 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | OLA | Chennai | 10 | 10050 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | TCS | Delhi | 12 | 15000 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | HCL | Banglore | 11 | 10500 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Mindtree | Delhi | 15 | 6550 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Wipro | Gurgaon | 21 | 9850 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Infosys | Mumbai | 19 | 8050 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Microsoft | Gurgaon | 16 | 5600 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Amazon | Mumbai | 13 | 10020 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | OLA | Chennai | 10 | 10050 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | TCS | Delhi | 12 | 15000 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | HCL | Banglore | 11 | 10500 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Mindtree | Delhi | 15 | 6550 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Wipro | Gurgaon | 21 | 9850 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Infosys | Mumbai | 19 | 8050 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Microsoft | Gurgaon | 16 | 5600 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Amazon | Mumbai | 13 | 10020 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | OLA | Chennai | 10 | 10050 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | TCS | Delhi | 12 | 15000 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | HCL | Banglore | 11 | 10500 |

```
96 rows in set (0.41 sec)
```

```
mysql> update v2
    -> set salary = salary + 1000
    -> where Designation = 'IT Head';
Query OK, 1 row affected (15.40 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from v2;
```

| empname | ecity | salary | enumber | eaddress | designation | cname | ccity | empnumber | numofemp |
|---|---|---|---|---|---|---|---|---|---|
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Mindtree | Delhi | 15 | 6550 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Wipro | Gurgaon | 21 | 9850 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Infosys | Mumbai | 19 | 8050 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Microsoft | Gurgaon | 16 | 5600 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | Amazon | Mumbai | 13 | 10020 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | OLA | Chennai | 10 | 10050 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | TCS | Delhi | 12 | 15000 |
| Anubha | Delhi | 40000 | 10 | Karol Bagh | Project Head | HCL | Banglore | 11 | 10500 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Mindtree | Delhi | 15 | 6550 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Infosys | Mumbai | 19 | 8050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | Amazon | Mumbai | 13 | 10020 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | OLA | Chennai | 10 | 10050 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | TCS | Delhi | 12 | 15000 |
| Anmol | Gurgaon | 29000 | 11 | Janakpuri | Programmer | HCL | Banglore | 11 | 10500 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | Mindtree | Delhi | 15 | 6550 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | Wipro | Gurgaon | 21 | 9850 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | Infosys | Mumbai | 19 | 8050 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | Microsoft | Gurgaon | 16 | 5600 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | Amazon | Mumbai | 13 | 10020 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | OLA | Chennai | 10 | 10050 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | TCS | Delhi | 12 | 15000 |
| Inderpreet | Banglore | 46000 | 12 | Subhash Nagar | IT Head | HCL | Banglore | 11 | 10500 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Mindtree | Delhi | 15 | 6550 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Wipro | Gurgaon | 21 | 9850 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Infosys | Mumbai | 19 | 8050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Microsoft | Gurgaon | 16 | 5600 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | Amazon | Mumbai | 13 | 10020 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | OLA | Chennai | 10 | 10050 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | TCS | Delhi | 12 | 15000 |
| Arpit | Chandigarh | 36000 | 13 | Uttam Nagar | Manager | HCL | Banglore | 11 | 10500 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Mindtree | Delhi | 15 | 6550 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Wipro | Gurgaon | 21 | 9850 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Infosys | Mumbai | 19 | 8050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Microsoft | Gurgaon | 16 | 5600 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | Amazon | Mumbai | 13 | 10020 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | OLA | Chennai | 10 | 10050 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | TCS | Delhi | 12 | 15000 |
| Siddharth | Gurgaon | 22000 | 15 | Patel Nagar | Programmer | HCL | Banglore | 11 | 10500 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | OLA | Chennai | 10 | 10050 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | TCS | Delhi | 12 | 15000 |
| Rishabh | Delhi | 65000 | 16 | Shadipur | Computer Science | HCL | Banglore | 11 | 10500 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | OLA | Chennai | 10 | 10050 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | TCS | Delhi | 12 | 15000 |
| Kris | Chandigarh | 80000 | 17 | Subhash Nagar | Computer Science | HCL | Banglore | 11 | 10500 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Mindtree | Delhi | 15 | 6550 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Wipro | Gurgaon | 21 | 9850 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Infosys | Mumbai | 19 | 8050 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Microsoft | Gurgaon | 16 | 5600 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | Amazon | Mumbai | 13 | 10020 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | OLA | Chennai | 10 | 10050 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | TCS | Delhi | 12 | 15000 |
| Tushar | Banglore | 62000 | 18 | Patel Nagar | Computer Science | HCL | Banglore | 11 | 10500 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Mindtree | Delhi | 15 | 6550 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Wipro | Gurgaon | 21 | 9850 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Infosys | Mumbai | 19 | 8050 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Microsoft | Gurgaon | 16 | 5600 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | Amazon | Mumbai | 13 | 10020 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | OLA | Chennai | 10 | 10050 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | TCS | Delhi | 12 | 15000 |
| Rajesh | Delhi | 58000 | 19 | Janakpuri | IT | HCL | Banglore | 11 | 10500 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Mindtree | Delhi | 15 | 6550 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Wipro | Gurgaon | 21 | 9850 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Infosys | Mumbai | 19 | 8050 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Microsoft | Gurgaon | 16 | 5600 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | Amazon | Mumbai | 13 | 10020 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | OLA | Chennai | 10 | 10050 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | TCS | Delhi | 12 | 15000 |
| Geeta | Mumbai | 75664 | 20 | Uttam Nagar | HR | HCL | Banglore | 11 | 10500 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Mindtree | Delhi | 15 | 6550 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Wipro | Gurgaon | 21 | 9850 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Infosys | Mumbai | 19 | 8050 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Microsoft | Gurgaon | 16 | 5600 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | Amazon | Mumbai | 13 | 10020 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | OLA | Chennai | 10 | 10050 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | TCS | Delhi | 12 | 15000 |
| Pankaj | Delhi | 55000 | 21 | Karol Bagh | HR | HCL | Banglore | 11 | 10500 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Mindtree | Delhi | 15 | 6550 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Wipro | Gurgaon | 21 | 9850 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Infosys | Mumbai | 19 | 8050 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Microsoft | Gurgaon | 16 | 5600 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | Amazon | Mumbai | 13 | 10020 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | OLA | Chennai | 10 | 10050 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | TCS | Delhi | 12 | 15000 |
| Nitin | Banglore | 78000 | 22 | Shadipur | HR | HCL | Banglore | 11 | 10500 |

```
96 rows in set (1 min 1.33 sec)
```

# Experiment-14

**Aim:** To study the commands involving and execute the following queries:

- create an index with attribute ename on the table employee.
- create a composite index with attributes ename and city on the table company
- drop all Indore created on table company.

**Theory:** Indexes are special lookup tables that the database engine can use to speed up data retrieval.

Create index command: Basic

**Syntax:**

CREATE INDEX index_name ON table_name;

1. **Single column indexes:** A single_column index ps created based on only one
   table column.
   **Syntax:**
   CREATEINDEX index_name
   ON table_name (column_name);
2. **Unique indexes:** unique index are used not only for performance also for data integrity. a unique index does not allow any duplicate values to be inserted into the table.
   **Syntax:**
   CREATE UNIQUE INDEX index_name.
   ON table_name (column_name);
3. **Composite indexes:** A composite index is an index on two or more columns of an table.
   **Syntax:**
   CREATE INDEX index_name
   ON table name (column1, column2);
4. **The drop INDEX Command:** the index can be dropped using SQL DROP command.
   **Syntax:**
   DROP INDEX name;

indexes should not be used on small tables ·

indexes should not be used on columns that contain a high number of NULL value c.

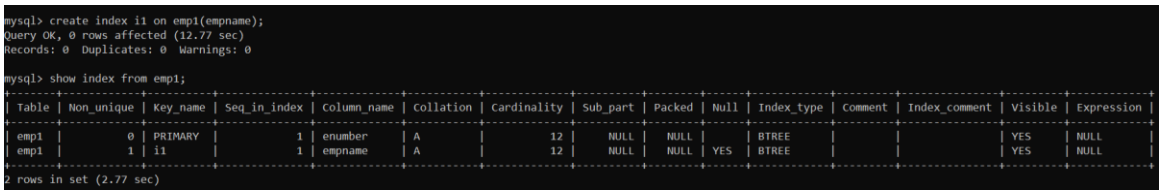
columns that are frequently manipulated should not be indexed.

**QUERIES:**

1. Create index i1 on emp1(empname);
2. Show index from emp1;
3. Create index i2 on company(cname,ccity);
4. Show index from company;

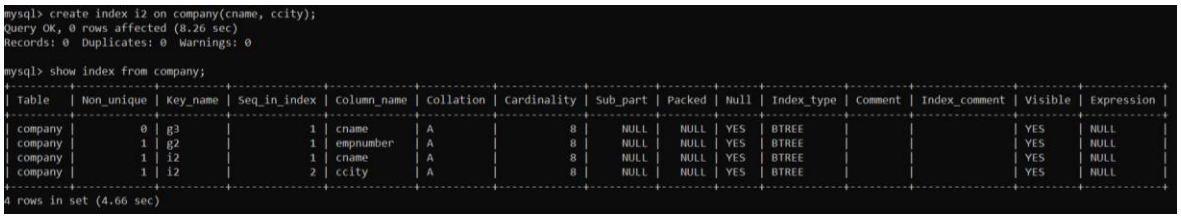
5. **Alter** table company
   Drop index i2;
6. Show index from company;

## Output:



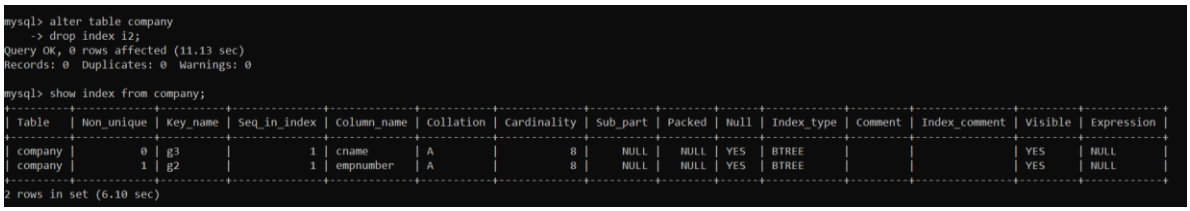
```
mysql> create index i1 on emp1(empname);
Query OK, 0 rows affected (12.77 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from emp1;
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| emp1  |          0 | PRIMARY  |            1 | enumber     | A         |          12 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| emp1  |          1 | i1       |            1 | empname     | A         |          12 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+-------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (2.77 sec)
```



```
mysql> create index i2 on company(cname, ccity);
Query OK, 0 rows affected (8.26 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from company;
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table   | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| company |          0 | g3       |            1 | cname       | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| company |          1 | g2       |            1 | empnumber   | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| company |          1 | i2       |            1 | cname       | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| company |          1 | i2       |            2 | ccity       | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
4 rows in set (4.66 sec)
```



```
mysql> alter table company
    -> drop index i2;
Query OK, 0 rows affected (11.13 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from company;
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table   | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| company |          0 | g3       |            1 | cname       | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| company |          1 | g2       |            1 | empnumber   | A         |           8 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+---------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
2 rows in set (6.10 sec)
```

# Experiment - 15

**Aim:** Introduction to PL-SQL

**Theory:**

PL-SQL is a block structured language that enables developer to combine the power of SQL with procedures statements. All the statement are passed to oracle engine at once which increase processing speed and decreases traffic.

**Features of PL SQL**

It's basically a procedural language which provides the functionality of decision making and many more features.

It can execute a no. Of query in one block using single command.

One can create a PL SQL unit such as procedures, functions, packages triggers which are stored in database for reuse of application

It provides a feature called exception building block

Application written in PL SQL can be portable to computer hardware or O.S. Where Oracle engine operates

It offers extensive error checking

**Structure of PL SQL block**

PL SQL block

Declare

Basic

Exception

End

**A block has following structure:**

Declare

Declaration statement;

Basic Executable statement;

Exception

Exception handling statement

End;

Declaration starts with declare keyword where variables, constant, record as cursor and declared and stored temporarily.

Execution starts with begin and end with END keywords Exception section is optional which counter statement that are executed when a run time error occurred

# Experiment -16

**Aim:** To study the conditional controls and case statement in LSQL and execute the following queries

- **Calculate the average salary from the employees and print increase the salary if the average salary is less than 10,000**
- **Print the deptno from the employees table using the case statement if the designation is project head, then designation number is 1 and if designation is computer science, then designation number is 2 LSD designation number is 3**

**Commands**:

Conditional statements:

**Syntax**:
If condition 1 then
Statement to be executed if condition 1 is true
Else if condition 2 then
Statement to be executed if condition two is true
Else
Statement to be executed if condition one and two is false
end if
Case statement:
It selects one sequence of statement to execute
**Syntax**:
case statement
When value 1 then
Statement to be executed when expression value = 1
When value 2 then
Statement two execute when expression = value 2
Else
Statement to execute when no value match
End case
**Queries**:

1. Delimiter //
2. Create procedure g()
   Begin
   Select avg(salary), if(avg(salary)<10000, 'Increase salary', 'ok') as decision from emp1;
   End;
   //
3. Delimiter ;
4. Call g;
5. Select designation,
   Case
   When designation = 'Project Head' then 1
   When designation = 'Computer Science' then 2
   Else 3
   End as designation

From emp1;

**Output:**

```
mysql> delimiter //
mysql> create procedure g()
    -> begin
    -> select avg(salary), if(avg(salary)<10000,'Increase Salary','Ok') as decision from emp1;
    -> end;
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql> call g;
+-------------+----------+
| avg(salary) | decision |
+-------------+----------+
|       55850 | Ok       |
+-------------+----------+
1 row in set (0.00 sec)
```

```
mysql> select designation,
    -> case
    -> when designation ='IT Head' then 1
    -> when designation = 'Programmer' then 2
    -> else 3
    -> end as designation
    -> from emp1;
+------------------+-------------+
| designation      | designation |
+------------------+-------------+
| Project Head     |           3 |
| Programmer       |           2 |
| IT Head          |           1 |
| Manager          |           3 |
| Programmer       |           2 |
| Computer Science |           3 |
| Computer Science |           3 |
| Computer Science |           3 |
| IT               |           3 |
| HR               |           3 |
| HR               |           3 |
| HR               |           3 |
+------------------+-------------+
12 rows in set (0.00 sec)
```

# Experiment-17

**Aim:** To study procedure and trigger in SQL and execute the following queries

- **Create a procedure on the table employee to display the details of employee by providing them values of salaries during execution**
- **Create a trigger on table company for insertion that if the number of employee entered by the user is less than 1000 the number new value of number of employees should be automatically set 2000**

**Command**:

1) Procedure it is sorted program that we can pass parent into it but does not return a value

   **Syntax:**

   create procedure procedure_name(parameter datatype)
   Begin
   Declaration section
   Executable section
   End
   # Note to drop procedure we use syntax
   Drop procedure procedure_name ;

2) Triggers triggers are stored programs which are automatically executed or prefixed when some events occur

   **Syntax**:

   Create trigger trigger name
   Before/ after delete/ update/ insert on table name for each row
   Begin
   Variable declaration
   Trigger code
   End

**Queries**:

1. Delimiter //
2. Create procedure sal(num int)
   Begin
   Select * from emp1 where salary = num;
   End;
   //
3. Delimiter ;
4. Call sal(21000);
5. Delimiter //
6. Create trigger t1 before insert on company for each row
   If New.numofemp < 1000 then
   Set New.numofemp = 1000;
   End if;
   //
7. Delimiter ;
8. Insert into company values(‘Realme’, ‘Noida’, ’37,500);

9. Select * from company;

Output:

```
mysql> use Employee_AAAI;
Database changed
mysql> delimiter //
mysql> create procedure sal(num int)
    -> begin
    -> select * from emp1 where salary = num;
    -> end;
    -> //
Query OK, 0 rows affected (3.59 sec)

mysql> delimiter ;
mysql> call sal(65000);
+---------+-------+--------+---------+----------+------------------+
| empname | ecity | salary | enumber | eaddress | designation      |
+---------+-------+--------+---------+----------+------------------+
| Rishabh | Delhi |  65000 |      16 | Shadipur | Computer Science |
+---------+-------+--------+---------+----------+------------------+
1 row in set (0.39 sec)

Query OK, 0 rows affected (0.40 sec)
```

```
mysql> delimiter //
mysql> create trigger t1 before insert on company for each row
    -> if New.numofemp < 1000 then
    -> set New.numofemp = 1000;
    -> end if;
    -> //
Query OK, 0 rows affected (0.14 sec)

mysql> delimiter ;
mysql> insert into company values('Realme','Noida',37,500);
Query OK, 1 row affected (0.16 sec)

mysql> select * from company;
+-------------+----------+-----------+-----------+
| cname       | ccity    | empnumber | numofemp  |
+-------------+----------+-----------+-----------+
| TCS         | Gurgaon  |        15 |     15000 |
| HCL         | Banglore |        12 |     11800 |
| OLA         | Chennai  |        13 |      1250 |
| Tata Motors | Delhi    |        29 |     11000 |
| Samsung     | Delhi    |        10 |      1050 |
| Realme      | Noida    |        37 |      1000 |
+-------------+----------+-----------+-----------+
6 rows in set (0.00 sec)

mysql>
```