**Lab 1, 2 & 3**

**Theory**

* The CREATE DATABASE statement is used to create a new SQL database.
* The CREATE TABLE statement is used to create a new table in a database.
* The DESC command is used to sort the data returned in descending order.
* The INSERT INTO statement is used to insert new records in a table.
* The SELECT statement is used to select data from a database.
* The UPDATE statement is used to modify the existing records in a table.
* The SELECT DISTINCT statement is used to return only distinct (different) values.
* The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
* MySQL's aggregate function is used to perform calculations on multiple values and return the result in a single value like the average of all values
  + **count():** It returns the number of rows, including rows with NULL values in a group.
  + **sum():** It returns the total summed values (Non-NULL) in a set.
  + **average():** It returns the average value of an expression.
  + **min():** It returns the minimum (lowest) value in a set.
  + **max():** It returns the maximum (highest) value in a set.
* The DELETE statement is used to delete existing records in a table.
* The PRIMARY KEY constraint uniquely identifies each record in a table.
  + Primary keys must contain UNIQUE values, and cannot contain NULL values.
  + A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.00 sec)

mysql> use college;

Database changed

mysql> create table student(enrollment\_num int, name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.01 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | YES | | NULL | |

| name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.00 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(140, "Ram Sharma", "M", 18, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(150, "Devansh Singh", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(155, "Shruti Malviya", "F", 19, "CE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(159, "Dipika Jain", "F", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(170, "Sakshi Sharma", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> select \* from student;

+----------------+----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+----------------+--------+------+--------+

| 34 | Garvit | M | 18 | CSE |

| 44 | Aayushi | F | 17 | EC |

| 47 | Kratik | M | 19 | ME |

| 59 | Diksha | F | 18 | AU |

| 39 | Gaurav | M | 18 | CSBS |

| 67 | Mohit | M | 19 | IT |

| 140 | Ram Sharma | M | 18 | CSE |

| 150 | Devansh Singh | M | 19 | ME |

| 155 | Shruti Malviya | F | 19 | CE |

| 159 | Dipika Jain | F | 19 | IT |

| 170 | Sakshi Sharma | F | 18 | AU |

+----------------+----------------+--------+------+--------+

11 rows in set (0.01 sec)

mysql> create table faculty(Faculty\_id mediumint, Name char(30), Gender char(1), Age int, Dept varchar(10), Salary float(5, 2), Year\_joined year);

Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> desc faculty;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| Faculty\_id | mediumint | YES | | NULL | |

| Name | char(30) | YES | | NULL | |

| Gender | char(1) | YES | | NULL | |

| Age | int | YES | | NULL | |

| Dept | varchar(10) | YES | | NULL | |

| Salary | float(5,2) | YES | | NULL | |

| Year\_joined | year | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

7 rows in set (0.00 sec)

mysql> insert into faculty values(101, "Rohit Mahajan", "M", 30, "ME", 344.9,'2008');

Query OK, 1 row affected (0.01 sec)

mysql> insert into faculty values(106, "Monika Jain", "F", 25, "CSE", 548.0,'2009');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(102, "Kiran Talwariya", "F", 34, "IT", 783.0,'2012');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(110, "Jitendra Sharma", "M", 45, "CE", 993.0,'2005');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(107, "Sheetal Agrawal", "F", 40, "AU", 678.0,'2009');

Query OK, 1 row affected (0.01 sec)

mysql> insert into faculty values(105, "Rahul Sharma", "M", 51, "ME", 693.0,'2010');

Query OK, 1 row affected (0.00 sec)

mysql> select \* from faculty;

+------------+-----------------+--------+------+------+--------+-------------+

| Faculty\_id | Name | Gender | Age | Dept | Salary | Year\_joined |

+------------+-----------------+--------+------+------+--------+-------------+

| 101 | Rohit Mahajan | M | 30 | ME | 344.90 | 2008 |

| 106 | Monika Jain | F | 25 | CSE | 548.00 | 2009 |

| 102 | Kiran Talwariya | F | 34 | IT | 783.00 | 2012 |

| 110 | Jitendra Sharma | M | 45 | CE | 993.00 | 2005 |

| 107 | Sheetal Agrawal | F | 40 | AU | 678.00 | 2009 |

| 105 | Rahul Sharma | M | 51 | ME | 693.00 | 2010 |

+------------+-----------------+--------+------+------+--------+-------------+

6 rows in set (0.00 sec)

mysql> update student set gender = 'F' where name = 'Anshika Gupta';

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0 Changed: 0 Warnings: 0

mysql> update student set age=19 where enrollment\_num in (34, 140, 59, 170);

Query OK, 4 rows affected (0.01 sec)

Rows matched: 4 Changed: 4 Warnings: 0

mysql> update student set age=18 where enrollment\_num not in (34, 140, 59, 170);

Query OK, 6 rows affected (0.00 sec)

Rows matched: 7 Changed: 6 Warnings: 0

mysql> update student set name = "Diksha Rathore" where name = "Diksha";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Neha Reddy" where name = "Neha";

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0 Changed: 0 Warnings: 0

mysql> update student set name = "Gaurav Sharma" where name = "Gaurav";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Garvit Paliwal" where name = "Garvit";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Kratik Mathur" where name = "Kratik";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Aayushi Talreja" where name = "Aayushi";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Mohit Pherwani" where name = "Mohit";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update faculty set Name = "Karan Singh" where Faculty\_id = 102;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select distinct name from student;

+-----------------+

| name |

+-----------------+

| Garvit Paliwal |

| Aayushi Talreja |

| Kratik Mathur |

| Diksha Rathore |

| Gaurav Sharma |

| Mohit Pherwani |

| Ram Sharma |

| Devansh Singh |

| Shruti Malviya |

| Dipika Jain |

| Sakshi Sharma |

+-----------------+

11 rows in set (0.00 sec)

mysql> select distinct name and enrollment\_num from student;

+-------------------------+

| name and enrollment\_num |

+-------------------------+

| 0 |

+-------------------------+

1 row in set, 11 warnings (0.00 sec)

mysql> select \* from student where gender='M';

+----------------+----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+----------------+--------+------+--------+

| 34 | Garvit Paliwal | M | 19 | CSE |

| 47 | Kratik Mathur | M | 18 | ME |

| 39 | Gaurav Sharma | M | 18 | CSBS |

| 67 | Mohit Pherwani | M | 18 | IT |

| 140 | Ram Sharma | M | 19 | CSE |

| 150 | Devansh Singh | M | 18 | ME |

+----------------+----------------+--------+------+--------+

6 rows in set (0.00 sec)

mysql> select \* from student where branch='CSE';

+----------------+----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+----------------+--------+------+--------+

| 34 | Garvit Paliwal | M | 19 | CSE |

| 140 | Ram Sharma | M | 19 | CSE |

+----------------+----------------+--------+------+--------+

2 rows in set (0.00 sec)

mysql> select name from student where branch='CSE';

+----------------+

| name |

+----------------+

| Garvit Paliwal |

| Ram Sharma |

+----------------+

2 rows in set (0.00 sec)

mysql> select \* from student where name like 'D%';

+----------------+----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+----------------+--------+------+--------+

| 59 | Diksha Rathore | F | 19 | AU |

| 150 | Devansh Singh | M | 18 | ME |

| 159 | Dipika Jain | F | 18 | IT |

+----------------+----------------+--------+------+--------+

3 rows in set (0.00 sec)

mysql> select \* from student where branch like 'CSE%';

+----------------+----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+----------------+--------+------+--------+

| 34 | Garvit Paliwal | M | 19 | CSE |

| 140 | Ram Sharma | M | 19 | CSE |

+----------------+----------------+--------+------+--------+

2 rows in set (0.00 sec)

mysql> select enrollment\_num, name from student where branch='CSE';

+----------------+----------------+

| enrollment\_num | name |

+----------------+----------------+

| 34 | Garvit Paliwal |

| 140 | Ram Sharma |

+----------------+----------------+

2 rows in set (0.00 sec)

mysql> select \* from faculty where Year\_joined between 2000 and 2009;

+------------+-----------------+--------+------+------+--------+-------------+

| Faculty\_id | Name | Gender | Age | Dept | Salary | Year\_joined |

+------------+-----------------+--------+------+------+--------+-------------+

| 101 | Rohit Mahajan | M | 30 | ME | 344.90 | 2008 |

| 106 | Monika Jain | F | 25 | CSE | 548.00 | 2009 |

| 110 | Jitendra Sharma | M | 45 | CE | 993.00 | 2005 |

| 107 | Sheetal Agrawal | F | 40 | AU | 678.00 | 2009 |

+------------+-----------------+--------+------+------+--------+-------------+

4 rows in set (0.00 sec)

mysql> select \* from faculty order by faculty\_id;

+------------+-----------------+--------+------+------+--------+-------------+

| Faculty\_id | Name | Gender | Age | Dept | Salary | Year\_joined |

+------------+-----------------+--------+------+------+--------+-------------+

| 101 | Rohit Mahajan | M | 30 | ME | 344.90 | 2008 |

| 102 | Karan Singh | F | 34 | IT | 783.00 | 2012 |

| 105 | Rahul Sharma | M | 51 | ME | 693.00 | 2010 |

| 106 | Monika Jain | F | 25 | CSE | 548.00 | 2009 |

| 107 | Sheetal Agrawal | F | 40 | AU | 678.00 | 2009 |

| 110 | Jitendra Sharma | M | 45 | CE | 993.00 | 2005 |

+------------+-----------------+--------+------+------+--------+-------------+

6 rows in set (0.00 sec)

mysql> select \* from student order by enrollment\_num;

+----------------+-----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+-----------------+--------+------+--------+

| 34 | Garvit Paliwal | M | 19 | CSE |

| 39 | Gaurav Sharma | M | 18 | CSBS |

| 44 | Aayushi Talreja | F | 18 | EC |

| 47 | Kratik Mathur | M | 18 | ME |

| 59 | Diksha Rathore | F | 19 | AU |

| 67 | Mohit Pherwani | M | 18 | IT |

| 140 | Ram Sharma | M | 19 | CSE |

| 150 | Devansh Singh | M | 18 | ME |

| 155 | Shruti Malviya | F | 18 | CE |

| 159 | Dipika Jain | F | 18 | IT |

| 170 | Sakshi Sharma | F | 19 | AU |

+----------------+-----------------+--------+------+--------+

11 rows in set (0.00 sec)

mysql> select max(salary) from faculty;

+-------------+

| max(salary) |

+-------------+

| 993.00 |

+-------------+

1 row in set (0.00 sec)

mysql> select min(age) from student;

+----------+

| min(age) |

+----------+

| 18 |

+----------+

1 row in set (0.00 sec)

mysql> select avg(age) from student;

+----------+

| avg(age) |

+----------+

| 18.3636 |

+----------+

1 row in set (0.00 sec)

mysql> select sum(salary) from faculty;

+-------------+

| sum(salary) |

+-------------+

| 4039.90 |

+-------------+

1 row in set (0.00 sec)

mysql> alter table faculty add dob date after Age;

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table faculty modify Faculty\_id varchar(10);

Query OK, 6 rows affected (0.03 sec)

Records: 6 Duplicates: 0 Warnings: 0

mysql> alter table faculty change column dob DOB date;

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table faculty drop column Salary;

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table faculty add primary key(Faculty\_id);

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table student add primary key(enrollment\_num);

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> delete from student where name = "Anshika Gupta";

Query OK, 0 rows affected (0.00 sec)

mysql> delete from student where enrollment\_num = 59;

Query OK, 1 row affected (0.00 sec)

mysql> create table cs\_dept(id int primary key, fname char(20) unique key, lname char(20), age smallint, dob date not null);

Query OK, 0 rows affected (0.01 sec)

mysql> insert into cs\_dept values(201, "Londyn", "Benson", 22,"2000-09-07");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(202, "Justice", "Boyd", 23,"1999-07-23");

Query OK, 1 row affected (0.01 sec)

mysql> insert into cs\_dept values(203, "Maddison", "Atkins", 22,"2000-09-16");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(204, "Kobe", "Olsen", 24,"1998-07-15");

;

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(205, "Angela", "Juarez", 20,"2001-05-08 ");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(206, "Angel", "Lee", 20,"2001-08-03");

Query OK, 1 row affected (0.01 sec)

mysql> insert into cs\_dept values(207, "Braeden", "Wiley", 22,"1999-05-27");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(208, "Killian", "Calhoun", 20,"2001-03-02");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(209, "Cheyanne", "Fisher", 20,"2001-03-02");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(210, "Bridget", "Horton", 21,"2000-08-29");

Query OK, 1 row affected (0.00 sec)

mysql> insert into cs\_dept values(211, "Leyla", "Webster", 22,"1999-05-16");

Query OK, 1 row affected (0.00 sec)

**Lab 4**

**Theory**

* The REGEXP\_LIKE() function in MySQL is used for pattern matching. It compares whether the given strings match a regular expression or not. It returns 1 if the strings match the regular expression and return 0 if no match is found.

+----------------------------------------------------------------------------+

**CODE:**

mysql> use college;

Database changed

mysql> alter table student drop primary key;

Query OK, 10 rows affected (0.05 sec)

Records: 10 Duplicates: 0 Warnings: 0

mysql> select \* from student;

+----------------+-----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+-----------------+--------+------+--------+

| 34 | Garvit Paliwal | M | 19 | CSE |

| 39 | Gaurav Sharma | M | 18 | CSBS |

| 44 | Aayushi Talreja | F | 18 | EC |

| 47 | Kratik Mathur | M | 18 | ME |

| 67 | Mohit Pherwani | M | 18 | IT |

| 140 | Ram Sharma | M | 19 | CSE |

| 150 | Devansh Singh | M | 18 | ME |

| 155 | Shruti Malviya | F | 18 | CE |

| 159 | Dipika Jain | F | 18 | IT |

| 170 | Sakshi Sharma | F | 19 | AU |

+----------------+-----------------+--------+------+--------+

10 rows in set (0.00 sec)

mysql> alter table student add primary key(name, branch);

Query OK, 0 rows affected (0.05 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | NO | | NULL | |

| name | char(20) | NO | PRI | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | NO | PRI | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.00 sec)

mysql> drop table cs\_dept;

Query OK, 0 rows affected (0.01 sec)

mysql> drop table cs\_dept\_table;

ERROR 1051 (42S02): Unknown table 'college.cs\_dept\_table'

mysql> select enrollment\_num as studentID from student;

+-----------+

| studentID |

+-----------+

| 44 |

| 150 |

| 159 |

| 34 |

| 39 |

| 47 |

| 67 |

| 140 |

| 170 |

| 155 |

+-----------+

10 rows in set (0.00 sec)

mysql> select faculty\_id from faculty as Teachers;

+------------+

| faculty\_id |

+------------+

| 101 |

| 102 |

| 105 |

| 106 |

| 107 |

| 110 |

+------------+

6 rows in set (0.00 sec)

mysql> select Name from faculty as Teachers;

+-----------------+

| Name |

+-----------------+

| Rohit Mahajan |

| Karan Singh |

| Rahul Sharma |

| Monika Jain |

| Sheetal Agrawal |

| Jitendra Sharma |

+-----------------+

6 rows in set (0.00 sec)

mysql> select enrollment\_num as studentID, name as Full\_Name from student;

+-----------+-----------------+

| studentID | Full\_Name |

+-----------+-----------------+

| 44 | Aayushi Talreja |

| 150 | Devansh Singh |

| 159 | Dipika Jain |

| 34 | Garvit Paliwal |

| 39 | Gaurav Sharma |

| 47 | Kratik Mathur |

| 67 | Mohit Pherwani |

| 140 | Ram Sharma |

| 170 | Sakshi Sharma |

| 155 | Shruti Malviya |

+-----------+-----------------+

10 rows in set (0.00 sec)

mysql> select \* from student;

+----------------+-----------------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+-----------------+--------+------+--------+

| 44 | Aayushi Talreja | F | 18 | EC |

| 150 | Devansh Singh | M | 18 | ME |

| 159 | Dipika Jain | F | 18 | IT |

| 34 | Garvit Paliwal | M | 19 | CSE |

| 39 | Gaurav Sharma | M | 18 | CSBS |

| 47 | Kratik Mathur | M | 18 | ME |

| 67 | Mohit Pherwani | M | 18 | IT |

| 140 | Ram Sharma | M | 19 | CSE |

| 170 | Sakshi Sharma | F | 19 | AU |

| 155 | Shruti Malviya | F | 18 | CE |

+----------------+-----------------+--------+------+--------+

10 rows in set (0.00 sec)

mysql> select enrollment\_num, concat(name, " , ", branch) as Student\_Name from student;

+----------------+----------------------+

| enrollment\_num | Student\_Name |

+----------------+----------------------+

| 44 | Aayushi Talreja , EC |

| 150 | Devansh Singh , ME |

| 159 | Dipika Jain , IT |

| 34 | Garvit Paliwal , CSE |

| 39 | Gaurav Sharma , CSBS |

| 47 | Kratik Mathur , ME |

| 67 | Mohit Pherwani , IT |

| 140 | Ram Sharma , CSE |

| 170 | Sakshi Sharma , AU |

| 155 | Shruti Malviya , CE |

+----------------+----------------------+

10 rows in set (0.00 sec)

mysql> select \* from faculty limit 3;

+------------+---------------+--------+------+------+------+-------------+

| Faculty\_id | Name | Gender | Age | DOB | Dept | Year\_joined |

+------------+---------------+--------+------+------+------+-------------+

| 101 | Rohit Mahajan | M | 30 | NULL | ME | 2008 |

| 102 | Karan Singh | F | 34 | NULL | IT | 2012 |

| 105 | Rahul Sharma | M | 51 | NULL | ME | 2010 |

+------------+---------------+--------+------+------+------+-------------+

3 rows in set (0.00 sec)

mysql> select \* from faculty where regexp\_like(name, '[a-g]');

+------------+-----------------+--------+------+------+------+-------------+

| Faculty\_id | Name | Gender | Age | DOB | Dept | Year\_joined |

+------------+-----------------+--------+------+------+------+-------------+

| 101 | Rohit Mahajan | M | 30 | NULL | ME | 2008 |

| 102 | Karan Singh | F | 34 | NULL | IT | 2012 |

| 105 | Rahul Sharma | M | 51 | NULL | ME | 2010 |

| 106 | Monika Jain | F | 25 | NULL | CSE | 2009 |

| 107 | Sheetal Agrawal | F | 40 | NULL | AU | 2009 |

| 110 | Jitendra Sharma | M | 45 | NULL | CE | 2005 |

+------------+-----------------+--------+------+------+------+-------------+

6 rows in set (0.01 sec)

mysql> select \* from faculty where regexp\_like(name, '[y-z]');

Empty set (0.00 sec)

mysql> select \* from faculty where regexp\_like(name, '[u-z]');

+------------+-----------------+--------+------+------+------+-------------+

| Faculty\_id | Name | Gender | Age | DOB | Dept | Year\_joined |

+------------+-----------------+--------+------+------+------+-------------+

| 105 | Rahul Sharma | M | 51 | NULL | ME | 2010 |

| 107 | Sheetal Agrawal | F | 40 | NULL | AU | 2009 |

+------------+-----------------+--------+------+------+------+-------------+

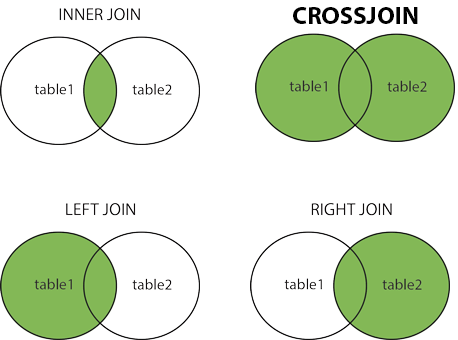
2 rows in set (0.00 sec)

mysql>

**Lab 5**

**Theory**

* The INNER JOIN keyword selects records that have matching values in both tables.
* The LEFT JOIN keyword returns all records from the left table (table1), and the matching records (if any) from the right table (table2).
* The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records (if any) from the left table (table1).
* The CROSS JOIN keyword returns all records from both tables (table1 and table2).



+----------------------------------------------------------------------------+

**CODE:**

mysql> create database company;

Query OK, 1 row affected (0.00 sec)

mysql> use company;

Database changed

mysql> create table customers(customerID int primary key, fname char(20) unique key, lname char(20));

Query OK, 0 rows affected (0.03 sec)

mysql> desc customers;

+------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+----------+------+-----+---------+-------+

| customerID | int | NO | PRI | NULL | |

| fname | char(20) | YES | UNI | NULL | |

| lname | char(20) | YES | | NULL | |

+------------+----------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> insert into customers values(100, "Monika", "Jain");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(101, "Salmon", "Khan");

Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(102, "Udit", "Sathe");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(103, "Utkarsh", "Chourasia");

Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(104, "Muskan", "Jain");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(105, "Tejas", "Shah");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(106, "Sneha", "Verma");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(107, "Vanhika", "Juneja");

Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(108, "Sparsh", "Garg");

Query OK, 1 row affected (0.00 sec)

mysql> insert into customers values(109, "Ujjwal", "Pawar");

Query OK, 1 row affected (0.01 sec)

mysql> insert into customers values(110, "Tanish", "Kohser");

Query OK, 1 row affected (0.00 sec)

mysql> create table orders(customerCODE int primary key, orderID int );

Query OK, 0 rows affected (0.02 sec)

mysql> insert into orders values(100, 1001);

Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(101, 1002);

Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(102, 1003);

Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(103, 1004);

Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(104, 1005);

Query OK, 1 row affected (0.01 sec)

mysql>

mysql> SELECT \* FROM customers INNER JOIN orders ON customers.customerID=orders.customerCODE;

+------------+---------+-----------+--------------+---------+

| customerID | fname | lname | customerCODE | orderID |

+------------+---------+-----------+--------------+---------+

| 100 | Monika | Jain | 100 | 1001 |

| 101 | Salmon | Khan | 101 | 1002 |

| 102 | Udit | Sathe | 102 | 1003 |

| 103 | Utkarsh | Chourasia | 103 | 1004 |

| 104 | Muskan | Jain | 104 | 1005 |

+------------+---------+-----------+--------------+---------+

5 rows in set (0.00 sec)

mysql> SELECT \* FROM customers LEFT JOIN orders ON customers.customerID = orders.customerCODE;

+------------+---------+-----------+--------------+---------+

| customerID | fname | lname | customerCODE | orderID |

+------------+---------+-----------+--------------+---------+

| 100 | Monika | Jain | 100 | 1001 |

| 101 | Salmon | Khan | 101 | 1002 |

| 102 | Udit | Sathe | 102 | 1003 |

| 103 | Utkarsh | Chourasia | 103 | 1004 |

| 104 | Muskan | Jain | 104 | 1005 |

| 105 | Tejas | Shah | NULL | NULL |

| 106 | Sneha | Verma | NULL | NULL |

| 107 | Vanhika | Juneja | NULL | NULL |

| 108 | Sparsh | Garg | NULL | NULL |

| 109 | Ujjwal | Pawar | NULL | NULL |

| 110 | Tanish | Kohser | NULL | NULL |

+------------+---------+-----------+--------------+---------+

11 rows in set (0.00 sec)

mysql> SELECT \* FROM customers RIGHT JOIN orders ON customers.customerID = orders.customerCODE;

+------------+---------+-----------+--------------+---------+

| customerID | fname | lname | customerCODE | orderID |

+------------+---------+-----------+--------------+---------+

| 100 | Monika | Jain | 100 | 1001 |

| 101 | Salmon | Khan | 101 | 1002 |

| 102 | Udit | Sathe | 102 | 1003 |

| 103 | Utkarsh | Chourasia | 103 | 1004 |

| 104 | Muskan | Jain | 104 | 1005 |

+------------+---------+-----------+--------------+---------+

5 rows in set (0.00 sec)

mysql> SELECT \* FROM customers CROSS JOIN orders;

+------------+---------+-----------+--------------+---------+

| customerID | fname | lname | customerCODE | orderID |

+------------+---------+-----------+--------------+---------+

| 100 | Monika | Jain | 104 | 1005 |

| 100 | Monika | Jain | 103 | 1004 |

| 100 | Monika | Jain | 102 | 1003 |

| 100 | Monika | Jain | 101 | 1002 |

| 100 | Monika | Jain | 100 | 1001 |

| 101 | Salmon | Khan | 104 | 1005 |

| 101 | Salmon | Khan | 103 | 1004 |

| 101 | Salmon | Khan | 102 | 1003 |

| 101 | Salmon | Khan | 101 | 1002 |

| 101 | Salmon | Khan | 100 | 1001 |

| 102 | Udit | Sathe | 104 | 1005 |

| 102 | Udit | Sathe | 103 | 1004 |

| 102 | Udit | Sathe | 102 | 1003 |

| 102 | Udit | Sathe | 101 | 1002 |

| 102 | Udit | Sathe | 100 | 1001 |

| 103 | Utkarsh | Chourasia | 104 | 1005 |

| 103 | Utkarsh | Chourasia | 103 | 1004 |

| 103 | Utkarsh | Chourasia | 102 | 1003 |

| 103 | Utkarsh | Chourasia | 101 | 1002 |

| 103 | Utkarsh | Chourasia | 100 | 1001 |

| 104 | Muskan | Jain | 104 | 1005 |

| 104 | Muskan | Jain | 103 | 1004 |

| 104 | Muskan | Jain | 102 | 1003 |

| 104 | Muskan | Jain | 101 | 1002 |

| 104 | Muskan | Jain | 100 | 1001 |

| 105 | Tejas | Shah | 104 | 1005 |

| 105 | Tejas | Shah | 103 | 1004 |

| 105 | Tejas | Shah | 102 | 1003 |

| 105 | Tejas | Shah | 101 | 1002 |

| 105 | Tejas | Shah | 100 | 1001 |

| 106 | Sneha | Verma | 104 | 1005 |

| 106 | Sneha | Verma | 103 | 1004 |

| 106 | Sneha | Verma | 102 | 1003 |

| 106 | Sneha | Verma | 101 | 1002 |

| 106 | Sneha | Verma | 100 | 1001 |

| 107 | Vanhika | Juneja | 104 | 1005 |

| 107 | Vanhika | Juneja | 103 | 1004 |

| 107 | Vanhika | Juneja | 102 | 1003 |

| 107 | Vanhika | Juneja | 101 | 1002 |

| 107 | Vanhika | Juneja | 100 | 1001 |

| 108 | Sparsh | Garg | 104 | 1005 |

| 108 | Sparsh | Garg | 103 | 1004 |

| 108 | Sparsh | Garg | 102 | 1003 |

| 108 | Sparsh | Garg | 101 | 1002 |

| 108 | Sparsh | Garg | 100 | 1001 |

| 109 | Ujjwal | Pawar | 104 | 1005 |

| 109 | Ujjwal | Pawar | 103 | 1004 |

| 109 | Ujjwal | Pawar | 102 | 1003 |

| 109 | Ujjwal | Pawar | 101 | 1002 |

| 109 | Ujjwal | Pawar | 100 | 1001 |

| 110 | Tanish | Kohser | 104 | 1005 |

| 110 | Tanish | Kohser | 103 | 1004 |

| 110 | Tanish | Kohser | 102 | 1003 |

| 110 | Tanish | Kohser | 101 | 1002 |

| 110 | Tanish | Kohser | 100 | 1001 |

+------------+---------+-----------+--------------+---------+

55 rows in set (0.00 sec)

mysql>

**Lab 6**

**Theory**

* In SQL, a view is a virtual table based on the result-set of an SQL statement.
* A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
* You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.
* A view is created with the CREATE VIEW statement.

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.01 sec)

mysql> use college;

Database changed

mysql> create table student(enrollment\_num int, name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.05 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | YES | | NULL | |

| name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.00 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.03 sec)

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

Query OK, 1 row affected (0.03 sec)

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(140, "Ram Sharma", "M", 18, "CSE");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(150, "Devansh Singh", "M", 19, "ME");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(155, "Shruti Malviya", "F", 19, "CE");

Query OK, 1 row affected (0.04 sec)

mysql> insert into student values(159, "Dipika Jain", "F", 19, "IT");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(170, "Sakshi Sharma", "F", 18, "AU");

Query OK, 1 row affected (0.01 sec)

mysql> update student set gender = 'F' where name = 'Anshika Gupta';

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0 Changed: 0 Warnings: 0

mysql> update student set age=19 where enrollment\_num in (34, 140, 59, 170);

Query OK, 4 rows affected (0.00 sec)

Rows matched: 4 Changed: 4 Warnings: 0

mysql> update student set age=18 where enrollment\_num not in (34, 140, 59, 170);

Query OK, 6 rows affected (0.01 sec)

Rows matched: 7 Changed: 6 Warnings: 0

mysql> update student set name = "Diksha Rathore" where name = "Diksha";

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Neha Reddy" where name = "Neha";

Query OK, 0 rows affected (0.00 sec)

Rows matched: 0 Changed: 0 Warnings: 0

mysql> update student set name = "Gaurav Sharma" where name = "Gaurav";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Garvit Paliwal" where name = "Garvit";

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Kratik Mathur" where name = "Kratik";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Aayushi Talreja" where name = "Aayushi";

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update student set name = "Mohit Pherwani" where name = "Mohit";

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> CREATE VIEW twotable AS SELECT student.name, student.branch FROM student;

Query OK, 0 rows affected (0.01 sec)

mysql> SELECT \* FROM twotable;

+-----------------+--------+

| name | branch |

+-----------------+--------+

| Garvit Paliwal | CSE |

| Aayushi Talreja | EC |

| Kratik Mathur | ME |

| Diksha Rathore | AU |

| Gaurav Sharma | CSBS |

| Mohit Pherwani | IT |

| Ram Sharma | CSE |

| Devansh Singh | ME |

| Shruti Malviya | CE |

| Dipika Jain | IT |

| Sakshi Sharma | AU |

+-----------------+--------+

11 rows in set (0.00 sec)

**Lab 7**

**Theory**

* The GROUP BY statement groups rows that have the same values into summary rows.
* The GROUP BY statement is often used with aggregate functions {COUNT(), MAX(), MIN(), SUM(), AVG()} to group the result-set by one or more columns.
* The ORDER BY keyword is used to sort the result-set in ascending or descending order.
* The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.
* The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.
* The UNION operator is used to combine the result-set of two or more SELECT statements.
  + Every SELECT statement within UNION must have the same number of columns
  + The columns must also have similar data types
  + The columns in every SELECT statement must also be in the same order

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.01 sec)

mysql> use college;

to student values(170, "Sakshi Sharma", "F", 18, "AU");Database changed

mysql> create table student(enrollment\_num int, full\_name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.02 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | YES | | NULL | |

| full\_name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.01 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(155, "Shruti Malviya", "F", 19, "CE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(159, "Dipika Jain", "F", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(170, "Sakshi Sharma", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(140, "Ram Sharma", "M", 18, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(150, "Devansh Singh", "M", 19, "ME");

Query OK, 1 row affected (0.01 sec)

mysql> SELECT COUNT(enrollment\_num), full\_name FROM student GROUP BY full\_name;

+-----------------------+----------------+

| COUNT(enrollment\_num) | full\_name |

+-----------------------+----------------+

| 1 | Garvit |

| 1 | Aayushi |

| 1 | Kratik |

| 1 | Diksha |

| 1 | Gaurav |

| 1 | Mohit |

| 2 | Ram Sharma |

| 1 | Shruti Malviya |

| 1 | Dipika Jain |

| 1 | Sakshi Sharma |

| 1 | Devansh Singh |

+-----------------------+----------------+

11 rows in set (0.00 sec)

mysql> SELECT COUNT(enrollment\_num), full\_name FROM student GROUP BY full\_name ORDER BY COUNT(branch) DESC;

+-----------------------+----------------+

| COUNT(enrollment\_num) | full\_name |

+-----------------------+----------------+

| 2 | Ram Sharma |

| 1 | Garvit |

| 1 | Aayushi |

| 1 | Kratik |

| 1 | Diksha |

| 1 | Gaurav |

| 1 | Mohit |

| 1 | Shruti Malviya |

| 1 | Dipika Jain |

| 1 | Sakshi Sharma |

| 1 | Devansh Singh |

+-----------------------+----------------+

11 rows in set (0.00 sec)

mysql> SELECT \* FROM student ORDER BY branch ;

+----------------+-----------+--------+------+--------+

| enrollment\_num | full\_name | gender | age | branch |

+----------------+-----------+--------+------+--------+

| 59 | Diksha | F | 18 | AU |

| 170 | Sakshi | F | 18 | AU |

| 155 | Shruti | F | 19 | CE |

| 39 | Gaurav | M | 18 | CSBS |

| 34 | Garvit | M | 18 | CSE |

| 140 | Ram | M | 18 | CSE |

| 44 | Aayushi | F | 17 | EC |

| 67 | Mohit | M | 19 | IT |

| 159 | Dipika | F | 19 | IT |

| 47 | Kratik | M | 19 | ME |

| 150 | Devansh | M | 19 | ME |

+----------------+-----------+--------+------+--------+

11 rows in set (0.00 sec)

mysql> SELECT \* FROM student;

+----------------+-----------+--------+------+--------+

| enrollment\_num | full\_name | gender | age | branch |

+----------------+-----------+--------+------+--------+

| 34 | Garvit | M | 18 | CSE |

| 44 | Aayushi | F | 17 | EC |

| 47 | Kratik | M | 19 | ME |

| 59 | Diksha | F | 18 | AU |

| 39 | Gaurav | M | 18 | CSBS |

| 67 | Mohit | M | 19 | IT |

| 140 | Ram | M | 18 | CSE |

| 150 | Devansh | M | 19 | ME |

| 155 | Shruti | F | 19 | CE |

| 159 | Dipika | F | 19 | IT |

| 170 | Sakshi | F | 18 | AU |

+----------------+-----------+--------+------+--------+

11 rows in set (0.01 sec)

mysql> SELECT \* FROM student where gender="F" Having age=18;

+----------------+-----------+--------+------+--------+

| enrollment\_num | full\_name | gender | age | branch |

+----------------+-----------+--------+------+--------+

| 59 | Diksha | F | 18 | AU |

| 170 | Sakshi | F | 18 | AU |

+----------------+-----------+--------+------+--------+

2 rows in set (0.00 sec)

mysql> SELECT \* FROM student where gender="F" Having age=18 Union SELECT \* FROM student where gender="M" Having age=18;

+----------------+-----------+--------+------+--------+

| enrollment\_num | full\_name | gender | age | branch |

+----------------+-----------+--------+------+--------+

| 59 | Diksha | F | 18 | AU |

| 170 | Sakshi | F | 18 | AU |

| 34 | Garvit | M | 18 | CSE |

| 39 | Gaurav | M | 18 | CSBS |

| 140 | Ram | M | 18 | CSE |

+----------------+-----------+--------+------+--------+

5 rows in set (0.00 sec)

**Lab 8**

**Theory**

* The EXISTS operator is used to test for the existence of any record in a subquery.
  + The EXISTS operator returns TRUE if the subquery returns one or more records.
* The ANY operator:
  + returns a boolean value as a result
  + returns TRUE if ANY of the subquery values meet the condition
* The ALL operator:
  + returns a boolean value as a result
  + returns TRUE if ALL of the subquery values meet the condition
  + is used with SELECT, WHERE and HAVING statements

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.00 sec)

mysql> use college;

Database changed

mysql> create table student(enrollment\_num int, full\_name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.01 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | YES | | NULL | |

| full\_name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.00 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(140, "Ram", "M", 18, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(150, "Devansh", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(155, "Shruti", "F", 19, "CE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(156, "Shruti", "F", 19, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(159, "Dipika", "F", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(170, "Saksh", "F", 18, "AU");

Query OK, 1 row affected (0.01 sec)

mysql> create table faculty(Faculty\_id mediumint, Name char(30), Gender char(1), Age int, branch char(10), Salary float(5, 2), Year\_joined year);

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> desc faculty;

+-------------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+------------+------+-----+---------+-------+

| Faculty\_id | mediumint | YES | | NULL | |

| Name | char(30) | YES | | NULL | |

| Gender | char(1) | YES | | NULL | |

| Age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

| Salary | float(5,2) | YES | | NULL | |

| Year\_joined | year | YES | | NULL | |

+-------------+------------+------+-----+---------+-------+

7 rows in set (0.00 sec)

mysql> insert into faculty values(101, "Rohit Mahajan", "M", 30, "ME", 344.9,'2008');

Query OK, 1 row affected (0.01 sec)

mysql> insert into faculty values(106, "Monika Jain", "F", 25, "CSE", 548.0,'2009');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(102, "Kiran Talwariya", "F", 34, "IT", 783.0,'2012');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(110, "Jitendra Sharma", "M", 45, "CE", 993.0,'2005');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(107, "Sheetal Agrawal", "F", 40, "AU", 678.0,'2009');

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(105, "Rahul Sharma", "M", 51, "ME", 693.0,'2010');

Query OK, 1 row affected (0.00 sec)

mysql> SELECT full\_name FROM student WHERE EXISTS (SELECT branch FROM faculty WHERE age=25 );

+-----------+

| full\_name |

+-----------+

| Garvit |

| Aayushi |

| Kratik |

| Diksha |

| Gaurav |

| Mohit |

| Ram |

| Devansh |

| Shruti |

| Shruti |

| Dipika |

| Saksh |

+-----------+

12 rows in set (0.00 sec)

mysql> SELECT full\_name FROM student WHERE branch=ANY (SELECT branch FROM faculty WHERE age>38 );

+-----------+

| full\_name |

+-----------+

| Kratik |

| Diksha |

| Devansh |

| Shruti |

| Saksh |

+-----------+

5 rows in set (0.00 sec)

mysql> SELECT full\_name FROM student WHERE branch=ALL (SELECT branch FROM faculty WHERE Gender="F" );

Empty set (0.00 sec)

**Lab 9**

**Theory**

* The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
  + A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.
* The ON DELETE CASCADE clause in MySQL is used to automatically remove the matching records from the child table when we delete the rows from the parent table. It is a kind of referential action related to the foreign key.
* A foreign key with "set null on delete" means that if a record in the parent table is deleted, then the corresponding records in the child table will have the foreign key fields set to NULL. The records in the child table will not be deleted in SQL Server.

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.01 sec)

mysql> use college;

Database changed

mysql> create table student(enrollment\_num int primary key, full\_name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.01 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | NO | PRI | NULL | |

| full\_name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.00 sec)

mysql> insert into student values(101, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(106, "Shruti", "F", 19, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(110, "Ram", "M", 18, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(102, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(107, "Shruti", "F", 19, "EC");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(105, "Devansh", "M", 19, "ME");

Query OK, 1 row affected (0.01 sec)

mysql> SET FOREIGN\_KEY\_CHECKS=0;

Query OK, 0 rows affected (0.00 sec)

mysql> create table faculty(Faculty\_id int primary key, Name char(30), Gender char(1), Age int, branch char(10), Salary float(5, 2), Year\_joined year,student\_en int, FOREIGN KEY (student\_en) REFERENCES student(enrollment\_num) ON DELETE SET NULL);

Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> desc faculty;

+-------------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+------------+------+-----+---------+-------+

| Faculty\_id | int | NO | PRI | NULL | |

| Name | char(30) | YES | | NULL | |

| Gender | char(1) | YES | | NULL | |

| Age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

| Salary | float(5,2) | YES | | NULL | |

| Year\_joined | year | YES | | NULL | |

| student\_en | int | YES | MUL | NULL | |

+-------------+------------+------+-----+---------+-------+

8 rows in set (0.01 sec)

mysql> insert into faculty values(101, "Rohit Mahajan", "M", 30, "ME", 344.9,'2008', 101);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(106, "Monika Jain", "F", 25, "CSE", 548.0,'2009', 106);

Query OK, 1 row affected (0.01 sec)

mysql> insert into faculty values(110, "Jitendra Sharma", "M", 45, "EC", 993.0,'2005', 102);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(107, "Sheetal Agrawal", "F", 40, "AU", 678.0,'2009', 105);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(105, "Rahul Sharma", "M", 51, "ME", 693.0,'2010', 105);

Query OK, 1 row affected (0.01 sec)

mysql> desc faculty;

+-------------+------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+------------+------+-----+---------+-------+

| Faculty\_id | int | NO | PRI | NULL | |

| Name | char(30) | YES | | NULL | |

| Gender | char(1) | YES | | NULL | |

| Age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

| Salary | float(5,2) | YES | | NULL | |

| Year\_joined | year | YES | | NULL | |

| student\_en | int | YES | MUL | NULL | |

+-------------+------------+------+-----+---------+-------+

8 rows in set (0.01 sec)

mysql> insert into faculty values(101, "Rohit Mahajan", "M", 30, "ME", 344.9,'2008', 101);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(106, "Monika Jain", "F", 25, "CSE", 548.0,'2009', 106);

Query OK, 1 row affected (0.01 sec)

mysql> insert into faculty values(110, "Jitendra Sharma", "M", 45, "EC", 993.0,'2005', 102);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(107, "Sheetal Agrawal", "F", 40, "AU", 678.0,'2009', 105);

Query OK, 1 row affected (0.00 sec)

mysql> insert into faculty values(105, "Rahul Sharma", "M", 51, "ME", 693.0,'2010', 105);

Query OK, 1 row affected (0.01 sec)

mysql> DELETE FROM student WHERE enrollment\_num=101;

Query OK, 1 row affected (0.01 sec)

mysql> select \* from student;

+----------------+-----------+--------+------+--------+

| enrollment\_num | full\_name | gender | age | branch |

+----------------+-----------+--------+------+--------+

| 102 | Aayushi | F | 17 | EC |

| 105 | Devansh | M | 19 | ME |

| 106 | Shruti | F | 19 | CSE |

| 107 | Shruti | F | 19 | EC |

| 110 | Ram | M | 18 | CSE |

+----------------+-----------+--------+------+--------+

5 rows in set (0.00 sec)

mysql> select \* from faculty;

+------------+-----------------+--------+------+--------+--------+-------------+------------+

| Faculty\_id | Name | Gender | Age | branch | Salary | Year\_joined | student\_en |

+------------+-----------------+--------+------+--------+--------+-------------+------------+

| 101 | Rohit Mahajan | M | 30 | ME | 344.90 | 2008 | 101 |

| 105 | Rahul Sharma | M | 51 | ME | 693.00 | 2010 | 105 |

| 106 | Monika Jain | F | 25 | CSE | 548.00 | 2009 | 106 |

| 107 | Sheetal Agrawal | F | 40 | AU | 678.00 | 2009 | 105 |

| 110 | Jitendra Sharma | M | 45 | EC | 993.00 | 2005 | 102 |

+------------+-----------------+--------+------+--------+--------+-------------+------------+

5 rows in set (0.00 sec)

**Lab 10**

**Theory**

* Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.
* SQL constraints are used to specify rules for the data in a table.
* Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
* The CASE statement goes through conditions and returns a value when the first condition is met (like an if-then-else statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.
* If there is no ELSE part and no conditions are true, it returns NULL.

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.01 sec)

mysql> use college;

Database changed

mysql> create table student(enrollment\_num int, name char(20), gender char(1), age int, branch char(10),CONSTRAINT CHK\_age CHECK (age>=18), CONSTRAINT CHK\_gender CHECK (gender="F"));

Query OK, 0 rows affected (0.03 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

ERROR 3819 (HY000): Check constraint 'CHK\_age' is violated.

mysql> insert into student values(44, "Salman", "M", 19, "EC");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(140, "Ram", "M", 18, "CSE");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(150, "Devansh", "M", 19, "ME");

ERROR 3819 (HY000): Check constraint 'CHK\_gender' is violated.

mysql> insert into student values(155, "Shruti", "F", 19, "CE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(159, "Dipika", "F", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(170, "Sakshi", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> ALTER TABLE student DROP CONSTRAINT CHK\_age;

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.01 sec)

mysql> select \* from student;

+----------------+---------+--------+------+--------+

| enrollment\_num | name | gender | age | branch |

+----------------+---------+--------+------+--------+

| 59 | Diksha | F | 18 | AU |

| 155 | Shruti | F | 19 | CE |

| 159 | Dipika | F | 19 | IT |

| 170 | Sakshi | F | 18 | AU |

| 44 | Aayushi | F | 17 | EC |

+----------------+---------+--------+------+--------+

5 rows in set (0.00 sec)

mysql> create table StudentBasic(name char(20), age int);

Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO StudentBasic(name, age) SELECT name, age FROM student;

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from StudentBasic;

+---------+------+

| name | age |

+---------+------+

| Diksha | 18 |

| Shruti | 19 |

| Dipika | 19 |

| Sakshi | 18 |

| Aayushi | 17 |

+---------+------+

5 rows in set (0.00 sec)

mysql> SELECT name, age, branch

-> FROM student

-> ORDER BY

-> (CASE

-> WHEN name THEN age

-> ELSE branch

-> END);

+---------+------+--------+

| name | age | branch |

+---------+------+--------+

| Diksha | 18 | AU |

| Sakshi | 18 | AU |

| Shruti | 19 | CE |

| Aayushi | 17 | EC |

| Dipika | 19 | IT |

+---------+------+--------+

5 rows in set, 5 warnings (0.00 sec)

**Lab 11**

**Theory**

* The stored procedure is SQL statements wrapped within the CREATE PROCEDURE statement. The stored procedure may contain a conditional statement like IF or CASE or the Loops. The stored procedure can also execute another stored procedure or a function that modularizes the code.
* **IN parameter:**It is the default mode. It takes a parameter as input, such as an attribute. When we define it, the calling program has to pass an argument to the stored procedure. This parameter's value is always protected.
* **OUT parameters:** It is used to pass a parameter as output. Its value can be changed inside the stored procedure, and the changed (new) value is passed back to the calling program. It is noted that a procedure cannot access the OUT parameter's initial value when it starts.
* **INOUT parameters:** It is a combination of IN and OUT parameters. It means the calling program can pass the argument, and the procedure can modify the INOUT parameter, and then passes the new value back to the calling program.
* you define a **DELIMITER** to tell the mysql client to treat the statements, functions, stored procedures or triggers as an entire statement.

+----------------------------------------------------------------------------+

**CODE:**

mysql> create database college;

Query OK, 1 row affected (0.01 sec)

mysql> use college;

Database changed

mysql>

mysql> create table student(enrollment\_num int, name char(20), gender char(1), age int, branch char(10));

Query OK, 0 rows affected (0.03 sec)

mysql> desc student;

+----------------+----------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------------+----------+------+-----+---------+-------+

| enrollment\_num | int | YES | | NULL | |

| name | char(20) | YES | | NULL | |

| gender | char(1) | YES | | NULL | |

| age | int | YES | | NULL | |

| branch | char(10) | YES | | NULL | |

+----------------+----------+------+-----+---------+-------+

5 rows in set (0.01 sec)

mysql> insert into student values(34, "Garvit", "M", 18, "CSE");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(44, "Aayushi", "F", 17, "EC");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(47, "Kratik", "M", 19, "ME");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(59, "Diksha", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(39, "Gaurav", "M", 18, "CSBS");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(67, "Mohit", "M", 19, "IT");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(140, "Ram", "M", 18, "CSE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(150, "Devansh", "M", 19, "ME");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(155, "Shruti", "F", 19, "CE");

Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(159, "Dipika", "F", 19, "IT");

Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(170, "Sakshi", "F", 18, "AU");

Query OK, 1 row affected (0.00 sec)

mysql> DELIMITER $$

mysql> CREATE PROCEDURE getName()

-> BEGIN

-> SELECT name FROM student WHERE age>18;

-> END $$

Query OK, 0 rows affected (0.01 sec)

mysql> DELIMITER ;

mysql> DELIMITER &&

mysql> CREATE PROCEDURE getAge (IN var1 INT)

-> BEGIN

-> SELECT name, age FROM student LIMIT var1;

-> END &&

Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;

mysql> CALL getName();

+---------+

| name |

+---------+

| Kratik |

| Mohit |

| Devansh |

| Shruti |

| Dipika |

+---------+

5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL getAge(5);

+---------+------+

| name | age |

+---------+------+

| Garvit | 18 |

| Aayushi | 17 |

| Kratik | 19 |

| Diksha | 18 |

| Gaurav | 18 |

+---------+------+

5 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)