EXPERIMENT NO : 5

AIM: Data retrieval from the already created database/ Create new sample database and necessary adding of data are made then perform the query selections. (Perform nested query selection using with comparison operators and Logical connectives)

RESULT:

```
sql> select*from student_mark;
+-------+-----------+------------------+------------------+
| Reg_No | Course_ID | Student_Internal | Student_External |
+-------+-----------+------------------+------------------+
|  4312 | C32       |               19 |               77 |
|  4512 | C12       |               30 |               55 |
|  5012 | C12       |               38 |               59 |
+-------+-----------+------------------+------------------+
rows in set (0.00 sec)

sql> select*from student_mark
  -> where Student_External >
  -> (select AVG(Student_External) from student_mark);
+-------+-----------+------------------+------------------+
| Reg_No | Course_ID | Student_Internal | Student_External |
+-------+-----------+------------------+------------------+
|  4312 | C32       |               19 |               77 |
+-------+-----------+------------------+------------------+
row in set (0.00 sec)
```

```
sql> select * from course;
+----------+-------------+--------+----------+---------------+---------------+-------------+
| Course_ID | Course_Name | Credit | Semester | Internal_mark | External_mark | Course_Type |
+----------+-------------+--------+----------+---------------+---------------+-------------+
| C12      | MCA         |    100 |        4 |            40 |            60 | Regular     |
| C17      | BCA         |     20 |        6 |            20 |            80 | Regular     |
| C32      | BTECH       |     50 |        8 |            20 |            80 | Regular     |
+----------+-------------+--------+----------+---------------+---------------+-------------+
rows in set (0.01 sec)

sql> select min(Student_Internal)
  -> from student_mark
  -> where Course_ID in (
  -> select Course_ID
  -> from course
  -> where Course_Name='MCA' or Course_Name='BCA');
+----------------------+
| min(Student_Internal) |
+----------------------+
|                   30 |
+----------------------+
row in set (0.00 sec)
```

EXPERIMENT NO : 6

AIM: Data retrieval from the already created database/ Create new sample database and necessary adding of data are made then perform the query selections. (Write queries that familiarize all string operations in SQL.)

RESULT:

```
sql> select*from course;
+----------+-------------+--------+----------+---------------+---------------+-------------+
| Course_ID | Course_Name | Credit | Semester | Internal_mark | External_mark | Course_Type |
+----------+-------------+--------+----------+---------------+---------------+-------------+
| C12       | MCA         |    100 |        4 |            40 |            60 | Regular     |
| C17       | BCA         |     20 |        6 |            20 |            80 | Regular     |
| C32       | BTECH       |     50 |        8 |            20 |            80 | Regular     |
+----------+-------------+--------+----------+---------------+---------------+-------------+
rows in set (0.00 sec)

sql> select Course_Name,length(Course_name) as length
  -> from course;
+------------+--------+
| Course_Name | length |
+------------+--------+
| MCA         |      3 |
| BCA         |      3 |
| BTECH       |      5 |
+------------+--------+
rows in set (0.00 sec)

sql> select Course_Name,locate('C',Course_Name) as Location_of_C
  -> from course;
+------------+---------------+
| Course_Name | Location_of_C |
+------------+---------------+
| MCA         |             2 |
| BCA         |             2 |
| BTECH       |             4 |
+------------+---------------+
rows in set (0.00 sec)
```

```
sql> select lower(Course_Name) as Course_Name
  -> from course;
+-------------+
| Course_Name |
+-------------+
| mca         |
| bca         |
| btech       |
+-------------+
rows in set (0.01 sec)

sql> select replace(Course_Name,'BTECH','BTech') as Course_Name
  -> from course;
+-------------+
| Course_Name |
+-------------+
| MCA         |
| BCA         |
| BTech       |
+-------------+
rows in set (0.00 sec)

sql> select Course_Name,reverse(Course_Name) as reverse
  -> from course;
+-------------+----------+
| Course_Name | reverse  |
+-------------+----------+
| MCA         | ACM      |
| BCA         | ACB      |
| BTECH       | HCETB    |
+-------------+----------+
rows in set (0.00 sec)
```

EXPERIMENT NO : 7

AIM: Data retrieval from the already created database/ Create new sample database and necessary adding of data are made then perform the query selections. (Write sample queries that familiarize all aggregate functions, group by and having clauses in SQL)

RESULT:

```
sql> select*from course;
+-----------+-------------+--------+----------+---------------+---------------+-------------+
| Course_ID | Course_Name | Credit | Semester | Internal_mark | External_mark | Course_Type |
+-----------+-------------+--------+----------+---------------+---------------+-------------+
| C12       | MCA         |    100 |        4 |            40 |            60 | Regular     |
| C17       | BCA         |     20 |        6 |            20 |            80 | Regular     |
| C32       | BTECH       |     50 |        8 |            20 |            80 | Regular     |
+-----------+-------------+--------+----------+---------------+---------------+-------------+
rows in set (0.00 sec)

sql> select count(Course_Name) as Courses_Available from course;
+------------------+
| Courses_Available |
+------------------+
|                3 |
+------------------+
row in set (0.00 sec)

sql> select*from course limit 1;
+-----------+-------------+--------+----------+---------------+---------------+-------------+
| Course_ID | Course_Name | Credit | Semester | Internal_mark | External_mark | Course_Type |
+-----------+-------------+--------+----------+---------------+---------------+-------------+
| C12       | MCA         |    100 |        4 |            40 |            60 | Regular     |
+-----------+-------------+--------+----------+---------------+---------------+-------------+
row in set (0.00 sec)
```

-

```
ysql> select Course_ID,count(Course_ID)
  -> from student_mark
  -> group by Course_ID;
+----------+------------------+
| Course_ID | count(Course_ID) |
+----------+------------------+
| C12       |                2 |
| C32       |                1 |
+----------+------------------+
rows in set (0.00 sec)

ysql> select Course_ID,count(Course_ID)
  -> from student_mark
  -> group by Course_ID
  -> having count(Course_ID) > 1;
+----------+------------------+
| Course_ID | count(Course_ID) |
+----------+------------------+
| C12       |                2 |
+----------+------------------+
row in set (0.00 sec)
```

```
ysql> select max(External_mark) from course;
+--------------------+
| max(External_mark) |
+--------------------+
|                 80 |
+--------------------+
row in set (0.00 sec)

ysql> select min(External_mark) from course;
+--------------------+
| min(External_mark) |
+--------------------+
|                 60 |
+--------------------+
row in set (0.00 sec)

ysql> select avg(External_mark) from course;
+--------------------+
| avg(External_mark) |
+--------------------+
|            73.3333 |
+--------------------+
row in set (0.00 sec)

ysql> select sum(External_mark) from course;
+--------------------+
| sum(External_mark) |
+--------------------+
|                220 |
+--------------------+
row in set (0.00 sec)
```

EXPERIMENT NO : 8

AIM: Data retrieval from the already created database/ Create new sample database and necessary adding of data are made then perform the query selections. (Write sample queries that familiarize all set operations in SQL)

RESULT:

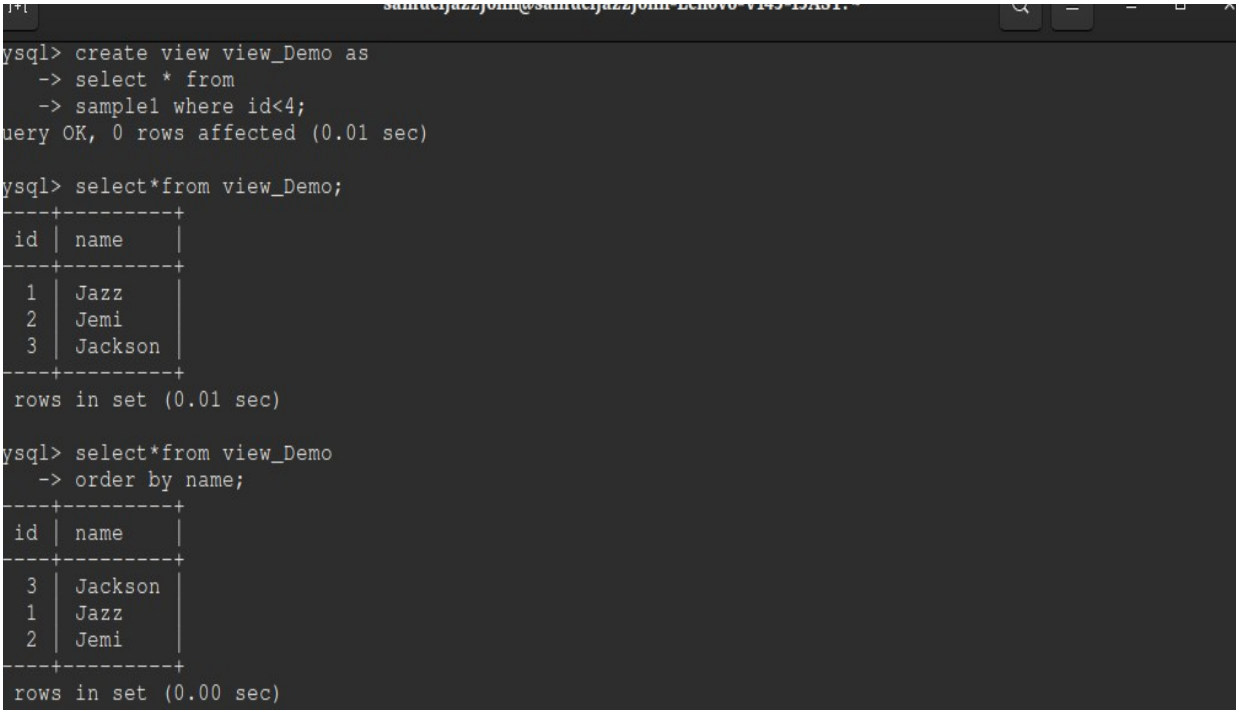
```
sql> select*from sample1;
+---+---------+
| id | name    |
+---+---------+
|  1 | Jazz    |
|  2 | Jemi    |
|  3 | Jackson |
+---+---------+
rows in set (0.00 sec)

sql> select*from sample2;
+---+----------+
| id | name     |
+---+----------+
|  2 | Jemi     |
|  4 | Jennifer |
+---+----------+
rows in set (0.00 sec)
```

```
ysql> select * from sample1
   -> union
   -> select * from sample2;
+----+----------+
| id | name     |
+----+----------+
|  1 | Jazz     |
|  2 | Jemi     |
|  3 | Jackson  |
|  4 | Jennifer |
+----+----------+
rows in set (0.01 sec)

ysql> select*from sample1
   -> union all
   -> select*from sample2;
+----+----------+
| id | name     |
+----+----------+
|  1 | Jazz     |
|  2 | Jemi     |
|  3 | Jackson  |
|  2 | Jemi     |
|  4 | Jennifer |
+----+----------+
rows in set (0.00 sec)
```

EXPERIMENT NO : 9

AIM: Define a view on the already created database and perform query selection on it(Create sample view and write sample queries on it)

RESULT:



EXPERIMENT NO : 10

AIM: Develop a tiny database system and do necessary adding of data and data retrieval from that (Create sample database systems such as Department Library system, College canteen system, Hostel system, College store system etc.)

RESULT:

Department Library System:

```
sql> create database Library_Management;
ery OK, 1 row affected (0.01 sec)

sql> show databases;
+--------------------+
| Database           |
+--------------------+
| Library_Management |
| db_mca             |
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
| sample             |
| sys                |
| wordpress          |
+--------------------+
rows in set (0.00 sec)
```

Tables:

Department

```
sql> create table department(
  -> department_id int(10) primary key,
  -> hod_name varchar(20) not null,
  -> contact_no int(10) not null)
  -> ;
ery OK, 0 rows affected, 2 warnings (0.03 sec)

sql> desc department;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| department_id | int       | NO   | PRI | NULL    |       |
| hod_name    | varchar(20) | NO   |     | NULL    |       |
| contact_no  | int         | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
rows in set (0.02 sec)
```

Student:

```
sql> create table student(
  -> student_id int(10) primary key,
  -> student_name varchar(20),
  -> student_address varchar(100) not null,
  -> registration_date date not null);
ery OK, 0 rows affected, 1 warning (0.03 sec)

sql> desc student;
+-----------------+--------------+------+-----+---------+-------+
Field             | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
student_id        | int          | NO   | PRI | NULL    |       |
student_name      | varchar(20)  | YES  |     | NULL    |       |
student_address   | varchar(100) | NO   |     | NULL    |       |
registration_date | date         | NO   |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
rows in set (0.00 sec)
```

book

```
sql> create table book(
  -> book_id int(20) not null primary key,
  -> book_title varchar(50) not null,
  -> category varchar(20) not null,
  -> rental_price int(10) not null,
  -> status varchar(20),
  -> auther varchar(20) not null,
  -> publisher varchar(20) not null);
ery OK, 0 rows affected, 2 warnings (0.03 sec)

sql> desc book;
+-------------+-------------+------+-----+---------+-------+
Field         | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
book_id       | int         | NO   | PRI | NULL    |       |
book_title    | varchar(50) | NO   |     | NULL    |       |
category      | varchar(20) | NO   |     | NULL    |       |
rental_price  | int         | NO   |     | NULL    |       |
status        | varchar(20) | YES  |     | NULL    |       |
auther        | varchar(20) | NO   |     | NULL    |       |
publisher     | varchar(20) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
rows in set (0.01 sec)
```

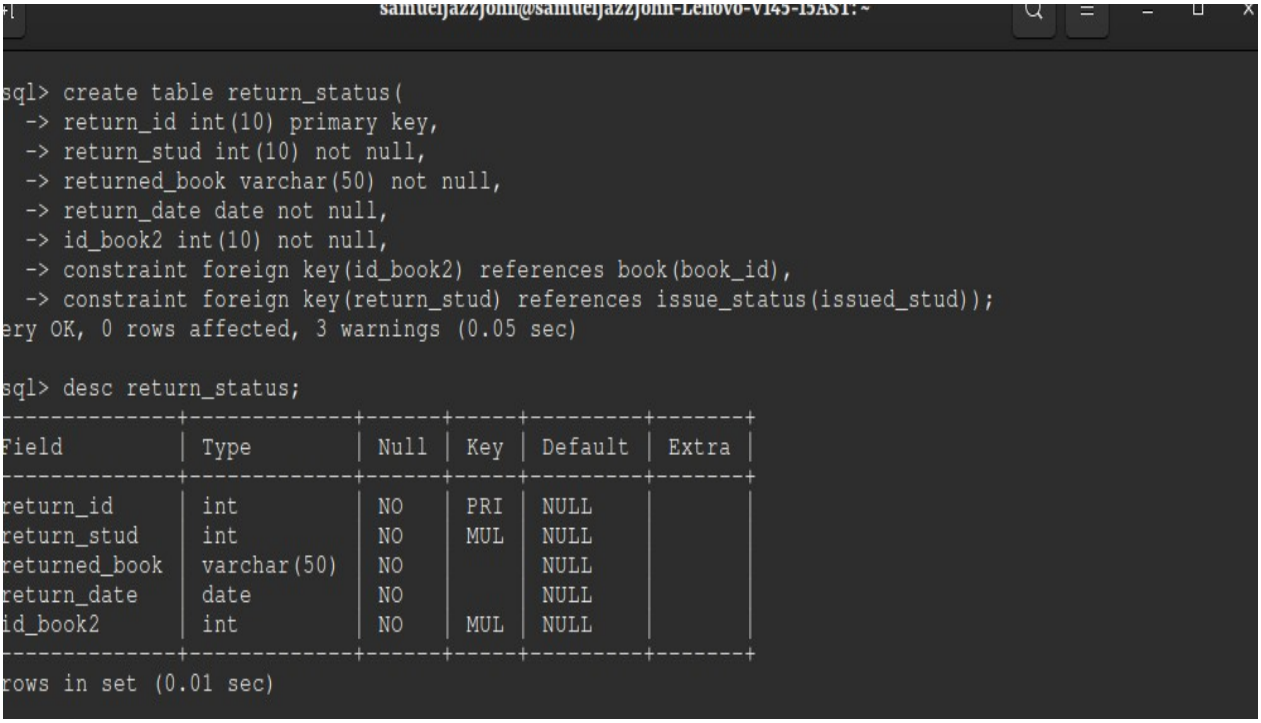
employee

```
sql> create table employee(
  -> employee_id int(10) not null primary key,
  -> employee_name varchar(20) not null,
  -> salary int(10) not null,
  -> position varchar(20) not null);
ery OK, 0 rows affected, 2 warnings (0.03 sec)

sql> desc employee;
+-------------+-------------+------+-----+---------+-------+
Field         | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
employee_id   | int         | NO   | PRI | NULL    |       |
employee_name | varchar(20) | NO   |     | NULL    |       |
salary        | int         | NO   |     | NULL    |       |
position      | varchar(20) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
rows in set (0.00 sec)
```

issue_status

```
sql> create table issue_status(
  -> issue_id int(10) primary key,
  -> issued_stud int(10) not null,
  -> issued_book_name varchar(50) not null,
  -> issue_date date not null,
  -> id_book int(10) not null,
  -> constraint foreign key(id_book) references book(book_id),
  -> constraint foreign key(issued_stud) references student(student_id));
ery OK, 0 rows affected, 3 warnings (0.04 sec)

sql> desc issue_status
  -> ;
+-----------------+-------------+------+-----+---------+-------+
Field             | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
issue_id          | int         | NO   | PRI | NULL    |       |
issued_stud       | int         | NO   | MUL | NULL    |       |
issued_book_name  | varchar(50) | NO   |     | NULL    |       |
issue_date        | date        | NO   |     | NULL    |       |
id_book           | int         | NO   | MUL | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
rows in set (0.01 sec)
```

return_status



```
sql> create table return_status(
  -> return_id int(10) primary key,
  -> return_stud int(10) not null,
  -> returned_book varchar(50) not null,
  -> return_date date not null,
  -> id_book2 int(10) not null,
  -> constraint foreign key(id_book2) references book(book_id),
  -> constraint foreign key(return_stud) references issue_status(issued_stud));
ery OK, 0 rows affected, 3 warnings (0.05 sec)

sql> desc return_status;
+-------------+-------------+------+-----+---------+-------+
Field         | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
return_id     | int         | NO   | PRI | NULL    |       |
return_stud   | int         | NO   | MUL | NULL    |       |
returned_book | varchar(50) | NO   |     | NULL    |       |
return_date   | date        | NO   |     | NULL    |       |
id_book2      | int         | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
rows in set (0.01 sec)
```