Name: Atharva shevate
Srn no: 202201727
Roll no: 02(E1)

BTECCE21405: Database Management Systems Lab

Assignment 2
Joins, Sets, Subqueries
I.

Consider the given database schema:

Student (studentid , studentname, instructorid, studentcity)
Instructor (instructorid, Instructorname, instructorcity, specialization)

```
mysql> desc instructorinfo;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| id             | int         | NO   | PRI | NULL    | auto_increment |
| instructorname | varchar(20) | YES  |     | NULL    |                |
| instructorcity | varchar(10) | YES  |     | NULL    |                |
| specialization | varchar(20) | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

```
mysql> desc studentinfo;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| studentid    | int         | NO   | PRI | NULL    | auto_increment |
| studentname  | varchar(20) | YES  |     | NULL    |                |
| instructorid | int         | YES  | MUL | NULL    |                |
| studentcity  | varchar(10) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql> select * from instructorinfo;
+--------------+----------------+----------------+----------------+
| instructorid | instructorname | instructorcity | specialization |
+--------------+----------------+----------------+----------------+
|          101 | ABC            | Pune           | CGG            |
|          102 | PQR            | Mumbai         | DBMS           |
|          103 | XYZ            | Pune           | Java           |
|          104 | LMN            | Ahmednagar     | TOC            |
|          105 | EFG            | Delhi          | ITSSPS         |
+--------------+----------------+----------------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> select * from studentinfo;
+-----------+--------------------+--------------+-------------+
| studentid | studentname        | instructorid | studentcity |
+-----------+--------------------+--------------+-------------+
|         1 | Anay Somani        |          101 | Pune        |
|         7 | Pranav Dagay       |          101 | Hyderabad   |
|         8 | Harsh Aru          |          105 | Aru Nagar   |
|         9 | Aarohi Samdani     |          102 | Nagpur      |
|        10 | Ayush Jain         |          103 | Pune        |
|        11 | Abhishek Bhagawati |          104 | Pimpri      |
+-----------+--------------------+--------------+-------------+
6 rows in set (0.00 sec)
```

Use all types of Joins and set operation:

1. Add primary and foreign keys
2. Find the instructor of each student.

```
mysql> select studentid,studentname,instructorinfo.instructorid,instructorname from studentinfo,instructorinfo
    -> where studentinfo.instructorid=instructorinfo.instructorid;
+-----------+--------------------+--------------+----------------+
| studentid | studentname        | instructorid | instructorname |
+-----------+--------------------+--------------+----------------+
|         1 | Anay Somani        |          101 | ABC            |
|         7 | Pranav Dagay       |          101 | ABC            |
|         9 | Aarohi Samdani     |          102 | PQR            |
|        10 | Ayush Jain         |          103 | XYZ            |
|        11 | Abhishek Bhagawati |          104 | LMN            |
|         8 | Harsh Aru          |          105 | EFG            |
+-----------+--------------------+--------------+----------------+
6 rows in set (0.00 sec)
```

2. Find the student who is not having any instructor.

```
mysql> select * from studentinfo where instructorid=NULL;
Empty set (0.01 sec)
```

3. Find the student who is not having any instructor as well as instructor who is not having student.

```
mysql> select * from studentinfo,instructorinfo where (studentinfo.instructorid=NULL);
Empty set (0.00 sec)
```

4. Find the students whose instructor’s specialization is computer.

```
mysql> select studentid,studentname,specialization from studentinfo,instructorinfo
    -> where (specialization='CGG')
    -> and
    -> (studentinfo.instructorid=instructorinfo.id);
+-----------+--------------+----------------+
| studentid | studentname  | specialization |
+-----------+--------------+----------------+
|         1 | Anay Somani  | CGG            |
|         7 | Pranav Dagay | CGG            |
+-----------+--------------+----------------+
2 rows in set (0.00 sec)
```

5. Create a view containing total number of students whose instructor belongs to “Pune**”.**

```
mysql> create view view2 as
    -> select studentid,studentname,instructorid,instructorname,instructorcity
    -> from studentinfo,instructorinfo
    -> where (instructorcity='Pune')
    -> and
    -> (studentinfo.instructorid=instructorinfo.id);
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select COUNT(studentname) from view2;
+---------------------+
| COUNT(studentname) |
+---------------------+
|                   3 |
+---------------------+
1 row in set (0.00 sec)
```

II. Consider following database. Execute each query given using join and subqueries.

CREATE TABLE departments (department_id INT (11) AUTO_INCREMENT PRIMARY KEY, department_name VARCHAR (30) NOT NULL, location_id INT (11) DEFAULT NULL,
);

CREATE TABLE employees (employee_id INT (11) AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR (20) DEFAULT NULL, last_name VARCHAR (25) NOT NULL, email VARCHAR (100) NOT NULL, phone_number VARCHAR (20) DEFAULT NULL, hire_date DATE NOT NULL, job_id INT (11) NOT NULL, salary

```
DECIMAL (8, 2) NOT NULL,    manager_id INT (11) DEFAULT NULL,   department_id INT
(11) DEFAULT NULL,
        FOREIGN KEY (department_id) REFERENCES departments (department_id) ON
DELETE CASCADE ON UPDATE CASCADE,
            FOREIGN KEY (manager_id) REFERENCES employees (employee_id)
);
```

```
mysql> desc departments;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| department_id   | int         | NO   | PRI | NULL    | auto_increment |
| department_name | varchar(30) | NO   |     | NULL    |                |
| location_id     | int         | YES  |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> desc employees;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| employee_id   | int         | NO   | PRI | NULL    | auto_increment |
| first_name    | varchar(20) | YES  |     | NULL    |                |
| last_name     | varchar(25) | NO   |     | NULL    |                |
| email         | varchar(100)| NO   |     | NULL    |                |
| phone_number  | varchar(20) | YES  |     | NULL    |                |
| hire_date     | date        | NO   |     | NULL    |                |
| job_id        | int         | NO   |     | NULL    |                |
| salary        | decimal(8,2)| NO   |     | NULL    |                |
| manager_id    | int         | YES  | MUL | NULL    |                |
| department_id | int         | YES  | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)

mysql> select * from departments;
+---------------+-----------------+-------------+
| department_id | department_name | location_id |
+---------------+-----------------+-------------+
|           101 | HR              |        1600 |
|           102 | Sales           |        1700 |
|           103 | Billing         |        1500 |
|           104 | Marketing       |        1700 |
|           105 | Advertising     |        1900 |
+---------------+-----------------+-------------+
5 rows in set (0.00 sec)

mysql> select * from employees;
+-------------+------------+-----------+-------------+--------------+------------+--------+----------+------------+---------------+
| employee_id | first_name | last_name | email       | phone_number | hire_date  | job_id | salary   | manager_id | department_id |
+-------------+------------+-----------+-------------+--------------+------------+--------+----------+------------+---------------+
|        1001 | Anay       | Somani    | as@vu.com   | 8340984345   | 2003-06-13 |    501 | 10000.00 |       1001 |           102 |
|        1003 | Pranav     | Dagay     | pd@vu.com   | 8340324345   | 2005-09-23 |    104 | 12000.00 |       1001 |           105 |
|        1004 | Harsh      | Aru       | ha@vu.com   | 8340324535   | 2006-05-13 |    210 |  8000.00 |       1001 |           105 |
|        1005 | Ayush      | Jain      | aj@vu.com   | 8340234535   | 2002-01-31 |    110 | 11000.00 |       1001 |           102 |
|        1006 | Abhishek   | Bhagawati | ab@vu.com   | 9847382344   | 2004-05-27 |    150 | 11000.00 |       1001 |           101 |
+-------------+------------+-----------+-------------+--------------+------------+--------+----------+------------+---------------+
5 rows in set (0.00 sec)
```

1. Find all employees who locate in the location with the id 1700

```
mysql> select employee_id,first_name,last_name,departments.department_id,department_name from employees,departments where (location_id=1700)
    -> and (employees.department_id=departments.department_id);
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|        1001 | Anay       | Somani    |           102 | Sales           |
|        1005 | Ayush      | Jain      |           102 | Sales           |
+-------------+------------+-----------+---------------+-----------------+
2 rows in set (0.00 sec)
```

2. Find all employees who do not locate at the location 1700

```
mysql> select employee_id,first_name,last_name,departments.department_id,department_name fro
m employees,departments where (location_id!=1700)
    -> and (employees.department_id=departments.department_id);
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|        1006 | Abhishek   | Bhagawati |           101 | HR              |
|        1003 | Pranav     | Dagay     |           105 | Advertising     |
|        1004 | Harsh      | Aru       |           105 | Advertising     |
+-------------+------------+-----------+---------------+-----------------+
3 rows in set (0.00 sec)
```

3. Finds the employees who have the highest salary

```
mysql> select * from employees order by salary desc;
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
| employee_id | first_name | last_name | email     | phone_number | hire_date  | job_id | salary   | manager_id | department_id |
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
|        1003 | Pranav     | Dagay     | pd@vu.com | 8340324345   | 2005-09-23 |    104 | 12000.00 |       1001 |           105 |
|        1005 | Ayush      | Jain      | aj@vu.com | 8340234535   | 2002-01-31 |    110 | 11000.00 |       1001 |           102 |
|        1006 | Abhishek   | Bhagawati | ab@vu.com | 9847382344   | 2004-05-27 |    150 | 11000.00 |       1001 |           101 |
|        1001 | Anay       | Somani    | as@vu.com | 8340984345   | 2003-06-13 |    501 | 10000.00 |       1001 |           102 |
|        1004 | Harsh      | Aru       | ha@vu.com | 8340324535   | 2006-05-13 |    210 |  8000.00 |       1001 |           105 |
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
5 rows in set (0.00 sec)
```

4. Finds all employees who salaries are greater than the average salary of all employees

```
mysql> select * from employees where salary>(select avg(salary) from employees);
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
| employee_id | first_name | last_name | email     | phone_number | hire_date  | job_id | salary   | manager_id | department_id |
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
|        1003 | Pranav     | Dagay     | pd@vu.com | 8340324345   | 2005-09-23 |    104 | 12000.00 |       1001 |           105 |
|        1005 | Ayush      | Jain      | aj@vu.com | 8340234535   | 2002-01-31 |    110 | 11000.00 |       1001 |           102 |
|        1006 | Abhishek   | Bhagawati | ab@vu.com | 9847382344   | 2004-05-27 |    150 | 11000.00 |       1001 |           101 |
+-------------+------------+-----------+-----------+--------------+------------+--------+----------+------------+---------------+
3 rows in set (0.00 sec)
```

5. Finds all departments which have at least one employee with the salary is greater than 10,000

```
mysql> select departments.department_id,department_name,location_id from departments,employees
    -> where (salary>10000)
    -> and
    -> (departments.department_id=employees.department_id);
+---------------+-----------------+-------------+
| department_id | department_name | location_id |
+---------------+-----------------+-------------+
|           105 | Advertising     |        1900 |
|           102 | Sales           |        1700 |
|           101 | HR              |        1600 |
+---------------+-----------------+-------------+
3 rows in set (0.00 sec)
```

6. Finds all departments that do not have any employee with the salary greater than 10,000

```
mysql> SELECT department_name,department_id from departments
    -> where NOT EXISTS(select 1 from employees where salary>10000
    -> and employees.department_id=departments.department_id)
    -> order by department_name;
+-----------------+---------------+
| department_name | department_id |
+-----------------+---------------+
| Billing         |           103 |
| Marketing       |           104 |
+-----------------+---------------+
2 rows in set (0.00 sec)
```

7. Finds all employees whose salaries are greater than the lowest salary of every department

```
mysql> select employee_id,first_name,last_name,salary,department_name from employees,departments
    -> where salary >=all(select min(salary) from employees group by department_id)
    -> and employees.department_id=departments.department_id;
+-------------+------------+-----------+----------+-----------------+
| employee_id | first_name | last_name | salary   | department_name |
+-------------+------------+-----------+----------+-----------------+
|        1003 | Pranav     | Dagay     | 12000.00 | Advertising     |
|        1005 | Ayush      | Jain      | 11000.00 | Sales           |
|        1006 | Abhishek   | Bhagawati | 11000.00 | HR              |
+-------------+------------+-----------+----------+-----------------+
3 rows in set (0.00 sec)
```

8. Finds all employees whose salaries are greater than or equal to the highest salary of every department

```
mysql> select employee_id,first_name,last_name,salary,department_name from employees,departm
ents
    -> where salary >=some(select max(salary) from employees group by department_id)
    -> and employees.department_id=departments.department_id;
+-------------+------------+-----------+----------+-----------------+
| employee_id | first_name | last_name | salary   | department_name |
+-------------+------------+-----------+----------+-----------------+
|        1003 | Pranav     | Dagay     | 12000.00 | Advertising     |
|        1005 | Ayush      | Jain      | 11000.00 | Sales           |
|        1006 | Abhishek   | Bhagawati | 11000.00 | HR              |
+-------------+------------+-----------+----------+-----------------+
3 rows in set (0.00 sec)
```

9. Finds the salaries of all employees, their average salary, and the difference between the salary of each employee and the average salary

```
)difference from employees' at line 1
mysql> select employee_id,first_name,last_name,salary,(select round(avg(salary),0) from empl
oyees) avg_salary,salary-(select round(avg(salary),0) from employees
    -> )difference from employees ;
+-------------+------------+-----------+----------+------------+------------+
| employee_id | first_name | last_name | salary   | avg_salary | difference |
+-------------+------------+-----------+----------+------------+------------+
|        1001 | Anay       | Somani    | 10000.00 |      10400 |    -400.00 |
|        1003 | Pranav     | Dagay     | 12000.00 |      10400 |    1600.00 |
|        1004 | Harsh      | Aru       |  8000.00 |      10400 |   -2400.00 |
|        1005 | Ayush      | Jain      | 11000.00 |      10400 |     600.00 |
|        1006 | Abhishek   | Bhagawati | 11000.00 |      10400 |     600.00 |
+-------------+------------+-----------+----------+------------+------------+
5 rows in set (0.00 sec)
```