Lab File

Name: T.Aswin Barath

USN:18BTRSE031

Subject: Database Management Systems Lab

Subject Code: 18CS4SP04L

Question 1: Create User in Oracle Database and grant and revoke the privileges and use of commit savepoint and rollback command.

Aim : To create a user in oracle database and grant and revoke the privileges and use of commit, savepoint and rollback command. And also analyze about the above mentioned command.

Consider the following table :-

```
SQL> select *from customers
  2  ;

        ID NAME                      AGE ADDRESS                SALARY
---------- -------------------- ---------- -------------------- ----------
         1 Manohar                      18 Hyderabad                  5000
         2 Sameendra                    19 Kukatpally                 6000
         3 JayaChandra                  19 Kurnool                    6500
         4 Chandrika                    21 Pune                       7500
         5 Sreenidhi                    25 Adoni                      5500
         6 Ragasree                     26 Vijayawada                 8500
         7 Geethika                     28 Vijayawada                 8600
         8 Chandu                       30 Rajamundry                 9600
        10 Reddy                        19 Hayathnagar                5600

9 rows selected.
```

SQL Queries :-

```
SQL> GRANT ALL ON customers to oe;

Grant succeeded.

SQL> grant select,update,insert on customers to oe with grant option;

Grant succeeded.

SQL> revoke all on customers from oe;

Revoke succeeded.

SQL> delete from customers where id=10;

1 row deleted.
```

```
SQL> select *from customers;

        ID NAME                          AGE ADDRESS                   SALARY
---------- -------------------- ---------- -------------------- ----------
         1 Manohar                      18 Hyderabad                  5000
         2 Sameendra                    19 Kukatpally                 6000
         3 JayaChandra                  19 Kurnool                    6500
         4 Chandrika                    21 Pune                       7500
         5 Sreenidhi                    25 Adoni                      5500
         6 Ragasree                     26 Vijayawada                 8500
         7 Geethika                     28 Vijayawada                 8600
         8 Chandu                       30 Rajamundry                 9600

8 rows selected.

SQL> commit;

Commit complete.

SQL>
```

Result: I have executed all the commands : grant, revoke, savepoint, rollback, commit) successfully.

Question 2: Create the following:
(a) Synonym, sequences and Index
(b) Create alter and update views.

Aim: To create synonym, sequence, index, alter view and update view in database. And also to observe how those are working.

SQL Queries :-

```
SQL> create sequence ord_no
  2  start with 1
  3  increment by 1
  4  minvalue 1
  5  maxvalue 25
  6  cycle
  7  cache 10;

Sequence created.

SQL> select ord_no.nextval from dual;

   NEXTVAL
----------
         1

SQL> select ord_no.nextval from dual;

   NEXTVAL
----------
         2
```

```
SQL> create or replace view customer_details as select ID,NAME,SALARY from customers where ID IN(1,2,3)
  2  with check option;

View created.

SQL> select *from customer_details;

        ID NAME                    SALARY
---------- -------------------- ----------
         1 Manohar                    5000
         2 Sameendra                  6000
         3 JayaChandra                6500

SQL> create bitmap index cus_dt on customers(id,name);

Index created.

SQL> create INDEX CUS_SAL on customers(salary);

Index created.
```

```
SQL> drop index cus_dt;

Index dropped.

SQL> drop index cus_sal;

Index dropped.

SQL> select *from user_synonyms;

SYNONYM_NAME                   TABLE_OWNER
------------------------------ ------------------------------
TABLE_NAME
------------------------------
DB_LINK
--------------------------------------------------------------------------------
SYSCATALOG                     SYS
SYSCATALOG


CATALOG                        SYS
CATALOG


SYNONYM_NAME                   TABLE_OWNER
------------------------------ ------------------------------
TABLE_NAME
------------------------------
DB_LINK
--------------------------------------------------------------------------------

TAB                            SYS
TAB


COL                            SYS
COL
```

```
SYNONYM_NAME                   TABLE_OWNER
------------------------------ ------------------------------
TABLE_NAME
------------------------------
DB_LINK
--------------------------------------------------------------------------------


TABQUOTAS                      SYS
TABQUOTAS


SYSFILES                       SYS

SYNONYM_NAME                   TABLE_OWNER
------------------------------ ------------------------------
TABLE_NAME
------------------------------
DB_LINK
--------------------------------------------------------------------------------
SYSFILES


PUBLICSYN                      SYS
PUBLICSYN



SYNONYM_NAME                   TABLE_OWNER
------------------------------ ------------------------------
TABLE_NAME
------------------------------
DB_LINK
--------------------------------------------------------------------------------
PRODUCT_USER_PROFILE           SYSTEM
SQLPLUS_PRODUCT_PROFILE



8 rows selected.
```

```
SQL> drop view custo

View dropped.

SQL> create view cus

View created.

SQL>
```

Result: I have successfully created Synonym, sequences and Index.

Create alter and update views and observed how all the statements are worked.

Question 3: Create PL/SQL program using cursors, control structure, exception handling

Aim: To create pl/sql program using cursor, control structure, exception handling

```
SQL> DECLARE
  2  c_id customers.id%type := 1;
  3  c_sal customers.salary%type;
  4  BEGIN
  5  SELECT salary
  6  INTO c_sal
  7  FROM customers
```

```
SQL> DECLARE
  2  c_id customers.id%type;
  3  c_name customers.name%type;
  4  c_addr customers.address%type;
  5  CURSOR c_customers is
  6  SELECT id,name,address FROM customers;
  7  BEGIN
  8  OPEN c_customers;
  9  LOOP
 10  FETCH c_customers into c_id,c_name,c_addr;
 11  EXIT WHEN c_customers%notfound;
 12  dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
 13  END LOOP;
 14  CLOSE c_customers;
 15  END;
```

```
SQL> DECLARE
  2  total_rows number(2);
  3  BEGIN
  4  UPDATE customers
  5  SET salary = salary + 500;
  6  IF sql%notfound THEN
  7  dbms_output.put_line('no customers selected');
  8  ELSIF sql%found THEN
  9  total_rows := sql%rowcount;
 10  dbms_output.put_line(total_rows || 'customers selected');
 11  END IF;
 12  END;
 13  /

PL/SQL procedure successfully completed.

SQL> select *from customers
  2  ;

        ID NAME                      AGE ADDRESS                SALARY
---------- -------------------- ---------- -------------------- ----------
         1 Manohar                      18 Hyderabad                  5500
         2 Sameendra                    19 Kukatpally                 6500
         3 JayaChandra                  19 Kurnool                    7000
         4 Chandrika                    21 Pune                       8000
         5 Sreenidhi                    25 Adoni                      6000
         6 Ragasree                     26 Vijayawada                 9000
         7 Geethika                     28 Vijayawada                 9100
         8 Chandu                       30 Rajamundry                10100

8 rows selected.
```

```
SQL> DECLARE
  2  c_id customers.id%type :=8;
  3  c_name customers.Name%type;
  4  c_addr customers.address%type;
  5  BEGIN
  6  SELECT name,address INTO c_name,c_addr
  7  FROM customers
  8  WHERE id = c_id;
  9  DBMS_OUTPUT.PUT_LINE ('Name: '|| c_name);
 10  DBMS_OUTPUT.PUT_LINE ('Address : '|| c_addr);
 11  EXCEPTION
 12  WHEN no_data_found THEN
 13  dbms_output.put_line('No such customer!');
 14  WHEN others THEN
```

Result: I have successfully created a Cursor and by using this I have updated the values of the table.

Question 4: Create following:
(a) Simple Triggers (b) Package using procedures and functions.

Aim: To know about how to create a simple Trigger, Procedures and Functions using pl/sql

```
SQL> CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE DELETE OR INSERT OR UPDATE ON customers
  3  FOR EACH ROW
  4  WHEN (NEW.ID > 0)
  5  DECLARE
  6   sal_diff number;
  7  BEGIN
  8   sal_diff := :NEW.salary - :OLD.salary;
```

```
SQL> select *from customers;

        ID NAME                      AGE ADDRESS                 SALARY
---------- -------------------- ---------- -------------------- ----------
         1 Manohar                      18 Hyderabad                  5500
         2 Sameendra                    19 Kukatpally                 6000
         3 JayaChandra                  19 Kurnool                    6500
         4 Chandrika                    21 Pune                       7500
         5 Sreenidhi                    25 Adoni                      5500
         6 Ragasree                     26 Vijayawada                 8500
         7 Geethika                     28 Vijayawada                 8600
         8 Chandu                       30 Rajamundry                 9600

8 rows selected.

SQL> CREATE PROCEDURE findMin(x IN number,y IN number,z OUT number)IS
  2  BEGIN
  3  IF x < y THEN
  4  z:= x;
  5  ELSE
  6  z:= y;
```

```
SQL> CREATE OR REPLACE FUNCTION totalCustomers
  2  RETURN number IS
  3  total number(2) := 0;
  4  BEGIN
  5  SELECT count(*) into total
  6  FROM customers;
  7  RETURN total;
  8  END;
  9  /

Function created.

SQL> DECLARE
  2  c number(2);
  3  BEGIN
  4  c := totalCustomers();
  5  dbms_output.put_line('Total no.of customers: ' || c);
```

Result: Successfully created the trigger, procedure and function and display the result using pl/sql.

Question 5: Create the table for
(a) COMPANY database
(b) STUDENT database
and Insert five records for each attributes.

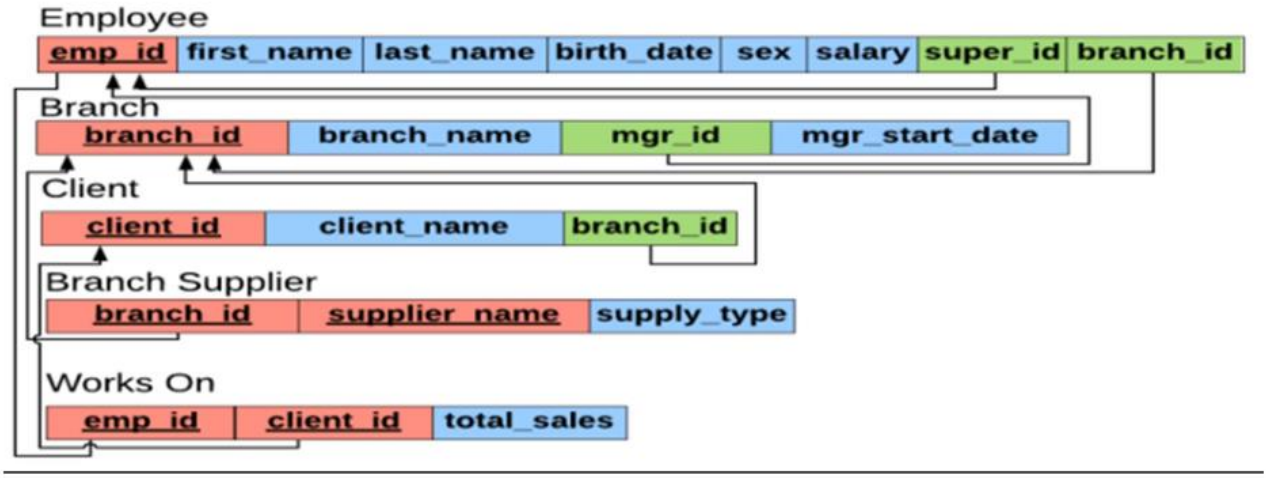
Aim: To create a table for company database and student database and insert 5 rows in each attributes.

(a) COMPANY database
SQL Queries :-

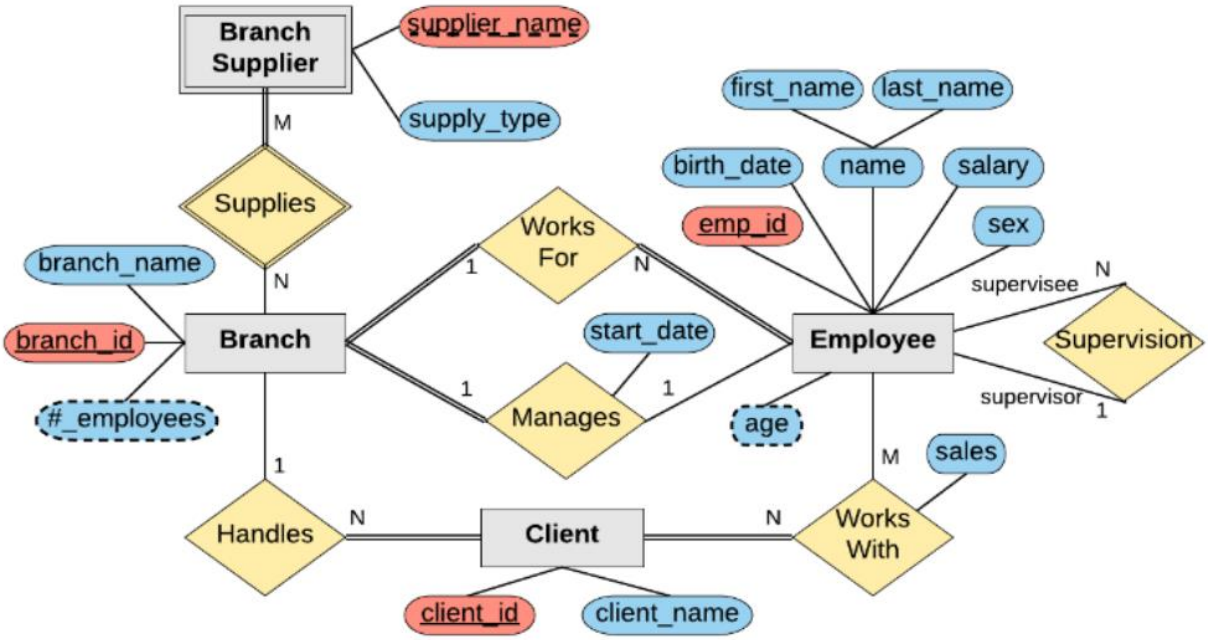
```
mysql> CREATE DATABASE COMPANY;
Query OK, 1 row affected (0.00 sec)

mysql> USE COMPANY;
Database changed
mysql> CREATE TABLE employee ( emp_id INT PRIMARY KEY, first_name VARCHAR(40),
last_name VARCHAR(40), birth_day DATE, sex VARCHAR(1), salary INT, super_id INT
, branch_id INT);
Query OK, 0 rows affected (0.81 sec)

mysql> CREATE TABLE branch (
    -> branch_id INT PRIMARY KEY,
    -> branch_name VARCHAR(40),
    -> mgr_id INT,
    -> mgr_start_date DATE,
    -> FOREIGN KEY(mgr_id) REFERENCES employee(emp_id) ON DELETE SET NULL);
Query OK, 0 rows affected (0.43 sec)

mysql> ALTER TABLE employee
    -> ADD FOREIGN KEY(branch_id)
    -> REFERENCES branch(branch_id)
    -> ON DELETE SET NULL;
Query OK, 0 rows affected (1.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> ALTER TABLE employee
    -> ADD FOREIGN KEY(super_id)
    -> REFERENCES employee(emp_id)
    -> ON DELETE SET NULL;
Query OK, 0 rows affected (0.91 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> CREATE TABLE client (
    -> client_id INT PRIMARY KEY,
    -> client_name VARCHAR(40),
    -> branch_id INT,
    -> FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE SET NULL);

Query OK, 0 rows affected (0.42 sec)

mysql> CREATE TABLE works_with (
    -> emp_id INT,
    -> client_id INT,
    -> total_sales INT,
    -> PRIMARY KEY(emp_id, client_id),
    -> FOREIGN KEY(emp_id) REFERENCES employee(emp_id) ON DELETE CASCADE,
    -> FOREIGN KEY(client_id) REFERENCES client(client_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.40 sec)

mysql> CREATE TABLE branch_supplier (
    -> branch_id INT,
    -> supplier_name VARCHAR(40),
    -> supply_type VARCHAR(40),
    -> PRIMARY KEY(branch_id, supplier_name),
    -> FOREIGN KEY(branch_id) REFERENCES branch(branch_id) ON DELETE CASCADE);
Query OK, 0 rows affected (0.35 sec)

mysql> SHOW TABLES;
+-------------------+
| Tables_in_COMPANY |
+-------------------+
| branch            |
| branch_supplier   |
| client            |
| employee          |
| works_with        |
+-------------------+
5 rows in set (0.00 sec)
```

Company Database Schema



Company ER Diagram



Insertion of rows to the tables employee, branch,branh_supplier, clients and works_with respectively:

```
mysql> INSERT INTO employee VALUES
    -> (100, 'Jeff', 'Bezos', '2017-01-01', 'M', 250000, NULL, NULL);
Query OK, 1 row affected (0.43 sec)

mysql> INSERT INTO employee VALUES
    -> (101, 'Tim', 'Cook', '2007-02-09', 'M', 450000, NULL, NULL);
Query OK, 1 row affected (0.60 sec)

mysql> INSERT INTO employee VALUES
    -> (102, 'Sachin', 'Bansal', '2019-03-20', 'M', 650000, NULL, NULL);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO employee VALUES
    -> (103, 'Bill', 'Gates', '2010-05-21', 'M', 850000, NULL, NULL);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO employee VALUES
    -> (104, 'Sundar', 'Pichai', '2015-09-05', 'M', 750000, NULL, NULL);
Query OK, 1 row affected (0.18 sec)
```

```
mysql> INSERT INTO branch VALUES(3, 'Seattle', 100, '2017-04-01');
Query OK, 1 row affected (0.44 sec)

mysql> INSERT INTO branch VALUES(3, 'California', 101, '2019-05-01');
ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'
mysql> INSERT INTO branch VALUES(4, 'California', 101, '2019-05-01');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO branch VALUES(5, 'Bengaluru', 102, '2018-02-01');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO branch VALUES(1, 'Washington', 103, '2017-01-21');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO branch VALUES(2, 'California', 104, '2014-12-20');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO branch_supplier VALUES(2, 'AWS', 'Cloud Computing');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO branch_supplier VALUES(2, 'Apple iSeries', 'Mobile Devices');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO branch_supplier VALUES(3, 'Office 365', 'Utilities');
Query OK, 1 row affected (0.06 sec)
```

```
mysql> INSERT INTO branch_supplier VALUES(3, 'Microsoft Azure', 'Cloud Computing');
Query OK, 1 row affected (0.44 sec)

mysql> INSERT INTO branch_supplier VALUES(3, 'FlipKart', 'E-Commerce');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO branch_supplier VALUES(3, 'Google Chrome', 'Search Engine');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO client VALUES(400, 'Apple Inc.', 2);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO client VALUES(401, 'AT&T and Verizon', 2);
Query OK, 1 row affected (0.17 sec)

mysql> INSERT INTO client VALUES(402, 'Pixar', 3);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO client VALUES(403, 'Walmart', 3);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO client VALUES(404, 'Amazon', 2);
Query OK, 1 row affected (0.07 sec)
```

```
mysql> INSERT INTO works_with VALUES(100, 400, 55000);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO works_with VALUES(101, 401, 267000);
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO works_with VALUES(102, 402, 22500);
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO works_with VALUES(103, 403, 5000);
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO works_with VALUES(104, 403, 12000);
Query OK, 1 row affected (0.07 sec)

mysql>
```

Display of All Tables :-

```
mysql> SHOW TABLES;
+-------------------+
| Tables_in_COMPANY |
+-------------------+
| branch            |
| branch_supplier   |
| client            |
| employee          |
| works_with        |
+-------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM branch;
+-----------+-------------+--------+----------------+
| branch_id | branch_name | mgr_id | mgr_start_date |
+-----------+-------------+--------+----------------+
|         1 | Washington  |    103 | 2017-01-21     |
|         2 | California  |    104 | 2014-12-20     |
|         3 | Seattle     |    100 | 2017-04-01     |
|         4 | California  |    101 | 2019-05-01     |
|         5 | Bengaluru   |    102 | 2018-02-01     |
+-----------+-------------+--------+----------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM branch_supplier;
+-----------+-----------------+-----------------+
| branch_id | supplier_name   | supply_type     |
+-----------+-----------------+-----------------+
|         2 | Apple iSeries   | Mobile Devices  |
|         2 | AWS             | Cloud Computing |
|         3 | FlipKart        | E-Commerce      |
|         3 | Google Chrome   | Search Engine   |
|         3 | Microsoft Azure | Cloud Computing |
|         3 | Office 365      | Utilities       |
+-----------+-----------------+-----------------+
6 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM client;
+-----------+-------------------+-----------+
| client_id | client_name       | branch_id |
+-----------+-------------------+-----------+
|       400 | Apple Inc.        |         2 |
|       401 | AT&T and Verizon  |         2 |
|       402 | Pixar             |         3 |
|       403 | Walmart           |         3 |
|       404 | Amazon            |         2 |
+-----------+-------------------+-----------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM employee;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM works_with;
+--------+-----------+-------------+
| emp_id | client_id | total_sales |
+--------+-----------+-------------+
|    100 |       400 |       55000 |
|    101 |       401 |      267000 |
|    102 |       401 |      267000 |
|    102 |       402 |       22500 |
|    103 |       403 |        5000 |
|    104 |       403 |       12000 |
+--------+-----------+-------------+
6 rows in set (0.00 sec)

mysql>
```

(b) STUDENT database
SQL Queries :-

```
mysql> create database STUDENT;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE TABLE student( sid int not null,name text not null, primary key(sid));
ERROR 1046 (3D000): No database selected
mysql> USE STUDENT;
Database changed
mysql> CREATE TABLE student( sid int not null,name text not null, primary key(sid));
Query OK, 0 rows affected (0.33 sec)

mysql> CREATE TABLE teachers(tid int not null,name text not null,primary key(tid));
Query OK, 0 rows affected (0.33 sec)

mysql> CREATE TABLE subjects(subid int not null,name text not null,primary key(subid));
Query OK, 0 rows affected (0.47 sec)

mysql> CREATE TABLE grades
    -> (studentID int not null references students(sid),
    -> teacherID int not null references teachers(tid),
    -> subjectID int not null references subjects(subid),
    -> grade varchar(3), primary key(studentID, teacherID, subjectID));
Query OK, 0 rows affected (0.34 sec)

mysql>
```

```
mysql> INSERT INTO subjects (subid, name) VALUES (1, 'Artificial Intelligence');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO subjects (subid, name) VALUES (2, 'Data Science');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO subjects (subid, name) VALUES (3, 'Software Engineering');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (1, 2, 1, 'A');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (1, 2, 2, 'B');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (7, 4, 3, 'C+');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (7, 3, 2, 'F');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (6, 2, 1, 'B+');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO grades (studentID, teacherID, subjectID, grade) VALUES (2, 4, 3, 'C');
Query OK, 1 row affected (0.13 sec)
```

```
mysql> INSERT INTO student (sid, name) VALUES(1, 'Aswin');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO student (sid, name) VALUES(2, 'Kishore');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO student (sid, name) VALUES(3, 'Venkat');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO student (sid, name) VALUES(4, 'Mudit');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO student (sid, name) VALUES(5, 'Vinay');
Query OK, 1 row affected (0.04 sec)

mysql> INSERT INTO student (sid, name) VALUES(6, 'Jenish');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO student (sid, name) VALUES(7, 'Vyshnav');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO teachers (tid, name) VALUES (1, 'Suresh');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO teachers (tid, name) VALUES (2, 'Maruthi');
Query OK, 1 row affected (0.05 sec)

mysql> INSERT INTO teachers (tid, name) VALUES (3, 'Nandan');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO teachers (tid, name) VALUES (4, 'Sandeep');
Query OK, 1 row affected (0.06 sec)
```

Result: I have successfully created company database and student database and also successfully inserted the values in both database.

Question 6: Illustrate the use of SELECT statement

Aim: To understand the uses of select statement in various cases

SQL Queries :-

```
mysql> SELECT * FROM client;
+-----------+------------------+-----------+
| client_id | client_name      | branch_id |
+-----------+------------------+-----------+
|       400 | Apple Inc.       |         2 |
|       401 | AT&T and Verizon |         2 |
|       402 | Pixar            |         3 |
|       403 | Walmart          |         3 |
|       404 | Amazon           |         2 |
+-----------+------------------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM employee;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * from employee ORDER BY salary ASC;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * from employee ORDER BY salary DESC;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * from employee ORDER BY sex, first_name, last_name;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> SELECT * from employee LIMIT 5;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT first_name, employee.last_name FROM employee;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Jeff       | Bezos     |
| Tim        | Cook      |
| Sachin     | Bansal    |
| Bill       | Gates     |
| Sundar     | Pichai    |
+------------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT first_name AS forename, employee.last_name AS surname
    -> FROM employee;
+----------+---------+
| forename | surname |
+----------+---------+
| Jeff     | Bezos   |
| Tim      | Cook    |
| Sachin   | Bansal  |
| Bill     | Gates   |
| Sundar   | Pichai  |
+----------+---------+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> SELECT DISTINCT sex FROM employee;
+------+
| sex  |
+------+
| M    |
+------+
1 row in set (0.01 sec)

mysql> SELECT * FROM employee WHERE sex = 'M';
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM employee WHERE branch_id = 2;
Empty set (0.02 sec)

mysql> SELECT emp_id, first_name, last_name FROM employee
    -> WHERE birth_day >= 1970-01-01;
+--------+------------+-----------+
| emp_id | first_name | last_name |
+--------+------------+-----------+
|    100 | Jeff       | Bezos     |
|    101 | Tim        | Cook      |
|    102 | Sachin     | Bansal    |
|    103 | Bill       | Gates     |
|    104 | Sundar     | Pichai    |
+--------+------------+-----------+
5 rows in set, 1 warning (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM employee
    -> WHERE (birth_day >= '1970-01-01' AND sex = 'F') OR salary > 80000;
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    101 | Tim        | Cook      | 2007-02-09 | M    | 450000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM employee
    -> WHERE birth_day BETWEEN '1970-01-01' AND '1975-01-01';
Empty set (0.00 sec)

mysql> SELECT * FROM employee WHERE first_name IN ('Jeff', 'Sachin', 'Bill', 'Sundar');
+--------+------------+-----------+------------+------+--------+----------+-----------+
| emp_id | first_name | last_name | birth_day  | sex  | salary | super_id | branch_id |
+--------+------------+-----------+------------+------+--------+----------+-----------+
|    100 | Jeff       | Bezos     | 2017-01-01 | M    | 250000 |     NULL |      NULL |
|    102 | Sachin     | Bansal    | 2019-03-20 | M    | 650000 |     NULL |      NULL |
|    103 | Bill       | Gates     | 2010-05-21 | M    | 850000 |     NULL |      NULL |
|    104 | Sundar     | Pichai    | 2015-09-05 | M    | 750000 |     NULL |      NULL |
+--------+------------+-----------+------------+------+--------+----------+-----------+
4 rows in set (0.00 sec)

mysql> SELECT COUNT(super_id)FROM employee;
+-----------------+
| COUNT(super_id) |
+-----------------+
|               0 |
+-----------------+
1 row in set (0.00 sec)

mysql>

mysql> SELECT AVG(salary) FROM employee;
+-------------+
| AVG(salary) |
+-------------+
| 590000.0000 |
+-------------+
1 row in set (0.00 sec)

mysql> SELECT SUM(salary) FROM employee;
+-------------+
| SUM(salary) |
+-------------+
|     2950000 |
+-------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(sex), sex FROM employee GROUP BY sex;
+------------+------+
| COUNT(sex) | sex  |
+------------+------+
|          5 | M    |
+------------+------+
1 row in set (0.00 sec)
```

```
mysql> SELECT SUM(total_sales), client_id FROM works_with
    -> GROUP BY client_id;
+------------------+-----------+
| SUM(total_sales) | client_id |
+------------------+-----------+
|            55000 |       400 |
|           534000 |       401 |
|            22500 |       402 |
|            17000 |       403 |
+------------------+-----------+
4 rows in set (0.00 sec)

mysql> SELECT employee.first_name AS Employee_Branch_Names
    -> FROM employee UNION
    -> SELECT branch.branch_name
    -> FROM branch;
+-----------------------+
| Employee_Branch_Names |
+-----------------------+
| Jeff                  |
| Tim                   |
| Sachin                |
| Bill                  |
| Sundar                |
| Washington            |
| California            |
| Seattle               |
| Bengaluru             |
+-----------------------+
9 rows in set (0.00 sec)
```

SQL Queries :-

```
mysql> select * from student;
+-----+---------+
| sid | name    |
+-----+---------+
|   1 | Aswin   |
|   2 | Kishore |
|   3 | Venkat  |
|   4 | Mudit   |
|   5 | Vinay   |
|   6 | Jenish  |
|   7 | Vyshnav |
+-----+---------+
7 rows in set (0.00 sec)

mysql>
mysql> select * from teachers;
+-----+---------+
| tid | name    |
+-----+---------+
|   1 | Suresh  |
|   2 | Maruthi |
|   3 | Nandan  |
|   4 | Sandeep |
+-----+---------+
4 rows in set (0.00 sec)

mysql>

mysql> select * from subjects;
+-------+-------------------------+
| subid | name                    |
+-------+-------------------------+
|     1 | Artificial Intelligence |
|     2 | Data Science            |
|     3 | Software Engineering    |
+-------+-------------------------+
3 rows in set (0.00 sec)

mysql>
mysql> select * from grades;
+-----------+-----------+-----------+-------+
| studentID | teacherID | subjectID | grade |
+-----------+-----------+-----------+-------+
|         1 |         2 |         1 | A     |
|         1 |         2 |         2 | B     |
|         2 |         4 |         3 | C     |
|         6 |         2 |         1 | B+    |
|         7 |         3 |         2 | F     |
|         7 |         4 |         3 | C+    |
+-----------+-----------+-----------+-------+
6 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from student order by name ASC;
+-----+---------+
| sid | name    |
+-----+---------+
|   1 | Aswin   |
|   6 | Jenish  |
|   2 | Kishore |
|   4 | Mudit   |
|   3 | Venkat  |
|   5 | Vinay   |
|   7 | Vyshnav |
+-----+---------+
7 rows in set (0.00 sec)

mysql> select name from student where sid in (select studentID from grades
 where teacherID in (select tid from teachers where name = 'Sandeep') );
+---------+
| name    |
+---------+
| Kishore |
| Vyshnav |
+---------+
2 rows in set (0.00 sec)

mysql> select name from teachers where tid in
    -> (select teacherID from grades where subjectID in
    -> (select subid from subjects where
    -> name = 'Software Engineering') );
+---------+
| name    |
+---------+
| Sandeep |
+---------+
1 row in set (0.00 sec)

mysql> select name from teachers where tid not in
    -> (select teacherID from grades);
+--------+
| name   |
+--------+
| Suresh |
+--------+
1 row in set (0.00 sec)
```

```
mysql> select name from student where sid not in
    -> (select studentID from grades);
+--------+
| name   |
+--------+
| Venkat |
| Mudit  |
| Vinay  |
+--------+
3 rows in set (0.00 sec)

mysql> select name from student where sid in
    -> (SELECT studentID FROM grades g1 WHERE
    -> (SELECT COUNT(*) FROM grades g2 WHERE
    -> g1.subjectID = g2.subjectID AND
    -> g1.teacherID = g2.teacherID ) > 1
    -> ORDER BY subjectID );
mysql> select t.name as "Teacher", sub.name as "Subject",
    -> s.name as "Student" from grades g1,
    -> grades g2, student s, teachers t,
    -> subjects sub where g1.teacherID = g2.teacherID
    -> and g1.subjectID = g2.subjectID
    -> and g1.studentID = s.sid
    -> and g1.teacherID = t.tid
    -> and g1.subjectID = sub.subid
    -> order by t.name, sub.name, s.name;
+---------+-------------------------+---------+
| Teacher | Subject                 | Student |
+---------+-------------------------+---------+
| Maruthi | Artificial Intelligence | Aswin   |
| Maruthi | Artificial Intelligence | Aswin   |
| Maruthi | Artificial Intelligence | Jenish  |
| Maruthi | Artificial Intelligence | Jenish  |
| Maruthi | Data Science            | Aswin   |
| Nandan  | Data Science            | Vyshnav |
| Sandeep | Software Engineering    | Kishore |
| Sandeep | Software Engineering    | Kishore |
| Sandeep | Software Engineering    | Vyshnav |
| Sandeep | Software Engineering    | Vyshnav |
+---------+-------------------------+---------+
10 rows in set (0.00 sec)

mysql>
```

Result: I have successfully observed and executed the various types of use of select statement.

Question 7 : Conditional retrieval - WHERE clause.

Aim: To observe the use in where clause in various process.

Consider the following table :-

```
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)
```

SQL Queries :-

```
mysql> SELECT * FROM EMPLOYEE WHERE DEPARTMENT_ID=31;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
+-------------+------------+-----------+--------+---------------+
1 row in set (0.05 sec)

mysql> SELECT * FROM EMPLOYEE WHERE LAST_NAME='JAIN';
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
1 row in set (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM EMPLOYEE WHERE SALARY>=60000;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE WHERE SALARY BETWEEN 57500 AND 62500;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
+-------------+------------+-----------+--------+---------------+
3 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM EMPLOYEE WHERE DEPARTMENT_ID IN(16,19,27,31);
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE WHERE FIRST_NAME LIKE('v%');
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          53 | venkat     | kavi      |  62500 |            19 |
+-------------+------------+-----------+--------+---------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM EMPLOYEE WHERE LAST_NAME LIKE('b%') OR LAST_NAME LIKE('%h')
    -> ;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
+-------------+------------+-----------+--------+---------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM EMPLOYEE WHERE LAST_NAME LIKE('k%') OR LAST_NAME LIKE('%n');

+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
2 rows in set (0.00 sec)

mysql>
```

Result: Various uses of where clause is observed carefully and executed successfully.

Question 8: Query sorted - ORDER BY clause

Aim: To know about order by clause and display the result of using order by clause.

Consider the following table :-

```
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)
```

SQL Queries :-

```
mysql> SELECT FIRST_NAME, LAST_NAME, SALARY FROM EMPLOYEE ORDER BY DEPARTMENT_ID;

+------------+-----------+--------+
| FIRST_NAME | LAST_NAME | SALARY |
+------------+-----------+--------+
| mudit      | jain      |  65500 |
| venkat     | kavi      |  62500 |
| sri        | sarvesh   |  57500 |
| aswin      | barath    |  60000 |
+------------+-----------+--------+
4 rows in set (0.01 sec)

mysql> SELECT FIRST_NAME, LAST_NAME, SALARY FROM EMPLOYEE ORDER BY SALARY DESC;
+------------+-----------+--------+
| FIRST_NAME | LAST_NAME | SALARY |
+------------+-----------+--------+
| mudit      | jain      |  65500 |
| venkat     | kavi      |  62500 |
| aswin      | barath    |  60000 |
| sri        | sarvesh   |  57500 |
+------------+-----------+--------+
4 rows in set (0.01 sec)

mysql>
```

```
mysql> SELECT FIRST_NAME, SALARY, DEPARTMENT_ID FROM EMPLOYEE ORDER BY LAST_NAME;

+------------+--------+---------------+
| FIRST_NAME | SALARY | DEPARTMENT_ID |
+------------+--------+---------------+
| aswin      |  60000 |            31 |
| mudit      |  65500 |            16 |
| venkat     |  62500 |            19 |
| sri        |  57500 |            27 |
+------------+--------+---------------+
4 rows in set (0.06 sec)

mysql> SELECT FIRST_NAME, SALARY, DEPARTMENT_ID FROM EMPLOYEE
    -> ORDER BY EMPLOYEE_ID DESC;
+------------+--------+---------------+
| FIRST_NAME | SALARY | DEPARTMENT_ID |
+------------+--------+---------------+
| mudit      |  65500 |            16 |
| venkat     |  62500 |            19 |
| sri        |  57500 |            27 |
| aswin      |  60000 |            31 |
+------------+--------+---------------+
4 rows in set (0.00 sec)

mysql>
```

Result: I have used the both order by clause and got the idea how it is working and also successfully attach the files.

Question 9(a): UNION, INTERSECTION and MINUS operations on tables

Aim: To get the full idea on UNION, INTERSECTION and MINUS on a table

Consider two tables named table1 and table2 with same attributes name and usn :-

```
mysql> create table table1(name varchar(20), usn int);
Query OK, 0 rows affected (1.46 sec)

mysql> create table table2(name varchar(20), usn int);
Query OK, 0 rows affected (0.38 sec)
```

SQL Queries :-

```
mysql> select * from table1;
+---------+------+
| name    | usn  |
+---------+------+
| aswin   |   31 |
| kishore |   12 |
| mudit   |   16 |
| vinay   |    6 |
+---------+------+
4 rows in set (0.02 sec)

mysql> select * from table2;
+---------+------+
| name    | usn  |
+---------+------+
| mudit   |   16 |
| vinay   |    6 |
| sarvesh |   27 |
| souvik  |   26 |
+---------+------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from table1 union select * from table2;
+---------+------+
| name    | usn  |
+---------+------+
| aswin   |   31 |
| kishore |   12 |
| mudit   |   16 |
| vinay   |    6 |
| sarvesh |   27 |
| souvik  |   26 |
+---------+------+
6 rows in set (0.00 sec)

mysql> select * from table1 union all select * from table2;
+---------+------+
| name    | usn  |
+---------+------+
| aswin   |   31 |
| kishore |   12 |
| mudit   |   16 |
| vinay   |    6 |
| mudit   |   16 |
| vinay   |    6 |
| sarvesh |   27 |
| souvik  |   26 |
+---------+------+
8 rows in set (0.00 sec)

mysql>
```

```
mysql> select t1.name,t1.usn from table1 t1, table2 t2 where t1.name=t2.name;
+-------+------+
| name  | usn  |
+-------+------+
| mudit |   16 |
| vinay |    6 |
+-------+------+
2 rows in set (0.00 sec)
```

Result: I have successfully completed the union, intersection, minus operation on a table.

Question 9(b): UPDATE, ALTER, DELETE, DROP operations on tables

Aim: To use update, alter, delete, drop statement in various ways in a table and observe the behavior of these command.

Consider the following table :-

```
mysql> select * from EMPLOYEE;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  57500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)
```

SQL Queries :-

```
mysql> UPDATE EMPLOYEE SET SALARY=65000 WHERE EMPLOYEE_ID=52;
Query OK, 1 row affected (0.44 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | aswin      | barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  65000 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)

mysql> UPDATE EMPLOYEE SET FIRST_NAME='Aswin', LAST_NAME='Barath'
    -> WHERE EMPLOYEE_ID=51;
Query OK, 1 row affected (0.43 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | Aswin      | Barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  65000 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM EMPLOYEE;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | Aswin      | Barath    |  60000 |            31 |
|          52 | sri        | sarvesh   |  65000 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)

mysql> UPDATE EMPLOYEE SET SALARY=67500 WHERE LAST_NAME LIKE('%h');
Query OK, 2 rows affected (0.47 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * FROM EMPLOYEE;
+-------------+------------+-----------+--------+---------------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | DEPARTMENT_ID |
+-------------+------------+-----------+--------+---------------+
|          51 | Aswin      | Barath    |  67500 |            31 |
|          52 | sri        | sarvesh   |  67500 |            27 |
|          53 | venkat     | kavi      |  62500 |            19 |
|          54 | mudit      | jain      |  65500 |            16 |
+-------------+------------+-----------+--------+---------------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> DESC EMPLOYEE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11)     | NO   |     | NULL    |       |
| FIRST_NAME    | varchar(10) | YES  |     | NULL    |       |
| LAST_NAME     | varchar(10) | YES  |     | NULL    |       |
| SALARY        | int(11)     | YES  |     | NULL    |       |
| DEPARTMENT_ID | int(11)     | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> ALTER TABLE EMPLOYEE ADD CITY VARCHAR(20);
Query OK, 0 rows affected (1.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMPLOYEE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11)     | NO   |     | NULL    |       |
| FIRST_NAME    | varchar(10) | YES  |     | NULL    |       |
| LAST_NAME     | varchar(10) | YES  |     | NULL    |       |
| SALARY        | int(11)     | YES  |     | NULL    |       |
| DEPARTMENT_ID | int(11)     | YES  |     | NULL    |       |
| CITY          | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql>
```

```
mysql> DESC EMPLOYEE;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID   | int(11)     | NO   |     | NULL    |       |
| FIRST_NAME    | varchar(10) | YES  |     | NULL    |       |
| LAST_NAME     | varchar(10) | YES  |     | NULL    |       |
| SALARY        | int(11)     | YES  |     | NULL    |       |
| DEPARTMENT_ID | int(11)     | YES  |     | NULL    |       |
| CITY          | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql> ALTER TABLE EMPLOYEE DROP COLUMN DEPARTMENT_ID;
Query OK, 0 rows affected (0.73 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC EMPLOYEE;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPLOYEE_ID | int(11)     | NO   |     | NULL    |       |
| FIRST_NAME  | varchar(10) | YES  |     | NULL    |       |
| LAST_NAME   | varchar(10) | YES  |     | NULL    |       |
| SALARY      | int(11)     | YES  |     | NULL    |       |
| CITY        | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>


mysql> DELETE FROM EMPLOYEE WHERE LAST_NAME='Barath';
Query OK, 1 row affected (0.07 sec)

mysql> SELECT * FROM EMPLOYEE;
+-------------+------------+-----------+--------+------+
| EMPLOYEE_ID | FIRST_NAME | LAST_NAME | SALARY | CITY |
+-------------+------------+-----------+--------+------+
|          52 | sri        | sarvesh   |  67500 | NULL |
|          53 | venkat     | kavi      |  62500 | NULL |
|          54 | mudit      | jain      |  65500 | NULL |
+-------------+------------+-----------+--------+------+
3 rows in set (0.00 sec)

mysql> DROP TABLE EMPLOYEE;
Query OK, 0 rows affected (0.56 sec)

mysql> DESC EMPLOYEE;
ERROR 1146 (42S02): Table 'where_clause.EMPLOYEE' doesn't exist
mysql>
```

Result: I have successfully executed UPDATE, ALTER, DELETE, DROP SQL commands.

Question 10: Query multiple tables using JOIN operation.

Aim: To use all join operation on two tables

Consider the following two tables for join operations :-

```
mysql> SELECT * FROM STUDENT;
+------+---------+-----------+------+
| NUM  | NAME    | CITY      | AGE  |
+------+---------+-----------+------+
|    1 | Aswin   | Ooty      |   19 |
|    2 | Kishore | Chennai   |   18 |
|    3 | Mudit   | Bangalore |   20 |
|    4 | Vinay   | Surat     |   20 |
+------+---------+-----------+------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM STUDENT_COURSE;
+------+------+
| C_NO | NO   |
+------+------+
|    3 |    1 |
|    3 |    2 |
|    2 |    3 |
|    1 |    4 |
+------+------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> SELECT STUDENT_COURSE.C_NO, STUDENT.NAME, STUDENT.AGE
    -> FROM STUDENT INNER JOIN STUDENT_COURSE
    -> ON STUDENT.NUM=STUDENT_COURSE.NO;
+------+---------+------+
| C_NO | NAME    | AGE  |
+------+---------+------+
|    3 | Aswin   |   19 |
|    3 | Kishore |   18 |
|    2 | Mudit   |   20 |
|    1 | Vinay   |   20 |
+------+---------+------+
4 rows in set (0.00 sec)

mysql> SELECT STUDENT_COURSE.C_NO, STUDENT.NAME, STUDENT.AGE
    -> FROM STUDENT LEFT JOIN STUDENT_COURSE
    -> ON STUDENT.NUM=STUDENT_COURSE.NO;
+------+---------+------+
| C_NO | NAME    | AGE  |
+------+---------+------+
|    3 | Aswin   |   19 |
|    3 | Kishore |   18 |
|    2 | Mudit   |   20 |
|    1 | Vinay   |   20 |
+------+---------+------+
4 rows in set (0.00 sec)

mysql> SELECT STUDENT_COURSE.C_NO, STUDENT.NAME, STUDENT.AGE
    -> FROM STUDENT RIGHT JOIN STUDENT_COURSE
    -> ON STUDENT.NUM=STUDENT_COURSE.NO;
+------+---------+------+
| C_NO | NAME    | AGE  |
+------+---------+------+
|    3 | Aswin   |   19 |
|    3 | Kishore |   18 |
|    2 | Mudit   |   20 |
|    1 | Vinay   |   20 |
+------+---------+------+
4 rows in set (0.00 sec)



mysql> SELECT STUDENT_COURSE.C_NO, NAME, AGE, NUM FROM STUDENT FULL JOIN S
TUDENT_COURSE ON NUM=STUDENT_COURSE.NO;
+------+---------+------+------+
| C_NO | NAME    | AGE  | NUM  |
+------+---------+------+------+
|    3 | Aswin   |   19 |    1 |
|    3 | Kishore |   18 |    2 |
|    2 | Mudit   |   20 |    3 |
|    1 | Vinay   |   20 |    4 |
+------+---------+------+------+
4 rows in set (0.00 sec)

mysql>
```

Result:I have successfully executed all types of join (Inner Join, Left Join, Right Join, Full Join) commands on two tables.

Question 11: Grouping the result of query - GROUP BY clause and HAVING clause

Aim: To use the group by clause and having clause. As we know also having clause is optional.

Consider the following table:

```
mysql> SELECT * FROM COUNTRIES;
+------+----------------+
| NO   | COUNTRY        |
+------+----------------+
|    1 | INDIA          |
|    2 | INDIA          |
|    3 | AMERICA        |
|    4 | AUSTRALIA      |
|    5 | UNITED KINGDOM |
|    6 | INDIA          |
|    7 | JAPAN          |
|    8 | RUSSIA         |
|    9 | AMERICA        |
|   10 | UNITED KINGDOM |
|   11 | AUSTRALIA      |
|   12 | SPAIN          |
|   13 | MEXICO         |
|   14 | CANADA         |
|   15 | SOUTH AFRICA   |
|   16 | JAPAN          |
|   17 | RUSSIA         |
|   18 | CANADA         |
|   19 | SPAIN          |
|   20 | MEXICO         |
+------+----------------+
20 rows in set (0.36 sec)

mysql>
```

```
mysql> SELECT COUNT(NO), COUNTRY FROM COUNTRIES
    -> GROUP BY COUNTRY
    -> HAVING COUNT(NO) > 1;
+-----------+----------------+
| COUNT(NO) | COUNTRY        |
+-----------+----------------+
|         2 | AMERICA        |
|         2 | AUSTRALIA      |
|         2 | CANADA         |
|         3 | INDIA          |
|         2 | JAPAN          |
|         2 | MEXICO         |
|         2 | RUSSIA         |
|         2 | SPAIN          |
|         2 | UNITED KINGDOM |
+-----------+----------------+
9 rows in set (0.00 sec)

mysql>
```

Result: I have successfully executed the “group by” and “having” clause commands.

Question 12: Query multiple tables using NATURAL and OUTER JOIN operation.

Aim: To see how the result is look like after using NATURAL and OUTER JOIN operation.

Consider the following two tables:

```
mysql> SELECT * FROM STUDENT;
+------+----------+------+
| USN  | NAME     | DEPT |
+------+----------+------+
|    1 | ASWIN    |    1 |
|    2 | KISHORE  |    4 |
|    3 | JENISH   |    3 |
|    4 | UVANESH  |    3 |
|    5 | SARVESH  |    2 |
|    6 | DHIRSITH |    5 |
+------+----------+------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM DEPARTMENT;
+------+-------+
| DEPT | DNAME |
+------+-------+
|    1 | SE    |
|    2 | MACT  |
|    3 | AI    |
+------+-------+
3 rows in set (0.00 sec)
```

SQL Queries :-

```
mysql> SELECT * FROM STUDENT NATURAL JOIN DEPARTMENT;
+------+------+---------+-------+
| DEPT | USN  | NAME    | DNAME |
+------+------+---------+-------+
|    1 |    1 | ASWIN   | SE    |
|    3 |    3 | JENISH  | AI    |
|    3 |    4 | UVANESH | AI    |
|    2 |    5 | SARVESH | MACT  |
+------+------+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> SELECT * FROM STUDENT LEFT OUTER JOIN DEPARTMENT ON
    -> STUDENT.USN=DEPARTMENT.DEPT;
+------+----------+------+------+-------+
| USN  | NAME     | DEPT | DEPT | DNAME |
+------+----------+------+------+-------+
|    1 | ASWIN    |    1 |    1 | SE    |
|    2 | KISHORE  |    4 |    2 | MACT  |
|    3 | JENISH   |    3 |    3 | AI    |
|    4 | UVANESH  |    3 | NULL | NULL  |
|    5 | SARVESH  |    2 | NULL | NULL  |
|    6 | DHIRSITH |    5 | NULL | NULL  |
+------+----------+------+------+-------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM STUDENT RIGHT OUTER JOIN DEPARTMENT ON STUDENT.USN
=DEPARTMENT.DEPT;
+------+---------+------+------+-------+
| USN  | NAME    | DEPT | DEPT | DNAME |
+------+---------+------+------+-------+
|    1 | ASWIN   |    1 |    1 | SE    |
|    2 | KISHORE |    4 |    2 | MACT  |
|    3 | JENISH  |    3 |    3 | AI    |
+------+---------+------+------+-------+
3 rows in set (0.00 sec)
```

We don’t have Full Join in MySQL, so I’ve emulated using Left and Right Join with the help of Union.

```
mysql> SELECT * FROM STUDENT LEFT JOIN DEPARTMENT
    -> ON STUDENT.USN=DEPARTMENT.DEPT
    -> UNION SELECT * FROM STUDENT RIGHT JOIN DEPARTMENT
    -> ON STUDENT.USN=DEPARTMENT.DEPT;
+------+----------+------+------+-------+
| USN  | NAME     | DEPT | DEPT | DNAME |
+------+----------+------+------+-------+
|    1 | ASWIN    |    1 |    1 | SE    |
|    2 | KISHORE  |    4 |    2 | MACT  |
|    3 | JENISH   |    3 |    3 | AI    |
|    4 | UVANESH  |    3 | NULL | NULL  |
|    5 | SARVESH  |    2 | NULL | NULL  |
|    6 | DHIRSITH |    5 | NULL | NULL  |
+------+----------+------+------+-------+
6 rows in set (0.36 sec)

mysql>
```

Result: I have successfully executed the NATURAL JOIN, OUTER JOIN (left outer join, right outer join, full outer join[“Using left and right join”]) operations on two tables STUDENT table and DEPARTMENT table.