PRACTICE QUESTIONS – DBMS

1. Create a Database with the name "university" and create all the required tables and insert data in to them.

Consider the following requirements list:

• The university has many Departments.

• The Departments offer one or more programs.

• A program is made up of one or more courses.

• A student must enroll in a program.

• A student takes the courses that are part of her program.

• A program has a name, a program identifier, department id, the total credit points required to graduate, and the year it commenced.

• A course has a name, a course identifier, program id, a credit point value, and the year it commenced. • Students have name, a surname, a student identifier, a date of birth, and the year they first enrolled. • When a student takes a course, the year and semester he attempted it are recorded. When he finishes the course, a grade (such as A or B) and a mark (such as 60 percent) are recorded. • Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).

```
mysql> CREATE DATABASE university;
Query OK, 1 row affected (0.02 sec)

mysql> USE university;
Database changed
mysql>
mysql> CREATE TABLE Departments (
    ->     department_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     department_name VARCHAR(25) NOT NULL
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql> DESCRIBE Departments;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| department_id   | int         | NO   | PRI | NULL    | auto_increment |
| department_name | varchar(25) | NO   |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

mysql> INSERT INTO Departments (department_name) VALUES
    ->     ('Science'),
    ->     ('Engineering'),
    ->     ('Arts');
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Departments;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             1 | Science         |
|             2 | Engineering     |
|             3 | Arts            |
+---------------+-----------------+
3 rows in set (0.00 sec)

mysql> CREATE TABLE Programs (
    ->     program_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     program_name VARCHAR(30) NOT NULL,
    ->     department_id INT,
    ->     total_credit_points_required INT,
    ->     year_commenced INT,
    ->     FOREIGN KEY (department_id) REFERENCES Departments(department_id)
    -> );
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> DESCRIBE PROGRAMS;
+------------------------------+-------------+------+-----+---------+----------------+
| Field                        | Type        | Null | Key | Default | Extra          |
+------------------------------+-------------+------+-----+---------+----------------+
| program_id                   | int         | NO   | PRI | NULL    | auto_increment |
| program_name                 | varchar(30) | NO   |     | NULL    |                |
| department_id                | int         | YES  | MUL | NULL    |                |
| total_credit_points_required | int         | YES  |     | NULL    |                |
| year_commenced               | int         | YES  |     | NULL    |                |
+------------------------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> INSERT INTO Programs (program_name, department_id, total_credit_points_required, year_commenced) VALUES
    ->     ('Bachelor of Science', 1, 120, 2019),
    ->     ('Bachelor of Engineering', 2, 150, 2018),
    ->     ('Bachelor of Arts', 3, 90, 2020);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Programs;
+------------+------------------------+---------------+------------------------------+----------------+
| program_id | program_name           | department_id | total_credit_points_required | year_commenced |
+------------+------------------------+---------------+------------------------------+----------------+
|          1 | Bachelor of Science    |             1 |                          120 |           2019 |
|          2 | Bachelor of Engineering |             2 |                          150 |           2018 |
|          3 | Bachelor of Arts       |             3 |                           90 |           2020 |
+------------+------------------------+---------------+------------------------------+----------------+
3 rows in set (0.00 sec)

mysql> CREATE TABLE Courses (
    ->     course_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     course_name VARCHAR(30) NOT NULL,
    ->     program_id INT,
    ->     credit_point_value INT,
    ->     year_commenced INT,
    ->     FOREIGN KEY (program_id) REFERENCES Programs(program_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> DESCRIBE courses;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| course_id          | int         | NO   | PRI | NULL    | auto_increment |
| course_name        | varchar(30) | NO   |     | NULL    |                |
| program_id         | int         | YES  | MUL | NULL    |                |
| credit_point_value | int         | YES  |     | NULL    |                |
| year_commenced     | int         | YES  |     | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> DESCRIBE courses;
+--------------------+-------------+------+-----+---------+----------
| Field              | Type        | Null | Key | Default | Extra
+--------------------+-------------+------+-----+---------+----------
| course_id          | int         | NO   | PRI | NULL    | auto_incre
| course_name        | varchar(30) | NO   |     | NULL    |
| program_id         | int         | YES  | MUL | NULL    |
| credit_point_value | int         | YES  |     | NULL    |
| year_commenced     | int         | YES  |     | NULL    |
+--------------------+-------------+------+-----+---------+----------
5 rows in set (0.00 sec)

mysql> INSERT INTO Courses (course_name, program_id, credit_point_valu
    ->     ('Mathematics', 1, 5, 2019),
    ->     ('Physics', 1, 5, 2019),
    ->     ('Computer Science', 2, 4, 2018),
    ->     ('Electrical Engineering', 2, 4, 2018),
    ->     ('History', 3, 3, 2020),
    ->     ('Literature', 3, 3, 2020);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Courses;
+-----------+------------------------+------------+-------------------
| course_id | course_name            | program_id | credit_point_value
+-----------+------------------------+------------+-------------------
|         1 | Mathematics            |          1 |                 5
|         2 | Physics                |          1 |                 5
|         3 | Computer Science       |          2 |                 4
|         4 | Electrical Engineering |          2 |                 4
|         5 | History                |          3 |                 3
|         6 | Literature             |          3 |                 3
+-----------+------------------------+------------+-------------------
6 rows in set (0.00 sec)

mysql> CREATE TABLE Students (
    ->     student_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     first_name VARCHAR(30) NOT NULL,
    ->     last_name VARCHAR(25) NOT NULL,
    ->     date_of_birth DATE,
    ->     year_enrolled INT
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE Students;
+---------------+-------------+------+-----+---------+----------------
| Field         | Type        | Null | Key | Default | Extra
+---------------+-------------+------+-----+---------+----------------
| student_id    | int         | NO   | PRI | NULL    | auto_increment
| first_name    | varchar(30) | NO   |     | NULL    |
| last_name     | varchar(25) | NO   |     | NULL    |
| date_of_birth | date        | YES  |     | NULL    |
| year_enrolled | int         | YES  |     | NULL    |
+---------------+-------------+------+-----+---------+----------------
```

```
mysql> DESCRIBE Students;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| student_id    | int         | NO   | PRI | NULL    | auto_increment |
| first_name    | varchar(30) | NO   |     | NULL    |                |
| last_name     | varchar(25) | NO   |     | NULL    |                |
| date_of_birth | date        | YES  |     | NULL    |                |
| year_enrolled | int         | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> INSERT INTO Students ( first_name, last_name, date_of_birth, year_enrolled)
    -> VALUES
    ->     ('Nisha', 'Elizabeth', '1992-02-15', 2018),
    ->     ('Alice', 'Koshy', '2000-03-20', 2019),
    ->     ('Emi', 'Johnson', '2001-07-10', 2020),
    -> ('Mathew', 'Luke', '2000-03-20', 2020),
    ->  ('Alice', 'Koshy', '2001-08-25', 2018),
    -> ('Anu', 'Mariam', '2000-03-20', 2018);
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Students;
+------------+------------+-----------+---------------+---------------+
| student_id | first_name | last_name | date_of_birth | year_enrolled |
+------------+------------+-----------+---------------+---------------+
|          1 | Nisha      | Elizabeth | 1992-02-15    |          2018 |
|          2 | Alice      | Koshy     | 2000-03-20    |          2019 |
|          3 | Emi        | Johnson   | 2001-07-10    |          2020 |
|          4 | Mathew     | Luke      | 2000-03-20    |          2020 |
|          5 | Alice      | Koshy     | 2001-08-25    |          2018 |
|          6 | Anu        | Mariam    | 2000-03-20    |          2018 |
+------------+------------+-----------+---------------+---------------+
6 rows in set (0.00 sec)

mysql> CREATE TABLE Enrollments (
    ->     enrollment_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     student_id INT,
    ->     program_id INT,
    ->     FOREIGN KEY (student_id) REFERENCES Students(student_id),
    ->     FOREIGN KEY (program_id) REFERENCES Programs(program_id)
    -> );
Query OK, 0 rows affected (0.10 sec)

mysql> DESCRIBE Enrollments;
+---------------+------+------+-----+---------+----------------+
| Field         | Type | Null | Key | Default | Extra          |
+---------------+------+------+-----+---------+----------------+
| enrollment_id | int  | NO   | PRI | NULL    | auto_increment |
| student_id    | int  | YES  | MUL | NULL    |                |
| program_id    | int  | YES  | MUL | NULL    |                |
+---------------+------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

```
mysql> DESCRIBE Enrollments;
+---------------+------+------+-----+---------+----------------+
| Field         | Type | Null | Key | Default | Extra          |
+---------------+------+------+-----+---------+----------------+
| enrollment_id | int  | NO   | PRI | NULL    | auto_increment |
| student_id    | int  | YES  | MUL | NULL    |                |
| program_id    | int  | YES  | MUL | NULL    |                |
+---------------+------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> INSERT INTO Enrollments (student_id, program_id)
    -> VALUES
    ->     (1, 1),
    ->     (2, 2),
    ->     (3, 1);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Enrollments;
+---------------+------------+------------+
| enrollment_id | student_id | program_id |
+---------------+------------+------------+
|             1 |          1 |          1 |
|             2 |          2 |          2 |
|             3 |          3 |          1 |
+---------------+------------+------------+
3 rows in set (0.00 sec)

mysql> CREATE TABLE CourseRegistrations (
    ->     registration_id INT AUTO_INCREMENT PRIMARY KEY,
    ->     student_id INT,
    ->     course_id INT,
    ->     year_attempted INT,
    ->     semester_attempted INT,
    ->     grade VARCHAR(2),
    ->     mark DECIMAL(5, 2),
    ->     FOREIGN KEY (student_id) REFERENCES Students(student_id),
    ->     FOREIGN KEY (course_id) REFERENCES Courses(course_id)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE CourseRegistrations;
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| registration_id    | int          | NO   | PRI | NULL    | auto_increment |
| student_id         | int          | YES  | MUL | NULL    |                |
| course_id          | int          | YES  | MUL | NULL    |                |
| year_attempted     | int          | YES  |     | NULL    |                |
| semester_attempted | int          | YES  |     | NULL    |                |
| grade              | varchar(2)   | YES  |     | NULL    |                |
| mark               | decimal(5,2) | YES  |     | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

```
mysql> INSERT INTO CourseRegistrations (student_id, course_id, year_attempted, semester_attempted, grade, mark)
    -> VALUES
    ->     (1, 1, 2023, 1, 'A+', 90.5),
    ->     (2, 6, 2023, 2, 'B+', 78.0),
    ->     (3, 3, 2023, 1, 'A', 88.0),
    ->     (4, 5, 2023, 2, 'A+', 92.5),
    ->     (5, 4, 2023, 1, 'A', 81.5),
    ->     (6, 2, 2023, 2, 'B', 65.8);
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM CourseRegistrations;
+-----------------+------------+-----------+----------------+--------------------+-------+-------+
| registration_id | student_id | course_id | year_attempted | semester_attempted | grade | mark  |
+-----------------+------------+-----------+----------------+--------------------+-------+-------+
|               7 |          1 |         1 |           2023 |                  1 | A+    | 90.50 |
|               8 |          2 |         6 |           2023 |                  2 | B+    | 78.00 |
|               9 |          3 |         3 |           2023 |                  1 | A     | 88.00 |
|              10 |          4 |         5 |           2023 |                  2 | A+    | 92.50 |
|              11 |          5 |         4 |           2023 |                  1 | A     | 81.50 |
|              12 |          6 |         2 |           2023 |                  2 | B     | 65.80 |
+-----------------+------------+-----------+----------------+--------------------+-------+-------+
6 rows in set (0.00 sec)
```

2. Alter the above tables with Foreign Keys as required for the scenario. And perform the following activities.

a. List all students in a specific department (e.g., Computer Science)

```
mysql> UPDATE Students
    -> SET department_id =  1
    -> ;
Query OK, 4 rows affected (0.03 sec)
Rows matched: 6  Changed: 4  Warnings: 0

mysql> SELECT * FROM Students;
+------------+------------+-----------+---------------+---------------+---------------+
| student_id | first_name | last_name | date_of_birth | year_enrolled | department_id |
+------------+------------+-----------+---------------+---------------+---------------+
|          1 | Nisha      | Elizabeth | 1992-02-15    |          2018 |             1 |
|          2 | Alice      | Koshy     | 2000-03-20    |          2019 |             1 |
|          3 | Emi        | Johnson   | 2001-07-10    |          2020 |             1 |
|          4 | Mathew     | Luke      | 2000-03-20    |          2020 |             1 |
|          5 | Alice      | Koshy     | 2001-08-25    |          2018 |             1 |
|          6 | Anu        | Mariam    | 2000-03-20    |          2018 |             1 |
+------------+------------+-----------+---------------+---------------+---------------+
6 rows in set (0.00 sec)

mysql> ALTER TABLE Students
    -> ADD CONSTRAINT FK_Students_Departments
    -> FOREIGN KEY (department_id)
    -> REFERENCES Departments(department_id);
Query OK, 6 rows affected (0.18 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Students;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| student_id    | int         | NO   | PRI | NULL    | auto_increment |
| first_name    | varchar(30) | NO   |     | NULL    |                |
| last_name     | varchar(25) | NO   |     | NULL    |                |
| date_of_birth | date        | YES  |     | NULL    |                |
| year_enrolled | int         | YES  |     | NULL    |                |
| department_id | int         | YES  | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> SELECT Students.student_id, Students.first_name, Students.last_name
    -> FROM Students
    -> INNER JOIN Departments ON Students.department_id = Departments.department_id
    -> WHERE Departments.department_name = 'Science';
+------------+------------+-----------+
| student_id | first_name | last_name |
+------------+------------+-----------+
|          1 | Nisha      | Elizabeth |
|          2 | Alice      | Koshy     |
|          3 | Emi        | Johnson   |
|          4 | Mathew     | Luke      |
|          5 | Alice      | Koshy     |
|          6 | Anu        | Mariam    |
+------------+------------+-----------+
6 rows in set (0.00 sec)
```

b. List all courses a specific student (e.g., with Student ID 123) is enrolled in

```
mysql> SELECT Courses.course_id, Courses.course_name
    -> FROM Courses
    -> JOIN CourseRegistrations ON Courses.course_id = CourseRegistrations.course_id
    -> WHERE CourseRegistrations.student_id = 1;
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|         1 | Mathematics |
+-----------+-------------+
1 row in set (0.00 sec)
```

c. Retrieve the total number of students enrolled in each course

```
mysql> SELECT Courses.course_id, Courses.course_name, COUNT(CourseRegistrations.student_id) AS EnrollmentCount
    -> FROM Courses
    -> LEFT JOIN CourseRegistrations ON Courses.course_id = CourseRegistrations.course_id
    -> GROUP BY Courses.course_id, Courses.course_name;
+-----------+-----------------------+-----------------+
| course_id | course_name           | EnrollmentCount |
+-----------+-----------------------+-----------------+
|         1 | Mathematics           |               1 |
|         2 | Physics               |               1 |
|         3 | Computer Science      |               1 |
|         4 | Electrical Engineering |               1 |
|         5 | History               |               1 |
|         6 | Literature            |               1 |
+-----------+-----------------------+-----------------+
6 rows in set (0.03 sec)
```

d. Find the course with the highest enrolment

```
mysql> SELECT Courses.course_id, Courses.course_name, COUNT(CourseRegistrations.student_id) AS EnrollmentCount
    -> FROM Courses
    -> LEFT JOIN CourseRegistrations ON Courses.course_id = CourseRegistrations.course_id
    -> GROUP BY Courses.course_id, Courses.course_name
    -> ORDER BY EnrollmentCount DESC
    -> LIMIT 1;
+-----------+-------------+-----------------+
| course_id | course_name | EnrollmentCount |
+-----------+-------------+-----------------+
|         1 | Mathematics |               1 |
+-----------+-------------+-----------------+
1 row in set (0.00 sec)

mysql> SELECT Courses.course_id, Courses.course_name, COUNT(CourseRegistrations.student_id) AS EnrollmentCount
    -> FROM Courses
    -> LEFT JOIN CourseRegistrations ON Courses.course_id = CourseRegistrations.course_id
    -> GROUP BY Courses.course_id, Courses.course_name
    -> ORDER BY EnrollmentCount DESC
    -> LIMIT 2;
+-----------+-------------+-----------------+
| course_id | course_name | EnrollmentCount |
+-----------+-------------+-----------------+
|         1 | Mathematics |               1 |
|         2 | Physics     |               1 |
+-----------+-------------+-----------------+
2 rows in set (0.00 sec)
```

e. List all Programs of a particular department

```
mysql>
mysql> SELECT program_id, program_name
    -> FROM Programs
    -> WHERE department_id = (
    ->     SELECT department_id
    ->     FROM Departments
    ->     WHERE department_name = 'Science'
    -> );
+------------+---------------------+
| program_id | program_name        |
+------------+---------------------+
|          1 | Bachelor of Science |
+------------+---------------------+
1 row in set (0.02 sec)
```

f. List all courses under a specific program.

```
mysql> SELECT course_id, course_name
    -> FROM Courses
    -> WHERE program_id = (
    ->     SELECT program_id
    ->     FROM Programs
    ->     WHERE program_name = 'Bachelor of Science'
    -> );
+-----------+-------------+
| course_id | course_name |
+-----------+-------------+
|         1 | Mathematics |
|         2 | Physics     |
+-----------+-------------+
2 rows in set (0.00 sec)
```

3. The data to be recorded according to the requirements are as follows.

Staff Id, Name, Designation, staff address, staff email, staff phone No, salary, branch Id, branch Description, Branch Address, branch phone No

a. Normalize the data and form the tables accordingly.

Q. StaffId, Name, Designation, Staff Address, staff email, staff phone, salary, branch Id, branch Description, Branch Address, branch phone No.

I Staff Table :
Staff Id (Primary key)
Name
Designation
Staff Address
Staff Email
Staff Phone No
Salary
Branch Id (Foreign Key)

I Branch Table :
Branch Id (primary key)
Branch Description
Branch Address
Branch phoneNo

Effiwently organiee Data, we can separate the values for given entities into two Separate Tables :-
Staff & Branch.

Here, Each column having atomic values
No repeating groups.

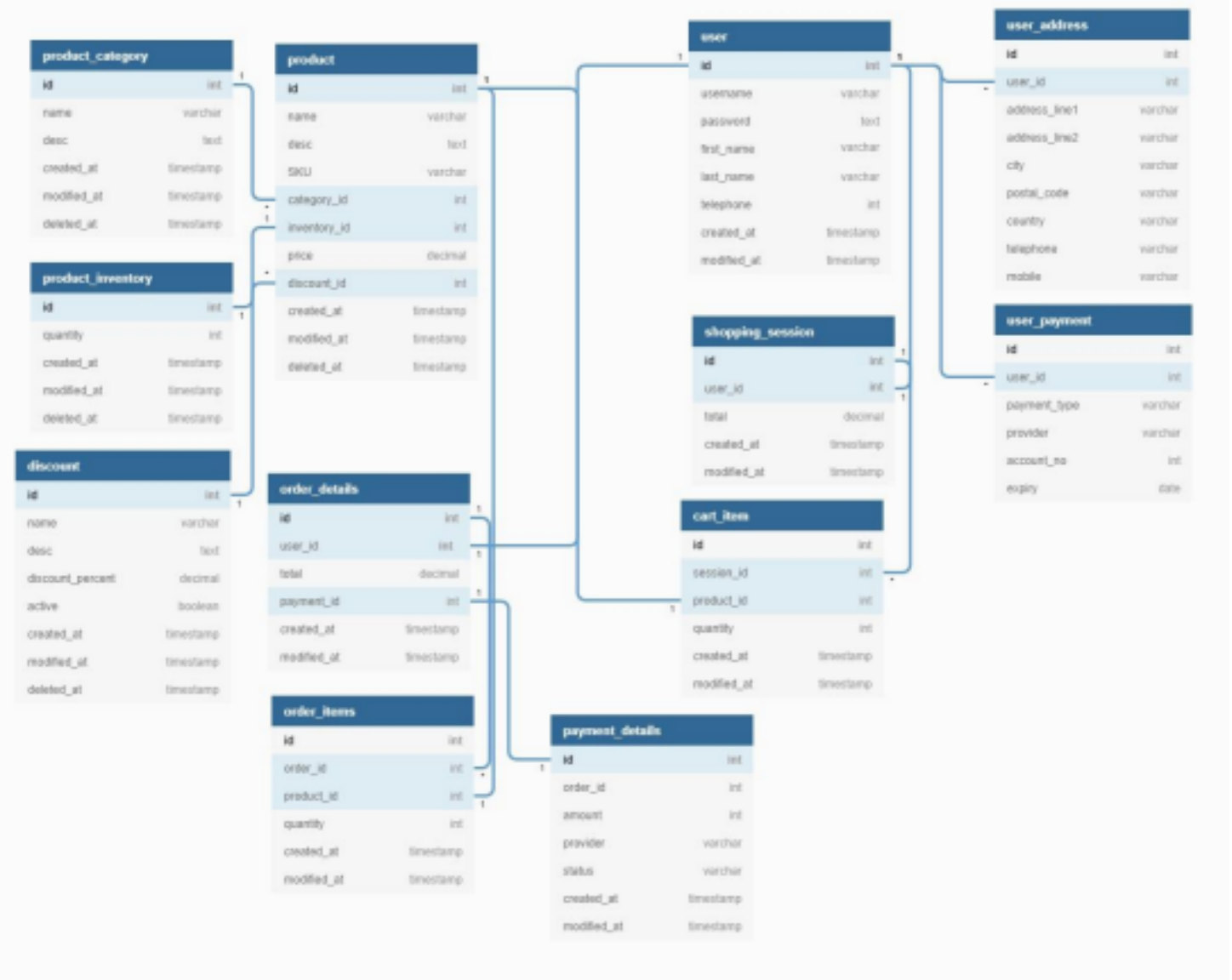
Now it is 3NF form,
Staff Table with Staff related Attributes
Branch Table with branch related Attributes

b. Create the normalized tables, with required relationships.

```
mysql> CREATE TABLE Branch (
    ->     BranchId INT PRIMARY KEY,
    ->     Description VARCHAR(255),
    ->     Address VARCHAR(255),
    ->     PhoneNo VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE Staff (
    ->     StaffId INT PRIMARY KEY,
    ->     Name VARCHAR(255),
    ->     Designation VARCHAR(255),
    ->     Address VARCHAR(255),
    ->     Email VARCHAR(255),
    ->     PhoneNo VARCHAR(20),
    ->     Salary DECIMAL(10, 2),
    ->     BranchId INT,
    ->     FOREIGN KEY (BranchId) REFERENCES Branch(BranchId)
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE Branch;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| BranchId    | int          | NO   | PRI | NULL    |       |
| Description | varchar(255) | YES  |     | NULL    |       |
| Address     | varchar(255) | YES  |     | NULL    |       |
| PhoneNo     | varchar(20)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM  Branch;
Empty set (0.00 sec)

mysql> DESCRIBE Staff;
+-------------+---------------+------+-----+---------+-------+
| Field       | Type          | Null | Key | Default | Extra |
+-------------+---------------+------+-----+---------+-------+
| StaffId     | int           | NO   | PRI | NULL    |       |
| Name        | varchar(255)  | YES  |     | NULL    |       |
| Designation | varchar(255)  | YES  |     | NULL    |       |
| Address     | varchar(255)  | YES  |     | NULL    |       |
| Email       | varchar(255)  | YES  |     | NULL    |       |
| PhoneNo     | varchar(20)   | YES  |     | NULL    |       |
| Salary      | decimal(10,2) | YES  |     | NULL    |       |
| BranchId    | int           | YES  | MUL | NULL    |       |
+-------------+---------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```


c. Perform all CRUD operations on these tables.

4 Create the following tables in a database “eShopping”



a. Insert data in all the tables.

b. Display the details of a Product, including the following data

   a. Product Id, Name, desc, Category name, quantity, discount percentage (simple join)

c. Display the details about an order to include the following data

   a. order ID, first name, product name, quantity. (inner join)

d. Display the following details

   a. First Name, product name(use left join)

e. Display the following data

   a. First name, product name (use right join)

f. Display the following data

   a. First name, product name (use full outer join)