**Name: Shumaila**

**ID: 6827**

**Department: BS (CS)**

**Semester: 8th**

**Course: Database System**

**Q1: Perform Normalization upto 3rd Normal Form on the following table. (13marks)**

Question # 01

Answer :-

Solution:- 1st Normal form:

| Studene ID | Student Name | Students Address | Course ID | Course Name | Grade |
|---|---|---|---|---|---|
| 01 | Farward | Karachi | SE - 01 | AI | A |
| 01 | farward | Karachi | SE-05 | SQE | B |
| 02 | waleed | Lahore | SE - 02 | DIP | C |
| 03 | Saira | Peshawar | SE - 03 | DB | A |
| 03 | Saira | peshawar | SE - 04 | SRE | B |
| 04 | Aiman | Karachi | SE-03 | DB | C |
| 05 | Daniyal | Lahore | SE - 01 | AI | A |
| 06 | Emaan | Peshawar | SE-01 | AI | B. |

# 2nd Normal form:-

=> Student ID -> Student Name, Student Address

=> Course - ID -> Course - Name

=> SID -> Course ID, Grade

S-ID -> S-Name, S-Add.

Grade [Key]

C-ID -> C-Name

| Student ID | Student Name | Student Address |
|---|---|---|
| 01 | Fawad | Karachi |
| 02 | waleed | Lahore |
| 03 | Saira | Peshawar |
| 04 | Aiman | Karachi |
| 05 | Daniyal | Lahore |
| 06 | Emaan | Peshawar |

=> Student Relation

| Course ID | Course Name |
|---|---|
| SE-01 | AI |
| SE-05 | SQE |
| SE-02 | DID |
| SE-03 | DB |
| SE-04 | SRE |
| SE-03 | DB |
| SE-01 | AI |
| SE-01 | AI |

=> Course - Relation

| Student ID | Course ID | Grade |
|---|---|---|
| 01 | SE-01 | A |
| 01 | SE-05 | B |
| 02 | SE-02 | C |
| 03 | SE-03 | A |
| 03 | SE-04 | B |
| 04 | SE-03 | C |
| 05 | SE-01 | A |
| 06 | SE-01 | B |

⇒ Registration Relation

# 3rd Normal form :-

⇒ Student ID ⇒ Student name, Student As we do previous in 2nd Normal form, also Similarly we can find it from in 2nd Normal form to 3rd normal.

⇒ Course ID → Course Name, Student ID.

⇒ Student ID → Course ID, Grade

[ Composition key (Student ID, Course ID)]

| Student ID | Student Name | Student Address |
|---|---|---|
| 01 | Fawad | Karachi |
| 02 | waleed | Lahore |
| 03 | Saira | Peshawar |
| 04 | Aimaar | Karachi |
| 05 | Daniyal | Lahore |
| 06 | Emaan | Peshawar |

3NF

| Student I.D | Course ID | Grade |
|---|---|---|
| 01 | SE-01 | A |
| 01 | SE-05 | B |
| 02 | SE-02 | C |
| 03 | SE-03 | A |
| 03 | SE-04 | B |
| 04 | SE-03 | C |
| 05 | SE-01 | A |
| 06 | SE-01 | B |

3NF

| Course-ID | Course-Name | Student-ID |
|---|---|---|
| SE-01 | AI | 01 |
| SE-05 | SRE | 01 |
| SE-02 | DIB | 02 |
| SE-03 | DB | 03 |
| SE-04 | SRE | 03 |
| SE-03 | DB | 04 |
| SE-01 | AI | 05 |
| SE-01 | AI | 06 |

here i's explained

⇒ The table in 2nd form Course-ID - Course Name has the data Redundancy So there was 2NF

**Q2: Write SQL queries for the following DDL Statements**

**1. Create a Database by the name Gallery**

```
CREATE DATABASE Gallery;
```

**2. Write a query to create a table by the name Movies which should have the following columns and restrictions:**

```
CREATE TABLE MOVIES(

ID INT

PRIMARY KEY Movie_Name varchar(50) NOT NULL,

Genre varchar(20)

Year INT,

Rating INT,

CHECK(year<=2020),

CHECK(rating<=5));
```

# Question no 3

**Part 1 : Write 2 SQL DML Queries to insert your data and your friend’s data in this Table.**

```
INSERT INTO University_table (Student_id, Student_name, age, cgpa)
VALUES ('45899', 'ali', '24' '2.9');


INSERT INTO University_table (Student_id, Student_name, age, cgpa)
VALUES ('79466', 'ahmad', '25' '3.0');
```

**2. Write SQL DML Query to delete all students' record whose CGPA is greater 3**

```
DELETE FROM Table

WHERE CGPA>3;
```

# Question no 4

**Consider you have the following 2 tables.**

**1. Write SQL Query for finding/displaying product names and ids of products whose unit price is less than 50 Rs.**

```
SELECT All.

  FROM  [Cateen_table]. [Order details]

  WHERE pro_price < 50

 ORDER BY pro_price DESC, pro_name;
```

**2. Write SQL Query for displaying sorted names of product names with Alias name as Product_List_Sorted.**

```
SELECT ALL

ORDER BY Product_name DESC
```

**3. Write output of the following query (5 Marks)**

```
SELECT Category, COUNT(Category)

 FROM Canteen_Table

GROUP BY Category

HAVING COUNT(Category) > 1;
```

Canteen_table

Category

1

2    1

3

**4.Write SQL INNER JOIN query and its output on the given two tables.**

The INNER JOIN selects all rows from both participating tables as long as there is a match between the columns. An SQL INNER JOIN is same as JOIN clause, combining rows from two or more tables.

### Syntax:

```
SELECT *
FROM table1 INNER JOIN table2
ON table1.column_name = table2.column_name;
```

## or

```
SELECT *
FROM table1
JOIN table2
ON table1.column_name = table2.column_name;
```