

**Name: Shumaila**

**ID: 6827**

**Department: BS (CS)**

**Semester: 8th**

**Course: Database System**

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

(1)  
Question # 01

Answer :-

Solution :- 1st Normal form:

Student ID	Student Name	Students Address	Course ID	Course Name	Grade
01	Faewad	Karachi	SE-01	AI	A
01	faewad	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

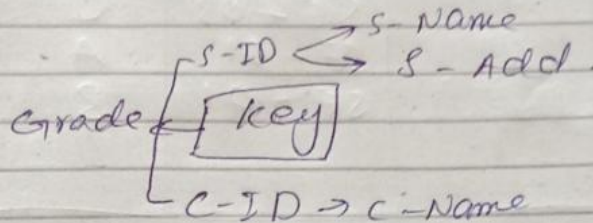
(2)

## 2<sup>nd</sup> Normal form:-

⇒ Student ID → Student Name, Student Address

⇒ Course → ID → Course Name

⇒ SID → Course ID, Grade



Student ID	Student Name	Student Address	Course ID	Course Name
01	Fawad	Karachi	SE-01	AI
02	waleed	Lahore	SE-05	SQE
03	Saira	Reskawas	SE-02	DIID
04	Aiman	Karachi	SE-03	DB
			SE-04	SRE
05	Daniyal	Lahore	SE-03	DB
			SE-01	AI
06	Emaan	Reskawas	SE-01	AI

⇒ Student Relation

⇒ Course - Relation



(3)

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

(4)

### 3<sup>rd</sup> Normal form :-

⇒ Student ID → Student Name,  
Student As we do previous  
in 2<sup>nd</sup> Normal form, also  
Similarly we can find it  
from in 2<sup>nd</sup> Normal  
form to 3<sup>rd</sup> normal.

⇒ Course ID → Course Name,  
Student ID.

⇒ Student ID → Course ID, Grade

[ Composition key (Student ID,  
Course ID) ]

Student ID	Student Name	Student Address	Student ID	Course ID	Grade
01	Fawad	Karachi	01	SE-01	A
02	Waleed	Lahore	01	SE-05	B
03	Saira	Peshawar	02	SE-02	C
04	Amaan	Karachi	03	SE-03	A
05	Daniyal	Lahore	03	SE-04	B
06	Emaan	Peshawar	04	SE-03	C
			05	SE-01	A
			06	SE-01	B

3NF

3NF



(5)

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  
is 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;
```