

Name: Aftab Khan ID:12985 Subject: Database System

Question 1:

Perform Normalization up-to 3rd Normal Form on the following table.

page (1)

Given Table

Student ID	Student Name	Student Address	Course ID	Course Name	Grade
01	Fawad	Karachi	SE-01	AI	A
			SE-05	SQE	B
02	Waliq	Lahore	SE-02	DIP	C
03	Saira	Peshawar	SE-03	DB	A
			SE-04	SRE	B
04	Aiman	Karachi	SE-03	DB	C
05	Danzal	Lahore	SE-01	AI	A
06	Emaan	Peshawar	SE-01	AI	B

Q=1 => Perform Normalization up to 3rd Normal form on the given table?

Sol:- 1st Normal Form:-

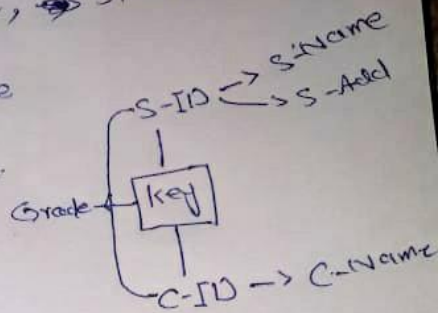
Student ID	Student Name	Student Address	Course ID	Course Name	Grade
01	Fawad	Karachi	SE-01	AI	A
01	Fawad	Karachi	SE-05	SQE	B
02	Waliq	Lahore	SE-02	DIP	C
03	Saira	Peshawar	SE-03	DB	A
03	Saira	Peshawar	SE-04	SRE	B
04	Aiman	Karachi	SE-03	DB	C
05	Danzal	Lahore	SE-01	AI	A
06	Emaan	Peshawar	SE-01	AI	B

⇒ 2nd Normal form :-

⇒ Student ID → Student Name, Student Address.

⇒ Course_ID → Course_Name

⇒ S-ID → Course_ID, Grade.



Student ID	Student Name	Student Address
01	Fahad	Iskachi
02	Mateel	Lahore
03	Saira	Peshawar
04	Aiman	Karachi
05	Daniyal	Lahore
06	Emaan	Peshawar

⇒ Student Relation.

Course ID	Course Name
SE-01	AI
SE-05	SDE
SE-02	DB
SE-03	DB
SE-04	SDE
SE-03	DB
SE-01	AI
SE-01	AI

⇒ Course Relation

⇒ Registration 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-03	A
05	SE-01	B
06	SE-01	B

=> 3rd Normal Form :-

=> Student ID -> Student Name, Student Address
 As we do previous in 2nd Normal form, also, similarly we can find it from the 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	Student ID	Course ID	Grade
01	Fakhrul	Karachi	01	SE-01	A
02	Muhammad	Lahore	01	SE-05	B
03	Saif	Peshawar	02	SE-02	C
04	Ahmad	Karachi	03	SE-03	A
05	Daniyal	Lahore	03	SE-04	B
06	Amaan	Peshawar	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	SE	01
SE-02	DB	02
SE-03	SE	03
SE-04	DB	03
SE-03	AI	04
SE-01	AI	05
SE-01	AI	06

=> here is the optional

3NF

=> Note => The Table in 2nd form - course ID - course name has the data redundancy so that was 2NF.

Question 2: 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(30) NOT NULL,  
  
Genre varchar(15),  
  
Year int,  
  
Rating int,  
  
CHECK(Year<=2020),  
  
CHECK (Rating<=5)  
  
);
```

Explanation:

Since we have set ID to be a primary key, it is automatically NOT NULL.

Question 3:

1. Write 2 SQL DML Queries to insert your data and your friend's data in this Table.

```
INSERT INTO Table (Student_Id,Student_Name, Age, CGPA)  
VALUES ('4001','Jason Roy','18','7');
```

```
INSERT INTO Table (Student_Id,Student_Name, Age, CGPA)  
VALUES ('4002','Steven Smith','17','3');
```

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

```
DELETE FROM Table WHERE CGPA > 3;
```

Question 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.

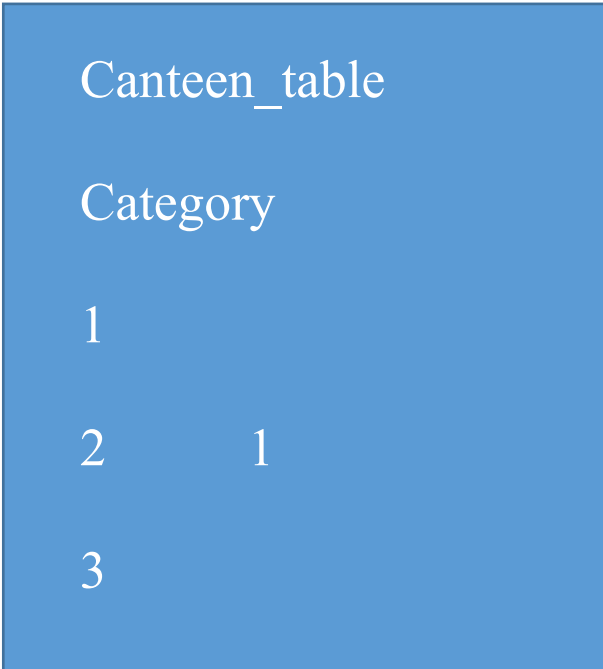
```
SELECT All.  
  
FROM [Cateen_table]. [Order details]  
  
WHERE pro_price < 50  
  
ORDER BY pro_price DESC, pro_name;
```

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.

```
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;
```