Haroon Rashid

Registration No# 16549

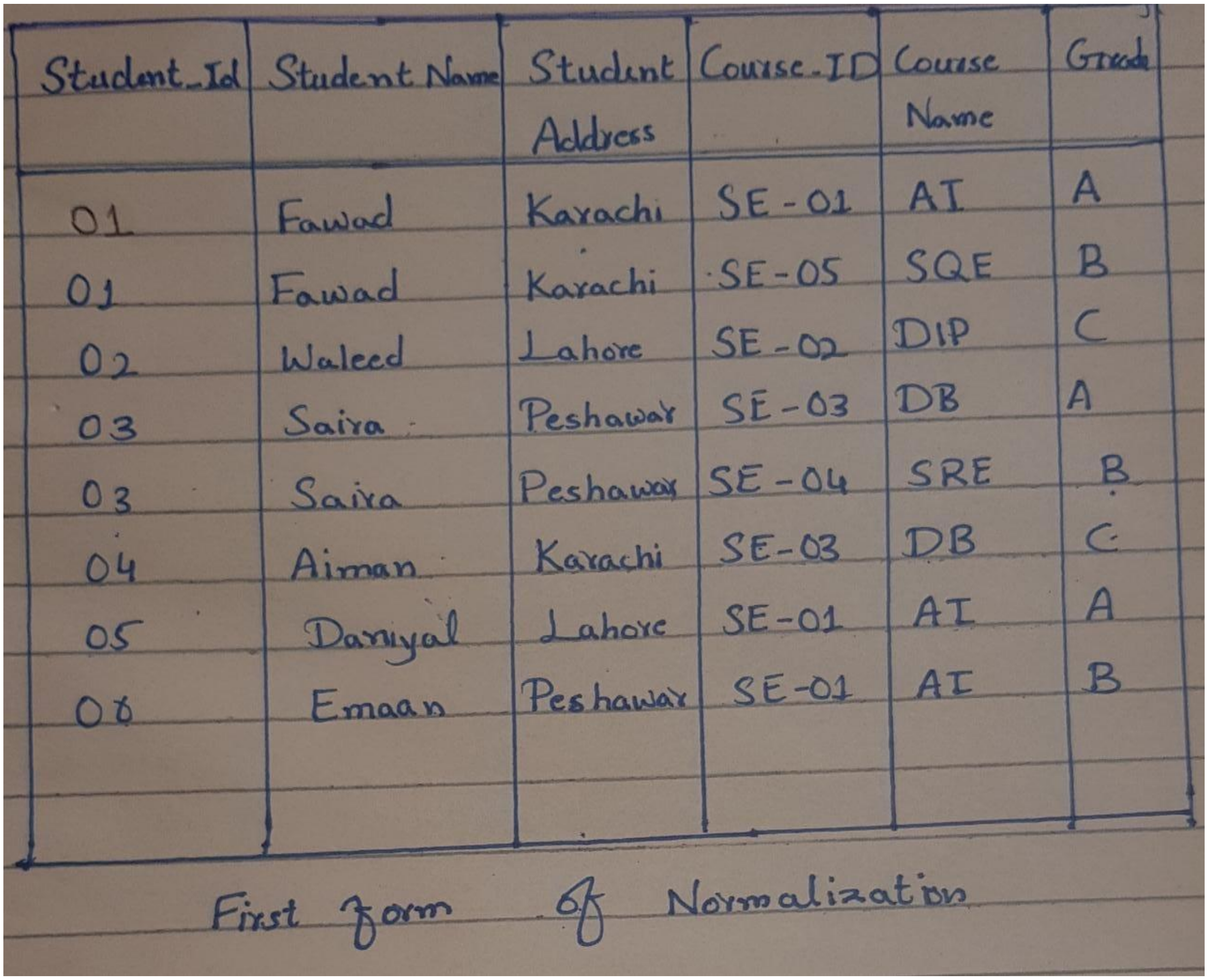
Semester: 6th

Final Assignment: Database Systems(T)

Submitted to: Madam Rimsha Khan

Q1: Perform Normalization upto 3rd Normal Form on the following table.

Answer:



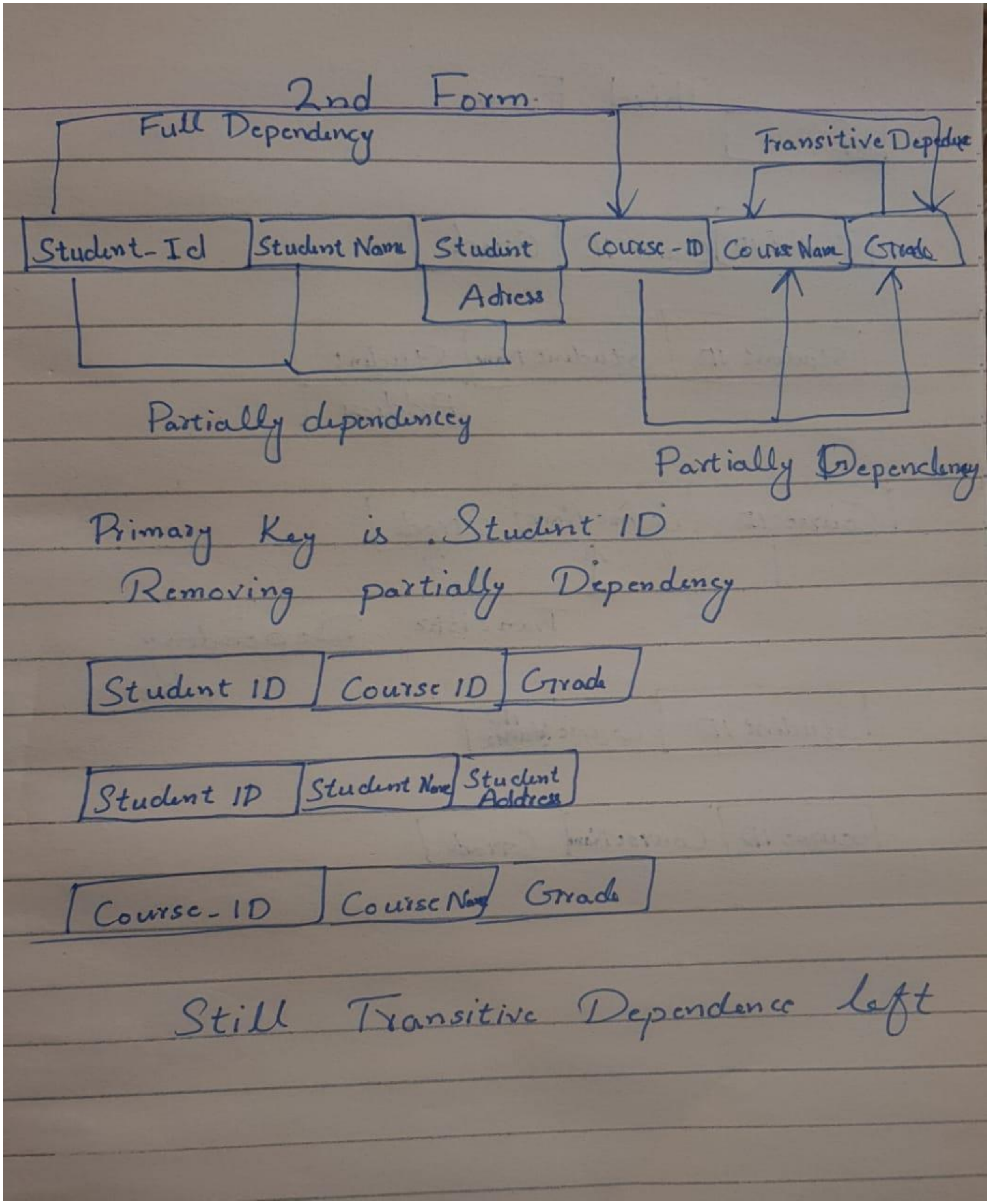
Student_Id	Student Name	Student Address	Course-ID	Course Name	Grade
01	Fawad	Karachi	SE-01	AI	A
01	Fawad	Karachi	SE-05	SQE	B
02	Waleed	Lahore	SE-02	DIP	C
03	Saira	Peshawar	SE-03	DB	A
03	Saira	Peshawar	SE-04	SRE	B
04	Aiman	Karachi	SE-03	DB	C
05	Daniyal	Lahore	SE-01	AI	A
06	Emaan	Peshawar	SE-01	AI	B

First form of Normalization

Anomalies in this Table:

- **Insertion–if new a new course of existing Student added, course data must be re-entered, causing duplication**
- **Deletion–if we delete the SQE 01, we lose information concerning this item's finish and price**
- **Update–changing the Grade of Student ID 01 requires update in several records**

2^ND Normalization Form:



2nd Form

Full Dependency
Transitive Depedue

Student-Id	Student Name	Student Adress	Course-ID	Course Name	Grade

Partially dependencey
Partially Dependency

Primary Key is .Student ID

Removing partially Dependency

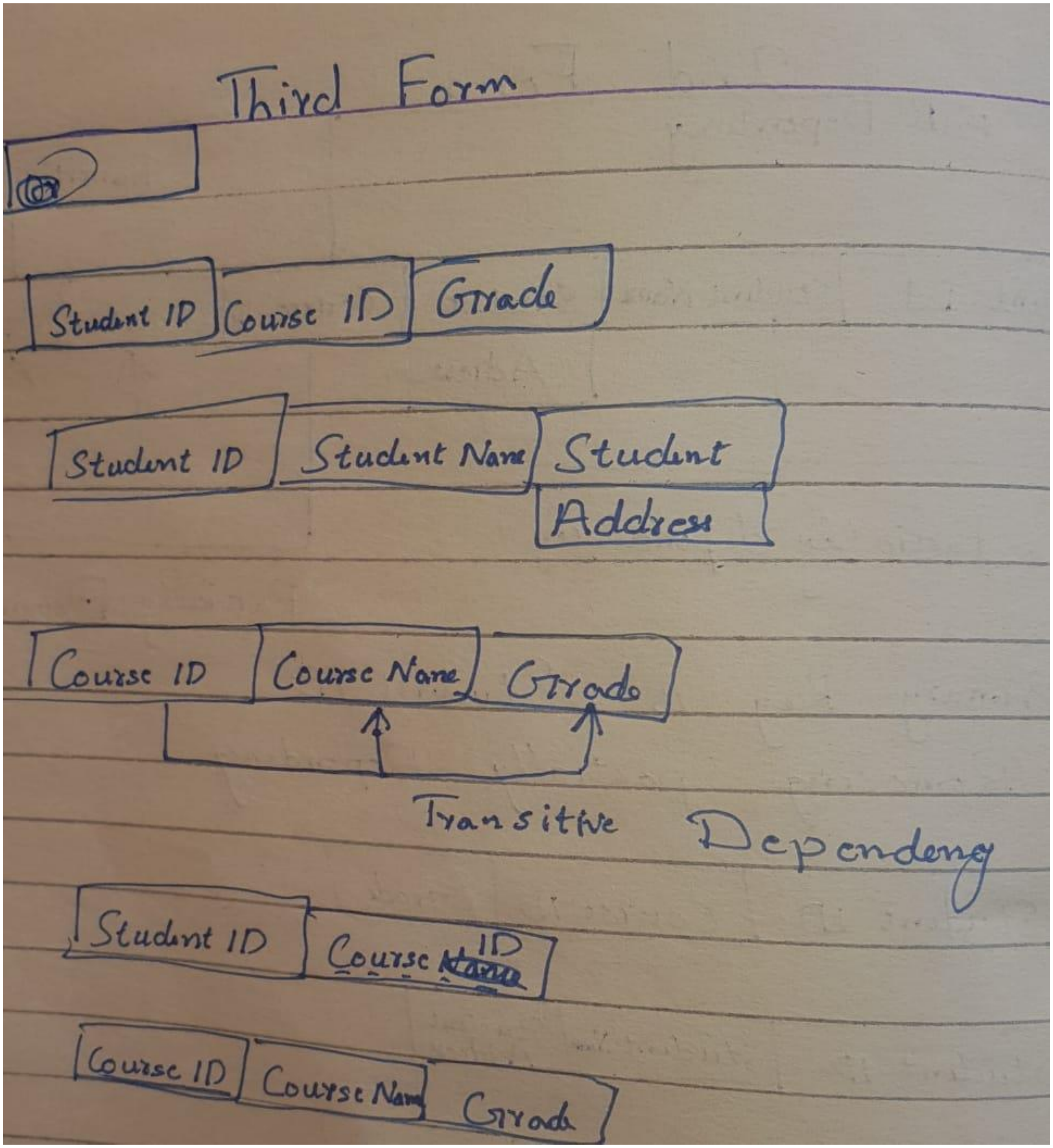
Student ID	Course ID	Grade

Student ID	Student Name	Student Address

Course-ID	Course Name	Grade

Still Transitive Dependence left

3RD Normalization Form:



Third Form

Student ID	Course ID	Grade

Student ID	Student Name	Student Address

Course ID	Course Name	Grado

Transitive Dependeng

Student ID	Course ~~Name~~ ID

Course ID	Course Nam	Grad

Atfer Removed Transitive Dependency Course_ID, Course name, Grade

Q2: Write SQL queries for the following DDL Statements

1. Create a Database by the name Gallery

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

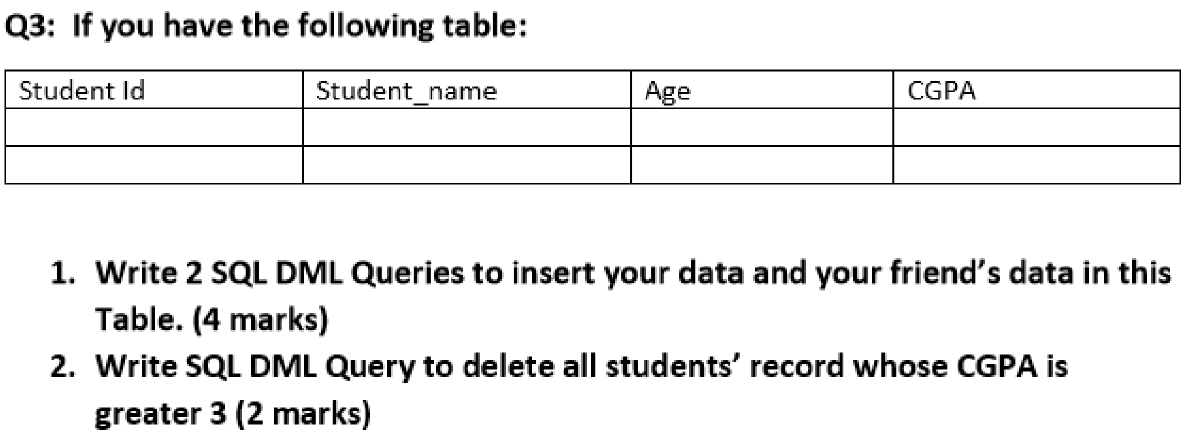
Column Name: ID Type: integer Column Name: Movie Name Type: varchar Column Name: Genre Type: varchar

Column Name: Year Type: integer Column Name: Rating Type: integer Restrictions: ID should be the primary key and NOT NULL. Movie Name should also be NOT NULL. Year should have a maximum value of 2020 and rating should have a maximum value of 5.

```
Answer: Part(1): CREATE DATABASE Gallery;
```

Part(2):

```
 create table "Movie"(

ID INT PK_ID Primary Key NOT Null,
Movie_Name varchar(100) NOT Null,

Genre varchar(250),

Year Integer
Select MAX(Year)
  From[year]
Where Year(2020)

Rating Integer
Select MAX(Rating)
  From[Rating]
Where Rating(5)

);
```



Q3: If you have the following table:

Student Id	Student_name	Age	CGPA

1. **Write 2 SQL DML Queries to insert your data and your friend's data in this Table. (4 marks)**
2. **Write SQL DML Query to delete all students' record whose CGPA is greater 3 (2 marks)**

Answer:
Part1:

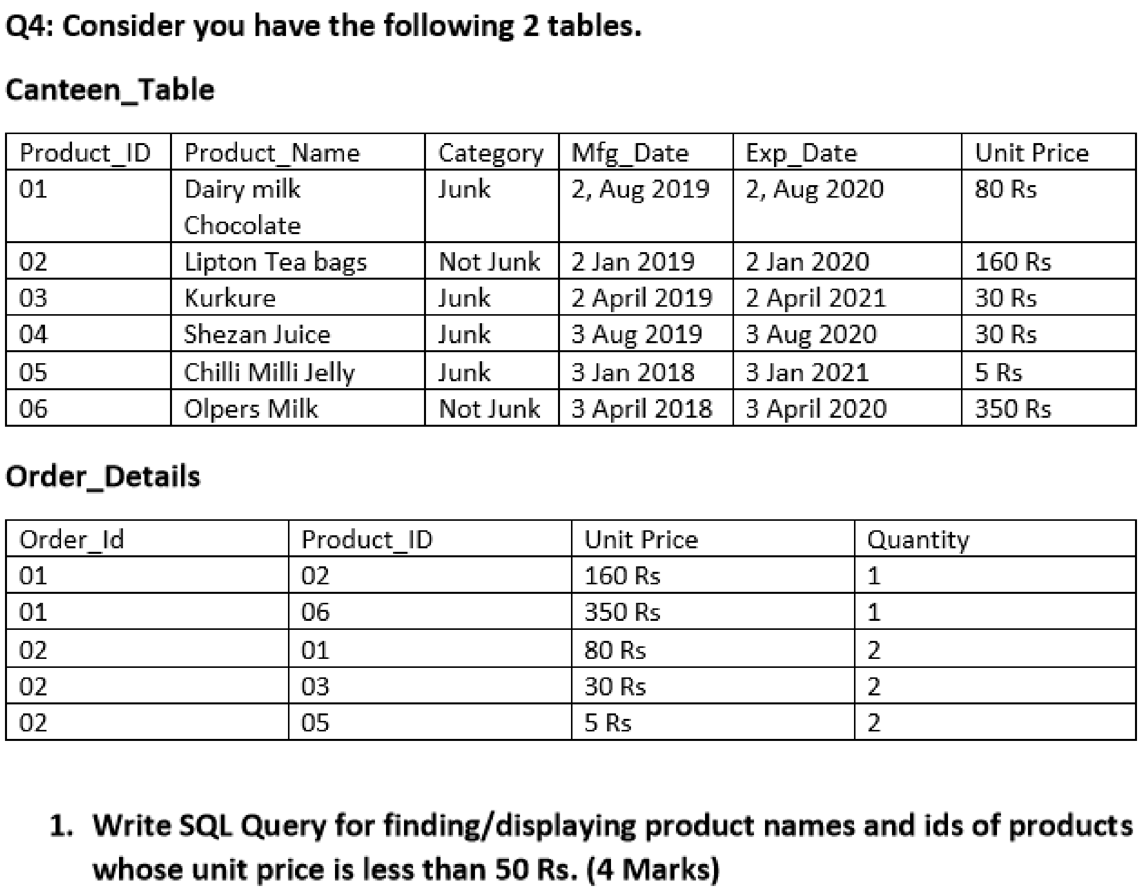
```
insert into (Student_id,Student_name,Age,CGPA)

      values (16549,'Haroon Rashid',21,3.8)
```

```
insert into (Student_id,Student_name,Age,CGPA)

      values (16660,'Arman Khan',22,3.5)
```

Part2:

```
delete from Student where CGPA Is > 3,
```

Q4 Is below:



Q4: Consider you have the following 2 tables.

Canteen_Table

Product_ID	Product_Name	Category	Mfg_Date	Exp_Date	Unit Price
01	Dairy milk Chocolate	Junk	2, Aug 2019	2, Aug 2020	80 Rs
02	Lipton Tea bags	Not Junk	2 Jan 2019	2 Jan 2020	160 Rs
03	Kurkure	Junk	2 April 2019	2 April 2021	30 Rs
04	Shezan Juice	Junk	3 Aug 2019	3 Aug 2020	30 Rs
05	Chilli Milli Jelly	Junk	3 Jan 2018	3 Jan 2021	5 Rs
06	Olpers Milk	Not Junk	3 April 2018	3 April 2020	350 Rs

Order_Details

Order_Id	Product_ID	Unit Price	Quantity
01	02	160 Rs	1
01	06	350 Rs	1
02	01	80 Rs	2
02	03	30 Rs	2
02	05	5 Rs	2

1. **Write SQL Query for finding/displaying product names and ids of products whose unit price is less than 50 Rs. (4 Marks)**

Answer:

Part1:

Canteen table:

```
 SELECT Product_name, Product_ID, UNIT_price

     FROM Canteen_Table
    WHERE _Unit_Price < 50
```

For Order:

```
SELECT ORDER_ID, Product_ID, UNIT PRICE

     FROM Canteen_Table
    WHERE _Unit_Price < 50
```

Part2: Write SQL Query for displaying sorted names of product names with Alias name as Product_List_Sorted.

Answer: Part2: In Ascending: By default, the sort is performed in ascending order. Therefore, there is no need to specify the **ASC** keyword.

```
SELECT ProductName AS [Product_Name]
FROM Canteen table;
```

3. Write output of the following query

```
SELECT Category, COUNT(Category)
 FROM Canteen_Table GROUP BY Category HAVING COUNT(Category) > 1;
```

```
Answer:  Category count

     Not junk    2

Product:
Olpers Milk
Lipton Tea bag

    junk    4
product:
Kurkure
Dairy milk Chocolate
Shezan Juice
Chilli Milli Jelly
```

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

Answer:

```
SELECT Canteen_Table.Product_ID, Canteen_Table.Product_Name,
Order,order_id, Order,order Quantity,

order.order_ID,order.order_quantity
FROM Canteen_Table
INNER JOIN Order_Details
ON Canteen_Table. order_id =order.order_id;
```

```
Canteen_Table.Product_ID  Canteen_Table.Product_Name order_ID order quantity
   01                  Diary Milk Chocolate    01                 1
   02                  Lepton tea bags         01                 1
   03                  Kurkure                 02                 2
```

```

------------------------ ----- ------------------------- -------
-------
```