UNIVERSITI TEKNOLOGI MALAYSIA

SEMESTER 1, SESSION 2023/2024

PROJECT PHASE 3

DATABASE LOGICAL DESIGN & SQL

SECD2523 : DATABASE

SECTION 10

GROUP NAME : **THOUSAND SUNNY**

1.	NUR IRDINA SYAFIQAH BINTI ABU BAKAR	B23CS0069
2.	MOHAMAD HAIRIL BIN ZAINAL	B23CS0045
3.	AHMAD AFIF AISY BIN AHMAD RIZAL	B23CS0018
4.	MUHAMMAD AFIQ KASYFI BIN NOR AZELAN	B23CS0049

LECTURER'S NAME : MADAM ROZILAWATI BINTI DOLLAH

TABLE OF CONTENTS

1.0 INTRODUCTION

For Project Phase 3, the Thousand Sunny team has gathered a comprehensive data set that includes all the important information needed for the construction of the TS Mart system. Based on the invaluable insights gained from our group's careful investigations during Project Phases 1 and 2, a concerted effort was made to refine and improve the envisioned system based on the insightful feedback gained through thorough reviews conducted by our group members.

This report serves as an extensive documentation and explanation of the complex process involved in the development of the TS Mart system. It will delve into an in-depth exploration of conceptual Entity Relationship Diagrams (ERDs) carefully designed to encapsulate the fundamentals of system structure. Furthermore, the report will explain the intricacies of the logical design of the database, explaining the thoughtful considerations and decisions made in shaping the system's underlying data architecture.

Finally, the report will begin the exploration of the practical application of SQL statements in the construction phase of the TS Mart system. This section will not only highlight the technical intricacies of SQL implementation but also explain the strategic decisions made in aligning system construction with the overall goals of the project.

2.0 OVERVIEW OF PROJECT

Our project is about providing services for purchasing groceries and customer needs through online platform. To improve customers shopping experience, TS Mart System improve e-commerce by allowing customers to buy items from more than one store or multiple destinations with just one payment instead of making separate payments from each shop.

In this phase, we will be updating of what we have done in phase 2 which is business rules, conceptual ERD, enhanced ERD, and logical enhanced ERD. Logical ERD should be finalised to do normalization process. Finally, SQL statements are required to create a complete system.

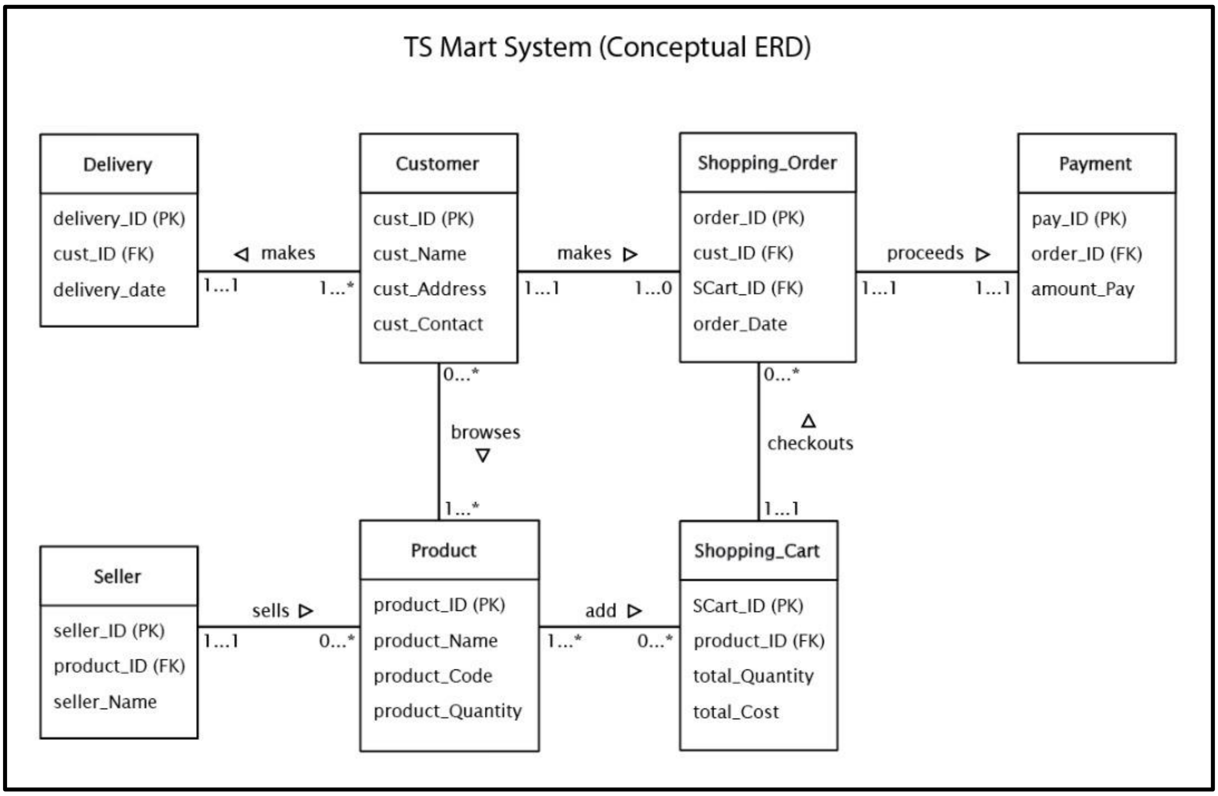
3.0 DATABASE CONCEPTUAL DESIGN

3.1 UPDATED BUSINESS RULE

TS Mart System business rules:

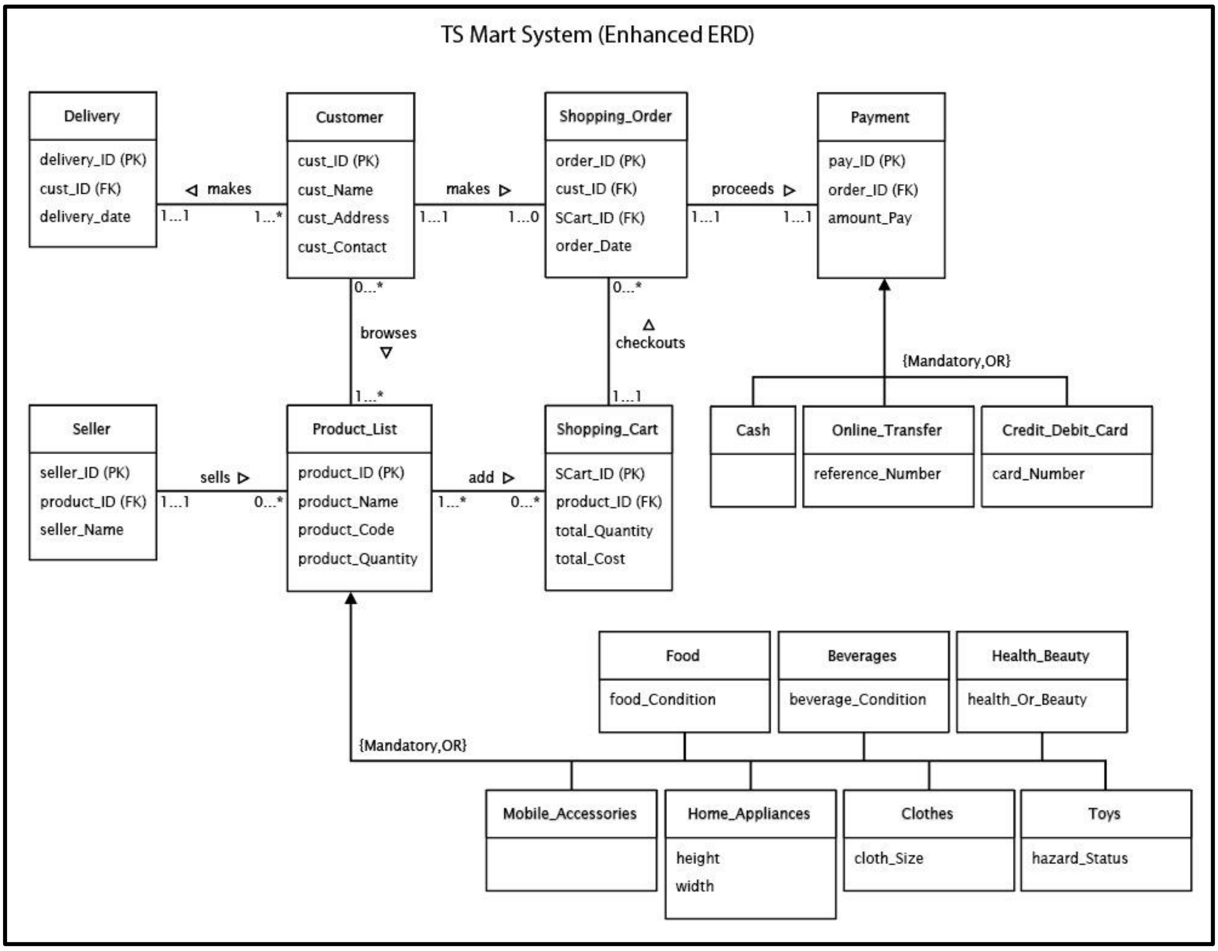
1. The company will start operating at 8am to 10 pm.
2. Display only available grocery products and update if an item is out of stock.
3. Customers can start their orders for delivery when the operating hours have started at the same time.
4. Shows the customer's total payment accurately, including if there are taxes applied to help customers plan their spending.
5. Give feedback or notifications in real time to customers when they have finished paying for their order.
6. Allow customers to make special requests when placing orders or make changes to cancel orders within a certain period before the order is delivered.
7. Provide various payment methods for customers to pay for orders such as credit or debit card, e-wallet, online transfer, and also Cash-On-Delivery (COD).
8. Ensure that the online payment process carried out by customers is safe to protect the personal and financial data of customers who use the TS Mart System.
9. Offer a delivery time according to the user's wishes.
10. Provide an easily accessible customer support channel for grocery related inquiries and issues to ensure a responsive and helpful customer service experience.
11. Protect customer rights and information by complying to all e-commerce laws and regulations, including privacy and data protection policies.
12. Ensure that the TS Mart System is optimised for various platforms, including websites, mobile phones, and tablets, to deliver an effortless and responsive user experience.

3.2 CONCEPTUAL ERD



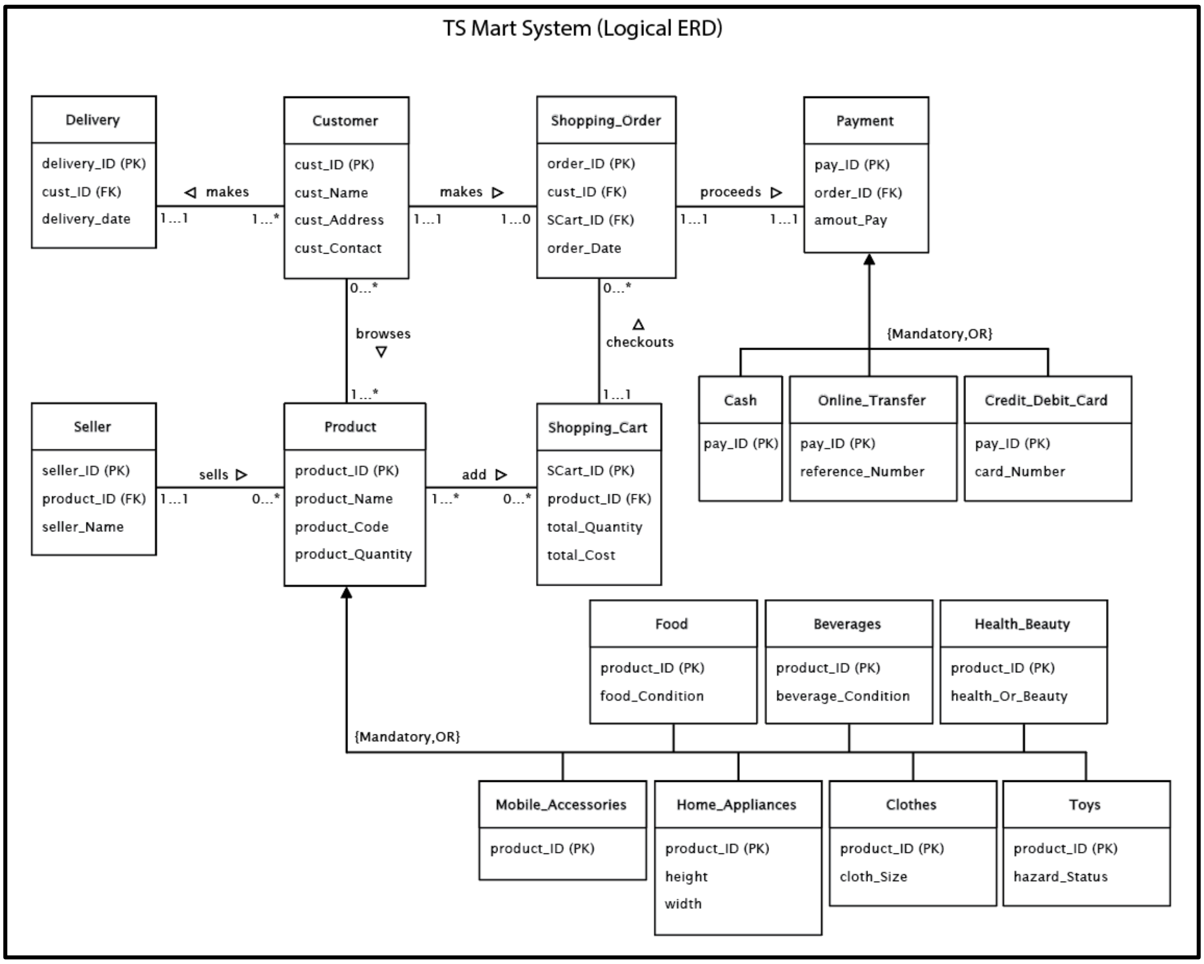
4.0 DATABASE LOGICAL DESIGN

Database logical design is the process of transforming a conceptual schema of the application domain into a schema for the data model of a specific database management system, such as relational or object-oriented. It involves creating tables, columns, keys, and constraints that can capture and distinguish all valid states of the conceptual schema, while also addressing issues related to performance, storage, and security.



4.1 LOGICAL ERD

A logical ERD is a diagram that includes the details of entities, their attributes, and the relationships among them. It is used to refine the database design and communicate with technical stakeholders.



4.2 UPDATED DATA DICTIONARY

Entity Name	Attributes	Data Type & Length	Description	Primary Key	Foreign Key
Customer	cust_ID	Decimal (10)	Identifier for the customer order ID	YES	NO
	cust_Name	Varchar2 (50)	Name of the customer	NO	NO
	cust_Address	Varchar2 (50)	Address of the customer	NO	NO
	cust_Contact	Varchar2 (15)	The contact number of the customer	NO	NO
Product	product_ID	Decimal (20)	Identifier for the product	YES	NO
	category_ID	Decimal (10)	Identifier for the category	NO	YES
	product_Name	Varchar (50)	Name of the product	NO	NO
Delivery	delivery_ID	Decimal (20)	Identifier for the delivery ID	YES	NO
	cust_ID	Decimal (10)	Identifier for the customer order	NO	YES
	delivery_date	Decimal (10)	Date of the delivery	NO	NO
Seller	seller_ID	Decimal (10)	Identifier for the seller ID	YES	NO

	product_ID	Decimal (20)	Identifier for the product	NO	YES
	seller_Name	Varchar2 (50)	Name of the product seller	NO	NO
Shopping_Order	order_ID	Decimal (20)	Identifier for the order	YES	NO
	cust_ID	Decimal (10)	Identifier for the customer order ID	NO	YES
	SCart_ID	Decimal (20)	Identifier for the shopping cart	NO	YES
	order_date	Decimal (10)	Date of the order	NO	NO
Shopping_Cart	SCart_ID	Decimal (20)	Identifier for the shopping cart	YES	NO
	product_ID	Decimal (20)	Identifier for the product	NO	YES
	total_Quantity	Decimal (1000)	Quantity of the order	NO	NO
	total_Cost	Decimal (1000)	Total cost of the order	NO	NO
Cash	pay_ID	Decimal (10)	Identifier for the order payment	YES	NO
	order_ID	Decimal (10)	Identifier for the order	NO	YES
	amount_pay	Double (1000)	Amount paid to the seller	NO	NO
Online_Transfer	pay_ID	Decimal (10)	Identifier for the order payment	YES	NO

	order_ID	Decimal (10)	Identifier for the order	NO	YES
	amount_pay	Double (1000)	Amount paid to the seller	NO	NO
	reference_Number	Decimal (15)	Reference number of the online transfer	NO	NO
Credit_Debit_Card	pay_ID	Decimal (10)	Identifier for the order payment	YES	NO
	order_ID	Decimal (10)	Identifier for the order	NO	YES
	amount_pay	Double (1000)	Amount paid to the seller	NO	NO
	card_Number	Decimal (20)	The number of the debit/credit card	NO	NO
Food	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO
	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	food_Condition	Varchar (20)	The condition of the food product	NO	NO
Beverages	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO

	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	beverage_Condition	Varchar (20)	The condition of the beverage product	NO	NO
Health_Beauty	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO
	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	health_Or_Beauty	Varchar (6)	The status whether it is for health or beauty	NO	NO
Mobile_Accessories	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO
	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
Home_Appliances	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO

	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	height	Double (1000)	The height of the home product	NO	NO
	width	Double (1000)	The width of the home product	NO	NO
Clothes	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO
	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	cloth_Size	Varchar (10)	The size of the cloth product	NO	NO
Toys	product_ID	Decimal (20)	Identifier for the product	YES	NO
	product_Name	Varchar (50)	Name of the product	NO	NO
	product_Code	Varchar (10)	The code of the product	NO	NO
	product_Quantity	Decimal (100)	The quantity of the product	NO	NO
	hazard_Status	Varchar (3)	The exist of hazard from the toy product	NO	NO

4.3 NORMALIZATION

Normalization in a database is a technique for creating database tables with suitable columns and keys by decomposing a large table into smaller logical units. It eliminates redundant and non-descriptive attributes and reduces data anomalies and inconsistency. Normalization is an iterative process that follows a series of steps or normal forms, each with specific rules and criteria.

***PK** stands for Primary Key, **FK** stands for Foreign Key, and **FD** stands for Functionality Dependency.

****Remark:** The underlined word is Primary Key.

1. **Customer (**cust_ID, cust_Name, cust_Address, cust_Contact**)**
   FD1: cust_ID ► cust_Name, cust_Address, cust_Contact (Primary Key)
   1NF, 2NF, 3NF, & BCNF:
   Customer (cust_ID, cust_Name, cust_Address, cust_Contact)
   PK: cust_ID

2. **Delivery (**delivery_ID, cust_ID, delivery_date**)**
   FD1: delivery_ID, cust_ID ► delivery_date
   1NF, 2NF, 3NF, & BCNF:
   Delivery (delivery_ID, cust_ID, delivery_date)
   PK: delivery_ID
   FK: cust_ID

3. **Product (**product_ID, product_Name, product_Code, product_Quantity**)**
   FD1: product_ID ► product_Name, product_Code, product_Quantity
   INF, 2NF, 3NF, & BCNF:
   Product (product_ID, product_Name, product_Code, product_Quantity)
   PK: product_ID

4. **Food (**product_ID, product_Name, product_Code, product_Quantity, food_Condition**)**

   FD1: product_ID ► food_Condition

   FD2: product_ID ► product_Name, product_Code, product_Quantity

   1NF:

   Food (product_ID, food_Condition)

   Product (product_ID, product_Name, product_Code, product_Quantity)

   2NF, 3NF, & BCNF:

   Food (product_ID, food_Condition)

   Product (product_ID, product_Name, product_Code, product_Quantity)

   PK: product_ID

5. **Beverages (**product_ID, product_Name, product_Code, product_Quantity, beverage_Condition**)**

   FD1: product_ID ► beverage_Condition

   FD2: product_ID ► product_Name, product_Code, product_Quantity

   1NF:

   Beverages (product_ID, beverage_Condition)

   Product (product_ID, product_Name, product_Code, product_Quantity)

   2NF, 3NF, & BCNF:

   Beverages (product_ID, beverage_Condition)

   Product (product_ID, product_Name, product_Code, product_Quantity)

   PK: product_ID

6. **Health_Beauty** (product_ID, product_Name, product_Code, product_Quantity, health_Or_Beauty)

   FD1: product_ID ► health_Or_Beauty

   FD2: product_ID ► product_Name, product_Code, product_Quantity

   1NF:

   Health_Beauty (product_ID, health_Or_Beauty)

   Product (product_ID, product_Name, product_Code, product_Quantity)

   2NF, 3NF, & BCNF:

   Health_Beauty (product_ID, health_Or_Beauty)

Product (product_ID, product_Name, product_Code, product_Quantity)

PK: product_ID

7. **Mobile_Accessories (**product_ID, product_Name, product_Code, product_Quantity**)**

FD1: product_ID (Primary Key)

FD2: product_ID ► product_Name, product_Code, product_Quantity

1NF:

Mobile_Accessories (product_ID)

Product (product_ID, product_Name, product_Code, product_Quantity)

2NF, 3NF, & BCNF:

Mobile_Accessories (product_ID)

Product (product_ID, product_Name, product_Code, product_Quantity)

PK: product_ID

8. **Home_Appliances** (product_ID, product_Name, product_Code, product_Quantity, height, width)

FD1: product_ID ► height, width

FD2: product_ID ► product_Name, product_Code, product_Quantity

1NF:

Home_Appliances (product_ID, height, width)

Product (product_ID, product_Name, product_Code, product_Quantity)

2NF, 3NF, & BCNF:

Home_Appliances (product_ID, height, width)

Product (product_ID, product_Name, product_Code, product_Quantity)

PK: product_ID

9. **Clothes** (product_ID, product_Name, product_Code, product_Quantity, cloth_Size)

FD1: product_ID ► cloth_Size

FD2: product_ID ► product_Name, product_Code, product_Quantity

1NF:

Cloth (product_ID, cloth_Size)

Product (product_ID, product_Name, product_Code, product_Quantity)

2NF, 3NF, & BCNF:

Cloth (product_ID, cloth_Size)

Product (product_ID, product_Name, product_Code, product_Quantity)

PK: product_ID

10. **Toys** (product_ID, product_Name, product_Code, product_Quantity, hazard_Status)

    FD1: product_ID ► hazard_Status

    FD2: product_ID ► product_Name, product_Code, product_Quantity

    1NF:

    Toys (product_ID, hazard_Status)

    Product (product_ID, product_Name, product_Code, product_Quantity)

    2NF, 3NF, & BCNF:

    Toys (product_ID, hazard_Status)

    Product (product_ID, product_Name, product_Code, product_Quantity)

    PK: product_ID

11. **Seller** (seller_ID, product_ID, seller_Name)

    FD1: seller_ID, product_ID ► seller_Name

    1NF, 2NF, 3NF, & BCNF:

    Delivery (seller_ID, product_ID, seller_Name)

    PK: seller_ID

    FK: product_ID

12. **Shopping_Cart (**SCart_ID, product_ID, total_Quantity, total_Cost**)**

    FD1: SCart_ID, product_ID ► total_Quantity, total_Cost

    1NF, 2NF, 3NF, & BCNF:

    Delivery (SCart_ID, product_ID, total_Quantity, total_Cost)

    PK: SCart_ID

    FK: product_ID

13. **Shopping_Order (**order_ID, cust_ID, SCart_ID, order_Date**)**

    FD1: order_ID, cust_ID, SCart_ID ► order_Date

    1NF, 2NF, 3NF, & BCNF:

    Shopping_Order (order_ID, cust_ID, SCart_ID, order_Date)

    PK: order_ID

    FK: cust_ID, SCart_ID

14. **Payment (**pay_ID, order_ID, amount_Pay)

    FD1: pay_ID ► order_ID, amount_Pay

    INF, 2NF, 3NF, & BCNF:

    Payment (pay_ID, order_ID, amount_Pay)

    PK: pay_ID

15. **Cash (**pay_ID, order_ID, amount_Pay**)**

    FD1: pay_ID (Primary Key)

    FD2: pay_ID ► order_ID, amount_Pay

    1NF:

    Cash (pay_ID)

    Payment (pay_ID, order_ID, amount_Pay)

    2NF, 3NF, & BCNF:

    Cash (pay_ID)

    Payment (pay_ID, order_ID, amount_Pay)

    PK: pay_ID

16. **Online_Transfer (**pay_ID, order_ID, amount_Pay, reference_Number**)**

    FD1: pay_ID ► reference_Number

    FD2: pay_ID ► order_ID, amount_Pay

    1NF:

    Online_Transfer (pay_ID, reference_Number)

    Payment (pay_ID, order_ID, amount_Pay)

    2NF, 3NF, & BCNF:

Online_Transfer (pay_ID, reference_Number)

Payment (pay_ID, order_ID, amount_Pay)

PK: pay_ID

17. **Credit_Debit_Card (**pay_ID, order_ID, amount_Pay, card_Number**)**

FD1: pay_ID ► card_Number

FD2: pay_ID ► order_ID, amount_Pay

1NF:

Credit_Debit_Card (pay_ID, card_Number)

Payment (pay_ID, order_ID, amount_Pay)

2NF, 3NF, & BCNF:

Credit_Debit_Card (pay_ID, card_Number)

Payment (pay_ID, order_ID, amount_Pay)

PK: pay_ID

5.0 RELATIONAL DB SCHEMAS

Delivery	(delivery_ID, cust_ID, delivery_date)
Customer	(cust_ID, cust_Name, cust_Address, cust_Contact)
Shopping_Order	(order_ID, cust_ID, SCart_ID, order_date)
Payment	(pay_ID, order_ID, amount_pay)
Cash	(pay_ID)
Online_Transfer	(pay_ID, reference_Number)
Credit_Debit_Card	(pay_ID, card_Number)
Product	(product_ID, product_Name, product_Code, product_Quantity)
Shopping_Cart	(SCart_ID, product_ID, total_Quantity, total_Cost)
Seller	(seller_ID, product_ID, seller_Name)
Mobile_Accessories	(product_ID)
Food	(product_ID, food_Condition)
Home_Appliances	(product_ID, height, width)
Beverages	(product_ID, beverage_Condition)
Clothes	(product_ID, cloth_Size)
Health_Beauty	(product_ID, health_Or_Beauty)
Toys	(product_ID, hazard_Status)

Delivery

delivery_ID	cust_ID	delivery_date

Customer

cust_ID	cust_Name	cust_Address	cust_Contact

Shopping Order

order_ID	cust_ID	SCart_ID	order_Date

Payment

pay_ID	order_ID	amount_pay

Cash

pay_ID

Online_Transfer

pay_ID, reference_Number

Credit_Debit_Card

pay_ID, card_Number

Product

product_ID	product_Name	product_Code	product_Quantity

Shopping_Cart

SCart_ID	produt_ID	total_Quantity	total_Cost

Seller

seller_ID	product_ID	seller_Name

Mobile_Accessories

product_ID

Food

product_ID	food_Condition

Home_Appliances

product_ID	height	width

Beverages

product_ID	beverage_Condition

Clothes

product_ID	cloth_Size

Health_Beauty

product_ID	health_Or_Beauty

Toys

product_ID	hazard_Status

6.0 SQL STATEMENTS (DDL & DML)

6.1 DATA DEFINITION LANGUAGE (DDL)

1. **Create Customer table**

```
CREATE TABLE Customer (
   cust_ID DECIMAL(10) PRIMARY KEY,
   cust_Name VARCHAR(50) NOT NULL,
   cust_Address VARCHAR(50) NOT NULL,
   cust_Contact VARCHAR(15) NOT NULL
);
```

2. **Create Delivery table**

```
CREATE TABLE Delivery (
   delivery_ID DECIMAL(20) PRIMARY KEY,
   cust_ID DECIMAL(10) NOT NULL,
   delivery_date DECIMAL(10) NOT NULL,
   FOREIGN KEY (cust_ID) REFERENCES Customer(cust_ID)
);
```

3. **Create Product table**

```
CREATE TABLE Product (
   product_ID DECIMAL(20) PRIMARY KEY,
   product_Name VARCHAR(50) NOT NULL,
   product_Code DECIMAL(20) NOT NULL,
    product_Quantity DECIMAL(10, 2) NOT NULL
);
```

4. **Create Food table**

```
CREATE TABLE Food (
    product_ID DECIMAL(20) PRIMARY KEY,
    food_Condition VARCHAR(20) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

5. **Create Beverages table**

```
CREATE TABLE Beverages (
    product_ID DECIMAL(20) PRIMARY KEY,
    beverage_Condition VARCHAR(20) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

6. **Create Health_Beauty table**

```
CREATE TABLE Health_Beauty (
    product_ID DECIMAL(20) PRIMARY KEY,
    health_Or_Beauty VARCHAR(10) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

7. Create Mobile_Accessories table

```
CREATE TABLE Mobile_Accessories (

    product_ID DECIMAL(20) PRIMARY KEY,

    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)

);
```

8. Create Home_Appliances table

```
CREATE TABLE Home_Appliances (

    product_ID DECIMAL(20) PRIMARY KEY,

    height DECIMAL(10, 2) NOT NULL,

    width DECIMAL(10, 2) NOT NULL,

    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)

);
```

9. Create Clothes table

```
CREATE TABLE Clothes (

    product_ID DECIMAL(20) PRIMARY KEY,

    cloth_Size VARCHAR(10) NOT NULL,

    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)

);
```

10. Create Toys table

```
CREATE TABLE Toys (

    product_ID DECIMAL(20) PRIMARY KEY,
    hazard_Status VARCHAR(10) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

11. Create Seller table

```
CREATE TABLE Seller (
    seller_ID DECIMAL(10) PRIMARY KEY,
    product_ID DECIMAL(20) NOT NULL,
    seller_Name VARCHAR(50) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

12. Create Shopping_Cart table

```
CREATE TABLE Shopping_Cart (
    SCart_ID DECIMAL(20) PRIMARY KEY,
    product_ID DECIMAL(20) NOT NULL,
    total_Quantity DECIMAL(10, 2) NOT NULL,
    total_Cost DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (product_ID) REFERENCES Product(product_ID)
);
```

13. Create Shopping_Order table

```
CREATE TABLE Shopping_Order (
    order_ID DECIMAL(20) PRIMARY KEY,
    cust_ID DECIMAL(10) NOT NULL,
    SCart_ID DECIMAL(20) NOT NULL,
    order_Date DECIMAL(10) NOT NULL,
    FOREIGN KEY (cust_ID) REFERENCES Customer(cust_ID),
    FOREIGN KEY (SCart_ID) REFERENCES Shopping_Cart(SCart_ID)
);
```

14. Create Payment table

```
CREATE TABLE Payment (
    pay_ID DECIMAL(10) PRIMARY KEY,
    order_ID DECIMAL(10) NOT NULL,
    amount_Pay DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_ID) REFERENCES Shopping_Order(order_ID)
);
```

15. Create Cash table

```
CREATE TABLE Cash (
    pay_ID DECIMAL(10) PRIMARY KEY NOT NULL,
    FOREIGN KEY (pay_ID) REFERENCES Payment(pay_ID)
);
```

16. Create Online_Transfer table

```
CREATE TABLE Online_Transfer (
    pay_ID DECIMAL(10) PRIMARY KEY NOT NULL,
    reference_Number DECIMAL(15) NOT NULL,
    FOREIGN KEY (pay_ID) REFERENCES Payment(pay_ID)
);
```

17. Create Credit_Debit_Card table

```
CREATE TABLE Credit_Debit_Card (
    pay_ID DECIMAL(10) PRIMARY KEY NOT NULL,
    card_Number DECIMAL(20) NOT NULL,
    FOREIGN KEY (pay_ID) REFERENCES Payment(pay_ID)
);
```

6.2 DATA MANIPULATION LANGUAGE (DML)

1. **Insert data into Customer table**

```
INSERT INTO Customer (cust_ID, cust_Name, cust_Address, cust_Contact)
VALUES (1, 'AFIQ', 'BANDAR PUTERI PUCHONG 47100 PUCHONG SELANGOR',
    '0123456789');

INSERT INTO Customer
VALUES (2, 'IRDINA', 'BANDAR BARU SALAK TINGGI 43900 SEPANG SELANGOR',
    '0176678902');

INSERT INTO Customer
VALUES (3, 'AHMAD', 'KAMPUNG GAJAH 36800 PERAK', '0198765432');
```

```
INSERT INTO Customer

VALUES (4, 'HAIRIL', 'KAMPUNG BATU EMBUN 27000 JERANTUT PAHANG',
    '0142345876');


INSERT INTO Customer

VALUES (5, 'MAISARAH', 'KAMPUNG KEDAP 17200 RANTAU PANJANG
    KELANTAN', '0123678890');


INSERT INTO Customer

VALUES (6, 'AMINAH', 'KAMPUNG RAJA 22200 KAMPUNG RAJA TERENGGANU',
    '0189087823');
```

2. Insert data into Delivery table

```
INSERT INTO Delivery (delivery_ID, cust_ID, delivery_date)

VALUES (101, 1, 20231213);


INSERT INTO Delivery

VALUES (102, 2, 20231213);


INSERT INTO Delivery

VALUES (103, 3, 20231220);


INSERT INTO Delivery

VALUES (104, 4, 20231221);
```

```sql
INSERT INTO Delivery
VALUES (105, 5, 20231225);

INSERT INTO Delivery
VALUES (106, 6, 20240108);
```

3. **Insert data into Product table**

```sql
INSERT INTO Product (product_ID, product_Name, product_Code, product_Quantity)
VALUES (1001, 'Laptop', 123456, 10);

INSERT INTO Product
VALUES (1002, 'Smartphone', 789012, 20);

INSERT INTO Product
VALUES (1003, 'Tablet', 345678, 15);

INSERT INTO Product
VALUES (1004, 'Desktop', 901234, 8);

INSERT INTO Product
VALUES (1005, 'Headphones', 567890, 30);

INSERT INTO Product
VALUES (1006, 'Camera', 123789, 12);
```

```sql
INSERT INTO Product

VALUES (1007, 'Printer', 456123, 5);


INSERT INTO Product

VALUES (1008, 'Smartwatch', 789345, 18);


INSERT INTO Product

VALUES (1009, 'External Hard Drive', 234567, 25);


INSERT INTO Product

VALUES (1010, 'Wireless Mouse', 890123, 15);
```

4. **Insert data into Food table**

```sql
INSERT INTO Food (product_ID, food_Condition)

VALUES (1001, 'Fresh');


INSERT INTO Food

VALUES (1002, 'Frozen');


INSERT INTO Food

VALUES (1003, 'Canned');


INSERT INTO Food

VALUES (1004, 'Dried');
```

```
INSERT INTO Food

VALUES (1005, 'Preserved');


INSERT INTO Food

VALUES (1006, 'Raw');


INSERT INTO Food

VALUES(1007, 'Cooked');
```

5. **Insert data into Beverages table**

```
INSERT INTO Beverages (product_ID, beverage_Condition)

VALUES (1001, 'Cooled');


INSERT INTO Beverages

VALUES (1002, 'Room Temperature');


INSERT INTO Beverages

VALUES (1003, 'Hot');


INSERT INTO Beverages

VALUES (1004, 'Iced');


INSERT INTO Beverages

VALUES (1005, 'Carbonated');
```

```
INSERT INTO Beverages

VALUES (1006, 'Non-Alcoholic');


INSERT INTO Beverages

VALUES (1007, 'Freshly Squeezed');
```

6. **Insert data into Health_Beauty table**

```
INSERT INTO Health_Beauty (product_ID, health_Or_Beauty)

VALUES (1001, 'Beauty');


INSERT INTO Health_Beauty

VALUES (1002, 'Health');


INSERT INTO Health_Beauty

VALUES (1003, 'Beauty');


INSERT INTO Health_Beauty

VALUES (1004, 'Beauty');


INSERT INTO Health_Beauty

VALUES (1005, 'Health');


INSERT INTO Health_Beauty

VALUES (1006, 'Beauty');
```

```
INSERT INTO Health_Beauty

VALUES (1007, 'Health');
```

7. **Insert data into Mobile_Accessories table**

```
INSERT INTO Mobile_Accessories (product_ID)

VALUES (1001);


INSERT INTO Mobile_Accessories (product_ID)

VALUES (1002);


INSERT INTO Mobile_Accessories (product_ID)

VALUES (1003);


INSERT INTO Mobile_Accessories (product_ID)

VALUES (1004);
```

8. **Insert data into Home_Appliances table**

```
INSERT INTO Home_Appliances (product_ID, height, width)

VALUES (1001, 15.5, 10.2);


INSERT INTO Home_Appliances

VALUES (1002, 5.0, 3.5);


INSERT INTO Home_Appliances
```

```
VALUES (1003, 12.0, 8.5);


INSERT INTO Home_Appliances

VALUES (1004, 18.2, 14.0);


INSERT INTO Home_Appliances

VALUES (1005, 8.5, 6.0);


INSERT INTO Home_Appliances

VALUES (1006, 20.0, 12.5);


INSERT INTO Home_Appliances

VALUES (1007, 10.0, 7.2);


INSERT INTO Home_Appliances

VALUES (1008, 15.0, 9.8);


INSERT INTO Home_Appliances

VALUES (1009, 6.5, 4.0);


INSERT INTO Home_Appliances

VALUES (1010, 22.0, 16.5);
```

9. Insert data into Clothes table

```
INSERT INTO Clothes (product_ID, cloth_Size)
```

```
VALUES (1001, 'Medium');


INSERT INTO Clothes

VALUES (1002, 'Large');


INSERT INTO Clothes

VALUES (1003, 'Small');


INSERT INTO Clothes

VALUES (1004, 'XL');


INSERT INTO Clothes

VALUES (1005, 'Medium');


INSERT INTO Clothes

VALUES (1006, 'Small');


INSERT INTO Clothes

VALUES (1007, 'Large');
```

10. Insert data into Toys table

```
INSERT INTO Toys (product_ID, hazard_Status)

VALUES (1001, 'Hazardous');
```

```
INSERT INTO Toys

VALUES (1002, 'Hazardous');


INSERT INTO Toys

VALUES (1003, 'Hazardous');


INSERT INTO Toys

VALUES (1004, 'Hazardous');


INSERT INTO Toys

VALUES (1005, 'Hazardous');
```

11. Insert data into Seller table

```
INSERT INTO Seller (seller_ID, product_ID, seller_Name)

VALUES (501, 1001, 'Electronics Store');


INSERT INTO Seller

VALUES (502, 1002, 'Toys and Gadgets');


INSERT INTO Seller

VALUES (503, 1003, 'Home Appliances Superstore');


INSERT INTO Seller

VALUES (504, 1004, 'Tech World');
```

```sql
INSERT INTO Seller

VALUES (505, 1005, 'Audio Paradise');


INSERT INTO Seller

VALUES (506, 1006, 'Camera Corner');


INSERT INTO Seller

VALUES (507, 1007, 'Print & More');


INSERT INTO Seller

VALUES (508, 1008, 'Gadget Galaxy');


INSERT INTO Seller

VALUES (509, 1009, 'Storage Solutions');


INSERT INTO Seller

VALUES (510, 1010, 'Tech Accessories Outlet');
```

12. Insert data into Shopping_Cart table

```sql
INSERT INTO Shopping_Cart (SCart_ID, product_ID, total_Quantity, total_Cost)

VALUES (10001, 1001, 2, 1500.00);


INSERT INTO Shopping_Cart

VALUES (10002, 1002, 3, 750.50);
```

```
INSERT INTO Shopping_Cart

VALUES (10003, 1003, 1, 300.00);


INSERT INTO Shopping_Cart

VALUES (10004, 1004, 2, 1200.50);


INSERT INTO Shopping_Cart

VALUES (10005, 1005, 5, 500.25);


INSERT INTO Shopping_Cart

VALUES (10006, 1006, 1, 800.00);


INSERT INTO Shopping_Cart

VALUES (10007, 1007, 3, 450.75);


INSERT INTO Shopping_Cart

VALUES (10008, 1008, 2, 600.50);


INSERT INTO Shopping_Cart

VALUES (10009, 1009, 4, 900.00);


INSERT INTO Shopping_Cart

VALUES (10010, 1010, 1, 150.75);
```

13. Insert data into Shopping_Order table

```
INSERT INTO Shopping_Order (order_ID, cust_ID, SCart_ID, order_Date)

VALUES (20001, 1, 10001, 20240116);


INSERT INTO Shopping_Order

VALUES (20002, 2, 10002, 20240117);


INSERT INTO Shopping_Order

VALUES (20003, 3, 10003, 20240118);


INSERT INTO Shopping_Order

VALUES (20004, 4, 10004, 20240119);


INSERT INTO Shopping_Order

VALUES (20005, 5, 10005, 20240120);


INSERT INTO Shopping_Order

VALUES (20006, 1, 10006, 20240121);


INSERT INTO Shopping_Order

VALUES (20007, 2, 10007, 20240122);


INSERT INTO Shopping_Order

VALUES (20008, 3, 10008, 20240123);
```

```
INSERT INTO Shopping_Order

VALUES (20009, 4, 10009, 20240124);


INSERT INTO Shopping_Order

VALUES (20010, 5, 10010, 20240125);
```

14. Insert data into Payment table

```
INSERT INTO Payment (pay_ID, order_ID, amount_Pay)

VALUES (30001, 20001, 1500.00);


INSERT INTO Payment

VALUES (30002, 20002, 750.50);


INSERT INTO Payment

VALUES (30003, 20003, 300.00);


INSERT INTO Payment

VALUES (30004, 20004, 1200.50);


INSERT INTO Payment

VALUES (30005, 20005, 500.25);


INSERT INTO Payment

VALUES (30006, 20006, 800.00);
```

```
INSERT INTO Payment

VALUES (30007, 20007, 450.75);


INSERT INTO Payment

VALUES (30008, 20008, 600.50);


INSERT INTO Payment

VALUES (30009, 20009, 900.00);


INSERT INTO Payment

VALUES (30010, 20010, 150.75);
```

15. Insert data into Cash table

```
INSERT INTO Cash (pay_ID)

VALUES (30001);


INSERT INTO Cash

VALUES (30003);


INSERT INTO Cash

VALUES (30005);
```

16. Insert data into Online_Transfer table

```
INSERT INTO Online_Transfer (pay_ID, reference_Number)
```

```
VALUES (30002, 123456789012345);


INSERT INTO Online_Transfer

VALUES (30004, 987654321012345);


INSERT INTO Online_Transfer

VALUES (30006, 543210987654321);
```

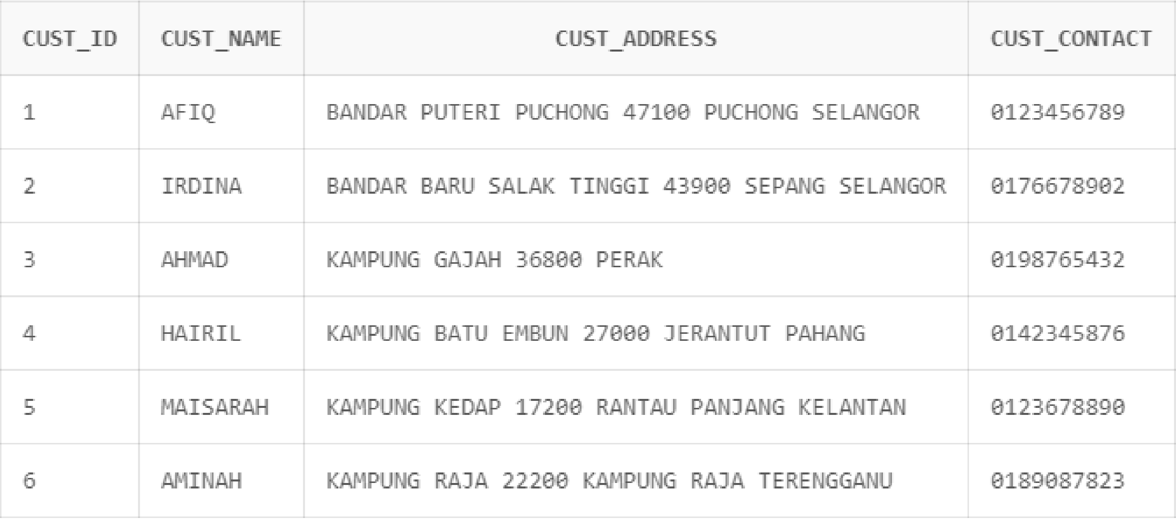
17. Insert data into Credit_Debit_Card table

```
INSERT INTO Credit_Debit_Card (pay_ID, card_Number)

VALUES (30007, 987654321098);


INSERT INTO Credit_Debit_Card

VALUES (30008, 123456789012);


INSERT INTO Credit_Debit_Card

VALUES (30009, 456789012345);


INSERT INTO Credit_Debit_Card

VALUES (30010, 876543210987);
```

6.3 TEST QUERY

1. View Customer Information

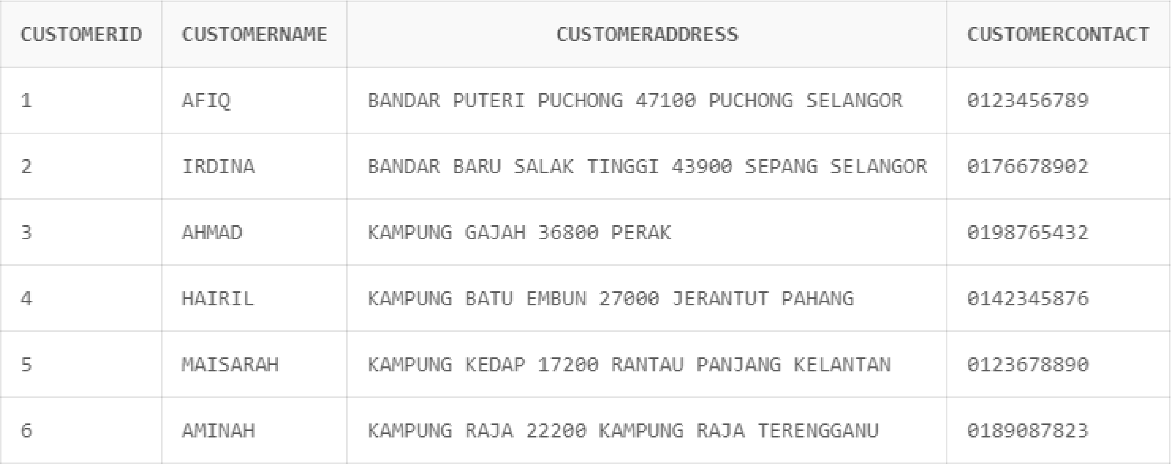
```
SELECT *
FROM Customer;
```



CUST_ID	CUST_NAME	CUST_ADDRESS	CUST_CONTACT
1	AFIQ	BANDAR PUTERI PUCHONG 47100 PUCHONG SELANGOR	0123456789
2	IRDINA	BANDAR BARU SALAK TINGGI 43900 SEPANG SELANGOR	0176678902
3	AHMAD	KAMPUNG GAJAH 36800 PERAK	0198765432
4	HAIRIL	KAMPUNG BATU EMBUN 27000 JERANTUT PAHANG	0142345876
5	MAISARAH	KAMPUNG KEDAP 17200 RANTAU PANJANG KELANTAN	0123678890
6	AMINAH	KAMPUNG RAJA 22200 KAMPUNG RAJA TERENGGANU	0189087823

2. Customize Customer Table Header

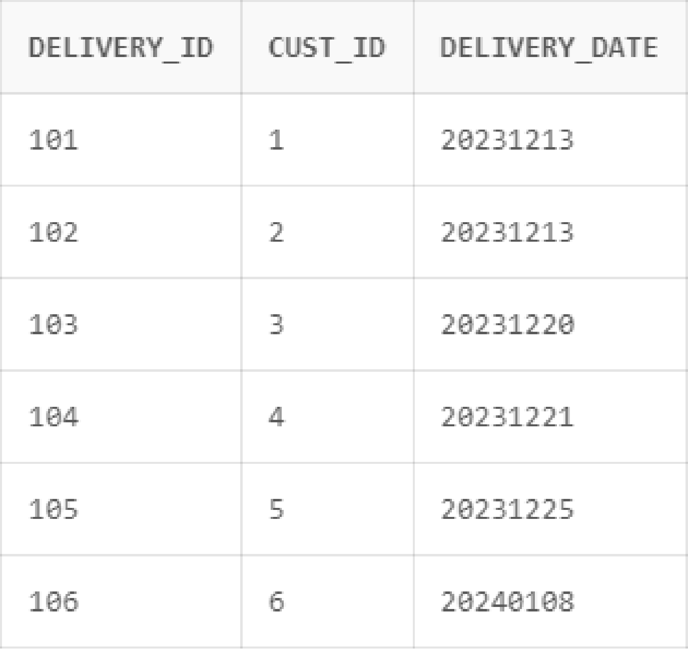
```
SELECT
  cust_ID AS CustomerID,
  cust_Name AS CustomerName,
  cust_Address AS CustomerAddress,
  cust_Contact AS CustomerContact
FROM Customer;
```



CUSTOMERID	CUSTOMERNAME	CUSTOMERADDRESS	CUSTOMERCONTACT
1	AFIQ	BANDAR PUTERI PUCHONG 47100 PUCHONG SELANGOR	0123456789
2	IRDINA	BANDAR BARU SALAK TINGGI 43900 SEPANG SELANGOR	0176678902
3	AHMAD	KAMPUNG GAJAH 36800 PERAK	0198765432
4	HAIRIL	KAMPUNG BATU EMBUN 27000 JERANTUT PAHANG	0142345876
5	MAISARAH	KAMPUNG KEDAP 17200 RANTAU PANJANG KELANTAN	0123678890
6	AMINAH	KAMPUNG RAJA 22200 KAMPUNG RAJA TERENGGANU	0189087823

3. View Delivery Information

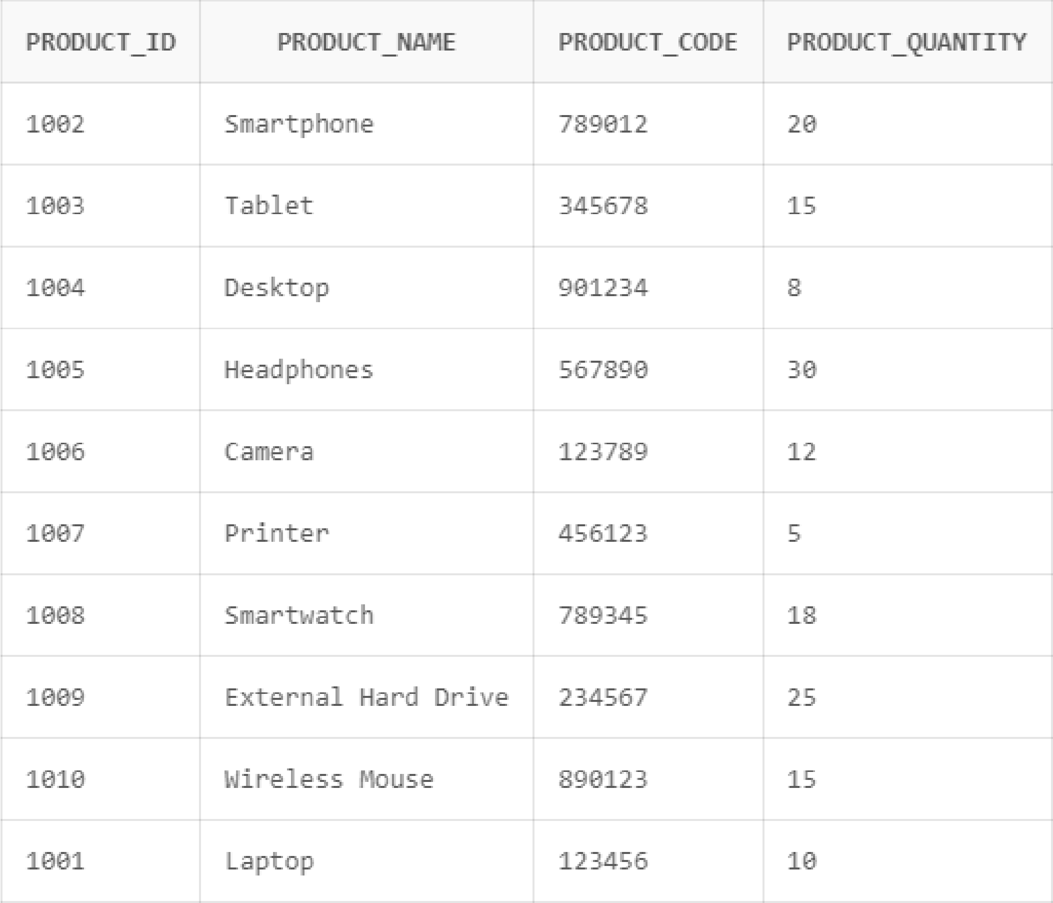
```
SELECT *
FROM Delivery;
```



DELIVERY_ID	CUST_ID	DELIVERY_DATE
101	1	20231213
102	2	20231213
103	3	20231220
104	4	20231221
105	5	20231225
106	6	20240108

4. View Product Information

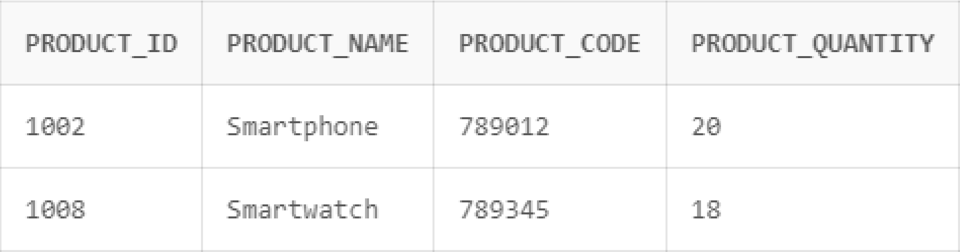
```
SELECT *
FROM Product;
```



PRODUCT_ID	PRODUCT_NAME	PRODUCT_CODE	PRODUCT_QUANTITY
1002	Smartphone	789012	20
1003	Tablet	345678	15
1004	Desktop	901234	8
1005	Headphones	567890	30
1006	Camera	123789	12
1007	Printer	456123	5
1008	Smartwatch	789345	18
1009	External Hard Drive	234567	25
1010	Wireless Mouse	890123	15
1001	Laptop	123456	10

5. View Product That Starts With Letter ‘S’

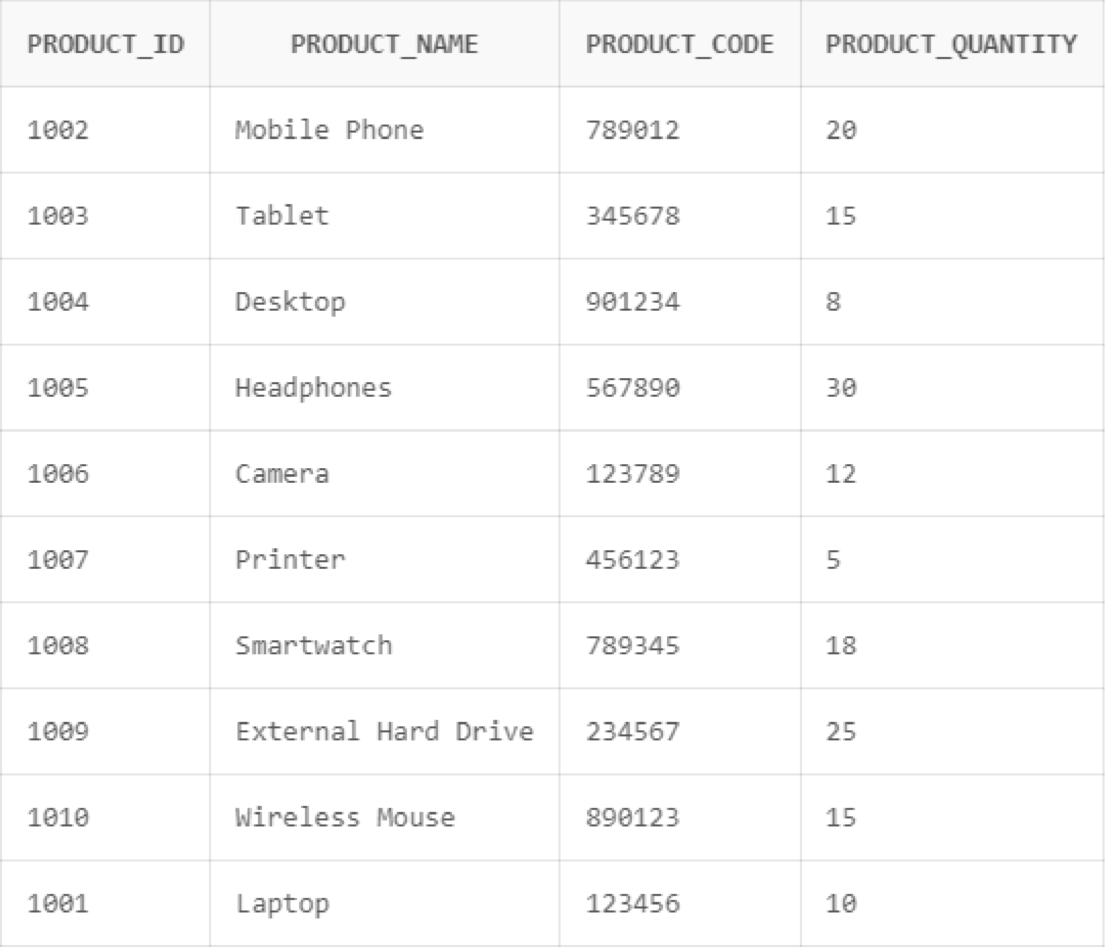
```
SELECT *
FROM Product
WHERE product_Name LIKE 'S%';
```



PRODUCT_ID	PRODUCT_NAME	PRODUCT_CODE	PRODUCT_QUANTITY
1002	Smartphone	789012	20
1008	Smartwatch	789345	18

6. Update Product Name

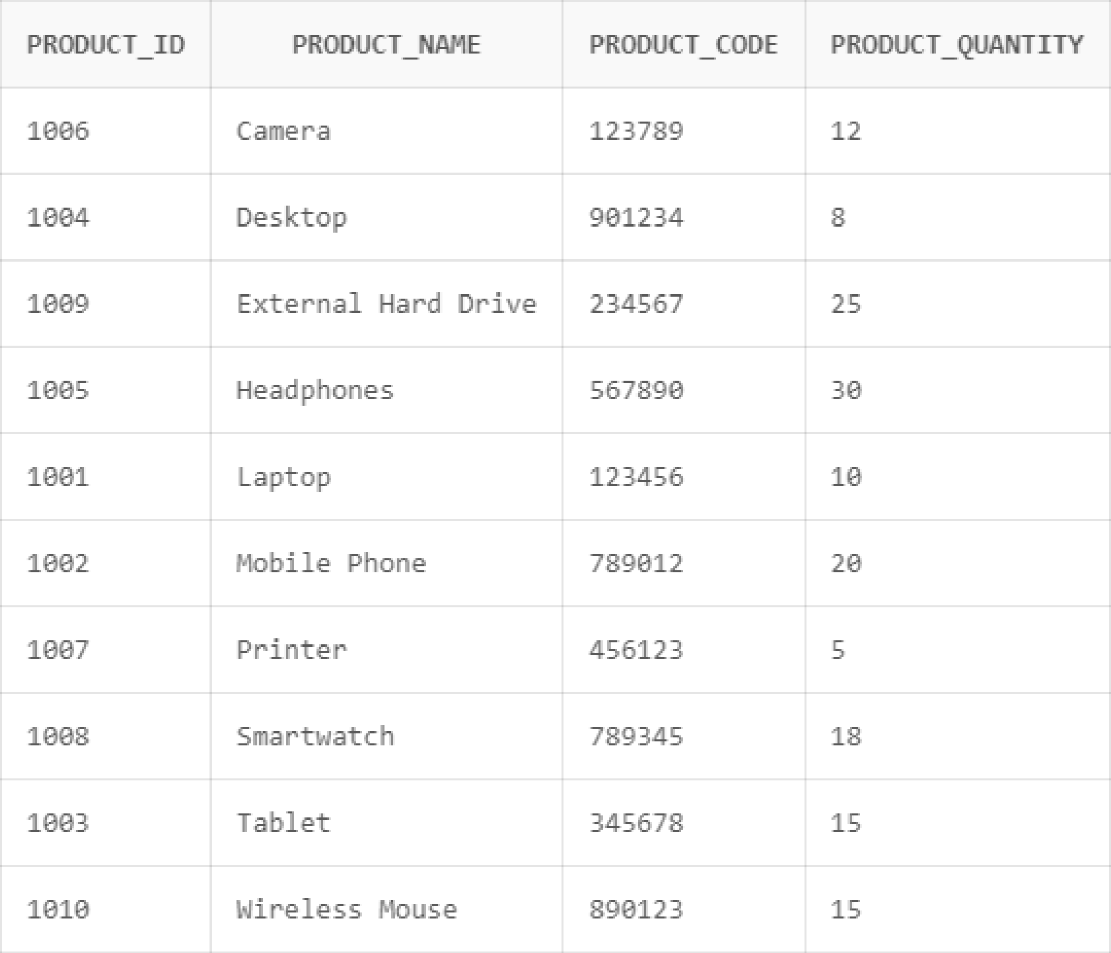
```
UPDATE Product
SET product_Name = 'Mobile Phone'
WHERE product_ID = '1002';
```



PRODUCT_ID	PRODUCT_NAME	PRODUCT_CODE	PRODUCT_QUANTITY
1002	Mobile Phone	789012	20
1003	Tablet	345678	15
1004	Desktop	901234	8
1005	Headphones	567890	30
1006	Camera	123789	12
1007	Printer	456123	5
1008	Smartwatch	789345	18
1009	External Hard Drive	234567	25
1010	Wireless Mouse	890123	15
1001	Laptop	123456	10

7. Sort Product Name in Ascending Order

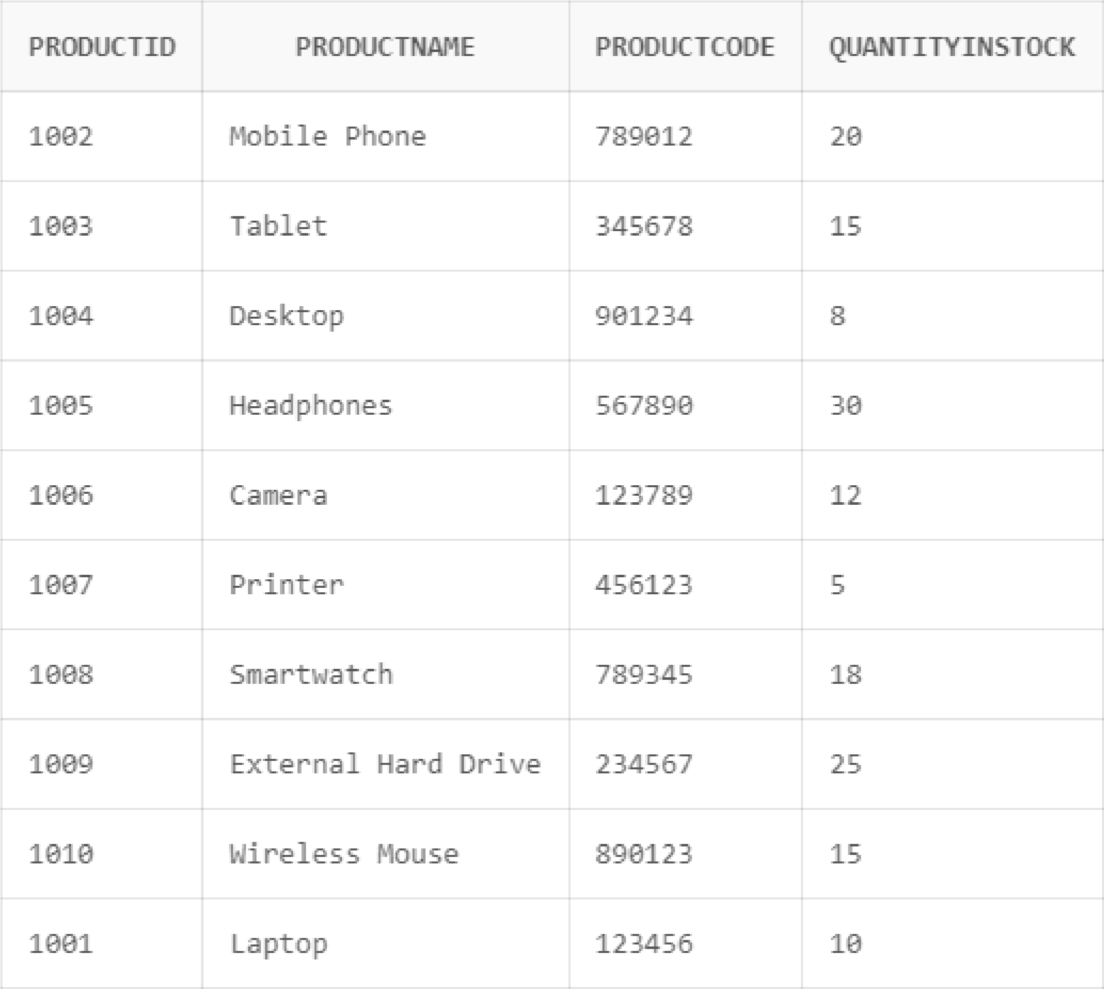
```
SELECT *
FROM Product
ORDER BY product_Name ASC;
```



PRODUCT_ID	PRODUCT_NAME	PRODUCT_CODE	PRODUCT_QUANTITY
1006	Camera	123789	12
1004	Desktop	901234	8
1009	External Hard Drive	234567	25
1005	Headphones	567890	30
1001	Laptop	123456	10
1002	Mobile Phone	789012	20
1007	Printer	456123	5
1008	Smartwatch	789345	18
1003	Tablet	345678	15
1010	Wireless Mouse	890123	15

8. Customize Product Table Header

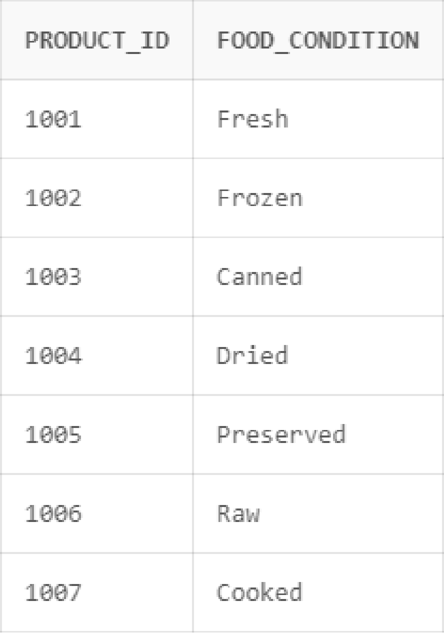
```
SELECT
  product_ID AS ProductID,
  product_Name AS ProductName,
  product_Code AS ProductCode,
  product_Quantity AS QuantityInStock
FROM Product;
```



PRODUCTID	PRODUCTNAME	PRODUCTCODE	QUANTITYINSTOCK
1002	Mobile Phone	789012	20
1003	Tablet	345678	15
1004	Desktop	901234	8
1005	Headphones	567890	30
1006	Camera	123789	12
1007	Printer	456123	5
1008	Smartwatch	789345	18
1009	External Hard Drive	234567	25
1010	Wireless Mouse	890123	15
1001	Laptop	123456	10

9. View Food Information

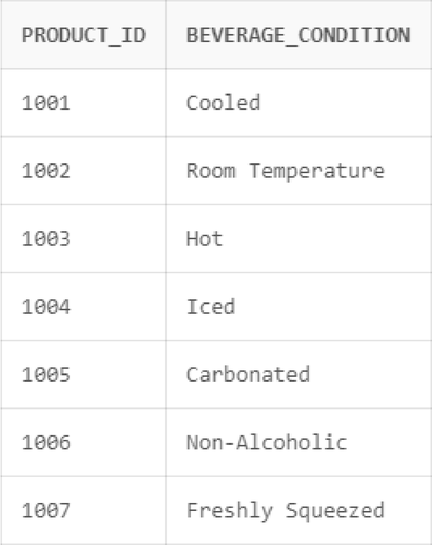
```
SELECT *
FROM Food;
```



PRODUCT_ID	FOOD_CONDITION
1001	Fresh
1002	Frozen
1003	Canned
1004	Dried
1005	Preserved
1006	Raw
1007	Cooked

10. View Beverages Information

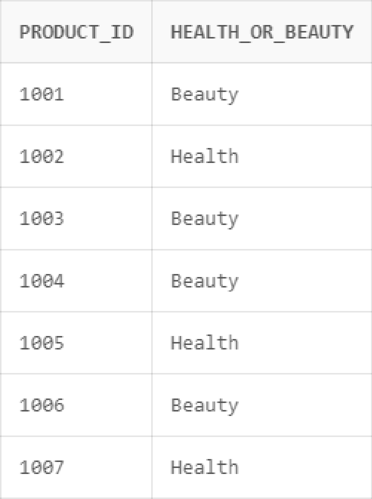
```
SELECT *
FROM Beverages;
```



PRODUCT_ID	BEVERAGE_CONDITION
1001	Cooled
1002	Room Temperature
1003	Hot
1004	Iced
1005	Carbonated
1006	Non-Alcoholic
1007	Freshly Squeezed

11. View Health_Beauty Information

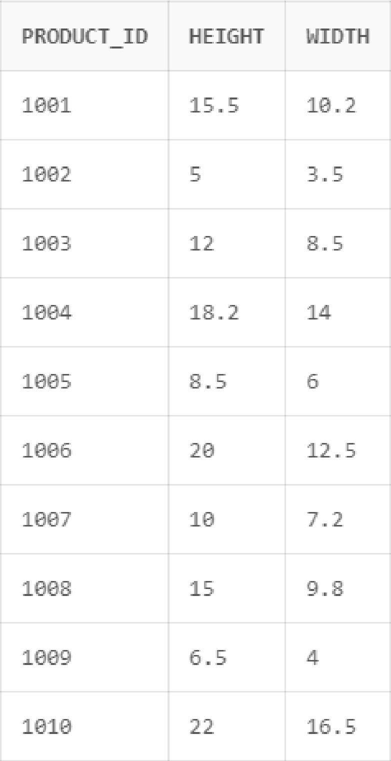
```
SELECT *
FROM Health_Beauty;
```



PRODUCT_ID	HEALTH_OR_BEAUTY
1001	Beauty
1002	Health
1003	Beauty
1004	Beauty
1005	Health
1006	Beauty
1007	Health

12. View Home_Appliances Information

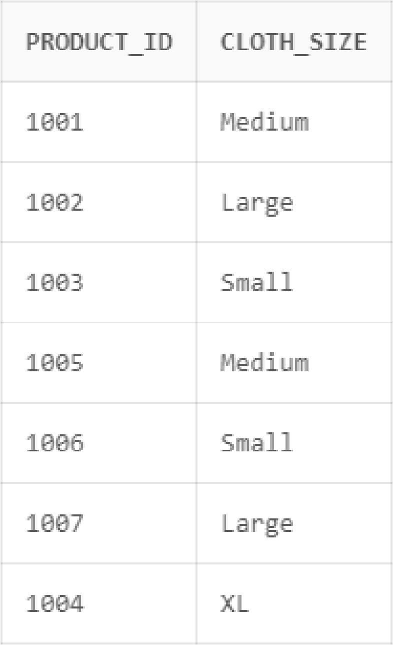
```
SELECT *
FROM Home_Appliances;
```



PRODUCT_ID	HEIGHT	WIDTH
1001	15.5	10.2
1002	5	3.5
1003	12	8.5
1004	18.2	14
1005	8.5	6
1006	20	12.5
1007	10	7.2
1008	15	9.8
1009	6.5	4
1010	22	16.5

13. View Clothes Information

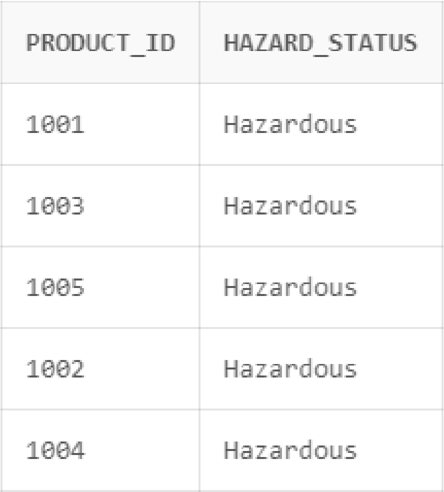
```
SELECT *
FROM Clothes;
```



PRODUCT_ID	CLOTH_SIZE
1001	Medium
1002	Large
1003	Small
1005	Medium
1006	Small
1007	Large
1004	XL

14. View Toys Information

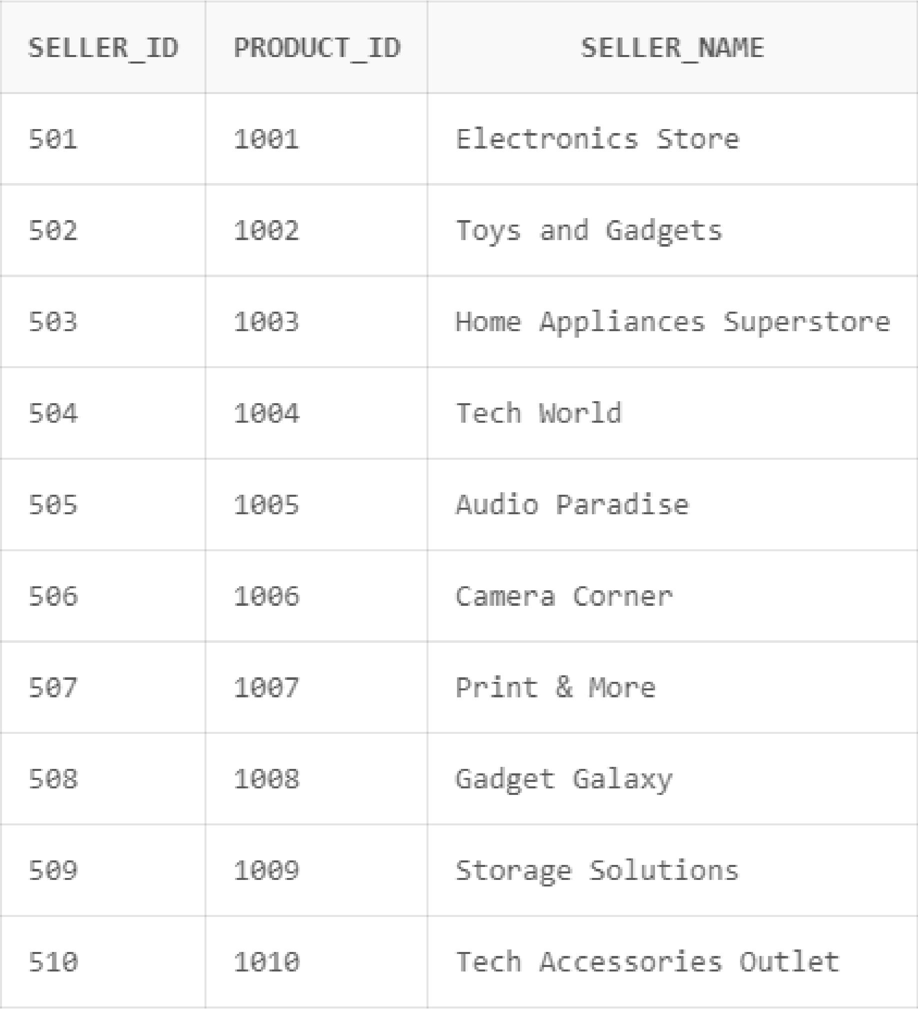
```
SELECT *
FROM Toys;
```



PRODUCT_ID	HAZARD_STATUS
1001	Hazardous
1003	Hazardous
1005	Hazardous
1002	Hazardous
1004	Hazardous

15. View Seller Information

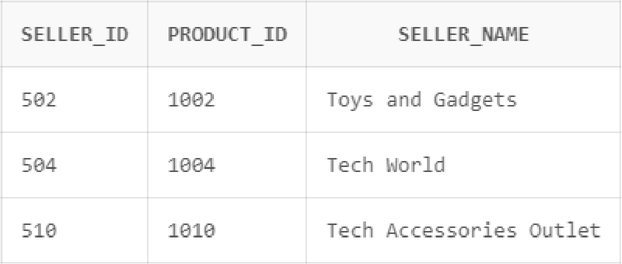
```
SELECT *

FROM Seller;
```



SELLER_ID	PRODUCT_ID	SELLER_NAME
501	1001	Electronics Store
502	1002	Toys and Gadgets
503	1003	Home Appliances Superstore
504	1004	Tech World
505	1005	Audio Paradise
506	1006	Camera Corner
507	1007	Print & More
508	1008	Gadget Galaxy
509	1009	Storage Solutions
510	1010	Tech Accessories Outlet

16. View Seller That Starts With Letter ‘T’

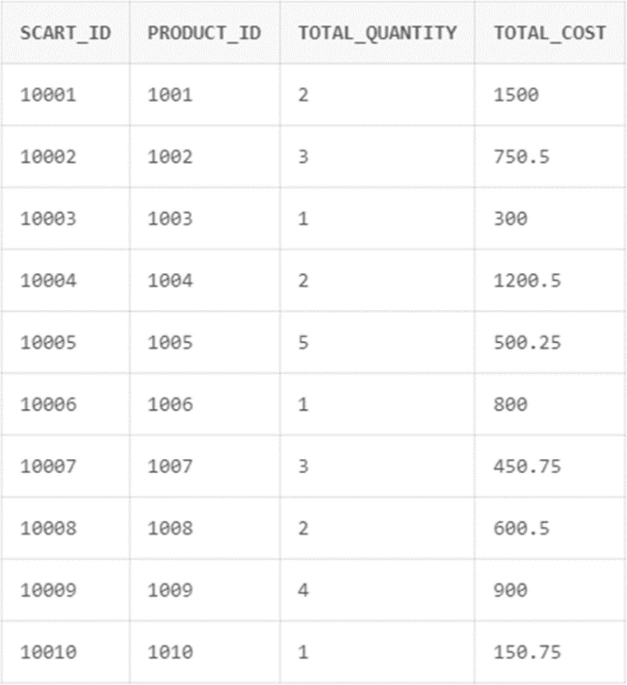
```
SELECT *

FROM Seller

WHERE seller_Name LIKE 'T%';
```



SELLER_ID	PRODUCT_ID	SELLER_NAME
502	1002	Toys and Gadgets
504	1004	Tech World
510	1010	Tech Accessories Outlet

17. View Shopping_Cart Information

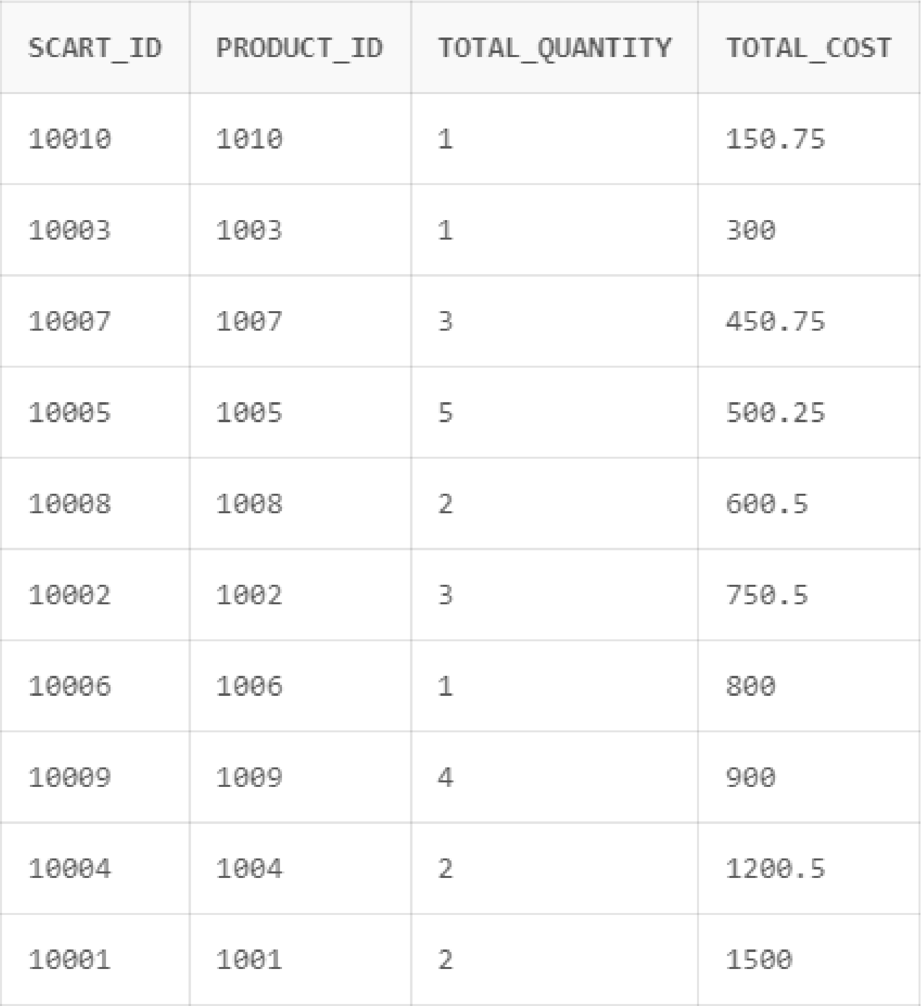
```
SELECT *

FROM Shopping_Cart;
```



SCART_ID	PRODUCT_ID	TOTAL_QUANTITY	TOTAL_COST
10001	1001	2	1500
10002	1002	3	750.5
10003	1003	1	300
10004	1004	2	1200.5
10005	1005	5	500.25
10006	1006	1	800
10007	1007	3	450.75
10008	1008	2	600.5
10009	1009	4	900
10010	1010	1	150.75

18. Sort Shopping Cart in Ascending Order of Total Cost

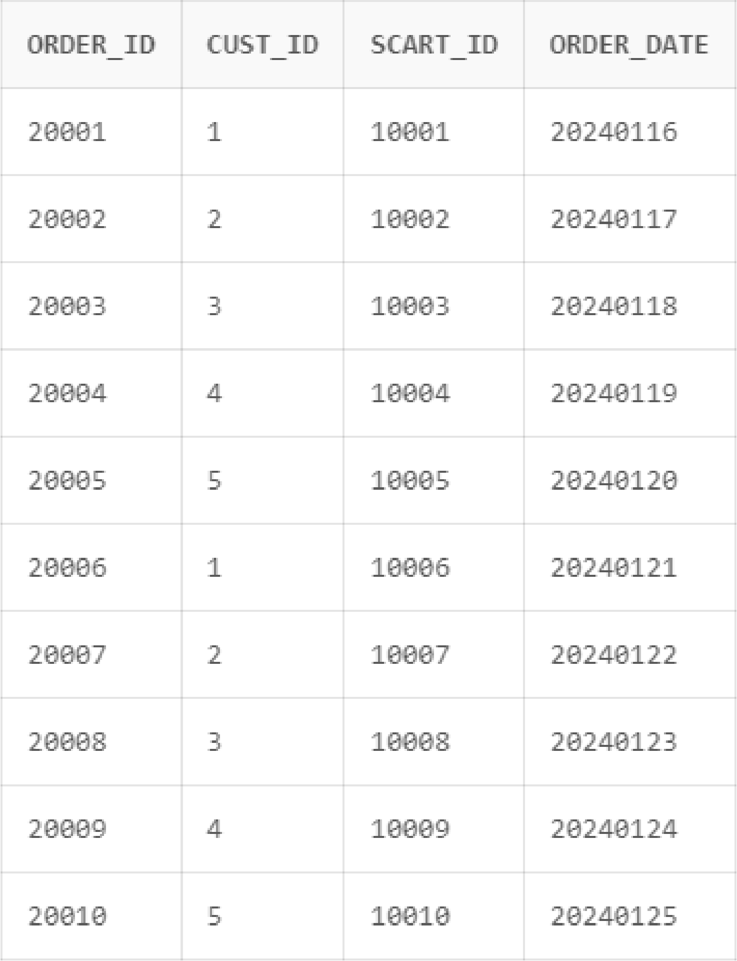
```
SELECT *

FROM Shopping_Cart

ORDER BY total_cost ASC;
```



SCART_ID	PRODUCT_ID	TOTAL_QUANTITY	TOTAL_COST
10010	1010	1	150.75
10003	1003	1	300
10007	1007	3	450.75
10005	1005	5	500.25
10008	1008	2	600.5
10002	1002	3	750.5
10006	1006	1	800
10009	1009	4	900
10004	1004	2	1200.5
10001	1001	2	1500

19. View Shopping Order Information

```
SELECT *

FROM Shopping_Order;
```

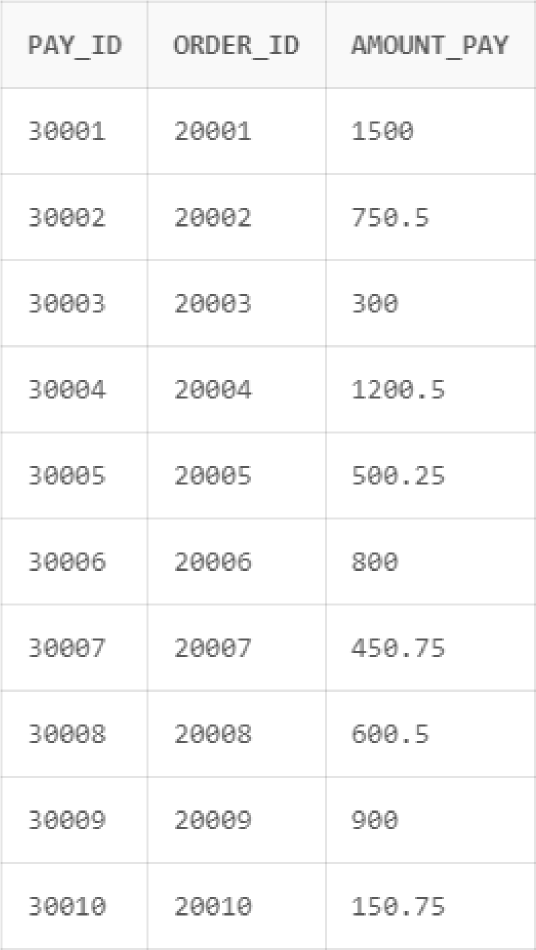


ORDER_ID	CUST_ID	SCART_ID	ORDER_DATE
20001	1	10001	20240116
20002	2	10002	20240117
20003	3	10003	20240118
20004	4	10004	20240119
20005	5	10005	20240120
20006	1	10006	20240121
20007	2	10007	20240122
20008	3	10008	20240123
20009	4	10009	20240124
20010	5	10010	20240125

20. View Payment Information

SELECT *

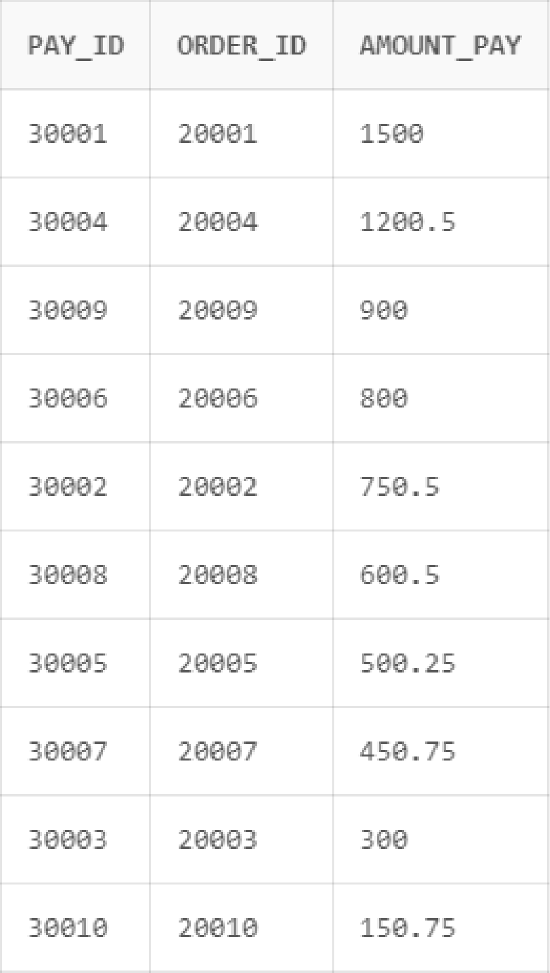
FROM Payment;



PAY_ID	ORDER_ID	AMOUNT_PAY
30001	20001	1500
30002	20002	750.5
30003	20003	300
30004	20004	1200.5
30005	20005	500.25
30006	20006	800
30007	20007	450.75
30008	20008	600.5
30009	20009	900
30010	20010	150.75

21. Sort Payment in Descending Order of Total Cost

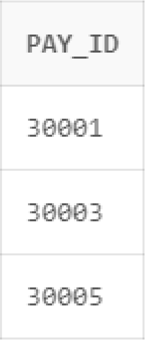
```
SELECT *

FROM Payment

ORDER BY amount_Pay DESC;
```



PAY_ID	ORDER_ID	AMOUNT_PAY
30001	20001	1500
30004	20004	1200.5
30009	20009	900
30006	20006	800
30002	20002	750.5
30008	20008	600.5
30005	20005	500.25
30007	20007	450.75
30003	20003	300
30010	20010	150.75

22. View Cash Payment

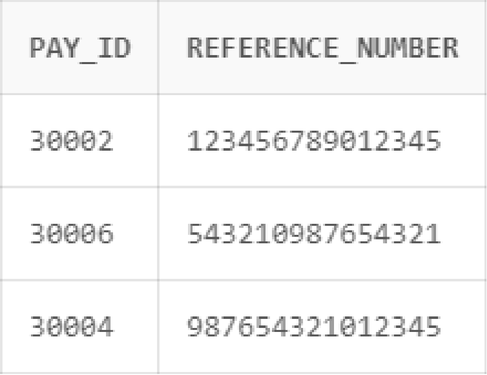
```
SELECT *

FROM Cash;
```



PAY_ID
30001
30003
30005

23. View Online Transfer Payment

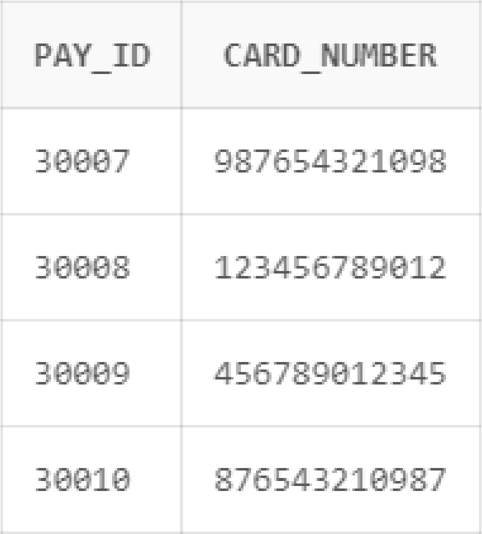
```
SELECT *

FROM Online_Transfer;
```



PAY_ID	REFERENCE_NUMBER
30002	123456789012345
30006	543210987654321
30004	987654321012345

24. View Credit or Debit Card Payment

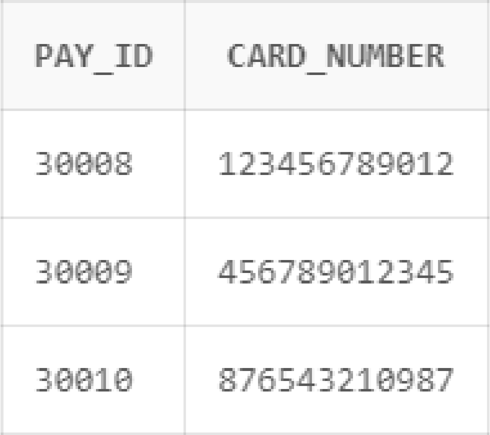
```
SELECT *

FROM Credit_Debit_Card;
```



PAY_ID	CARD_NUMBER
30007	987654321098
30008	123456789012
30009	456789012345
30010	876543210987

25. Delete Credit or Debit Card Payment.

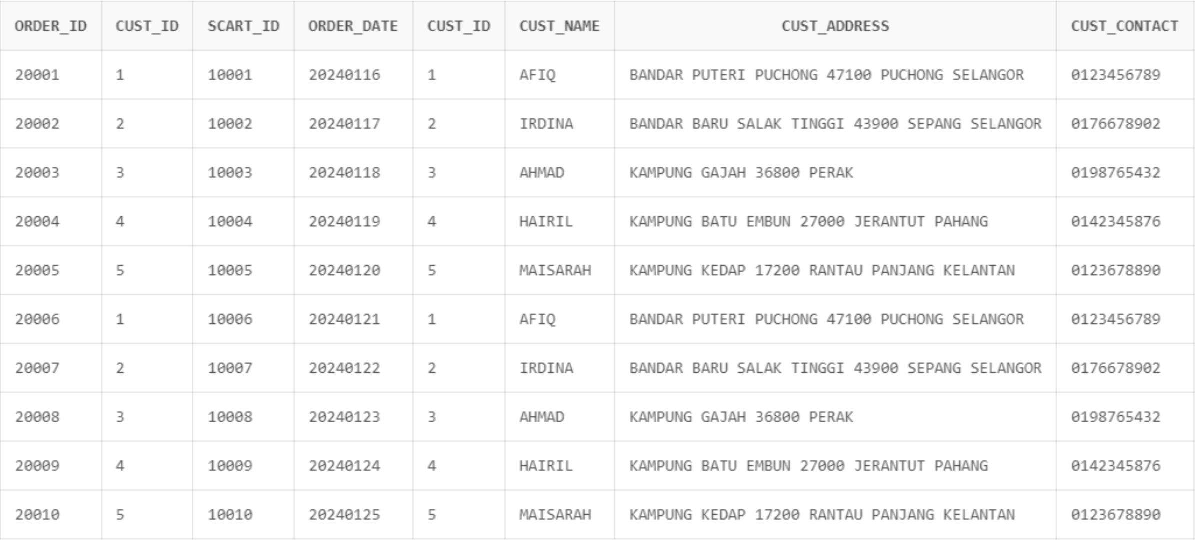
```
DELETE FROM Credit_Debit_Card

WHERE pay_ID = 30007;
```



PAY_ID	CARD_NUMBER
30008	123456789012
30009	456789012345
30010	876543210987

26. View Join Shopping Order With Customer Details

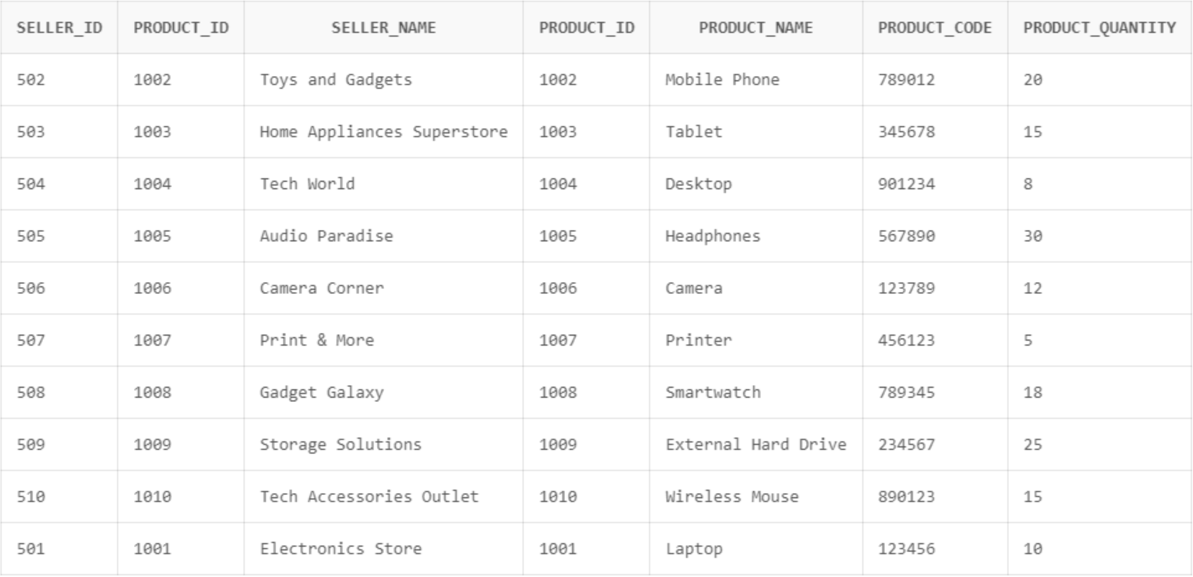
```
SELECT *

FROM Shopping_Order so

JOIN Customer c ON so.cust_ID = c.cust_ID;
```



ORDER_ID	CUST_ID	SCART_ID	ORDER_DATE	CUST_ID	CUST_NAME	CUST_ADDRESS	CUST_CONTACT
20001	1	10001	20240116	1	AFIQ	BANDAR PUTERI PUCHONG 47100 PUCHONG SELANGOR	0123456789
20002	2	10002	20240117	2	IRDINA	BANDAR BARU SALAK TINGGI 43900 SEPANG SELANGOR	0176678902
20003	3	10003	20240118	3	AHMAD	KAMPUNG GAJAH 36800 PERAK	0198765432
20004	4	10004	20240119	4	HAIRIL	KAMPUNG BATU EMBUN 27000 JERANTUT PAHANG	0142345876
20005	5	10005	20240120	5	MAISARAH	KAMPUNG KEDAP 17200 RANTAU PANJANG KELANTAN	0123678890
20006	1	10006	20240121	1	AFIQ	BANDAR PUTERI PUCHONG 47100 PUCHONG SELANGOR	0123456789
20007	2	10007	20240122	2	IRDINA	BANDAR BARU SALAK TINGGI 43900 SEPANG SELANGOR	0176678902
20008	3	10008	20240123	3	AHMAD	KAMPUNG GAJAH 36800 PERAK	0198765432
20009	4	10009	20240124	4	HAIRIL	KAMPUNG BATU EMBUN 27000 JERANTUT PAHANG	0142345876
20010	5	10010	20240125	5	MAISARAH	KAMPUNG KEDAP 17200 RANTAU PANJANG KELANTAN	0123678890

27. View Join Seller with Product Information

```
SELECT *

FROM Seller s

JOIN Product p ON s.product_ID = p.product_ID;
```



SELLER_ID	PRODUCT_ID	SELLER_NAME	PRODUCT_ID	PRODUCT_NAME	PRODUCT_CODE	PRODUCT_QUANTITY
502	1002	Toys and Gadgets	1002	Mobile Phone	789012	20
503	1003	Home Appliances Superstore	1003	Tablet	345678	15
504	1004	Tech World	1004	Desktop	901234	8
505	1005	Audio Paradise	1005	Headphones	567890	30
506	1006	Camera Corner	1006	Camera	123789	12
507	1007	Print & More	1007	Printer	456123	5
508	1008	Gadget Galaxy	1008	Smartwatch	789345	18
509	1009	Storage Solutions	1009	External Hard Drive	234567	25
510	1010	Tech Accessories Outlet	1010	Wireless Mouse	890123	15
501	1001	Electronics Store	1001	Laptop	123456	10

7.0 SUMMARY

In the TS Mart System database, our group has completed Phase 3 of logical design and SQL statements. Considering grocery shopping can be done online, this database was created to help customers buy groceries faster and more efficiently while also saving them time and energy. Proper constraints, data types, and attributes are incorporated into the design of every entity. These systems are known for capturing financial transactions, order and delivery records, product details, and customer information. In order to guarantee data integrity, we additionally implement foreign key connections among tables. In the retail setting of TS Mart, this system offers a strong basis for handling transactions and customer interactions.

DEMONSTRATIONS

The Thousand Sunny group made a presentation video regarding Project Phase 3 as well as an SQL statement demonstration for the completed TS Mart System. Below is a link to the demonstration presentation video;

https://youtu.be/SofC-QLl6LM