1. INTRODUCTION

In this case study assignment, we will explore the design and management of a relational database system for a fictional company called "TechMart." TechMart is an online electronics retail store that sells a wide range of electronic products, including smartphones, laptops, cameras, and accessories. The goal of this assignment is to develop a comprehensive database system that can efficiently store and manage information about TechMart's products, customers, orders, and suppliers.

2. ASSIGNMENT OBJECTS

1. Develop Entity-Relationship (ER) diagram/s to model the TechMart database.

Entities :

Based on the information provided , the entities are

- Products
- Customers
- Orders
- Suppliers

Attributes :

For each entities , here are some attributes

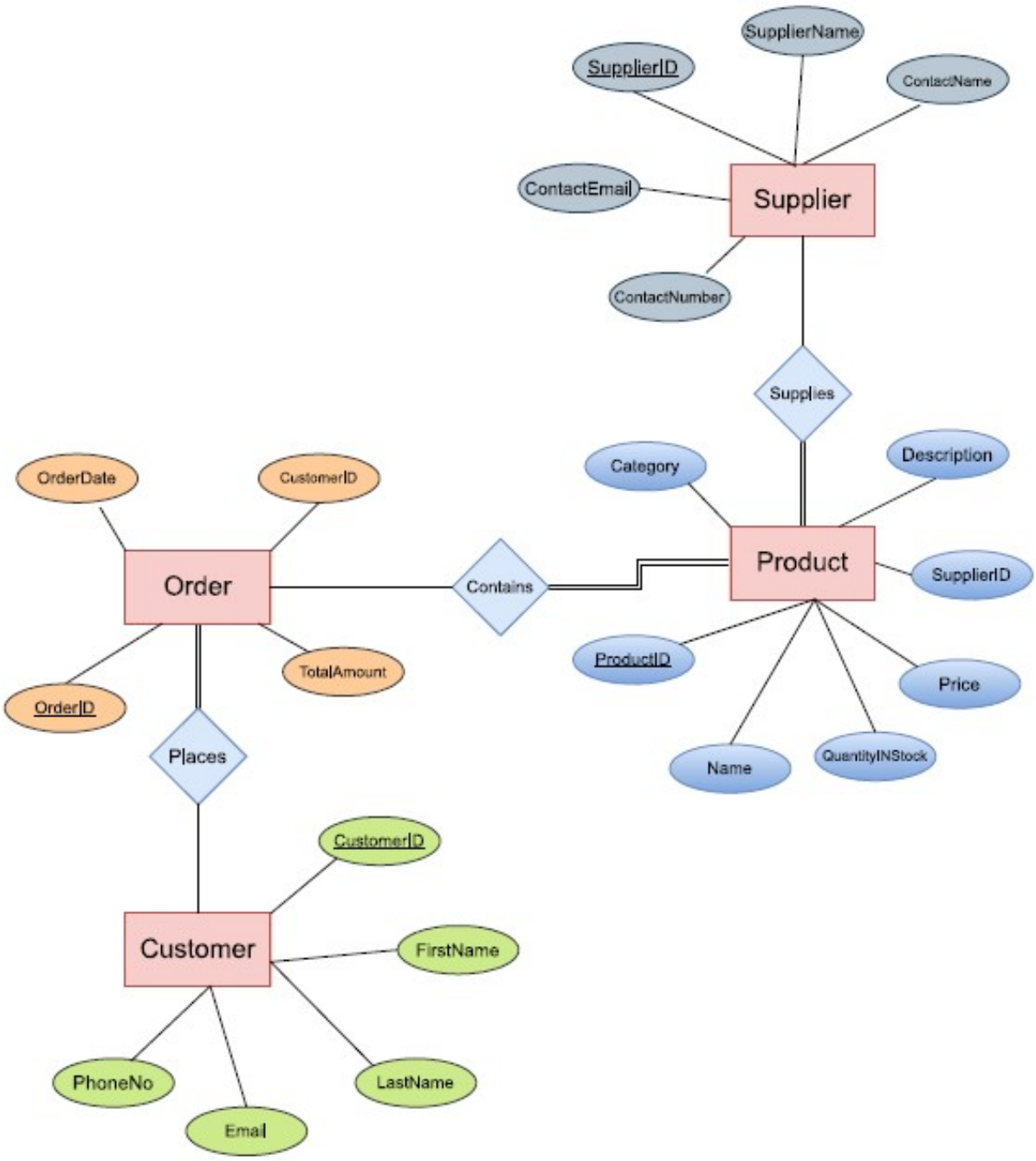
- Products : ProductID , Name , Description , Category ,SupplierID , Price , Stock INQunatity.
- Suppliers : SupplierID , SupplierName , ContactName , ContactEmail ,Contact Phone .
- Customers : CustomerID , FirstName ,LastName ,Email , PhoneNo.
  Orders : OrderID , OrderDate , TotalAmount.

Relationships :

The relationships between these entities are given below :

- A customer places many orders (One-to-many relationships).
- An Order contains one or more Products (One-to-many relationships).
- Products are supplied by Suppliers(many-to-one relationship).

ER Diagrams :



2. Create relational tables based on the ER diagram.

(I)Create the Product table

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Description VARCHAR(255),
    Price DECIMAL(10, 2) NOT NULL,
    Category VARCHAR(50),
    StockINQuantity INT,
    SupplierID INT,
    FOREIGN KEY (SupplierID) REFERENCES Suppliers(SupplierID)
);
```

(II) Create the Customers table

```
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) ,
    Phone VARCHAR(15)
);
```

(III) Create the Orders table

```
CREATE TABLE Order(
    OrderID INT PRIMARY KEY,
    OrderDate DATE,
    CustomerID INT,
    TotalAmount DECIMAL(10, 2) NOT NULL,
     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

(IV) Create the Suppliers table

```
CREATE TABLE Supplier (
   SupplierID INT PRIMARY KEY,
   SupplierName VARCHAR(100) NOT NULL,
   ContactName VARCHAR(100),
   ContactEmail VARCHAR(100),
   ContactPhone VARCHAR(15)
);
```

(V) Create the OrderProduct table

```
CREATE TABLE OrderProduct (
   OrderID INT,
   ProductID INT,
   Quantity INT,
   Price DECIMAL(10, 2),
   PRIMARY KEY(OrderID,ProductID),
   FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
   FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

3.Populate the tables with sample data.

Product Table :

ProductID	Name	Description	Category	StockIN Quentity	Price	SupplierID
401	Smart Phone A	4GB,64GB	Electronics	50	749	101
402	Laptop Pro B	16GB,512SSD	Electronics	20	1000	102
403	Camera C	16GB	Electronics	10	799	103
404	Tablet D	4GB,64GB	Electronics	50	399	101

Supplier Table :

SupplierID	SupplierName	ContactName	contactEmail	ContactPhone
101	TechSupplies	Sayak Das	dassayak@techsupplies.com	(555)555-1111
102	Gadgets R Us	Samir Das	samirdas@gadgetsrus.com	(555)555-2222
103	Electronics Hub	Rajiv Mishra	mishrarrajiv@ehub.com	(555)555-3333

Customer Table :

CustomerID	FirstName	LastName	Email	PhoneNo
301	Sayak	Das	sayakdas@email.com	555-123-4567
302	Samir	Jana	samirjana@email.com	555-125-9788
303	Rajiv	Mishra	mishrarrajiv@email.com	555-186-6798

Order Table :

OrderID	OrderDate	TotalAmount	CustomerID
801	2023-10-01	499	301
802	2023-10-02	569	303
803	2023-10-03	799	302

OrderProduct Table :

OrderID	ProductID	Quantity	Price
801	402	20	499
802	401	30	599
802	403	49	899
803	402	39	1000

3. Apply normalization techniques to ensure data integrity.

To apply normalization techniques to the tables, we'll aim for Third Normal Form (3NF) which involves the following steps:

First Normal Form (1NF):

- Ensure all attributes are atomic (indivisible).
- All values must be of a specific data type.
- There should be a primary key.

All the provided tables already meet the requirements for 1NF.

Second Normal Form (2NF):

- All attributes must be functionally dependent on the entire primary key.
- If a table has a composite primary key, each non-key attribute should be functionally dependent on the entire composite key.

Looking at the tables, they are already in 2NF.

Third Normal Form (3NF):

- All attributes must be functionally dependent only on the primary key. There should be no transitive dependencies.

Now, let's examine the tables*:*

Product Table :

ProductID	Name	Description	Category	StockIN Quantity	Price	SupplierID
401	Smart Phone A	4GB,64GB	Electronics	50	749	101
402	Laptop Pro B	16GB,512SSD	Electronics	20	1000	102
403	Camera C	16GB	Electronics	10	799	103
404	Tablet D	4GB,64GB	Electronics	50	399	101

All the attributes are directly related to the ProductID, which is the primary key. Therefore, the Product table is in 3NF.

Supplier Table :

SupplierID	SupplierName	ContactName	ContactEmail	ContactPhone
101	TechSupplies	Sayak Sen	dassayak@techsupplies.com	(555)555-1111
102	Gadgets R Us	Samir Das	samirdas@gadgetsrus.com	(555)555-2222
103	Electronics Hub	Rajiv Mishra	mishrarrajiv@ehub.com	(555)555-3333

Similar to the Product table, all attributes are directly related to the SupplierID, which is the primary key. Therefore, the Supplier table is in 3NF.

Customer Table :

CustomerID	FirstName	LastName	Email	PhoneNo
301	Sayak	Sen	sayaksen@email.com	555-123-4567
302	Samir	Jana	samirjana@email.com	555-125-9788
303	Rajiv	Mishra	mishrarrajiv@email.com	555-186-6798

Similar to the Product table, all attributes are directly related to the CustomerID, which is the primary key. Therefore, the Customer table is in 3NF.

Order Table :

OrderID	OrderDate	TotalAmount	CustomerID
801	2023-09-01	499	301
802	2023-09-02	569	303
803	2023-09-03	799	302

Similar to the Product table, all attributes are directly related to the OrderID, which is the primary key. Therefore, the Order table is in 3NF.

OrderProduct Table :

OrderID	ProductID	Quantity	Price
801	402	20	499
802	401	30	599
802	403	49	899
803	402	39	1000

The OrderProduct table is also in 3NF because all attributes are directly related to the combination of OrderID and ProductID, which together form the primary key for this table.
All the tables are already in Third Normal Form (3NF), ensuring data integrity and minimizing redundancy.

Q5. Demonstrate an understanding of basic database concepts.

Here's the demonstration of an understanding of basic database concepts based on the objectives mentioned:
Entity-Relationship (ER) Modeling:

- An ER diagram is a visual representation of the relationships among different entities in a database. It helps in understanding the structure and interactions within a database system.
- In the TechMart case study, we created an ER diagram to model the relationships between entities like Product, Supplier, Customer, and Order. This diagram illustrated how these entities are related and what attributes they have.

Normalization:

- Normalization is a process of organizing data in a database to minimize redundancy and dependency by organizing data into separate tables.

- In the TechMart case study, we ensured that the tables were in Third Normal Form (3NF). This means that each table is free from transitive dependencies, and all attributes are functionally dependent only on the primary key.

Relational Database Design:

- A relational database organizes data into tables, where each table represents an entity, and the relationships between entities are established through keys.
- In the TechMart case study, we designed tables for Product, Supplier, Customer, Order, and Order_Product, each with its set of attributes. We established relationships using primary and foreign keys to link the entities.

Data Integrity:

- Data integrity ensures that data is accurate, consistent, and reliable. This is achieved through mechanisms like primary keys, foreign keys, constraints, and normalization.
- In the TechMart database, we used primary keys to uniquely identify records in each table. We also used foreign keys to establish relationships between tables, ensuring referential integrity.

SQL Statements:

- SQL (Structured Query Language) is used to interact with databases. It allows users to perform tasks like querying, updating, and managing data.
- In the TechMart case study, we provided SQL statements to create the tables and populate them with sample data. These statements followed the syntax and structure of SQL.

6. SQL queries to retrieve and manipulate data.

I. Retrieve a list of all products in the database.

SQL Query:

```
SELECT * FROM Product;
```

Output :

ProductID	Name	Description	Category	StockIN Quantity	Price	SupplierID
401	Smart Phone A	4GB,64GB	Electronics	50	749	101
402	Laptop Pro B	16GB,512SSD	Electronics	20	1000	102
403	Camera C	16GB	Electronics	10	799	103
404	Tablet D	4GB,64GB	Electronics	50	399	101

II. Find the total number of orders placed by each customer.

SQL Query:

```
SELECT CustomerID, COUNT(*) AS TotalOrders
FROM Order
GROUP BY CustomerID;
```

Output :

CustomerID	TotalOrder
301	1
302	1
303	1

III. List all suppliers who provide products in a specific category.

SQL query :

```
SELECT DISTINCT Supplier.SupplierID, SupplierName,
ContactName, ContactEmail, ContactPhone
FROM Supplier
JOIN Product ON Supplier.SupplierID = Product.SupplierID
WHERE Product.Category = 'Electronics';
```

Output :

SupplierID	SupplierName	ContactName	ContactEmail	ContactPhone
101	TechSupplies	John Dae	Jhon.dae@techsupplies.com	(555)555-1111

102	Gadgets R Us	Jane Smith	janesmith@gadgetsrus.com	(555)555-2222
103	Electronics Hub	Mark Johnson	markjohnson@ehub.com	(555)555-3333

IV.Calculate the total revenue generated by TechMart in a given month.

Let us consider the September 2023 sales reports

```
SELECT SUM(Order.TotalAmount) AS TotalRevenue
FROM Order
WHERE OrderDate >= '2023-09-01' AND OrderDate <= '2023-09-31';
```

Output:

TotalRevenue
1867

V.Update the price of a specific product.

Let's say we want to update the price of 'Laptop pro B' to $899.

SQL query:

```
UPDATE Product
SET Price = 899
WHERE Name = 'Laptop Pro B';
```

ProductID	Name	Description	Category	StockIN Quantity	Price	SupplierID
401	Smart Phone A	4GB,64GB	Electronics	50	749	101
402	Laptop Pro B	16GB,512SSD	Electronics	20	899	102
403	Camera C	16GB	Electronics	10	799	103
404	Tablet D	4GB,64GB	Electronics	50	399	101

VI. Delete a customer's order and update the order total accordingly.

SQL query:

```
DELETE FROM Order
WHERE OrderID = 801;
```

OrderID	ProductID	Quantity	Price
802	401	30	599
802	403	49	899
803	402	39	1000

```
SELECT sum(Price) as Update_Sum
FROM Order
```

Update_Sum
2498