

Mokapuff Bakery - Bread Bags & Accessories

Team members

- Thi Diem Quynh Bui (Katherine), 301416482
- Wenxi Zhong (Simone), 301476512
- Khanh Huyen (Emily) Le, 301393966
- Minh Nguyen, 301411105

Project Title: Mokapuff Bakery - Bread Bags & Accessories

Project URL:

<https://www.kickstarter.com/projects/mokapuff/mokapuff-bakery-bread-bags-and-accessories?ref=section-food-craft-view-more-discovery-p1>

Final Milestone

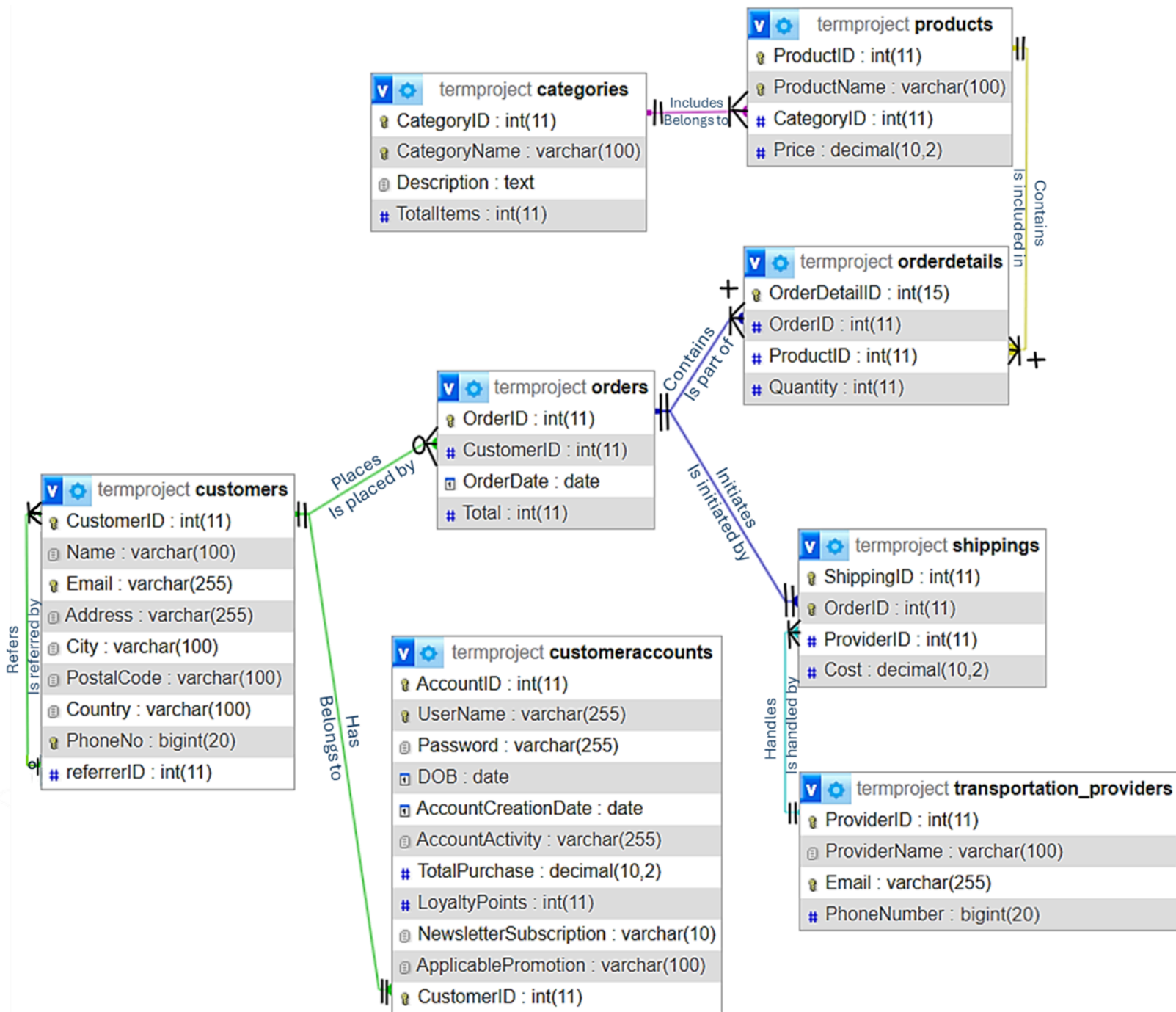
Total mark: 10 (out of 20 of Term project mark)

The goal is to model a complete database of at least 6 entities for the Kickstarter project you have selected

A) ERD of your design (2 marks)

Marks will be deducted for :

- Not identifying modalities and cardinalities in ERDs
- Not denoting weak entities
- Not displaying relationships descriptors in your ERDs
- Using tinny fonts



B) Questions (5 marks)

1. Indicate a 1:1 relationship and the two entities involved in it. Also, provide the SQL query that established that 1:1 relationship (the SQL statement that created the two tables and the corresponding constraints)

- "Customers" Table and "CustomerAccounts" Table have 1:1 relationship

Customers

```
CREATE TABLE Customers (  
  CustomerID int(11) NOT NULL AUTO_INCREMENT,  
  Name varchar(100) NOT NULL,  
  Email varchar(255) UNIQUE,  
  Address varchar(255) NOT NULL,  
  City varchar(100) NOT NULL,  
  PostalCode varchar(100) NOT NULL,  
  Country varchar(100) NOT NULL,  
  PhoneNo BIGINT UNIQUE DEFAULT NULL,  
  PRIMARY KEY(CustomerID)  
) ENGINE=InnoDB;
```

Customer Accounts

```
CREATE TABLE `CustomerAccounts` (  
  AccountID int(11) AUTO_INCREMENT,  
  UserName varchar(255) UNIQUE NOT NULL,  
  Password varchar(255) NOT NULL,  
  DOB Date NOT NULL,  
  AccountCreationDate Date,  
  AccountActivity varchar(255),  
  TotalPurchase Decimal(10,2),  
  LoyaltyPoints INT(11),  
  NewsletterSubscription varchar(10),  
  ApplicablePromotion varchar(100),  
  CustomerID int(11) UNIQUE,  
  PRIMARY KEY (AccountID),  
  CONSTRAINT account_belongs_1customer FOREIGN KEY(CustomerID)  
REFERENCES Customers(CustomerID)  
)ENGINE=INNODB
```

2. Indicate an M:M relationship and the two entities and the weak entity involved in it and the identifier of the weak entity (the composite key). Also, provide the SQL query that established that M;M relationship (the SQL statement that created the tables and the corresponding constraints)

- “Products” table and “Orders” table have M:M relationship
- “OrderDetails” table is the *weak entity*.
- The identifier for the weak entity "OrderDetails" is a composite key made up of **OrderID** and **ProductID**. This composite key uniquely identifies each record in the "OrderDetails" table and is crucial in maintaining the relationship between orders and products.

Products

```
CREATE TABLE Products (  
    ProductID int(11) NOT NULL AUTO_INCREMENT,  
    ProductName VARCHAR(100) UNIQUE NOT NULL,  
    CategoryID int(11) NOT NULL,  
    Price Decimal (10, 2),  
    PRIMARY KEY(ProductID),  
    CONSTRAINT fk_Categories FOREIGN KEY (CategoryID)  
    REFERENCES Categories(CategoryID)  
    ) ENGINE=innnoDB;
```

Orders

```
CREATE TABLE Orders(  
    OrderID int(11) NOT NULL AUTO_INCREMENT,  
    CustomerID int(11) NOT NULL,  
    OrderDate date NOT NULL ,  
    Total int (11) NOT NULL,  
    PRIMARY KEY (OrderID),  
    Constraint fk_Orders_Customers FOREIGN KEY (CustomerID) REFERENCES  
    Customers(CustomerID)  
    )ENGINE=INNODB;
```

Order Details

```
CREATE TABLE OrderDetails (  
    OrderDetailID INT(15) NOT NULL AUTO_INCREMENT,
```

```

OrderID INT(11) NOT NULL,
ProductID INT(11) NOT NULL,
Quantity INT(11) NOT NULL CHECK(Quantity>0),
Primary Key (OrderDetailID),
Constraint fk_orders FOREIGN KEY(OrderID) References Orders(orderID),
Constraint fk_products FOREIGN KEY(ProductID) References
Products(ProductID)
)ENGINE=innnoDB;

```

3. Provide at least one recursive relationship (the table with the recursion, the SQL statement that defines the table in text + the screenshot of ERD)

- “**Customers**” table
- Recursive relationship is through the **referrerID** which refers back to **CustomerID** within the same “**Customers**” table, which tracks referrals (such as a customer who was referred by another customer).

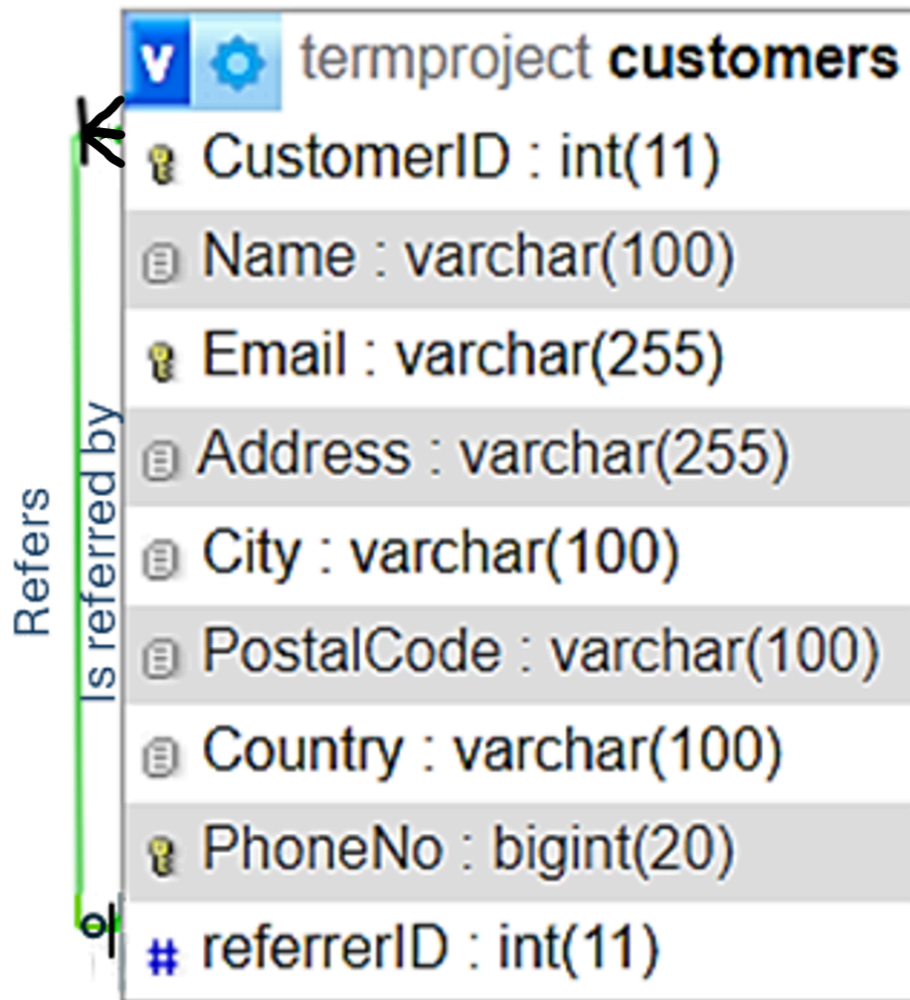
Customers

```

CREATE TABLE Customers (
CustomerID int(11) NOT NULL AUTO_INCREMENT,
Name varchar(100) NOT NULL,
Email varchar(255) UNIQUE,
Address varchar(255) NOT NULL,
City varchar(100) NOT NULL,
PostalCode varchar(100) NOT NULL,
Country varchar(100) NOT NULL,
PhoneNo BIGINT UNIQUE DEFAULT NULL,
PRIMARY KEY(CustomerID)
) ENGINE=innnoDB;

ALTER TABLE Customers
ADD COLUMN referrerID INT,
ADD CONSTRAINT fk_customers_referral
FOREIGN KEY (referrerID) REFERENCES Customers(CustomerID);

```



4. Explain how you made sure 2NF was not violated in your DB. None of your tables can violate 2NF (explain by example, e.g. we broke down this table into this and this ...)

- "Customers" Table

- a. Primary Key: **CustomerID**
- b. Attributes: **Name, Email, Address, City, PostalCode, Country, PhoneNo, referrerID**

→ 2NF is ensured: Each attribute directly depends on 'CustomerID'. There are no composite keys, and each non-key attribute is fully functionally dependent on the primary key 'CustomerID'.

- "CustomerAccounts" Table

- Primary Key: **AccountID**

- Foreign Key: **CustomerID** references **Customer(CustomerID)**
- Attributes: **UserName, Password, DOB, AccountCreationDate, AccountActivity, TotalPurchase, LoyaltyPoints, NewsletterSubscription, ApplicablePromotion**

→ 2NF is ensured: We designated '**AccountID**' as a unique primary key to ensure all account-related attributes depend on it, not just '**CustomerID**'. This change eliminated any potential partial dependency that would have occurred if CustomerID was part of a composite key.

- **"Shippings" Table**

- Primary Key: **ShippingID**
- Foreign Key: **OrderID** references **Orders(OrderID)**; **ProviderID** references **transportation_providers(ProviderID)**
- Attributes: **Cost**

→ 2NF is ensured: By defining '**OrderID**' and '**ProviderID**' as foreign keys that are not part of a composite primary key but rather linked uniquely to each shipment, we ensured full dependency of shipping cost on '**ShippingID**' alone.

- **"Products" Table**

- Primary Key: **ProductID**
- Foreign Key: **CategoryID** references **Categories(CategoryID)**
- Attributes: **ProductName, Price**

→ 2NF is ensured: There's a direct full dependency of all attributes on **ProductID**. In this table, there is no composite key and **CategoryID** is used just to link products to their categories.

- **"Categories" Table**

- Primary Key: **CategoryID**
- Attributes: **CategoryName, Description, TotalItems**

→ 2NF is ensured: All attributes are fully dependent on **CategoryID**, which is the only primary key, hence satisfying 2NF.

- **"Transportation_providers" Table**

- Primary Key: **ProviderID**
- Attributes: **ProviderName, Email, PhoneNumber**

→ 2NF is ensured: All attributes are fully dependent on **ProviderID**, which is the only primary key, hence satisfying 2NF.

- **"Orders" Table**

- Primary Key: **OrderID**
- Foreign Key: **CustomerID** references **Customers(CustomerID)**
- Attributes: **OrderDate, Total**

→ 2NF is ensured: The table ensures that all order details depend solely on **OrderID**. Since **CustomerID** serves to link each order to a customer, there's no partial dependency.

- **"OrderDetails" Table**

- Primary Key: **OrderDetailID**
- Foreign Keys: **OrderID** references **Orders(OrderID)**; **ProductID** references **Products(ProductID)**
- Attributes: **Quantity**

→ 2NF is ensured: If **OrderID** and **ProductID** were considered a composite key in the '**OrderDetails**' table, it could lead to partial dependencies. Therefore, we created **OrderDetailID** as a unique primary key for '**OrderDetails**' table; and maintained **OrderID** as a primary key in '**Orders**' table and **ProductID** as a primary key in '**Products**' table. This structure ensures that all attributes in each table (**OrderDetails, Orders, Products**) are fully dependent on their respective primary keys, not on a combination of keys.

5. In the table that stores user credentials, provide a constraint that makes sure the length of password is at least 6 characters long. Provide the check constraints in your SQL script that enforces such constraint

```
ALTER TABLE CustomerAccounts  
ADD CHECK (`Password` LIKE '_____%')
```

6. Your data model must account for all the exceptions. Provide at least one exception and your solution to it

There is a potential for customers to exploit the referral program by providing their own information as the referrer, allowing them to obtain discounts.

```
ALTER TABLE Customers  
Add CHECK (CustomerID <> referrerID)
```

Note: Since MyDBMaria doesn't allow the use of AUTO_INCREMENT together with CHECK constraints, we need to remove the AUTO_INCREMENT attribute before altering the table.

It's not our desire to remove AUTO_INCREMENT, but this is the only way to demonstrate the solution in MyDBMaria. Our ideal design would be to have both AUTO_INCREMENT and CHECK constraints in our database to optimize the business process.

```
ALTER TABLE Customers MODIFY CustomerID INT(11);
```

7. Provide an example of running a query on a M:M relationship of your model (the query in text, followed by the result)

What are the products' names ordered with only 1 quantity?

```
Select DISTINCT products.ProductID, productname
From products, orderdetails, orders
Where quantity=1
And orderdetails.orderid= orders.orderid
And orderdetails.productid= products.productid;
```

ProductID	productname
1	Babie Bear Baguette Crossbody Bag
5	Bunny Cupcake Backpack

8. Provide an example of running a query on a 1:1 relationship of your model (the query in text, followed by the result)

Retrieve all the shipping information with order information

```
SELECT Orders.OrderID, Orders.CustomerID, Orders.OrderDate,
Orders.Total, shippings.ShippingID, shippings.ProviderID, shippings.Cost
FROM Orders
INNER JOIN shippings ON orders.OrderID = shippings.OrderID;
```

OrderID	CustomerID	OrderDate	Total	ShippingID	ProviderID	Cost
1	1	2024-01-01	75	1	1	10.00
2	2	2024-01-02	30	2	2	35.00
3	2	2024-01-03	150	3	2	45.00
4	3	2024-03-14	60	4	4	50.00
5	4	2024-03-15	40	5	5	30.00

9. Provide an example of running a query on a recursive relationship of your model (the query in text, followed by the result)

List all customers and their referrer's names (even if they don't have refers)

```
select a.name as customers, b.name as referrername from customers a
LEFT JOIN customers b on b.CustomerID= a.referrerID;
```

customers	referrername
John Doe	<i>NULL</i>
Jane Nguyen	John Doe
Alice Wang	Jane Nguyen
Crystal Huynh	Jane Nguyen
Peter Lee	Crystal Huynh

C) 12 queries (3 marks)

Write 12 queries on your DB to utilize all you have learned by the end of this course (SELECT, UNIQUE, PRIMARY KEY, JOIN, IN, subqueries, EXISTS, HAVING,...)

- insert at least three entries in each of your tables (one more row compared to the previous milestone). You may need more rows to make sure your design is complete and flawless

Insertion

1. Customers

```
INSERT INTO `Customers` (`CustomerID`, `Name`, `Email`, `Address`, `City`,
`postalcode`, `Country`, `PhoneNo`, `referrerID`) VALUES
```

```
(1, 'John Doe', 'john.doe@example.com', '123 Elm St', 'Springfield', '02310', 'USA',
1234567890, NULL),
```

```
(2, 'Jane Nguyen', 'jane.nguyen@example.com', '456 Pham Ngoc Thach St',
'HaNoi', 'V314X6', 'Vietnam', 2345678901, 1),
```

```
(3, 'Alice Wang', 'alice.wang@gmail.com', '36 Pine St', 'Guangzhou', '510832',
'China', 13724356598, 2),
```

```
(4, 'Crystal Huynh', 'crystal.huynh@gmail.com', '120 Hoan Kiem St', 'HaNoi',
'V5M6A8', 'Vietnam', 2346662020, 2),
```

```
(5, 'Peter Lee', 'peter.lee@gmail.com', '420 Hamilton St', 'Seattle', '08110', 'USA',
1238887220, 4);
```

2. Customer Accounts

```
INSERT INTO CustomerAccounts VALUES
```

```
(1,'john.doe', 'password123', '1985-05-15', '2024-01-01', 'Review product',
2500.00, 300, 'Yes', 'Mid-year Sale', 1),
(2,'jane.nguyen', 'securepass', '1990-07-20', '2024-02-15', 'Share product',
1500.00, 150, 'No', 'Birthday Discount', 2),
(3,'alice.wang', 'alicepass', '1988-03-22', '2024-03-10', 'Save product', 3500.00,
450, 'Yes', NULL, 3),
(4,'crystal.huynh', 'crystal123', '1995-03-20', '2024-03-14', 'Share Product',
2000.00, 200, 'No', 'Birthday Discount', 4),
(5, 'peter.lee', 'peter456', '2000-04-24', '2024-06-22', 'Share Product', 3000.00,
300, 'Yes', 'Summer Sale', 5);
```

3. Products

```
INSERT INTO Products (ProductName, CategoryID, Price) VALUES
('Babie Bear Baguette Crossbody Bag', 1, 29.99),
('Shoku-pan Bear Tote Bag', 2, 24.99),
('Banada Fruits Sando Plushie', 3, 19.99),
('Melon-pan Coin Pouch', 4, 9.99),
('Bunny Cupcake Backpack', 5, 49.99),
('Pudding Lollipop Keychain', 6, 8.99);
```

4. Categories

```
INSERT INTO Categories VALUES
(1,'Crossbody Bags', 'Stylish and functional crossbody bags for everyday use', 8),
(2,'Tote Bags', 'Spacious and trendy tote bags', 10),
(3,'Plushies', 'Cute and cuddly plush toys', 10),
(4,'Coin Pouches', 'Convenient and cute coin pouches', 10),
(5,'Backpacks', 'Durable and fashionable backpacks', 5),
(6,'Keychains', 'Cute and adorable keychains', 6);
```

5. Shippings

```
INSERT INTO `Shippings` (`ShippingID`, `OrderID`, `ProviderID`, `Cost`)
VALUES (1, 1, 1, 10), (2, 2, 2, 35), (3, 3, 2, 45), (4, 4, 4, 50), (5, 5, 5, 30);
```

6. Transportation Providers

```
INSERT INTO `transportation_providers`
```

```
(`ProviderID`, `ProviderName`, `Email`, `PhoneNumber`)
```

```
VALUES
```

```
(1, 'CanadaPost', 'cp@canadapost.com', 9876543210),
```

```
(2, 'International', 'international@quickship.com', 9876543211),
```

```
(3, 'Backup', 'backup@reliabletrans.com', 9876543212),
```

```
(4, 'DHL', 'dhl@globalshipping.com', 9876543213),
```

```
(5, 'FedEx', 'fedex@homedelivery.com', 9876543214);
```

7. Orders

```
INSERT INTO `Orders` (`OrderID`, `CustomerID`, `OrderDate`, `Total`)
```

```
VALUES
```

```
(1, 1, '2024-01-01', 75),
```

```
(2, 2, '2024-01-02', 30),
```

```
(3, 2, '2024-01-03', 150),
```

```
(4, 3, '2024-03-14', 60),
```

```
(5, 4, '2024-03-15', 40);
```

8. Order Details

```
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
```

```
(1, 1, 1),
```

```
(2, 3, 2),
```

```
(3, 5, 1),
```

```
(4, 2, 2),
```

```
(5, 1, 1);
```

12 Queries:

1. SELECT

```
SELECT * FROM Customers WHERE CustomerID = 2
```

CustomerID	Name	Email	Address	City	PostalCode	Country	PhoneNo	referrerID
2	Jane Nguyen	jane.nguyen@example.com	456 Pham Ngoc Thach St	HaNoi	V3I4X6	Vietnam	2345678901	1

2. ORDER BY

SELECT * FROM Customers ORDER BY Name ASC;

CustomerID	Name	Email	Address	City	PostalCode	Country	PhoneNo	referrerID
3	Alice Wang	alice.wang@gmail.com	36 Pine St	Guangzhou	510832	China	13724356598	2
4	Crystal Huynh	crystal.huynh@gmail.com	120 Hoan Kiem St	HaNoi	V5M6A8	Vietnam	2346662020	2
2	Jane Nguyen	jane.nguyen@example.com	456 Pham Ngoc Thach St	HaNoi	V3I4X6	Vietnam	2345678901	1
1	John Doe	john.doe@example.com	123 Elm St	Springfield	02310	USA	1234567890	NULL
5	Peter Lee	peter.lee@gmail.com	420 Hamilton St	Seattle	08110	USA	1238887220	4

3. LIKE

SELECT * FROM Customers WHERE name LIKE 'A%';

CustomerID	Name	Email	Address	City	PostalCode	Country	PhoneNo	referrerID
3	Alice Wang	alice.wang@gmail.com	36 Pine St	Guangzhou	510832	China	13724356598	2

4. BETWEEN

SELECT AccountID, CustomerID, UserName, TotalPurchase FROM
`CustomerAccounts` WHERE TotalPurchase BETWEEN 2500 AND 3000;

AccountID	CustomerID	UserName	TotalPurchase
1	1	john.doe	2500.00
5	5	peter.lee	3000.00

5. MIN

SELECT MIN(DOB) from CustomerAccounts;

MIN(DOB)

1985-05-15

6. MAX

SELECT MAX(Total) FROM Orders;

MAX(Total)

150

7. COUNT

SELECT COUNT(OrderID) FROM Orders WHERE CustomerID = 3

COUNT(OrderID)

1

8. IN

SELECT * FROM Customers WHERE CustomerID IN
(SELECT CustomerID FROM Orders WHERE Total =
(SELECT MAX(Total) FROM Orders));

CustomerID	Name	Email	Address	City	PostalCode	Country	PhoneNo	referrerID
2	Jane Nguyen	jane.nguyen@example.com	456 Pham Ngoc Thach St	HaNoi	V3l4X6	Vietnam	2345678901	1

9. Average

SELECT AVG(Total) FROM Orders;

AVG(Total)

71.0000

10. GROUP BY

SELECT COUNTRY, COUNT(CustomerID) AS Nofcustomers FROM
Customers GROUP BY Country;

COUNTRY	Nofcustomers
China	1
USA	2
Vietnam	2

11. HAVING

SELECT count(AccountID) as nofsubscription from CustomerAccounts
GROUP BY NewsletterSubscription

HAVING NewsletterSubscription='yes';

nofsubscription

3

12. LEFT JOIN

```
SELECT DISTINCT Products.ProductID, ProductName, Quantity FROM  
Products LEFT JOIN OrderDetails
```

```
ON Products.ProductID=OrderDetails.ProductID
```

```
ORDER BY QUANTITY DESC;
```

ProductID	ProductName	Quantity
3	Banada Fruits Sando Plushie	2
2	Shoku-pan Bear Tote Bag	2
1	Babie Bear Baguette Crossbody Bag	1
5	Bunny Cupcake Backpack	1
6	Pudding Lollipop Keychain	NULL
4	Melon-pan Coin Pouch	NULL

D) SQL dump (schema + data SQL statements)

```
-- phpMyAdmin SQL Dump  
-- version 5.2.1  
-- https://www.phpmyadmin.net/  
--  
-- Host: 127.0.0.1  
-- Generation Time: Jul 29, 2024 at 10:19 PM  
-- Server version: 10.4.32-MariaDB  
-- PHP Version: 8.0.30
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";  
START TRANSACTION;  
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;  
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;  
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```

```

/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `termpproject`
--

-----

--
-- Table structure for table `categories`
--

CREATE TABLE `categories` (
  `CategoryID` int(11) NOT NULL,
  `CategoryName` varchar(100) NOT NULL,
  `Description` text DEFAULT NULL,
  `TotalItems` int(11) DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `categories`
--

INSERT INTO `categories` (`CategoryID`, `CategoryName`, `Description`, `TotalItems`)
VALUES
(1, 'Crossbody Bags', 'Stylish and functional crossbody bags for everyday use', 8),
(2, 'Tote Bags', 'Spacious and trendy tote bags', 10),
(3, 'Plushies', 'Cute and cuddly plush toys', 10),
(4, 'Coin Pouches', 'Convenient and cute coin pouches', 10),
(5, 'Backpacks', 'Durable and fashionable backpacks', 5),
(6, 'Keychains', 'Cute and adorable keychains', 6);

-----

--
-- Table structure for table `customeraccounts`
--

```



```

CREATE TABLE `customeraccounts` (
  `AccountID` int(11) NOT NULL,
  `UserName` varchar(255) NOT NULL,
  `Password` varchar(255) NOT NULL,
  `DOB` date NOT NULL,
  `AccountCreationDate` date DEFAULT NULL,
  `AccountActivity` varchar(255) DEFAULT NULL,
  `TotalPurchase` decimal(10,2) DEFAULT NULL,
  `LoyaltyPoints` int(11) DEFAULT NULL,
  `NewsletterSubscription` varchar(10) DEFAULT NULL,
  `ApplicablePromotion` varchar(100) DEFAULT NULL,
  `CustomerID` int(11) DEFAULT NULL
);

--

-- Dumping data for table `customeraccounts`
--

INSERT INTO `customeraccounts` (`AccountID`, `UserName`, `Password`, `DOB`,
`AccountCreationDate`, `AccountActivity`, `TotalPurchase`, `LoyaltyPoints`,
`NewsletterSubscription`, `ApplicablePromotion`, `CustomerID`) VALUES
(1, 'john.doe', 'password123', '1985-05-15', '2024-01-01', 'Review product', 2500.00, 300,
'Yes', 'Mid-year Sale', 1),
(2, 'jane.nguyen', 'securepass', '1990-07-20', '2024-02-15', 'Share product', 1500.00, 150,
'No', 'Birthday Discount', 2),
(3, 'alice.wang', 'alicepass', '1988-03-22', '2024-03-10', 'Save product', 3500.00, 450,
'Yes', NULL, 3),
(4, 'crystal.huynh', 'crystal123', '1995-03-20', '2024-03-14', 'Share Product', 2000.00, 200,
'No', 'Birthday Discount', 4),
(5, 'peter.lee', 'peter456', '2000-04-24', '2024-06-22', 'Share Product', 3000.00, 300, 'Yes',
'Summer Sale', 5);

-- -----

--

-- Table structure for table `customers`
--

```

```

CREATE TABLE `customers` (
  `CustomerID` int(11) NOT NULL,
  `Name` varchar(100) NOT NULL,
  `Email` varchar(255) DEFAULT NULL,
  `Address` varchar(255) NOT NULL,
  `City` varchar(100) NOT NULL,
  `PostalCode` varchar(100) NOT NULL,
  `Country` varchar(100) NOT NULL,
  `PhoneNo` bigint(20) DEFAULT NULL,
  `referrerID` int(11) DEFAULT NULL
);

--
-- Dumping data for table `customers`
--

INSERT INTO `customers` (`CustomerID`, `Name`, `Email`, `Address`, `City`, `PostalCode`,
`Country`, `PhoneNo`, `referrerID`) VALUES
(1, 'John Doe', 'john.doe@example.com', '123 Elm St', 'Springfield', '02310', 'USA',
1234567890, NULL),
(2, 'Jane Nguyen', 'jane.nguyen@example.com', '456 Pham Ngoc Thach St', 'HaNoi',
'V3I4X6', 'Vietnam', 2345678901, 1),
(3, 'Alice Wang', 'alice.wang@gmail.com', '36 Pine St', 'Guangzhou', '510832', 'China',
13724356598, 2),
(4, 'Crystal Huynh', 'crystal.huynh@gmail.com', '120 Hoan Kiem St', 'HaNoi', 'V5M6A8',
'Vietnam', 2346662020, 2),
(5, 'Peter Lee', 'peter.lee@gmail.com', '420 Hamilton St', 'Seattle', '08110', 'USA',
1238887220, 4);

-- -----

--
-- Table structure for table `orderdetails`
--

CREATE TABLE `orderdetails` (
  `OrderDetailID` int(15) NOT NULL,
  `OrderID` int(11) NOT NULL,

```

```
`ProductID` int(11) NOT NULL,  
`Quantity` int(11) NOT NULL CHECK (`Quantity` > 0)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `orderdetails`

--

```
INSERT INTO `orderdetails` (`OrderDetailID`, `OrderID`, `ProductID`, `Quantity`) VALUES  
(1, 1, 1, 1),  
(2, 2, 3, 2),  
(3, 3, 5, 1),  
(4, 4, 2, 2),  
(5, 5, 1, 1);
```

-- -----

--

-- Table structure for table `orders`

--

```
CREATE TABLE `orders` (  
  `OrderID` int(11) NOT NULL,  
  `CustomerID` int(11) NOT NULL,  
  `OrderDate` date NOT NULL,  
  `Total` int(11) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `orders`

--

```
INSERT INTO `orders` (`OrderID`, `CustomerID`, `OrderDate`, `Total`) VALUES  
(1, 1, '2024-01-01', 75),  
(2, 2, '2024-01-02', 30),  
(3, 2, '2024-01-03', 150),  
(4, 3, '2024-03-14', 60),  
(5, 4, '2024-03-15', 40);
```

--

-- Table structure for table `products`

--

```
CREATE TABLE `products` (  
  `ProductID` int(11) NOT NULL,  
  `ProductName` varchar(100) NOT NULL,  
  `CategoryID` int(11) NOT NULL,  
  `Price` decimal(10,2) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `products`

--

```
INSERT INTO `products` (`ProductID`, `ProductName`, `CategoryID`, `Price`) VALUES  
(1, 'Babie Bear Baguette Crossbody Bag', 1, 29.99),  
(2, 'Shoku-pan Bear Tote Bag', 2, 24.99),  
(3, 'Banada Fruits Sando Plushie', 3, 19.99),  
(4, 'Melon-pan Coin Pouch', 4, 9.99),  
(5, 'Bunny Cupcake Backpack', 5, 49.99),  
(6, 'Pudding Lollipop Keychain', 6, 8.99);
```

--

-- Table structure for table `shippings`

--

```
CREATE TABLE `shippings` (  
  `ShippingID` int(11) NOT NULL,  
  `OrderID` int(11) NOT NULL,  
  `ProviderID` int(11) NOT NULL,  
  `Cost` decimal(10,2) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `shippings`

--

```
INSERT INTO `shippings` (`ShippingID`, `OrderID`, `ProviderID`, `Cost`) VALUES
(1, 1, 1, 10.00),
(2, 2, 2, 35.00),
(3, 3, 2, 45.00),
(4, 4, 4, 50.00),
(5, 5, 5, 30.00);
```

-- -----

--

-- Table structure for table `transportation_providers`

--

```
CREATE TABLE `transportation_providers` (
  `ProviderID` int(11) NOT NULL,
  `ProviderName` varchar(100) NOT NULL,
  `Email` varchar(255) NOT NULL,
  `PhoneNumber` bigint(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

--

-- Dumping data for table `transportation_providers`

--

```
INSERT INTO `transportation_providers` (`ProviderID`, `ProviderName`, `Email`,
`PhoneNumber`) VALUES
(1, 'CanadaPost', 'cp@canadapost.com', 9876543210),
(2, 'International', 'international@quickship.com', 9876543211),
(3, 'Backup', 'backup@reliabletrans.com', 9876543212),
(4, 'DHL', 'dhl@globalshipping.com', 9876543213),
(5, 'FedEx', 'fedex@homedelivery.com', 9876543214);
```

--

```
-- Indexes for dumped tables
--

--
-- Indexes for table `categories`
--
ALTER TABLE `categories`
  ADD PRIMARY KEY (`CategoryID`),
  ADD UNIQUE KEY `CategoryName` (`CategoryName`);

--
-- Indexes for table `customeraccounts`
--
ALTER TABLE `customeraccounts`
  ADD PRIMARY KEY (`AccountID`),
  ADD UNIQUE KEY `UserName` (`UserName`),
  ADD UNIQUE KEY `CustomerID` (`CustomerID`);

--
-- Indexes for table `customers`
--
ALTER TABLE `customers`
  ADD PRIMARY KEY (`CustomerID`),
  ADD UNIQUE KEY `Email` (`Email`),
  ADD UNIQUE KEY `PhoneNo` (`PhoneNo`),
  ADD KEY `fk_customers_referral` (`referrerID`);

--
-- Indexes for table `orderdetails`
--
ALTER TABLE `orderdetails`
  ADD PRIMARY KEY (`OrderDetailID`),
  ADD KEY `fk_orders` (`OrderID`),
  ADD KEY `fk_products` (`ProductID`);

--
-- Indexes for table `orders`
--
```

```

ALTER TABLE `orders`
  ADD PRIMARY KEY (`OrderID`),
  ADD KEY `fk_Orders_Customers` (`CustomerID`);

--

-- Indexes for table `products`
--
ALTER TABLE `products`
  ADD PRIMARY KEY (`ProductID`),
  ADD UNIQUE KEY `ProductName` (`ProductName`),
  ADD KEY `fk_Categories` (`CategoryID`);

--

-- Indexes for table `shippings`
--
ALTER TABLE `shippings`
  ADD PRIMARY KEY (`ShippingID`),
  ADD UNIQUE KEY `OrderID` (`OrderID`),
  ADD KEY `fk_Providers_Shipping` (`ProviderID`);

--

-- Indexes for table `transportation_providers`
--
ALTER TABLE `transportation_providers`
  ADD PRIMARY KEY (`ProviderID`),
  ADD UNIQUE KEY `Email` (`Email`);

--

-- AUTO_INCREMENT for dumped tables
--

--

-- AUTO_INCREMENT for table `categories`
--
ALTER TABLE `categories`
  MODIFY `CategoryID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--

```

```

-- AUTO_INCREMENT for table `customeraccounts`
--
ALTER TABLE `customeraccounts`
  MODIFY `AccountID` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `orderdetails`
--
ALTER TABLE `orderdetails`
  MODIFY `OrderDetailID` int(15) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `orders`
--
ALTER TABLE `orders`
  MODIFY `OrderID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `products`
--
ALTER TABLE `products`
  MODIFY `ProductID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

--
-- AUTO_INCREMENT for table `shippings`
--
ALTER TABLE `shippings`
  MODIFY `ShippingID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT for table `transportation_providers`
--
ALTER TABLE `transportation_providers`
  MODIFY `ProviderID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- Constraints for dumped tables
--

```



```

--
-- Constraints for table `customeraccounts`
--
ALTER TABLE `customeraccounts`
  ADD CONSTRAINT `account_belongs_1customer` FOREIGN KEY (`CustomerID`)
REFERENCES `customers` (`CustomerID`);

--
-- Constraints for table `customers`
--
ALTER TABLE `customers`
  ADD CONSTRAINT `fk_customers_referral` FOREIGN KEY (`referrerID`) REFERENCES
`customers` (`CustomerID`);

--
-- Constraints for table `orderdetails`
--
ALTER TABLE `orderdetails`
  ADD CONSTRAINT `fk_orders` FOREIGN KEY (`OrderID`) REFERENCES `orders`
(`OrderID`),
  ADD CONSTRAINT `fk_products` FOREIGN KEY (`ProductID`) REFERENCES `products`
(`ProductID`);

--
-- Constraints for table `orders`
--
ALTER TABLE `orders`
  ADD CONSTRAINT `fk_Orders_Customers` FOREIGN KEY (`CustomerID`) REFERENCES
`customers` (`CustomerID`);

--
-- Constraints for table `products`
--
ALTER TABLE `products`
  ADD CONSTRAINT `fk_Categories` FOREIGN KEY (`CategoryID`) REFERENCES `categories`
(`CategoryID`);

```

```
--  
-- Constraints for table `shippings`  
--  
ALTER TABLE `shippings`  
  ADD CONSTRAINT `fk_Orders_Shipping` FOREIGN KEY (`OrderID`) REFERENCES `orders`  
  (`OrderID`),  
  ADD CONSTRAINT `fk_Providers_Shipping` FOREIGN KEY (`ProviderID`) REFERENCES  
  `transportation_providers` (`ProviderID`);  
COMMIT;  
  
/!*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/!*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/!*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```