Report

Table of Contents

1. Problem Description

The EasyDelivery Company provides food/products delivery service for customers. Customer can place takeout/delivery orders through the EasyDelivery App. Now, EasyDelivery would like one relational database to store the information about their management system to be able to carry out their work in an organized way. They have some major modules such as Employee and Customer, Restaurant and Order.

Employee's information is stored in the database, such as employee ID, Name (First, Middle, Last), Address, Gender, Date of Birth (should be constrained as 16 years or older) and Phone number (one person may have more than one phone number). The Employee's ID should have the format "EXXX" ("E" followed by exactly 3 digits), where X is a number from 0-9. (Hint: you can use regexp_like() function).

Details of a customer such as ID, Name (First, Middle, Last), Delivery Address and Phone number, Joining Date are recorded. You are free to define the format of customer ID. A Customer may have multiple delivery addresses. Customer is further classified as Ordinary Customer and Silver Member. Every silver member owns one unique member card issued by Staff. Card issuing date is stored. Employee and Silver Member can be Premium Member who have extra privileges. Every premium member owns a Meal Pass, which can allow the member has free delivery fee for their orders 10 times monthly. The Meal pass has effective dates and expiration dates.

Employee is further classified as Area Manager, Deliverer and Staff. The start date of the designation of each employee is stored. Every deliverer is supervised by an area manager, an area manager may supervise many deliverers. Every deliverer needs to register at least one Vehicle for delivery. Vehicle's information such as Plate Number, Maker, Model and Color are stored. An area manager works in an Area (Richardson, Frisco, Plano, Dallas, etc.), and is responsible for making contracts with the Shops in his working area. The contract start time of each shop is stored.

There are two types of Shops: Restaurants and Supermarkets. Restaurants' Name, Type (Fast Food, BBQ, Buffet, Drink, etc.), Address, Area, Business Phone Number are recorded. A Restaurant can have more than one type. Supermarkets' Name, Address, Business Phone Number are recorded. Supermarkets sell various Products; one product may be sold in different supermarkets at different price. Supermarkets maintain Inventories of their products showing how many products in stock. Every product has ID which is only guaranteed unique in each store, but different product may have the same ID in different stores. Product names and description are also stored.

Each Shop opens and closes at specific times following a schedule table. Shops may have different open and close time in different days in a week. Shops sometimes may provide Promotion. Each Promotion includes a unique Promotion code, and its description. When

Customers place orders, customer may choose to use promotion codes. Customers can make Comments to the Shops. The comments include rating score (can be 1,2,3,4,5), and comment contents.

Details of Orders such as Order ID, Order Contents, Subtotals are recorded. Each order belongs to corresponding Shops. Customer can select different payment types to pay for the order. Details of Payment such as unique Payment Confirm Number, Payment Type, Payment Time is recorded. Deliverer will deliver the order with a specific registered vehicle.

2. Project Questions

1. Is the ability to model superclass/subclass relationships likely to be important in the shopping mall management system like above? Why or why not?

   I think it's important, because we can create various tables with different characteristics by inheriting from a superclass, which helps to manage the database efficiently and increase the scalability of the database. It is suitable for shop delivery systems that must deal with various persons. By defining common attributes and relations for the superclass using superclass/subclass relationships, we may reduce redundancy. For example, "Employee" and "Customer" tables inherit "Person" superclass. With help of superclass "PERSON", it is not necessary to create new attributes for each table with similar attributes such as the name of the person. Hence, data can be managed efficiently without duplication. In addition, while creating other types of "PERSON" entities in future, it can be easily created by inheriting "PERSON".

   We were able to avoid having to declare First_name, Middle_name, and Last_name for the relations Employee and Customer, as well as their subclasses, by using the Person superclass.

   For both the restaurant and the supermarket shop types, we defined the common relations involving Area Manager, Schedule, Order, Promotion, and Comment using the Shop superclass.

2. Can you think of 5 more business rules (other than those explicitly described above) that are likely to be used in a shopping mall environment? Add your rules to the above requirement to be implemented.

   - Customers may only review and leave comments for a shop if they have recently (within the past two days) placed an order there.
   - We can add points to premium and silver customers' member cards in accordance with the value of their orders.
   - Promotional codes dedicated to employees.
   - Expedited delivery service: If your order is a part of a stacked delivery and you choose priority delivery, it will be delivered first.
   - Customer service module to store the details of customer queries.
   - Gift cards that may be bought, added to the wallet, and used to place orders.

- Ability to schedule recursive orders.

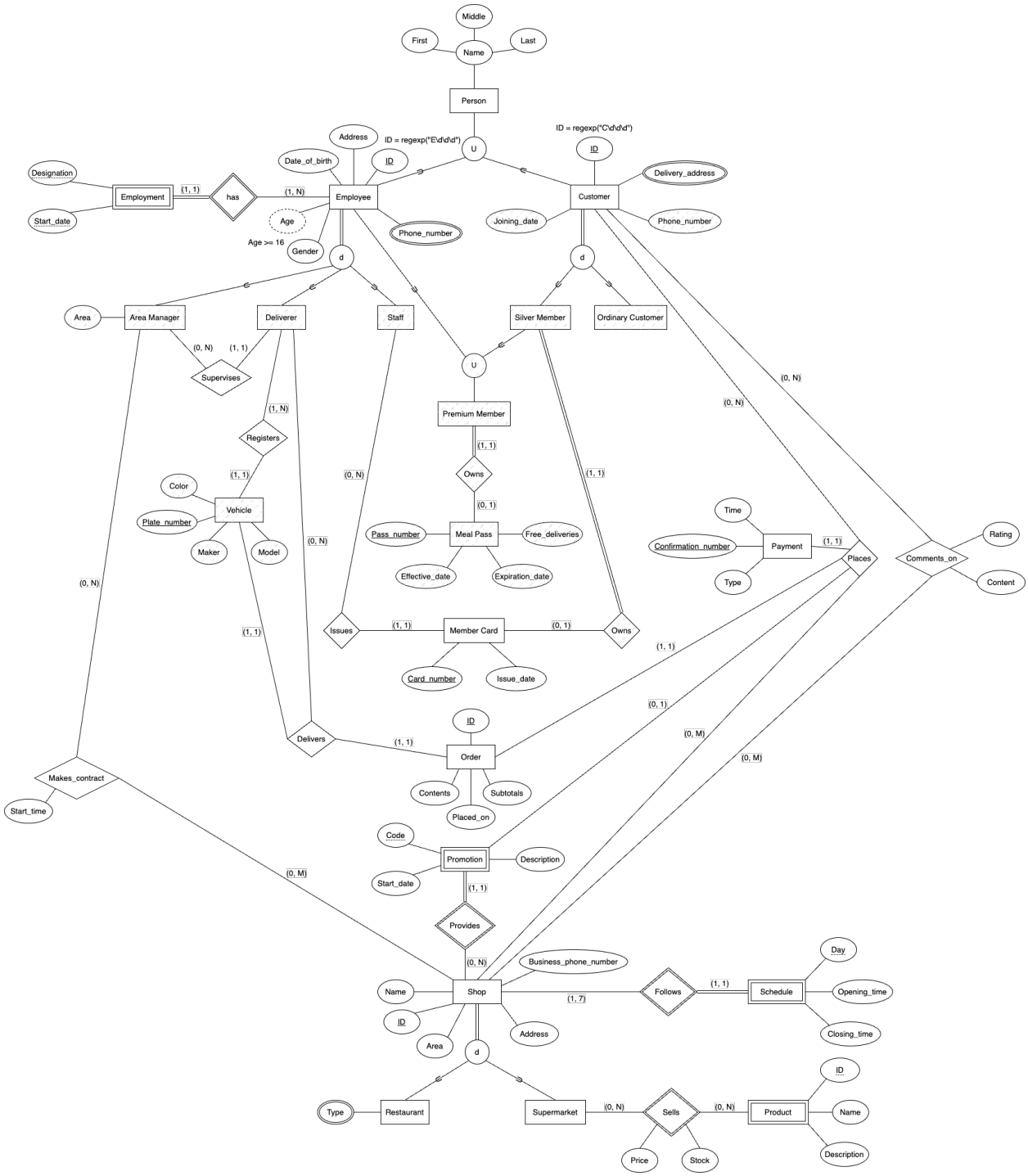
- Top deliverers should be assigned for delivery orders placed by premium and silver customer.

3. Justify using a Relational DBMS like Oracle for this project.

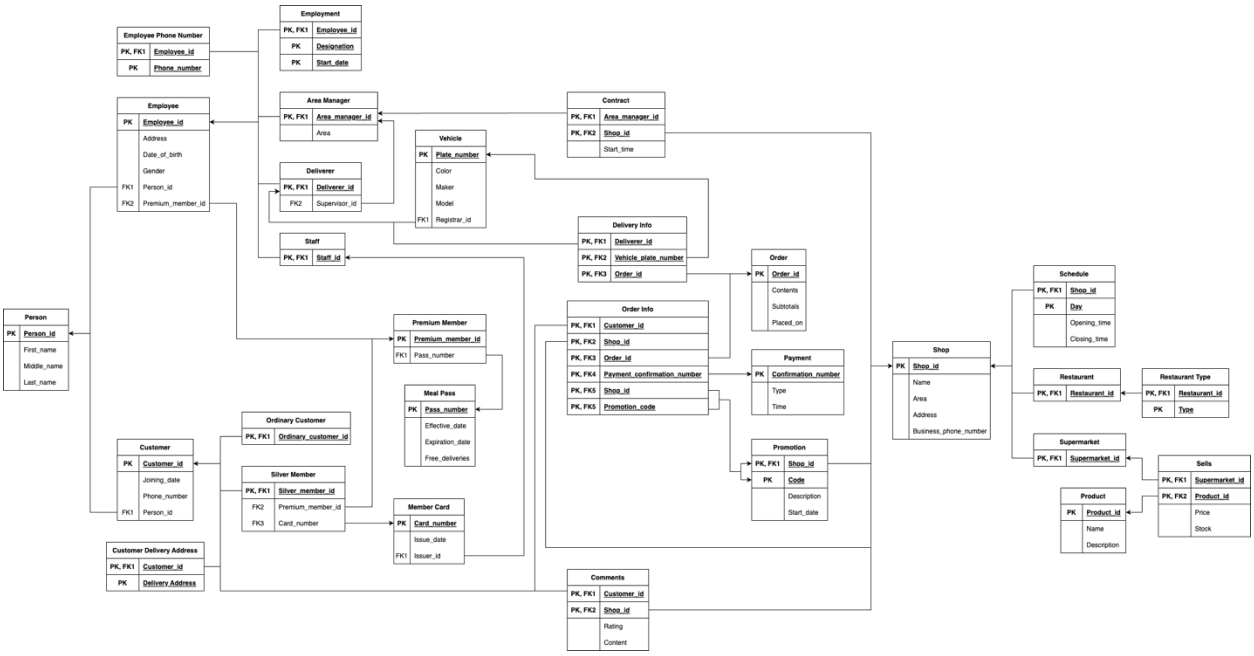
   For this project, a relational DBMS works well.
   1. The database's data won't change frequently and won't need to be updated constantly.
   2. Fast query times are preferred.

   These two findings support the use of a relational database management system.
   We used MySQL as the DBMS due to its speed, security, and ease of use.

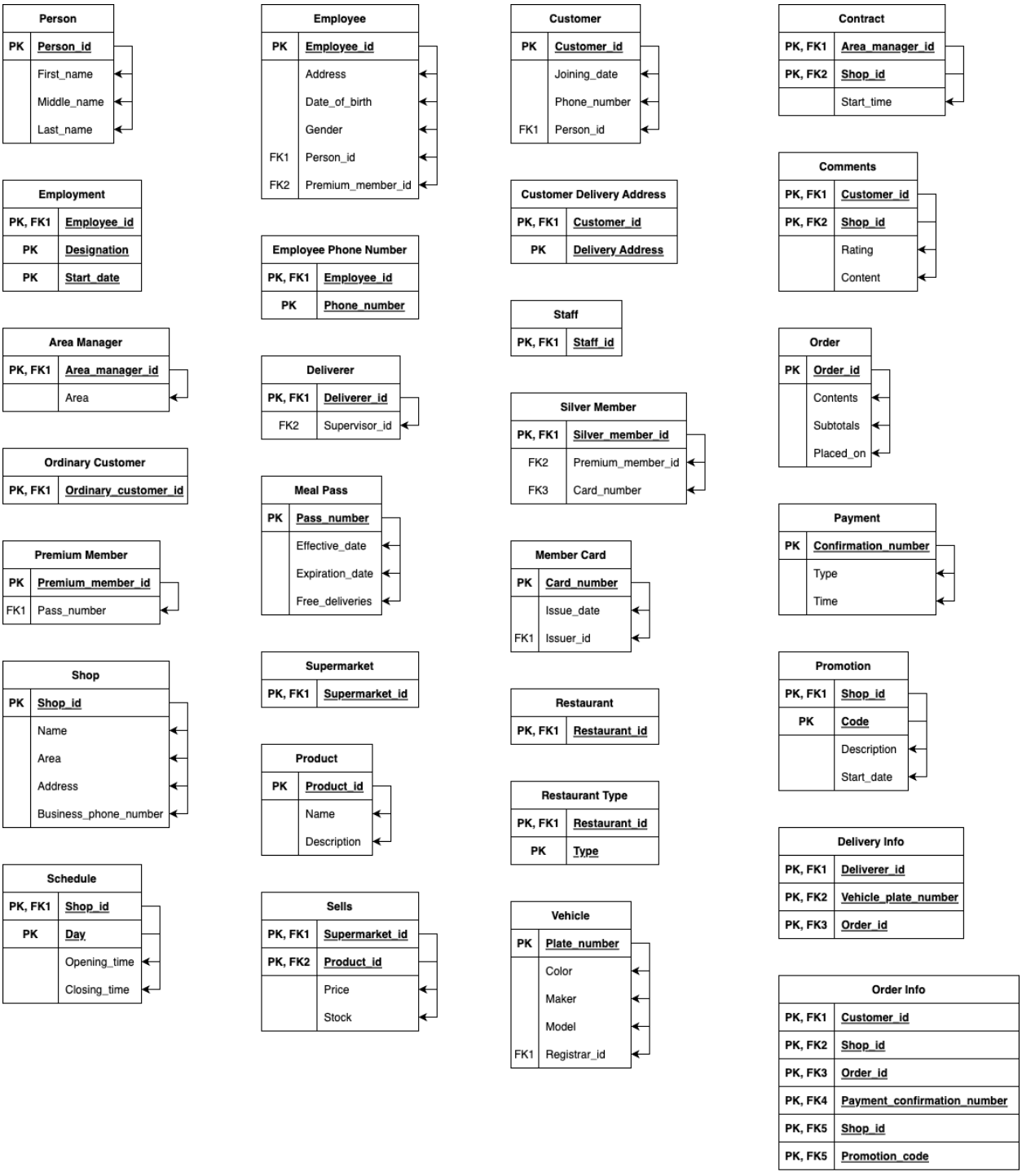
3. Enhanced Entity Relationship (EER) Diagram



4. Relational Schema



5. Dependency Diagram



6. SQL Statements

6.1. Creating database, tables, and all other structures

```
DROP SCHEMA IF EXISTS EASY_DELIVERY;
CREATE SCHEMA EASY_DELIVERY;

USE EASY_DELIVERY;

-- PERSON
DROP TABLE IF EXISTS PERSON;
CREATE TABLE PERSON (
    Person_id INT(9) NOT NULL AUTO_INCREMENT,
    First_name VARCHAR(63) NOT NULL,
    Middle_name VARCHAR(63),
    Last_name VARCHAR(63) NOT NULL,
    PRIMARY KEY (Person_id)
);

-- MEAL_PASS
DROP TABLE IF EXISTS MEAL_PASS;
CREATE TABLE MEAL_PASS (
    Pass_number INT(9) NOT NULL AUTO_INCREMENT,
    Effective_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    Expiration_date DATETIME NOT NULL,
    Free_deliveries INT(2) DEFAULT 10,
    PRIMARY KEY (Pass_number)
);

DROP TRIGGER IF EXISTS setDefaultExpirationDate;
CREATE TRIGGER setDefaultExpirationDate
BEFORE INSERT ON MEAL_PASS
FOR EACH ROW
SET NEW.Expiration_date = DATE_ADD(New.Effective_date, INTERVAL 1 MONTH);

-- PREMIUM_MEMBER
DROP TABLE IF EXISTS PREMIUM_MEMBER;
CREATE TABLE PREMIUM_MEMBER (
    Premium_member_id INT(9) NOT NULL AUTO_INCREMENT,
    Pass_number INT(9) NOT NULL,
    PRIMARY KEY (Premium_member_id),
    FOREIGN KEY (Pass_number) REFERENCES MEAL_PASS(Pass_number)
);

-- EMPLOYEE
DROP TABLE IF EXISTS EMPLOYEE;
```

```
CREATE TABLE EMPLOYEE (
    Employee_id VARCHAR(4) NOT NULL,
    Address VARCHAR(511) NOT NULL,
    Date_of_birth DATE NOT NULL,
    Gender ENUM('Male', 'Female') NOT NULL,
    Person_id INT(9) NOT NULL,
    Premium_member_id INT(9),
    PRIMARY KEY (Employee_id),
    FOREIGN KEY (Person_id) REFERENCES PERSON(Person_id),
    FOREIGN KEY (Premium_member_id) REFERENCES PREMIUM_MEMBER(Premium_member_id),
    CONSTRAINT Employee_id_check CHECK (REGEXP_LIKE(Employee_id, 'E[0-9]{3}'))
);

DROP PROCEDURE IF EXISTS Validate_employee_age;
DELIMITER $$
CREATE PROCEDURE Validate_employee_age(
    IN Date_of_birth DATE
)
DETERMINISTIC
NO SQL
BEGIN
    IF ((DATEDIFF(CURRENT_TIMESTAMP, Date_of_birth) / 365.25) - (CASE WHEN
MONTH(Date_of_birth) = MONTH(CURRENT_TIMESTAMP) AND DAY(Date_of_birth) >
DAY(CURRENT_TIMESTAMP) THEN 1 ELSE 0 END)) < 16 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Age must be not be less than 16';
    END IF;
END$$
DELIMITER ;

DROP TRIGGER IF EXISTS Validate_employee_insert;
DELIMITER $$
CREATE TRIGGER Validate_employee_insert
BEFORE INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
    CALL Validate_employee_age(NEW.Date_of_birth);
END$$
DELIMITER ;

DROP TRIGGER IF EXISTS Validate_employee_update;
DELIMITER $$
CREATE TRIGGER Validate_employee_update
BEFORE UPDATE ON EMPLOYEE
FOR EACH ROW
BEGIN
    CALL Validate_employee_age(NEW.Date_of_birth);
END$$
```

```
DELIMITER ;

-- EMPLOYEE_PHONE_NUMBER
DROP TABLE IF EXISTS EMPLOYEE_PHONE_NUMBER;
CREATE TABLE EMPLOYEE_PHONE_NUMBER (
    Employee_id VARCHAR(4) NOT NULL,
    Phone_number VARCHAR(10) NOT NULL,
    PRIMARY KEY (Employee_id, Phone_number),
    FOREIGN KEY (Employee_id) REFERENCES EMPLOYEE(Employee_id)
);

-- EMPLOYMENT
DROP TABLE IF EXISTS EMPLOYMENT;
CREATE TABLE EMPLOYMENT (
    Employee_id VARCHAR(4) NOT NULL,
    Designation ENUM('Area Manager', 'Deliverer', 'Staff') NOT NULL,
    Start_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Employee_id, Designation, Start_date),
    FOREIGN KEY (Employee_id) REFERENCES EMPLOYEE(Employee_id)
);

-- AREA_MANAGER
DROP TABLE IF EXISTS AREA_MANAGER;
CREATE TABLE AREA_MANAGER (
    Area_manager_id VARCHAR(4) NOT NULL,
    Area VARCHAR(31) NOT NULL,
    PRIMARY KEY (Area_manager_id),
    FOREIGN KEY (Area_manager_id) REFERENCES EMPLOYEE(Employee_id)
);

-- DELIVERER
DROP TABLE IF EXISTS DELIVERER;
CREATE TABLE DELIVERER (
    Deliverer_id VARCHAR(4) NOT NULL,
    Supervisor_id VARCHAR(4) NOT NULL,
    PRIMARY KEY (Deliverer_id),
    FOREIGN KEY (Deliverer_id) REFERENCES EMPLOYEE(Employee_id),
    FOREIGN KEY (Supervisor_id) REFERENCES AREA_MANAGER(Area_manager_id)
);

-- STAFF
DROP TABLE IF EXISTS STAFF;
CREATE TABLE STAFF (
    Staff_id VARCHAR(4) NOT NULL,
    PRIMARY KEY (Staff_id),
    FOREIGN KEY (Staff_id) REFERENCES EMPLOYEE(Employee_id)
);
```

```sql
-- VEHICLE
DROP TABLE IF EXISTS VEHICLE;
CREATE TABLE VEHICLE (
    Plate_number VARCHAR(7) NOT NULL,
    Color VARCHAR(31) NOT NULL,
    Maker VARCHAR(31) NOT NULL,
    Model VARCHAR(31) NOT NULL,
    Registrar_id VARCHAR(4) NOT NULL,
    PRIMARY KEY (Plate_number),
    FOREIGN KEY (Registrar_id) REFERENCES DELIVERER(Deliverer_id)
);

-- MEMBER_CARD
DROP TABLE IF EXISTS MEMBER_CARD;
CREATE TABLE MEMBER_CARD (
    Card_number INT(9) NOT NULL AUTO_INCREMENT,
    Issue_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    Issuer_id VARCHAR(4) NOT NULL,
    PRIMARY KEY (Card_number),
    FOREIGN KEY (Issuer_id) REFERENCES STAFF(Staff_id)
);

-- CUSTOMER
DROP TABLE IF EXISTS CUSTOMER;
CREATE TABLE CUSTOMER (
    Customer_id VARCHAR(4) NOT NULL,
    Joining_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    Phone_number VARCHAR(10) NOT NULL,
    Person_id INT(9) NOT NULL,
    PRIMARY KEY (Customer_id),
    FOREIGN KEY (Person_id) REFERENCES PERSON(Person_id),
    CONSTRAINT Customer_id_check CHECK (REGEXP_LIKE(Customer_id, 'C[0-9]{3}'))
);

-- CUSTOMER_DELIVERY_ADDRESS
DROP TABLE IF EXISTS CUSTOMER_DELIVERY_ADDRESS;
CREATE TABLE CUSTOMER_DELIVERY_ADDRESS (
    Customer_id VARCHAR(4) NOT NULL,
    Delivery_address VARCHAR(511) NOT NULL,
    PRIMARY KEY (Customer_id, Delivery_address),
    FOREIGN KEY (Customer_id) REFERENCES CUSTOMER(Customer_id)
);

-- ORDINARY_CUSTOMER
DROP TABLE IF EXISTS ORDINARY_CUSTOMER;
CREATE TABLE ORDINARY_CUSTOMER (
```

```
    Ordinary_customer_id VARCHAR(4) NOT NULL,
    PRIMARY KEY (Ordinary_customer_id),
    FOREIGN KEY (Ordinary_customer_id) REFERENCES CUSTOMER(Customer_id)
);

-- SILVER_MEMBER
DROP TABLE IF EXISTS SILVER_MEMBER;
CREATE TABLE SILVER_MEMBER (
    Silver_member_id VARCHAR(4) NOT NULL,
    Premium_member_id INT(9),
    Card_number INT(9) NOT NULL,
    PRIMARY KEY (Silver_member_id),
    FOREIGN KEY (Silver_member_id) REFERENCES CUSTOMER(Customer_id),
    FOREIGN KEY (Premium_member_id) REFERENCES PREMIUM_MEMBER(Premium_member_id),
    FOREIGN KEY (Card_number) REFERENCES MEMBER_CARD(Card_number)
);

-- SHOP
DROP TABLE IF EXISTS SHOP;
CREATE TABLE SHOP (
    Shop_id INT(9) NOT NULL AUTO_INCREMENT,
    Name VARCHAR(63) NOT NULL,
    Area VARCHAR(31) NOT NULL,
    Address VARCHAR(255) NOT NULL,
    Business_phone_number VARCHAR(10) NOT NULL,
    PRIMARY KEY (Shop_id)
);

-- RESTAURANT
DROP TABLE IF EXISTS RESTAURANT;
CREATE TABLE RESTAURANT (
    Restaurant_id INT(9) NOT NULL,
    PRIMARY KEY (Restaurant_id),
    FOREIGN KEY (Restaurant_id) REFERENCES SHOP(Shop_id)
);

-- RESTAURANT_TYPE
DROP TABLE IF EXISTS RESTAURANT_TYPE;
CREATE TABLE RESTAURANT_TYPE (
    Restaurant_id INT(9) NOT NULL,
    Type VARCHAR(31) NOT NULL,
    PRIMARY KEY (Restaurant_id, Type),
    FOREIGN KEY (Restaurant_id) REFERENCES RESTAURANT(Restaurant_id)
);

-- SUPERMARKET
DROP TABLE IF EXISTS SUPERMARKET;
```

```sql
CREATE TABLE SUPERMARKET (
    Supermarket_id INT(9) NOT NULL,
    PRIMARY KEY (Supermarket_id),
    FOREIGN KEY (Supermarket_id) REFERENCES SHOP(Shop_id)
);

-- PRODUCT
DROP TABLE IF EXISTS PRODUCT;
CREATE TABLE PRODUCT (
    Product_id INT(9) NOT NULL AUTO_INCREMENT,
    Name VARCHAR(127) NOT NULL,
    Description VARCHAR(1023) NOT NULL,
    PRIMARY KEY (Product_id)
);

-- SELLS
DROP TABLE IF EXISTS SELLS;
CREATE TABLE SELLS (
    Supermarket_id INT(9) NOT NULL,
    Product_id INT(9) NOT NULL,
    Price FLOAT(8, 2) NOT NULL,
    Stock INT(6) NOT NULL,
    PRIMARY KEY (Supermarket_id, Product_id),
    FOREIGN KEY (Supermarket_id) REFERENCES SUPERMARKET(Supermarket_id),
    FOREIGN KEY (Product_id) REFERENCES PRODUCT(Product_id)
);

-- SCHEDULE
DROP TABLE IF EXISTS SCHEDULE;
CREATE TABLE SCHEDULE (
    Shop_id INT(9) NOT NULL,
    Day ENUM('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday') NOT NULL,
    Opening_time TIME NOT NULL,
    Closing_time TIME NOT NULL,
    PRIMARY KEY (Shop_id, Day),
    FOREIGN KEY (Shop_id) REFERENCES SHOP(Shop_id)
);

-- PROMOTION
DROP TABLE IF EXISTS PROMOTION;
CREATE TABLE PROMOTION (
    Shop_id INT(9) NOT NULL,
    Code VARCHAR(15) NOT NULL,
    Description VARCHAR(255) NOT NULL,
    Start_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Shop_id, Code),
```

```
    FOREIGN KEY (Shop_id) REFERENCES SHOP(Shop_id)
);

-- ORDER
DROP TABLE IF EXISTS `ORDER`;
CREATE TABLE `ORDER` (
    Order_id INT(9) NOT NULL AUTO_INCREMENT,
    Contents VARCHAR(511) NOT NULL,
    Subtotals FLOAT(9, 2) NOT NULL,
    Placed_on DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Order_id)
);

-- PAYMENT
DROP TABLE IF EXISTS PAYMENT;
CREATE TABLE PAYMENT (
    Confirmation_number INT(9) NOT NULL AUTO_INCREMENT,
    Type ENUM('Credit/Debit Card', 'Cash', 'Check', 'EFT', 'Crypto') NOT NULL,
    Time DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Confirmation_number)
);

-- DELIVERY_INFO
DROP TABLE IF EXISTS DELIVERY_INFO;
CREATE TABLE DELIVERY_INFO (
    Deliverer_id VARCHAR(4) NOT NULL,
    Vehicle_plate_number VARCHAR(7) NOT NULL,
    Order_id INT(9) NOT NULL,
    PRIMARY KEY (Deliverer_id, Vehicle_plate_number, Order_id),
    FOREIGN KEY (Deliverer_id) REFERENCES DELIVERER(Deliverer_id),
    FOREIGN KEY (Vehicle_plate_number) REFERENCES VEHICLE(Plate_number),
    FOREIGN KEY (Order_id) REFERENCES `ORDER`(Order_id)
);

-- ORDER_INFO
DROP TABLE IF EXISTS ORDER_INFO;
CREATE TABLE ORDER_INFO (
    Order_id INT(9) NOT NULL,
    Shop_id INT(9) NOT NULL,
    Customer_id VARCHAR(4) NOT NULL,
    Payment_confirmation_number INT(9) NOT NULL,
    Promotion_code VARCHAR(15) DEFAULT 'N/A',
    PRIMARY KEY (Order_id, Shop_id, Customer_id, Payment_confirmation_number,
Promotion_code),
    FOREIGN KEY (Order_id) REFERENCES `ORDER`(Order_id),
    FOREIGN KEY (Shop_id) REFERENCES SHOP(Shop_id),
    FOREIGN KEY (Customer_id) REFERENCES CUSTOMER(Customer_id),
```

```
    FOREIGN KEY (Payment_confirmation_number) REFERENCES PAYMENT(Confirmation_number),
    FOREIGN KEY (Shop_id, Promotion_code) REFERENCES PROMOTION(Shop_id, Code)
);

-- CONTRACT
DROP TABLE IF EXISTS CONTRACT;
CREATE TABLE CONTRACT (
    Shop_id INT(9) NOT NULL,
    Area_manager_id VARCHAR(4) NOT NULL,
    Start_time DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Shop_id, Area_manager_id),
    FOREIGN KEY (Shop_id) REFERENCES SHOP(Shop_id),
    FOREIGN KEY (Area_manager_id) REFERENCES AREA_MANAGER(Area_manager_id)
);

-- COMMENTS
DROP TABLE IF EXISTS COMMENTS;
CREATE TABLE COMMENTS (
    Shop_id INT(9) NOT NULL,
    Customer_id VARCHAR(4) NOT NULL,
    Rating ENUM('1', '2', '3', '4', '5') NOT NULL,
    Content VARCHAR(255) NOT NULL,
    PRIMARY KEY (Shop_id, Customer_id),
    FOREIGN KEY (Shop_id) REFERENCES SHOP(Shop_id),
    FOREIGN KEY (Customer_id) REFERENCES CUSTOMER(Customer_id)
);
```

6.2. Populating data into tables

```
-- PERSON
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("John", "B", "Smith");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Franklin", "T",
"Wong");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Alicia", "J",
"Zelaya");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Jennifer", "S",
"Wallace");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Ramesh", "K",
"Narayan");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Joyce", "A",
"English");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("Ahmad", "V",
"Jabbar");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("James", "E", "Borg");
INSERT INTO PERSON (First_name, Middle_name, Last_name) VALUES ("John", NULL, "Doe");

-- MEAL_PASS
INSERT INTO MEAL_PASS (Effective_date) VALUES ('2022-01-01');
INSERT INTO MEAL_PASS (Effective_date) VALUES ('2022-04-13');
INSERT INTO MEAL_PASS (Effective_date) VALUES ('2022-01-01');
INSERT INTO MEAL_PASS (Effective_date) VALUES ('2022-04-01');
INSERT INTO MEAL_PASS (Effective_date) VALUES ('2022-09-09');

-- PREMIUM_MEMBER
INSERT INTO PREMIUM_MEMBER (Pass_number) VALUES (1);
INSERT INTO PREMIUM_MEMBER (Pass_number) VALUES (2);
INSERT INTO PREMIUM_MEMBER (Pass_number) VALUES (3);
INSERT INTO PREMIUM_MEMBER (Pass_number) VALUES (4);
INSERT INTO PREMIUM_MEMBER (Pass_number) VALUES (5);

-- EMPLOYEE
INSERT INTO EMPLOYEE VALUES ('E001', '731 Fondren, Houston, TX', '1965-01-09', 'Male',
1, 4);
INSERT INTO EMPLOYEE VALUES ('E002', '638 Voss, Houston, TX', '1955-11-08', 'Male', 2,
1);
INSERT INTO EMPLOYEE VALUES ('E003', '3321 Castle, Spring, TX', '1968-01-19',
'Female', 3, 3);
INSERT INTO EMPLOYEE VALUES ('E004', '291 Berry, Bellaire, TX', '1941-06-20',
'Female', 4, NULL);
INSERT INTO EMPLOYEE VALUES ('E005', '601 Causley Avenue, Austin, TX', '1962-09-15',
'Male', 5, NULL);
INSERT INTO EMPLOYEE VALUES ('E006', '800 W. Campbell Road, Richardson, TX', '2000-09-
15', 'Female', 6, NULL);
```

```
INSERT INTO EMPLOYEE VALUES ('E007', '5631 Rice, Houston, TX', '1974-07-12', 'Male',
7, NULL);
INSERT INTO EMPLOYEE VALUES ('E008', '980 Dallas, Houston, TX', '1983-08-19', 'Male',
8, NULL);
INSERT INTO EMPLOYEE VALUES ('E009', '975 Fire Oak, Humble, TX', '1984-03-28', 'Male',
9, NULL);

-- EMPLOYEE_PHONE_NUMBER
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E001', '1234567890');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E002', '4534534533');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E003', '9876543214');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E004', '9879879872');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E005', '6668844445');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E005', '5684182639');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E006', '6524372867');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E007', '7623864523');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E008', '3476543890');
INSERT INTO EMPLOYEE_PHONE_NUMBER VALUES ('E009', '8367297224');

-- EMPLOYMENT
INSERT INTO EMPLOYMENT VALUES ('E001', 'Area Manager', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E002', 'Staff', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E003', 'Deliverer', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E004', 'Staff', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E005', 'Deliverer', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E006', 'Area Manager', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E007', 'Area Manager', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E008', 'Area Manager', '2022-01-01');
INSERT INTO EMPLOYMENT VALUES ('E009', 'Area Manager', '2022-01-01');

-- AREA_MANAGER
INSERT INTO AREA_MANAGER VALUES ('E001', 'Plano');
INSERT INTO AREA_MANAGER VALUES ('E006', 'Richardson');
INSERT INTO AREA_MANAGER VALUES ('E007', 'Dallas');
INSERT INTO AREA_MANAGER VALUES ('E008', 'Irving');
INSERT INTO AREA_MANAGER VALUES ('E009', 'Plano');

-- DELIVERER
INSERT INTO DELIVERER VALUES ('E003', 'E001');
INSERT INTO DELIVERER VALUES ('E005', 'E001');

-- STAFF
INSERT INTO STAFF VALUES ('E002');
INSERT INTO STAFF VALUES ('E004');

-- VEHICLE
INSERT INTO VEHICLE VALUES ('77KR117', 'Silver', 'Hyundai', 'Elantra', 'E003');
```

```
INSERT INTO VEHICLE VALUES ('FGY5597', 'Black', 'Honda', 'Accord', 'E003');
INSERT INTO VEHICLE VALUES ('AAA0286', 'Blue', 'Hyundai', 'Accent', 'E005');

-- MEMBER_CARD
INSERT INTO MEMBER_CARD (Issue_date, Issuer_id) VALUES ('2022-04-13', 'E002');
INSERT INTO MEMBER_CARD (Issue_date, Issuer_id) VALUES ('2022-01-30', 'E002');
INSERT INTO MEMBER_CARD (Issue_date, Issuer_id) VALUES ('2022-09-09', 'E002');
INSERT INTO MEMBER_CARD (Issue_date, Issuer_id) VALUES ('2022-09-20', 'E004');

-- CUSTOMER
INSERT INTO CUSTOMER VALUES ('C001', '2022-01-01', '4534534533', 2);
INSERT INTO CUSTOMER VALUES ('C002', '2022-01-01', '9876543214', 3);
INSERT INTO CUSTOMER VALUES ('C003', '2022-01-01', '9879879872', 4);
INSERT INTO CUSTOMER VALUES ('C004', '2022-01-01', '6524372867', 6);
INSERT INTO CUSTOMER VALUES ('C005', '2022-01-01', '7623864523', 7);
INSERT INTO CUSTOMER VALUES ('C006', '2022-01-01', '3476543890', 8);

-- CUSTOMER_DELIVERY_ADDRESS
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C001', '638 Voss, Houston, TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C001', '800 W. Campbell Road,
Richardson, TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C003', '291 Berry, Bellaire, TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C004', '5631 Rice, Houston, TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C005', '7575 Frankford Rd, Dallas,
TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C005', '980 Dallas, Houston, TX');
INSERT INTO CUSTOMER_DELIVERY_ADDRESS VALUES ('C006', '450 Stone, Houston, TX');

-- ORDINARY_CUSTOMER
INSERT INTO ORDINARY_CUSTOMER VALUES ('C003');
INSERT INTO ORDINARY_CUSTOMER VALUES ('C004');

-- SILVER_MEMBER
INSERT INTO SILVER_MEMBER VALUES ('C001', 2, 1);
INSERT INTO SILVER_MEMBER VALUES ('C002', NULL, 2);
INSERT INTO SILVER_MEMBER VALUES ('C005', 5, 3);
INSERT INTO SILVER_MEMBER VALUES ('C006', NULL, 4);

-- SHOP
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ("McDonald's",
'Plano', '216 Coit Rd, Plano, TX', '9727580282');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Taco Bell',
'Richardson', '2828 Rutford Ave Ste 4, Richardson, TX', '9728837471');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Braums',
'Dallas', '17879 Preston Road, Dallas, TX', '9722487379');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('The Irishman
Pub', 'Dallas', '18101 Preston Rd Suite 102, Dallas, TX', '9727331852');
```

```
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Great Wall
Super Buffet', 'Plano', '901 W Plano Pkwy, Plano, TX', '9725788887');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('BBQ Chicken',
'Plano', '121 Spring Creek Pkwy Suite 311, Plano, TX', '4693669040');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ("Domino's",
'Plano', '4120 W 15th St #101, Plano, TX', '4695430420');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('KFC', 'Plano',
'1504 Custer Rd, Plano, TX', '9725094869');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Walmart',
'Irving', '1635 Market Pl Blvd, Irving, TX', '2145744517');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Target',
'Plano', '725 Hebron Pkwy, Plano, TX', '9724592605');
INSERT INTO SHOP (Name, Area, Address, Business_phone_number) VALUES ('Costco
Wholesale', 'Plano', '1701 Dallas Pkwy, Plano, TX', '9722462203');

-- RESTAURANT
INSERT INTO RESTAURANT VALUES (1);
INSERT INTO RESTAURANT VALUES (2);
INSERT INTO RESTAURANT VALUES (3);
INSERT INTO RESTAURANT VALUES (4);
INSERT INTO RESTAURANT VALUES (5);
INSERT INTO RESTAURANT VALUES (6);
INSERT INTO RESTAURANT VALUES (7);
INSERT INTO RESTAURANT VALUES (8);

-- RESTAURANT_TYPE
INSERT INTO RESTAURANT_TYPE VALUES (1, 'Fast Food');
INSERT INTO RESTAURANT_TYPE VALUES (2, 'Fast Food');
INSERT INTO RESTAURANT_TYPE VALUES (3, 'Fast Food');
INSERT INTO RESTAURANT_TYPE VALUES (4, 'Drinks');
INSERT INTO RESTAURANT_TYPE VALUES (5, 'Buffet');
INSERT INTO RESTAURANT_TYPE VALUES (6, 'BBQ');
INSERT INTO RESTAURANT_TYPE VALUES (7, 'Fast Food');
INSERT INTO RESTAURANT_TYPE VALUES (8, 'Fast Food');

-- SUPERMARKET
INSERT INTO SUPERMARKET VALUES (9);
INSERT INTO SUPERMARKET VALUES (10);
INSERT INTO SUPERMARKET VALUES (11);

-- PRODUCT
INSERT INTO PRODUCT (Name, Description) VALUES ("Crocs", "The most delightfully
comfortable shoes in the world");
INSERT INTO PRODUCT (Name, Description) VALUES ("Fresh Roma Tomato, Each", "With Roma
Tomatoes, it's easy to make a wholesome, delicious meal");
INSERT INTO PRODUCT (Name, Description) VALUES ("Ferrero Rocher Premium Gourmet Milk
Chocolate", "Ferrero Rocher offers a unique taste experience of contrasting layers: a
```

```
whole crunchy hazelnut in the heart, a delicious creamy hazelnut filling, a crisp wafer shell covered with chocolate and gently roasted pieces. And thanks to its inimitable golden wrapper Ferrero Rocher is even more unique and special. The iconic original.");
INSERT INTO PRODUCT (Name, Description) VALUES ("Great Value Whole Vitamin D Milk, Gallon, 128 fl oz", "Enjoy the wholesome goodness of Great Value Whole Milk at any time. This Grade A quality milk is pasteurized and delivers fresh from the farm taste. This kitchen staple is great for using in cereal, to bake desserts and more. This gallon of milk is also wonderful for milkshakes, smoothies, chocolate milk and more. Enjoy a delicious classic and serve a cold glass of this milk with soft, chocolate chip cookies for dipping. It offers an abundance of nutritional benefits as it is a rich source of protein, calcium, potassium and vitamin D. Our farms have pledged to not treat any cows with any artificial growth hormones.");
INSERT INTO PRODUCT (Name, Description) VALUES ("Yaheetech Convertible Sofa Bed", "Yaheetech Convertible Sofa Bed with USB Ports Sleeper Couch Futon Daybed Sleeper Sofa for Living Room");
INSERT INTO PRODUCT (Name, Description) VALUES ("Beats Studio3 Over-Ear Noise Canceling Bluetooth Wireless Headphones", "Beats Studio3 Wireless headphones deliver a premium listening experience with Pure Active Noise Cancelling (Pure ANC). Beats' Pure ANC actively blocks external noise and uses real-time audio calibration to preserve clarity, range and emotion. They continuously pinpoint external sounds to block while automatically responding to individual fit in real time, optimising sound output to preserve a premium listening experience the way artists intended.");
INSERT INTO PRODUCT (Name, Description) VALUES ("Lindor Holiday Sugar Cookie Bag - 6oz", "Remind a friend or loved one of how much they mean to you this holiday season with Lindt LINDOR Sugar Cookie White Chocolate Truffles. Featuring festive holiday packaging, these white chocolate candy truffles make perfect gifts and stocking stuffers for family and friends. Bring a bit of sparkle to your Christmas candy dishes and table decor with the addition of these LINDOR holiday chocolates. Each Lindt LINDOR chocolate truffle is finely crafted with a delicate white chocolate shell that envelops an irresistibly smooth sugar cookie truffle filling. Make Holiday Moments Magical with Lindt.");
INSERT INTO PRODUCT (Name, Description) VALUES ("MacBook Air", 'MacBook Air 13.3" - Apple M1 Chip 8-core CPU, 7-core GPU - 8GB Memory - 256GB SSD Space Gray');
INSERT INTO PRODUCT (Name, Description) VALUES ("Holiday Extravaganza Gift Basket", "Divine chocolates, decadent sweets and savory nibbles abound in this beautiful gift that a special someone deserves.  They'll love the sweets from Harry & David®,The Popcorn Factory, Ghirardelli ® and more.  A truly impressive gift!  After the snacks are gone, the basket makes a great daily decor or storage of numerous collections for years to come!");
INSERT INTO PRODUCT (Name, Description) VALUES ("Snapware Pyrex 18-piece Glass Food Storage Set", "This Snapware® set is made of Pyrex® tempered glass, which is safe to use in both the microwave and oven. The lids are airtight to keep food fresher longer, yet are easy to put on and take off. Plus, you can write on the lids to eliminate the guesswork of what's inside. Then, erase the lid and reuse.");
INSERT INTO PRODUCT (Name, Description) VALUES ("Microsoft Office Home and Student 2022", "For students and families who want classic Office 2022 versions of Word,
```

```
Excel, PowerPoint and OneNote for Windows 11, installed on one PC or Mac for use at
home or school, including 60 days of Microsoft support at no extra cost.");

-- SELLS
INSERT INTO SELLS VALUES (9, 1, 29.99, 60);
INSERT INTO SELLS VALUES (9, 2, 0.29, 1000);
INSERT INTO SELLS VALUES (9, 3, 10.48, 20);
INSERT INTO SELLS VALUES (9, 4, 2.98, 200);
INSERT INTO SELLS VALUES (9, 8, 749.99, 13);
INSERT INTO SELLS VALUES (10, 2, 0.33, 500);
INSERT INTO SELLS VALUES (10, 4, 3.29, 500);
INSERT INTO SELLS VALUES (10, 5, 325.99, 11);
INSERT INTO SELLS VALUES (10, 6, 169.99, 45);
INSERT INTO SELLS VALUES (10, 7, 5.29, 120);
INSERT INTO SELLS VALUES (10, 8, 799.99, 20);
INSERT INTO SELLS VALUES (11, 8, 799.99, 9);
INSERT INTO SELLS VALUES (11, 9, 159.99, 2);
INSERT INTO SELLS VALUES (11, 10, 30.99, 25);
INSERT INTO SELLS VALUES (11, 11, 119.19, 5);

-- SCHEDULE
INSERT INTO SCHEDULE VALUES (1, 'Sunday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Monday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Tuesday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Wednesday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Thursday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Friday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (1, 'Saturday', '00:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (2, 'Sunday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Monday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Tuesday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Wednesday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Thursday', '10:00:00', '01:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Friday', '10:00:00', '01:00:00');
INSERT INTO SCHEDULE VALUES (2, 'Saturday', '10:00:00', '01:00:00');
INSERT INTO SCHEDULE VALUES (3, 'Sunday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Monday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Tuesday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Wednesday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Thursday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Friday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (3, 'Saturday', '06:00:00', '22:45:00');
INSERT INTO SCHEDULE VALUES (4, 'Sunday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (4, 'Monday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (4, 'Tuesday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (4, 'Wednesday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (4, 'Thursday', '11:00:00', '02:00:00');
```

```
INSERT INTO SCHEDULE VALUES (4, 'Friday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (4, 'Saturday', '11:00:00', '02:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Sunday', '11:00:00', '21:30:00');
INSERT INTO SCHEDULE VALUES (5, 'Monday', '11:00:00', '21:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Tuesday', '11:00:00', '21:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Wednesday', '11:00:00', '21:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Thursday', '11:00:00', '21:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Friday', '11:00:00', '21:00:00');
INSERT INTO SCHEDULE VALUES (5, 'Saturday', '11:00:00', '21:30:00');
INSERT INTO SCHEDULE VALUES (6, 'Sunday', '11:30:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (6, 'Monday', '11:30:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (6, 'Tuesday', '11:30:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (6, 'Wednesday', '11:30:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (6, 'Thursday', '11:30:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (6, 'Friday', '11:30:00', '22:30:00');
INSERT INTO SCHEDULE VALUES (6, 'Saturday', '11:30:00', '22:30:00');
INSERT INTO SCHEDULE VALUES (7, 'Sunday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Monday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Tuesday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Wednesday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Thursday', '10:00:00', '00:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Friday', '10:00:00', '01:00:00');
INSERT INTO SCHEDULE VALUES (7, 'Saturday', '10:00:00', '01:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Sunday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Monday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Tuesday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Wednesday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Thursday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Friday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (8, 'Saturday', '10:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Sunday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Monday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Tuesday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Wednesday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Thursday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Friday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (9, 'Saturday', '06:00:00', '23:00:00');
INSERT INTO SCHEDULE VALUES (10, 'Sunday', '07:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (10, 'Monday', '07:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (10, 'Tuesday', '07:00:00', '23:59:59');
INSERT INTO SCHEDULE VALUES (10, 'Wednesday', '08:00:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (10, 'Thursday', '08:00:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (10, 'Friday', '08:00:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (10, 'Saturday', '08:00:00', '22:00:00');
INSERT INTO SCHEDULE VALUES (11, 'Sunday', '10:00:00', '18:00:00');
INSERT INTO SCHEDULE VALUES (11, 'Monday', '10:00:00', '20:30:00');
INSERT INTO SCHEDULE VALUES (11, 'Tuesday', '10:00:00', '20:30:00');
```

```
INSERT INTO SCHEDULE VALUES (11, 'Wednesday', '10:00:00', '20:30:00');
INSERT INTO SCHEDULE VALUES (11, 'Thursday', '10:00:00', '20:30:00');
INSERT INTO SCHEDULE VALUES (11, 'Friday', '10:00:00', '20:30:00');
INSERT INTO SCHEDULE VALUES (11, 'Saturday', '09:30:00', '19:00:00');

-- PROMOTION
INSERT INTO PROMOTION VALUES (1, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (1, '15%OFF', 'Get 15% off on spending 40$ and above',
'2022-12-01');
INSERT INTO PROMOTION VALUES (1, '2$OFF', 'Get 2$ off on spending 10$ and above',
'2022-04-01');
INSERT INTO PROMOTION VALUES (1, 'FREEDEL', 'Free Delivery', '2022-11-15');
INSERT INTO PROMOTION VALUES (2, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (3, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (4, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (5, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (6, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (7, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (7, 'BUY1GET1', 'Buy one get one free', '2022-04-01');
INSERT INTO PROMOTION VALUES (8, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (8, '10$BUCKET', 'Get any bucket for 10$', '2022-12-01');
INSERT INTO PROMOTION VALUES (8, 'BUY1GET1', 'Buy one bucket worth minimum of 18$ and
get another bucket for free', '2022-09-01');
INSERT INTO PROMOTION VALUES (9, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (9, 'FREEDEL', 'Free Delivery', '2022-01-01');
INSERT INTO PROMOTION VALUES (10, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (10, '20$OFF', 'Get 20$ off on spending 10$ and above',
'2022-11-20');
INSERT INTO PROMOTION VALUES (11, 'N/A', 'Not Applied', '2022-01-01');
INSERT INTO PROMOTION VALUES (11, '50%OFF', 'Get 50% off', '2022-11-20');

-- ORDER
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('6 x Hot and Spicy
McChicken', 11.63, '2022-01-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x 20 Piece McNuggets',
10.75, '2022-04-27');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x McChicken Biscuit &
1 x Hash Browns', 4.09, '2022-11-16');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Chipotle Ranch
Grilled Chicken Burrito', 2.17, '2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ("6 x Braum's - Nutty
Cone", 5.41, '2022-11-24');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x Golden Original
Whole Chicken', 17.32, '2022-11-19');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x Memphis BBQ
Chicken', 20.56, '2022-11-13');
```

```
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x 12 pc. Chicken
Bucket', 28.13, '2022-11-09');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 9.44, '2022-11-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Crocs', 32.46,
'2022-11-12');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x Ferrero Rocher
Premium Gourmet Milk Chocolate', 20.96, '2022-11-27');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Yaheetech
Convertible Sofa Bed', 332.88, '2022-11-22');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x MacBook Air',
432.99, '2022-11-24');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Holiday Extravaganza
Gift Basket & 1 x Microsoft Office Home and Student 2022', 302.21, '2022-11-25');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-21');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-22');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-23');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-24');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-25');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-26');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-27');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-28');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-29');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-11-30');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('12 x Fresh Roma Tomato,
Each & 2 x Great Value Whole Vitamin D Milk, Gallon, 128 fl oz', 11.44, '2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Chipotle Ranch
Grilled Chicken Burrito', 2.17, '2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ("6 x Braum's - Nutty
Cone", 5.41, '2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Mojito', 6.48,
'2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('1 x Meal', 12.99, '2022-
12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x Golden Original
Whole Chicken', 17.32, '2022-12-01');
```

```sql
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x Memphis BBQ
Chicken', 20.56, '2022-12-01');
INSERT INTO `ORDER` (Contents, Subtotals, Placed_on) VALUES ('2 x 12 pc. Chicken
Bucket', 28.13, '2022-12-01');

-- PAYMENT
INSERT INTO PAYMENT (Type, Time) VALUES ('Cash', '2022-01-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-04-27');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-16');
INSERT INTO PAYMENT (Type, Time) VALUES ('Cash', '2022-01-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Cash', '2022-11-24');
INSERT INTO PAYMENT (Type, Time) VALUES ('EFT', '2022-05-19');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-04-13');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-09-09');
INSERT INTO PAYMENT (Type, Time) VALUES ('Cash', '2022-11-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Check', '2022-11-12');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-27');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-22');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-24');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-25');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-21');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-22');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-23');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-24');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-25');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-26');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-27');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-28');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-29');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-11-30');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');
INSERT INTO PAYMENT (Type, Time) VALUES ('Credit/Debit Card', '2022-12-01');

-- DELIVERY_INFO
INSERT INTO DELIVERY_INFO VALUES ('E005', 'AAA0286', 3);
INSERT INTO DELIVERY_INFO VALUES ('E003', 'FGY5597', 10);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 11);
INSERT INTO DELIVERY_INFO VALUES ('E005', 'AAA0286', 15);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 16);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 17);
INSERT INTO DELIVERY_INFO VALUES ('E005', 'AAA0286', 18);
```

```sql
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 19);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 20);
INSERT INTO DELIVERY_INFO VALUES ('E005', 'AAA0286', 21);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 22);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 23);
INSERT INTO DELIVERY_INFO VALUES ('E005', 'AAA0286', 24);
INSERT INTO DELIVERY_INFO VALUES ('E003', '77KR117', 25);

-- ORDER_INFO
INSERT INTO ORDER_INFO VALUES (1, 1, 'C004', 1, 'N/A');
INSERT INTO ORDER_INFO VALUES (2, 1, 'C006', 2, '2$OFF');
INSERT INTO ORDER_INFO VALUES (3, 1, 'C001', 3, 'FREEDEL');
INSERT INTO ORDER_INFO VALUES (4, 2, 'C005', 4, 'N/A');
INSERT INTO ORDER_INFO VALUES (5, 3, 'C003', 5, 'N/A');
INSERT INTO ORDER_INFO VALUES (6, 6, 'C003', 6, 'N/A');
INSERT INTO ORDER_INFO VALUES (7, 7, 'C004', 7, 'BUY1GET1');
INSERT INTO ORDER_INFO VALUES (8, 8, 'C005', 8, 'BUY1GET1');
INSERT INTO ORDER_INFO VALUES (9, 9, 'C006', 9, 'N/A');
INSERT INTO ORDER_INFO VALUES (10, 9, 'C005', 10, 'FREEDEL');
INSERT INTO ORDER_INFO VALUES (11, 9, 'C005', 11, 'FREEDEL');
INSERT INTO ORDER_INFO VALUES (12, 10, 'C002', 12, '20$OFF');
INSERT INTO ORDER_INFO VALUES (13, 11, 'C001', 13, '50%OFF');
INSERT INTO ORDER_INFO VALUES (14, 11, 'C002', 14, 'N/A');
INSERT INTO ORDER_INFO VALUES (15, 9, 'C004', 15, 'N/A');
INSERT INTO ORDER_INFO VALUES (16, 9, 'C004', 16, 'N/A');
INSERT INTO ORDER_INFO VALUES (17, 9, 'C004', 17, 'N/A');
INSERT INTO ORDER_INFO VALUES (18, 9, 'C004', 18, 'N/A');
INSERT INTO ORDER_INFO VALUES (19, 9, 'C004', 19, 'N/A');
INSERT INTO ORDER_INFO VALUES (20, 9, 'C004', 20, 'N/A');
INSERT INTO ORDER_INFO VALUES (21, 9, 'C004', 21, 'N/A');
INSERT INTO ORDER_INFO VALUES (22, 9, 'C004', 22, 'N/A');
INSERT INTO ORDER_INFO VALUES (23, 9, 'C004', 23, 'N/A');
INSERT INTO ORDER_INFO VALUES (24, 9, 'C004', 24, 'N/A');
INSERT INTO ORDER_INFO VALUES (25, 9, 'C004', 25, 'N/A');
INSERT INTO ORDER_INFO VALUES (26, 2, 'C001', 26, 'N/A');
INSERT INTO ORDER_INFO VALUES (27, 3, 'C001', 27, 'N/A');
INSERT INTO ORDER_INFO VALUES (28, 4, 'C001', 28, 'N/A');
INSERT INTO ORDER_INFO VALUES (29, 5, 'C001', 29, 'N/A');
INSERT INTO ORDER_INFO VALUES (30, 6, 'C001', 30, 'N/A');
INSERT INTO ORDER_INFO VALUES (31, 7, 'C001', 31, 'N/A');
INSERT INTO ORDER_INFO VALUES (32, 8, 'C001', 32, 'N/A');


-- CONTRACT
INSERT INTO CONTRACT VALUES (1, 'E001', '2022-01-01');
INSERT INTO CONTRACT VALUES (2, 'E006', '2022-01-01');
INSERT INTO CONTRACT VALUES (3, 'E007', '2022-01-01');
```

```sql
INSERT INTO CONTRACT VALUES (4, 'E007', '2022-01-01');
INSERT INTO CONTRACT VALUES (5, 'E001', '2022-01-01');
INSERT INTO CONTRACT VALUES (6, 'E001', '2022-01-01');
INSERT INTO CONTRACT VALUES (7, 'E001', '2022-01-01');
INSERT INTO CONTRACT VALUES (8, 'E001', '2022-01-01');
INSERT INTO CONTRACT VALUES (9, 'E008', '2022-01-01');
INSERT INTO CONTRACT VALUES (10, 'E009', '2022-01-01');
INSERT INTO CONTRACT VALUES (11, 'E001', '2022-01-01');

-- COMMENTS
INSERT INTO COMMENTS VALUES (1, 'C001', '1', 'Customer service is not good');
INSERT INTO COMMENTS VALUES (2, 'C005', '5', 'Amazing Food');
INSERT INTO COMMENTS VALUES (11, 'C001', '5', 'Has great deals');
```

6.3. Views

1. Annual Top Customers: This view returns the First Name, Last Name, Total Order Subtotals of the customers who paid top 3 total subtotals of all orders in past 1 year.

```
DROP VIEW IF EXISTS ANNUAL_TOP_CUSTOMERS;
CREATE VIEW ANNUAL_TOP_CUSTOMERS
AS
SELECT First_name, Last_name, SUM(Subtotals) AS Total_order_subtotals
FROM PERSON
NATURAL JOIN CUSTOMER
NATURAL JOIN ORDER_INFO
NATURAL JOIN `ORDER`
WHERE Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 YEAR)
GROUP BY Customer_id
ORDER BY Total_order_subtotals DESC
LIMIT 3;
```

2. Popular Restaurant Type: This view returns the Type of restaurants that have the most number of orders in past 1 year.

```
DROP VIEW IF EXISTS NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_TYPE_IN_PAST_ONE_YEAR;
CREATE VIEW NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_TYPE_IN_PAST_ONE_YEAR
AS
SELECT Type, COUNT(*) AS Total_orders
FROM RESTAURANT
NATURAL JOIN RESTAURANT_TYPE
INNER JOIN SHOP
ON RESTAURANT.Restaurant_id = SHOP.Shop_id
NATURAL JOIN ORDER_INFO
NATURAL JOIN `ORDER`
WHERE Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 YEAR)
GROUP BY Type;

DROP VIEW IF EXISTS POPULAR_RESTAURANT_TYPE;
CREATE VIEW POPULAR_RESTAURANT_TYPE
AS
SELECT Type, Total_orders
FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_TYPE_IN_PAST_ONE_YEAR
WHERE Total_orders = (
    SELECT MAX(Total_orders)
    FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_TYPE_IN_PAST_ONE_YEAR
);
```

3. Potential Silver Member: This view returns the information of the customers (not a silver member yet) who have placed orders more than 10 times in the past 1 month.

```
DROP VIEW IF EXISTS POTENTIAL_SILVER_MEMBER;
CREATE VIEW POTENTIAL_SILVER_MEMBER
AS
SELECT First_name, Middle_name, Last_name, COUNT(*) AS Number_of_orders
FROM PERSON
NATURAL JOIN CUSTOMER
NATURAL JOIN ORDER_INFO
NATURAL JOIN `ORDER`
WHERE Customer_id NOT IN (
    SELECT Silver_member_id
    FROM SILVER_MEMBER
) AND Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 MONTH)
GROUP BY Person_id
HAVING Number_of_orders > 10
ORDER BY Number_of_orders DESC;
```

4. Best Area Manager: This view returns the information of the area manager who successfully made the most number of contracts with shops in her/his working area in past 1 year.

```
DROP VIEW IF EXISTS NUMBER_OF_CONTRACTS_BY_AREA_MANAGERS_IN_PAST_ONE_YEAR;
CREATE VIEW NUMBER_OF_CONTRACTS_BY_AREA_MANAGERS_IN_PAST_ONE_YEAR
AS
SELECT First_name, Middle_name, Last_name, Area, Total_contracts
FROM PERSON
NATURAL JOIN EMPLOYEE
INNER JOIN (
    SELECT Area_manager_id, Area, COUNT(*) AS Total_contracts
    FROM AREA_MANAGER
    NATURAL JOIN CONTRACT
    WHERE Start_time >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 YEAR)
    GROUP BY Area_manager_id, Area
) T
ON T.Area_manager_id = EMPLOYEE.Employee_id;

CREATE VIEW BEST_AREA_MANAGER
AS
SELECT First_name, Middle_name, Last_name, Area, Total_contracts
FROM NUMBER_OF_CONTRACTS_BY_AREA_MANAGERS_IN_PAST_ONE_YEAR T1
WHERE Total_contracts = (
    SELECT MAX(Total_contracts)
    FROM NUMBER_OF_CONTRACTS_BY_AREA_MANAGERS_IN_PAST_ONE_YEAR T2
    WHERE T1.Area = T2.Area
```

```
)
ORDER BY Area;
```

5. Top Restaurants: This view returns the top restaurant in each restaurant type that have the most orders in past 1 month.

```
DROP VIEW IF EXISTS NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH;
CREATE VIEW NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH
AS
SELECT Name, Type, COUNT(*) AS Number_of_orders
FROM SHOP
INNER JOIN RESTAURANT
ON SHOP.Shop_id = RESTAURANT.Restaurant_id
NATURAL JOIN RESTAURANT_TYPE
NATURAL JOIN `ORDER`
NATURAL JOIN ORDER_INFO
WHERE Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 MONTH)
GROUP BY Name, Type;

DROP VIEW IF EXISTS TOP_RESTAURANTS;
CREATE VIEW TOP_RESTAURANTS
AS
SELECT Name, Type, Number_of_orders
FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH T1
WHERE Number_of_orders = (
    SELECT MAX(Number_of_orders)
    FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH T2
    WHERE T1.Type = T2.Type
)
ORDER BY Type;
```

6.4. Queries

1. Find the names of employee who supervises the most number of deliverers.

```
DROP VIEW IF EXISTS NUMBER_OF_DELIVERERS_SUPERVISED;
CREATE VIEW NUMBER_OF_DELIVERERS_SUPERVISED
AS
SELECT Supervisor_id, COUNT(*) AS Number_of_deliverers_supervised
FROM DELIVERER
GROUP BY Supervisor_id;

SELECT First_name, Middle_name, Last_name, Number_of_deliverers_supervised
FROM PERSON
NATURAL JOIN EMPLOYEE
INNER JOIN (
    SELECT *
    FROM NUMBER_OF_DELIVERERS_SUPERVISED
    WHERE Number_of_deliverers_supervised = (
        SELECT MAX(Number_of_deliverers_supervised)
        FROM NUMBER_OF_DELIVERERS_SUPERVISED
    )
) T
ON EMPLOYEE.Employee_id = T.Supervisor_id;
```

First_name	Middle_name	Last_name	Number_of_deliverers_supervised
John	B	Smith	2

2. Find the average number of orders placed by Potential Silver Member.

```
SELECT AVG(Number_of_orders) AS Average_number_of_orders
FROM POTENTIAL_SILVER_MEMBER;
```

Average_number_of_orders
12

3. Find all the customers who placed orders of the restaurants that belong to Popular Restaurant Type. Please also report the name of restaurants.

```
SELECT PERSON.First_name, PERSON.Middle_name, PERSON.Last_name, Name
FROM POPULAR_RESTAURANT_TYPE
NATURAL JOIN RESTAURANT_TYPE
NATURAL JOIN RESTAURANT
INNER JOIN SHOP
```

```
ON RESTAURANT.Restaurant_id = SHOP.Shop_id
NATURAL JOIN ORDER_INFO
NATURAL JOIN CUSTOMER
NATURAL JOIN PERSON;
```

First_name	Middle_name	Last_name	Name
James	E	Borg	McDonald's
Franklin	T	Wong	McDonald's
Joyce	A	English	McDonald's
Ahmad	V	Jabbar	Taco Bell
Franklin	T	Wong	Taco Bell
Jennifer	S	Wallace	Braums
Franklin	T	Wong	Braums
Joyce	A	English	Domino's
Franklin	T	Wong	Domino's
Ahmad	V	Jabbar	KFC
Franklin	T	Wong	KFC

4. List all the customers that have become a silver member within a month of joining the system.

```
SELECT First_name, Middle_name, Last_name
FROM PERSON
NATURAL JOIN CUSTOMER
INNER JOIN SILVER_MEMBER
ON CUSTOMER.Customer_id = SILVER_MEMBER.Silver_member_id
NATURAL JOIN MEMBER_CARD
WHERE Joining_date >= DATE_SUB(Issue_date, INTERVAL 1 MONTH);
```

First_name	Middle_name	Last_name
Alicia	J	Zelaya

5. Find the names of deliverers who delivered the most orders in past 1 month.

```
DROP VIEW IF EXISTS NUMBER_OF_DELIVERIES_MADE_BY_DELIVERERS_IN_PAST_ONE_MONTH;
CREATE VIEW NUMBER_OF_DELIVERIES_MADE_BY_DELIVERERS_IN_PAST_ONE_MONTH
AS
SELECT Deliverer_id, COUNT(*) AS Number_of_deliveries
FROM DELIVERY_INFO
NATURAL JOIN ORDER_INFO
NATURAL JOIN `ORDER`
WHERE Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 MONTH)
```

```
GROUP BY Deliverer_id;

SELECT First_name, Middle_name, Last_name, Number_of_deliveries
FROM PERSON
NATURAL JOIN EMPLOYEE
INNER JOIN (
    SELECT *
    FROM NUMBER_OF_DELIVERIES_MADE_BY_DELIVERERS_IN_PAST_ONE_MONTH
    WHERE Number_of_deliveries = (
        SELECT MAX(Number_of_deliveries)
        FROM NUMBER_OF_DELIVERIES_MADE_BY_DELIVERERS_IN_PAST_ONE_MONTH
    )
) T
ON EMPLOYEE.Employee_id = T.Deliverer_id;
```

First_name	Middle_name	Last_name	Number_of_deliveries
Alicia	J	Zelaya	9

6. Find the restaurants that provide most promotion in past 1 month.

```
DROP VIEW IF EXISTS
NUMBER_OF_PROMOTIONS_PROVIDED_BY_RESTAURANTS_IN_PAST_ONE_MONTH;
CREATE VIEW NUMBER_OF_PROMOTIONS_PROVIDED_BY_RESTAURANTS_IN_PAST_ONE_MONTH
AS
SELECT Shop_id, COUNT(*) AS Number_of_promotions
FROM PROMOTION
INNER JOIN RESTAURANT
ON PROMOTION.Shop_id = RESTAURANT.Restaurant_id
WHERE Start_date >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 MONTH)
GROUP BY Shop_id;

SELECT Name, Number_of_promotions
FROM SHOP
NATURAL JOIN (
    SELECT *
    FROM NUMBER_OF_PROMOTIONS_PROVIDED_BY_RESTAURANTS_IN_PAST_ONE_MONTH
    WHERE Number_of_promotions = (
        SELECT MAX(Number_of_promotions)
        FROM NUMBER_OF_PROMOTIONS_PROVIDED_BY_RESTAURANTS_IN_PAST_ONE_MONTH
    )
) T;
```

Name	Number_of_promotions
McDonald's	2

7. Find the customer who have place orders of all Fast Food restaurants.

```
SELECT First_name, Middle_name, Last_name
FROM PERSON
NATURAL JOIN CUSTOMER
NATURAL JOIN (
    SELECT Customer_id
    FROM ORDER_INFO
    INNER JOIN RESTAURANT
    ON ORDER_INFO.Shop_id = RESTAURANT.Restaurant_id
    NATURAL JOIN RESTAURANT_TYPE
    WHERE Type = "Fast Food"
    GROUP BY Customer_id
    HAVING COUNT(DISTINCT(Restaurant_id)) = (
        SELECT COUNT(*)
        FROM RESTAURANT_TYPE
        WHERE Type = "Fast Food"
    )
) T;
-- alternate query (using division)
SELECT DISTINCT First_name, Middle_name, Last_name
FROM ORDER_INFO O1
NATURAL JOIN CUSTOMER
NATURAL JOIN PERSON
WHERE NOT EXISTS (
    (SELECT Restaurant_id
    FROM RESTAURANT_TYPE
    WHERE Type = "Fast Food")
    EXCEPT
    (SELECT Shop_id
    FROM ORDER_INFO O2
    WHERE O1.Customer_id = O2.Customer_id)
);
```

First_name	Middle_name	Last_name
Franklin	T	Wong

8. For each restaurant, list all the customers who placed the order there, and the price of each order.

```
SELECT Name, First_name, Middle_name, Last_name, Subtotals
FROM RESTAURANT
INNER JOIN SHOP
ON RESTAURANT.Restaurant_id = SHOP.Shop_id
NATURAL JOIN ORDER_INFO
NATURAL JOIN `ORDER`
NATURAL JOIN CUSTOMER
NATURAL JOIN PERSON
ORDER BY Name;
```

Name	First_name	Middle_name	Last_name	Subtotals
BBQ Chicken	Jennifer	S	Wallace	17.32
BBQ Chicken	Franklin	T	Wong	17.32
Braums	Jennifer	S	Wallace	5.41
Braums	Franklin	T	Wong	5.41
Domino's	Joyce	A	English	20.56
Domino's	Franklin	T	Wong	20.56
Great Wall Super Buffet	Franklin	T	Wong	12.99
KFC	Ahmad	V	Jabbar	28.13
KFC	Franklin	T	Wong	28.13
McDonald's	James	E	Borg	10.75
McDonald's	Franklin	T	Wong	4.09
McDonald's	Joyce	A	English	11.63
Taco Bell	Ahmad	V	Jabbar	2.17
Taco Bell	Franklin	T	Wong	2.17
The Irishman Pub	Franklin	T	Wong	6.48

9. Find the area that have the most number of restaurants located.

```
DROP VIEW IF EXISTS NUMBER_OF_RESTAURANTS_IN_EACH_AREA;
CREATE VIEW NUMBER_OF_RESTAURANTS_IN_EACH_AREA
AS
SELECT Area, COUNT(*) AS Number_of_restaurants
FROM RESTAURANT
INNER JOIN SHOP
ON RESTAURANT.Restaurant_id = SHOP.Shop_id
GROUP BY Area;

SELECT Area, Number_of_restaurants
FROM NUMBER_OF_RESTAURANTS_IN_EACH_AREA
```

```
WHERE Number_of_restaurants = (
    SELECT MAX(Number_of_restaurants)
    FROM NUMBER_OF_RESTAURANTS_IN_EACH_AREA
);
```

Area	Number_of_restaurants
Plano	5

10. Find the schedule of the restaurant that have the most orders in past 1 month.

```
DROP VIEW IF EXISTS NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH;
CREATE VIEW NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH
AS
SELECT Shop_id, COUNT(*) AS Number_of_orders
FROM RESTAURANT
INNER JOIN ORDER_INFO
ON RESTAURANT.Restaurant_id = ORDER_INFO.Shop_id
NATURAL JOIN `ORDER`
WHERE Placed_on >= DATE_SUB(CURRENT_TIMESTAMP, INTERVAL 1 MONTH)
GROUP BY Shop_id;

SELECT Name, Day, Opening_time, Closing_time
FROM SCHEDULE
NATURAL JOIN SHOP
NATURAL JOIN (
    SELECT *
    FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH
    WHERE Number_of_orders = (
        SELECT MAX(Number_of_orders)
        FROM NUMBER_OF_ORDERS_IN_EACH_RESTAURANT_IN_PAST_ONE_MONTH
    )
) T;
```

Name	Day	Opening_time	Closing_time
Taco Bell	Sunday	10:00:00	0:00:00
Taco Bell	Monday	10:00:00	0:00:00
Taco Bell	Tuesday	10:00:00	0:00:00
Taco Bell	Wednesday	10:00:00	0:00:00
Taco Bell	Thursday	10:00:00	1:00:00
Taco Bell	Friday	10:00:00	1:00:00
Taco Bell	Saturday	10:00:00	1:00:00
Braums	Sunday	6:00:00	22:45:00
Braums	Monday	6:00:00	22:45:00
Braums	Tuesday	6:00:00	22:45:00
Braums	Wednesday	6:00:00	22:45:00
Braums	Thursday	6:00:00	22:45:00
Braums	Friday	6:00:00	22:45:00
Braums	Saturday	6:00:00	22:45:00
BBQ Chicken	Sunday	11:30:00	22:00:00
BBQ Chicken	Monday	11:30:00	22:00:00
BBQ Chicken	Tuesday	11:30:00	22:00:00
BBQ Chicken	Wednesday	11:30:00	22:00:00
BBQ Chicken	Thursday	11:30:00	22:00:00
BBQ Chicken	Friday	11:30:00	22:30:00
BBQ Chicken	Saturday	11:30:00	22:30:00
Domino's	Sunday	10:00:00	0:00:00
Domino's	Monday	10:00:00	0:00:00
Domino's	Tuesday	10:00:00	0:00:00
Domino's	Wednesday	10:00:00	0:00:00
Domino's	Thursday	10:00:00	0:00:00
Domino's	Friday	10:00:00	1:00:00
Domino's	Saturday	10:00:00	1:00:00
KFC	Sunday	10:00:00	23:00:00
KFC	Monday	10:00:00	23:00:00
KFC	Tuesday	10:00:00	23:00:00
KFC	Wednesday	10:00:00	23:00:00
KFC	Thursday	10:00:00	23:00:00
KFC	Friday	10:00:00	23:00:00
KFC	Saturday	10:00:00	23:00:00

11. Find the names of employee who are also a Premium Member.

```
SELECT First_name, Middle_name, Last_name
FROM PERSON
NATURAL JOIN EMPLOYEE
WHERE Premium_member_id IS NOT NULL;
```

First_name	Middle_name	Last_name
Franklin	T	Wong
Alicia	J	Zelaya
John	B	Smith

12. Find the supermarket that have most different products in stock.

```
DROP VIEW IF EXISTS
NUMBER_OF_DIFFERENT_PRODUCTS_IN_STOCK_FOR_EACH_SUPERMARKETS;
CREATE VIEW NUMBER_OF_DIFFERENT_PRODUCTS_IN_STOCK_FOR_EACH_SUPERMARKETS
AS
SELECT Supermarket_id, COUNT(*) AS Number_of_different_products_in_stock
FROM SUPERMARKET
NATURAL JOIN SELLS
NATURAL JOIN PRODUCT
WHERE Stock > 0
GROUP BY Supermarket_id;

SELECT Name, Number_of_different_products_in_stock
FROM SHOP
INNER JOIN (
    SELECT *
    FROM NUMBER_OF_DIFFERENT_PRODUCTS_IN_STOCK_FOR_EACH_SUPERMARKETS
    WHERE Number_of_different_products_in_stock = (
        SELECT MAX(Number_of_different_products_in_stock)
        FROM NUMBER_OF_DIFFERENT_PRODUCTS_IN_STOCK_FOR_EACH_SUPERMARKETS
    )
) T
ON SHOP.Shop_id = T.Supermarket_id;
```

Name	Number_of_different_products_in_stock
Target	6

13. For each product, list all the supermarket selling it, and the price of the product at the supermarket.

```
SELECT PRODUCT.Name AS Product_name, SHOP.Name AS Supermarket_name, Price
FROM SELLS
NATURAL JOIN PRODUCT
NATURAL JOIN SUPERMARKET
INNER JOIN SHOP
ON SUPERMARKET.Supermarket_id = SHOP.Shop_id
ORDER BY Product_name;
```

Product_name	Supermarket_name	Price
Beats Studio3 Over-Ear Noise Canceling Bluetooth Wireless Headphones	Target	169.99
Crocs	Walmart	29.99
Ferrero Rocher Premium Gourmet Milk Chocolate	Walmart	10.48
Fresh Roma Tomato, Each	Walmart	0.29
Fresh Roma Tomato, Each	Target	0.33
Great Value Whole Vitamin D Milk, Gallon, 128 fl oz	Walmart	2.98
Great Value Whole Vitamin D Milk, Gallon, 128 fl oz	Target	3.29
Holiday Extravaganza Gift Basket	Costco Wholesale	159.99
Lindor Holiday Sugar Cookie Bag - 6oz	Target	5.29
MacBook Air	Walmart	749.99
MacBook Air	Target	799.99
MacBook Air	Costco Wholesale	799.99
Microsoft Office Home and Student 2022	Costco Wholesale	119.19
Snapware Pyrex 18-piece Glass Food Storage Set	Costco Wholesale	30.99
Yaheetech Convertible Sofa Bed	Target	325.99