CPSC 304 Project Cover Page

Milestone #: 2

Date: July 18

Group Number: 34

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jack Yao	29843927	s9n0s	jackyao2004@gmail.com
Joshua Calalang	66829102	u6x4h	joshuacalalang@hotmail.com
Rahul Kamath	25650656	t9i7n	rahul.kamath2512@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects so this will help them remember details about your project.

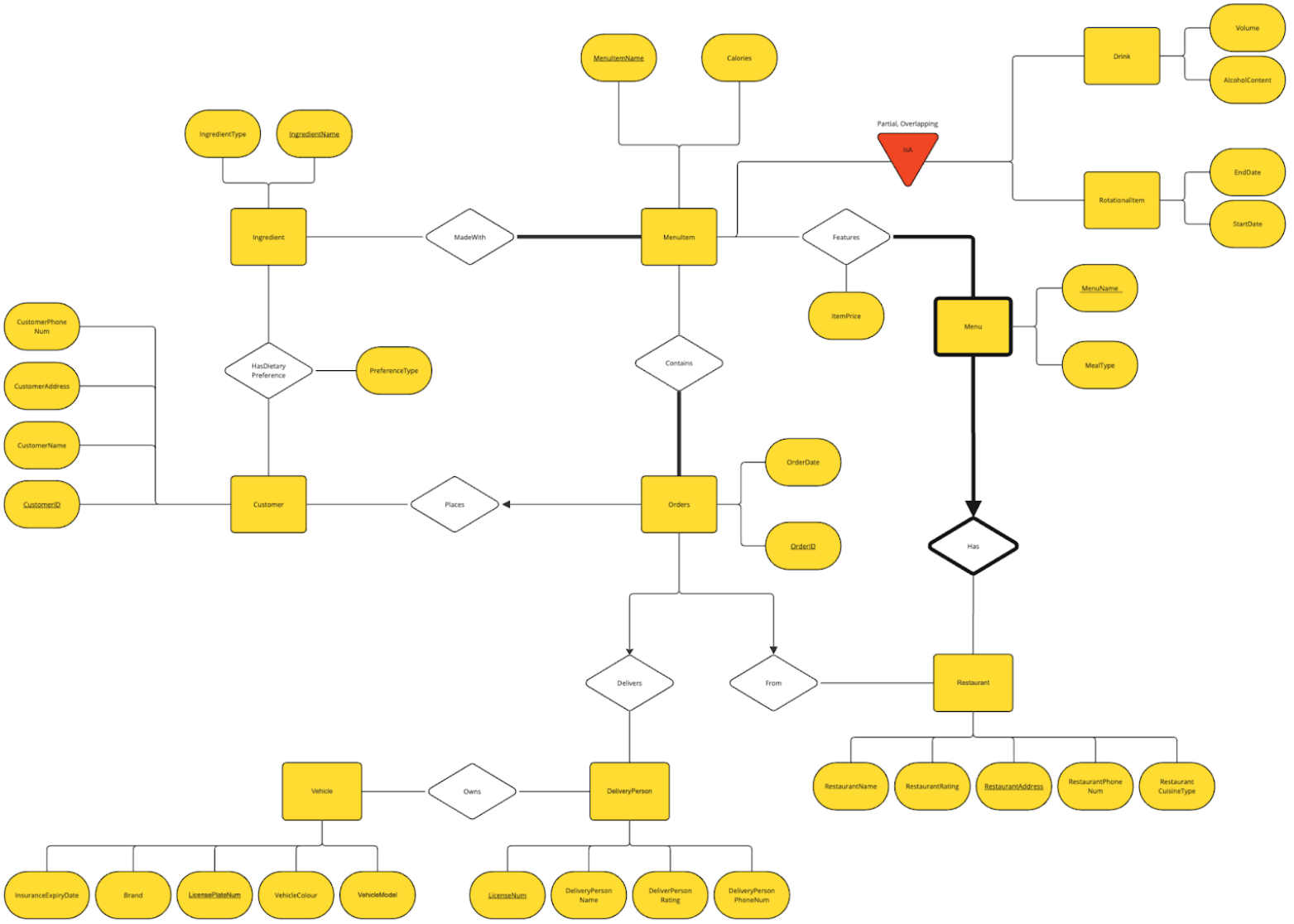
Our database application relates to the food and online delivery industry, specifically the management between customers, delivery drivers, and restaurants. This application intends to manage data of customers (e.g. food preferences/allergies and residing location), delivery drivers (e.g. working time and locations), and restaurants (e.g. food items and its ingredients). The emphasis is not on the retail/sales of food, but rather organizing the collection of data from the “end users” (i.e. the three mentioned above) of the industry.

3. The ER diagram you are basing your item #3 (below) on. This ER diagram may be the same as your milestone 1 submission or it might be different. If you have made changes from the version submitted in milestone 1, attach a note indicating what changes have been made and why. If you have decided not to implement the suggestions given by your project mentor, please be sure to leave a note stating why. This is not to say that you must do everything that your project mentor says. In many instances, there are trade-offs between design choices and your decision may be influenced by different factors. Your TAs will often leave suggestions that are meant to help massage your project into a form that will fit with the requirements in future project milestones. If you choose not to take their advice, it would be helpful for them to know why to better assist the group moving forward.

Since milestone #1, some changes have been made to the ER diagram:

- The Ingredient entity has another attribute IngredientType, which would show whether the ingredient is meat, vegetable, etc.
- The primary key of the Restaurant entity was changed to RestaurantAddress (assuming the address is unique for the regions that the application serves) and RestaurantID was removed. Customer, on the other hand, remains unchanged, as we argue that neither phone number, address, nor name are unique to a customer. Likewise with DeliveryPerson.
- New RestaurantRating attribute for Restaurant.
- New RestaurantCuisineType attribute for Restaurant.
- The IsA of the MenuItem entity has been identified to be partial and overlapping: a MenuItem does not necessarily have to be a Drink or a RotationalItem. Furthermore, a MenuItem can be both or only one of Drink or RotationalItem.
- Any attribute with # has been replaced with Num as to not interfere with SQL syntax.
- Order has been renamed to Orders to avoid the order keyword in SQL.
- Some renaming of attributes for clarity in the following sections below.

- Another note regarding the PhoneNum design decisions: we decided that PhoneNum are not unique for customers, in the case that a home/household number is used. The uniqueness of PhoneNum should, however, apply to restaurants and drivers.


MenuItemName
Calories
Drink
Volume
AlcoholContent
IngredientType
IngredientName
Partial, Overlapping
IsA
RotationalItem
EndDate
StartDate
Ingredient
MadeWith
MenuItem
Features
ItemPrice
Menu
MenuName
MealType
CustomerPhone Num
CustomerAddress
CustomerName
CustomerID
HasDietary Preference
PreferenceType
Contains
Customer
Places
Orders
OrderDate
OrderID
Has
Delivers
From
Restaurant
Vehicle
Owns
DeliveryPerson
RestaurantName
RestaurantRating
RestaurantAddress
RestaurantPhone Num
Restaurant CuisineType
InsuranceExpiryDate
Brand
LicensePlateNum
VehicleColour
VehicleModel
LicenseNum
DeliveryPerson Name
DeliverPerson Rating
DeliveryPerson PhoneNum

4. The schema derived from your ER diagram (above). For the translation of the ER diagram to the relational model, follow the same instructions as in your lectures. The process should be reasonably straightforward. For each table:

a. List the table definition (e.g., Table1(attr1: domain1, attr2: domain2, ...)). Make sure to include the domains for each attribute.

b. Specify the primary key (PK), candidate key, (CK) foreign keys (FK), and other constraints (e.g., not null, unique, etc.) that the table must maintain.

Ingredient(IngredientName, IngredientType)
IngredientName: varchar, PK
IngredientType: varchar

Customer(CustomerID, CustomerName, CustomerAddress, CustomerPhoneNum)
CustomerID: varchar, PK
CustomerName: varchar
CustomerAddress: varchar, not null
CustomerPhoneNum: char[10]

HasDietaryPreference(**CustomerID**, **IngredientName**, PreferenceType)
CustomerID: varchar, PK FK
IngredientName: varchar, PK FK
PreferenceType: varchar

Orders(OrderID, OrderDate, **CustomerID**, **LicenseNum**, **RestaurantAddress**)
OrderID: varchar, PK
OrderDate: date
CustomerID: varchar, FK
LicenseNum: varchar, FK
RestaurantAddress: varchar, FK

DeliveryPerson(LicenseNum, DeliveryPersonName, DeliveryPersonRating, PhoneNum)
LicenseNum: char[8], PK
DeliveryPersonName: varchar
DeliveryPersonRating: float
PhoneNum: char[10], CK unique

Vehicle(LicensePlateNum, InsuranceExpiryDate, Brand, VehicleModel, VehicleColour)
LicensePlateNum: varchar, PK
InsuranceExpiryDate: date
Brand: varchar
VehicleModel: varchar
VehicleColour: varchar

OwnsVehicle(**LicenseNum**, **LicensePlateNum**)
LicenseNum: varchar, PK FK
LicensePlateNum: varchar, PK FK

Restaurant(RestaurantAddress, RestaurantName, RestaurantRating, RestaurantPhoneNum, RestaurantCuisineType)

RestaurantAddress: varchar, PK
RestaurantName: varchar
RestaurantRating: float
RestaurantPhoneNum: char[10], CK unique
RestaurantCuisineType: varchar

Menu(MenuName, **RestaurantAddress**, MealType)
MenuName: varchar, PK
RestaurantAddress: varchar, PK FK
MealType: varchar

MenuItem(MenuItemName, Calories)
MenuItemName: varchar, PK
Calories: int

MenuFeaturesItem(**MenuItemName**, **MenuName**, ItemPrice)
MenuItemName: varchar, PK FK
MenuName: varchar, PK FK
ItemPrice: float, not null

ItemMadeWith(**MenuItemName**, **IngredientName**)
MenuItemName: varchar, PK FK
IngredientName: varchar, PK FK

OrderContains(**OrderID**, **MenuItemName**)
OrderID: varchar, PK FK
MenuItemName: varchar, PK FK

Drink(**MenuItemName**, Volume, AlcoholContent)
MenuItemName: varchar, PK FK
Volume: int
AlcoholContent: float

RotationalItem(**MenuItemName**, StartDate, EndDate)
MenuItemName: varchar, PK FK
StartDate: date, not null
EndDate: date, not null

RotationalDrink(**MenuItemName**, Volume, AlcoholContent, StartDate, EndDate)
MenuItemName: varchar, PK FK
StartDate: date, not null
EndDate: date, not null
Volume: int
AlcoholContent: float

5. Functional Dependencies (FDs)

a. Identify the functional dependencies in your relations, including the ones involving all candidate keys (including the primary key). PKs and CKs are considered functional dependencies and should be included in the list of FDs. You do not need to include trivial FDs such as A → A.

Note: In your list of FDs, there must be some kind of valid FD other than those identified by a PK or CK. If you observe that no relations have FDs other than the PK and CK(s), then you will have to intentionally add some (meaningful) attributes to show valid FDs. We want you to get a good normalization exercise. Your design must go through a normalization process. You do not need to have a non-PK/CK FD for each relation but be reasonable. If your TA feels that some non-PK/CK FDs have been omitted, your grade will be adjusted accordingly.

The FDs in our relations are:

Ingredient

IngredientName → IngredientType

Customer

CustomerID → CustomerName CustomerAddress CustomerPhoneNum

HasDietaryPreference

CustomerID IngredientName → PreferenceType

Orders

OrderID → OrderDate CustomerID LicenseNum RestaurantAddress

DeliveryPerson

LicenseNum → DeliveryPersonName DeliveryPersonRating DeliveryPersonPhoneNum
PhoneNum → DeliveryPersonName, DeliveryPersonRating, LicenseNum

Vehicle

LicensePlateNum → InsuranceExpiryDate, Brand, VehicleModel, VehicleColour

Restaurant

RestaurantAddress → RestaurantName, RestaurantRating, RestaurantPhoneNum, RestaurantCuisineType
RestaurantPhoneNum → RestaurantRating, RestaurantAddress, RestaurantCuisineType
RestaurantName → RestaurantCuisineType

Menu

MenuName, RestaurantAddress → MealType

MenuItem

MenuItemName → Calories

MenuFeaturesItem

MenuItemName, MenuName → ItemPrice

Drink

MenuItemName → Volume, AlcoholContent

RotationalItem

MenuItemName → StartDate, EndDate

RotationalDrink

MenuItemName → StartDate, EndDate, Volume, AlcoholContent

6. Normalization

a. Normalize each of your tables to be in 3NF or BCNF. Give the list of tables, their primary keys, their candidate keys, and their foreign keys after normalization. You should show the steps taken for the decomposition. Should there be errors, and no work is shown, no partial credit can be awarded without steps shown. The format should be the same as Step 3, with tables listed similar to Table1(attr1:domain1, attr2:domain2, ...). ALL Tables must be listed, not only the ones post normalization.

For simplicity, the attributes will be acronymized according to their CamelCase formatting (crossed out meaning it has been converted below):

Ingredient
IN → IT

Customer
~~CID → CN CA CPN~~

HasDietaryPreference
CID IN → PT

Orders
~~OID → OD CID LN RA~~

DeliveryPerson
~~LN → DPN DPR DPPN~~
~~DPPN → DPN DPR LN~~

Vehicle
~~LPN → IED B VM VC~~

Restaurant
~~RA → RN RR RPN RCT~~
~~RPN → RR RA RCT~~

RN → RCT

Menu
MN RA → MT

MenuItem
MIN → C

MenuFeaturesItem
MIN MN → IP

Drink
~~MIN → V AC~~

RotationalItem
~~MIN → SD ED~~

RotationalDrink
~~MIN → SD ED V AC~~

Converting to minimal cover

One attribute on RHS
CID → CN
CID → CA
CID → CPN

OID → OD
OID → CID
OID → LN
OID → RA

LN → DPN
LN → DPR
LN → DPPN

~~DPPN → DPN~~
~~DPPN → DPR~~
DPPN → LN

LPN → IED
LPN → B
LPN → VM
LPN → VC

RA → RN
RA → RR
RA → RPN
RA → RCT

~~RPN → RN~~
~~RPN → RR~~
RPN → RA

~~RPN → RCT~~

MIN → V
MIN → AC

MIN → SD
MIN → ED

Minimize LHS
CID IN → PT
CID+ = CID CN CA CPN
IN+ = IN IT
Thus can not minimize

RN RPN → RR
RN+ = RN RCT
RPN+ = RPN RA RR RCT
Thus can not minimize

RN RPN → RA
RN+ = RN RCT
RPN+ = RPN RA RR RCT
Thus can not minimize

MN RA → MT
MN+ = MN
RA+ = RA RR RPN RCT
Thus can not minimize

MIN MN → IP
MIN+ = MIN C V AC SD ED
MN+ = MN
Thus can not minimize

Remove Redundant FDs

DPPN → DPN and DPPN → DPR are redundant
DPPN+ = DPPN LN DPN DPR

RPN → RN, RPN → RR, and RPN → RCT are redundant
RPN+ = RA RPN RN RR RCT

Finally, the minimal cover are as follows:
IN → IT

CID IN → PT

RN → RCT

MN RA → MT

MIN → C

MIN MN → IP

CID → CN
CID → CA
CID → CPN

OID → OD
OID → CID
OID → LN
OID → RA

LN → DPN
LN → DPR
LN → DPPN

DPPN → LN

LPN → IED
LPN → B
LPN → VM
LPN → VC

RA → RN
RA → RR
RA → RPN
RA → RCT
RPN → RA
RN → RCT

MIN → V
MIN → AC

MIN → S
MIN → ED

Normalization to 3NF

We have decided to normalize to 3NF. With our FDs converted to minimal cover, we can see that only the table Restaurant violates 3NF, since only this relation has an FD (i.e RN → RCT) which violates the 3NF criteria (RN is not part of a superkey, and RCT is not part of a key in the FD RN → RCT).

Splitting the relation with respect to this FD we get the following:

Restaurant1(RestaurantAddress, **RestaurantName**, RestaurantRating, RestaurantPhoneNum)
- RestaurantAddress: varchar, PK
- RestaurantName: varchar, FK
- RestaurantRating: float
- RestaurantPhoneNum: char[10], CK unique

Restaurant2(RestaurantName, RestaurantCuisineType)
- RestaurantName: varchar, PK
- RestaurantCuisineType: varchar

Both these relations satisfy 3NF, so our final list are as follows:

Final list of all relations

Ingredient(IngredientName, IngredientType)
- IngredientName: varchar, PK
- IngredientType: varchar

Customer(CustomerID, CustomerName, CustomerAddress, CustomerPhoneNum)
- CustomerID: varchar, PK
- CustomerName: varchar
- CustomerAddress: varchar, not null
- CustomerPhoneNum: char[10]

HasDietaryPreference(**CustomerID**, **IngredientName**, PreferenceType)
- CustomerID: varchar, PK FK
- IngredientName: varchar, PK FK
- PreferenceType: varchar

Orders(OrderID, OrderDate, **CustomerID**, **LicenseNum**, **RestaurantAddress**)
- OrderID: varchar, PK
- OrderDate: date
- CustomerID: varchar, FK
- LicenseNum: varchar, FK
- RestaurantAddress: varchar, FK

DeliveryPerson(LicenseNum, DeliveryPersonName, DeliveryPersonRating, PhoneNum)
- LicenseNum: char[8], PK
- DeliveryPersonName: varchar
- DeliveryPersonRating: float
- PhoneNum: char[10], CK unique

Vehicle(LicensePlateNum, InsuranceExpiryDate, Brand, VehicleModel, VehicleColour)
- LicensePlateNum: varchar, PK
- InsuranceExpiryDate: date
- Brand: varchar
- VehicleModel: varchar
- VehicleColour: varchar

OwnsVehicle(**LicenseNum**, **LicensePlateNum**)
- LicenseNum: char[8], PK FK
- LicensePlateNum: varchar, PK FK

Restaurant(RestaurantAddress, **RestaurantName**, RestaurantRating, RestaurantPhoneNum)
- RestaurantAddress: varchar, PK
- RestaurantName: varchar, FK
- RestaurantRating: float
- RestaurantPhoneNum: char[10], CK unique

RestaurantCuisine(RestaurantName, RestaurantCuisineType)
RestaurantName: varchar, PK
RestaurantCuisineType: varchar

Menu(MenuName, **RestaurantAddress**, MealType)
MenuName: varchar, PK
RestaurantAddress: varchar, PK FK
MealType: varchar

MenuItem(MenuItemName, Calories)
MenuItemName: varchar, PK
Calories: int

MenuFeaturesItem(**MenuItemName**, **MenuName**, ItemPrice)
MenuItemName: varchar, PK FK
MenuName: varchar, PK FK
ItemPrice: float, not null

ItemMadeWith(**MenuItemName**, **IngredientName**)
MenuItemName: varchar, PK FK
IngredientName: varchar, PK FK

OrderContains(**OrderID**, **MenuItemName**)
OrderID: varchar, PK FK
MenuItemName: varchar, PK FK

Drink(**MenuItemName**, Volume, AlcoholContent)
MenuItemName: varchar, PK FK
Volume: int
AlcoholContent: float

RotationalItem(**MenuItemName**, StartDate, EndDate)
MenuItemName: varchar, PK FK
StartDate: date, not null
EndDate: date, not null

RotationalDrink(**MenuItemName**, Volume, AlcoholContent, StartDate, EndDate)
MenuItemName: varchar, PK FK
StartDate: date, not null
EndDate: date, not null
Volume: int
AlcoholContent: float

7. The SQL DDL statements required to create all the tables from item #6. The statements should use the appropriate foreign keys, primary keys, UNIQUE constraints, etc. Unless you know that you will always have exactly x characters for a given character, it is better to use the VARCHAR data type as opposed to a CHAR(Y). For example, UBC courses always use four characters to represent which department offers a course. In that case, you will want to use CHAR(4) for the department attribute in your SQL DDL statement. If you are trying to represent the name of a UBC course, you will want to use VARCHAR as the number of characters in a course name can vary greatly.

```
CREATE TABLE Ingredient (

     IngredientName VARCHAR(50) PRIMARY KEY,
     IngredientType VARCHAR(50)

);

CREATE TABLE Customer (

     CustomerID          VARCHAR(12) PRIMARY KEY,
     CustomerName   VARCHAR(50),
     CustomerAddress     VARCHAR(50) NOT NULL,
     CustomerPhoneNum    CHAR(10)

);

CREATE TABLE HasDietaryPreference (

     CustomerID          VARCHAR(10),
     IngredientName VARCHAR(50),
     PreferenceType VARCHAR(20),

     PRIMARY KEY    (CustomerID, IngredientName),
     FOREIGN KEY    (CustomerID) REFERENCES
          Customer(CustomerID)
          ON DELETE CASCADE
          ON UPDATE CASCADE,
     FOREIGN KEY    (IngredientName) REFERENCES
          Ingredient(IngredientName)
          ON DELETE CASCADE
          ON UPDATE CASCADE

);

CREATE TABLE DeliveryPerson (

     LicenseNum               CHAR(8) PRIMARY KEY,
     DeliveryPersonName       VARCHAR(50),
     DeliveryPersonRating     FLOAT,
     PhoneNum            CHAR(12) UNIQUE

);

CREATE TABLE Vehicle (

     LicensePlateNum          VARCHAR(8) PRIMARY KEY,
     InsuranceExpiryDate      DATE,
     Brand                    VARCHAR(20),
     VehicleModel             VARCHAR(20),
     VehicleColour            VARCHAR(20)
```

```
);

CREATE TABLE OwnsVehicle (

     LicenseNum          CHAR(8),
     LicensePlateNum     VARCHAR(8),

     PRIMARY KEY    (LicenseNum, LicensePlateNum),
     FOREIGN KEY    (LicenseNum) REFERENCES
          DeliveryPerson(LicenseNum),
     FOREIGN KEY    (LicensePlateNum) REFERENCES
          Vehicle(LicensePlateNum)

);

CREATE TABLE Restaurant (

     RestaurantAddress   VARCHAR(50) PRIMARY KEY,
     RestaurantName VARCHAR(50) ,
     RestaurantRating    FLOAT,
     RestaurantPhoneNum  CHAR(10) UNIQUE,

     FOREIGN KEY    (RestaurantName) REFERENCES
          RestaurantCuisine(RestaurantName)
          ON DELETE CASCADE
          ON UPDATE CASCADE

);

CREATE TABLE RestaurantCuisine (

     RestaurantName      VARCHAR(50) PRIMARY KEY,
     RestaurantCuisineType    VARCHAR(20)

);

CREATE TABLE Orders (

     OrderID        VARCHAR(10) PRIMARY KEY,
     OrderDate      DATETIME,
     CustomerID          VARCHAR(10),
     LicenseNum          CHAR(8),
     RestaurantAddress   VARCHAR(50),

     FOREIGN KEY    (CustomerID) REFERENCES
          Customer(CustomerID)
          ON DELETE SET NULL
          ON UPDATE CASCADE,
     FOREIGN KEY    (LicenseNum) REFERENCES
          DeliveryPerson(LicenseNum)
          ON DELETE SET NULL
```

```
          ON UPDATE CASCADE,
     FOREIGN KEY    (RestaurantAddress) REFERENCES
          Restaurant(RestaurantAddress)
          ON DELETE SET NULL
          ON UPDATE CASCADE

);

CREATE TABLE Menu (

     MenuName       VARCHAR(20),
     RestaurantAddress   VARCHAR(50),
     MealType       VARCHAR(20),

     PRIMARY KEY    (MenuName, RestaurantAddress),
     FOREIGN KEY    (RestaurantAddress) REFERENCES
          Restaurant(RestaurantAddress)
          ON DELETE CASCADE
          ON UPDATE CASCADE
);


CREATE TABLE MenuItem    (

     MenuItemName   VARCHAR(50) PRIMARY KEY,
     Calories       INT

);


CREATE TABLE MenuFeaturesItem (

     MenuItemName   VARCHAR(50),
     MenuName       VARCHAR(20),
     ItemPrice      FLOAT NOT NULL,

     PRIMARY KEY    (MenuItemName, MenuName),
     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName),
     FOREIGN KEY    (MenuName) REFERENCES Menu(MenuName)
          ON DELETE CASCADE
          ON UPDATE CASCADE

);


CREATE TABLE ItemMadeWith (

     MenuItemName   VARCHAR(50),
     IngredientName VARCHAR(50),
```

```
     PRIMARY KEY    (MenuItemName, IngredientName),
     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName),
     FOREIGN KEY    (IngredientName) REFERENCES
          Ingredient(IngredientName)

);


CREATE TABLE OrderContains (

     OrderID        VARCHAR(10),
     MenuItemName   VARCHAR(50),

     PRIMARY KEY    (OrderID, MenuItemName),
     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName),
     FOREIGN KEY    (OrderID) REFERENCES Orders(OrderID)
          ON DELETE CASCADE
          ON UPDATE CASCADE

);


CREATE TABLE Drink (

     MenuItemName   VARCHAR(50) PRIMARY KEY,
     Volume         INT,
     AlcoholContent FLOAT,

     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName)

);

CREATE TABLE RotationalItem(

     MenuItemName   VARCHAR(50) PRIMARY KEY,
     StartDate           DATE NOT NULL,
     EndDate        DATE NOT NULL,

     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName)

);

CREATE TABLE RotationalDrink(
```

```
     MenuItemName   VARCHAR(50) PRIMARY KEY,
     StartDate           DATE NOT NULL,
     EndDate        DATE NOT NULL,
     Volume         INT,
     AlcoholContent FLOAT,

     FOREIGN KEY    (MenuItemName) REFERENCES
          MenuItem(MenuItemName)

);
```

8. INSERT statements to populate each table with at least 5 tuples. You will likely want to have more than 5 tuples so that you can have meaningful queries later.

Note: Be consistent with the names used in your ER diagram, schema, and FDs. Make a note if the name has been intentionally changed

```
INSERT INTO Customer(CustomerID, CustomerName,
CustomerAddress, CustomerPhoneNum)
VALUES
     (1, 'Esther Murty', 1234 Granville Street,
     '604-132-2523'),
     (2, 'Melvin Cottis', 5678 Burrard Street,
     '778-233-4961'),
     (3, 'Raff Aspinell', 4321 Main Street, '604-632-8177'),
     (4, 'Sadye Tutsell', 8765 Cambie Street, '604-991-2784'),
     (5, 'Jonah Essberger', 9101 Oak Street, '604-377-0667'),
     (6, 'Terese Musselwhite', 3456 West 4th Avenue,
     '778-828-8315'),
     (7, 'Nanice Spinola', 7890 Kingsway, '604-214-1623'),
     (8, 'Janey Bernucci', 6543 Commercial Drive,
     '778-488-5825'),
     (9, 'Karee Garter', 3210 Fraser Street, '778-263-5321'),
     (10, 'Gillan Heinzler', 9876 Broadway, '604-699-8676');



INSERT INTO DeliveryPerson (LicenseNum, DeliveryPersonName,
DeliveryPersonRating,
     PhoneNum)
VALUES
     ('81685719', 'Hildagarde Scothorne', 0.78,
     '604-220-0356'),
     ('01983055', 'Whitman Goburn', 0.60, '778-704-3887'),
     ('78276122', 'Kayley Massey', 0.86, '604-621-7985'),
     ('85681253', 'Sky McElhargy', 0.83, '778-279-7384'),
     ('29795860', 'Tansy Jakeman', 0.65, '604-916-1407'),
     ('09571867', 'Germayne Shillabeer', 0.74,
     '778-226-0243'),
```

```
     ('99220121', 'Merwin Bartley', 0.86, '604-977-0612'),
     ('69154261', 'Perrine Griffitt', 0.92, '778-547-8319'),
     ('83174286', 'Ciel Bahl', 0.94, '604-130-0961'),
     ('72761716', 'Teddie Townsend', 0.73, '778-434-3030');


INSERT INTO Vehicle (LicensePlateNum, InsuranceExpiryDate,
Brand, VehicleModel, VehicleColour)
VALUES
     ('LAD-201', '2024-12-31', 'Toyota', 'Camry', 'Blue'),
     ('A42-LO2', '2024-11-30', 'Honda', 'Civic', 'Red'),
     ('K6G-2F5', '2024-10-15', 'Ford', 'Escape', 'Silver'),
     ('FGB-343', '2024-09-20', 'Chevrolet', 'Malibu',
'Black'),
     ('H3B-L35', '2024-08-10', 'Nissan', 'Altima', 'White'),
     ('423-JKW', '2024-07-05', 'Hyundai', 'Elantra', 'Gray'),
     ('B9D-23L', '2024-06-15', 'Volkswagen', 'Jetta',
'Green'),
     ('DD5-41E', '2024-05-25', 'Subaru', 'Outback', 'Brown'),
     ('LE4-41A', '2024-03-12', 'Mazda', 'CX-5', 'Orange'),
     ('SIGMA'', '2024-09-12', 'Tesla', 'Model 3', 'White');


INSERT INTO OwnsVehicle (LicenseNum, LicensePlateNum)
VALUES
     ('81685719', 'LAD-201'),
     ('01983055', 'A42-LO2'),
     ('78276122', 'K6G-2F5'),
     ('85681253', 'FGB-343'),
     ('29795860', 'H3B-L35'),
     ('09571867', '423-JKW'),
     ('99220121', 'B9D-23L'),
     ('69154261', 'DD5-41E'),
     ('83174286', 'LE4-41A'),
     ('72761716', 'SIGMA');


INSERT INTO Restaurant (RestaurantAddress, RestaurantName,
RestaurantRating,
     RestaurantPhoneNum)
VALUES
     ('123 Main St', 'Tasty Bites', 4.5, '555-1234'),
     ('456 Elm Ave', 'Spice Haven', 4.2, '555-5678'),
     ('789 Oak Rd', 'Sushi Delight', 4.8, '555-9876'),
     ('101 Maple Ln', 'Pizza Paradise', 4.0, '555-4321'),
     ('222 Cherry Blvd', 'Burger Joint', 3.7, '555-2468'),
     ('333 Pine Dr', 'Mexican Fiesta', 4.3, '555-1357'),
```

```
      ('444 Walnut Ct', 'Café Bliss', 4.6, '555-6789'),
      ('555 Cedar Ave', 'Thai Orchid', 4.4, '555-7890'),
      ('666 Birch St', 'Greek Taverna', 4.1, '555-5432'),
      ('777 Willow Rd', 'Indian Spice', 4.7, '555-8765');


INSERT INTO RestaurantCuisine (RestaurantName,
RestaurantCuisineType)
VALUES
      ('Tasty Bites', 'Indian'),
      ('Spice Haven', 'Thai'),
      ('Sushi Delight', 'Japanese'),
      ('Pizza Paradise', 'Italian'),
      ('Burger Joint', 'American'),
      ('Mexican Fiesta', 'Mexican'),
      ('Café Bliss', 'French'),
      ('Thai Orchid', 'Thai'),
      ('Greek Taverna', 'Greek'),
      ('Indian Spice', 'Indian');


INSERT INTO Orders (OrderID, OrderDate, CustomerID,
LicenseNum, RestaurantAddress)
VALUES
      ('ORD001', '2024-07-15 10:30:00', '8', '81685719', '123 Main St'),
      ('ORD002', '2024-07-16 12:45:00', '4', '01983055', '456 Elm Ave'),
      ('ORD003', '2024-07-17 14:20:00', '7', '78276122', '789 Oak Rd'),
      ('ORD004', '2024-07-18 16:10:00', '5', '85681253', '101 Maple Ln'),
      ('ORD005', '2024-07-19 18:00:00', '6', '29795860', '222 Cherry
Blvd'),
      ('ORD006', '2024-07-20 20:30:00', '9', '09571867', '333 Pine Dr'),
      ('ORD007', '2024-07-21 09:15:00', '1', '99220121', '444 Walnut Ct'),
      ('ORD008', '2024-07-22 11:40:00', '6', '69154261', '555 Cedar Ave'),
      ('ORD009', '2024-07-23 13:55:00', '1', '83174286', '666 Birch St'),
      ('ORD010', '2024-07-24 15:25:00', '4', '72761716', '777 Willow Rd'),
      ('ORD011', '2024-07-25 17:30:00', '9', '81685719', '123 Main St'),
      ('ORD012', '2024-07-26 19:45:00', '6', '01983055', '456 Elm Ave'),
      ('ORD013', '2024-07-27 08:20:00', '9', '78276122', '789 Oak Rd'),
      ('ORD014', '2024-07-28 10:10:00', '7', '85681253', '101 Maple Ln'),
      ('ORD015', '2024-07-29 12:00:00', '8', '29795860', '222 Cherry
Blvd'),
      ('ORD016', '2024-07-30 14:30:00', '2', '09571867', '333 Pine Dr'),
      ('ORD017', '2024-07-31 09:45:00', '7', '99220121', '444 Walnut Ct'),
      ('ORD018', '2024-08-01 11:20:00', '9', '69154261', '555 Cedar Ave'),
      ('ORD019', '2024-08-02 13:35:00', '8', '83174286', '666 Birch St'),
      ('ORD020', '2024-08-03 15:55:00', '2', '72761716', '777 Willow Rd');
```

```
INSERT INTO Menu (MenuName, RestaurantAddress, MealType)
VALUES
     ('Breakfast Specials', '123 Main St', 'Breakfast'),
     ('Lunch Combos', '123 Main St', 'Lunch'),
     ('Dinner Feasts', '123 Main St', 'Dinner'),
     ('Spicy Platters', '456 Elm Ave', 'Lunch'),
     ('Evening Delight', '456 Elm Ave', 'Dinner'),
     ('Lunch Specials', '789 Oak Rd', 'Lunch'),
     ('Dinner Deals', '789 Oak Rd', 'Dinner'),
     ('2 for $10', '101 Maple Ln', 'Lunch'),
     ('Desserts', '101 Maple Ln', 'Dinner'),
     ('Burgers', '222 Cherry Blvd', 'Lunch'),
     ('Milkshakes', '222 Cherry Blvd', 'Dinner'),
     ('Fiesta Lunch', '333 Pine Dr', 'Lunch'),
     ('Fiesta Dinners', '333 Pine Dr', 'Dinner'),
     ('Breakfast Specials', '444 Walnut Ct', 'Breakfast'),
     ('Lunch Specials', '444 Walnut Ct', 'Lunch'),
     (`Dinner Specials', '444 Walnut Ct', 'Dinner'),
     ('Lunch Combos', '555 Cedar Ave', 'Lunch'),
     ('Thai Dinner', '555 Cedar Ave', 'Dinner'),
     ('Greek Lunch', '666 Birch St', 'Lunch'),
     ('Happy Hour', '666 Birch St', 'Dinner'),
     (`Lunch Specials', '777 Willow Rd', 'Lunch'),
     ('Dinner Feasts', '777 Willow Rd', 'Dinner');

INSERT INTO MenuItem (MenuItemName, Calories)
VALUES
     ('Cheeseburger', 550),
     ('Grilled Chicken Salad', 350),
     ('Veggie Wrap', 300),
     ('Spaghetti Bolognese', 600),
     ('Margherita Pizza', 700),
     ('Sushi Roll', 250),
     ('Pad Thai', 650),
     ('Tacos', 500),
     ('French Fries', 400),
     ('Caesar Salad', 350),
     ('Breakfast Burrito', 450),
     ('Pancakes', 500),
     ('Chocolate Cake', 600),
     ('Apple Pie', 450),
```

```
    ('Chicken Curry', 550),
    ('Beef Steak', 700),
    ('Fish and Chips', 650),
    ('Vegetable Stir Fry', 400),
    ('Lentil Soup', 200),
    ('Chicken Quesadilla', 600),
    ('Avocado Toast', 300),
    ('BLT Sandwich', 450),
    ('Pepperoni Pizza', 750),
    ('Fettuccine Alfredo', 800),
    ('Shrimp Tacos', 450),
    ('Clam Chowder', 300),
    ('BBQ Ribs', 900),
    ('Garden Salad', 200),
    ('Eggplant Parmesan', 600),
    ('Turkey Sandwich', 400);

INSERT INTO MenuFeaturesItem (MenuItemName, MenuName,
ItemPrice)
VALUES
    ('Cheeseburger', 'Burgers', 8.99),
    ('Grilled Chicken Salad', 'Lunch Specials', 10.99),
    ('Veggie Wrap', 'Lunch Specials', 7.99),
    ('Spaghetti Bolognese', 'Dinner Feasts', 12.99),
    ('Margherita Pizza', 'Dinner Feasts', 11.99),
    ('Sushi Roll', 'Lunch Specials', 8.49),
    ('Pad Thai', 'Thai Dinner', 11.49),
    ('Tacos', 'Fiesta Lunch', 9.49),
    ('French Fries', 'Burgers', 3.99),
    ('Caesar Salad', 'Lunch Combos', 6.99),
    ('Breakfast Burrito', 'Breakfast Specials', 5.99),
    ('Pancakes', 'Breakfast Specials', 7.99),
    ('Chocolate Cake', 'Desserts', 6.49),
    ('Apple Pie', 'Desserts', 5.49),
    ('Chicken Curry', 'Dinner Feasts', 11.99),
    ('Beef Steak', 'Dinner Feasts', 14.99),
    ('Fish and Chips', 'Lunch Combos', 10.99),
    ('Vegetable Stir Fry', 'Lunch Combos', 8.99),
    ('Lentil Soup', 'Lunch Combos', 6.49),
    ('Chicken Quesadilla', 'Lunch Combos', 9.49),
    ('Avocado Toast', 'Breakfast Specials', 6.99),
    ('BLT Sandwich', 'Lunch Combos', 7.99),
```

```
    ('Pepperoni Pizza', 'Dinner Deals', 12.99),
    ('Fettuccine Alfredo', 'Dinner Feasts', 13.99),
    ('Shrimp Tacos', 'Fiesta Dinners', 11.49),
    ('Clam Chowder', 'Lunch Combos', 7.99),
    ('BBQ Ribs', 'Dinner Feasts', 15.99),
    ('Garden Salad', 'Lunch Combos', 5.99),
    ('Eggplant Parmesan', 'Dinner Feasts', 12.49),
    ('Turkey Sandwich', 'Lunch Combos', 8.49);

INSERT INTO Drink (MenuItemName, Volume, AlcoholContent)
VALUES
    ('Beer', 355, 0.05),
    ('Red Wine', 150, 0.125),
    ('White Wine', 150, 0.11),
    ('Whiskey', 50, 0.40),
    ('Cocktail', 200, 0.15),
    ('Vodka', 50, 0.40),
    ('Gin', 50, 0.375),
    ('Rum', 50, 0.35),
    ('Champagne', 150, 0.12),
    ('Liqueur', 50, 0.25),
    ('Cola', 355, 0.0),
    ('Diet Cola', 355, 0.0),
    ('Lemonade', 355, 0.0),
    ('Orange Juice', 355, 0.0),
    ('Espresso', 60, 0.0),
    ('Latte', 240, 0.0),
    ('Cappuccino', 240, 0.0);

INSERT INTO RotationalItem (MenuItemName, StartDate, EndDate)
VALUES
    ('Summer BBQ Cheeseburger Extravaganza', '2024-07-01',
    '2024-07-31'),
    ('Fall Harvest Grilled Chicken Salad', '2024-08-01',
    '2024-08-31'),
    ('Autumn Spaghetti Bolognese Delight', '2024-09-01',
    '2024-09-30'),
    ('Halloween Margherita Pizza Special', '2024-10-01',
    '2024-10-31'),
    ('Thanksgiving Sushi Roll Medley', '2024-11-01',
    '2024-11-30'),
    ('Winter Pad Thai Comfort', '2024-12-01', '2024-12-31'),
```

```
    ('New Year’s Tantalizing Taco Trio', '2025-01-01',
    '2025-01-31'),
    ('Valentine’s Golden French Fries', '2025-02-01',
    '2025-02-28'),
    ('Spring Caesar Salad Celebration', '2025-03-01',
    '2025-03-31'),
    ('Easter Breakfast Burrito Surprise', '2025-04-01',
    '2025-04-30'),
    ('Mother’s Day Fluffy Pancake Stack', '2025-05-01',
    '2025-05-31'),
    ('Summer Solstice Chocolate Cake Bliss', '2025-06-01',
    '2025-06-30'),
    ('4th of July Apple Pie Treat', '2025-07-01',
    '2025-07-31'),
    ('Back to School Chicken Curry Feast', '2025-08-01',
    '2025-08-31'),
    ('Labor Day Prime Beef Steak Special', '2025-09-01',
    '2025-09-30'),
    ('Oktoberfest Fish and Chips Extravaganza', '2025-10-01',
    '2025-10-31'),
    ('Thanksgiving Vegetable Stir Fry Fusion', '2025-11-01',
    '2025-11-30'),
    ('Holiday Lentil Soup Comfort Bowl', '2025-12-01',
    '2025-12-31'),
    ('New Year’s Zesty Chicken Quesadilla', '2026-01-01',
    '2026-01-31'),
    ('Valentine’s Avocado Toast Heaven', '2026-02-01',
    '2026-02-28');

INSERT INTO RotationalDrink (MenuItemName, StartDate, EndDate,
    Volume, AlcoholContent)
VALUES
    ('Summer Breeze Mojito', '2024-07-01', '2024-07-31', 300,
    0.08),
    ('Autumn Spiced Apple Cider', '2024-08-01', '2024-08-31',
    250, 0.05),
    ('Halloween Pumpkin Ale', '2024-09-01', '2024-09-30',
    330, 0.06),
    ('Thanksgiving Cranberry Punch', '2024-10-01',
    '2024-10-31', 200, 0.04),
    ('Holiday Eggnog Delight', '2024-11-01', '2024-11-30',
    250, 0.12),
    ('Winter Wonderland Mulled Wine', '2024-12-01',
    '2024-12-31', 300, 0.15),
    ('New Year’s Champagne Sparkler', '2025-01-01',
    '2025-01-31', 150, 0.12),
```

```
    ('Valentine’s Raspberry Mojito', '2025-02-01',
    '2025-02-28', 300, 0.08),
    ('Spring Blossom Gin Fizz', '2025-03-01', '2025-03-31',
    250, 0.10),
    ('Easter Mimosa Burst', '2025-04-01', '2025-04-30', 200,
    0.08),
    ('Mother’s Day Lavender Lemonade', '2025-05-01',
    '2025-05-31', 250, 0.0),
    ('Summer Solstice Sangria', '2025-06-01', '2025-06-30',
    300, 0.10),
    ('4th of July Firework Punch', '2025-07-01',
    '2025-07-31', 250, 0.09),
    ('Back to School Iced Coffee', '2025-08-01',
    '2025-08-31', 200, 0.0),
    ('Labor Day Spiked Lemonade', '2025-09-01', '2025-09-30',
    300, 0.07),
    ('Oktoberfest Bavarian Lager', '2025-10-01',
    '2025-10-31', 330, 0.06),
    ('Thanksgiving Maple Bourbon', '2025-11-01',
    '2025-11-30', 150, 0.14),
    ('Holiday Peppermint Martini', '2025-12-01',
    '2025-12-31', 200, 0.13),
    ('Winter Chai Latte', '2025-12-01', '2025-12-31', 240,
    0.0),
    ('Spring Citrus Lemonade', '2026-03-01', '2026-03-31',
    250, 0.0),
    ('Summer Iced Matcha', '2026-07-01', '2026-07-31', 250,
    0.0),
    ('Autumn Spiced Pumpkin Latte', '2026-10-01',
    '2026-10-31', 240, 0.0),
    ('Holiday Gingerbread Latte', '2026-12-01', '2026-12-31',
    240, 0.0);

INSERT INTO OrderContains (OrderID, MenuItemName)
VALUES
    ('ORD001', 'Cheeseburger'),
    ('ORD001', 'French Fries'),
    ('ORD002', 'Grilled Chicken Salad'),
    ('ORD002', 'Vegetable Stir Fry'),
    ('ORD003', 'Margherita Pizza'),
    ('ORD003', 'Pancakes'),
    ('ORD004', 'Chicken Curry'),
    ('ORD004', 'Fish and Chips'),
    ('ORD005', 'Shrimp Tacos'),
    ('ORD005', 'Garden Salad'),
```

```
    ('ORD006', 'BBQ Ribs'),
    ('ORD006', 'Fettuccine Alfredo');
    ('ORD007', 'Breakfast Burrito'),
    ('ORD007', 'Avocado Toast'),
    ('ORD008', 'Pad Thai'),
    ('ORD008', 'Thai Dinner'),
    ('ORD009', 'BLT Sandwich'),
    ('ORD009', 'Pepperoni Pizza'),
    ('ORD010', 'Chicken Curry'),
    ('ORD010', 'Fish and Chips'),
    ('ORD011', 'Cheeseburger'),
    ('ORD011', 'French Fries'),
    ('ORD012', 'Grilled Chicken Salad'),
    ('ORD012', 'Vegetable Stir Fry'),
    ('ORD013', 'Chocolate Cake'),
    ('ORD013', 'Apple Pie'),
    ('ORD014', 'Margherita Pizza'),
    ('ORD014', 'Pancakes'),
    ('ORD015', 'Shrimp Tacos'),
    ('ORD015', 'Garden Salad'),
    ('ORD016', 'Clam Chowder'),
    ('ORD016', 'Eggplant Parmesan'),
    ('ORD017', 'Beef Steak'),
    ('ORD017', 'Lentil Soup'),
    ('ORD018', 'Chicken Quesadilla'),
    ('ORD018', 'Spaghetti Bolognese'),
    ('ORD019', 'BBQ Ribs'),
    ('ORD019', 'Fettuccine Alfredo'),
    ('ORD020', 'Turkey Sandwich'),
    ('ORD020', 'Cheeseburger');

INSERT INTO Ingredient (IngredientName, IngredientType)
VALUES
    ('Ground Beef', 'Meat'),
    ('Cheddar Cheese', 'Dairy'),
    ('Onions', 'Vegetable'),
    ('Potatoes', 'Vegetable'),
    ('Olive Oil', 'Oil'),
    ('Chicken Breast', 'Meat'),
    ('Bell Peppers', 'Vegetable'),
    ('Broccoli', 'Vegetable'),
    ('Tofu', 'Soy Product'),
```

```
    ('Tomatoes', 'Vegetable'),
    ('Garlic', 'Spice'),
    ('Rice', 'Grain'),
    ('Nori', 'Seaweed'),
    ('Peanuts', 'Nut'),
    ('Lettuce', 'Vegetable'),
    ('Parmesan Cheese', 'Dairy'),
    ('Eggs', 'Dairy'),
    ('Flour Tortilla', 'Grain'),
    ('Flour', 'Grain'),
    ('Milk', 'Dairy'),
    ('Sugar', 'Sweetener'),
    ('Butter', 'Dairy'),
    ('Apples', 'Fruit'),
    ('Coconut Milk', 'Dairy Substitute'),
    ('Salt', 'Seasoning'),
    ('Black Pepper', 'Spice'),
    ('Salmon', 'Fish'),
    ('Cream', 'Dairy'),
    ('Cabbage', 'Vegetable'),
    ('Clams', 'Seafood'),
    ('Pork Ribs', 'Meat'),
    ('BBQ Sauce', 'Sauce'),
    ('Cucumber', 'Vegetable'),
    ('Eggplant', 'Vegetable'),
    ('Marinara Sauce', 'Sauce'),
    ('Turkey Breast', 'Meat');

INSERT INTO ItemMadeWith (MenuItemName, IngredientName)
VALUES
    ('Cheeseburger', 'Ground Beef'),
    ('Cheeseburger', 'Cheddar Cheese'),
    ('Cheeseburger', 'Onions'),
    ('French Fries', 'Potatoes'),
    ('French Fries', 'Olive Oil'),
    ('Grilled Chicken Salad', 'Chicken Breast'),
    ('Grilled Chicken Salad', 'Bell Peppers'),
    ('Grilled Chicken Salad', 'Broccoli'),
    ('Veggie Wrap', 'Tofu'),
    ('Veggie Wrap', 'Bell Peppers'),
    ('Veggie Wrap', 'Tomatoes'),
    ('Spaghetti Bolognese', 'Ground Beef'),
```

```
('Spaghetti Bolognese', 'Tomatoes'),
('Spaghetti Bolognese', 'Garlic'),
('Margherita Pizza', 'Cheddar Cheese'),
('Margherita Pizza', 'Tomatoes'),
('Margherita Pizza', 'Olive Oil'),
('Sushi Roll', 'Salmon'),
('Sushi Roll', 'Rice'),
('Sushi Roll', 'Nori'),
('Pad Thai', 'Rice'),
('Pad Thai', 'Tofu'),
('Pad Thai', 'Peanuts'),
('Tacos', 'Ground Beef'),
('Tacos', 'Bell Peppers'),
('Tacos', 'Onions'),
('Caesar Salad', 'Chicken Breast'),
('Caesar Salad', 'Lettuce'),
('Caesar Salad', 'Parmesan Cheese'),
('Breakfast Burrito', 'Eggs'),
('Breakfast Burrito', 'Cheddar Cheese'),
('Breakfast Burrito', 'Flour Tortilla'),
('Pancakes', 'Flour'),
('Pancakes', 'Eggs'),
('Pancakes', 'Milk'),
('Chocolate Cake', 'Flour'),
('Chocolate Cake', 'Sugar'),
('Chocolate Cake', 'Butter'),
('Apple Pie', 'Apples'),
('Apple Pie', 'Flour'),
('Apple Pie', 'Sugar'),
('Chicken Curry', 'Chicken Breast'),
('Chicken Curry', 'Garlic'),
('Chicken Curry', 'Coconut Milk'),
('Beef Steak', 'Ground Beef'),
('Beef Steak', 'Salt'),
('Beef Steak', 'Black Pepper'),
('Fish and Chips', 'Salmon'),
('Fish and Chips', 'Flour'),
('Fish and Chips', 'Potatoes'),
('Vegetable Stir Fry', 'Broccoli'),
('Vegetable Stir Fry', 'Bell Peppers'),
('Vegetable Stir Fry', 'Garlic'),
('Lentil Soup', 'Lentils'),
```

```
     ('Lentil Soup', 'Tomatoes'),
     ('Lentil Soup', 'Garlic'),
     ('Chicken Quesadilla', 'Chicken Breast'),
     ('Chicken Quesadilla', 'Cheddar Cheese'),
     ('Chicken Quesadilla', 'Flour Tortilla'),
     ('Avocado Toast', 'Avocado'),
     ('Avocado Toast', 'Bread'),
     ('Avocado Toast', 'Olive Oil'),
     ('BLT Sandwich', 'Bacon'),
     ('BLT Sandwich', 'Lettuce'),
     ('BLT Sandwich', 'Tomatoes'),
     ('Pepperoni Pizza', 'Pepperoni'),
     ('Pepperoni Pizza', 'Cheddar Cheese'),
     ('Pepperoni Pizza', 'Tomatoes'),
     ('Fettuccine Alfredo', 'Fettuccine'),
     ('Fettuccine Alfredo', 'Cream'),
     ('Fettuccine Alfredo', 'Parmesan Cheese'),
     ('Shrimp Tacos', 'Shrimp'),
     ('Shrimp Tacos', 'Corn Tortilla'),
     ('Shrimp Tacos', 'Cabbage'),
     ('Clam Chowder', 'Clams'),
     ('Clam Chowder', 'Cream'),
     ('Clam Chowder', 'Potatoes'),
     ('BBQ Ribs', 'Pork Ribs'),
     ('BBQ Ribs', 'BBQ Sauce'),
     ('BBQ Ribs', 'Garlic'),
     ('Garden Salad', 'Lettuce'),
     ('Garden Salad', 'Tomatoes'),
     ('Garden Salad', 'Cucumber'),
     ('Eggplant Parmesan', 'Eggplant'),
     ('Eggplant Parmesan', 'Marinara Sauce'),
     ('Eggplant Parmesan', 'Parmesan Cheese'),
     ('Turkey Sandwich', 'Turkey Breast'),
     ('Turkey Sandwich', 'Bread'),
     ('Turkey Sandwich', 'Lettuce');

INSERT INTO HasDietaryPreference (CustomerID, IngredientName,
     PreferenceType)
VALUES
     (2, 'Peanut', 'Nut Allergy'),
     (5, 'Salmon', 'Seafood Allergy'),
     (6, 'Clams', 'Seafood Allergy),
```

```
(6, 'Butter', 'Lactose Intolerant'),
(7, 'Chicken Breast', 'Vegetarian');
```