Each team is required to submit the SQL code necessary to implement the database design, including the insertion of a minimal amount of data (at least ten rows per table) using SQL INSERT statements or the Data Import Wizard.

Teams are expected to create the following database objects:

- At least three table-level CHECK constraints.

- Each table must have a primary key.

- Relationships between tables must enforce foreign key constraints.

- Appropriate rules, such as nullability, must be enforced.

- Each column should be assigned the appropriate data type.

Include an identity column or equivalent mechanism for automatic ID generation.

1 .sql file containing all DDLs for all database objects created. The script must drop all of the objects if exists, so if there is failure the script can be re-run.

1 .sql script for INSERTs

--sql code for DDLs:

CREATE DATABASE FoodDeliverySystem;

GO

USE FoodDeliverySystem;

GO

DROP TABLE IF EXISTS Review, PaymentMethod, OrderItem, Delivery, [Order], Address, Customer, Driver, Restaurant, MenuItem, MenuCategory;

GO

-- Customer Table

CREATE TABLE Customer (

CustomerID INT IDENTITY(1,1),

CustomerFirstName VARCHAR(50) NOT NULL,

CustomerLastName VARCHAR(50) NOT NULL,

PhoneNumber VARCHAR(15) NOT NULL,

Email VARCHAR(100) NOT NULL UNIQUE,

Password VARCHAR(255) NOT NULL,

CONSTRAINT PK\_Customer PRIMARY KEY (CustomerID),

CONSTRAINT chk\_PhoneNumber CHECK (PhoneNumber LIKE '[0-9]%'),

CONSTRAINT chk\_EmailFormat CHECK (Email LIKE '%@%.%')

);

-- Address Table

CREATE TABLE Address (

AddressID INT IDENTITY(1,1),

CustomerID INT NOT NULL,

StreetName VARCHAR(100) NOT NULL,

StreetNumber VARCHAR(20) NOT NULL,

City VARCHAR(50) NOT NULL,

State VARCHAR(50) NOT NULL,

ZipCode VARCHAR(10) NOT NULL,

CONSTRAINT PK\_Address PRIMARY KEY (AddressID),

CONSTRAINT FK\_Address\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Driver Table

CREATE TABLE Driver (

DriverID INT IDENTITY(1,1),

DriverName VARCHAR(100) NOT NULL,

PhoneNumber VARCHAR(15) NOT NULL,

VehicleInfo VARCHAR(100) NOT NULL,

CONSTRAINT PK\_Driver PRIMARY KEY (DriverID),

CONSTRAINT CHK\_Driver\_Phone CHECK (PhoneNumber LIKE '[0-9]%')

);

-- Restaurant Table

CREATE TABLE Restaurant (

RestaurantID INT IDENTITY(1,1),

Name VARCHAR(100) NOT NULL,

Address VARCHAR(255) NOT NULL,

Rating DECIMAL(3,2),

CuisineType VARCHAR(50),

CONSTRAINT PK\_Restaurant PRIMARY KEY (RestaurantID)

);

-- Payment Method Table

CREATE TABLE PaymentMethod (

PaymentID INT IDENTITY(1,1),

CustomerID INT NOT NULL,

CardNumber VARCHAR(16),

ExpDate DATE,

PaymentType VARCHAR(50),

CONSTRAINT PK\_PaymentMethod PRIMARY KEY (PaymentID),

CONSTRAINT FK\_PaymentMethod\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Order Table

CREATE TABLE [Order] (

OrderID INT IDENTITY(1,1),

CustomerID INT NOT NULL,

PaymentID INT NOT NULL,

TotalAmount DECIMAL(10,2) NOT NULL,

Status VARCHAR(50) NOT NULL,

OrderDate DATE,

DeliveryType VARCHAR(50),

CONSTRAINT PK\_Order PRIMARY KEY (OrderID),

CONSTRAINT CHK\_Delivery\_Type CHECK (DeliveryType IN ('Standard', 'Express', 'Pickup')),

CONSTRAINT FK\_Order\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

CONSTRAINT FK\_Order\_PaymentMethod FOREIGN KEY (PaymentID) REFERENCES PaymentMethod(PaymentID),

CONSTRAINT CHK\_Order\_Status CHECK (Status IN ('Pending', 'Completed', 'Cancelled')),

CONSTRAINT CHK\_Total\_Amount CHECK (TotalAmount > 0)

);

-- Delivery Table

CREATE TABLE Delivery (

OrderID INT,

DriverID INT NOT NULL,

AddressID INT NOT NULL,

DeliveryStatus VARCHAR(50) NOT NULL,

PickupTime DATETIME,

ArrivalTime DATETIME,

CONSTRAINT PK\_Delivery PRIMARY KEY (OrderID),

CONSTRAINT FK\_Delivery\_Order FOREIGN KEY (OrderID) REFERENCES [Order](OrderID),

CONSTRAINT FK\_Delivery\_Driver FOREIGN KEY (DriverID) REFERENCES Driver(DriverID),

CONSTRAINT FK\_Delivery\_Address FOREIGN KEY (AddressID) REFERENCES Address(AddressID)

);

-- MenuCategory Table

CREATE TABLE MenuCategory (

MenuCategoryID INT IDENTITY(1,1),

MenuCategoryName VARCHAR(100),

CONSTRAINT PK\_MenuCategory PRIMARY KEY (MenuCategoryID)

);

-- MenuItem Table

CREATE TABLE MenuItem (

MenuItemID INT IDENTITY(1,1),

MenuCategoryID INT,

Name VARCHAR(100) NOT NULL,

Description VARCHAR(255),

Price DECIMAL(8,2) NOT NULL,

Available BIT,

CONSTRAINT PK\_MenuItem PRIMARY KEY (MenuItemID),

CONSTRAINT FK\_MenuItem\_MenuCategory FOREIGN KEY (MenuCategoryID) REFERENCES MenuCategory(MenuCategoryID)

);

-- OrderItem Table

CREATE TABLE OrderItem (

OrderItemID INT IDENTITY(1,1),

RestaurantID INT NOT NULL,

OrderID INT NOT NULL,

MenuItemID INT NOT NULL,

Quantity INT NOT NULL,

UnitPrice DECIMAL(10,2) NOT NULL,

Notes VARCHAR(255),

CONSTRAINT PK\_OrderItem PRIMARY KEY (OrderItemID),

CONSTRAINT FK\_OrderItem\_Restaurant FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID),

CONSTRAINT FK\_OrderItem\_Order FOREIGN KEY (OrderID) REFERENCES [Order](OrderID),

CONSTRAINT FK\_OrderItem\_MenuItem FOREIGN KEY (MenuItemID) REFERENCES MenuItem(MenuItemID)

);

-- Review Table

CREATE TABLE Review (

ReviewID INT IDENTITY(1,1),

RestaurantID INT NOT NULL,

CustomerID INT NOT NULL,

Rating INT NOT NULL,

Comment VARCHAR(255),

ReviewDate DATE NOT NULL,

CONSTRAINT PK\_Review PRIMARY KEY (ReviewID),

CONSTRAINT FK\_Review\_Restaurant FOREIGN KEY (RestaurantID) REFERENCES Restaurant(RestaurantID),

CONSTRAINT FK\_Review\_Customer FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),

CONSTRAINT CHK\_Review\_Rating CHECK (Rating BETWEEN 1 AND 5)

);

--Insert code

--Customer data

INSERT INTO Customer (CustomerFirstName, CustomerLastName, PhoneNumber, Email, Password) VALUES

('John', 'Doe', '1234567890', 'johndoe@example.com', 'password123'), ('Jane', 'Smith', '2345678901', 'janesmith@example.com', 'securepass'), ('Mike', 'Johnson', '3456789012', 'mikej@example.com', 'passMike99'), ('Sarah', 'Brown', '4567890123', 'sarahb@example.com', 'sarahPass!'), ('David', 'Wilson', '5678901234', 'davidw@example.com', 'DavidSecure'), ('Emily', 'Jones', '6789012345', 'emilyj@example.com', 'EmilyPass12'), ('Chris', 'Taylor', '7890123456', 'christ@example.com', 'TaylorPass34'), ('Laura', 'White', '8901234567', 'lauraw@example.com', 'WhiteLaura56'), ('Robert', 'Martin', '9012345678', 'robertm@example.com', 'RobMart78'), ('Sophia', 'Lee', '0123456789', 'sophial@example.com', 'SophiaLee90');

--Restaurant data

INSERT INTO Restaurant (Name, Address, Rating, CuisineType)

VALUES

('The Gourmet Spot', '123 Main St, New York, NY', 4.5, 'Italian'),

('Sushi Haven', '456 Elm St, Los Angeles, CA', 4.7, 'Japanese'),

('Burger Palace', '789 Oak St, Chicago, IL', 4.3, 'American'),

('Taco Fiesta', '321 Maple St, Houston, TX', 4.2, 'Mexican'),

('Curry House', '654 Pine St, Miami, FL', 4.6, 'Indian'),

('Vegan Delight', '987 Cedar St, San Francisco, CA', 4.8, 'Vegan'),

('BBQ Kingdom', '147 Walnut St, Dallas, TX', 4.4, 'BBQ'),

('French Bistro', '258 Spruce St, Seattle, WA', 4.7, 'French'),

('Dim Sum Corner', '369 Birch St, Boston, MA', 4.5, 'Chinese'),

('Greek Taverna', '852 Chestnut St, Denver, CO', 4.6, 'Greek');

--Address Data

INSERT INTO Address (CustomerID, StreetName, StreetNumber, City, State, ZipCode) VALUES

(1, 'Main St', '123', 'New York', 'NY', '10001'),

(2, 'Elm St', '456', 'Los Angeles', 'CA', '90001'),

(3, 'Oak St', '789', 'Chicago', 'IL', '60601'),

(4, 'Maple St', '321', 'Houston', 'TX', '77001'),

(5, 'Pine St', '654', 'Miami', 'FL', '33101'),

(6, 'Cedar St', '987', 'San Francisco', 'CA', '94101'),

(7, 'Walnut St', '147', 'Dallas', 'TX', '75201'),

(8, 'Spruce St', '258', 'Seattle', 'WA', '98101'),

(9, 'Birch St', '369', 'Boston', 'MA', '02101'),

(10, 'Chestnut St', '852', 'Denver', 'CO', '80201');

--Review Data

INSERT INTO Review (RestaurantID, CustomerID, Rating, Comment, ReviewDate) VALUES (1, 1, 5, 'Amazing food and great service!', '2024-03-01'),

(2, 2, 4, 'Fresh sushi, but a bit pricey.', '2024-03-02'),

(3, 3, 5, 'Best burgers in town!', '2024-03-03'),

(4, 4, 3, 'Tacos were good, but service was slow.', '2024-03-04'), (5, 5, 5, 'Excellent curry! Will come again.', '2024-03-05'),

(6, 6, 4, 'Great vegan options!', '2024-03-06'),

(7, 7, 5, 'BBQ ribs were delicious!', '2024-03-07'),

(8, 8, 4, 'Authentic French flavors.', '2024-03-08'),

(9, 9, 5, 'Dim sum was fresh and tasty.', '2024-03-09'),

(10, 10, 4, 'Greek food was good, but portions were small.', '2024-03-10');

--MenuCategory Data

INSERT INTO MenuCategory (MenuCategoryName) VALUES ('Appetizers'), ('Main Course'), ('Desserts'), ('Beverages'), ('Seafood'), ('Vegetarian'), ('Fast Food'), ('BBQ'), ('Chinese Cuisine'), ('Italian Cuisine');

--MenuItem

INSERT INTO MenuItem (MenuCategoryID, Name, Description, Price, Available) VALUES (1, 'Spring Rolls', 'Crispy rolls stuffed with vegetables', 5.99, 1),

(2, 'Grilled Chicken', 'Juicy grilled chicken breast', 12.99, 1),

(3, 'Chocolate Cake', 'Rich chocolate layered cake', 6.50, 1),

(4, 'Iced Coffee', 'Cold brewed coffee with ice', 3.99, 1),

(5, 'Lobster Bisque', 'Creamy lobster soup', 9.99, 1),

(6, 'Vegan Burger', 'Plant-based burger with lettuce and tomato', 10.50, 1),

(7, 'Cheeseburger', 'Classic beef burger with cheese', 8.99, 1),

(8, 'BBQ Ribs', 'Slow-cooked ribs with BBQ sauce', 14.99, 1),

(9, 'Kung Pao Chicken', 'Spicy stir-fried chicken with peanuts', 11.99, 1),

(10, 'Margherita Pizza', 'Classic Italian pizza with fresh basil and mozzarella', 12.50, 1);

--paymentMethod Data

INSERT INTO PaymentMethod (CustomerID, CardNumber, ExpDate, PaymentType) VALUES

(1, '4111111111111111', '2026-12-31', 'Credit Card'),

(2, '5500000000000004', '2025-10-15', 'Credit Card'),

(3, '340000000000009', '2027-05-20', 'Credit Card'),

(4, '6011000000000004', '2026-08-22', 'Debit Card'),

(5, '4111111111111129', '2028-03-12', 'Credit Card'),

(6, '5500000000000020', '2025-11-10', 'Debit Card'),

(7, '340000000000020', '2026-07-18', 'Credit Card'),

(8, '6011000000000036', '2027-09-25', 'Credit Card'),

(9, '4111111111111145', '2026-06-14', 'Debit Card'),

(10, '5500000000000053', '2025-12-30', 'Credit Card');

--Order Data

INSERT INTO [Order] (CustomerID, PaymentID, TotalAmount, Status, OrderDate, DeliveryType) VALUES

(1, 1, 25.99, 'Completed', '2025-03-01', 'Standard'),

(2, 2, 45.50, 'Pending', '2025-03-02', 'Express'),

(3, 3, 32.75, 'Completed', '2025-03-03', 'Pickup'),

(4, 4, 15.99, 'Cancelled', '2025-03-04', 'Standard'),

(5, 5, 67.80, 'Completed', '2025-03-05', 'Express'),

(6, 6, 12.50, 'Pending', '2025-03-06', 'Standard'),

(7, 7, 98.25, 'Completed', '2025-03-07', 'Pickup'),

(8, 8, 43.60, 'Cancelled', '2025-03-08', 'Express'),

(9, 9, 27.30, 'Completed', '2025-03-09', 'Standard'),

(10, 10, 55.99, 'Pending', '2025-03-10', 'Express');

--OrderItem Data

INSERT INTO OrderItem (RestaurantID, OrderID, MenuItemID, Quantity, UnitPrice, Notes) VALUES

(1, 1, 1, 2, 5.99, 'Extra sauce'),

(2, 2, 2, 1, 12.99, 'No onions'),

(3, 3, 3, 3, 6.50, NULL),

(4, 4, 4, 1, 3.99, 'Less ice'),

(5, 5, 5, 2, 9.99, 'Extra spicy'),

(6, 6, 6, 1, 10.50, NULL),

(7, 7, 7, 2, 8.99, 'Add bacon'),

(8, 8, 8, 1, 14.99, 'BBQ sauce on side'),

(9, 9, 9, 3, 11.99, NULL),

(10, 10, 10, 2, 12.50, 'Extra cheese');

--Driver Data

INSERT INTO Driver (DriverName, PhoneNumber, VehicleInfo) VALUES

('Michael Johnson', '1234567890', 'Toyota Corolla'), ('Jessica Brown', '2345678901', 'Honda Civic'),

('Daniel Smith', '3456789012', 'Ford Focus'),

('Emily White', '4567890123', 'Nissan Altima'),

('David Wilson', '5678901234', 'Chevrolet Malibu'),

('Sophia Anderson', '6789012345', 'Hyundai Sonata'),

('James Taylor', '7890123456', 'Volkswagen Jetta'),

('Olivia Martin', '8901234567', 'Mazda 3'),

('William Clark', '9012345678', 'Tesla Model 3'),

('Emma Harris', '0123456789', 'Subaru Impreza');

--Delivery Data

INSERT INTO Delivery (OrderID, DriverID, AddressID, DeliveryStatus, PickupTime, ArrivalTime) VALUES

(1, 1, 1, 'Delivered', '2025-03-01 12:30:00', '2025-03-01 13:00:00'),

(2, 2, 2, 'Out for Delivery', '2025-03-02 14:00:00', NULL),

(3, 3, 3, 'Pending', NULL, NULL),

(4, 4, 4, 'Delivered', '2025-03-04 16:45:00', '2025-03-04 17:30:00'),

(5, 5, 5, 'Out for Delivery', '2025-03-05 18:00:00', NULL),

(6, 6, 6, 'Pending', NULL, NULL),

(7, 7, 7, 'Delivered', '2025-03-07 11:00:00', '2025-03-07 11:45:00'),

(8, 8, 8, 'Out for Delivery', '2025-03-08 15:30:00', NULL),

(9, 9, 9, 'Delivered', '2025-03-09 19:00:00', '2025-03-09 19:40:00'),

(10, 10, 10, 'Pending', NULL, NULL);