

Hackathon Submission (Level-2-Solution)

Use Case Title: Online Food Ordering .

Student Name: N.SRIBALA

Register Number: U22CSE32255.

Institution: Sri Meenakshi Govt Arts college for Women(Autonomous), Madurai.

Department: Computer Science.

1. Problem Statement

Traditional food ordering systems rely on phone calls or in-person visits, leading to inefficiencies such as:

- Long wait times for order processing.
- Human errors in order taking.
- Lack of real-time tracking for customers.
- Difficulty in managing restaurant operations (inventory, orders, payments).

An Online Food Ordering System solves these issues by:

- ❖ Allowing customers to browse menus, place orders, and pay online.
- ❖ Enabling restaurants to manage orders efficiently.
- ❖ Providing real-time order status updates.
- ❖ Automating inventory and sales tracking

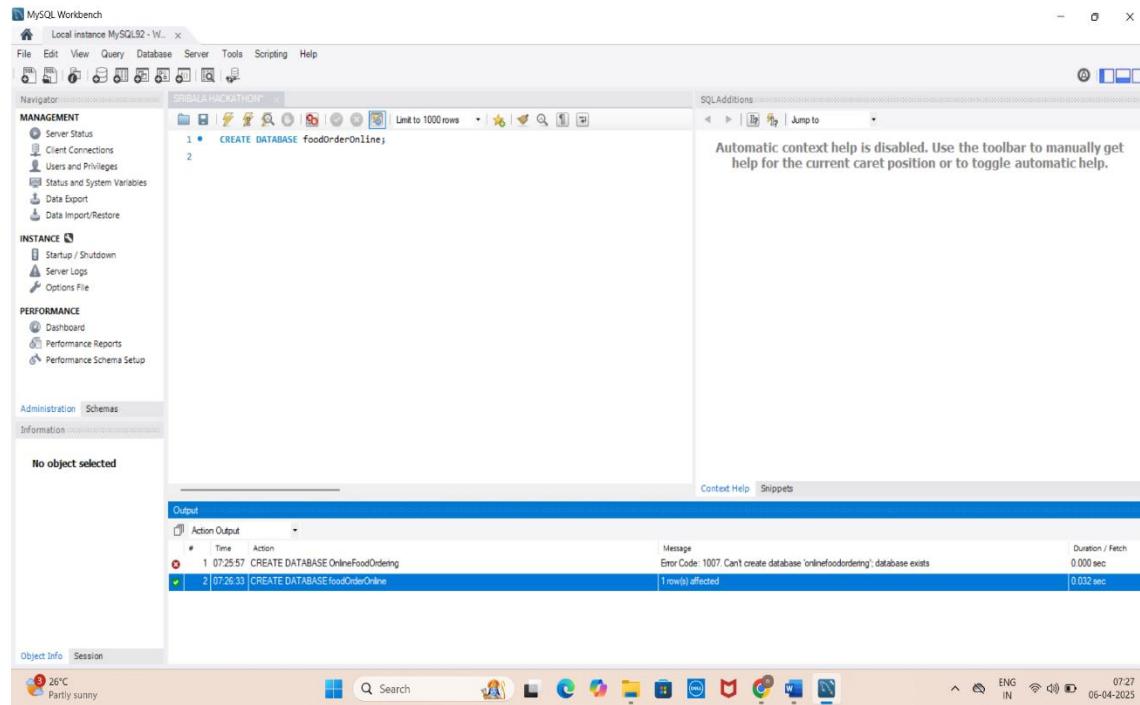
2.DATABASE DESIGN & IMPLEMENTATION

Database

Creation & Table

Database Creation

```
CREATE DATABASE foodOrderOnline;
```



Tables Creation

```
USE foodOrderOnline;
```

RESTAURANT TABLE

```
CREATE TABLE Restaurants (
    RestaurantID INT PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR(100) NOT NULL,
    Description TEXT,
    Location VARCHAR(150),
    ContactNumber VARCHAR(20),
    Email VARCHAR(100) UNIQUE,
    OpeningHours VARCHAR(100),
    Rating DECIMAL(2,1) DEFAULT 0.0,
    IsActive BOOLEAN DEFAULT TRUE,
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

MENUITEMS TABLE

```
CREATE TABLE MenuItems (
    ItemID INT PRIMARY KEY AUTO_INCREMENT,
    RestaurantID INT NOT NULL,
    Name VARCHAR(100) NOT NULL,
    Description TEXT,
    Price DECIMAL(10,2) NOT NULL CHECK (Price >= 0),
    Category VARCHAR(50),
    IsVegetarian BOOLEAN DEFAULT FALSE,
    Availability BOOLEAN DEFAULT TRUE,
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID) ON DELETE
    CASCADE
);
```

CUSTOMERS TABLE

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    FullName VARCHAR(100) NOT NULL,
    Email VARCHAR(100) UNIQUE,
    PhoneNumber VARCHAR(20),
    AddressLine1 VARCHAR(150),
    AddressLine2 VARCHAR(150),
    City VARCHAR(50),
    State VARCHAR(50),
    PostalCode VARCHAR(10),
    CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP
)
```

ORDERS TABLE

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY AUTO_INCREMENT,
    CustomerID INT NOT NULL,
    RestaurantID INT NOT NULL,
```

```

OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
EstimatedDeliveryTime DATETIME,
DeliveryMethod ENUM('Pickup', 'Delivery') DEFAULT 'Delivery',
Status ENUM('Pending', 'Confirmed', 'Preparing', 'Ready', 'Out for Delivery', 'Delivered',
'Cancelled') DEFAULT 'Pending',
PaymentMethod ENUM('Cash', 'Card', 'UPI', 'Wallet') DEFAULT 'Cash',
SubTotal DECIMAL(10,2) NOT NULL CHECK (SubTotal >= 0),
TaxAmount DECIMAL(10,2) DEFAULT 0.00 CHECK (TaxAmount >= 0),
DeliveryFee DECIMAL(10,2) DEFAULT 0.00 CHECK (DeliveryFee >= 0),
TotalAmount DECIMAL(10,2) NOT NULL CHECK (TotalAmount >= 0),
DeliveryAddress TEXT,
Notes TEXT,
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE
CASCADE,
FOREIGN KEY (RestaurantID) REFERENCES Restaurants(RestaurantID) ON DELETE
CASCADE
)

```

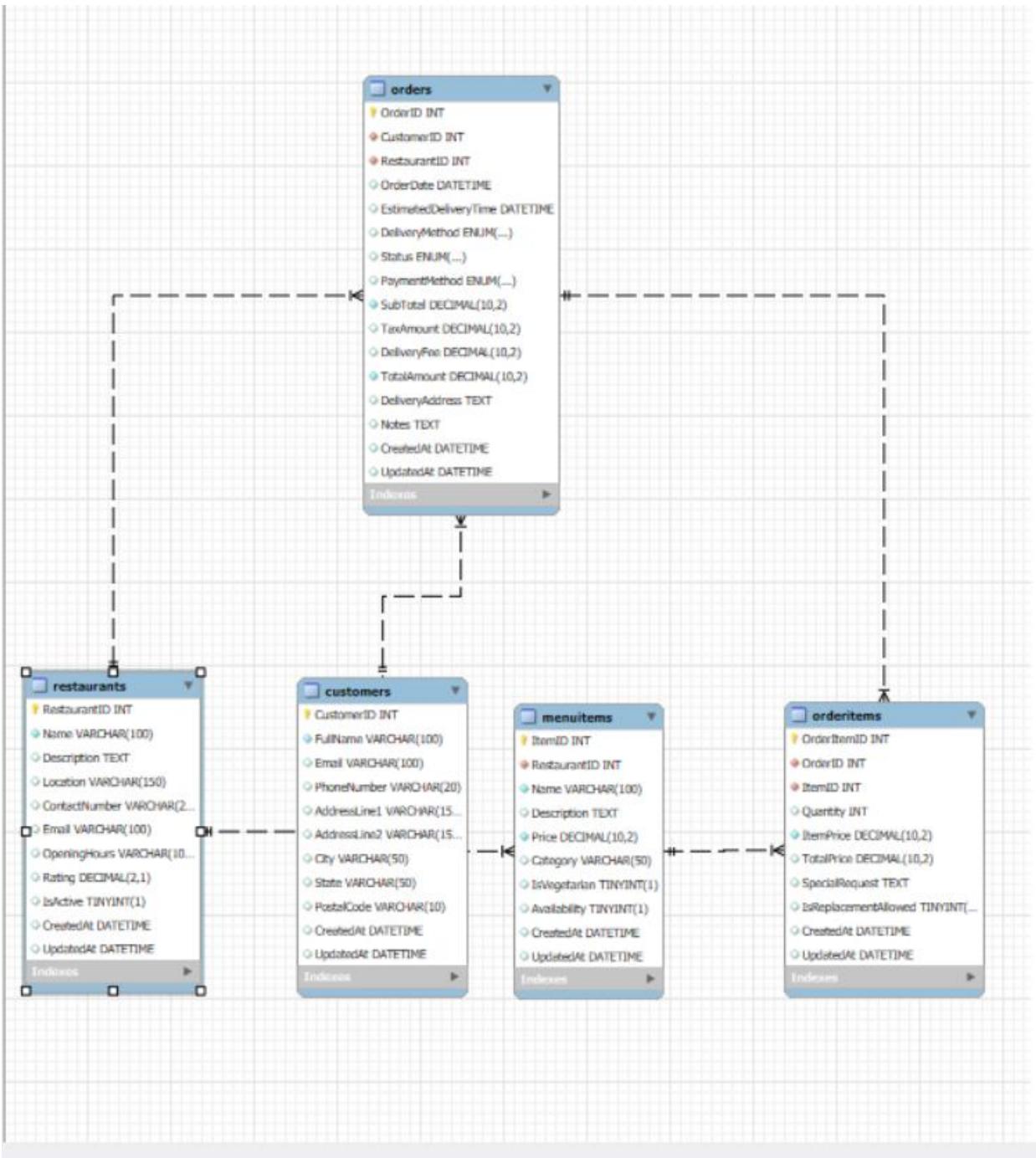
ORDER ITEMS TABLE

```

CREATE TABLE OrderItems (
OrderItemID INT PRIMARY KEY AUTO_INCREMENT,
OrderID INT NOT NULL,
ItemID INT NOT NULL,
Quantity INT DEFAULT 1 CHECK (Quantity > 0),
ItemPrice DECIMAL(10,2) NOT NULL CHECK (ItemPrice >= 0), -- Price at the time of
order
TotalPrice DECIMAL(10,2) GENERATED ALWAYS AS (Quantity * ItemPrice) STORED, -
Calculated
SpecialRequest TEXT,
IsReplacementAllowed BOOLEAN DEFAULT FALSE,
CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
UpdatedAt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE CASCADE,
FOREIGN KEY (ItemID) REFERENCES MenuItems(ItemID) ON DELETE CASCADE
);

```

ER Diagram (Reverse Engineered)



3.QUERIES FOR DATA MANAGEMENT

3.1 Insert Sample Data

1.REATAURANT TABLE

```
INSERT INTO Restaurants (Name, Description, Location, ContactNumber, Email, OpeningHours, Rating)
```

```
VALUES
```

```
('Spicy Garden', 'Authentic Indian Cuisine', 'Hyderabad', '9876543210', 'spicy@garden.com', '10:00 AM - 10:00 PM', 4.5),
```

```
('Pizza World', 'Delicious Italian Pizzas', 'Mumbai', '9123456780', 'contact@pizzaworld.com', '11:00 AM - 11:00 PM', 4.2),
```

```
('Burger House', 'Best Burgers in Town', 'Delhi', '9988776655', 'hello@burgerhouse.com', '09:00 AM - 09:00 PM', 4.0),
```

```
('Vegan Delight', '100% Plant-Based Meals', 'Bangalore', '9001122334', 'info@vegandelight.com', '10:30 AM - 08:00 PM', 4.7),
```

```
("Tandoori Flames", "Tandoori Specials & Kebabs", "Chennai", "9345678901", "support@tandooriflames.com", '12:00 PM - 12:00 AM', 4.3);
```

2.MENU ITEMS TABLE

```
INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian)
```

```
VALUES
```

```
(1, 'Paneer Butter Masala', 'Creamy cottage cheese curry', 220.00, 'Main Course', TRUE),
```

```
(1, 'Chicken Biryani', 'Spicy Hyderabadi biryani', 250.00, 'Main Course', FALSE),
```

```
(2, 'Margherita Pizza', 'Classic cheese pizza', 180.00, 'Pizza', TRUE),
```

```
(3, 'Double Patty Burger', 'Beef burger with double patties', 200.00, 'Burgers', FALSE),
```

```
(4, 'Vegan Bowl', 'Quinoa, tofu and veggies', 210.00, 'Healthy', TRUE);
```

4 CUSTOMERS TABLE

```
INSERT INTO Customers (FullName, Email, PhoneNumber, AddressLine1, AddressLine2, City, State, PostalCode)
```

```
VALUES
```

```
('Aman Verma', 'aman@example.com', '9898989898', '123 Street A', 'Near Market', 'Delhi', 'Delhi', '110001'),
```

```
('Sneha Rao', 'sneha@example.com', '9777788888', '456 Avenue B', NULL, 'Mumbai', 'Maharashtra', '400001'),
```

```
('Karthik Reddy', 'karthik@example.com', '9666677777', '789 Road C', 'Apt 101',  
'Hyderabad', 'Telangana', '500001'),  
(('Neha Sharma', 'neha@example.com', '9555566666', '321 Colony D', NULL,  
'Bangalore', 'Karnataka', '560001'),  
(('Ravi Kumar', 'ravi@example.com', '9444455555', '654 Lane E', 'Opp School',  
'Chennai', 'Tamil Nadu', '600001');
```

5.ORDER TABLE

```
INSERT INTO Orders (CustomerID, RestaurantID, EstimatedDeliveryTime, DeliveryMethod,  
Status, PaymentMethod, SubTotal, TaxAmount, DeliveryFee, TotalAmount, DeliveryAddress,  
Notes)  
VALUES  
(1, 1, NOW() + INTERVAL 45 MINUTE, 'Delivery', 'Preparing', 'UPI', 470.00, 47.00, 30.00,  
547.00, '123 Street A, Delhi', 'Extra spicy'),  
(2, 2, NOW() + INTERVAL 30 MINUTE, 'Pickup', 'Confirmed', 'Card', 180.00, 18.00, 0.00,  
198.00, NULL, NULL),  
(3, 3, NOW() + INTERVAL 50 MINUTE, 'Delivery', 'Pending', 'Cash', 200.00, 20.00, 20.00,  
240.00, '789 Road C, Hyderabad', 'No onions'),  
(4, 4, NOW() + INTERVAL 25 MINUTE, 'Pickup', 'Ready', 'Wallet', 210.00, 21.00, 0.00, 231.00,  
NULL, 'Pack it tight'),  
(5, 5, NOW() + INTERVAL 60 MINUTE, 'Delivery', 'Out for Delivery', 'Card', 250.00, 25.00,  
30.00, 305.00, '654 Lane E, Chennai', NULL);
```

6.ORDER ITEMS TABLE

```
INSERT INTO OrderItems (OrderID, ItemID, Quantity, ItemPrice, SpecialRequest,  
IsReplacementAllowed)  
VALUES  
(1, 1, 1, 220.00, 'Extra gravy', FALSE),  
(1, 2, 1, 250.00, 'Less spicy', TRUE),  
(2, 3, 1, 180.00, NULL, FALSE),  
(3, 4, 1, 200.00, 'No mayo', TRUE),  
(4, 5, 1, 210.00, 'No tofu', FALSE);
```

3.2 RETRIEVAL QUERIES

1. BASIC: GET ALL ORDERS WITH STATUS PENDING

```
SELECT  
OrderID,  
CustomerID,
```

```
RestaurantID,  
OrderDate,  
Status,  
TotalAmount  
FROM Orders  
WHERE Status = 'Pending';
```

2. DETAILED: JOIN WITH CUSTOMERS AND REATAURANTS

```
SELECT  
o.OrderID,  
o.OrderDate,  
o.Status,  
o.TotalAmount,  
c.FullName AS CustomerName,  
c.PhoneNumber,  
r.Name AS RestaurantName,  
r.Location,  
o.DeliveryMethod,  
o.EstimatedDeliveryTime  
FROM Orders o  
JOIN Customers c ON o.CustomerID = c.CustomerID  
JOIN Restaurants r ON o.RestaurantID = r.RestaurantID  
WHERE o.Status = 'Pending'  
ORDER BY o.OrderDate DESC;
```

3.OPTIONAL: FILTER BY TODAY PENDING ORDER ONLY

```
SELECT  
o.OrderID,  
c.FullName AS Customer,  
r.Name AS Restaurant,  
o.OrderDate,  
o.EstimatedDeliveryTime,  
o.TotalAmount  
FROM Orders o  
JOIN Customers c ON o.CustomerID = c.CustomerID  
JOIN Restaurants r ON o.RestaurantID = r.RestaurantID  
WHERE o.Status = 'Pending'  
AND DATE(o.OrderDate) = CURDATE();
```

4.PENDING ORDERS QUERY(WITH CUSTOMERS AND RESTAURANTS INFO)

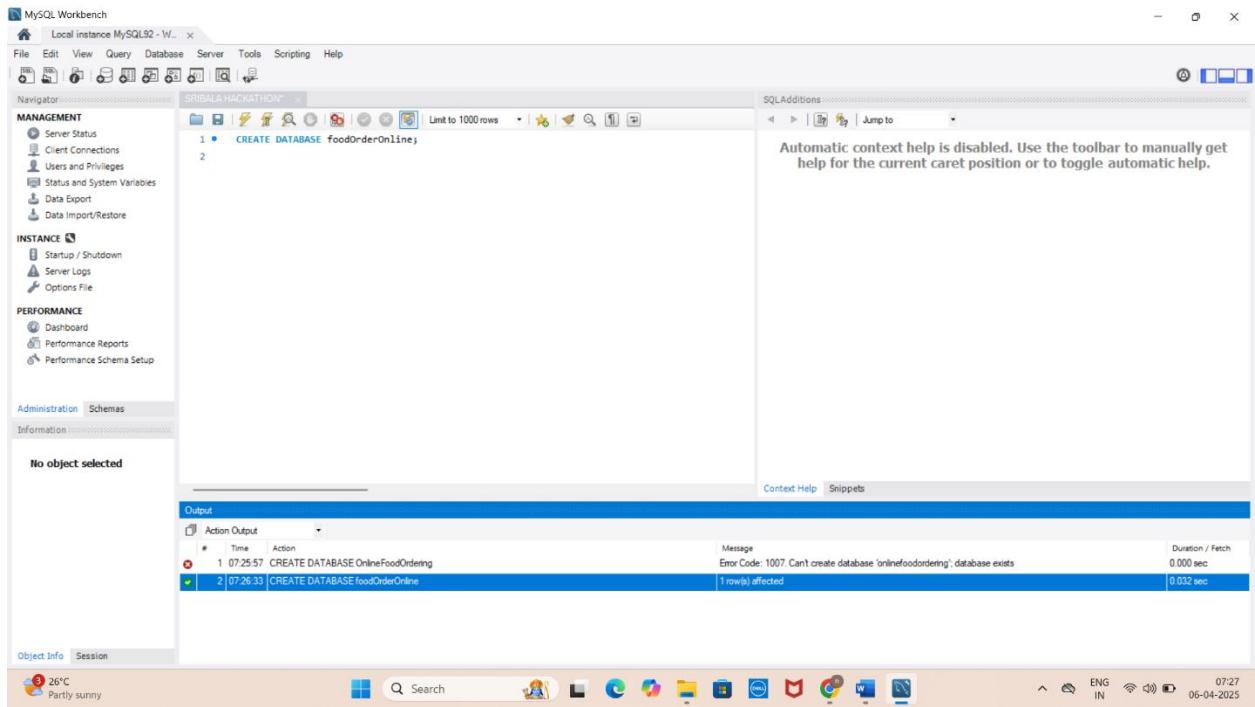
```
SELECT o.OrderID,o.OrderDate,o.EstimatedDeliveryTime,  
o.Status,o.TotalAmount,o.PaymentMethod,o.DeliveryMethod,  
o.Notes,c.FullName AS CustomerName,  
c.PhoneNumber,c.City,r.Name AS RestaurantName,  
r.Location AS RestaurantLocation  
FROM Orders oJOIN Customers c ON o.CustomerID = c.CustomerID  
JOIN Restaurants r ON o.RestaurantID = r.RestaurantID  
WHERE o.Status = 'Pending'  
ORDER BY o.OrderDate DESC;
```

4.IMPLEMENTATION AND RESULT

4.1Execution Environment

- ★ MySQL Workbench (Version 9.2)

1.TABLE CREATED



Output					
Action Output			Message	Duration / Fetch	
1	19:13:23	USE FoodOrder	0 row(s) affected	0.000 sec	
2	19:13:23	CREATE TABLE Restaurants (RestaurantID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(100) N...	0 row(s) affected	0.031 sec	
3	19:13:23	CREATE TABLE MenuItems (ItemID INT PRIMARY KEY AUTO_INCREMENT, RestaurantID INT, Name VARC...	0 row(s) affected	0.031 sec	
4	19:13:24	CREATE TABLE Customers (CustomerID INT PRIMARY KEY AUTO_INCREMENT, Name VARCHAR(50) NOT ...	0 row(s) affected	0.031 sec	
5	19:13:24	CREATE TABLE Orders (OrderID INT PRIMARY KEY AUTO_INCREMENT, CustomerID INT, RestaurantID INT,...	0 row(s) affected	0.047 sec	
6	19:13:24	CREATE TABLE OrderItems (OrderItemID INT PRIMARY KEY AUTO_INCREMENT, OrderID INT, ItemID INT, Q...	0 row(s) affected	0.063 sec	

2.SAMPLE INSERTION DATA

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. The 'ORDERS' schema is currently active. The 'Tables' section shows the 'orderitems' table with its columns: OrderItemID, OrderID, ItemID, Quantity, ItemPrice, TotalPrice, SpecialRequest, IsReplacementAllowed, CreatedAt, and UpdatedAt.

The 'SQL' tab displays the following SQL code:

```

USE foodOrderOnline;
INSERT INTO Restaurants (Name, Description, Location, ContactNumber, Email, OpeningHours, Rating)
VALUES
('Spicy Garden', 'Authentic Indian Cuisine', 'Hyderabad', '9876543210', 'spicy@garden.com', '11:00-22:00', 4.5),
('Pizza World', 'Delicious Italian Pizzas', 'Mumbai', '9123456780', 'contact@pizzaworld.com', '12:00-21:00', 4.8),
('Burger House', 'Best Burgers in Town', 'Delhi', '9988776655', 'hello@burgerhouse.com', '10:00-20:00', 4.2),
('Vegan Delight', '100% Plant-Based Meals', 'Bangalore', '9801233454', 'info@vegandelight.com', '11:00-19:00', 4.9),
('Tandoori Flames', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com', '12:00-21:00', 4.6);

INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian)
VALUES
(1, 'Paneer Butter Masala', 'Creamy cottage cheese curry', 220.00, 'Main Course', TRUE),
(1, 'Chicken Biryani', 'Spicy Hyderabadi biryani', 250.00, 'Main Course', FALSE),
(2, 'Margherita Pizza', 'Classic cheese pizza', 180.00, 'Pizza', TRUE),
(2, 'Double Patty Burger', 'Beef burger with double patties', 200.00, 'Burgers', FALSE),
(3, 'Vegan Bowl', 'Quinoa, tofu and veggies', 210.00, 'Healthy', TRUE),
(4, 'Tandoori Chores', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com', '12:00-21:00', 4.6);

INSERT INTO Customers (FullName, Email, PhoneNumber, AddressLine1, AddressLine2, City, State, PostalCode)
VALUES
('Aman Verma', 'aman@example.com', '9898989898', '123 Street A', 'Near Market', 'Delhi', 'India', '110001'),
('Sneha Rao', 'sneha@example.com', '9777708888', '456 Avenue B', NULL, 'Mumbai', 'Maharashtra', '400001'),
('Karthik Reddy', 'karthik@example.com', '9666677777', '789 Road C', 'Apt 101', 'Hyderabad', 'Andhra Pradesh', '500001');

INSERT INTO Orders (CustomerID, RestaurantID, EstimatedDeliveryTime, DeliveryMethod, Status, PaymentMethod, IsCancelled)
VALUES
(1, 1, '2023-04-01 12:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(2, 2, '2023-04-01 13:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(3, 3, '2023-04-01 14:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(4, 4, '2023-04-01 15:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(5, 5, '2023-04-01 16:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(6, 6, '2023-04-01 17:00:00', 'Delivery', 'Placed', 'Credit Card', 0);

INSERT INTO OrderItems (OrderID, ItemID, Quantity, ItemPrice, TotalPrice, SpecialRequest, IsReplacementAllowed)
VALUES
(1, 1, 2, 220.00, 440.00, 'No', 0),
(1, 2, 1, 250.00, 250.00, 'Yes', 1),
(2, 3, 1, 180.00, 180.00, 'No', 0),
(2, 4, 1, 200.00, 200.00, 'Yes', 1),
(3, 5, 1, 210.00, 210.00, 'No', 0),
(3, 6, 1, 210.00, 210.00, 'Yes', 1),
(4, 7, 1, 220.00, 220.00, 'No', 0),
(4, 8, 1, 250.00, 250.00, 'Yes', 1),
(5, 9, 1, 180.00, 180.00, 'No', 0),
(5, 10, 1, 200.00, 200.00, 'Yes', 1),
(6, 11, 1, 210.00, 210.00, 'No', 0),
(6, 12, 1, 210.00, 210.00, 'Yes', 1);

```

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. The 'ORDERS' schema is currently active. The 'Tables' section shows the 'restaurants' table with its columns: RestaurantID, Name, Description, Location, ContactNumber, Email, OpeningHours, Rating, Active, and UpdatedAt.

The 'SQL' tab displays the following SQL code:

```

USE foodOrderOnline;
INSERT INTO Restaurants (Name, Description, Location, ContactNumber, Email, OpeningHours, Rating)
VALUES
('Spicy Garden', 'Authentic Indian Cuisine', 'Hyderabad', '9876543210', 'spicy@garden.com', '11:00-22:00', 4.5),
('Pizza World', 'Delicious Italian Pizzas', 'Mumbai', '9123456780', 'contact@pizzaworld.com', '12:00-21:00', 4.8),
('Burger House', 'Best Burgers in Town', 'Delhi', '9988776655', 'hello@burgerhouse.com', '10:00-20:00', 4.2),
('Vegan Delight', '100% Plant-Based Meals', 'Bangalore', '9801233454', 'info@vegandelight.com', '11:00-19:00', 4.9),
('Tandoori Flames', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com', '12:00-21:00', 4.6);

INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian)
VALUES
(1, 'Paneer Butter Masala', 'Creamy cottage cheese curry', 220.00, 'Main Course', TRUE),
(1, 'Chicken Biryani', 'Spicy Hyderabadi biryani', 250.00, 'Main Course', FALSE),
(2, 'Margherita Pizza', 'Classic cheese pizza', 180.00, 'Pizza', TRUE),
(2, 'Double Patty Burger', 'Beef burger with double patties', 200.00, 'Burgers', FALSE),
(3, 'Vegan Bowl', 'Quinoa, tofu and veggies', 210.00, 'Healthy', TRUE),
(4, 'Tandoori Chores', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com', '12:00-21:00', 4.6);

INSERT INTO Customers (FullName, Email, PhoneNumber, AddressLine1, AddressLine2, City, State, PostalCode)
VALUES
('Aman Verma', 'aman@example.com', '9898989898', '123 Street A', 'Near Market', 'Delhi', 'India', '110001'),
('Sneha Rao', 'sneha@example.com', '9777708888', '456 Avenue B', NULL, 'Mumbai', 'Maharashtra', '400001'),
('Karthik Reddy', 'karthik@example.com', '9666677777', '789 Road C', 'Apt 101', 'Hyderabad', 'Andhra Pradesh', '500001');

INSERT INTO Orders (CustomerID, RestaurantID, EstimatedDeliveryTime, DeliveryMethod, Status, PaymentMethod, IsCancelled)
VALUES
(1, 1, '2023-04-01 12:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(2, 2, '2023-04-01 13:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(3, 3, '2023-04-01 14:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(4, 4, '2023-04-01 15:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(5, 5, '2023-04-01 16:00:00', 'Delivery', 'Placed', 'Credit Card', 0),
(6, 6, '2023-04-01 17:00:00', 'Delivery', 'Placed', 'Credit Card', 0);

INSERT INTO OrderItems (OrderID, ItemID, Quantity, ItemPrice, TotalPrice, SpecialRequest, IsReplacementAllowed)
VALUES
(1, 1, 2, 220.00, 440.00, 'No', 0),
(1, 2, 1, 250.00, 250.00, 'Yes', 1),
(2, 3, 1, 180.00, 180.00, 'No', 0),
(2, 4, 1, 200.00, 200.00, 'Yes', 1),
(3, 5, 1, 210.00, 210.00, 'No', 0),
(3, 6, 1, 210.00, 210.00, 'Yes', 1),
(4, 7, 1, 220.00, 220.00, 'No', 0),
(4, 8, 1, 250.00, 250.00, 'Yes', 1),
(5, 9, 1, 180.00, 180.00, 'No', 0),
(5, 10, 1, 200.00, 200.00, 'Yes', 1),
(6, 11, 1, 210.00, 210.00, 'No', 0),
(6, 12, 1, 210.00, 210.00, 'Yes', 1);

```

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Filter objects

fooderonline

- Tables
 - customers
 - menuitems
- Columns
- Indexes
- Foreign Keys
- Triggers

Administration Schemas Information

Table: menuitems

Columns:

ItemID	int AI PK
RestaurantID	int
Name	varchar(1)
Description	text
Price	decimal(10,2)
Category	varchar(5)
IsVegetarian	tinyint(1)
Availability	tinyint(1)
CreatedAt	datetime
UpdatedAt	datetime

Object Info Session

Output

Action Output

Time	Action	Message	Duration / Fetch
1 11:38:07	USE foodOrderOnline	0 rows affected	0.000 sec
2 11:42:07	INSERT INTO Restaurants (Name, Description, Location, ContactNumber, Email, OpeningHours, Rating) VALUES ('Spicy Garden', 'Authentic Indian Cuisine', 'Hyderabad', '9876543210', 'spicy@garden.com', 'Pizza World', 'Delicious Italian Pizzas', 'Mumbai', '9123456780', 'contact@pizzaworld.com', 'Burger House', 'Best Burgers in Town', 'Delhi', '9988776655', 'hello@burgerhouse.com', 'Vegan Delight', '100% Plant-Based Meals', 'Bangalore', '9801223344', 'info@vegandelight.com', 'Tandoori Flames', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com')	5 rows affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
3 11:42:59	INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian) VALUES (1, 'Paneer Butter Masala', 'Creamy cottage cheese curry', 220.00, 'Main Course', TRUE), (1, 'Chicken Biryani', 'Spicy Hyderabadi biryani', 250.00, 'Main Course', FALSE), (2, 'Margherita Pizza', 'Classic cheese pizza', 180.00, 'Pizza', TRUE), (3, 'Double Patty Burger', 'Beef burger with double patties', 200.00, 'Burgers', FALSE), (4, 'Vegan Bowl', 'Quinoa, tofu and veggies', 210.00, 'Healthy', TRUE);	5 rows affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec

Context Help Snippets

32°C Mostly sunny

Search

11:43 ENG IN 06-04-2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Filter objects

fooderonline

- Tables
 - customers
 - menuitems
- Columns
- Indexes
- Foreign Keys
- Triggers

Administration Schemas Information

Table: customers

Columns:

CustomerID	int AI PK
FullName	varchar(1)
Email	varchar(1)
PhoneNumber	varchar(1)
AddressLine1	varchar(1)
AddressLine2	varchar(1)
City	varchar(5)
State	varchar(5)
PostalCode	varchar(1)
CreatedAt	datetime
UpdatedAt	datetime

Object Info Session

Output

Action Output

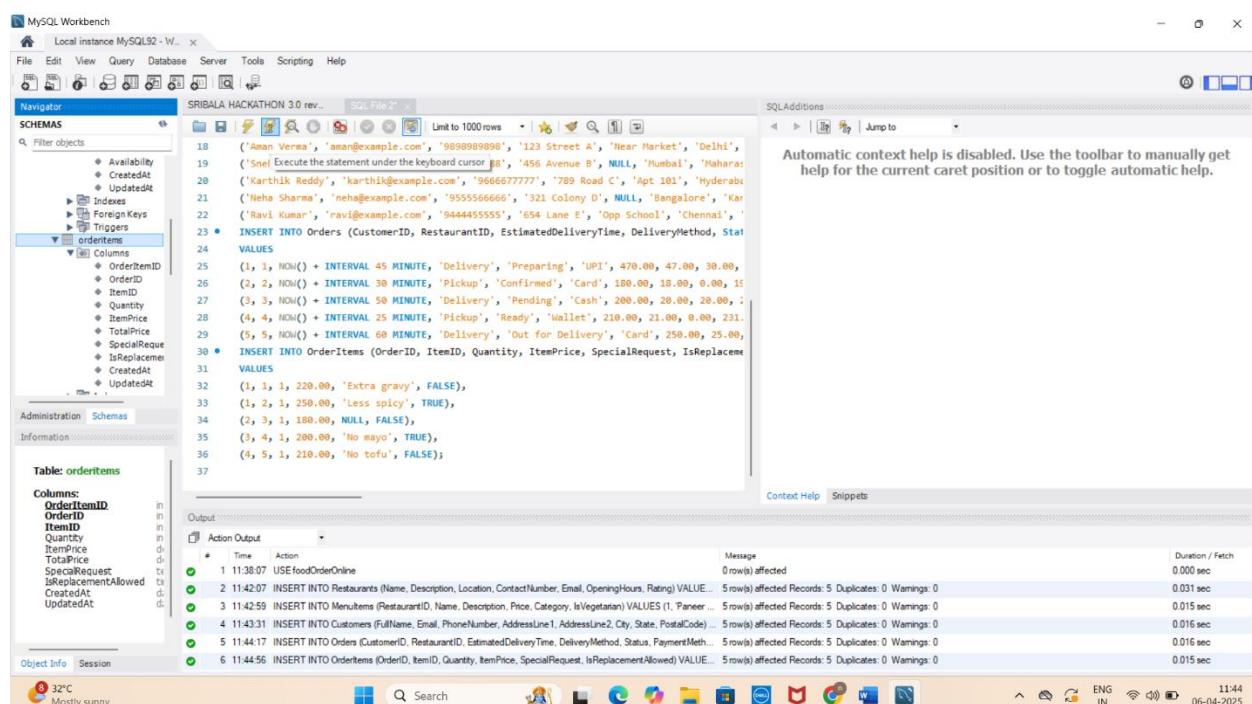
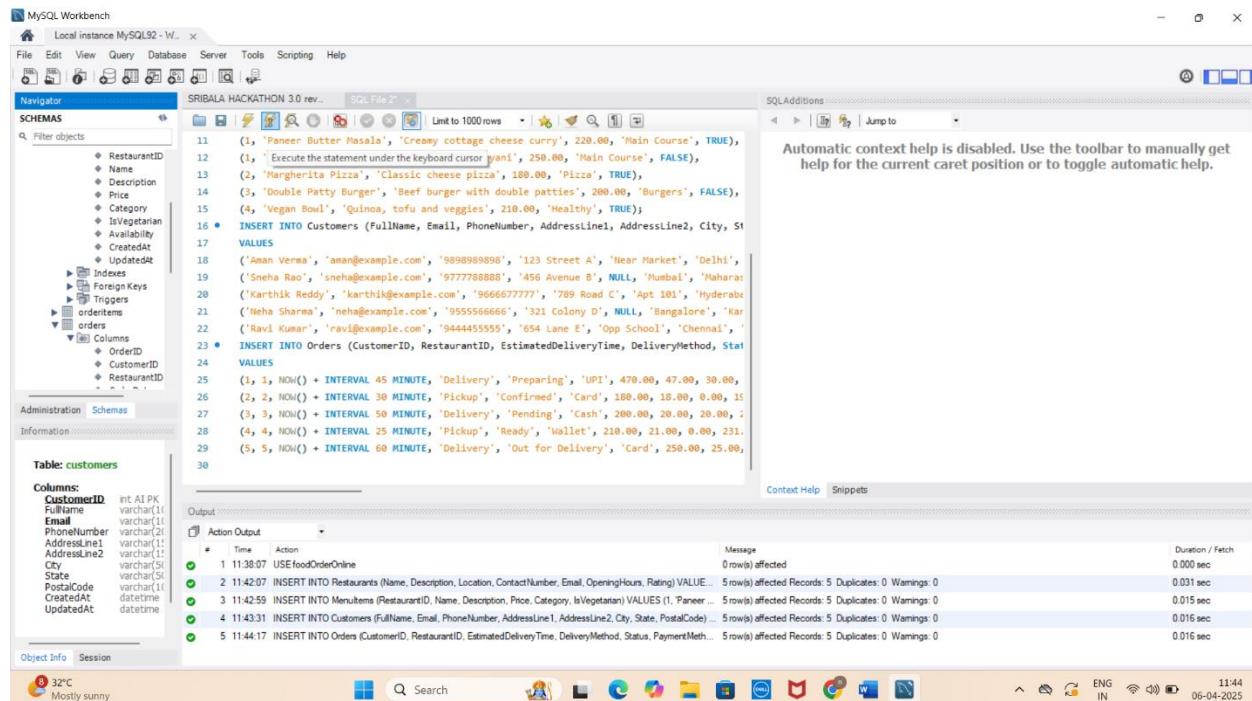
Time	Action	Message	Duration / Fetch
1 11:38:07	USE foodOrderOnline	0 rows affected	0.000 sec
2 11:42:07	INSERT INTO Restaurants (Name, Description, Location, ContactNumber, Email, OpeningHours, Rating) VALUES ('Spicy Garden', 'Authentic Indian Cuisine', 'Hyderabad', '9876543210', 'spicy@garden.com', 'Pizza World', 'Delicious Italian Pizzas', 'Mumbai', '9123456780', 'contact@pizzaworld.com', 'Burger House', 'Best Burgers in Town', 'Delhi', '9988776655', 'hello@burgerhouse.com', 'Vegan Delight', '100% Plant-Based Meals', 'Bangalore', '9801223344', 'info@vegandelight.com', 'Tandoori Flames', 'Tandoori Specials & Kebabs', 'Chennai', '9345678901', 'support@tandooriflames.com')	5 rows affected Records: 5 Duplicates: 0 Warnings: 0	0.031 sec
3 11:42:59	INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian) VALUES (1, 'Paneer Butter Masala', 'Creamy cottage cheese curry', 220.00, 'Main Course', TRUE), (1, 'Chicken Biryani', 'Spicy Hyderabadi biryani', 250.00, 'Main Course', FALSE), (2, 'Margherita Pizza', 'Classic cheese pizza', 180.00, 'Pizza', TRUE), (3, 'Double Patty Burger', 'Beef burger with double patties', 200.00, 'Burgers', FALSE), (4, 'Vegan Bowl', 'Quinoa, tofu and veggies', 210.00, 'Healthy', TRUE);	5 rows affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
4 11:43:31	INSERT INTO Customers (FullName, Email, PhoneNumber, AddressLine1, AddressLine2, City, State, PostalCode) VALUES ('Aman Verma', 'aman@example.com', '9898989898', '123 Street A', 'Near Market', 'Delhi', 'Sneha Rao', 'sneha@example.com', '9777788888', '456 Avenue B', NULL, 'Mumbai', 'Maharashtra', 'Karthik Reddy', 'karthik@example.com', '9666677777', '89 Road C', 'Apartment 101', 'Hyderabad', 'Reha Sharma', 'reha@example.com', '9555566666', '321 Colony D', NULL, 'Bangalore', 'Ravi Kumar', 'ravi@example.com', '9444455555', '654 Lane E', 'Opp School', 'Chennai', 'Vidya Patel', 'vidya@example.com', '9333344444', '789 Main St', NULL, 'Chennai')	5 rows affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec

Context Help Snippets

32°C Mostly sunny

Search

11:43 ENG IN 06-04-2025



3.RETRIEVAL QUERIES

The screenshot shows the MySQL Workbench interface with a query editor window titled "SIRIBALA HACKATHON 3.0 rev...". The query is:

```
12    o.EstimatedDeliveryTime
13  FROM Orders o
14  JOIN Customers c ON o.CustomerID = c.CustomerID
15  JOIN Restaurants r ON o.RestaurantID = r.RestaurantID
16  WHERE o.Status = 'Pending'
17  ORDER BY o.OrderDate DESC
18
```

The results grid shows one row of data:

OrderID	OrderDate	Status	TotalAmount	CustomerName	PhoneNumber	RestaurantName	Location
3	2025-04-06 11:44:17	Pending	240.00	Karthik Reddy	9666677777	Burger House	Delhi

The "Action Output" pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
5	11:44:17	INSERT INTO Orders (CustomerID, RestaurantID, EstimatedDeliveryTime, DeliveryMethod, Status, PaymentMethod, TotalAmount)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
6	11:44:56	INSERT INTO OrderItems (OrderID, ItemID, Quantity, ItemPrice, SpecialRequest, IsReplacementAllowed) VALUES (1, 1, 1, 100, 0, 0)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
7	12:07:33	USE foodOrderOnline	0 row(s) affected	0.000 sec
8	12:11:16	SELECT OrderID, CustomerID, RestaurantID, OrderDate, Status, TotalAmount FROM Orders WHERE Status = 'Pending'	1 row(s) returned	0.000 sec / 0.000 sec
9	12:15:28	USE foodOrderOnline	0 row(s) affected	0.000 sec
10	12:16:03	SELECT o.OrderID, o.OrderDate, o.Status, o.TotalAmount, c.FullName AS CustomerName, c.PhoneNumber AS PhoneNumber, r.RestaurantName, r.Location	1 row(s) returned	0.016 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface with a query editor window titled "SIRIBALA HACKATHON 3.0 rev...". The query is:

```
5  RestaurantID,
6  OrderDate,
7  Status,
8  TotalAmount
9  FROM Orders
10 WHERE Status = 'Pending';
11
```

The results grid shows one row of data:

OrderID	CustomerID	RestaurantID	OrderDate	Status	TotalAmount
3	3	3	2025-04-06 11:44:17	Pending	240.00

The "Action Output" pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
3	11:42:59	INSERT INTO MenuItems (RestaurantID, Name, Description, Price, Category, IsVegetarian) VALUES (1, 'Pasta', 'A delicious pasta dish.', 100, 'Italian', 1)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
4	11:43:31	INSERT INTO Customers (FullName, Email, PhoneNumber, AddressLine1, AddressLine2, City, State, PostalCode)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
5	11:44:17	INSERT INTO Orders (CustomerID, RestaurantID, EstimatedDeliveryTime, DeliveryMethod, Status, PaymentMethod, TotalAmount)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
6	11:44:56	INSERT INTO OrderItems (OrderID, ItemID, Quantity, ItemPrice, SpecialRequest, IsReplacementAllowed) VALUES (1, 1, 1, 100, 0, 0)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec
7	12:07:33	USE foodOrderOnline	0 row(s) affected	0.000 sec
8	12:11:16	SELECT OrderID, CustomerID, RestaurantID, OrderDate, Status, TotalAmount FROM Orders WHERE Status = 'Pending'	1 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

IBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SQL File 6*

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

1 USE foodOrderOnline;
2 
3     o.TotalAmount
4     FROM Orders o
5     JOIN Customers c ON o.CustomerID = c.CustomerID
6     JOIN Restaurants r ON o.RestaurantID = r.RestaurantID
7     WHERE o.Status = 'Pending'
8     AND DATE(o.OrderDate) = CURDATE();
9 
10    3
11    Karthik Reddy Burger House 2025-04-06 11:44:17 2025-04-06 12:34:17 240.00

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

Result 1 x Output Action Output

OrderID	Customer	Restaurant	OrderDate	EstimatedDeliveryTime	TotalAmount
3	Karthik Reddy	Burger House	2025-04-06 11:44:17	2025-04-06 12:34:17	240.00

Object Info Session

34°C Mostly sunny

Search ENG IN 12:18 06-04-2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

IBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SQL File 7*

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

1 USE foodOrderOnline;
2 
3     SELECT
4         o.OrderID,
5         o.OrderDate,
6         o.EstimatedDeliveryTime,
7         o.Status,
8         o.TotalAmount,
9         o.PaymentMethod,
10        o.DeliveryMethod,
11        o.Notes,
12        c.FullName AS CustomerName,
13        c.PhoneNumber,
14        c.City,
15        r.Name AS RestaurantName,
16        r.Location AS RestaurantLocation
17     FROM Orders o
18     JOIN Customers c ON o.CustomerID = c.CustomerID
19     JOIN Restaurants r ON o.RestaurantID = r.RestaurantID
20     WHERE o.Status = 'Pending'
21     ORDER BY o.OrderDate DESC;

```

Context Help Snippets

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: IBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SQL File 7... SQLAdditions

Filter objects

SCHEMAS

foodorderonline

- Tables
 - customers
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - menuitems
 - Columns

Administration Schemas

Information

Schema: foodorderonline

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

```

15     r.Location AS Restaurantlocation
16   FROM Orders o
17   JOIN Customers c ON o.CustomerID = c.CustomerID
18   JOIN Restaurants r ON o.RestaurantID = r.RestaurantID
19   WHERE o.Status = 'Pending'
20   ORDER BY o.OrderDate DESC;
21

```

OrderID OrderDate EstimatedDeliveryTime Status TotalAmount PaymentMethod DeliveryMethod

3	2025-04-06 11:44:17	2025-04-06 12:34:17	Pending	240.00	Cash	Delivery
---	---------------------	---------------------	---------	--------	------	----------

Form Editor

Field Types

Result 1 x Read Only Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
8	12:11:16	SELECT OrderID, CustomerID, RestaurantID, OrderDate, Status, TotalAmount FROM Orders ...	1 row(s) returned	0.000 sec / 0.000 sec
9	12:15:28	USE foodorderonline	0 row(s) affected	0.000 sec
10	12:16:03	SELECT o.OrderID, o.OrderDate, o.Status, o.TotalAmount, c.FullName AS CustomerName, c.P... 1 row(s) returned		0.016 sec / 0.000 sec
11	12:18:00	USE foodorderonline	0 row(s) affected	0.000 sec
12	12:18:20	SELECT o.OrderID, c.FullName AS Customer, r.Name AS Restaurant, o.OrderDate, o.EstimatedD... 1 row(s) returned		0.016 sec / 0.000 sec
13	12:22:04	SELECT o.OrderID, o.OrderDate, o.EstimatedDeliveryTime, o.Status, o.TotalAmount, o.Payme... 1 row(s) returned		0.016 sec / 0.000 sec

Object Info Session

34°C Mostly sunny

Search

ENG IN 12:22 06-04-2025

4.SELECT THE TABLE OF DATA

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: SRIBALA HACKATHON*

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	07:25:57	CREATE DATABASE foodOrderOnline;	Error Code: 1007: Can't create database 'foodOrderOnline'; database exists	0.000 sec
2	07:26:33	CREATE DATABASE foodOrderOnline	1 row(s) affected	0.032 sec

Context Help Snippets

Object Info Session

26°C Partly sunny

Search

ENG IN 07:27 06-04-2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance MySQL92 - W...

SCHEMAS: foodorderonline

Tables: customers, menuitems, orderdetails, orders, restaurants

Views, Stored Procedures, Functions, sys

BALAHACKTHON 3.0 rev... SIRBALAHACKTHON 3.0 rev... SIRBALAHACKTHON 3.0 rev... SQL File 8*

1 USE foodOrderOnline;

2 SELECT * FROM restaurants;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

RestaurantID	Name	Description	Location	ContactNumber	Email
1	Spicy Garden	Authentic Indian Cuisine	Hyderabad	9876543210	spicy@garden.com
2	Pizza World	Delicious Italian Pizza	Mumbai	91234567890	contact@pizzaworld.com
3	Burger House	Best Burgers in Town	Delhi	9898776655	hello@burgerhouse.com
4	Veggie Delight	100% Plant-Based Meals	Bangalore	9001122334	info@veggiadelight.com
5	Tandoori Flames	Tandoori Specials & Kebabs	Chennai	9345678901	support@tandooriflames.com

No object selected

Output: Action Output | Time Action | Message | Duration / Fetch

1 12:37:34 SELECT * FROM restaurants LIMIT 0, 1000 5 row(s) returned 0.015 sec / 0.000 sec

Object Info Session

Heavy rain Today

12:37 ENG IN 06-04-2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Local instance MySQL92 - W...

SCHEMAS: foodorderonline

Tables: customers, menuitems, orderdetails, orders, restaurants

Views, Stored Procedures, Functions, sys

BALAHACKTHON 3.0 rev... SIRBALAHACKTHON 3.0 rev... SIRBALAHACKTHON 3.0 rev... SQL File 9*

1 USE foodOrderOnline;

2 SELECT * FROM customers;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types

CustomerID	FullName	Email	PhoneNumber	AddressLine1	AddressLine2	City	State
1	Aman Verma	aman@example.com	9898989988	123 Street A	New Market	Delhi	Delhi
2	Sneha Rao	sneha@example.com	9777998888	456 Avenue B	Mark	Mumbai	Maharashtra
3	Karthik Reddy	karthik@example.com	9666677777	789 Road C	Apt 101	Hyderabad	Andhra
4	Hema Sharma	hema@example.com	9555566666	321 Colony D	Opp School	Bangalore	Karnataka
5	Ravi Kumar	ravi@example.com	9444455555	654 Lane E	Mark	Chennai	Tamil Nadu

No object selected

Output: Action Output | Time Action | Message | Duration / Fetch

1 12:37:34 SELECT * FROM restaurants LIMIT 0, 1000 5 row(s) returned 0.015 sec / 0.000 sec

2 12:39:35 SELECT * FROM customers LIMIT 0, 1000 5 row(s) returned 0.016 sec / 0.000 sec

Object Info Session

Hot weather Now

12:39 ENG IN 06-04-2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: foodorderonline (Tables: customers, menuitems, orderitems, orders, restaurants), sakila, sys

SQL Editor: BALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev...

```
1 • USE FoodOrderOnline;
2 • SELECT * FROM orderitems;
```

Result Grid: OrderID, OrderID, ItemID, Quantity, ItemPrice, TotalPrice, SpecialRequest, IsReplacementAllowed, Create...

OrderID	OrderID	ItemID	Quantity	ItemPrice	TotalPrice	SpecialRequest	IsReplacementAllowed	Create...
1	1	1	1	220.00	220.00	Extra gravy	0	2025-0...
2	1	2	1	250.00	250.00	Less spicy	1	2025-0...
3	2	3	1	180.00	180.00		0	2025-0...
4	3	4	1	200.00	200.00	No mayo	1	2025-0...
5	4	5	1	210.00	210.00	No tofu	0	2025-0...

Output: Action Output

#	Time	Action	Message	Duration / Fetch
1	12:37:34	SELECT * FROM restaurants LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
2	12:39:35	SELECT * FROM customers LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec
3	12:41:02	SELECT * FROM menuitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	12:42:42	SELECT * FROM orderitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Feels hotter Now

Search

12:42 ENG IN 06-04-2025

MySQL Workbench

Local instance MySQL92 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

SCHEMAS: foodorderonline (Tables: customers, menuitems, orderitems, orders, restaurants), sakila, sys

SQL Editor: 0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev... SRIBALA HACKATHON 3.0 rev...

```
1 • USE FoodOrderOnline;
2 • SELECT * FROM orders;
```

Result Grid: OrderID, CustomerID, RestaurantID, OrderDate, EstimatedDeliveryTime, DeliveryMethod, Status...

OrderID	CustomerID	RestaurantID	OrderDate	EstimatedDeliveryTime	DeliveryMethod	Status
1	1	1	2025-04-06 11:44:17	2025-04-06 12:29:17	Delivery	Preparing
2	2	2	2025-04-06 11:44:17	2025-04-06 12:14:17	Pickup	Confirmed
3	3	3	2025-04-06 11:44:17	2025-04-06 12:34:17	Delivery	Pending
4	4	4	2025-04-06 11:44:17	2025-04-06 12:09:17	Pickup	Ready
5	5	5	2025-04-06 11:44:17	2025-04-06 12:44:17	Delivery	Out for Delivery

Output: Action Output

#	Time	Action	Message	Duration / Fetch
1	12:37:34	SELECT * FROM restaurants LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
2	12:39:35	SELECT * FROM customers LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec
3	12:41:02	SELECT * FROM menuitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	12:42:42	SELECT * FROM orderitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
5	12:43:40	SELECT * FROM orderitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	12:46:50	SELECT * FROM orders LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

34°C Mostly sunny

Search

12:47 ENG IN 06-04-2025

#	Time	Action	Message	Duration / Fetch
1	12:37:34	SELECT * FROM restaurants LIMIT 0, 1000	5 row(s) returned	0.015 sec / 0.000 sec
2	12:39:35	SELECT * FROM customers LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec
3	12:41:02	SELECT * FROM menuitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	12:42:42	SELECT * FROM orderitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
5	12:43:40	SELECT * FROM orderitems LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	12:46:50	SELECT * FROM orders LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

4.GITHUB REPOSITORY

3.3 Repository Link

<https://github.com/Sribala2005/Online-food-ordering.git>

3.4 Uploaded Files in Repository



Files Included:

- 3.4.1 `database_schema.sql` (SQL scripts for DB creation)
- 3.4.2 `sample_data_inserts.sql` (Sample data insertion queries)
- 3.4.3 `queries.sql` (Retrieval and management queries)
- 3.4.4 `ER_Diagram.png` (Reverse-engineered ER Diagram)
- 3.4.5 `README.md` (Project documentation)

CONCLUSION

This Online Food Ordering System automates restaurant operations, improves customer experience, and reduces manual errors. Future enhancements could include real-time delivery tracking and AI-based recommendations.