INFO 5707

Spring 2023

Optimizing Operations of Restaurant Management System

Submitted by

S.No	Name	ID
1	Devendra Bupathi Rapolu (Team Coordinator)	11639592
2	Raja Diwakar Moyyi	11589070
3	Sasi Kiran Katamneni	11571141
4	Tirumala Aditya Yadagiri	11594277

Introduction

The restaurant sector is booming and continuously developing in today's fast-paced society. The need for an effective and cost-efficient system for managing restaurants has become critical to give consumers the finest eating experience possible. The Restaurant Management System (RMS) is intended to improve the entire customer experience by facilitating the restaurant's everyday activities, such as handling orders, inventory, personnel, and client bookings. The technology will optimize the restaurant's business procedures by simplifying duties and providing an integrated solution.

Objectives

The primary goal of this undertaking is to create a reliable and robust database to run a management system for restaurants System. This project's primary goals are as follows:

1. To develop a user-friendly database that is simple to manage and update.
2. To create a database schema capable of storing any relevant data for efficient and successful restaurant administration.
3. Ensure that the database architecture is scalable to handle future growth and development.
4. Ensuring that the database structure fulfills the necessary performance criteria, such as fast response times and minimal latency.
5. To maintain data security and privacy, adequate authentication and authorization mechanisms must be implemented.

Scope

This project's goal is to create a structure for a database of a system for managing restaurants. The database will hold all essential information about the restaurant's activities, such as customer information, personnel information, inventory, menu items, and orders. This project's scope includes:

1. Development of a theoretical data model to determine the database structure.

2. Development of a logic-based data framework to specify the connections between entities and characteristics.
3. Develop a physical data representation to specify the tables, their columns, and data types.
4. Creating the database schema to assure the integrity, consistency, and correctness of the data.
5. Creating the indexes and restrictions required for effective retrieval of information and storage.
6. Creating the stored procedures and triggers required for carrying out business rules and validations.
7. Putting adequate authentication and permission processes in place to safeguard data integrity and confidentiality.
8. Testing the relational database design to confirm it satisfies the performance requirements.

Project Requirements

OS: Windows / Mac

Database: MYSQL

Applications: MS word and MySQL Workbench

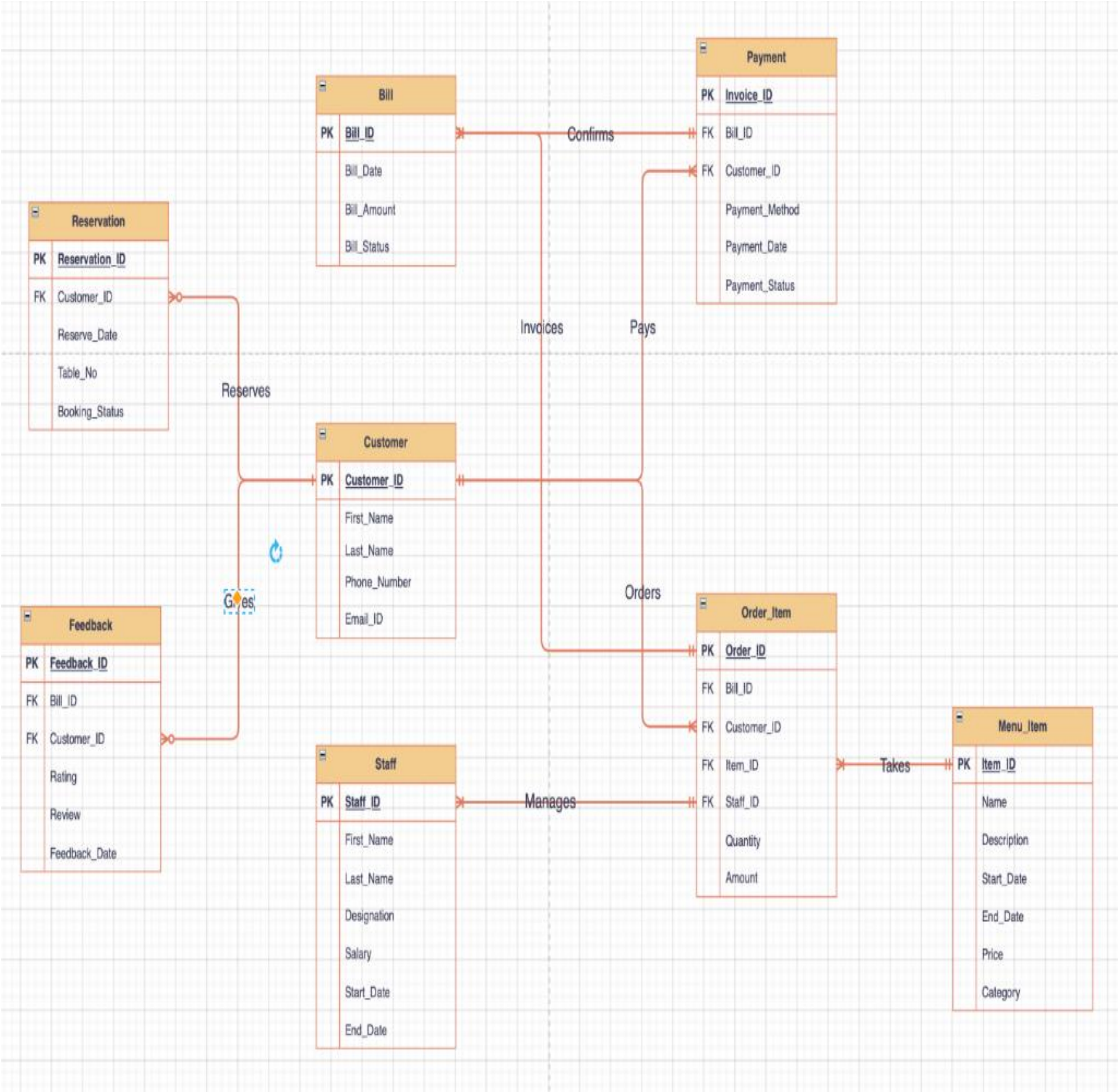
User Requirements

1. Customers should be able to make bookings using the system.
2. The system should display a menu with pricing and descriptions for all available foods.
3. Customers should be able to place orders using the system.
4. Customers should be able to pay for their orders using a variety of methods of payment, like credit cards, cash, or electronic payments.
5. The system should offer clients real-time information on the progress of their orders.
6. The system should enable personnel to manage the restaurant's inventory levels, including ordering fresh goods and maintaining track of current supplies.
7. The technology should allow personnel to maintain and update the restaurant's menu regularly.

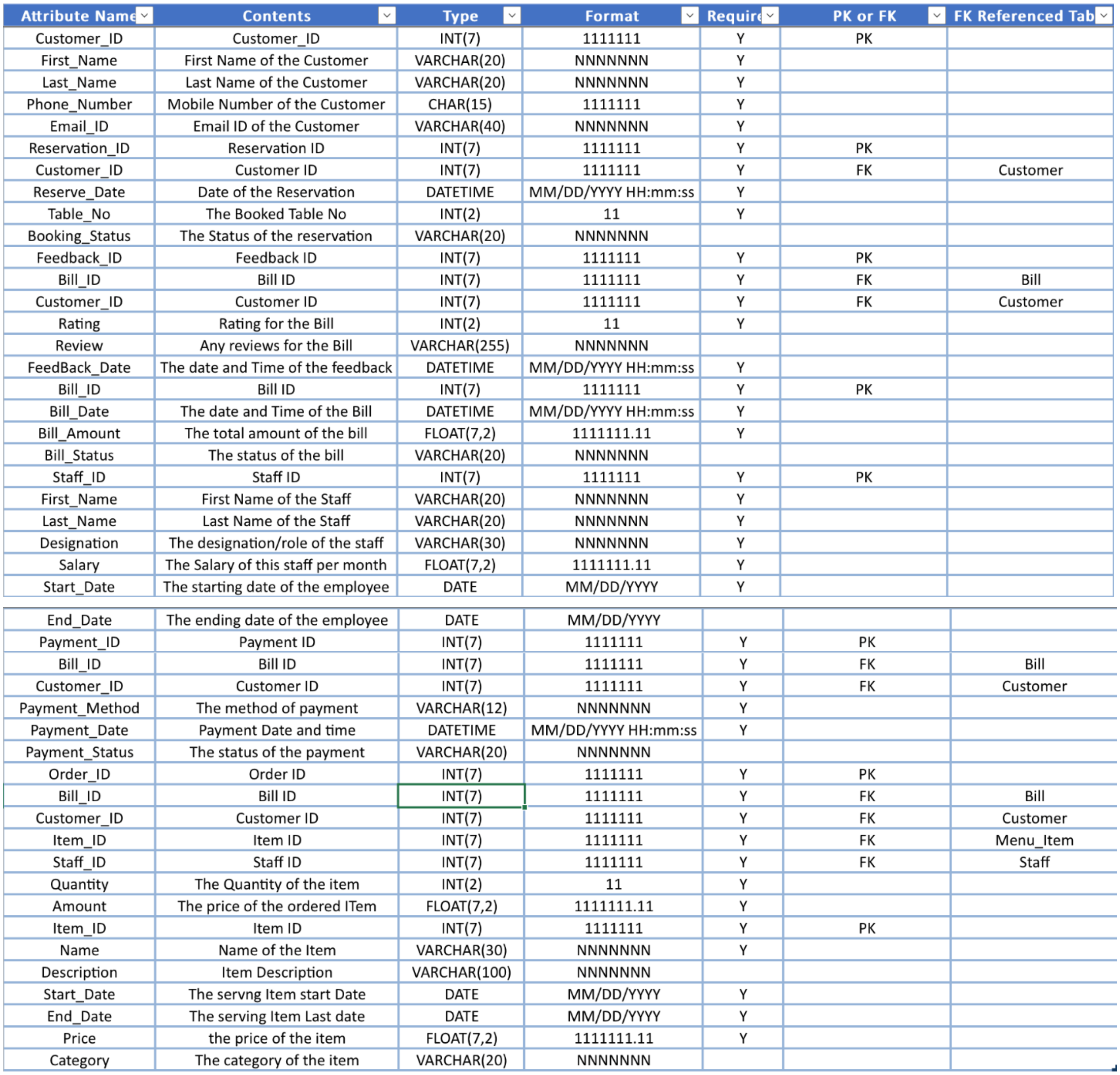
Business Rules

1. Customers should not be able to make orders for meals currently out of stock.
2. Customers should not be able to place orders for foods not available for distribution or pick-up.
3. Customers should not be allowed to make bookings for times that have already been reserved or unavailable.
4. The system should determine taxes and service costs based on the restaurant's location and laws.
5. The system ought to let the restaurant establish delivery prices based on the customer's distance and location.
6. By establishing adequate authentication and authorization methods, the system should assure data privacy and security.
7. The system must adhere to all applicable rules and regulations governing data privacy and security.

Entity-Relationship Diagram (ERD)



Data Dictionary



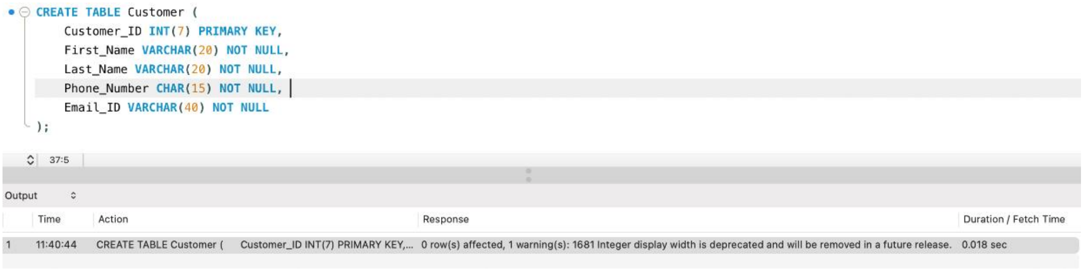
Attribute Name	Contents	Type	Format	Require	PK or FK	FK Referenced Tab
Customer_ID	Customer_ID	INT(7)	1111111	Y	PK	
First_Name	First Name of the Customer	VARCHAR(20)	NNNNNNN	Y		
Last_Name	Last Name of the Customer	VARCHAR(20)	NNNNNNN	Y		
Phone_Number	Mobile Number of the Customer	CHAR(15)	1111111	Y		
Email_ID	Email ID of the Customer	VARCHAR(40)	NNNNNNN	Y		
Reservation_ID	Reservation ID	INT(7)	1111111	Y	PK	
Customer_ID	Customer ID	INT(7)	1111111	Y	FK	Customer
Reserve_Date	Date of the Reservation	DATETIME	MM/DD/YYYY HH:mm:ss	Y		
Table_No	The Booked Table No	INT(2)	11	Y		
Booking_Status	The Status of the reservation	VARCHAR(20)	NNNNNNN			
Feedback_ID	Feedback ID	INT(7)	1111111	Y	PK	
Bill_ID	Bill ID	INT(7)	1111111	Y	FK	Bill
Customer_ID	Customer ID	INT(7)	1111111	Y	FK	Customer
Rating	Rating for the Bill	INT(2)	11	Y		
Review	Any reviews for the Bill	VARCHAR(255)	NNNNNNN			
FeedBack_Date	The date and Time of the feedback	DATETIME	MM/DD/YYYY HH:mm:ss	Y		
Bill_ID	Bill ID	INT(7)	1111111	Y	PK	
Bill_Date	The date and Time of the Bill	DATETIME	MM/DD/YYYY HH:mm:ss	Y		
Bill_Amount	The total amount of the bill	FLOAT(7,2)	1111111.11	Y		
Bill_Status	The status of the bill	VARCHAR(20)	NNNNNNN			
Staff_ID	Staff ID	INT(7)	1111111	Y	PK	
First_Name	First Name of the Staff	VARCHAR(20)	NNNNNNN	Y		
Last_Name	Last Name of the Staff	VARCHAR(20)	NNNNNNN	Y		
Designation	The designation/role of the staff	VARCHAR(30)	NNNNNNN	Y		
Salary	The Salary of this staff per month	FLOAT(7,2)	1111111.11	Y		
Start_Date	The starting date of the employee	DATE	MM/DD/YYYY	Y		
End_Date	The ending date of the employee	DATE	MM/DD/YYYY			
Payment_ID	Payment ID	INT(7)	1111111	Y	PK	
Bill_ID	Bill ID	INT(7)	1111111	Y	FK	Bill
Customer_ID	Customer ID	INT(7)	1111111	Y	FK	Customer
Payment_Method	The method of payment	VARCHAR(12)	NNNNNNN	Y		
Payment_Date	Payment Date and time	DATETIME	MM/DD/YYYY HH:mm:ss	Y		
Payment_Status	The status of the payment	VARCHAR(20)	NNNNNNN			
Order_ID	Order ID	INT(7)	1111111	Y	PK	
Bill_ID	Bill ID	INT(7)	1111111	Y	FK	Bill
Customer_ID	Customer ID	INT(7)	1111111	Y	FK	Customer
Item_ID	Item ID	INT(7)	1111111	Y	FK	Menu_Item
Staff_ID	Staff ID	INT(7)	1111111	Y	FK	Staff
Quantity	The Quantity of the item	INT(2)	11	Y		
Amount	The price of the ordered ITem	FLOAT(7,2)	1111111.11	Y		
Item_ID	Item ID	INT(7)	1111111	Y	PK	
Name	Name of the Item	VARCHAR(30)	NNNNNNN	Y		
Description	Item Description	VARCHAR(100)	NNNNNNN			
Start_Date	The servng Item start Date	DATE	MM/DD/YYYY	Y		
End_Date	The serving Item Last date	DATE	MM/DD/YYYY	Y		
Price	the price of the item	FLOAT(7,2)	1111111.11	Y		
Category	The category of the item	VARCHAR(20)	NNNNNNN			

Entity Generation and Data Entry

Customer Table

Explanation: This table contains personal information about the customer.

Create

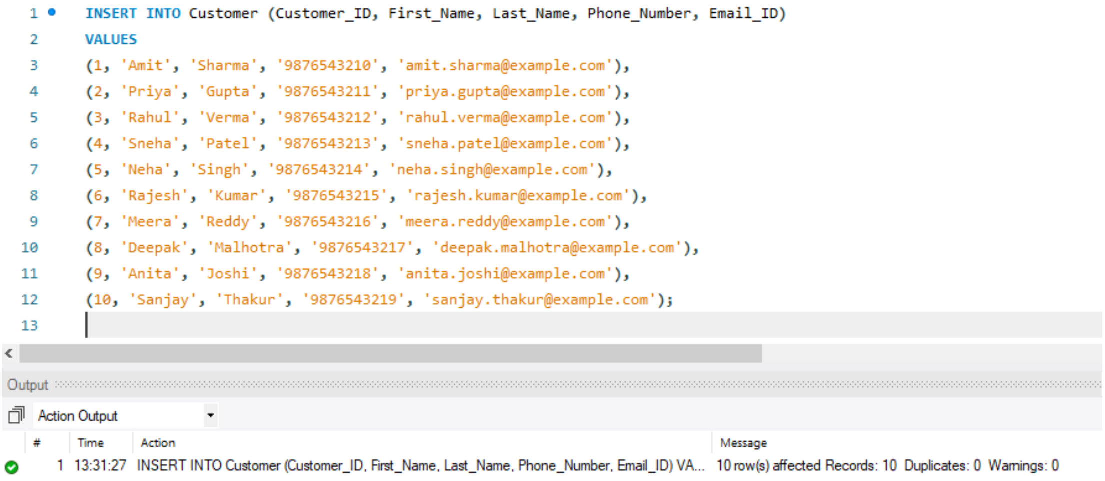


```sql
CREATE TABLE Customer (
    Customer_ID INT(7) PRIMARY KEY,
    First_Name VARCHAR(20) NOT NULL,
    Last_Name VARCHAR(20) NOT NULL,
    Phone_Number CHAR(15) NOT NULL,
    Email_ID VARCHAR(40) NOT NULL
);
```

```
37:5

Output
     Time      Action                                                    Response                                                                                                 Duration / Fetch Time
1    11:40:44  CREATE TABLE Customer (    Customer_ID INT(7) PRIMARY KEY,...  0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.  0.018 sec
```

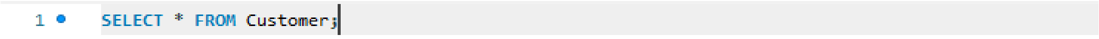
Insert



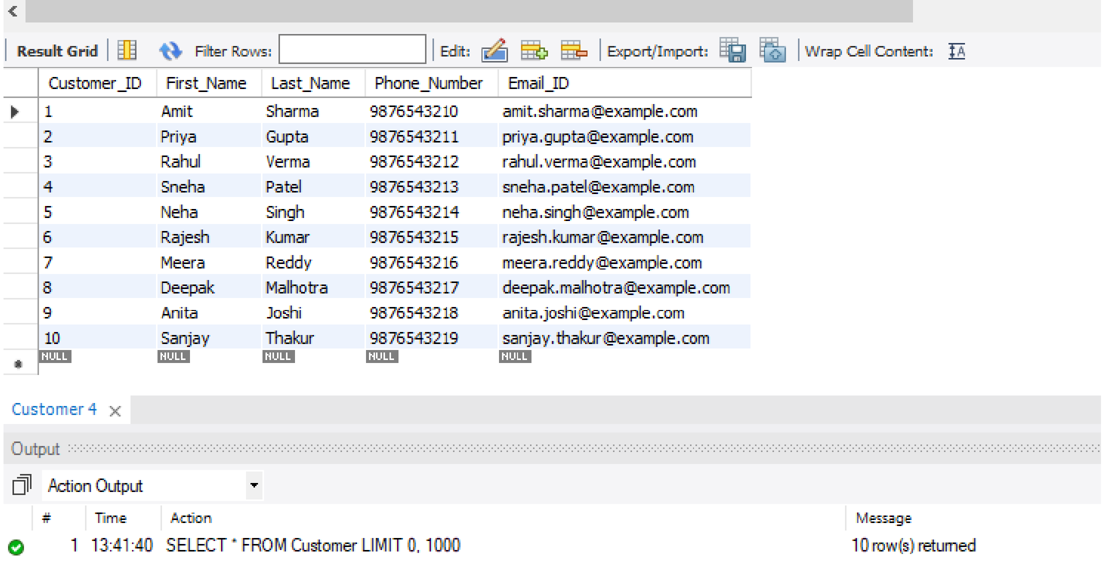
```sql
INSERT INTO Customer (Customer_ID, First_Name, Last_Name, Phone_Number, Email_ID)
VALUES
(1, 'Amit', 'Sharma', '9876543210', 'amit.sharma@example.com'),
(2, 'Priya', 'Gupta', '9876543211', 'priya.gupta@example.com'),
(3, 'Rahul', 'Verma', '9876543212', 'rahul.verma@example.com'),
(4, 'Sneha', 'Patel', '9876543213', 'sneha.patel@example.com'),
(5, 'Neha', 'Singh', '9876543214', 'neha.singh@example.com'),
(6, 'Rajesh', 'Kumar', '9876543215', 'rajesh.kumar@example.com'),
(7, 'Meera', 'Reddy', '9876543216', 'meera.reddy@example.com'),
(8, 'Deepak', 'Malhotra', '9876543217', 'deepak.malhotra@example.com'),
(9, 'Anita', 'Joshi', '9876543218', 'anita.joshi@example.com'),
(10, 'Sanjay', 'Thakur', '9876543219', 'sanjay.thakur@example.com');
```

```
Output
Action Output
   #  Time      Action                                                                   Message
✓  1  13:31:27  INSERT INTO Customer (Customer_ID, First_Name, Last_Name, Phone_Number, Email_ID) VA...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Select



```sql
SELECT * FROM Customer;
```



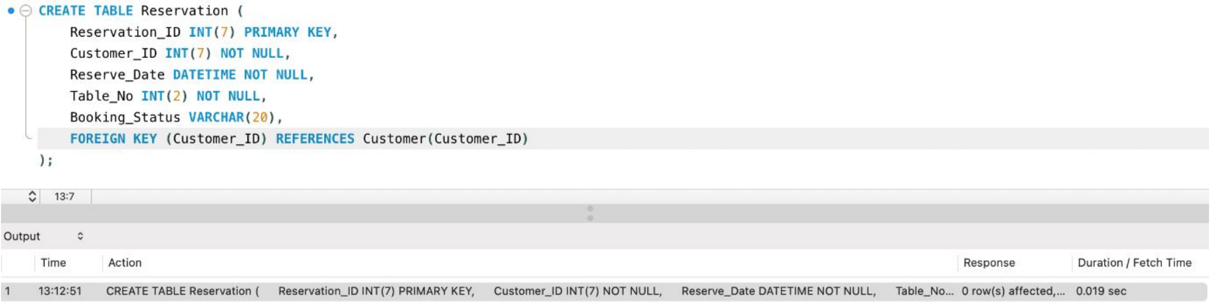
Customer_ID	First_Name	Last_Name	Phone_Number	Email_ID
1	Amit	Sharma	9876543210	amit.sharma@example.com
2	Priya	Gupta	9876543211	priya.gupta@example.com
3	Rahul	Verma	9876543212	rahul.verma@example.com
4	Sneha	Patel	9876543213	sneha.patel@example.com
5	Neha	Singh	9876543214	neha.singh@example.com
6	Rajesh	Kumar	9876543215	rajesh.kumar@example.com
7	Meera	Reddy	9876543216	meera.reddy@example.com
8	Deepak	Malhotra	9876543217	deepak.malhotra@example.com
9	Anita	Joshi	9876543218	anita.joshi@example.com
10	Sanjay	Thakur	9876543219	sanjay.thakur@example.com
NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                               Message
1   13:41:40  SELECT * FROM Customer LIMIT 0, 1000  10 row(s) returned
```

Reservation Table

Explanation: This table contains the information about the reservation made by a customer.

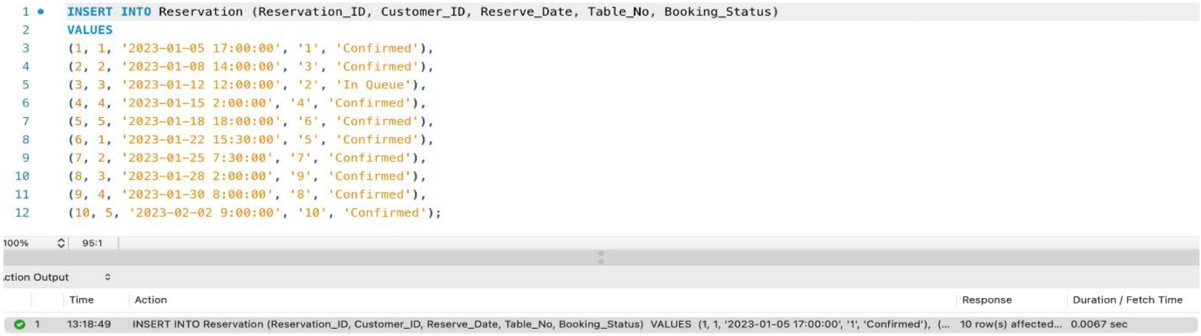
Create



```sql
CREATE TABLE Reservation (
    Reservation_ID INT(7) PRIMARY KEY,
    Customer_ID INT(7) NOT NULL,
    Reserve_Date DATETIME NOT NULL,
    Table_No INT(2) NOT NULL,
    Booking_Status VARCHAR(20),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
```

```
   Time      Action                                                                                                                           Response              Duration / Fetch Time
1  13:12:51  CREATE TABLE Reservation (   Reservation_ID INT(7) PRIMARY KEY,   Customer_ID INT(7) NOT NULL,   Reserve_Date DATETIME NOT NULL,   Table_No...  0 row(s) affected,...  0.019 sec
```

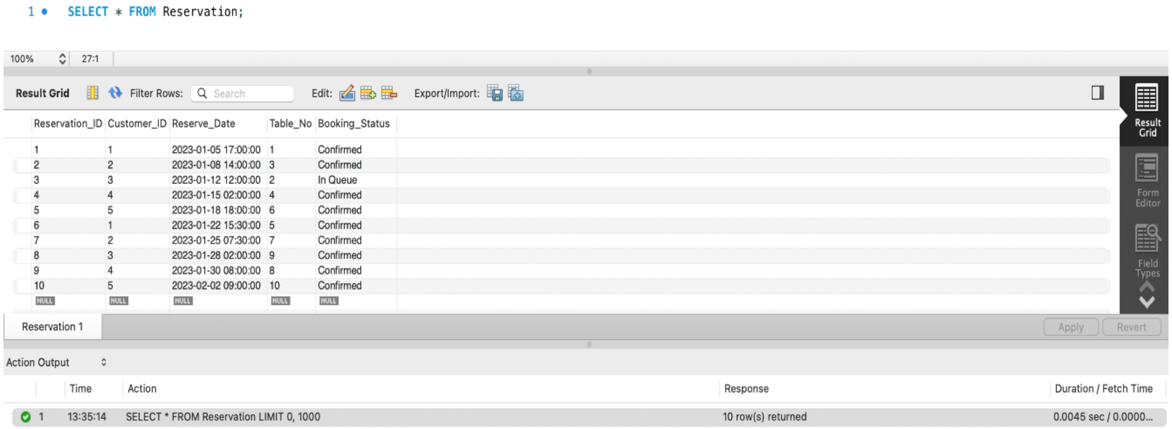
Insert



```sql
INSERT INTO Reservation (Reservation_ID, Customer_ID, Reserve_Date, Table_No, Booking_Status)
VALUES
(1, 1, '2023-01-05 17:00:00', '1', 'Confirmed'),
(2, 2, '2023-01-08 14:00:00', '3', 'Confirmed'),
(3, 3, '2023-01-12 12:00:00', '2', 'In Queue'),
(4, 4, '2023-01-15 2:00:00', '4', 'Confirmed'),
(5, 5, '2023-01-18 18:00:00', '6', 'Confirmed'),
(6, 1, '2023-01-22 15:30:00', '5', 'Confirmed'),
(7, 2, '2023-01-25 7:30:00', '7', 'Confirmed'),
(8, 3, '2023-01-28 2:00:00', '9', 'Confirmed'),
(9, 4, '2023-01-30 8:00:00', '8', 'Confirmed'),
(10, 5, '2023-02-02 9:00:00', '10', 'Confirmed');
```

```
    Time      Action                                                                                                                              Response                Duration / Fetch Time
1   13:18:49  INSERT INTO Reservation (Reservation_ID, Customer_ID, Reserve_Date, Table_No, Booking_Status)  VALUES  (1, 1, '2023-01-05 17:00:00', '1', 'Confirmed'),  (...   10 row(s) affected...   0.0067 sec
```

Select



```sql
SELECT * FROM Reservation;
```

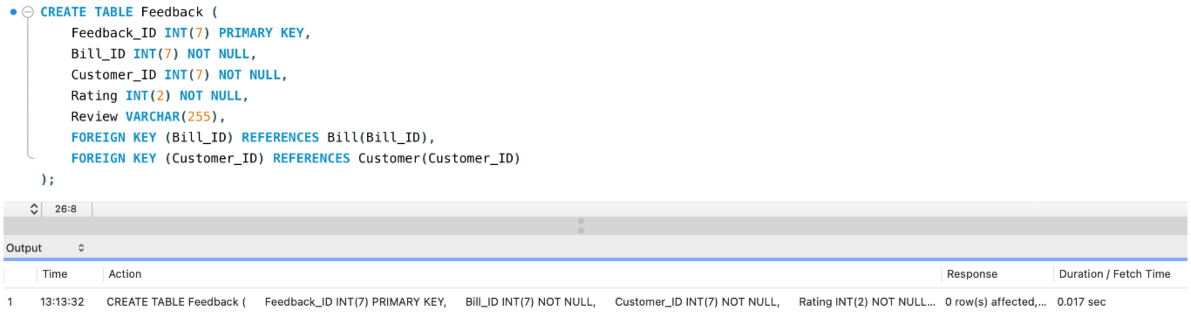
Reservation_ID	Customer_ID	Reserve_Date	Table_No	Booking_Status
1	1	2023-01-05 17:00:00	1	Confirmed
2	2	2023-01-08 14:00:00	3	Confirmed
3	3	2023-01-12 12:00:00	2	In Queue
4	4	2023-01-15 02:00:00	4	Confirmed
5	5	2023-01-18 18:00:00	6	Confirmed
6	1	2023-01-22 15:30:00	5	Confirmed
7	2	2023-01-25 07:30:00	7	Confirmed
8	3	2023-01-28 02:00:00	9	Confirmed
9	4	2023-01-30 08:00:00	8	Confirmed
10	5	2023-02-02 09:00:00	10	Confirmed
NULL	NULL	NULL	NULL	NULL

```
    Time      Action                                    Response              Duration / Fetch Time
1   13:35:14  SELECT * FROM Reservation LIMIT 0, 1000   10 row(s) returned    0.0045 sec / 0.0000...
```

Feedback Table

Explanation: This table contains the information about the feedback given by the customer for a particular bill.

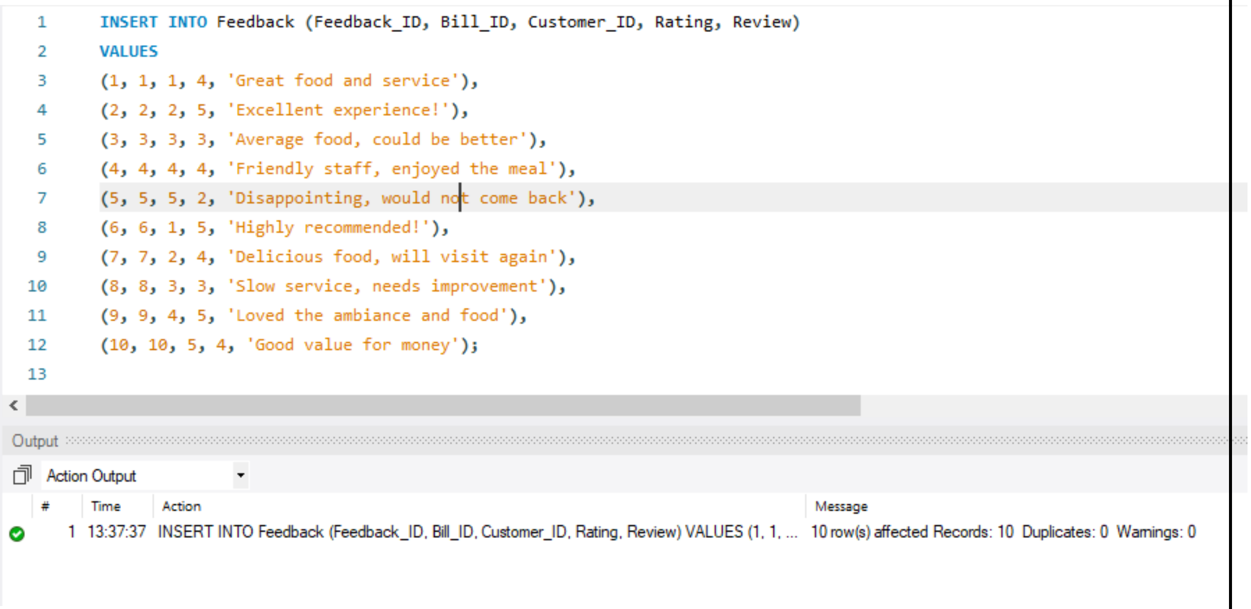
Create



```sql
CREATE TABLE Feedback (
    Feedback_ID INT(7) PRIMARY KEY,
    Bill_ID INT(7) NOT NULL,
    Customer_ID INT(7) NOT NULL,
    Rating INT(2) NOT NULL,
    Review VARCHAR(255),
    FOREIGN KEY (Bill_ID) REFERENCES Bill(Bill_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
```

```
    Time      Action                                                                                                                        Response              Duration / Fetch Time
1   13:13:32  CREATE TABLE Feedback (     Feedback_ID INT(7) PRIMARY KEY,     Bill_ID INT(7) NOT NULL,     Customer_ID INT(7) NOT NULL,     Rating INT(2) NOT NULL...   0 row(s) affected,...   0.017 sec
```

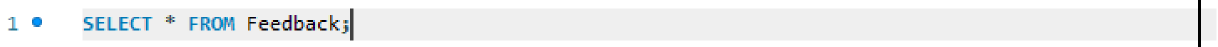
Insert



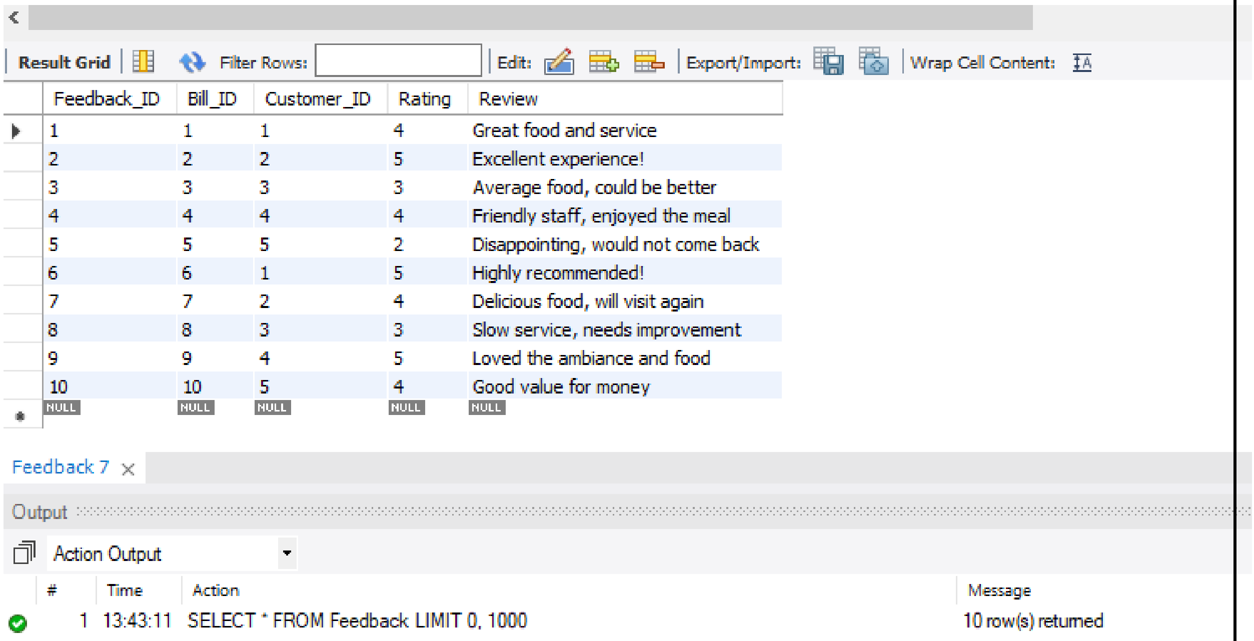
```sql
INSERT INTO Feedback (Feedback_ID, Bill_ID, Customer_ID, Rating, Review)
VALUES
(1, 1, 1, 4, 'Great food and service'),
(2, 2, 2, 5, 'Excellent experience!'),
(3, 3, 3, 3, 'Average food, could be better'),
(4, 4, 4, 4, 'Friendly staff, enjoyed the meal'),
(5, 5, 5, 2, 'Disappointing, would not come back'),
(6, 6, 1, 5, 'Highly recommended!'),
(7, 7, 2, 4, 'Delicious food, will visit again'),
(8, 8, 3, 3, 'Slow service, needs improvement'),
(9, 9, 4, 5, 'Loved the ambiance and food'),
(10, 10, 5, 4, 'Good value for money');
```

```
#   Time      Action                                                                          Message
1   13:37:37  INSERT INTO Feedback (Feedback_ID, Bill_ID, Customer_ID, Rating, Review) VALUES (1, 1, ...   10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Select



```sql
SELECT * FROM Feedback;
```



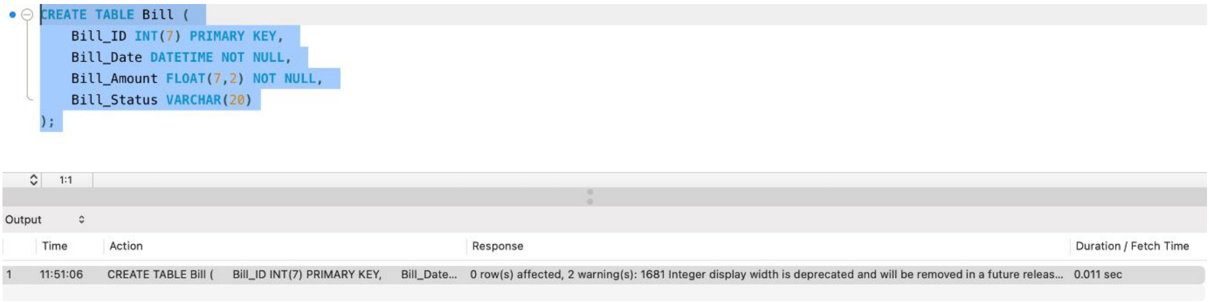
Feedback_ID	Bill_ID	Customer_ID	Rating	Review
1	1	1	4	Great food and service
2	2	2	5	Excellent experience!
3	3	3	3	Average food, could be better
4	4	4	4	Friendly staff, enjoyed the meal
5	5	5	2	Disappointing, would not come back
6	6	1	5	Highly recommended!
7	7	2	4	Delicious food, will visit again
8	8	3	3	Slow service, needs improvement
9	9	4	5	Loved the ambiance and food
10	10	5	4	Good value for money
NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                              Message
1   13:43:11  SELECT * FROM Feedback LIMIT 0, 1000  10 row(s) returned
```

Bill Table

Explanation: This table contains the information about the total bill of the order.

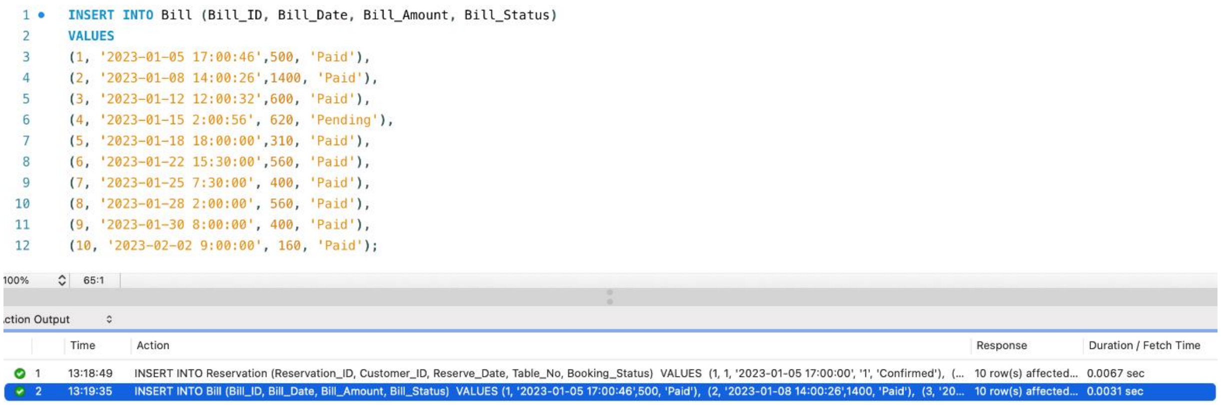
Create



```sql
CREATE TABLE Bill (
    Bill_ID INT(7) PRIMARY KEY,
    Bill_Date DATETIME NOT NULL,
    Bill_Amount FLOAT(7,2) NOT NULL,
    Bill_Status VARCHAR(20)
);
```

```
    Time      Action                                                  Response                                                                               Duration / Fetch Time
1   11:51:06  CREATE TABLE Bill (    Bill_ID INT(7) PRIMARY KEY,    Bill_Date...  0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future releas...  0.011 sec
```

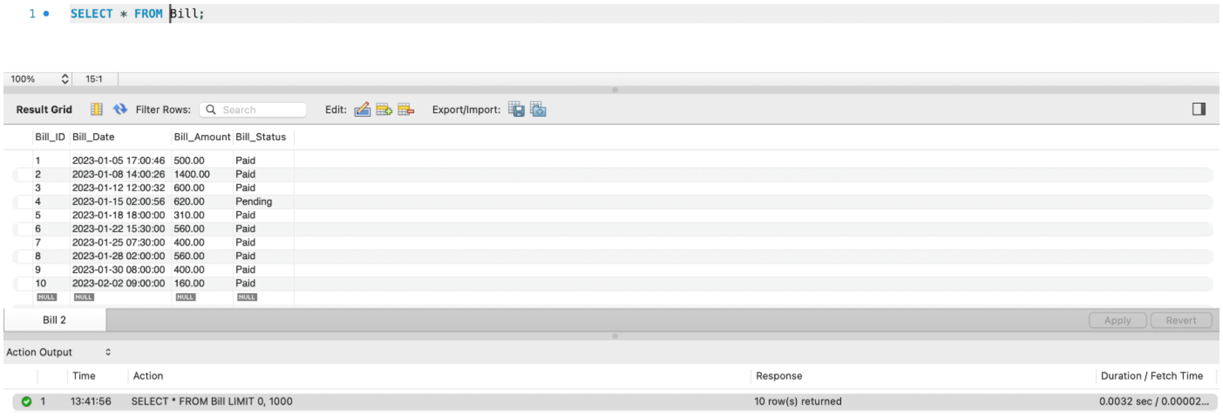
Insert



```sql
INSERT INTO Bill (Bill_ID, Bill_Date, Bill_Amount, Bill_Status)
VALUES
(1, '2023-01-05 17:00:46',500, 'Paid'),
(2, '2023-01-08 14:00:26',1400, 'Paid'),
(3, '2023-01-12 12:00:32',600, 'Paid'),
(4, '2023-01-15 2:00:56', 620, 'Pending'),
(5, '2023-01-18 18:00:00',310, 'Paid'),
(6, '2023-01-22 15:30:00',560, 'Paid'),
(7, '2023-01-25 7:30:00', 400, 'Paid'),
(8, '2023-01-28 2:00:00', 560, 'Paid'),
(9, '2023-01-30 8:00:00', 400, 'Paid'),
(10, '2023-02-02 9:00:00', 160, 'Paid');
```

```
     Time      Action                                                                                                                              Response             Duration / Fetch Time
✓ 1  13:18:49  INSERT INTO Reservation (Reservation_ID, Customer_ID, Reserve_Date, Table_No, Booking_Status)  VALUES  (1, 1, '2023-01-05 17:00:00', '1', 'Confirmed'),  (...  10 row(s) affected...  0.0067 sec
✓ 2  13:19:35  INSERT INTO Bill (Bill_ID, Bill_Date, Bill_Amount, Bill_Status)  VALUES (1, '2023-01-05 17:00:46',500, 'Paid'),  (2, '2023-01-08 14:00:26',1400, 'Paid'),  (3, '20...  10 row(s) affected...  0.0031 sec
```

Select



```sql
SELECT * FROM Bill;
```

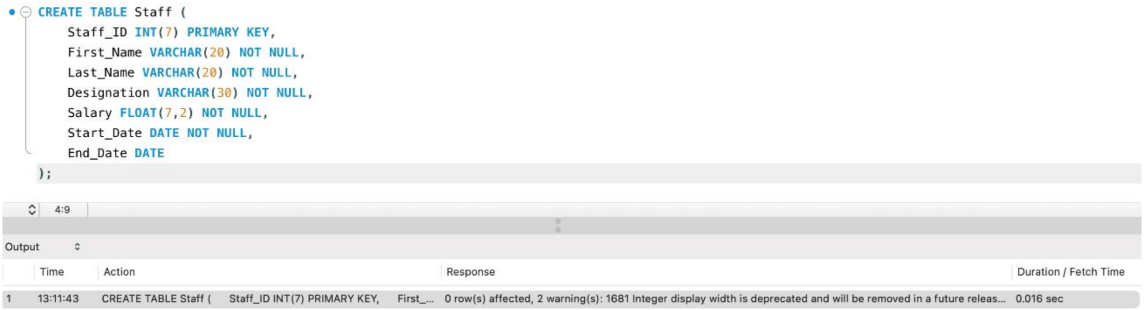
Bill_ID	Bill_Date	Bill_Amount	Bill_Status
1	2023-01-05 17:00:46	500.00	Paid
2	2023-01-08 14:00:26	1400.00	Paid
3	2023-01-12 12:00:32	600.00	Paid
4	2023-01-15 02:00:56	620.00	Pending
5	2023-01-18 18:00:00	310.00	Paid
6	2023-01-22 15:30:00	560.00	Paid
7	2023-01-25 07:30:00	400.00	Paid
8	2023-01-28 02:00:00	560.00	Paid
9	2023-01-30 08:00:00	400.00	Paid
10	2023-02-02 09:00:00	160.00	Paid
NULL	NULL	NULL	NULL

```
     Time      Action                          Response            Duration / Fetch Time
✓ 1  13:41:56  SELECT * FROM Bill LIMIT 0, 1000  10 row(s) returned  0.0032 sec / 0.00002...
```

Staff Table

Explanation: This table contains information about the staff managing the order.

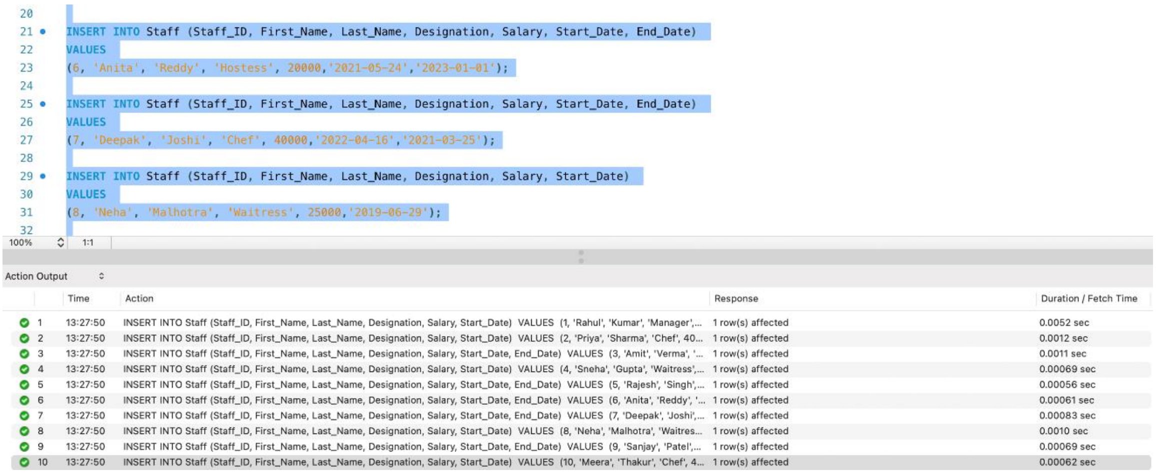
Create



```sql
CREATE TABLE Staff (
    Staff_ID INT(7) PRIMARY KEY,
    First_Name VARCHAR(20) NOT NULL,
    Last_Name VARCHAR(20) NOT NULL,
    Designation VARCHAR(30) NOT NULL,
    Salary FLOAT(7,2) NOT NULL,
    Start_Date DATE NOT NULL,
    End_Date DATE
);
```

```
    Time      Action                                                   Response                                                                                      Duration / Fetch Time
1   13:11:43  CREATE TABLE Staff (    Staff_ID INT(7) PRIMARY KEY,    First_...  0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future releas...  0.016 sec
```

Insert



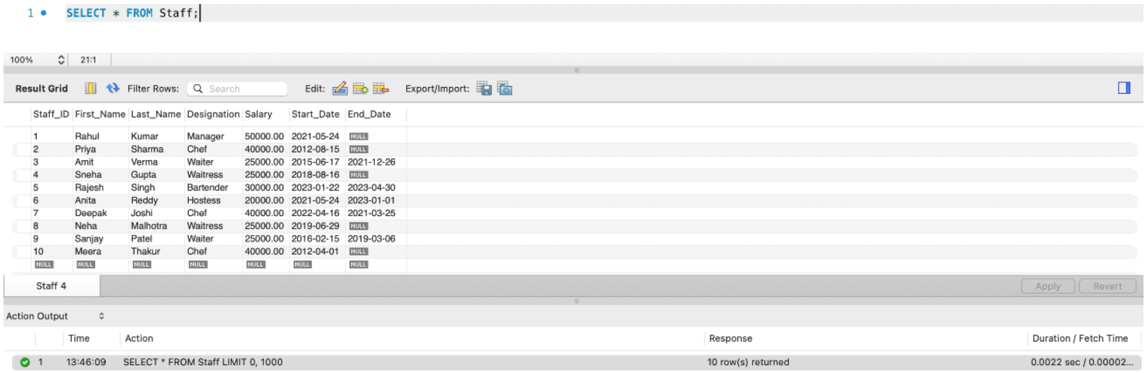
```sql
INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)
VALUES
(6, 'Anita', 'Reddy', 'Hostess', 20000,'2021-05-24','2023-01-01');

INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)
VALUES
(7, 'Deepak', 'Joshi', 'Chef', 40000,'2022-04-16','2021-03-25');

INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)
VALUES
(8, 'Neha', 'Malhotra', 'Waitress', 25000,'2019-06-29');
```

```
     Time      Action                                                                                                              Response            Duration / Fetch Time
1    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)  VALUES  (1, 'Rahul', 'Kumar', 'Manager',...  1 row(s) affected  0.0052 sec
2    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)  VALUES  (2, 'Priya', 'Sharma', 'Chef', 40...  1 row(s) affected  0.0012 sec
3    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)  VALUES  (3, 'Amit', 'Verma', '...  1 row(s) affected  0.0011 sec
4    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)  VALUES  (4, 'Sneha', 'Gupta', 'Waitress',...  1 row(s) affected  0.00069 sec
5    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)  VALUES  (5, 'Rajesh', 'Singh',...  1 row(s) affected  0.00056 sec
6    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)  VALUES  (6, 'Anita', 'Reddy', '...  1 row(s) affected  0.00061 sec
7    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)  VALUES  (7, 'Deepak', 'Joshi',...  1 row(s) affected  0.00083 sec
8    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)  VALUES  (8, 'Neha', 'Malhotra', 'Waitres...  1 row(s) affected  0.0010 sec
9    13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date, End_Date)  VALUES  (9, 'Sanjay', 'Patel',...  1 row(s) affected  0.00069 sec
10   13:27:50  INSERT INTO Staff (Staff_ID, First_Name, Last_Name, Designation, Salary, Start_Date)  VALUES  (10, 'Meera', 'Thakur', 'Chef', 4...  1 row(s) affected  0.00062 sec
```

Select



```sql
SELECT * FROM Staff;
```

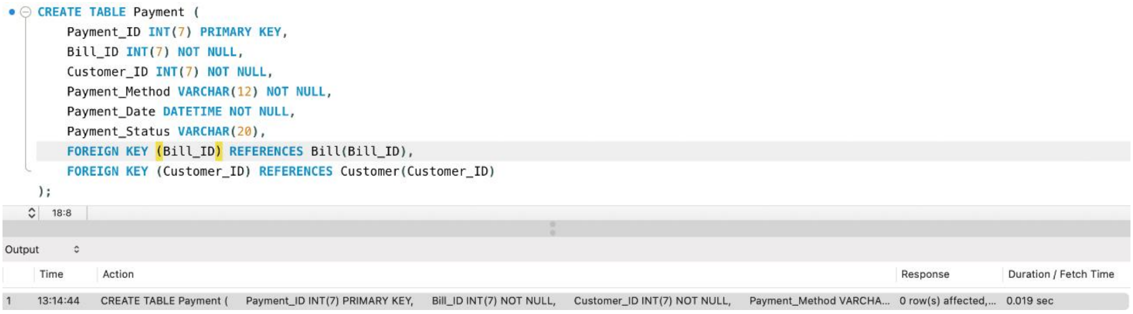
Staff_ID	First_Name	Last_Name	Designation	Salary	Start_Date	End_Date
1	Rahul	Kumar	Manager	50000.00	2021-05-24	NULL
2	Priya	Sharma	Chef	40000.00	2012-08-15	NULL
3	Amit	Verma	Waiter	25000.00	2015-06-17	2021-12-26
4	Sneha	Gupta	Waitress	25000.00	2018-08-16	NULL
5	Rajesh	Singh	Bartender	30000.00	2023-01-22	2023-04-30
6	Anita	Reddy	Hostess	20000.00	2021-05-24	2023-01-01
7	Deepak	Joshi	Chef	40000.00	2022-04-16	2021-03-25
8	Neha	Malhotra	Waitress	25000.00	2019-06-29	NULL
9	Sanjay	Patel	Waiter	25000.00	2016-02-15	2019-03-06
10	Meera	Thakur	Chef	40000.00	2012-04-01	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
     Time      Action                            Response            Duration / Fetch Time
1    13:46:09  SELECT * FROM Staff LIMIT 0, 1000  10 row(s) returned  0.0022 sec / 0.00002...
```

Payment Table

Explanation: This table contains information about the staff managing the order.

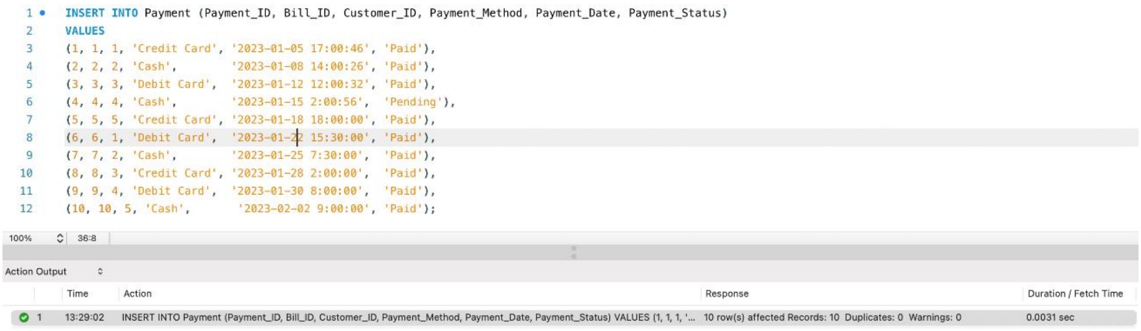
Create



```
CREATE TABLE Payment (
    Payment_ID INT(7) PRIMARY KEY,
    Bill_ID INT(7) NOT NULL,
    Customer_ID INT(7) NOT NULL,
    Payment_Method VARCHAR(12) NOT NULL,
    Payment_Date DATETIME NOT NULL,
    Payment_Status VARCHAR(20),
    FOREIGN KEY (Bill_ID) REFERENCES Bill(Bill_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
```

```
    Time      Action                                                                                                                    Response               Duration / Fetch Time
1   13:14:44  CREATE TABLE Payment (    Payment_ID INT(7) PRIMARY KEY,    Bill_ID INT(7) NOT NULL,    Customer_ID INT(7) NOT NULL,    Payment_Method VARCHA...  0 row(s) affected,...  0.019 sec
```

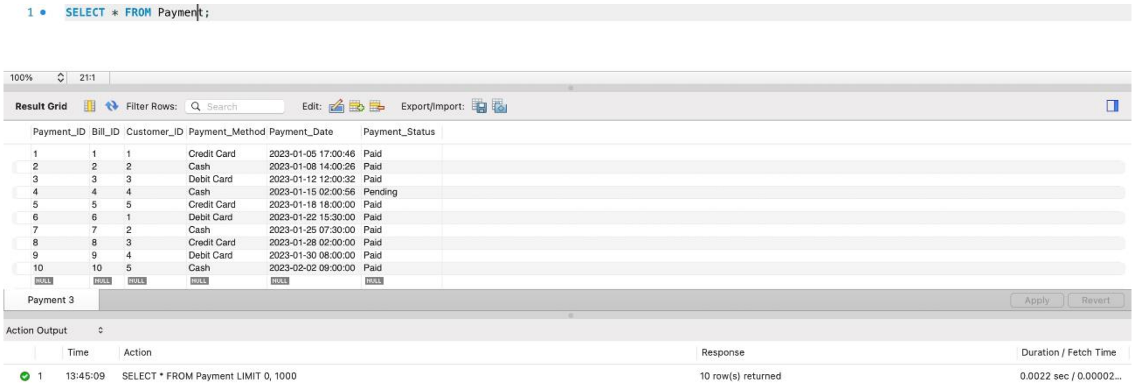
Insert



```
INSERT INTO Payment (Payment_ID, Bill_ID, Customer_ID, Payment_Method, Payment_Date, Payment_Status)
VALUES
(1, 1, 1, 'Credit Card', '2023-01-05 17:00:46', 'Paid'),
(2, 2, 2, 'Cash',        '2023-01-08 14:00:26', 'Paid'),
(3, 3, 3, 'Debit Card',  '2023-01-12 12:00:32', 'Paid'),
(4, 4, 4, 'Cash',        '2023-01-15 2:00:56',  'Pending'),
(5, 5, 5, 'Credit Card', '2023-01-18 18:00:00', 'Paid'),
(6, 6, 1, 'Debit Card',  '2023-01-22 15:30:00', 'Paid'),
(7, 7, 2, 'Cash',        '2023-01-25 7:30:00',  'Paid'),
(8, 8, 3, 'Credit Card', '2023-01-28 2:00:00',  'Paid'),
(9, 9, 4, 'Debit Card',  '2023-01-30 8:00:00',  'Paid'),
(10, 10, 5, 'Cash',       '2023-02-02 9:00:00', 'Paid');
```

```
    Time      Action                                                                                                      Response                                              Duration / Fetch Time
1   13:29:02  INSERT INTO Payment (Payment_ID, Bill_ID, Customer_ID, Payment_Method, Payment_Date, Payment_Status) VALUES (1, 1, 1, '...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0  0.0031 sec
```

Select



```
SELECT * FROM Payment;
```

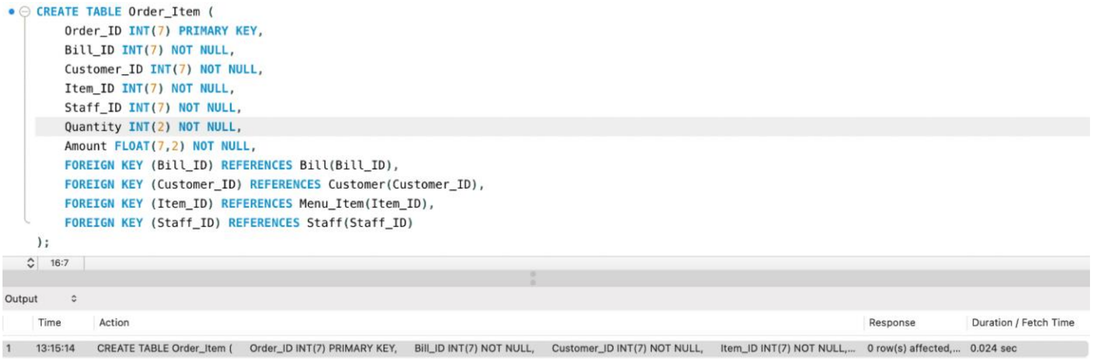
Payment_ID	Bill_ID	Customer_ID	Payment_Method	Payment_Date	Payment_Status
1	1	1	Credit Card	2023-01-05 17:00:46	Paid
2	2	2	Cash	2023-01-08 14:00:26	Paid
3	3	3	Debit Card	2023-01-12 12:00:32	Paid
4	4	4	Cash	2023-01-15 02:00:56	Pending
5	5	5	Credit Card	2023-01-18 18:00:00	Paid
6	6	1	Debit Card	2023-01-22 15:30:00	Paid
7	7	2	Cash	2023-01-25 07:30:00	Paid
8	8	3	Credit Card	2023-01-28 02:00:00	Paid
9	9	4	Debit Card	2023-01-30 08:00:00	Paid
10	10	5	Cash	2023-02-02 09:00:00	Paid
NULL	NULL	NULL	NULL	NULL	NULL

```
    Time      Action                              Response           Duration / Fetch Time
1   13:45:09  SELECT * FROM Payment LIMIT 0, 1000  10 row(s) returned  0.0022 sec / 0.00002...
```

Order_Item Table:

Explanation: This table contains information about the staff managing the order.

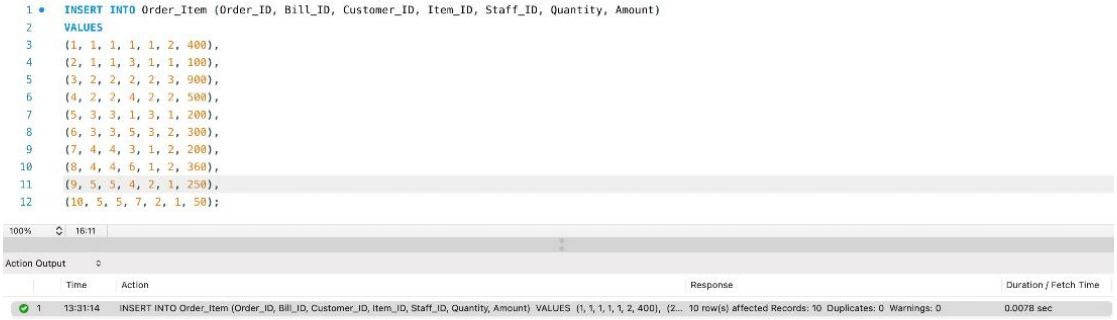
Create



```
CREATE TABLE Order_Item (
    Order_ID INT(7) PRIMARY KEY,
    Bill_ID INT(7) NOT NULL,
    Customer_ID INT(7) NOT NULL,
    Item_ID INT(7) NOT NULL,
    Staff_ID INT(7) NOT NULL,
    Quantity INT(2) NOT NULL,
    Amount FLOAT(7,2) NOT NULL,
    FOREIGN KEY (Bill_ID) REFERENCES Bill(Bill_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Item_ID) REFERENCES Menu_Item(Item_ID),
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
);
```

```
     Time      Action                                                                                                                           Response                  Duration / Fetch Time
1    13:15:14  CREATE TABLE Order_Item (    Order_ID INT(7) PRIMARY KEY,    Bill_ID INT(7) NOT NULL,    Customer_ID INT(7) NOT NULL,    Item_ID INT(7) NOT NULL,...  0 row(s) affected,...  0.024 sec
```

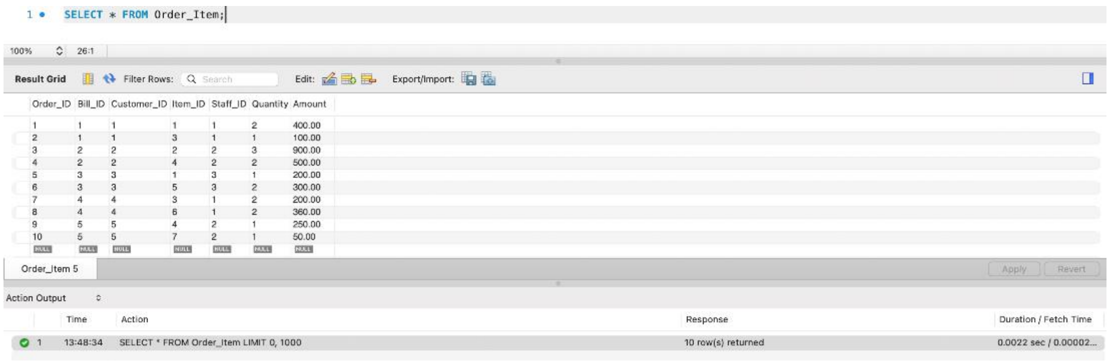
Insert



```
INSERT INTO Order_Item (Order_ID, Bill_ID, Customer_ID, Item_ID, Staff_ID, Quantity, Amount)
VALUES
(1, 1, 1, 1, 1, 2, 400),
(2, 1, 1, 3, 1, 1, 100),
(3, 2, 2, 2, 2, 3, 900),
(4, 2, 2, 4, 2, 2, 500),
(5, 3, 3, 1, 3, 1, 200),
(6, 3, 3, 5, 3, 2, 300),
(7, 4, 4, 3, 1, 2, 200),
(8, 4, 4, 6, 1, 2, 360),
(9, 5, 5, 4, 2, 1, 250),
(10, 5, 5, 7, 2, 1, 50);
```

```
     Time      Action                                                                                                         Response                                                    Duration / Fetch Time
✓ 1  13:31:14  INSERT INTO Order_Item (Order_ID, Bill_ID, Customer_ID, Item_ID, Staff_ID, Quantity, Amount)  VALUES  (1, 1, 1, 1, 1, 2, 400),  (2...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0  0.0078 sec
```

Select



```
SELECT * FROM Order_Item;
```

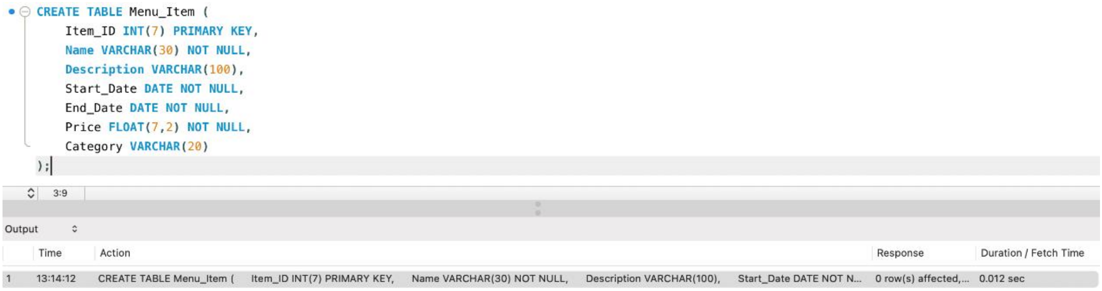
Order_ID	Bill_ID	Customer_ID	Item_ID	Staff_ID	Quantity	Amount
1	1	1	1	1	2	400.00
2	1	1	3	1	1	100.00
3	2	2	2	2	3	900.00
4	2	2	4	2	2	500.00
5	3	3	1	3	1	200.00
6	3	3	5	3	2	300.00
7	4	4	3	1	2	200.00
8	4	4	6	1	2	360.00
9	5	5	4	2	1	250.00
10	5	5	7	2	1	50.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
     Time      Action                                  Response            Duration / Fetch Time
✓ 1  13:48:34  SELECT * FROM Order_Item LIMIT 0, 1000  10 row(s) returned  0.0022 sec / 0.00002...
```

Menu_Item Table

Explanation: This table contains information about the staff managing the order.

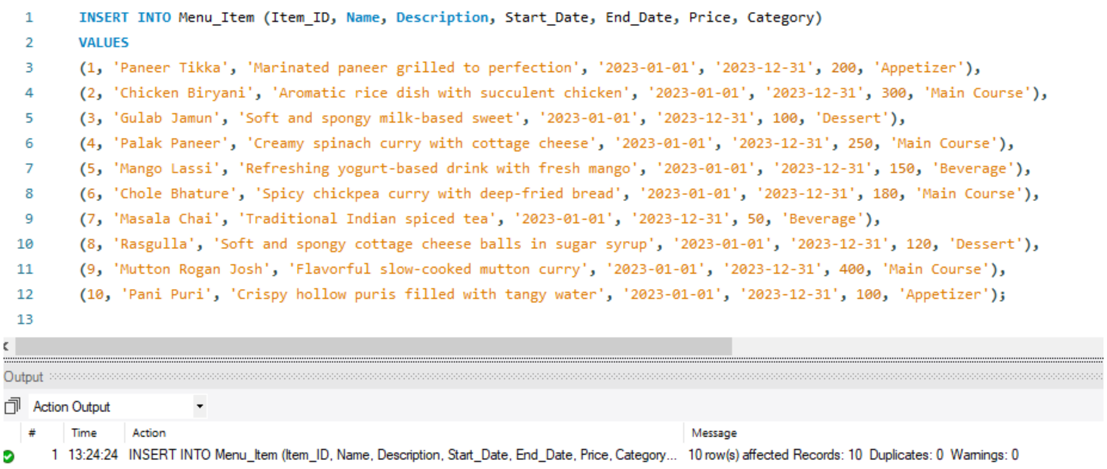
Create



```
CREATE TABLE Menu_Item (
    Item_ID INT(7) PRIMARY KEY,
    Name VARCHAR(30) NOT NULL,
    Description VARCHAR(100),
    Start_Date DATE NOT NULL,
    End_Date DATE NOT NULL,
    Price FLOAT(7,2) NOT NULL,
    Category VARCHAR(20)
);
```

```
     Time      Action                                                                                                                   Response              Duration / Fetch Time
1    13:14:12  CREATE TABLE Menu_Item (    Item_ID INT(7) PRIMARY KEY,    Name VARCHAR(30) NOT NULL,    Description VARCHAR(100),    Start_Date DATE NOT N...  0 row(s) affected,...  0.012 sec
```

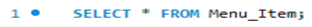
Insert



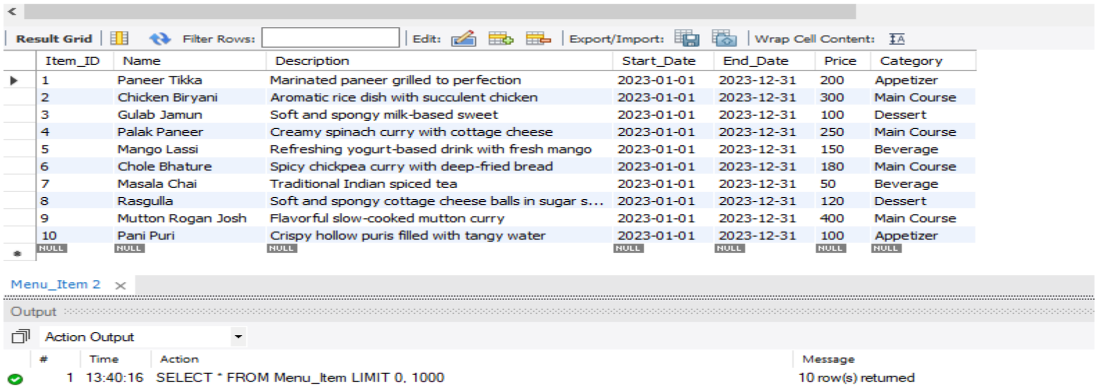
```
INSERT INTO Menu_Item (Item_ID, Name, Description, Start_Date, End_Date, Price, Category)
VALUES
(1, 'Paneer Tikka', 'Marinated paneer grilled to perfection', '2023-01-01', '2023-12-31', 200, 'Appetizer'),
(2, 'Chicken Biryani', 'Aromatic rice dish with succulent chicken', '2023-01-01', '2023-12-31', 300, 'Main Course'),
(3, 'Gulab Jamun', 'Soft and spongy milk-based sweet', '2023-01-01', '2023-12-31', 100, 'Dessert'),
(4, 'Palak Paneer', 'Creamy spinach curry with cottage cheese', '2023-01-01', '2023-12-31', 250, 'Main Course'),
(5, 'Mango Lassi', 'Refreshing yogurt-based drink with fresh mango', '2023-01-01', '2023-12-31', 150, 'Beverage'),
(6, 'Chole Bhature', 'Spicy chickpea curry with deep-fried bread', '2023-01-01', '2023-12-31', 180, 'Main Course'),
(7, 'Masala Chai', 'Traditional Indian spiced tea', '2023-01-01', '2023-12-31', 50, 'Beverage'),
(8, 'Rasgulla', 'Soft and spongy cottage cheese balls in sugar syrup', '2023-01-01', '2023-12-31', 120, 'Dessert'),
(9, 'Mutton Rogan Josh', 'Flavorful slow-cooked mutton curry', '2023-01-01', '2023-12-31', 400, 'Main Course'),
(10, 'Pani Puri', 'Crispy hollow puris filled with tangy water', '2023-01-01', '2023-12-31', 100, 'Appetizer');
```

```
   #  Time      Action                                                                          Message
   1  13:24:24  INSERT INTO Menu_Item (Item_ID, Name, Description, Start_Date, End_Date, Price, Category...  10 row(s) affected Records: 10  Duplicates: 0  Warnings: 0
```

Select



```sql
SELECT * FROM Menu_Item;
```



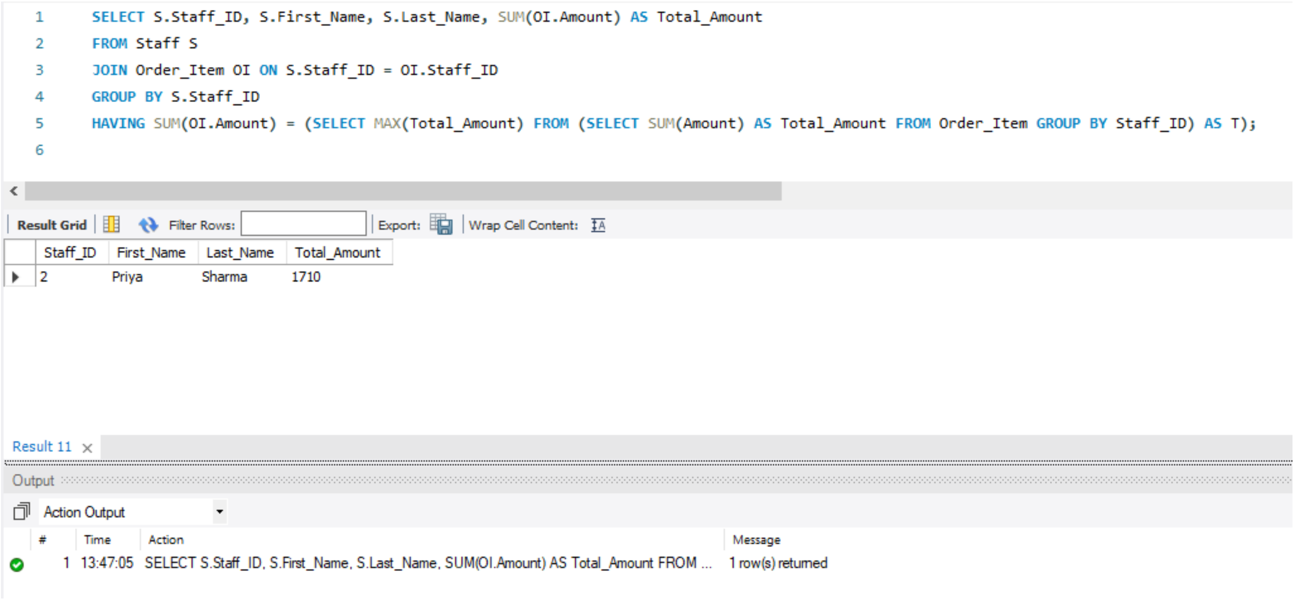
Item_ID	Name	Description	Start_Date	End_Date	Price	Category
1	Paneer Tikka	Marinated paneer grilled to perfection	2023-01-01	2023-12-31	200	Appetizer
2	Chicken Biryani	Aromatic rice dish with succulent chicken	2023-01-01	2023-12-31	300	Main Course
3	Gulab Jamun	Soft and spongy milk-based sweet	2023-01-01	2023-12-31	100	Dessert
4	Palak Paneer	Creamy spinach curry with cottage cheese	2023-01-01	2023-12-31	250	Main Course
5	Mango Lassi	Refreshing yogurt-based drink with fresh mango	2023-01-01	2023-12-31	150	Beverage
6	Chole Bhature	Spicy chickpea curry with deep-fried bread	2023-01-01	2023-12-31	180	Main Course
7	Masala Chai	Traditional Indian spiced tea	2023-01-01	2023-12-31	50	Beverage
8	Rasgulla	Soft and spongy cottage cheese balls in sugar s...	2023-01-01	2023-12-31	120	Dessert
9	Mutton Rogan Josh	Flavorful slow-cooked mutton curry	2023-01-01	2023-12-31	400	Main Course
10	Pani Puri	Crispy hollow puris filled with tangy water	2023-01-01	2023-12-31	100	Appetizer
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                                  Message
1   13:40:16  SELECT * FROM Menu_Item LIMIT 0, 1000   10 row(s) returned
```

Data Retrieval and Reports

Query 1

Retrieve the staff members who have processed orders of the highest total amount.



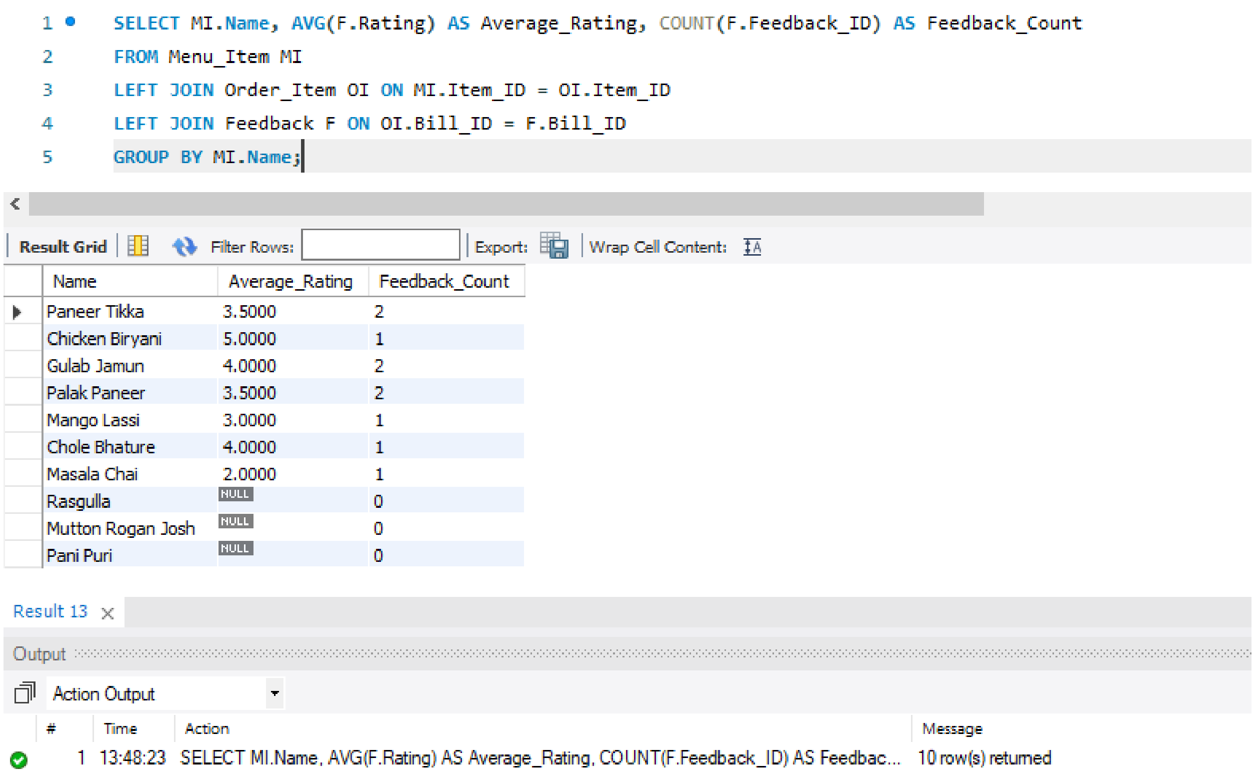
```sql
SELECT S.Staff_ID, S.First_Name, S.Last_Name, SUM(OI.Amount) AS Total_Amount
FROM Staff S
JOIN Order_Item OI ON S.Staff_ID = OI.Staff_ID
GROUP BY S.Staff_ID
HAVING SUM(OI.Amount) = (SELECT MAX(Total_Amount) FROM (SELECT SUM(Amount) AS Total_Amount FROM Order_Item GROUP BY Staff_ID) AS T);
```

Staff_ID	First_Name	Last_Name	Total_Amount
2	Priya	Sharma	1710

```
#   Time      Action                                                                    Message
1   13:47:05  SELECT S.Staff_ID, S.First_Name, S.Last_Name, SUM(OI.Amount) AS Total_Amount FROM ...   1 row(s) returned
```

Query 2

Retrieve the menu items along with the average rating and the count of feedback received for each item



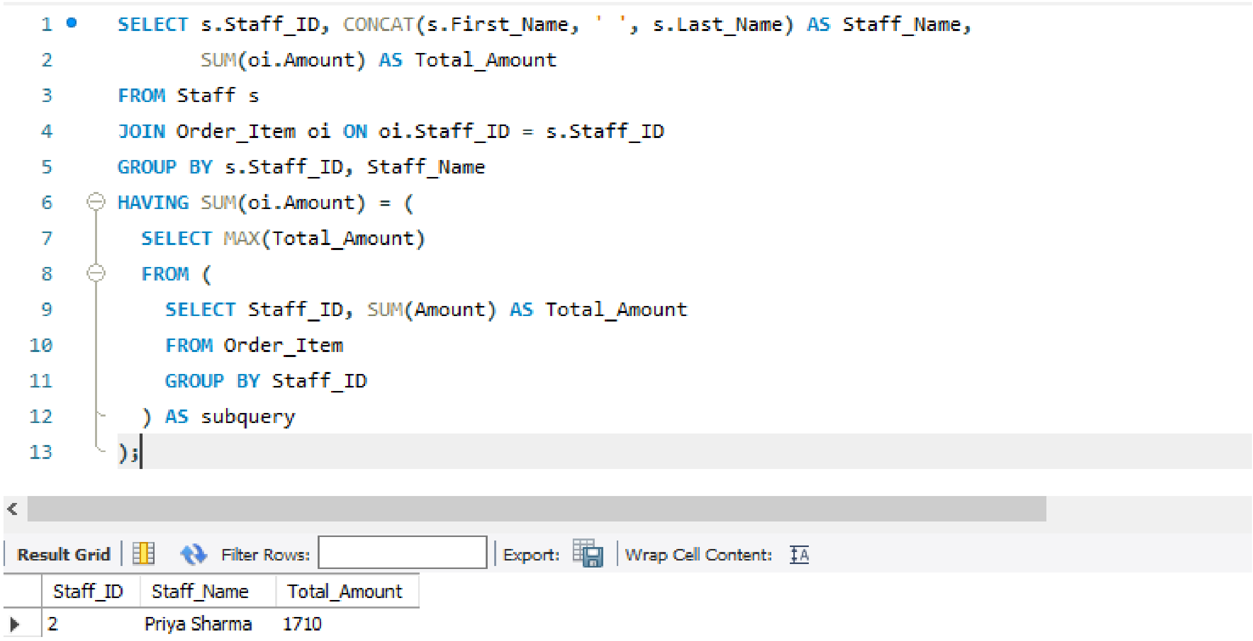
```
SELECT MI.Name, AVG(F.Rating) AS Average_Rating, COUNT(F.Feedback_ID) AS Feedback_Count
FROM Menu_Item MI
LEFT JOIN Order_Item OI ON MI.Item_ID = OI.Item_ID
LEFT JOIN Feedback F ON OI.Bill_ID = F.Bill_ID
GROUP BY MI.Name;
```

Name	Average_Rating	Feedback_Count
Paneer Tikka	3.5000	2
Chicken Biryani	5.0000	1
Gulab Jamun	4.0000	2
Palak Paneer	3.5000	2
Mango Lassi	3.0000	1
Chole Bhature	4.0000	1
Masala Chai	2.0000	1
Rasgulla	NULL	0
Mutton Rogan Josh	NULL	0
Pani Puri	NULL	0

	#	Time	Action	Message
✓	1	13:48:23	SELECT MI.Name, AVG(F.Rating) AS Average_Rating, COUNT(F.Feedback_ID) AS Feedbac...	10 row(s) returned

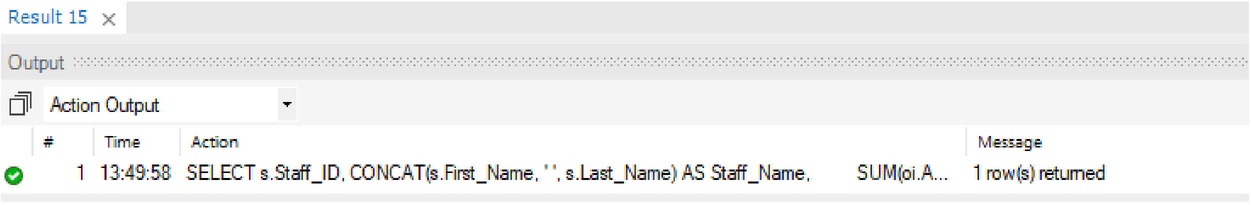
Query 3

To retrieve the staff members who have handled orders with the highest total amount



```sql
SELECT s.Staff_ID, CONCAT(s.First_Name, ' ', s.Last_Name) AS Staff_Name,
       SUM(oi.Amount) AS Total_Amount
FROM Staff s
JOIN Order_Item oi ON oi.Staff_ID = s.Staff_ID
GROUP BY s.Staff_ID, Staff_Name
HAVING SUM(oi.Amount) = (
  SELECT MAX(Total_Amount)
  FROM (
    SELECT Staff_ID, SUM(Amount) AS Total_Amount
    FROM Order_Item
    GROUP BY Staff_ID
  ) AS subquery
);
```

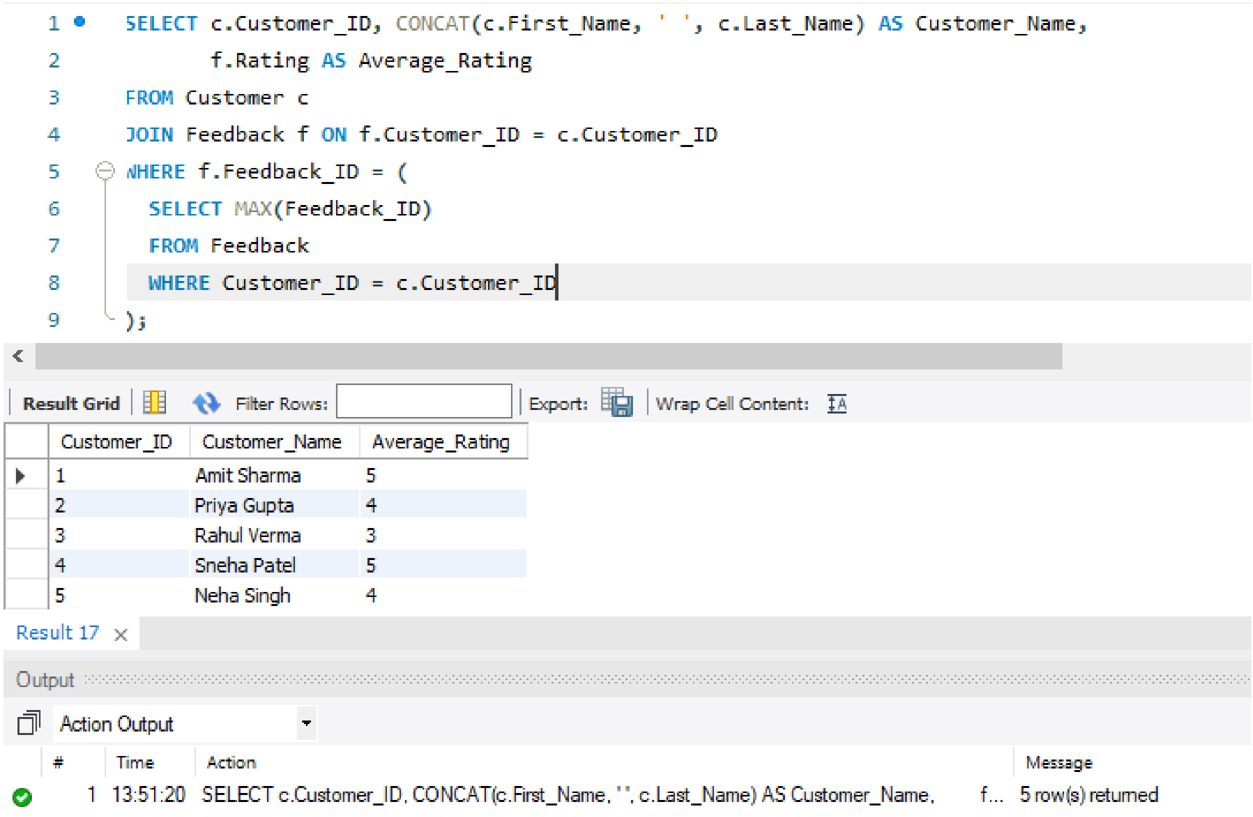
Staff_ID	Staff_Name	Total_Amount
2	Priya Sharma	1710



```
#  Time      Action                                                                      Message
1  13:49:58  SELECT s.Staff_ID, CONCAT(s.First_Name, ' ', s.Last_Name) AS Staff_Name,  SUM(oi.A...  1 row(s) returned
```

Query 4

To retrieve the customers who have provided feedback along with their average rating



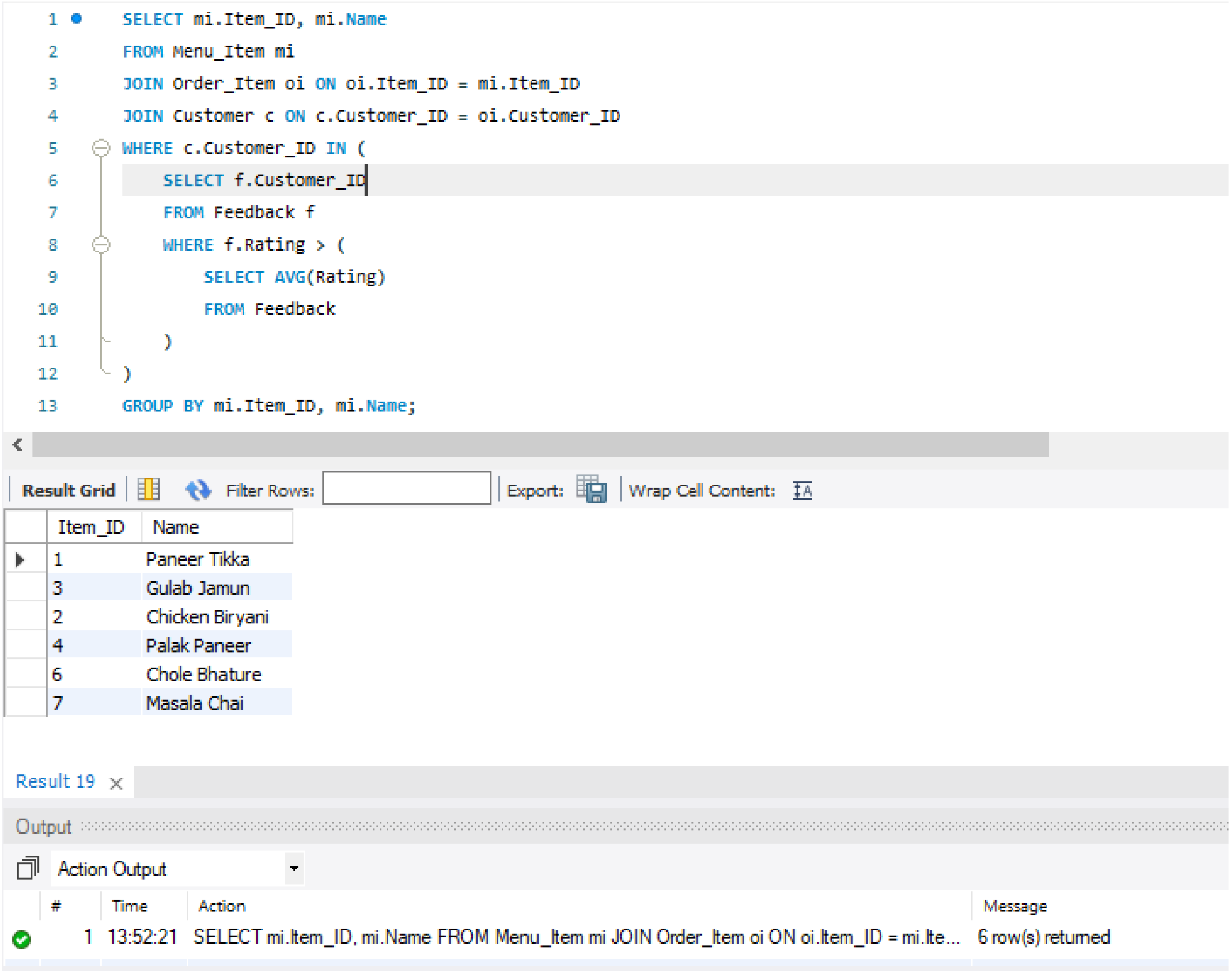
```sql
SELECT c.Customer_ID, CONCAT(c.First_Name, ' ', c.Last_Name) AS Customer_Name,
       f.Rating AS Average_Rating
FROM Customer c
JOIN Feedback f ON f.Customer_ID = c.Customer_ID
WHERE f.Feedback_ID = (
  SELECT MAX(Feedback_ID)
  FROM Feedback
  WHERE Customer_ID = c.Customer_ID
);
```

Customer_ID	Customer_Name	Average_Rating
1	Amit Sharma	5
2	Priya Gupta	4
3	Rahul Verma	3
4	Sneha Patel	5
5	Neha Singh	4

```
#  Time      Action                                                                 Message
1  13:51:20  SELECT c.Customer_ID, CONCAT(c.First_Name, ' ', c.Last_Name) AS Customer_Name,   f...  5 row(s) returned
```

Query 5

Retrieve the menu items that have been ordered by customers who have provided feedback with a rating greater than the average rating



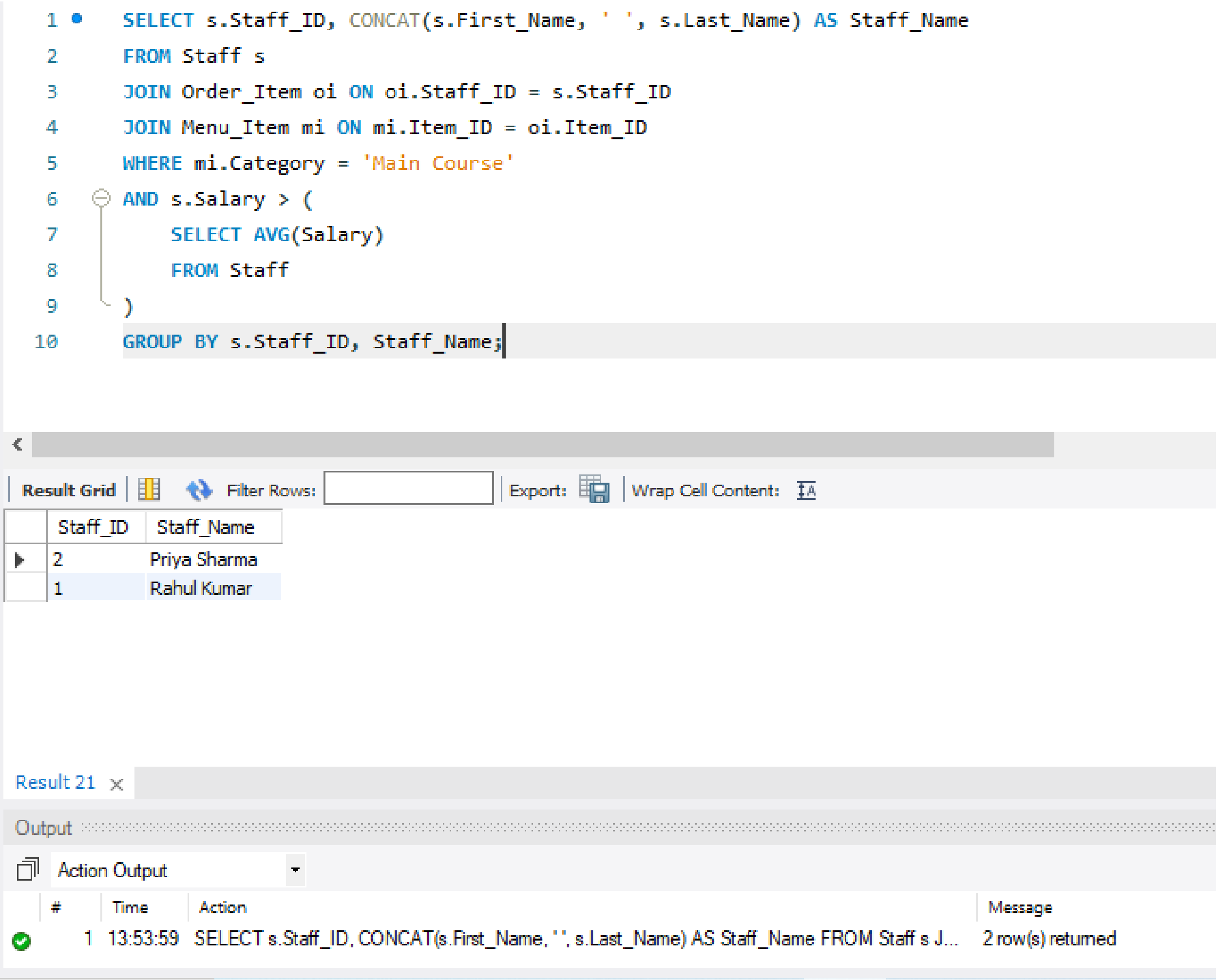
```sql
SELECT mi.Item_ID, mi.Name
FROM Menu_Item mi
JOIN Order_Item oi ON oi.Item_ID = mi.Item_ID
JOIN Customer c ON c.Customer_ID = oi.Customer_ID
WHERE c.Customer_ID IN (
    SELECT f.Customer_ID
    FROM Feedback f
    WHERE f.Rating > (
        SELECT AVG(Rating)
        FROM Feedback
    )
)
GROUP BY mi.Item_ID, mi.Name;
```

Item_ID	Name
1	Paneer Tikka
3	Gulab Jamun
2	Chicken Biryani
4	Palak Paneer
6	Chole Bhature
7	Masala Chai

#	Time	Action	Message
1	13:52:21	SELECT mi.Item_ID, mi.Name FROM Menu_Item mi JOIN Order_Item oi ON oi.Item_ID = mi.Ite...	6 row(s) returned

Query 6

Retrieve the staff members who have handled orders for menu items in a specific category and have a salary greater than the average salary of all staff members



```
SELECT s.Staff_ID, CONCAT(s.First_Name, ' ', s.Last_Name) AS Staff_Name
FROM Staff s
JOIN Order_Item oi ON oi.Staff_ID = s.Staff_ID
JOIN Menu_Item mi ON mi.Item_ID = oi.Item_ID
WHERE mi.Category = 'Main Course'
AND s.Salary > (
    SELECT AVG(Salary)
    FROM Staff
)
GROUP BY s.Staff_ID, Staff_Name;
```

Staff_ID	Staff_Name
2	Priya Sharma
1	Rahul Kumar

```
#   Time      Action                                                                              Message
1   13:53:59  SELECT s.Staff_ID, CONCAT(s.First_Name, ' ', s.Last_Name) AS Staff_Name FROM Staff s J...  2 row(s) returned
```