**Restaurant Management Organization Database**

**PROJECT DESCRIPTION:**

A restaurant is a place where people go to eat diverse types of cuisine, hold parties, and have business lunches. Restaurant management must keep track of customers and employees for legal and communication purposes.

However, relying on paper records can lead to data loss. To avoid this, we created the **Restaurant Management Organization Database** system. This system assists in the recording of customer information, services, and bills, such as money transfer, banquet hall service details, parking service details, and staff information such as working hours, salary, and personal information.

To meet the needs of users, the system can be easily updated, corrected, and changed. The primary goal of this project is to transition from manual paper-based record keeping to a database system, which reduces manual labor while improving the security of customer and employee data.

**OBJECTIVES:**

* To book tables for customers based on availability and their preferences.
* To order any kind of food.
* Minimize the usage of paper or manual work and move digitally.
* To record any details of staff, including their personal details and working services and hours.
* To track payment details of customer for the food.
* Records parking service details.
* Stock inventory maintenance.
* Helps in accepting different modes of payments from customers.
* Records Details of banquet hall used by customers.

**SCOPE:**

1. The goal of this project is to switch from conventional paper-based operations to digital ones, which will result in better time management and service quality.
2. Customer access is made easier and their ability to offer feedback or report problems is made more convenient thanks to the automation of processes.
3. Additionally, the system guarantees customer information confidentiality, lowering the possibility of fraud. The system has the option of including real-time tracking.
4. Customer information can be saved for future reference, assisting marketing efforts. Furthermore, the system reduces the possibility of financial mismanagement and facilitates data backup for future use.
5. The system allows for online booking and banquet hall reservations.
6. Management of the workforce, including pay information and service history, is also made simpler and more efficient.
7. Finally, resource status can be tracked and updated in real time.

**PROJECT REQUIREMENTS:**

Operating System: Windows

Database: Microsoft SQL SERVER Management Studio

Applications: Microsoft word

**DATABASE REQUIREMENTS:**

The following information contains the data tables for the database collection:

1. Restaurant Tables
2. Customer Table
3. Table Bookings Table
4. Order Table
5. Menu Table
6. Staff Table
7. Valet Parking Table
8. Delivery Table
9. Transaction Table
10. Payment Table
11. Banquet Halls Table
12. Banquet Booking Table

**USER REQUIREMENTS:**

* Users will be able to book banquet halls and tables allocations.
* User wants to know the availability of banquet halls and tables during booking.
* User wants to know the services provided by restaurant.
* If a user booked a banquet hall, they require the details of staff include their working hours, the services they provide.
* The user wants to know the mode of payment for any transaction.
* User wants to know the details include timings of restaurant, opening and closing hours, weekly or monthly holidays.
* Users will get to know the availability of banquet halls and booking time based on the hours booked.
* Users can reserve a table before dining based on the table status.
* Users will have access to menu items and staff take care of placing an order.

**BUSINESS RULES:**

* + One customer can book any number of tables.
  + One table can accommodate single customers’ ID.
  + Each of the customers will have a unique ID.
  + Customer can order multiple dishes to tables.
  + Each staff has unique ID.
  + Each Delivery is associated with one order id.
  + Customers can request multiple services from multiple staff.
  + Each customer will have a unique valet parking ID.
  + Customer can have multiple bills and multiple payments under single ID.
  + Each customer will have specific check-in and check-out timings based on their bookings.
  + Payment information is stored and has more bill cycles.
  + Real time access to management and customers.

**ENTITY RELATIONSHIP DIAGRAM (ERD):**

An entity relationship (ER) diagram is a form of process diagram that shows the relationships between "entities" like individuals, things, or thoughts inside a system.

For this Restaurant Management Organization Database, we will have entities like:

**A picture containing graphical user interface

Description automatically generated**

**DATA DICTIONARY:**

A data dictionary is a list of descriptions of the data objects or other things in a data model that programmers and other people may refer to.

**Data Dictionary Importance: Developer use the data dictionary for the following reasons.**

* To manage the large restaurant operations.
* To allocate staff to a particular task.
* To facilitate banquet bookings based on availability.
* To accommodate several types of orders.
* To check the status of the tables used for dining.



References for the shortcuts used in the Data Dictionary

****

**ENTITY GENERATION AND INSERTION:**

**Creating Database:**

We have created a database named **“Restaurant”** in MS Server SQL as shown below.

**Graphical user interface, application

Description automatically generated**

**Creating Staff Table:**

**Explanation:** Creating the table staff to include all the details and their job title.

**Query:**

CREATE TABLE [Staff] (

[StaffID] INT NOT NULL,

[StaffName] VARCHAR (100) NOT NULL,

[JobTitle] VARCHAR (100) NOT NULL,

[StartDate] DATE NOT NULL,

[DateofBirth] DATE NOT NULL,

[PhoneNumber] INT NOT NULL,

[SSN] VARCHAR (100) NOT NULL,

[Address] VARCHAR (100) NOT NULL,

[County] CHAR (32) NOT NULL,

[State] CHAR (32) NOT NULL,

[ZipCode] INT NOT NULL,

PRIMARY KEY ([StaffID])

);

**Graphical user interface, text, application

Description automatically generated**

**Inserting records for Staff Table:**

**Explanation:** Inserting the records into staff tables to include all the personal details and their job title.

**Query:**

INSERT INTO [Staff] ([StaffID], [StaffName], [JobTitle], [StartDate], [DateofBirth], [PhoneNumber], [SSN], [Address], [County], [State], [ZipCode])

VALUES

(1, 'John Doe', 'Manager', '2020-01-01', '1990-01-01', 1234567890, '123-45-6789', '123 Main St', 'Anytown', 'CA', 12345),

( 2, 'Jane Doe', 'Assistant Manager', '2020-02-01', '1995-01-01', 1345678901, '234-56-7890', '456 Second St', 'Othertown', 'NY', 23456),

(3, 'Bob Smith', 'Supervisor', '2020-03-01', '1992-01-01', 1456789012, '345-67-8901', '789 Third St', 'Newtown', 'TX', 34567),

(4, 'Sara Johnson', 'Sales Associate', '2020-04-01', '1994-01-01', 1567890123, '456-78-9012', '12 Fourth St', 'Smalltown', 'NC', 45678),

(5, 'David Brown', 'Customer Service Representative', '2020-05-01', '1993-01-01', 1678901234, '567-89-0123', '345 Fifth St', 'Bigtown', 'FL', 56789),

(6, 'Emily Jones', 'Marketing Specialist', '2020-06-01', '1991-01-01', 1789012345, '678-90-1234', '678 Sixth St', 'Hometown', 'GA', 67890),

(7, 'Karen Wilson', 'Sales Manager', '2020-07-01', '1988-01-01', 1890123456, '789-01-2345', '901 Seventh St', 'Cityville', 'CA', 78901),

(8, 'Samuel Davis', 'Supervisor', '2020-08-01', '1989-01-01', 1901234567, '890-12-3456', '234 Eighth St', 'Countytown', 'PA', 89012),

(9, 'Megan Taylor', 'Assistant Manager', '2020-09-01', '1992-01-01', 1012345678, '901-23-4567', '567 Ninth St', 'Villagetown', 'VA', 90123),

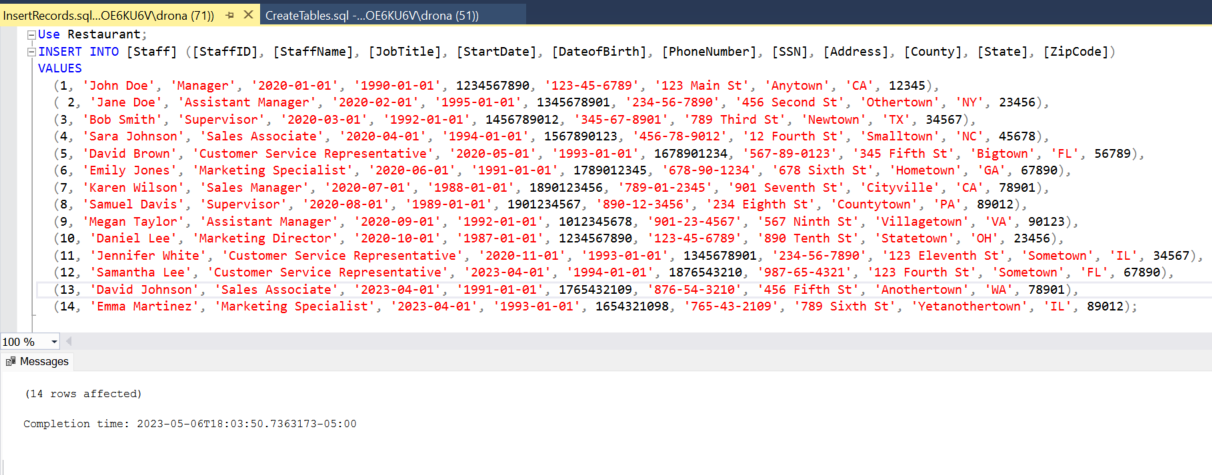
(10, 'Daniel Lee', 'Marketing Director', '2020-10-01', '1987-01-01', 1234567890, '123-45-6789', '890 Tenth St', 'Statetown', 'OH', 23456),

(11, 'Jennifer White', 'Customer Service Representative', '2020-11-01', '1993-01-01', 1345678901, '234-56-7890', '123 Eleventh St', 'Sometown', 'IL', 34567),

(12, 'Samantha Lee', 'Customer Service Representative', '2023-04-01', '1994-01-01', 1876543210, '987-65-4321', '123 Fourth St', 'Sometown', 'FL', 67890),

(13, 'David Johnson', 'Sales Associate', '2023-04-01', '1991-01-01', 1765432109, '876-54-3210', '456 Fifth St', 'Anothertown', 'WA', 78901),

(14, 'Emma Martinez', 'Marketing Specialist', '2023-04-01', '1993-01-01', 1654321098, '765-43-2109', '789 Sixth St', 'Yetanothertown', 'IL', 89012);

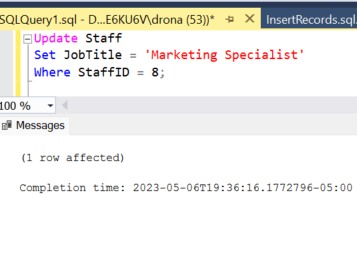
****

**Update Staff Table:**

**Explanation:** Updating a staff record whose name is Samuel Davis, job title from Supervisor to Marketing Specialist.

**Query:**

Update Staff   
Set JobTitle = 'Marketing Specialist'  
Where StaffID = 8;

****

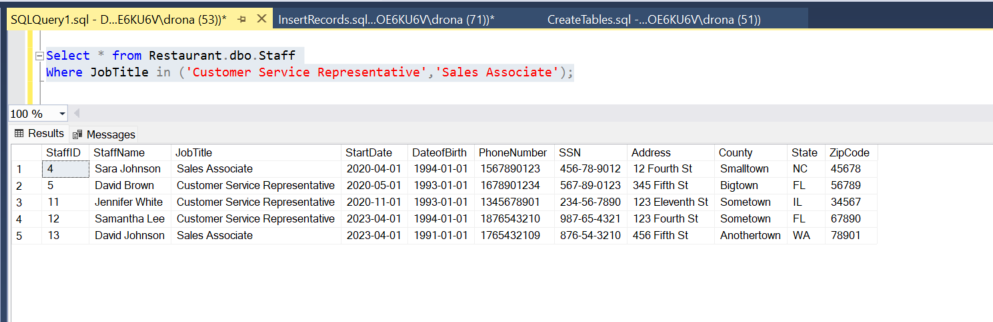
**Data Retrieval and Reports:**

1. To get the staff details whose Job title is Customer Service Representative and Sales Associative.
2. To get the staff details.

**Explanation-1:** It gives the records of staff with the job title Customer Service Representative and Sales Associative.

**Query-1:**

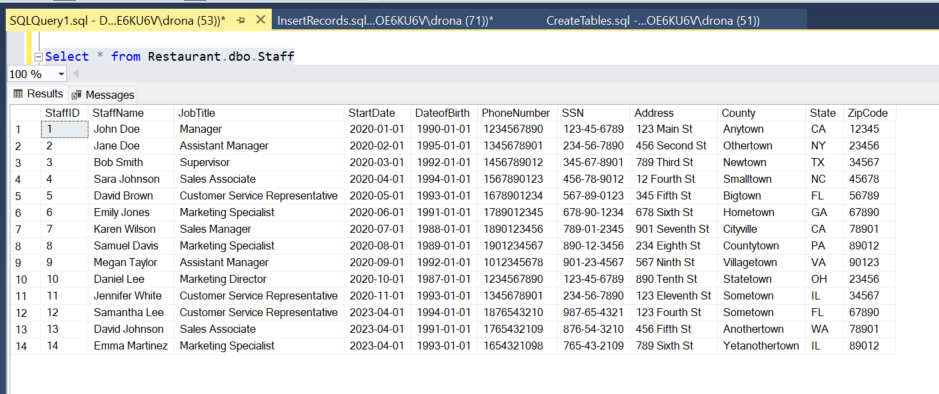
Select \* from Restaurant.dbo.Staff  
Where JobTitle in ('Customer Service Representative’, ‘Sales Associate');



**Explanation-2:** It gives the records of staff with the job title.

**Query-2:**

Select \* from Restaurant.dbo.Staff

****

**Creating Menu Table:**

**Explanation:** Creating the table menu to include all the details of the items with their descriptions and price.

**Query:**

CREATE TABLE [Menu] (

[MenuID] INT NOT NULL,

[MenuName] VARCHAR(100) NOT NULL,

[MenuDescription] VARCHAR(200) NOT NULL,

[MenuType] CHAR(32) NOT NULL,

[Price] INT NOT NULL,

PRIMARY KEY ([MenuID])

);

**Graphical user interface, text, application

Description automatically generated**

**Inserting records for Menu Table:**

**Explanation:** Inserting the records to the menu table include all the details of the items with their descriptions and price.

**Query:**

INSERT INTO Menu (MenuID, MenuName, MenuDescription, MenuType, Price) VALUES

(1, 'Cheeseburger', 'Classic burger with American cheese', 'Entree', 10),

(2, 'Spaghetti Bolognese', 'Spaghetti with meat sauce', 'Entree', 12),

(3, 'Caesar Salad', 'Romaine lettuce with croutons and Caesar dressing', 'Appetizer', 8),

(4, 'Margherita Pizza', 'Tomato sauce, mozzarella, and basil', 'Entree', 14),

(5, 'Fettuccine Alfredo', 'Fettuccine with creamy Parmesan sauce', 'Entree', 13),

(6, 'Mushroom Risotto', 'Arborio rice with mushrooms and Parmesan', 'Entree', 15),

(7, 'Garlic Bread', 'Toasted bread with garlic and butter', 'Appetizer', 5),

(8, 'Grilled Chicken', 'Grilled chicken breast with vegetables', 'Entree', 12),

(9, 'French Fries', 'Crispy fried potatoes', 'Side', 4),

(10, 'Onion Rings', 'Crispy fried onion rings', 'Side', 5),

(11, 'Tiramisu', 'Mascarpone cheese with ladyfingers and espresso', 'Dessert', 8),

(12, 'Chocolate Cake', 'Rich chocolate cake with chocolate ganache', 'Dessert', 9),

(13, 'Pesto Linguine', 'Linguine with basil pesto sauce', 'Entree', 13),

(14, 'Caprese Salad', 'Tomatoes, mozzarella, and basil', 'Appetizer', 9),

(15, 'Grilled Salmon', 'Grilled salmon with vegetables', 'Entree', 16),

(16, 'Chicken Caesar Salad', 'Romaine lettuce with grilled chicken and Caesar dressing', 'Entree', 11),

(17, 'Margarita Cocktail', 'Tequila, lime juice, and triple sec', 'Beverage', 8),

(18, 'Red Wine', 'Cabernet Sauvignon', 'Beverage', 12),

(19, 'Green Salad', 'Mixed greens with vegetables', 'Appetizer', 7),

(20, 'Bruschetta', 'Toasted bread with tomatoes and basil', 'Appetizer', 6),

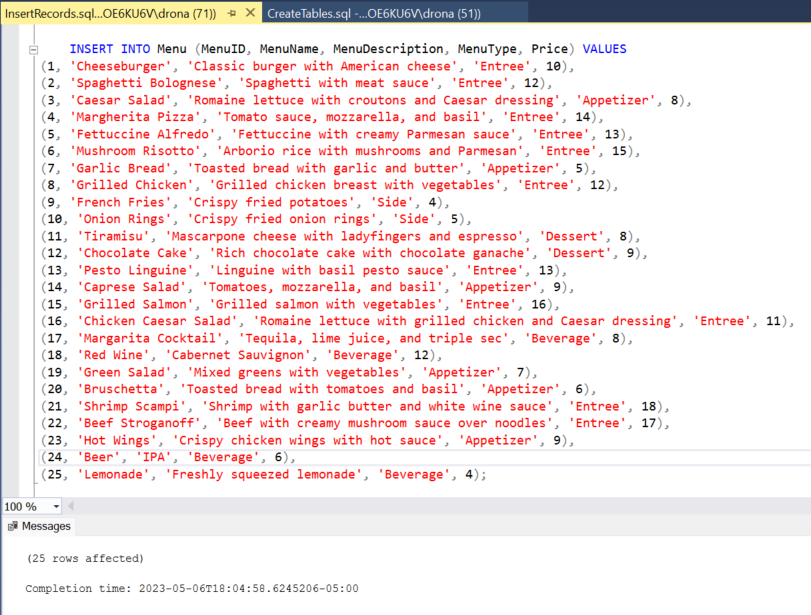
(21, 'Shrimp Scampi', 'Shrimp with garlic butter and white wine sauce', 'Entree', 18),

(22, 'Beef Stroganoff', 'Beef with creamy mushroom sauce over noodles', 'Entree', 17),

(23, 'Hot Wings', 'Crispy chicken wings with hot sauce', 'Appetizer', 9),

(24, 'Beer', 'IPA', 'Beverage', 6),

(25, 'Lemonade', 'Freshly squeezed lemonade', 'Beverage', 4);

****

**Data Retrieval and Reports:**

1. To get the Menu Items Where Menu type is Entrée with Price < 15
2. To get the Menu Items.

**Explanation-1:** It gives the records of Menu with the Menu type is Entrée and Price less than 15.

**Query-1:**

Select \* from Restaurant.dbo.Menu  
Where MenuType = 'Entree' AND Price < 15;

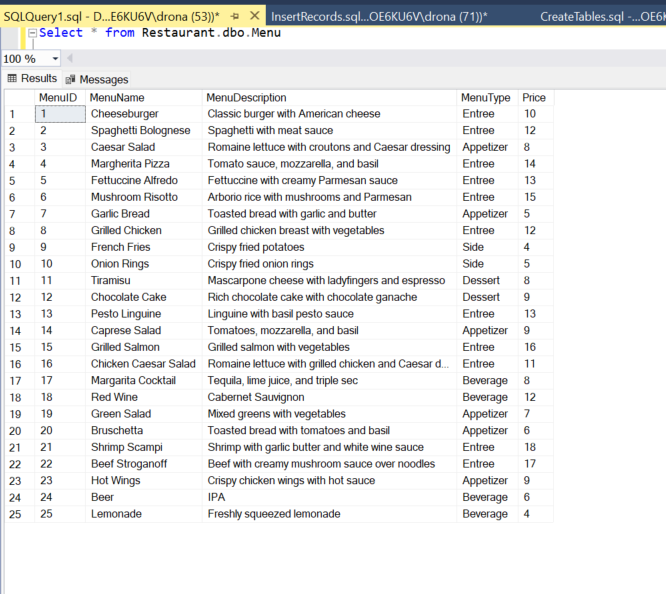
Graphical user interface, text, email

Description automatically generated

**Explanation-2:** It gives the records of all Menu items in the Menu.

**Query-2:**

Select \* from Restaurant.dbo.Menu

  
**Creating Customer Table:**

**Explanation:** Creating the table customer to include all the personal details and menu order.

**Query:**

CREATE TABLE [Customer] (

[CustID] INT NOT NULL,

[CustName] VARCHAR(100) NOT NULL,

[CustEmail] VARCHAR(100) NOT NULL,

[CustPhoneNo] INT NOT NULL,

[MenuID] INT NOT NULL,

PRIMARY KEY ([CustID]),

CONSTRAINT [FK\_Customer\_Menu]

FOREIGN KEY ([MenuID])

REFERENCES [Menu]([MenuID])

);

**Graphical user interface, text, application

Description automatically generated**

**Inserting records for Customer Table:**

**Explanation:** Inserting the records to the customer table include all the personal details and menu order.

**Query:**

INSERT INTO [Customer] ([CustID], [CustName], [CustEmail], [CustPhoneNo], [MenuID])

VALUES

(1, 'David', 'david@gmail.com', 1234567890,1),

(2, 'Jane', 'jane@gmail.com', 1345678901,8),

(3, 'Bob', 'bob@gmail.com', 1456789012,1),

(4, 'Sarah', 'sarah@gmail.com', 1567890123,7),

(5, 'Brown', 'brown@gmail.com', 1678901234,12),

(6, 'Laura', 'laura@gmail.com', 1789012345,1),

(7, 'Kevin', 'kevin@gmail.com', 1890123456,2),

(8, 'Samantha', 'samantha@gmail.com', 1901234567,15),

(9, 'Johnson', 'Johnson@gmail.com', 1012345678,18),

(10, 'Emily', 'emily@gmail.com', 0123456789,23),

(11, 'Brian', 'brian@gmail.com', 1234567890,24),

(12, 'Alexa', 'alexa@gmail.com', 1345678901,25),

(13, 'Jacob', 'jacob@gmail.com', 1456789012,1),

(14, 'Olivia', 'olivia@gmail.com', 1567890123,8),

(15, 'Emma', 'emma@gmail.com', 1678901234,7),

(16, 'Mason', 'mason@gmail.com', 1789012345,12),

(17, 'Ava', 'ava@gmail.com', 1890123456,1),

(18, 'Noah', 'noah@gmail.com', 1901234567,2),

(19, 'Sophia', 'sophia@gmail.com', 1012345678,15),

(20, 'Ethan', 'ethan@gmail.com', 0123456789,18),

(21, 'Isabella', 'isabella@gmail.com', 1234567890,23),

(22, 'Michael', 'michael@gmail.com', 1345678901,24),

(23, 'Avery', 'avery@gmail.com', 1456789012,25),

(24, 'Abigail', 'abigail@gmail.com', 1567890123,1),

(25, 'Mia', 'mia@gmail.com', 1678901234,20),

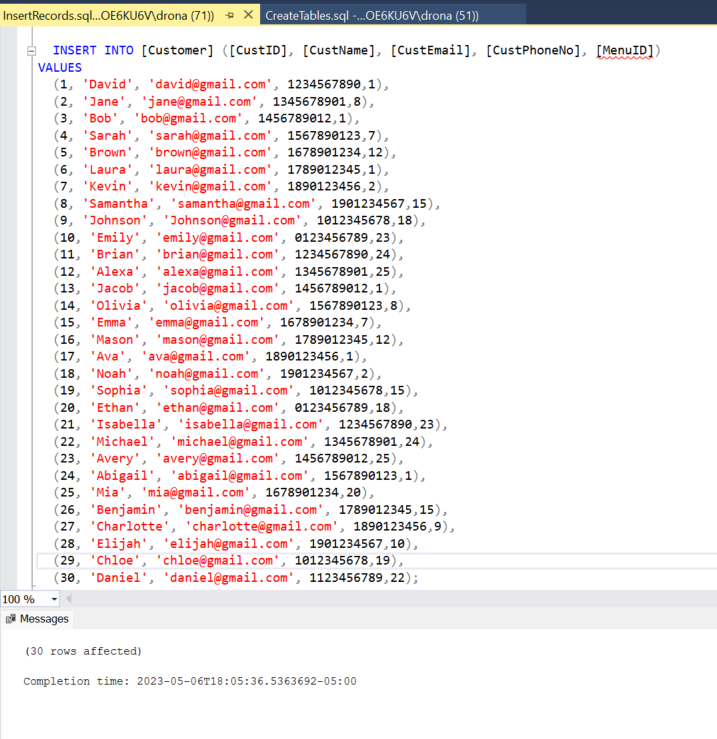
(26, 'Benjamin', 'benjamin@gmail.com', 1789012345,15),

(27, 'Charlotte', 'charlotte@gmail.com', 1890123456,9),

(28, 'Elijah', 'elijah@gmail.com', 1901234567,10),

(29, 'Chloe', 'chloe@gmail.com', 1012345678,19),

(30, 'Daniel', 'daniel@gmail.com', 1123456789,22);

****

**Creating Payment Table:**

**Explanation:** Creating the table payment to include all the payment details and date.

**Query:**

CREATE TABLE [Payment] (

[PaymentID] INT NOT NULL,

[StaffID] INT NOT NULL,

[PayDate] DATETIME NOT NULL,

[PaymentMode] CHAR(32) NOT NULL,

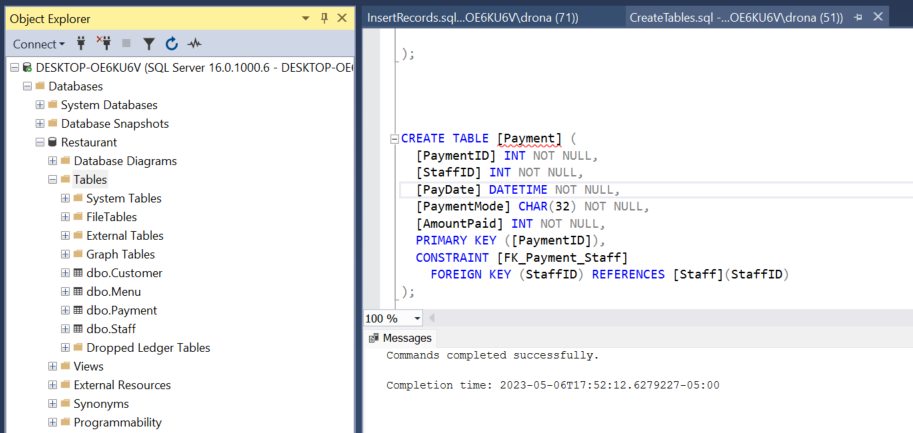
[AmountPaid] INT NOT NULL,

PRIMARY KEY ([PaymentID]),

CONSTRAINT [FK\_Payment\_Staff]

FOREIGN KEY (StaffID) REFERENCES [Staff](StaffID)

);

****

**Inserting records for Payment Table:**

**Explanation:** Inserting the records to the payment table include all the payment details and date.

**Query:**

INSERT INTO [dbo].[Payment]

([PaymentID]

,[StaffID]

,[PayDate]

,[PaymentMode]

,[AmountPaid])

VALUES

('1000001', 1, '2023-01-01', 'Credit Card', 5000.00),

('1000002', 2, '2023-01-01', 'Cash', 4000.00),

('1000003', 3, '2023-01-01', 'Debit Card', 3000.00),

('1000004', 4, '2023-01-01', 'Cheque', 2000.00),

('1000005', 5, '2023-01-01', 'Credit Card', 1800.00),

('1000006', 6, '2023-01-01', 'Cash', 2000.00),

('1000007', 7, '2023-01-01', 'Debit Card', 5000.00),

('1000008', 8, '2023-01-01', 'Cheque', 3000.00),

('1000009', 9, '2023-01-01', 'Credit Card', 4000.00),

('1000010', 10, '2023-01-01', 'Cash', 5500.00),

('1000011', 11, '2023-01-01', 'Debit Card', 1800.00),

('1000012', 12, '2023-01-01', 'Cheque', 1800.00),

('1000013', 13, '2023-01-01', 'Credit Card', 2000.00),

('1000014', 14, '2023-01-01', 'Cash', 2000.00),

('1000015', 1, '2023-02-01', 'Debit Card', 5000.00),

('1000016', 2, '2023-02-01', 'Cheque', 4000.00),

('1000017', 3, '2023-02-01', 'Credit Card', 3000.00),

('1000018', 4, '2023-02-01', 'Cash', 2000.00),

('1000019', 5, '2023-02-01', 'Debit Card', 1800.00),

('1000020', 6, '2023-02-01', 'Cheque', 2000.00),

('1000021', 7, '2023-02-01', 'Credit Card', 5000.00),

('1000022', 8, '2023-02-01', 'Cash', 3000.00),

('1000023', 9, '2023-02-01', 'Debit Card', 4000.00),

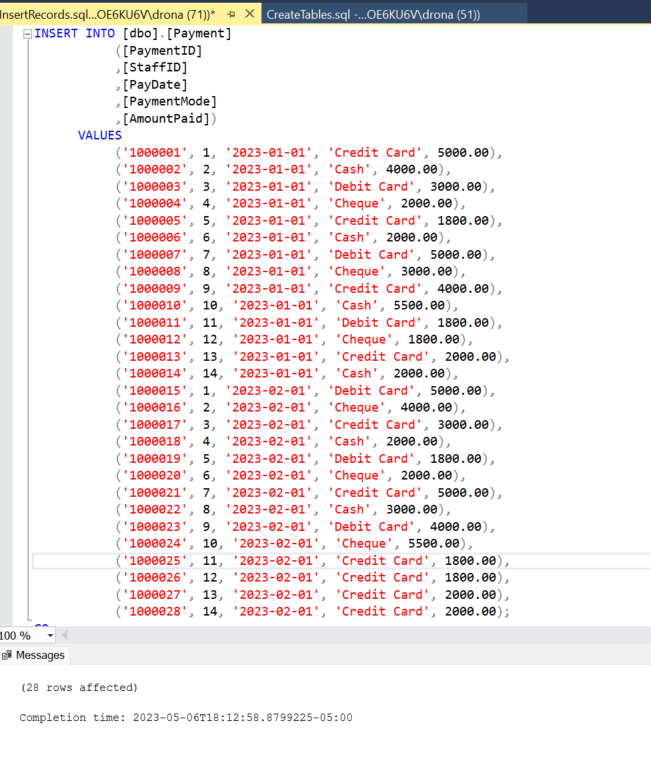
('1000024', 10, '2023-02-01', 'Cheque', 5500.00),

('1000025', 11, '2023-02-01', 'Credit Card', 1800.00),

('1000026', 12, '2023-02-01', 'Credit Card', 1800.00),

('1000027', 13, '2023-02-01', 'Credit Card', 2000.00),

('1000028', 14, '2023-02-01', 'Credit Card', 2000.00);

****

**Creating Transaction Table:**

**Explanation:** Creating the table transaction to include all the details of the transactions with datetime.

**Query:**

CREATE TABLE [Transaction] (

[TransactionID] INT NOT NULL,

[Amount] INT NOT NULL,

[PaymentMethod] CHAR(32) NOT NULL,

[Transationdate] DATETIME NOT NULL,

PRIMARY KEY ([TransactionID])

);

****

**Inserting records for Transaction Table:**

**Explanation:** Inserting the records to the transaction table including all the details of the transactions with datetime.

**Query:**

INSERT INTO [dbo].[Transaction]

([TransactionID]

,[Amount]

,[PaymentMethod]

,[Transationdate])

VALUES

(1, 400, 'Cash', '2023-01-01 10:35:00'),

(2, 175, 'Credit Card', '2023-01-02 15:00:00'),

(3, 250, 'Debit Card', '2023-01-03 12:00:00'),

(4, 600, 'Credit Card', '2023-01-04 16:20:00'),

(5, 1000, 'Credit Card', '2023-01-05 18:30:00'),

(6, 175, 'Debit Card', '2023-01-06 09:15:00'),

(7, 800, 'Cash', '2023-01-07 14:00:00'),

(8, 75, 'Credit Card', '2023-01-08 17:30:00'),

(9, 600, 'Debit Card', '2023-01-09 11:45:00'),

(10, 560, 'Cash', '2023-01-10 13:20:00'),

(11, 235, 'Credit Card', '2023-01-11 16:45:00'),

(12, 230, 'Debit Card', '2023-01-12 10:15:00'),

(13, 160, 'Cash', '2023-01-13 13:00:00'),

(14, 275, 'Credit Card', '2023-01-14 17:00:00'),

(15, 360, 'Cash', '2023-01-15 19:45:00'),

(16, 670, 'Cash', '2023-01-16 12:30:00'),

(17, 475, 'Credit Card', '2023-01-17 15:45:00'),

(18, 160, 'Debit Card', '2023-01-18 09:00:00'),

(19, 150, 'Cash', '2023-01-19 14:15:00'),

(20, 275, 'Credit Card', '2023-01-20 18:30:00'),

(21, 100, 'Debit Card', '2023-01-21 10:45:00'),

(22, 120, 'Cash', '2023-01-22 13:00:00'),

(23, 80, 'Credit Card', '2023-01-23 16:15:00'),

(24, 200, 'Debit Card', '2023-01-24 19:30:00'),

(25, 510, 'Cash', '2023-01-25 12:45:00'),

(26, 50, 'Credit Card', '2023-01-26 15:00:00'),

(27, 175, 'Debit Card', '2023-01-27 18:15:00'),

(28, 160, 'Cash', '2023-01-28 11:30:00'),

(29, 175, 'Debit Card', '2023-01-29 14:45:00'),

(30, 100, 'Debit Card', '2023-01-11 14:45:00'),

(31, 200, 'Cash', '2023-01-12 14:45:00'),

(32, 400, 'Debit Card', '2023-01-13 12:45:00'),

(33, 600, 'Cash', '2023-01-14 14:45:00'),

(34, 1000, 'Debit Card', '2023-01-15 11:45:00'),

(35, 600, 'Debit Card', '2023-01-16 12:45:00'),

(36, 100, 'Debit Card', '2023-01-17 14:45:00'),

(37, 200, 'Cash', '2023-01-18 14:45:00'),

(38, 400, 'Debit Card', '2023-01-19 10:45:00'),

(39, 600, 'Debit Card', '2023-01-20 09:45:00'),

(40, 1000, 'Cash', '2023-01-21 14:45:00'),

(41, 600, 'Debit Card', '2023-01-22 12:45:00'),

(42, 100, 'Cash', '2023-01-23 14:45:00'),

(43, 200, 'Debit Card', '2023-01-24 14:45:00'),

(1001, 20,'Debit Card','2023-01-01 10:40:00'),

(1002,36,'Credit Card','2023-01-01 10:55:00'),

(1003, 10,'Credit Card','2023-01-01 11:05:00'),

(1004, 40,'Debit Card','2023-01-02 10:05:00'),

(1005, 20,'Debit Card','2023-01-02 10:40:00'),

(1006,36,'Credit Card','2023-01-03 10:55:00'),

(1007, 10,'Credit Card','2023-01-03 11:05:00'),

(1008, 40,'Debit Card','2023-01-03 10:05:00'),

(1009, 20,'Debit Card','2023-01-03 11:40:00'),

(1010,36,'Credit Card','2023-01-03 11:55:00'),

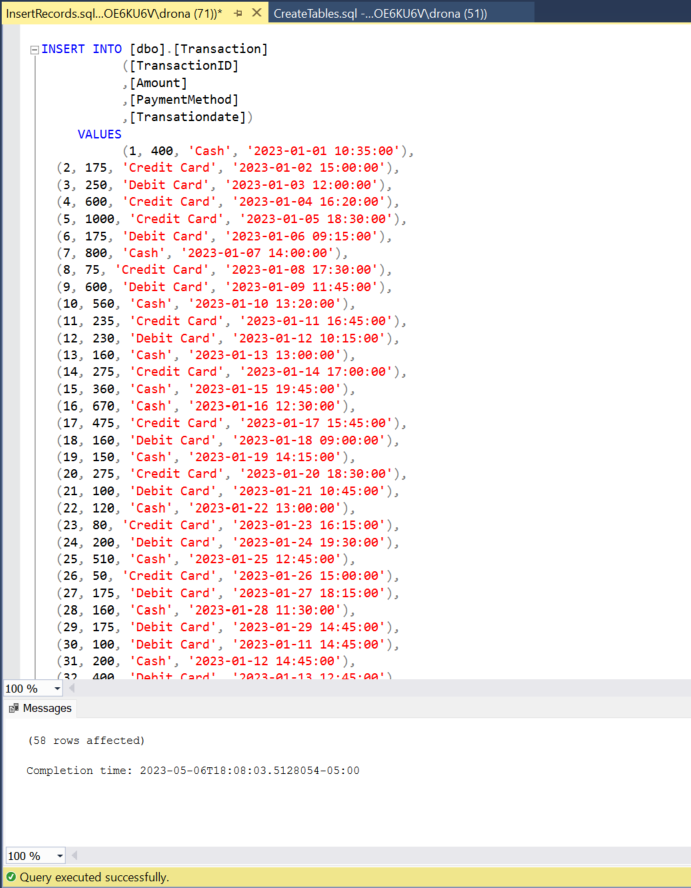
(1011, 10,'Credit Card','2023-01-03 12:05:00'),

(1012, 40,'Debit Card','2023-01-04 10:05:00'),

(1013,36,'Credit Card','2023-01-04 11:55:00'),

(1014, 40,'Debit Card','2023-01-04 10:05:00'),

(1015, 10,'Credit Card','2023-01-04 12:05:00');

****

**Creating Valet Parking Table:**

**Explanation:** Creating the table valet parking to include all the vehicle number and staff details.

**Query:**

CREATE TABLE [Valet Parking] (

[VehicleNo] CHAR(32) NOT NULL,

[CustID] INT NOT NULL,

[StaffID] INT NOT NULL,

PRIMARY KEY ([VehicleNo]),

CONSTRAINT [FK\_Valet\_Customer]

FOREIGN KEY ([CustID])

REFERENCES [Customer]([CustID]),

CONSTRAINT [FK\_Valet\_Staff]

FOREIGN KEY ([StaffID])

REFERENCES [Staff]([StaffID])

);

****

**Inserting records for Valet Parking Table:**

**Explanation:** Inserting the records to the valet parking table including all the vehicle number and staff details.

**Query:**

INSERT INTO [Valet Parking] ([VehicleNo], [CustID], [StaffID])

VALUES ('ABC123', 1, 3),

('DEF456', 2, 8),

('GHI789', 1, 3),

('JKL012', 4, 8),

('MNO345', 15, 3),

('PQR678', 6, 3),

('STU901', 17, 8),

('VWX234', 8, 3),

('YZA567', 9, 3),

('BCD890', 10, 3),

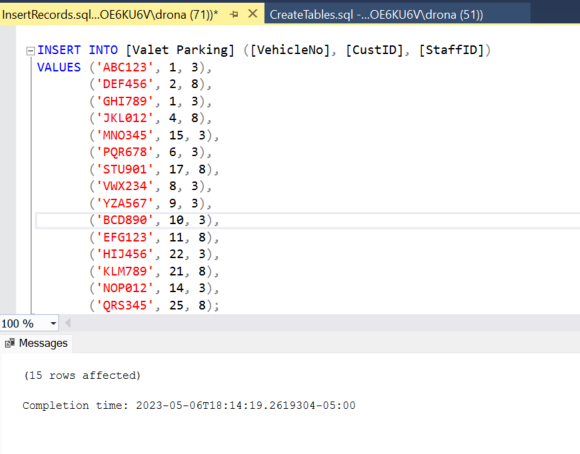
('EFG123', 11, 8),

('HIJ456', 22, 3),

('KLM789', 21, 8),

('NOP012', 14, 3),

('QRS345', 25, 8);

****

**Creating Order Table:**

**Explanation:** Creating the table order to include all the order details from the customers.

**Query:**

CREATE TABLE [Order] (

[OrderID] INT NOT NULL,

[CustID] INT NOT NULL,

[OrderDate] DATETIME NOT NULL,

[OrderType] CHAR(32) NOT NULL,

[Quantity] INT NOT NULL,

[Price] INT NOT NULL,

[TransactionID] INT NOT NULL,

[StaffID] INT NOT NULL,

PRIMARY KEY ([OrderID]),

CONSTRAINT [FK\_Order\_Customer]

FOREIGN KEY ([CustID])

REFERENCES [Customer]([CustID]),

CONSTRAINT [FK\_Order\_Staff]

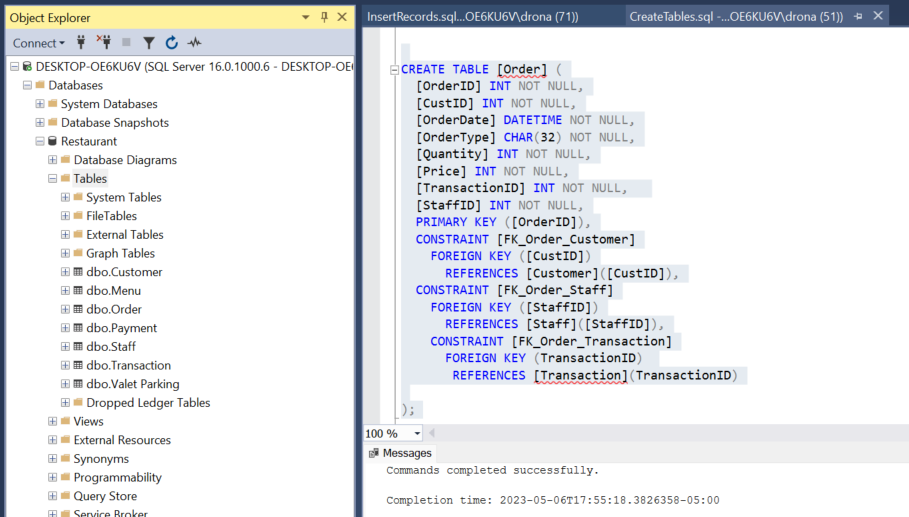
FOREIGN KEY ([StaffID])

REFERENCES [Staff]([StaffID]),

CONSTRAINT [FK\_Order\_Transaction]

FOREIGN KEY (TransactionID)

REFERENCES [Transaction](TransactionID));

****

**Inserting records for Order Table:**

**Explanation:** Inserting the records to the order table including all the order details from the customers.

**Query:**

INSERT INTO [dbo].[Order]

([OrderID]

,[CustID]

,[OrderDate]

,[OrderType]

,[Quantity]

,[Price]

,[TransactionID]

,[StaffID])

VALUES

(1, 1, '2023-01-01', 'Takeout', 2, 20, 1001, 5),

(2, 6, '2023-01-01', 'Dine-in', 3, 36, 1002, 11),

(3, 3, '2023-01-01', 'Takeout', 1, 10, 1003, 12),

(4, 4, '2023-01-02', 'Delivery', 4, 40, 1004, 5),

(5, 5, '2023-01-02', 'Takeout', 2, 20, 1005, 5),

(6, 1, '2023-01-03', 'Dine-in', 3, 36, 1006, 11),

(7, 6, '2023-01-03', 'Takeout', 1, 10, 1007, 12),

(8, 3, '2023-01-03', 'Delivery', 4, 40, 1008, 5),

(9, 7, '2023-01-03', 'Delivery', 2, 20, 1009, 11),

(10, 8, '2023-01-03', 'Delivery', 3, 36, 1010, 12),

(11, 9, '2023-01-03', 'Takeout', 1, 10, 1011, 5),

(12, 10, '2023-01-04', 'Dine-in', 4, 40, 1012, 11),

(13, 1, '2023-01-04', 'Takeout', 3, 36, 1013, 12),

(14, 4, '2023-01-04', 'Delivery', 4, 40, 1014, 5),

(15, 5, '2023-01-04', 'Delivery', 1, 10, 1015, 12);

**Graphical user interface, application

Description automatically generated**

**Creating Delivery Table:**

**Explanation:** Creating the table delivery to include all the delivery details and their order

**Query:**

CREATE TABLE [Delivery] (

[DeliveryID] INT NOT NULL,

[DeliveryStatus] CHAR(32) NOT NULL,

[OrderID] INT NOT NULL,

PRIMARY KEY ([DeliveryID]),

CONSTRAINT [FK\_Delivery\_Order]

FOREIGN KEY ([OrderID])

REFERENCES [Order]([OrderID])

);

****

**Inserting records for Delivery Table:**

**Explanation:** Inserting the records to the delivery table including all the delivery details and their order**.**

**Query:**

INSERT INTO [dbo].[Delivery]

([DeliveryID]

,[DeliveryStatus]

,[OrderID])

VALUES

(1, 'DELIVERED', 1),

(2, 'PROCESSING', 3),

(3, 'DELIVERED', 4),

(4, 'DELIVERED', 5),

(5, 'PROCESSING', 7),

(6, 'DELIVERED', 8),

(7, 'DELIVERED', 9),

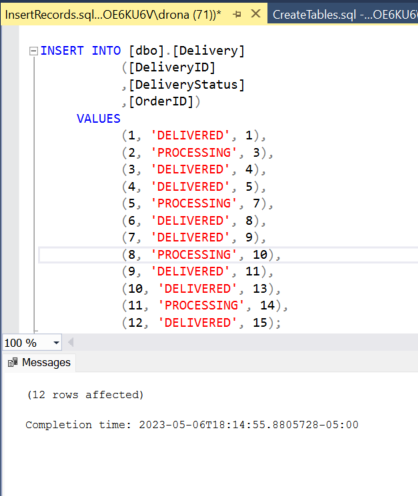
(8, 'PROCESSING', 10),

(9, 'DELIVERED', 11),

(10, 'DELIVERED', 13),

(11, 'PROCESSING', 14),

(12, 'DELIVERED', 15);

****

**Creating Banquet Halls Table:**

**Explanation:** Creating the table staff to include all the details and their job title.

**Query:**

CREATE TABLE [BanquetHalls] (

[HallID] INT NOT NULL,

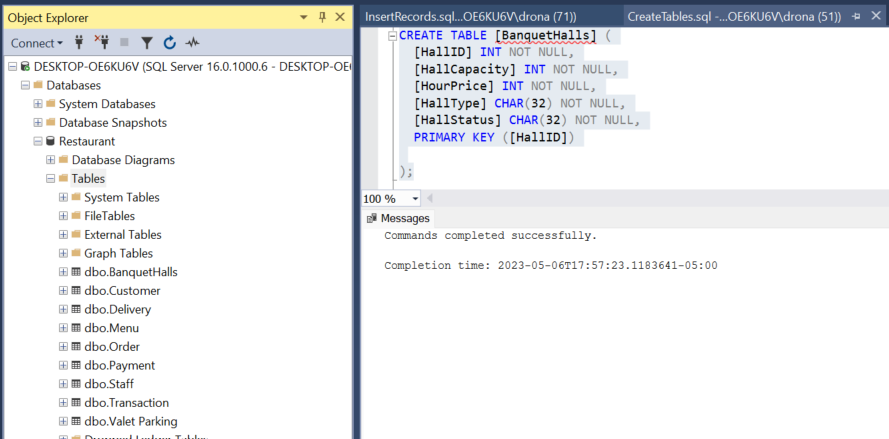
[HallCapacity] INT NOT NULL,

[HourPrice] INT NOT NULL,

[HallType] CHAR(32) NOT NULL,

[HallStatus] CHAR(32) NOT NULL,

PRIMARY KEY ([HallID]));

****

**Inserting records for Banquet Halls Table:**

**Explanation:** Inserting the records to the customer table include all the personal details and menu order.

**Query:**

INSERT INTO BanquetHalls (HallID, HallCapacity, HourPrice, HallType, HallStatus)

VALUES

(1, 50, 100, 'Grand Room', 'Available'),

(2, 100, 200, 'Outdoor Terrace', 'Reserved'),

(3, 45, 75, 'Garden Room', 'Available'),

(4, 75, 150, 'Crystal Room', 'Occupied'),

(5, 60, 120, 'Diamond Hall', 'Reserved'),

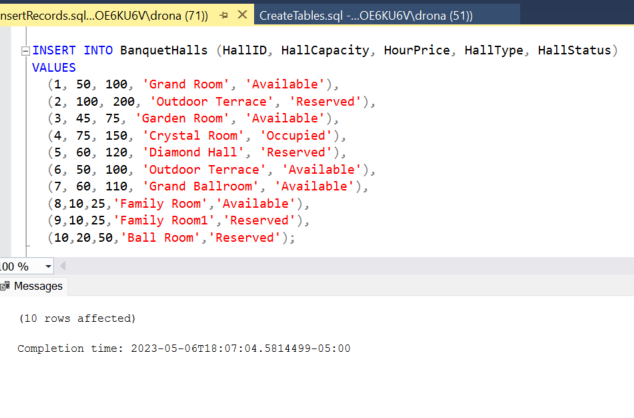
(6, 50, 100, 'Outdoor Terrace', 'Available'),

(7, 60, 110, 'Grand Ballroom', 'Available'),

(8,10,25,'Family Room','Available'),

(9,10,25,'Family Room1','Reserved'),

(10,20,50,'Ball Room','Reserved');

****

**Creating Banquet Booking Table:**

**Explanation:** Creating the table banquet booking to include all the details about the banquet booking with booking hours and price.

**Query:**

CREATE TABLE [BanquetBooking] (

[BookingID] INT NOT NULL,

[BookingDate] DATE NOT NULL,

[BookingTime] TIME NOT NULL,

[HoursBooked] INT NOT NULL,

[Price] INT NOT NULL,

[HallID] INT NOT NULL,

[TransactionID] INT NOT NULL,

[CustID] INT NOT NULL,

[StaffID] INT NOT NULL,

PRIMARY KEY ([BookingID]),

CONSTRAINT [FK\_BanquetBooking.CustID]

FOREIGN KEY ([CustID])

REFERENCES [Customer]([CustID]),

CONSTRAINT [FK\_BanquetBooking.StaffID]

FOREIGN KEY ([StaffID])

REFERENCES [Staff]([StaffID]),

CONSTRAINT [FK\_BanquetBooking.HallID]

FOREIGN KEY ([HallID])

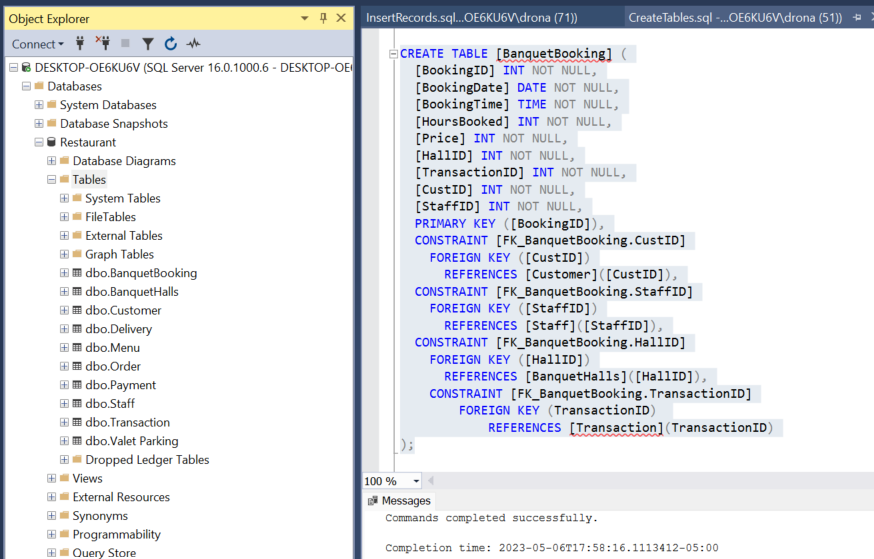
REFERENCES [BanquetHalls]([HallID]),

CONSTRAINT [FK\_BanquetBooking.TransactionID]

FOREIGN KEY (TransactionID)

REFERENCES [Transaction](TransactionID)

);

****

**Inserting records for Banquet Booking Table:**

**Explanation:** Inserting the records to the banquet booking table include all the details about the banquet booking with booking hours and price.

**Query:**

INSERT INTO [dbo].[BanquetBooking] ([BookingID], [BookingDate], [BookingTime], [HoursBooked], [Price], [HallID], [TransactionID], [CustID], [StaffID])

VALUES

(1, '2023-01-01', '14:00:00', 4, 400, 1, 1, 1, 1),

(2, '2023-01-04', '12:00:00', 3, 600, 2, 4, 4, 2),

(3, '2023-01-06', '15:00:00', 5, 1000, 2, 5, 3, 9),

(4, '2023-01-10', '16:00:00', 6, 600, 6, 9, 8, 1),

(5, '2023-01-12', '10:00:00', 4, 100, 8, 30, 5, 2),

(6, '2023-01-13', '13:00:00', 4, 200, 10, 31, 6, 9),

(7, '2023-01-14', '11:00:00', 4, 400, 1, 32, 1, 1),

(8, '2023-01-15', '14:00:00', 3, 600, 2,33, 8, 2),

(9, '2023-01-16', '15:00:00', 5, 1000, 2, 34, 9, 9),

(10, '2023-01-17', '12:00:00', 6, 600, 6, 35, 10, 1),

(11, '2023-01-18', '16:00:00', 4, 100, 8, 36, 11, 2),

(12, '2023-01-19', '13:00:00', 4, 200, 10, 37, 12, 9),

(13, '2023-01-20', '11:00:00', 4, 400, 1, 38, 5, 1),

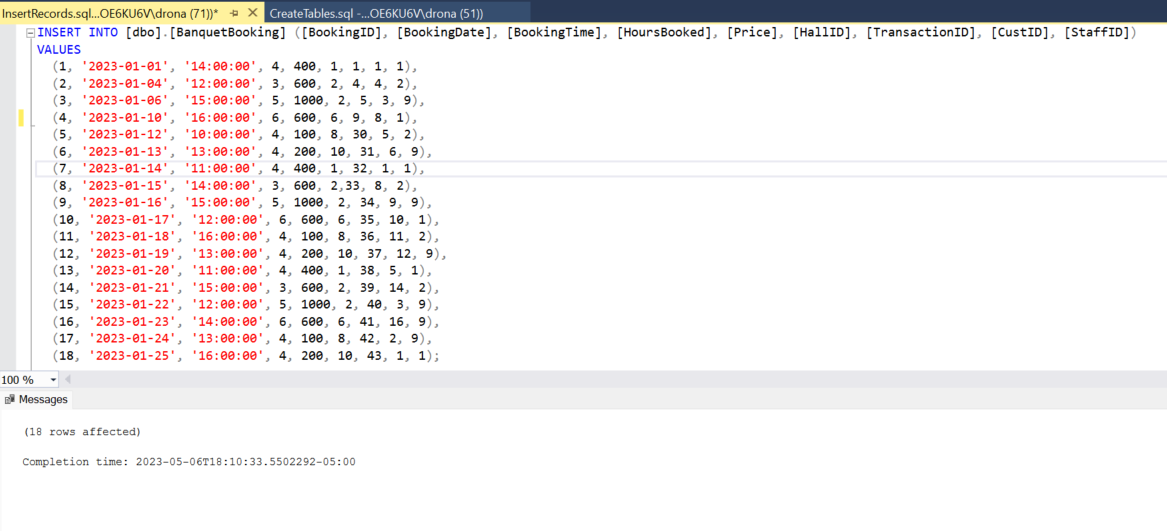
(14, '2023-01-21', '15:00:00', 3, 600, 2, 39, 14, 2),

(15, '2023-01-22', '12:00:00', 5, 1000, 2, 40, 3, 9),

(16, '2023-01-23', '14:00:00', 6, 600, 6, 41, 16, 9),

(17, '2023-01-24', '13:00:00', 4, 100, 8, 42, 2, 9),

(18, '2023-01-25', '16:00:00', 4, 200, 10, 43, 1, 1);

**Creating Table Tables:**

**Explanation:** Creating the table tables to include all the details about the tables and their availability**.**

**Query:**

CREATE TABLE [Tables] (

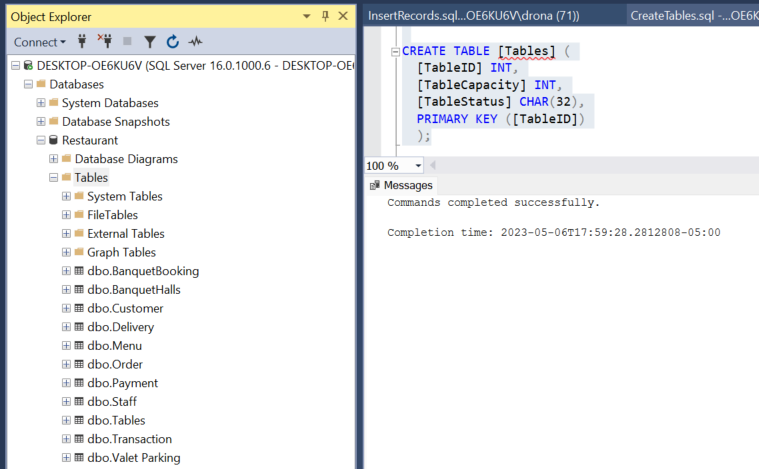
[TableID] INT,

[TableCapacity] INT,

[TableStatus] CHAR(32),

PRIMARY KEY ([TableID])

);

****

**Inserting records for Tables Table:**

**Explanation:** Inserting the records to the tables table include all the details about the tables and their availability**.**

**Query:**

INSERT INTO Tables(TableID, TableCapacity, TableStatus) VALUES

(1, 4, 'Available'),

(2, 6, 'Occupied'),

(3, 2, 'Available'),

(4, 8, 'Available'),

(5, 4, 'Occupied'),

(6, 6, 'Available'),

(7, 2, 'Available'),

(8, 8, 'Occupied'),

(9, 4, 'Available'),

(10, 6, 'Available'),

(11, 2, 'Occupied'),

(12, 8, 'Available'),

(13, 4, 'Available'),

(14, 6, 'Occupied'),

(15, 2, 'Available'),

(16,2,'Available'),

(17,2,'Available'),

(18,4,'Available'),

(19,6,'Available'),

(20,10,'Available'),

(21,8,'Available'),

(22,8,'Available'),

(23,2,'Available'),

(24,2,'Available'),

(25,2,'Available'),

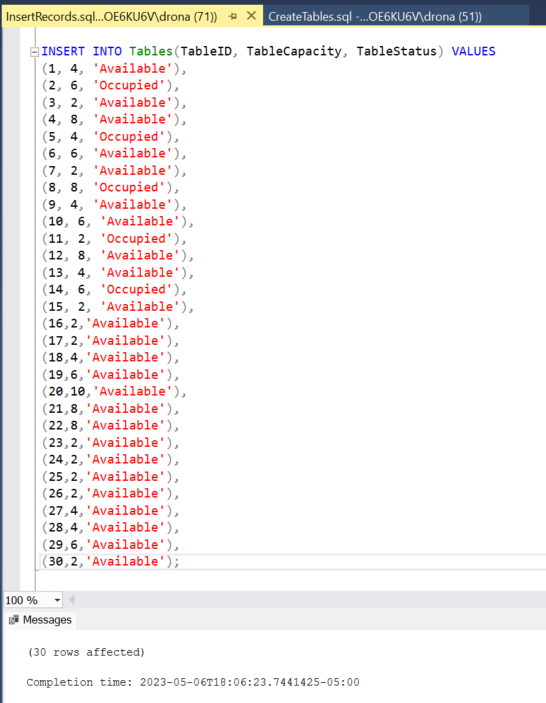
(26,2,'Available'),

(27,4,'Available'),

(28,4,'Available'),

(29,6,'Available'),

(30,2,'Available');

****

**Creating Table Bookings Table:**

**Explanation:** Creating the table booking table to include all the booking details with date and time**.**

**Query:**

CREATE TABLE [TableBooking] (

[TableBookingID] INT,

[CustID] INT,

[BookingDate] DATE,

[BookingTime] TIME,

[TableID] INT,

[StaffID] INT,

[Capacity] INT,

PRIMARY KEY ([TableBookingID]),

CONSTRAINT [FK\_TableBooking.CustID]

FOREIGN KEY ([CustID])

REFERENCES [Customer]([CustID]),

CONSTRAINT [FK\_TableBooking.StaffID]

FOREIGN KEY ([StaffID])

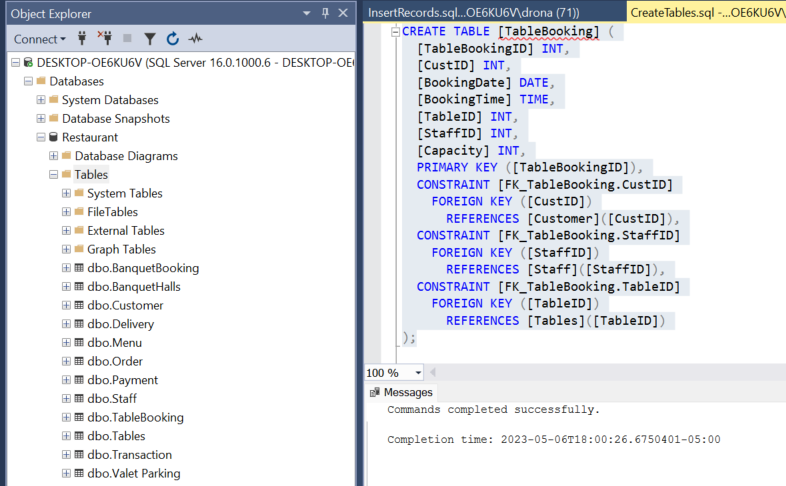
REFERENCES [Staff]([StaffID]),

CONSTRAINT [FK\_TableBooking.TableID]

FOREIGN KEY ([TableID])

REFERENCES [Tables]([TableID])

);

****

**Inserting records for Table Bookings Table:**

**Explanation:** Inserting the records to the table booking table including all the booking details with date and time**.**

**Query:**

INSERT INTO [dbo].[TableBooking]

([TableBookingID]

,[CustID]

,[BookingDate]

,[BookingTime]

,[TableID]

,[StaffID]

,[Capacity])

VALUES

(101,9, '2023-01-01', '11:00:00', 4, 7, 8),

(102,10, '2023-01-01', '11:30:00', 8, 7, 8),

(103,15, '2023-01-01', '12:30:00', 12, 7, 8),

(104,16, '2023-01-02', '16:00:00', 15, 7, 2),

(105,9, '2023-01-02', '17:00:00', 4, 7, 8),

(106,10, '2023-01-02', '18:30:00', 8, 7, 8),

(107,15, '2023-01-03', '12:30:00', 12, 7, 8),

(108,16, '2023-01-03', '13:00:00', 15, 7, 2),

(109,9, '2023-01-03', '17:00:00', 4, 7, 8),

(110,10, '2023-01-03', '18:30:00', 8, 7, 8),

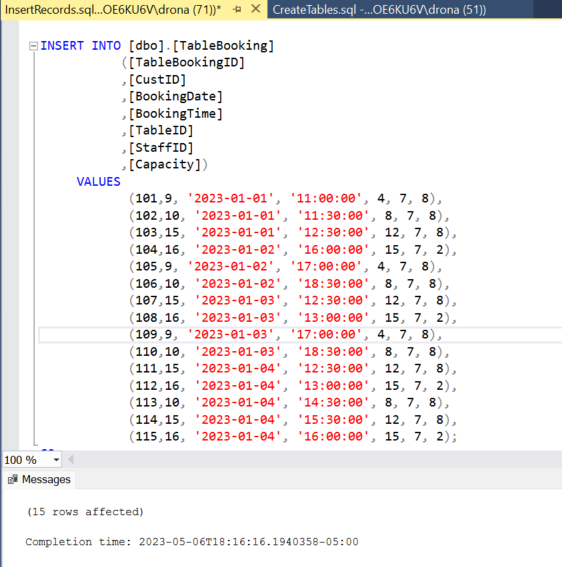
(111,15, '2023-01-04', '12:30:00', 12, 7, 8),

(112,16, '2023-01-04', '13:00:00', 15, 7, 2),

(113,10, '2023-01-04', '14:30:00', 8, 7, 8),

(114,15, '2023-01-04', '15:30:00', 12, 7, 8),

(115,16, '2023-01-04', '16:00:00', 15, 7, 2);

****

**Data Retrieval and Simple Reports:**

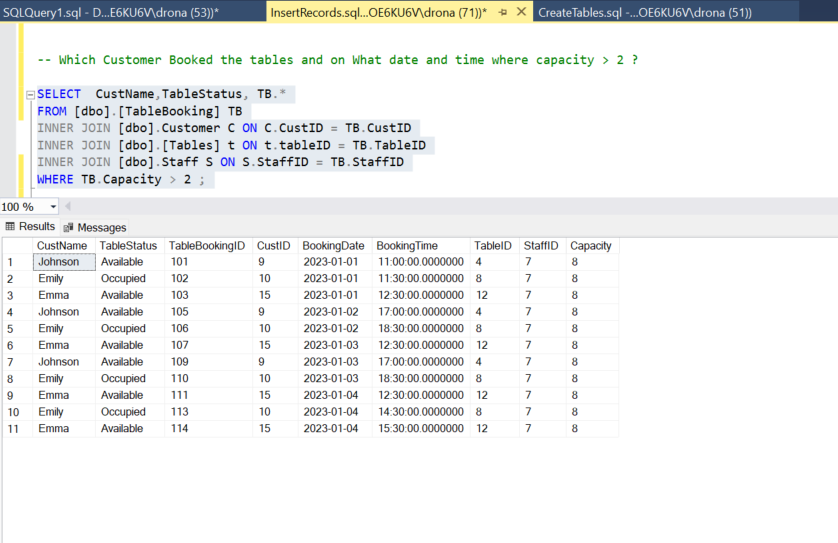
1. Which Customer Booked the tables and on What date and time where capacity > 2?
2. How Frequently Banquet hall - Outdoor Terrace is booked and What the amount paid with the Hall Capacity?
3. What is the count of Different type of Delivery Orders?
4. What is the Total revenue in a month?
5. Which Staff is looking after the Dine in Orders?
6. Which Staff is looking after the Banquet Orders?

**Explanation-1:** Here we tried to retrieve the customer who had booked the tables with more than capacity 2 on which date and time.

**Query-1:**

SELECT  CustName,TableStatus, TB.\*  
FROM [dbo].[TableBooking] TB  
INNER JOIN [dbo].Customer C ON C.CustID = TB.CustID  
INNER JOIN [dbo].[Tables] t ON t.tableID = TB.TableID  
INNER JOIN [dbo].Staff S ON S.StaffID = TB.StaffID  
WHERE TB.Capacity > 2 ;

**Results-1:**

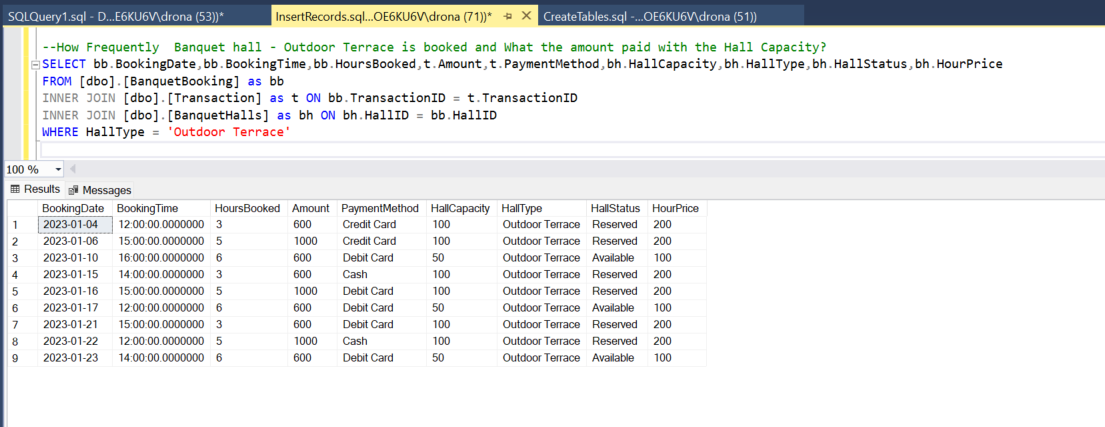
****

**Explanation-2:** Here we tried to retrieve the frequently booked outdoor terrace banquet halls, along with the capacity and the amount paid for it.

**Query-2:**

SELECT bb.BookingDate,bb.BookingTime,bb.HoursBooked,t.Amount,t.PaymentMethod,bh.HallCapacity,bh.HallType,bh.HallStatus,bh.HourPrice  
FROM [dbo].[BanquetBooking] as bb  
INNER JOIN [dbo].[Transaction] as t ON bb.TransactionID = t.TransactionID  
INNER JOIN [dbo].[BanquetHalls] as bh ON bh.HallID = bb.HallID  
WHERE HallType = 'Outdoor Terrace';

**Results-2:**

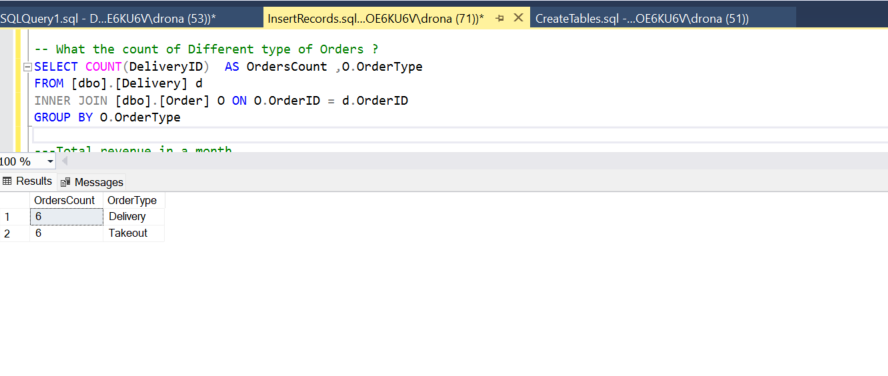
****

**Explanation-3:** Here we tried to retrieve the count of the distinct types of Delivery orders.

**Query-3:**

SELECT COUNT(DeliveryID)  AS OrdersCount ,O.OrderType   
FROM [dbo].[Delivery] d  
INNER JOIN [dbo].[Order] O ON O.OrderID = d.OrderID  
GROUP BY O.OrderType;

**Results-3:**

****

**Explanation-4:** Here we tried to retrieve the Total revenue in a month from the Transactions.

**Query-4:**

Select SUM(t.Amount) as Revenue, MONTH(t.TransationDate) AS TransationMonth  
From Restaurant.dbo.[Transaction] t  
Group by MONTH(t.Transationdate);

**Results-4:**

**Graphical user interface, application, Word

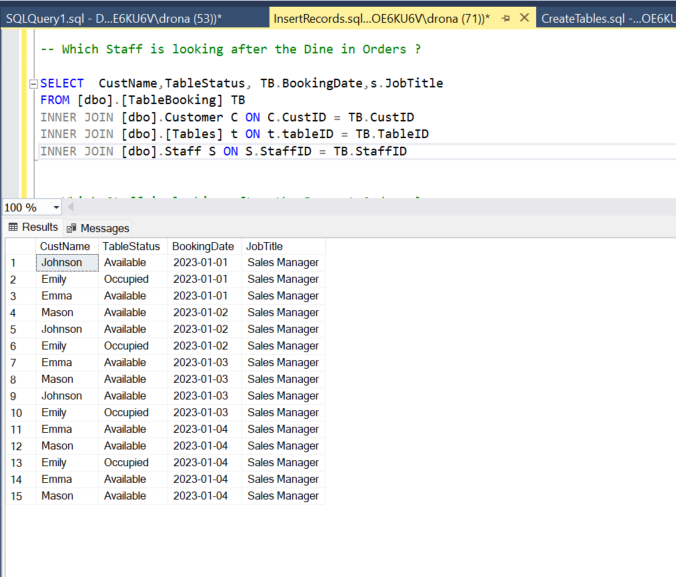
Description automatically generated**

**Explanation-5:** Here we tried to retrieve the staff looking after the Dine in Orders.

**Query-5:**

SELECT  CustName,TableStatus, TB.BookingDate,s.JobTitle  
FROM [dbo].[TableBooking] TB  
INNER JOIN [dbo].Customer C ON C.CustID = TB.CustID  
INNER JOIN [dbo].[Tables] t ON t.tableID = TB.TableID  
INNER JOIN [dbo].Staff S ON S.StaffID = TB.StaffID;

**Results-5:**

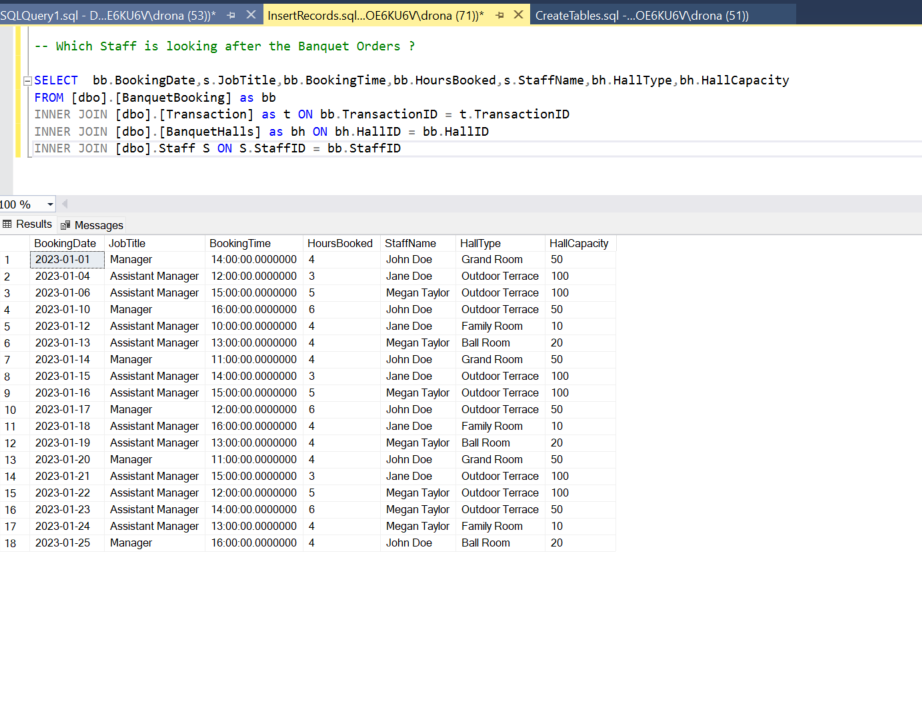
****

**Explanation-6:** Here we tried to retrieve the staff looking after the Banquet Orders.

**Query-6:**

SELECT  bb.BookingDate,s.JobTitle,bb.BookingTime,bb.HoursBooked,s.StaffName,bh.HallType,bh.HallCapacity  
FROM [dbo].[BanquetBooking] as bb  
INNER JOIN [dbo].[Transaction] as t ON bb.TransactionID = t.TransactionID  
INNER JOIN [dbo].[BanquetHalls] as bh ON bh.HallID = bb.HallID  
INNER JOIN [dbo].Staff S ON S.StaffID = bb.StaffID;

**Results-6:**



**Conclusion:**

As a result, Restaurant operations is crucial in determining whether a restaurant revenue from sales increases or decreases. It is also used for the numerous day- to-day operations to perform smoothly by allocating staff with their respective job titles. This enables management to speed up their operations and provide the best customer service using this restaurant management system. This also covers the table booking, banquet booking for the users to simplify their interactions without any wait time.