Case Study Article on Restaurant business

Heritage Bites Restaurant

Written by Sahil Thummar
Professor Dr. Junaid Qazi

1. Introduction

1.1 Purpose of the Database Design

The purpose of this case study is to design a comprehensive database for Heritage Bites Restaurant, a small restaurant with 8-9 tables. The goal is to streamline key operations such as customer management, reservations, orders, and inventory management to improve overall restaurant efficiency and customer satisfaction.

1.2 Overview of Heritage Bites Restaurant

Heritage Bites Restaurant is a cozy dining establishment focused on providing quality food with locally sourced ingredients. With a limited number of tables and a strong emphasis on customer service, the restaurant needs a reliable system to manage its operations effectively.

2. Mission and Objectives

2.1 Mission Statement

To be leading southeast ethnic restaurant that ensure to deliver delightful dining experience.

2.2 Business Objectives

- Efficient Table Management– Minimize wait times and maximize seating utilization.
- Order Tracking – Ensure timely preparation and delivery of food.
- Inventory Management– Maintain stock for menu items and reduce waste.
- Customer Data– Personalize services to enhance the dining experience.

3. Database Design Overview

Database design is important because it helps keep data organized and easy to use. It makes things run smoothly, like tracking orders and customers, and helps avoid errors. This is key for a business to work efficiently.

3.1 Identified Tables

Based on the restaurant's operations, eight core tables were identified for the database:

- Customers
- Employees
- Reservations
- Tables
- Menu_Items
- Orders

- Order_Details
- Payments

3.2 Focus Areas (Reservations, Orders)

The database will focus on efficiently handling:

- Customer reservations, including time, table, and guest count.
- Orders placed by customers, tracked by employee and table.

4. Tables and Their Attribute

4.1 Customers Table

Description: Stores customer details for contact and loyalty tracking. Used for managing customer interactions and personalized services.

Fields: Customer_ID, First_Name, Last_Name, Phone, Email

4.2 Employees Table

Description: Holds employee data, including role and salary. Helps with shift scheduling, payroll, and staff management.

Fields: Employee_ID, First_Name, Last_Name, Role, Shift_Time, Salary

4.3 Reservations Table

Description: Manages reservation information, linking customers to tables. Tracks reservation status (confirmed, pending) for efficient table use.

Fields: Reservation_ID, Customer_ID, Table_ID, Reservation_Time, No_of_Guests, Status

4.4 Tables Table

Description: Details table capacities and locations within the restaurant. Helps in managing seating and reservations efficiently.

Fields: Table_ID, Capacity, Location

4.5 Menu_Items Table

Description: Stores details of food and drinks offered on the menu. Helps in managing what is available for customers to order.

Fields: Item_ID, Name, Category, Price, Description

4.6 Orders Table

Description: Tracks customer orders, including which employee took the order. Used to manage the flow of orders during service hours.

Fields: Order_ID, Customer_ID, Employee_ID, Table_ID, Order_Time

4.7 Order_Details Table

Description: Contains detailed information about each order's contents. Helps to track the specific items ordered and the quantities.

Fields: Order_Detail_ID, Order_ID, Item_ID, Quantity

4.8 Payments Table

Description: Stores payment transaction details like method and amount. Tracks how orders are paid and manages financial records.

Fields: Payment_ID, Order_ID, Payment_Method, Total_Amount, Payment_Date

5. Relationships Between Tables

Here's a brief explanation of the relationships between the tables in the restaurant database:

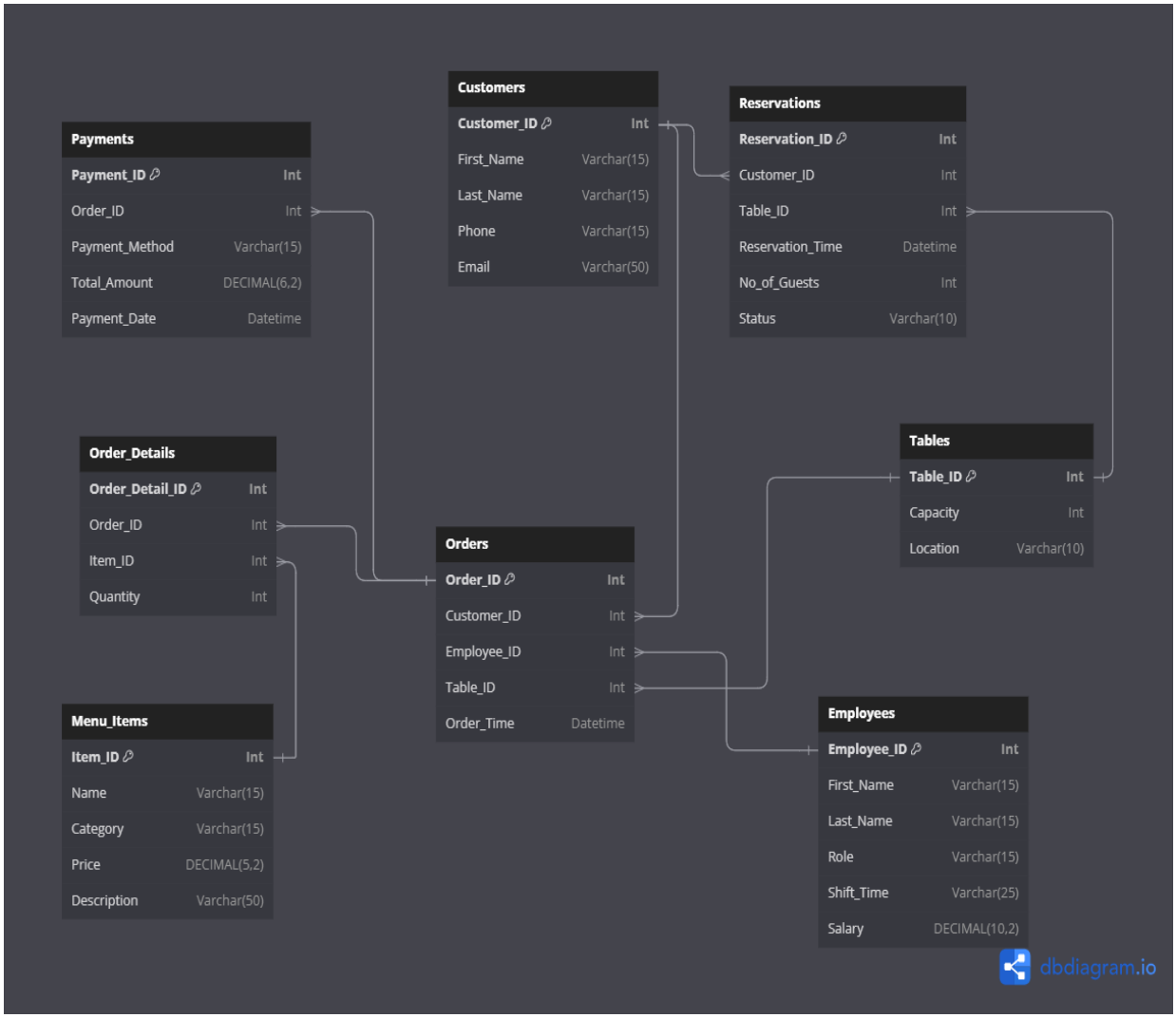
1. **Customers → Reservations (One-to-Many)**
   A customer can make multiple reservations, but each reservation belongs to one customer.
2. **Tables → Reservations (One-to-Many)**
   A table can be reserved many times, but each reservation is for one specific table.
3. **Employees → Orders (One-to-Many)**
   An employee can handle multiple orders, but each order is managed by one employee.
4. **Customers → Orders (One-to-Many)**
   A customer can place multiple orders, but each order is tied to a single customer.
5. **Tables → Orders (One-to-Many)**
   A table can have multiple orders over time, but each order is associated with one table.
6. **Orders → Order_Details (One-to-Many)**
   An order can contain multiple items (order details), but each detail belongs to one order.
7. **Menu_Items → Order_Details (One-to-Many)**
   A menu item can appear in multiple order details, but each order detail is for one menu item.
8. **Orders → Payments (One-to-One)**
   Each order has exactly one payment, and each payment is linked to one order.

6. Entity Relationship Diagram (ERD)

6.1 Overview of the ERD

The ERD provides a visual representation of the database structure, showing how different tables are related. Key tables like Customers, Orders, Employees, and Menu_Items are interconnected through various relationships.

6.2 ER-diagram



7. Conclusion

We created a database for Heritage Bites to help the restaurant run smoothly by managing reservations, orders, staff, menu items, and inventory. This design is based on the restaurant's goals and needs, making daily operations more efficient and improving customer satisfaction. Using a visual diagram (ERD), we made it easier for the restaurant to understand how everything is connected. The database is also flexible, so the restaurant can add features like loyalty programs and online reservations in the future.

8.Appendix

8.1 Table and description

Each table contains fields that store essential data required for smooth restaurant operations.

Customers Table

Field	Data Type	Description
Customer_ID	INT	Unique identifier for each customer (Primary Key)
First_Name	VARCHAR(50)	Customer's first name
Last_Name	VARCHAR(50)	Customer's last name
Phone	VARCHAR(15)	Customer's phone number
Email	VARCHAR(100)	Customer's email address
Loyalty_Points	INT	Points earned by customer for loyalty programs

Employees Table

Field	Data Type	Description
Employee_ID	INT	Unique identifier for each employee (Primary Key)
First_Name	VARCHAR(50)	Employee's first name
Last_Name	VARCHAR(50)	Employee's last name
Role	VARCHAR(50)	Employee's job role (e.g., Waiter, Chef)
Shift_Time	VARCHAR(20)	Employee's working shift (e.g., 9:00 AM - 5:00 PM)
Salary	DECIMAL(8,2)	Employee's salary

Reservations Table

Field	Data Type	Description
Reservation_ID	INT	Unique identifier for each reservation (Primary Key)
Customer_ID	INT	Foreign key linking to the customer who made the reservation
Table_ID	INT	Foreign key linking to the table being reserved
Reservation_Time	DATETIME	Time of the reservation
No_of_Guests	INT	Number of guests for the reservation
Status	VARCHAR(20)	Reservation status (e.g., Confirmed, Pending)

Tables Table

Field	Data Type	Description
Table_ID	INT	Unique identifier for each table (Primary Key)
Capacity	INT	Number of seats available at the table
Location	VARCHAR(20)	Table's location (e.g., Indoors, Outdoors)

Menu_Items Table

Field	Data Type	Description
Item_ID	INT	Unique identifier for each menu item (Primary Key)
Name	VARCHAR(100)	Name of the menu item
Category	VARCHAR(50)	Category of the menu item (e.g., Main Course, Dessert)
Price	DECIMAL(5,2)	Price of the menu item
Description	TEXT	Description of the menu item

Orders Table

Field	Data Type	Description
Order_ID	INT	Unique identifier for each order (Primary Key)
Customer_ID	INT	Foreign key linking to the customer who placed the order
Employee_ID	INT	Foreign key linking to the employee who took the order
Table_ID	INT	Foreign key linking to the table associated with the order
Order_Time	DATETIME	Time the order was placed

Order_Details Table

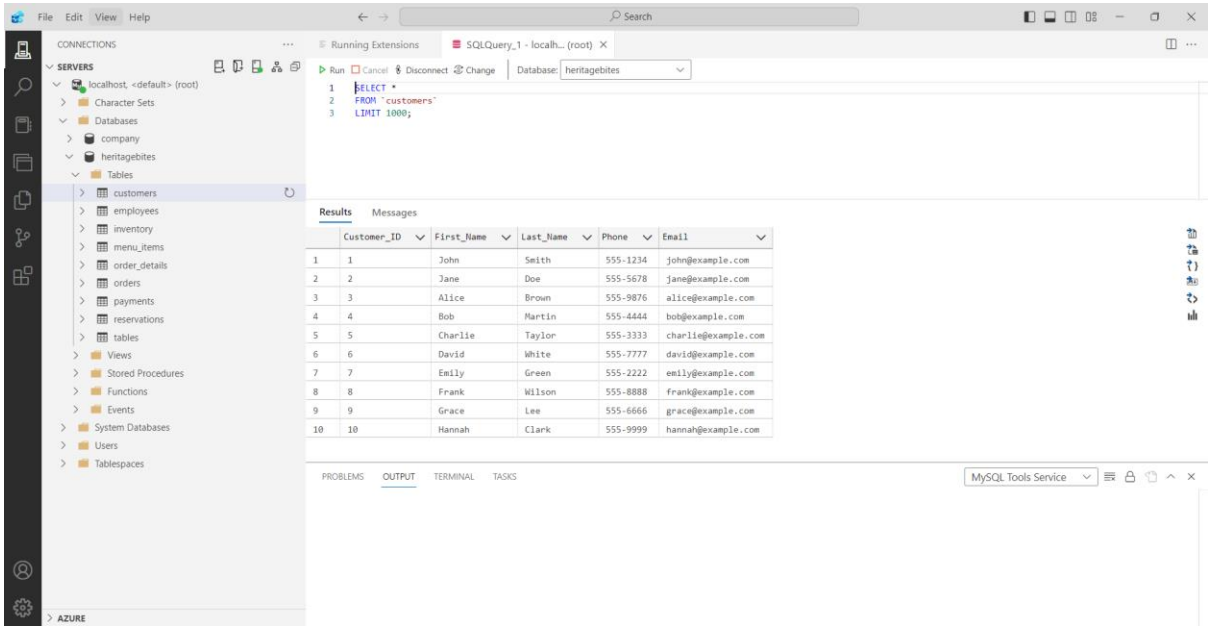
Field	Data Type	Description
Order_Detail_ID	INT	Unique identifier for each order detail (Primary Key)
Order_ID	INT	Foreign key linking to the corresponding order
Item_ID	INT	Foreign key linking to the menu item ordered
Quantity	INT	Quantity of the item ordered

Payments Table

Field	Data Type	Description
Payment_ID	INT	Unique identifier for each payment (Primary Key)
Order_ID	INT	Foreign key linking to the order for which payment is made
Payment_Method	VARCHAR(20)	Method of payment (e.g., Credit Card, Cash)
Total_Amount	DECIMAL(8,2)	Total amount paid for the order
Payment_Date	DATETIME	Date and time of payment

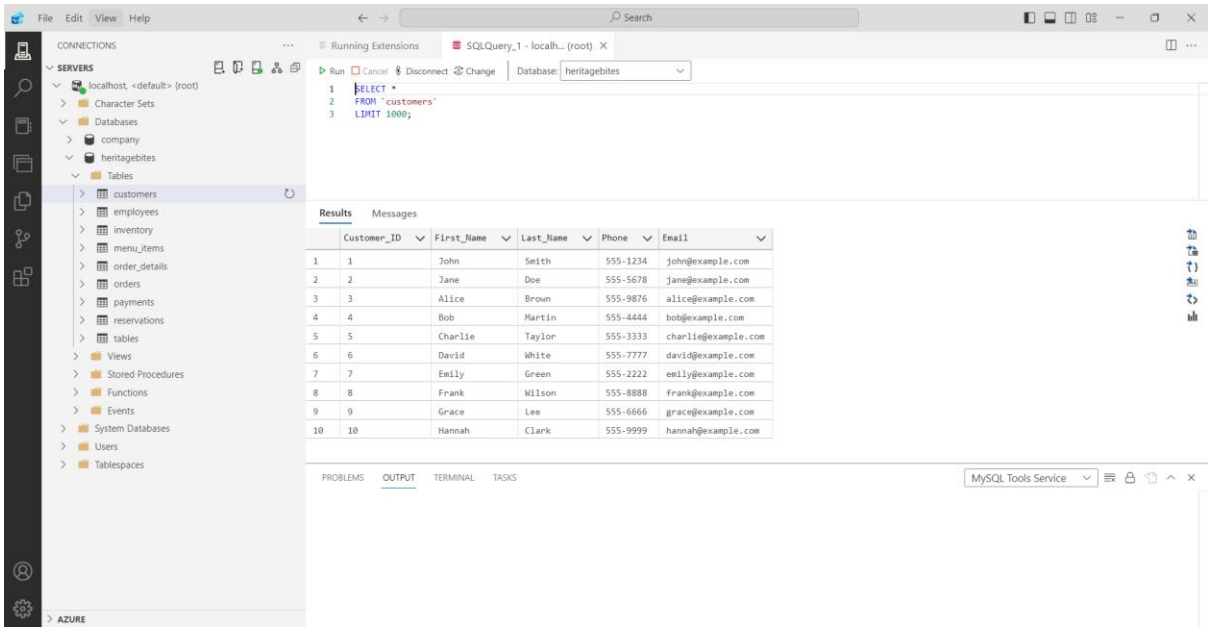
8.2 Testing database and Query

Create database of Heritagebites and create all tables.

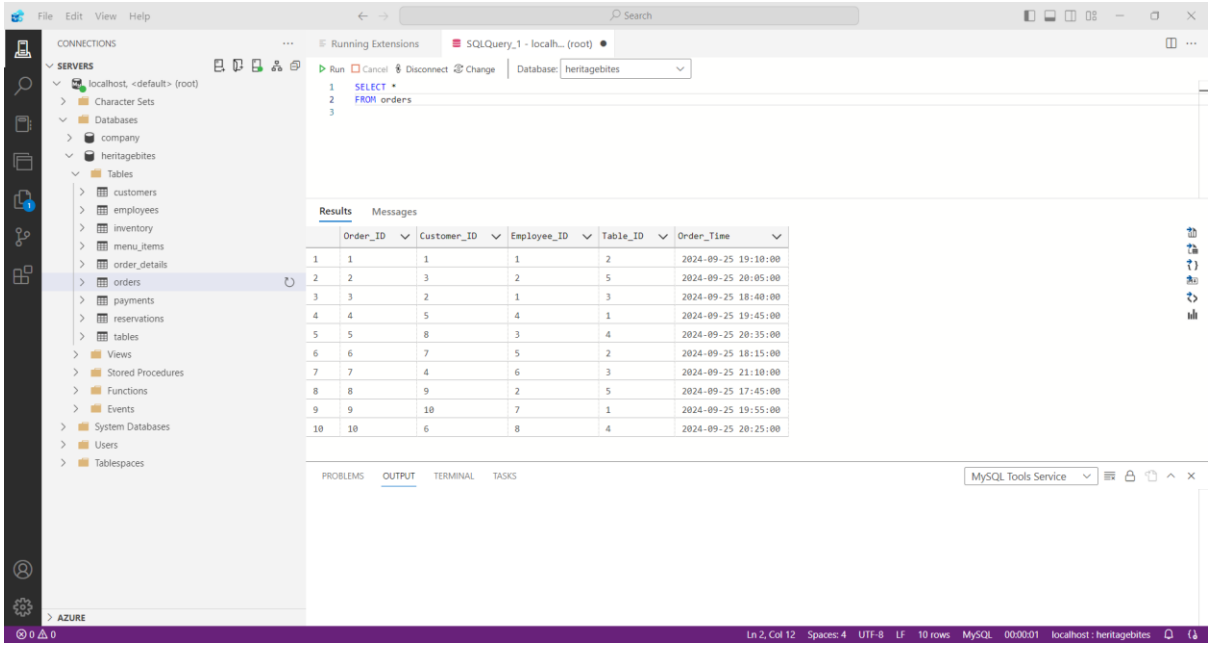


- **Select query**

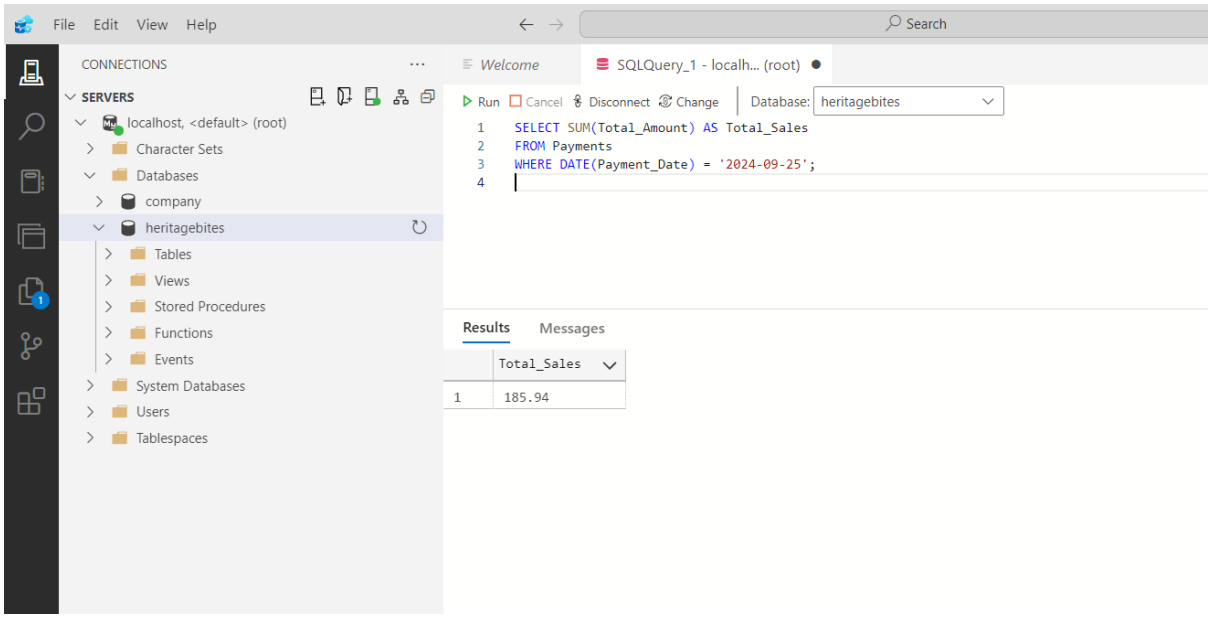
Select * FROM customers;



Select * FROM orders;

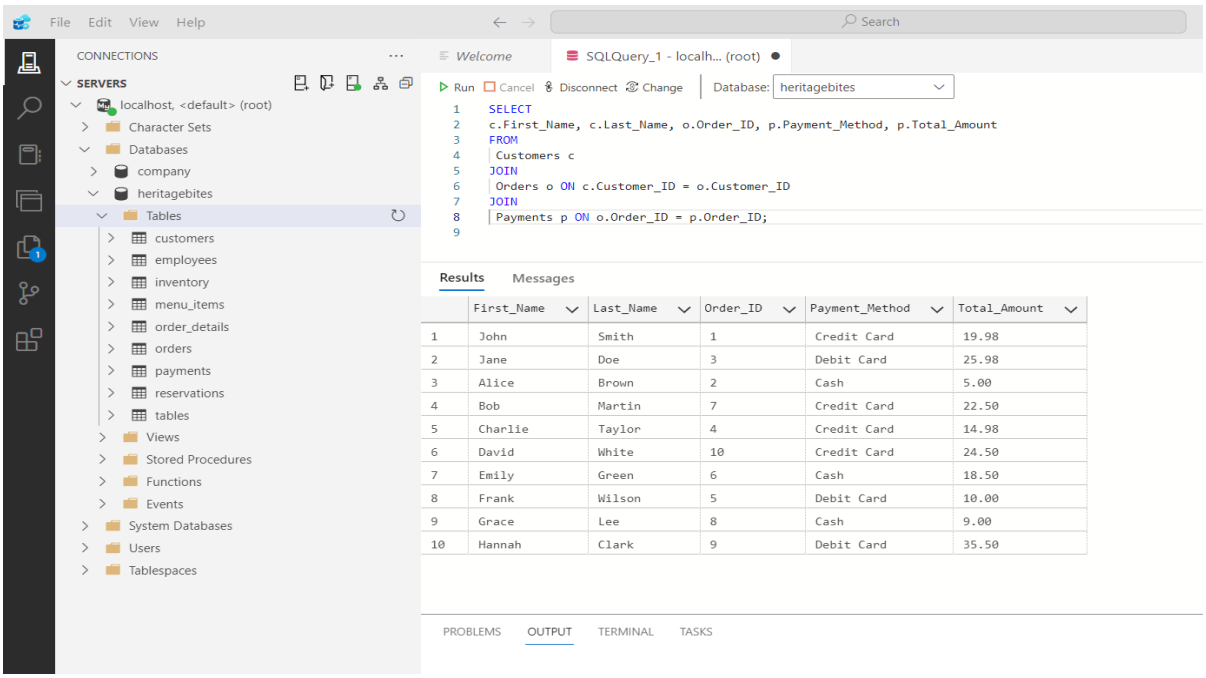


SELECT SUM(Total_Amount) AS Total_Sales FROM Payments WHERE DATE(Payment_Date) = '2024-09-25';



- **join query**

```
SELECT
    C.First_Name, C.Last_Name, O.Order_ID, P.Payment_Method, P.Total_Amount
FROM
    Customers C
JOIN
    Orders O ON C.Customer_ID = O.Customer_ID
JOIN
    Payments P ON O.Order_ID = P.Order_ID;
```



```
SELECT Customers.First_Name, Customers.Last_Name, Orders.Order_ID,
Orders.Order_Time
FROM Customers
INNER JOIN Orders ON Customers.Customer_ID = Orders.Customer_ID;
```