

Digna Hernandez

CIS 344

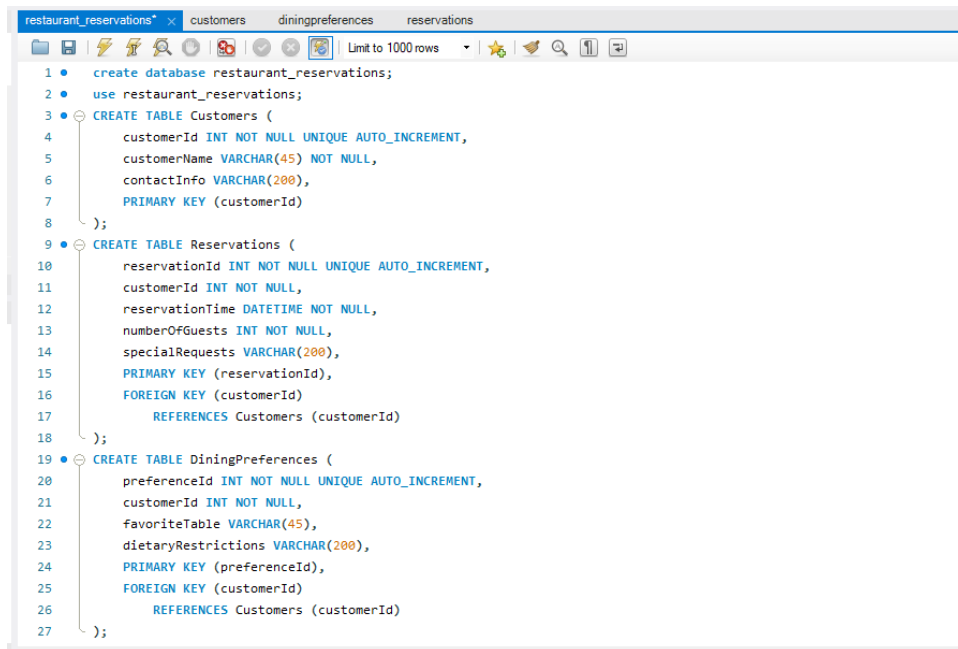
Prof. Yanilda Peralta

May 23, 2024

Final Project

In the project the specific task is to develop a database for reservations that would be useful for restaurants to manage their business. I will try to explain how I did this project step by step.

First, I had to create a database named 'restaurant_reservations' Then, I had to use the database to build appropriate tables in the database. The tables were called customers, reservations, and Dining Preferences.



```
1 • create database restaurant_reservations;
2 • use restaurant_reservations;
3 • CREATE TABLE Customers (
4     customerId INT NOT NULL UNIQUE AUTO_INCREMENT,
5     customerName VARCHAR(45) NOT NULL,
6     contactInfo VARCHAR(200),
7     PRIMARY KEY (customerId)
8 );
9 • CREATE TABLE Reservations (
10     reservationId INT NOT NULL UNIQUE AUTO_INCREMENT,
11     customerId INT NOT NULL,
12     reservationTime DATETIME NOT NULL,
13     numberOfGuests INT NOT NULL,
14     specialRequests VARCHAR(200),
15     PRIMARY KEY (reservationId),
16     FOREIGN KEY (customerId)
17         REFERENCES Customers (customerId)
18 );
19 • CREATE TABLE DiningPreferences (
20     preferenceId INT NOT NULL UNIQUE AUTO_INCREMENT,
21     customerId INT NOT NULL,
22     favoriteTable VARCHAR(45),
23     dietaryRestrictions VARCHAR(200),
24     PRIMARY KEY (preferenceId),
25     FOREIGN KEY (customerId)
26         REFERENCES Customers (customerId)
27 );
```

I then fill out these tables with data to analyze the outcome with a view to confirming the operation of the tool. This activity also entails the use of stored procedures to, for example, retrieve information regarding the reservations made by the customer, change the special requests made for the certain reservation, making new reservations if needed and any other subsequent activities as well as the creation of a customer where necessary. Finally, we need to run these stored procedures and observe whether they work as expected in a way to fulfill all the prescribed criteria in the given task.

```
restaurant_reservations* x customers diningpreferences reservations
Limit to 1000 rows
28 • INSERT INTO Customers (customerName, contactInfo) VALUES
29 ('Marla Hernandez', 'marla.h@example.com'),
30 ('John Smith', 'john.smith@example.com'),
31 ('Marcos Garcia', 'marcos.garcia@example.com'),
32 ('Alaia Lugo', 'alaia_lugo@example.com'),
33 ('David Mendez', 'david0123@example.com');
34
35 • INSERT INTO Reservations (customerId, reservationTime, numberOfGuests, specialRequests) VALUES
36 (1, '2024-06-01 19:00:00', 4, 'Window view'),
37 (2, '2024-06-02 20:00:00', 2, 'Quiet table'),
38 (3, '2024-06-03 18:00:00', 3, 'Birthday cake'),
39 (4, '2024-05-21 20:00:00', 4, 'No Apple'),
40 (5, '2024-05-23 19:45:00', 5, 'toddler chair');
41
42 • INSERT INTO DiningPreferences (customerId, favoriteTable, dietaryRestrictions) VALUES
43 (1, 'Table 5', 'None'),
44 (2, 'Table 3', 'Vegetarian'),
45 (3, 'Table 7', 'Gluten-Free'),
46 (4, 'Table 15', 'dairy-free'),
47 (5, 'Table 1', 'Vegan');
48
```

```
restaurant_reservations* x customers diningpreferences reservations
Limit to 1000 rows
55 DELIMITER //
56 • CREATE PROCEDURE AddSpecialRequest(IN resId INT, IN req VARCHAR(200))
57 BEGIN
58     UPDATE Reservations SET specialRequests = req WHERE reservationId = resId;
59 END //
60 DELIMITER ;
61 DELIMITER //
62 • CREATE PROCEDURE AddReservation(
63     IN custName VARCHAR(45),
64     IN contact VARCHAR(200),
65     IN resTime DATETIME,
66     IN numGuests INT,
67     IN specRequests VARCHAR(200)
68 )
```

The screenshot shows a database IDE with a SQL editor and a result grid. The SQL editor contains a PL/SQL block that declares a variable, finds a customer ID, inserts a new customer if none is found, and then inserts a reservation. It also includes three procedure calls: FindReservations, AddSpecialRequest, and Addreservation. The result grid below shows two rows of reservation data.

```
68 )
69 BEGIN
70     DECLARE custId INT;
71
72     SELECT customerId INTO custId FROM Customers WHERE customerName = custName AND contactInfo = contact;
73
74     IF custId IS NULL THEN
75         INSERT INTO Customers (customerName, contactInfo) VALUES (custName, contact);
76         SET custId = last_insert_id();
77     END IF;
78
79     INSERT INTO Reservations (customerId, reservationTime, numberOfGuests, specialRequests)
80     VALUES (custId, resTime, numGuests, specRequests);
81 END //
82 DELIMITER ;
83
84 • CALL FindReservations(1);
85
86 • CALL AddSpecialRequest(2, 'quiet table');
87
88 • CALL Addreservation('Marla Hernandez', 'marla.h@example.com', '2024-06-02 20:00:00', 2, 'quiet table');
89
```

Result Grid

reservationId	customerId	reservationTime	numberOfGuests	specialRequests
1	1	2024-06-01 19:00:00	4	Window view
6	1	2024-06-02 20:00:00	2	quiet table

This solution offers an organized method of handling restaurant bookings, which includes receiving customers, processing reservations, and customizing the experience with unique requests and eating preferences.

We had to connect MySQL to python, but I had too many problems with it. It was not connecting it was saying that I had an error, and I could not connect.