AIRLINE RESERVATION SYSTEM

This **Airline Reservation System** is a database-driven project that manages the core functionalities of airline operations, such as managing flights, bookings, passengers, and airports. It allows users to retrieve available flights, calculate passenger load factors, and generate revenue reports efficiently.

Features:

- Manage Airports: Store airport details like name, city, and country.
- Flight Management: Store and manage flight schedules and capacities.
- Passenger Bookings: Track passenger details and seat assignments.
- Revenue Reports: Generate reports based on flight bookings.
- Passenger Load Factor: Calculate flight efficiency by tracking booked seats.

Database Schema:

This project uses the following tables:

1. Airports: Stores airport details (name, code, city, etc.).
2. Flights: Stores flight schedules and seat capacities.
3. Passengers: Stores passenger information (name, contact, etc.).
4. Bookings: Manages bookings for specific flights with fare details.

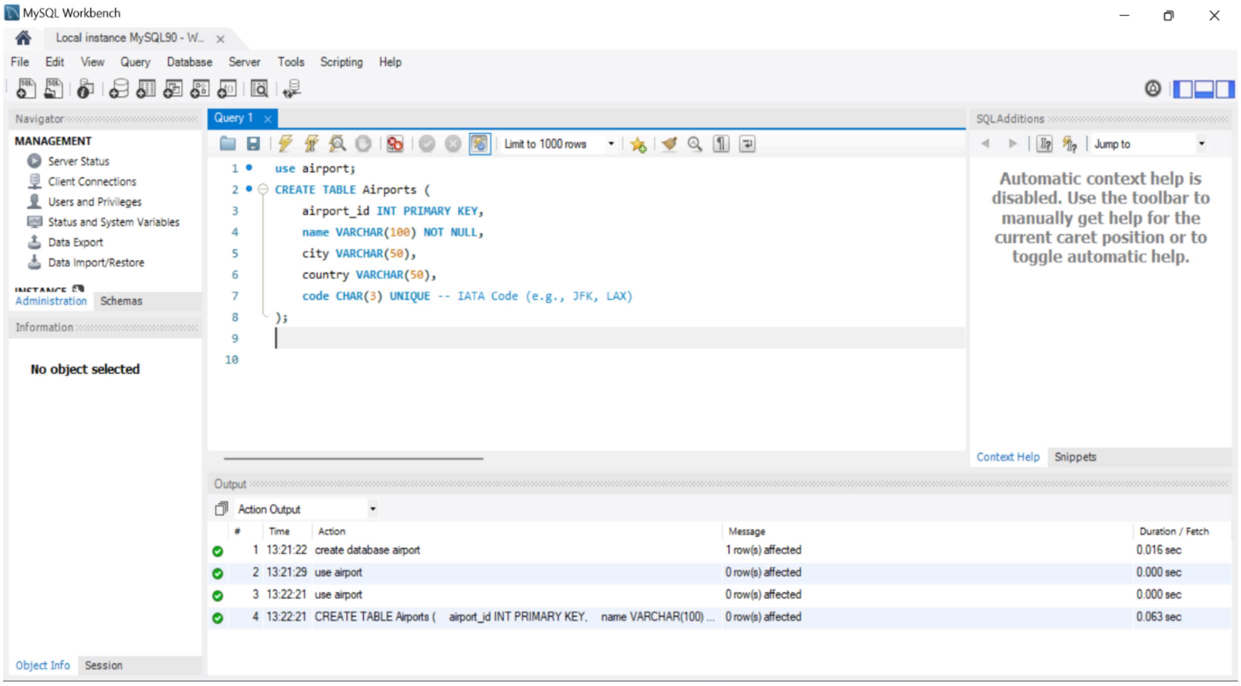
1. Tables Overview:

We'll create three main tables:

- **Flights**: To store information about flights, including departure and arrival details.
- **Airports**: To store information about different airports.
- **Bookings**: To manage the relationship between passengers and flights, including booking status and seat allocation.
- **Passengers**: To store personal details of passengers.

Database Schema:

1. Airports Table



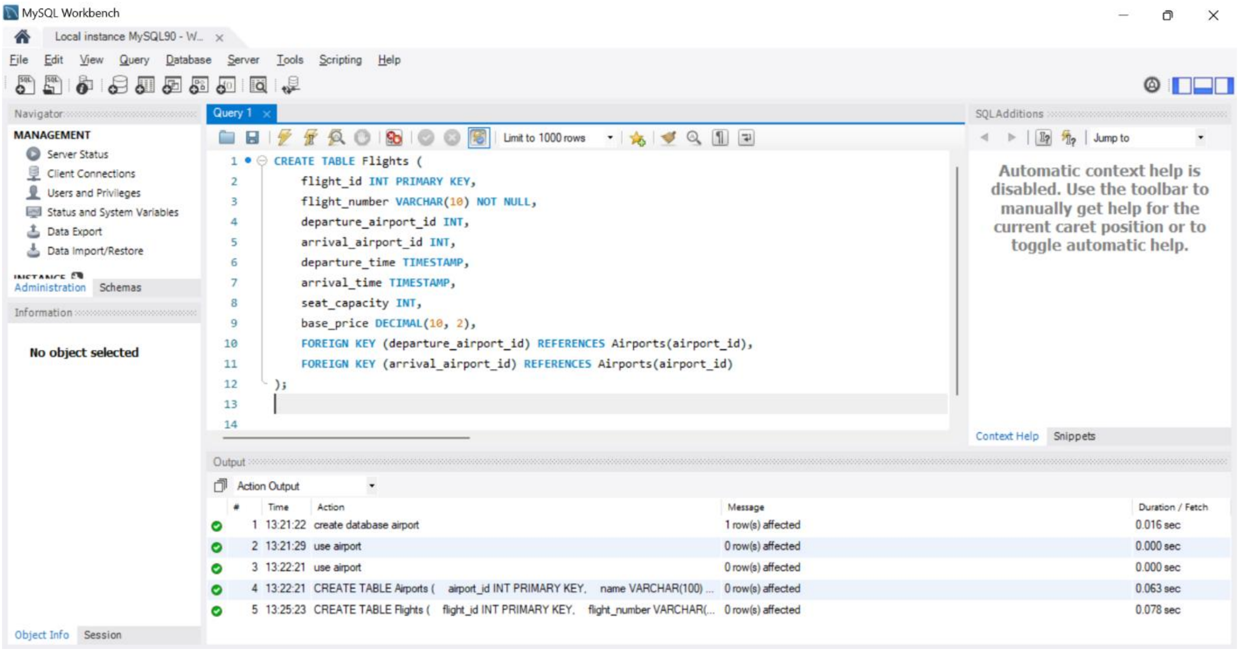
```sql
use airport;
CREATE TABLE Airports (
    airport_id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    city VARCHAR(50),
    country VARCHAR(50),
    code CHAR(3) UNIQUE -- IATA Code (e.g., JFK, LAX)
);
```

No object selected

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

#	Time	Action	Message	Duration / Fetch
1	13:21:22	create database airport	1 row(s) affected	0.016 sec
2	13:21:29	use airport	0 row(s) affected	0.000 sec
3	13:22:21	use airport	0 row(s) affected	0.000 sec
4	13:22:21	CREATE TABLE Airports (airport_id INT PRIMARY KEY, name VARCHAR(100) ...	0 row(s) affected	0.063 sec

2. Flights Table



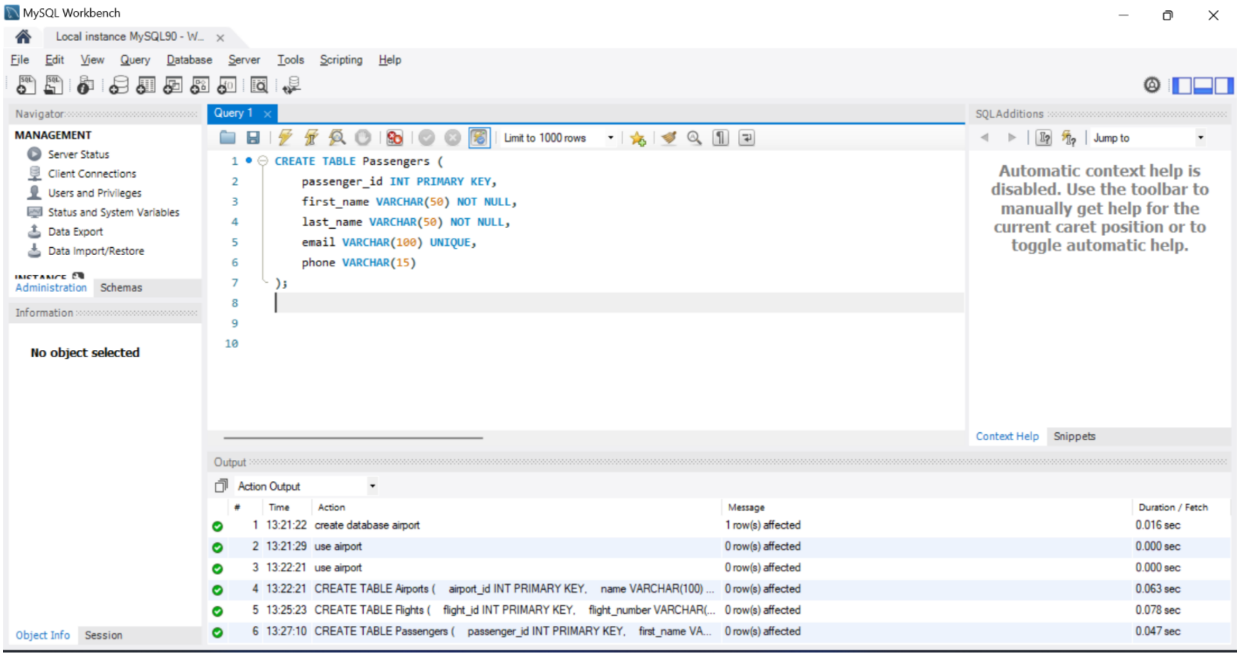
```sql
CREATE TABLE Flights (
    flight_id INT PRIMARY KEY,
    flight_number VARCHAR(10) NOT NULL,
    departure_airport_id INT,
    arrival_airport_id INT,
    departure_time TIMESTAMP,
    arrival_time TIMESTAMP,
    seat_capacity INT,
    base_price DECIMAL(10, 2),
    FOREIGN KEY (departure_airport_id) REFERENCES Airports(airport_id),
    FOREIGN KEY (arrival_airport_id) REFERENCES Airports(airport_id)
);
```

No object selected

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

#	Time	Action	Message	Duration / Fetch
1	13:21:22	create database airport	1 row(s) affected	0.016 sec
2	13:21:29	use airport	0 row(s) affected	0.000 sec
3	13:22:21	use airport	0 row(s) affected	0.000 sec
4	13:22:21	CREATE TABLE Airports (airport_id INT PRIMARY KEY, name VARCHAR(100) ...	0 row(s) affected	0.063 sec
5	13:25:23	CREATE TABLE Flights (flight_id INT PRIMARY KEY, flight_number VARCHAR(...	0 row(s) affected	0.078 sec

3. Passengers Table

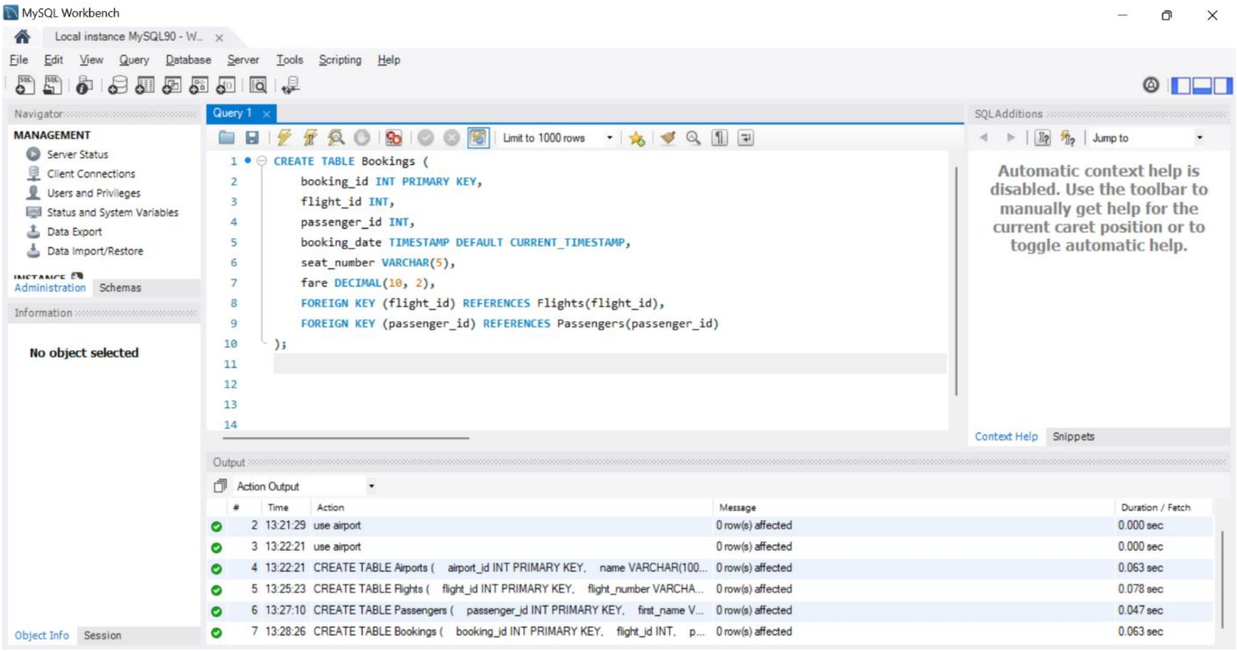


```sql
CREATE TABLE Passengers (
    passenger_id INT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(15)
);
```

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

```
#  Time      Action                                                                          Message             Duration / Fetch
1  13:21:22  create database airport                                                         1 row(s) affected   0.016 sec
2  13:21:29  use airport                                                                     0 row(s) affected   0.000 sec
3  13:22:21  use airport                                                                     0 row(s) affected   0.000 sec
4  13:22:21  CREATE TABLE Airports (   airport_id INT PRIMARY KEY,   name VARCHAR(100) ...   0 row(s) affected   0.063 sec
5  13:25:23  CREATE TABLE Flights (   flight_id INT PRIMARY KEY,   flight_number VARCHAR(... 0 row(s) affected   0.078 sec
6  13:27:10  CREATE TABLE Passengers (   passenger_id INT PRIMARY KEY,   first_name VA...    0 row(s) affected   0.047 sec
```

4. Bookings Table



```sql
CREATE TABLE Bookings (
    booking_id INT PRIMARY KEY,
    flight_id INT,
    passenger_id INT,
    booking_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    seat_number VARCHAR(5),
    fare DECIMAL(10, 2),
    FOREIGN KEY (flight_id) REFERENCES Flights(flight_id),
    FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id)
);
```

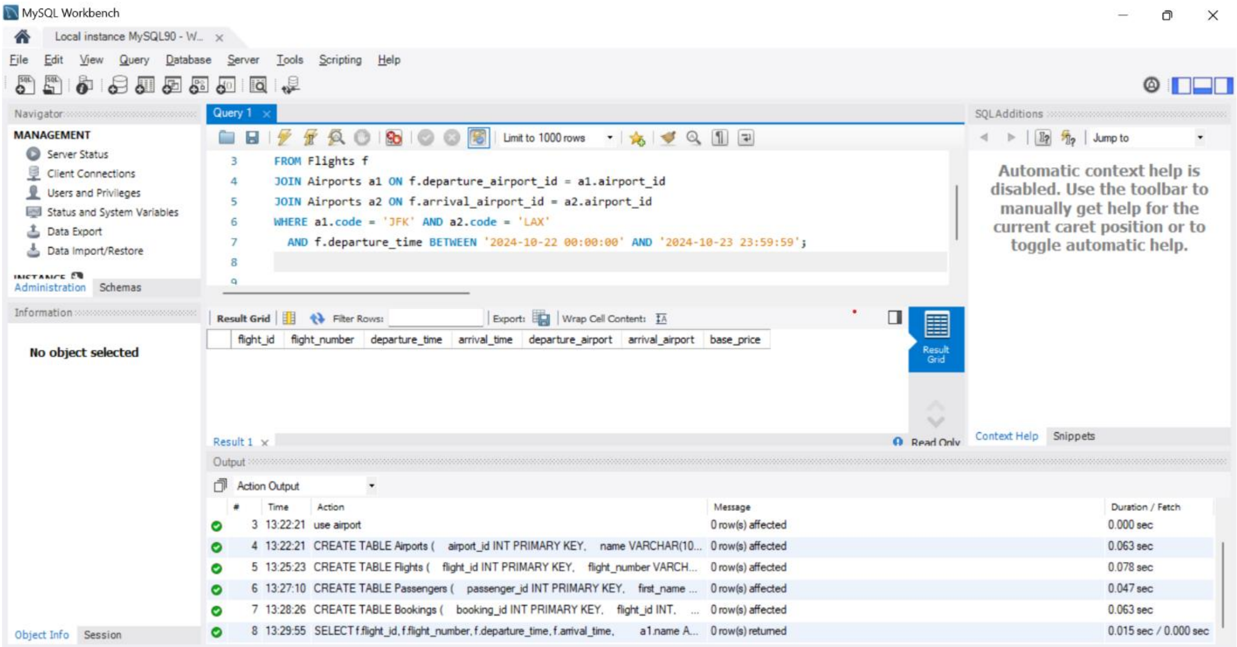
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
#  Time      Action                                                                          Message             Duration / Fetch
2  13:21:29  use airport                                                                     0 row(s) affected   0.000 sec
3  13:22:21  use airport                                                                     0 row(s) affected   0.000 sec
4  13:22:21  CREATE TABLE Airports (   airport_id INT PRIMARY KEY,   name VARCHAR(100...     0 row(s) affected   0.063 sec
5  13:25:23  CREATE TABLE Flights (   flight_id INT PRIMARY KEY,   flight_number VARCHA...   0 row(s) affected   0.078 sec
6  13:27:10  CREATE TABLE Passengers (   passenger_id INT PRIMARY KEY,   first_name V...     0 row(s) affected   0.047 sec
7  13:28:26  CREATE TABLE Bookings (   booking_id INT PRIMARY KEY,   flight_id INT,   p...   0 row(s) affected   0.063 sec
```

SQL Queries:

1. Find Available Flights

Retrieve all flights that depart from a specific airport and arrive at another airport within a given date range.



```sql
FROM Flights f
JOIN Airports a1 ON f.departure_airport_id = a1.airport_id
JOIN Airports a2 ON f.arrival_airport_id = a2.airport_id
WHERE a1.code = 'JFK' AND a2.code = 'LAX'
  AND f.departure_time BETWEEN '2024-10-22 00:00:00' AND '2024-10-23 23:59:59';
```

flight_id	flight_number	departure_time	arrival_time	departure_airport	arrival_airport	base_price

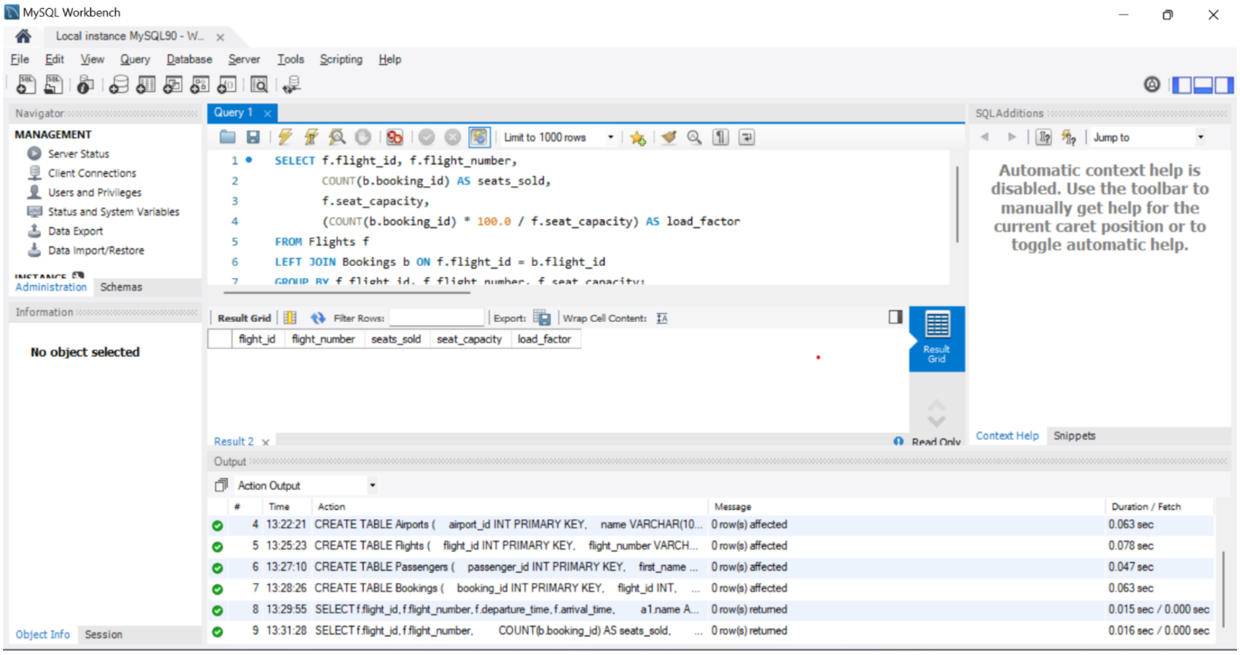
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#  Time      Action                                                                       Message              Duration / Fetch
3  13:22:21  use airport                                                                  0 row(s) affected    0.000 sec
4  13:22:21  CREATE TABLE Airports (   airport_id INT PRIMARY KEY,   name VARCHAR(10...   0 row(s) affected    0.063 sec
5  13:25:23  CREATE TABLE Flights (   flight_id INT PRIMARY KEY,   flight_number VARCH... 0 row(s) affected    0.078 sec
6  13:27:10  CREATE TABLE Passengers (   passenger_id INT PRIMARY KEY,   first_name ...   0 row(s) affected    0.047 sec
7  13:28:26  CREATE TABLE Bookings (   booking_id INT PRIMARY KEY,   flight_id INT,   ... 0 row(s) affected    0.063 sec
8  13:29:55  SELECT f.flight_id, f.flight_number, f.departure_time, f.arrival_time,   a1.name A...  0 row(s) returned  0.015 sec / 0.000 sec
```

2. Calculate Passenger Load Factor:

The Passenger Load Factor (PLF) measures how efficiently seats are filled on a flight. It is calculated as:
PLF = (Seats Sold / Total Seats) × 100



```sql
SELECT f.flight_id, f.flight_number,
       COUNT(b.booking_id) AS seats_sold,
       f.seat_capacity,
       (COUNT(b.booking_id) * 100.0 / f.seat_capacity) AS load_factor
FROM Flights f
LEFT JOIN Bookings b ON f.flight_id = b.flight_id
GROUP BY f.flight_id, f.flight_number, f.seat_capacity;
```

flight_id	flight_number	seats_sold	seat_capacity	load_factor

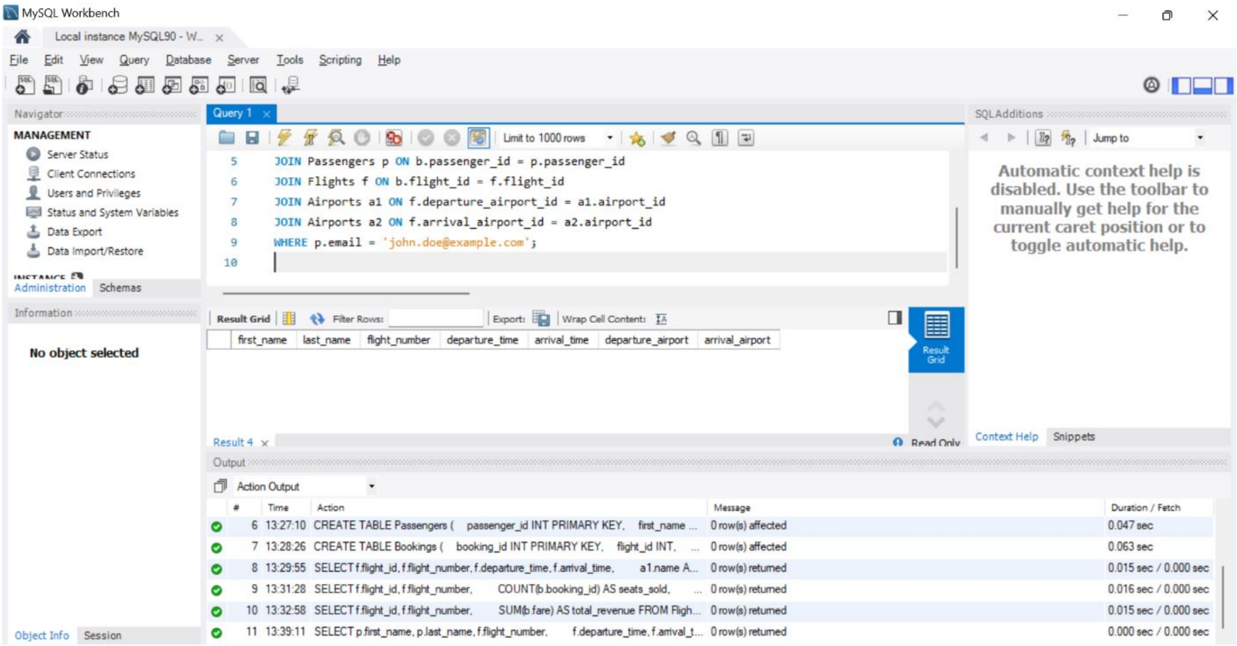
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

No object selected

```
#  Time      Action                                                                       Message              Duration / Fetch
4  13:22:21  CREATE TABLE Airports (   airport_id INT PRIMARY KEY,   name VARCHAR(10...   0 row(s) affected    0.063 sec
5  13:25:23  CREATE TABLE Flights (   flight_id INT PRIMARY KEY,   flight_number VARCH... 0 row(s) affected    0.078 sec
6  13:27:10  CREATE TABLE Passengers (   passenger_id INT PRIMARY KEY,   first_name ...   0 row(s) affected    0.047 sec
7  13:28:26  CREATE TABLE Bookings (   booking_id INT PRIMARY KEY,   flight_id INT,   ... 0 row(s) affected    0.063 sec
8  13:29:55  SELECT f.flight_id, f.flight_number, f.departure_time, f.arrival_time,   a1.name A...  0 row(s) returned  0.015 sec / 0.000 sec
9  13:31:28  SELECT f.flight_id, f.flight_number,   COUNT(b.booking_id) AS seats_sold,   ...  0 row(s) returned  0.016 sec / 0.000 sec
```

3. Find All Flights for a Passenger

Retrieve all the flights a specific passenger has booked.



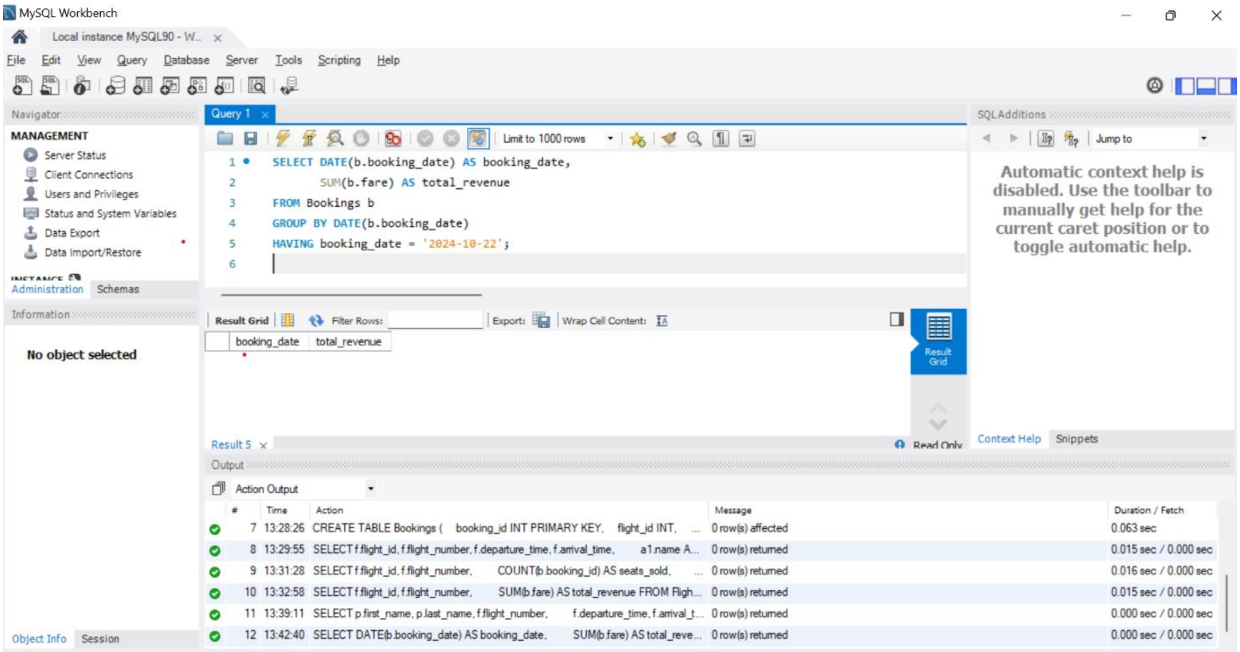
```sql
JOIN Passengers p ON b.passenger_id = p.passenger_id
JOIN Flights f ON b.flight_id = f.flight_id
JOIN Airports a1 ON f.departure_airport_id = a1.airport_id
JOIN Airports a2 ON f.arrival_airport_id = a2.airport_id
WHERE p.email = 'john.doe@example.com';
```

first_name	last_name	flight_number	departure_time	arrival_time	departure_airport	arrival_airport

```
#   Time      Action                                                                  Message            Duration / Fetch
6   13:27:10  CREATE TABLE Passengers (    passenger_id INT PRIMARY KEY,    first_name ...  0 row(s) affected  0.047 sec
7   13:28:26  CREATE TABLE Bookings (    booking_id INT PRIMARY KEY,    flight_id INT,    ...  0 row(s) affected  0.063 sec
8   13:29:55  SELECT f.flight_id, f.flight_number, f.departure_time, f.arrival_time,    a1.name A...  0 row(s) returned  0.015 sec / 0.000 sec
9   13:31:28  SELECT f.flight_id, f.flight_number,    COUNT(b.booking_id) AS seats_sold,    ...  0 row(s) returned  0.016 sec / 0.000 sec
10  13:32:58  SELECT f.flight_id, f.flight_number,    SUM(b.fare) AS total_revenue FROM Fligh...  0 row(s) returned  0.015 sec / 0.000 sec
11  13:39:11  SELECT p.first_name, p.last_name, f.flight_number,    f.departure_time, f.arrival_t...  0 row(s) returned  0.000 sec / 0.000 sec
```

4. Daily Revenue Report:

Generate a report of total revenue for all flights on a specific date.



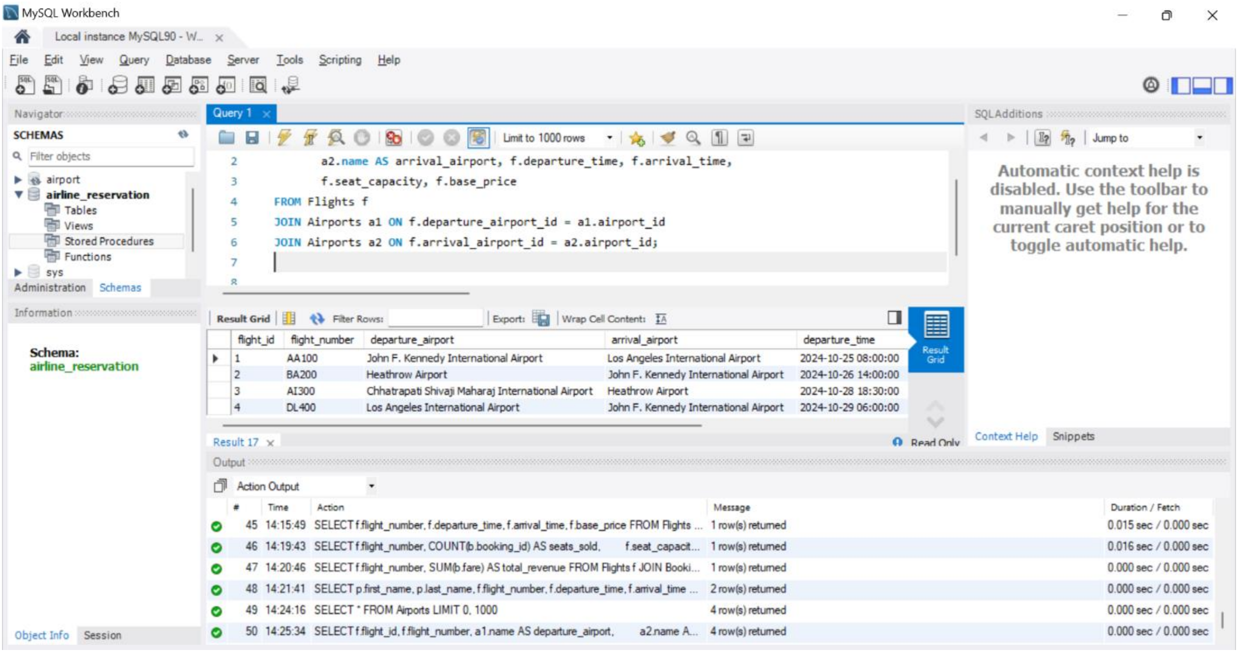
```sql
SELECT DATE(b.booking_date) AS booking_date,
       SUM(b.fare) AS total_revenue
FROM Bookings b
GROUP BY DATE(b.booking_date)
HAVING booking_date = '2024-10-22';
```

booking_date	total_revenue

```
#   Time      Action                                                                  Message            Duration / Fetch
7   13:28:26  CREATE TABLE Bookings (    booking_id INT PRIMARY KEY,    flight_id INT,    ...  0 row(s) affected  0.063 sec
8   13:29:55  SELECT f.flight_id, f.flight_number, f.departure_time, f.arrival_time,    a1.name A...  0 row(s) returned  0.015 sec / 0.000 sec
9   13:31:28  SELECT f.flight_id, f.flight_number,    COUNT(b.booking_id) AS seats_sold,    ...  0 row(s) returned  0.016 sec / 0.000 sec
10  13:32:58  SELECT f.flight_id, f.flight_number,    SUM(b.fare) AS total_revenue FROM Fligh...  0 row(s) returned  0.015 sec / 0.000 sec
11  13:39:11  SELECT p.first_name, p.last_name, f.flight_number,    f.departure_time, f.arrival_t...  0 row(s) returned  0.000 sec / 0.000 sec
12  13:42:40  SELECT DATE(b.booking_date) AS booking_date,    SUM(b.fare) AS total_reve...  0 row(s) returned  0.000 sec / 0.000 sec
```

Retrieve All Flights

This query shows all the flights along with their departure and arrival times, airports, and seat capacities.



```
        a2.name AS arrival_airport, f.departure_time, f.arrival_time,
        f.seat_capacity, f.base_price
FROM Flights f
JOIN Airports a1 ON f.departure_airport_id = a1.airport_id
JOIN Airports a2 ON f.arrival_airport_id = a2.airport_id;
```

flight_id	flight_number	departure_airport	arrival_airport	departure_time
1	AA100	John F. Kennedy International Airport	Los Angeles International Airport	2024-10-25 08:00:00
2	BA200	Heathrow Airport	John F. Kennedy International Airport	2024-10-26 14:00:00
3	AI300	Chhatrapati Shivaji Maharaj International Airport	Heathrow Airport	2024-10-28 18:30:00
4	DL400	Los Angeles International Airport	John F. Kennedy International Airport	2024-10-29 06:00:00

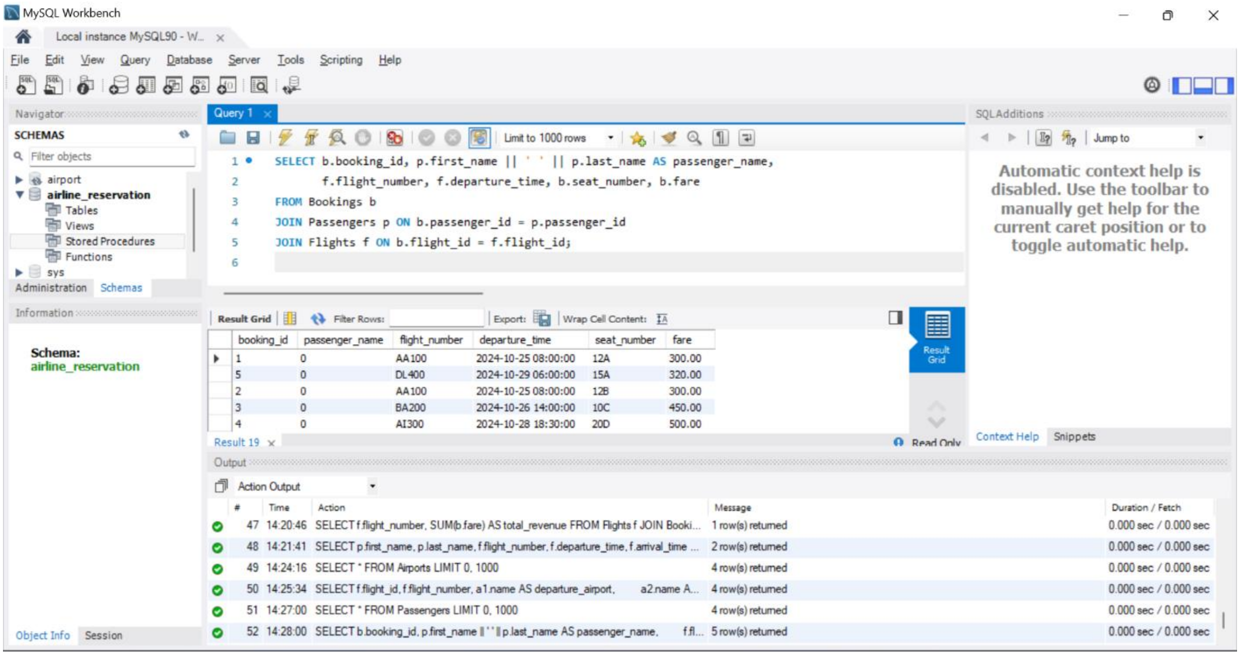
Schema:
airline_reservation

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

```
 #   Time      Action                                                                      Message            Duration / Fetch
 45  14:15:49  SELECT f.flight_number, f.departure_time, f.arrival_time, f.base_price FROM Flights ...  1 row(s) returned  0.015 sec / 0.000 sec
 46  14:19:43  SELECT f.flight_number, COUNT(b.booking_id) AS seats_sold,     f.seat_capacit...  1 row(s) returned  0.016 sec / 0.000 sec
 47  14:20:46  SELECT f.flight_number, SUM(b.fare) AS total_revenue FROM Flights f JOIN Booki...  1 row(s) returned  0.000 sec / 0.000 sec
 48  14:21:41  SELECT p.first_name, p.last_name, f.flight_number, f.departure_time, f.arrival_time ...  2 row(s) returned  0.000 sec / 0.000 sec
 49  14:24:16  SELECT * FROM Airports LIMIT 0, 1000                                        4 row(s) returned  0.000 sec / 0.000 sec
 50  14:25:34  SELECT f.flight_id, f.flight_number, a1.name AS departure_airport,     a2.name A...  4 row(s) returned  0.000 sec / 0.000 sec
```

Retrieve All Bookings

This query lists all the bookings with flight details and passengers associated with them.



```
SELECT b.booking_id, p.first_name || ' ' || p.last_name AS passenger_name,
        f.flight_number, f.departure_time, b.seat_number, b.fare
FROM Bookings b
JOIN Passengers p ON b.passenger_id = p.passenger_id
JOIN Flights f ON b.flight_id = f.flight_id;
```

booking_id	passenger_name	flight_number	departure_time	seat_number	fare
1	0	AA100	2024-10-25 08:00:00	12A	300.00
5	0	DL400	2024-10-29 06:00:00	15A	320.00
2	0	AA100	2024-10-25 08:00:00	12B	300.00
3	0	BA200	2024-10-26 14:00:00	10C	450.00
4	0	AI300	2024-10-28 18:30:00	20D	500.00

Schema:
airline_reservation

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

```
 #   Time      Action                                                                      Message            Duration / Fetch
 47  14:20:46  SELECT f.flight_number, SUM(b.fare) AS total_revenue FROM Flights f JOIN Booki...  1 row(s) returned  0.000 sec / 0.000 sec
 48  14:21:41  SELECT p.first_name, p.last_name, f.flight_number, f.departure_time, f.arrival_time ...  2 row(s) returned  0.000 sec / 0.000 sec
 49  14:24:16  SELECT * FROM Airports LIMIT 0, 1000                                        4 row(s) returned  0.000 sec / 0.000 sec
 50  14:25:34  SELECT f.flight_id, f.flight_number, a1.name AS departure_airport,     a2.name A...  4 row(s) returned  0.000 sec / 0.000 sec
 51  14:27:00  SELECT * FROM Passengers LIMIT 0, 1000                                      4 row(s) returned  0.000 sec / 0.000 sec
 52  14:28:00  SELECT b.booking_id, p.first_name || ' ' || p.last_name AS passenger_name,     f.fl...  5 row(s) returned  0.000 sec / 0.000 sec
```

Conclusion :

The Airline Reservation System plays a crucial role in streamlining the process of booking and managing airline tickets for passengers. By automating various functions such as flight scheduling, passenger information management, and booking tracking, the system enhances efficiency, reduces errors, and provides a seamless experience for both airline staff and customers. Additionally, the integration of database management ensures that flight, airport, passenger, and booking information is securely stored and easily accessible. This system not only optimizes resource management but also improves customer satisfaction by enabling smooth operations, making it an essential tool for modern airline management.