Use Cases + Queries

This document serves as a comprehensive overview of the use cases in our airline reservation system and the corresponding SQL queries that support them.

Table of Contents

Public Access Features

Flight Search

Description: All users can search for future flights based on source city/airport name, destination city/airport name, and dates.

Query:

```
SELECT flight_number, airline_name, departure_datetime, arrival_datetime,
       departure_airport_code, arrival_airport_code, base_price, status,
airplane_ID
FROM flight
WHERE departure_airport_code = [source]
AND arrival_airport_code = [destination]
AND DATE(departure_datetime) = [departure_date]
```

Authentication

Customer Registration

Description: New customers can register by providing their personal information.

Query:

```
INSERT INTO customer
VALUES(email, name, md5(password), building_number, street, city,
       state, phone_number, passport_number, passport_expiration,
       passport_country, date_of_birth)
```

Airline Staff Registration

Description: New airline staff members can register with their airline credentials.

Queries:

```
-- Insert staff member
INSERT INTO airline_staff
VALUES(username, md5(password), first_name, last_name, date_of_birth,
airline_name)

-- Insert staff phone numbers
INSERT INTO staff_phone VALUES(username, phone_number)

-- Insert staff email addresses
INSERT INTO airline_email VALUES(username, email)
```

Login Authentication

Description: Users can log in using their credentials.

Queries:

```
-- Customer Login
SELECT * FROM customer
WHERE email = [email] AND password = md5([password])

-- Staff Login
SELECT * FROM airline_staff
WHERE username = [username] AND password = md5([password])
```

Customer Features

View My Flights

Description: Customers can view all flights they have purchased tickets for.

Query:

```
SELECT f.*, t.ticket_ID
FROM purchase p
JOIN ticket t ON p.ticket_ID = t.ticket_ID
JOIN flight f ON t.airline_name = f.airline_name
    AND t.flight_number = f.flight_number
    AND t.departure_datetime = f.departure_datetime
WHERE p.email = [customer_email]
ORDER BY f.departure_datetime DESC
```

Search for Flights

Description: Customers can search for future flights, including both one-way and round-trip options.

Queries:

```
-- Outbound flight search
SELECT flight_number, airline_name, departure_datetime, arrival_datetime,
       departure_airport_code, arrival_airport_code, base_price, status,
airplane_ID
FROM flight
WHERE departure_airport_code = [source]
AND arrival_airport_code = [destination]
AND DATE(departure_datetime) = [departure_date]

-- Return flight search (for round-trip)
SELECT flight_number, airline_name, departure_datetime, arrival_datetime,
       departure_airport_code, arrival_airport_code, base_price, status,
airplane_ID
FROM flight
WHERE departure_airport_code = [destination]
AND arrival_airport_code = [source]
AND DATE(arrival_datetime) = [return_date]
AND DATE(departure_datetime) >= DATE([departure_date])
ORDER BY departure_datetime ASC
```

Purchase Tickets

Description: Customers can purchase tickets for flights.

Queries:

```
-- Create ticket
INSERT INTO ticket (ticket_ID, airline_name, flight_number, departure_datetime)
VALUES ([ticket_id], [airline_name], [flight_number], [departure_datetime])

-- Record purchase
INSERT INTO purchase (ticket_ID, email, card_type, card_number, card_name,
                     card_expiry, purchase_datetime, sold_price)
VALUES ([ticket_id], [customer_email], [card_type], [card_number], [card_name],
        [expiry_date], datetime.datetime.now(), [ticket_price])
```

Cancel Trip

Description: Customers can cancel tickets for flights departing more than 24 hours in the future.

Query:

```
DELETE FROM purchase
WHERE ticket_ID = [ticket_id] AND email = [customer_email]
```

Rate and Comment on Flights

Description: Customers can rate and comment on their past flights.

Query:

```
INSERT INTO review (email, airline_name, flight_number, departure_datetime,
review, comment)
VALUES ([customer_email], [airline_name], [flight_number], [departure_datetime],
[rating], [comment])
```

Logout

Description: End the customer's session.

Implementation Note: This is handled at the application level by destroying the session, no direct database queries are involved.

Airline Staff Features

View Flights

Description: Staff can view all flights for their airline within a date range. Default shows flights for next 30 days.

Query:

```
SELECT flight_number, departure_datetime, arrival_datetime,
       departure_airport_code, arrival_airport_code, status,
       airplane_ID, base_price
FROM flight
WHERE airline_name = [staff_airline]
AND DATE(departure_datetime) BETWEEN [start_date] AND [end_date]
AND (departure_airport_code = [source_airport] OR [source_airport] IS NULL)
AND (arrival_airport_code = [destination_airport] OR [destination_airport] IS
NULL)
ORDER BY departure_datetime ASC
```

View Flight Customers

Description: Staff can view all customers booked on a specific flight.

Query:

```
SELECT c.name, c.email, t.ticket_ID, p.sold_price
FROM ticket t
JOIN purchase p ON t.ticket_ID = p.ticket_ID
JOIN customer c ON p.email = c.email
WHERE t.flight_number = [flight_number] AND t.airline_name = [staff_airline]
```

Create New Flight

Description: Staff can create new flights for their airline.

Query:

```
INSERT INTO flight (airline_name, flight_number, departure_datetime,
                   arrival_datetime, departure_airport_code,
                   arrival_airport_code, base_price, status, airplane_ID)
VALUES ([airline_name], [flight_number], [departure_datetime],
        [arrival_datetime], [departure_airport], [arrival_airport],
        [base_price], [status], [airplane_id])
```

Update Flight Status

Description: Staff can update the status of flights.

Query:

```
UPDATE flight
SET status = [new_status]
WHERE airline_name = [staff_airline] AND flight_number = [flight_number]
```

Add Airplane

Description: Staff can add new airplanes to their airline's fleet.

Query:

```
INSERT INTO airplane (airline_name, airplane_ID, num_seats, manufacturing_company)
VALUES([airline_name], [airplane_id], [num_seats], [manufacturing_company])
```

Add Airport

Description: Staff can add new airports to the system.

Query:

```
INSERT INTO airport (code, name, city, country)
VALUES([code], [name], [city], [country])
```

View Flight Ratings

Description: Staff can view average ratings and comments for flights.

Queries:

```
-- View average ratings
SELECT flight_number, departure_datetime,
       ROUND(AVG(review), 2) AS avg_review, COUNT(*) AS num_reviews
FROM review
WHERE airline_name = [staff_airline]
GROUP BY flight_number, departure_datetime
ORDER BY departure_datetime DESC

-- View detailed comments
SELECT flight_number, departure_datetime, review, comment
FROM review
WHERE airline_name = [staff_airline]
ORDER BY departure_datetime DESC
```

View Reports

Description: Staff can view ticket sales and revenue reports.

Query:

```
SELECT f.flight_number, COUNT(p.ticket_ID) AS tickets_sold,
       SUM(p.sold_price) AS total_revenue
FROM flight f
LEFT JOIN ticket t ON f.flight_number = t.flight_number
    AND f.airline_name = t.airline_name
LEFT JOIN purchase p ON t.ticket_ID = p.ticket_ID
WHERE f.airline_name = [staff_airline]
GROUP BY f.flight_number
```

Logout

Description: End the staff member's session.

Implementation Note: This is handled at the application level by destroying the session, no direct database queries are involved.