Brandon Kim bmk7319, Serena Nam sn3659

Homepage

View public info

- **Search for future flights**

  Round trip:

  query = '"SELECT * FROM Flight WHERE
  (departure_airport_code = “{{departure_airport_code}}” and arrival_airport_code = “{{arrival_airport_code}}” and DATE(departure_date_time) = “{{departure_date}}”)
  OR (departure_airport_code = “{{arrival_airport_code}}” and
  arrival_airport_code = “{{departure_airport_code}}” and
  DATE(arrival_date_time) = “{{return_date}}”)"'

  One way:

  query = '"SELECT * FROM Flight WHERE
  departure_airport_code = “{{departure_airport_code}}” and arrival_airport_code = “{{arrival_airport_code}}” and DATE(departure_date_time) = “{{departure_date}}”"'

  See flight status

  query = '"SELECT * FROM Flight WHERE
  airline_name = “{{airline}}” and
  flight_num = “{{flight_num}}” and
  DATE(departure_date_time) = “{{departure_date}}” "'

Register

- **Register customer**

  ins = 'INSERT INTO customer VALUES(“{{fname}}”, “{{lname}}”, “{{email}}, “{{password}}”, “{{building}}”, “{{street}}”, “{{apt}}”, “{{city}}”, “{{state}}”, “{{zipcode}}”, “{{passport}}”,”{{expiration}}”, “{{country}}”,”{{dob}}”),

  insPhone = 'INSERT INTO customer_phone VALUES(“{{email}}”, %s)'

- **Register airline staff**

  insert_q = 'INSERT INTO AirlineStaff
  VALUES(“{{username}}”,”{{password}}”,”{{fname}}”,”{{lname}}”,”{{dob}}”)'

  insert_email_q = 'INSERT INTO AirlineStaff_email VALUES(“{{username}}”,”{{email}}”)'

  insert_phone_q = 'INSERT INTO AirlineStaff_phone VALUES(“{{username}}”, “{{phone}}”)'

  insert_works_q = 'INSERT INTO works VALUES(“{{username}}”, “{{airline_name}}”)'

Login

- **Login customer**

query = 'SELECT * FROM customer WHERE email = “{{email}}” and password = {{password}}”'

- **Login airline staff**

query = 'SELECT * FROM AirlineStaff WHERE username = “{{username}}” and password = “{{password}}”'

query = 'SELECT airline_name FROM works WHERE username = “{{username}}”'

Customer use cases

View my flights

customer’s upcoming flights:

query = '''SELECT *
FROM purchases NATURAL JOIN flight
WHERE email=”{{email}}” and departure_date_time > “{{curDate}}” GROUP BY airline_name, airplane_ID, flight_num, departure_date_time'''

customer’s past flights:

query = '''SELECT *
FROM purchases NATURAL JOIN flight NATURAL LEFT OUTER JOIN reviews
WHERE email=”{{email}}” and arrival_date_time < “{{curDate}}” GROUP BY airline_name, airplane_ID, flight_num, departure_date_time'''

Search for flights

Round trip:

query = '''SELECT airline_name, airplane_ID, flight_num, departure_date_time, arrival_date_time, departure_airport_code, arrival_airport_code, flight_status, ROUND(IF (num_seats*0.8 <= num_purchased, base_ticket_price*1.25, base_ticket_price),2) as ticket_price
FROM
(SELECT airline_name, airplane_ID, flight_num, departure_date_time,
COUNT(*) as num_purchased from Purchases
GROUP BY airline_name, airplane_ID, flight_num, departure_date_time) as C
NATURAL JOIN Airplane NATURAL RIGHT OUTER JOIN Flight
WHERE num_seats > num_purchased AND (departure_airport_code = “{{departure_airport_code}}” and
arrival_airport_code = “{{arrival_airport_code}}” and DATE(departure_date_time) = “{{departure_date}}”)
OR (departure_airport_code = “{{arrival_airport_code}}” and arrival_airport_code = “{{departure_airport_code}}” and DATE(arrival_date_time) = “{{return_date}})'''

One way:

query = '''SELECT airline_name, airplane_ID, flight_num, departure_date_time, arrival_date_time, departure_airport_code, arrival_airport_code, flight_status, ROUND(IF

(num_seats*0.8 <= num_purchased, base_ticket_price*1.25, base_ticket_price),2) as ticket_price
FROM
(SELECT airline_name, airplane_ID, flight_num, departure_date_time,
COUNT(*) as num_purchased from Purchases
GROUP BY airline_name, airplane_ID, flight_num, departure_date_time) as C
NATURAL JOIN Airplane NATURAL RIGHT OUTER JOIN Flight
WHERE num_seats > num_purchased and departure_airport_code =
“{{departure_airport_code}}” and arrival_airport_code = “{{arrival_airport_code}}” and
DATE(departure_date_time) = “{{departure_date}}””'

Purchase tickets
Check for unclaimed ticket:
query = 'SELECT MAX(ticket_ID) as exist_ticket FROM ticket NATURAL LEFT OUTER JOIN purchases where airline_name=”{{airline_name}}” and airplane_ID=”{{airplane_id}} and flight_num=”{{flight_num}}” and departure_date_time=”{{departure_date_time}}” and email is NULL'
Create new ticket:
query = 'SELECT COUNT(ticket_ID) as maxTicket FROM ticket'
ins = 'INSERT INTO ticket
VALUES(”{{ticket_id}}”,”{{airline_name}}”,”{{airplane_id}}”,”{{flight_num}}”,
“{{departure_date_time}}”)'
Create new purchase
ins = 'INSERT INTO purchases
VALUES(“{{email}}”,”{{ticket_id}}”,”{{airline_name}}”,”{{airplane_id}}”,”{{flight_num}}”,
“{{departure_date_time}}”,”{{first_name}}”,”{{last_name}}”,”{{date_of_birth}}”,”{{ticket_price}}”,”{{card_type}}”,”{{card_number}}”,”{{card_expiration}}”,”{{card_name}}”,”{{today}}”)'

Cancel Purchase
NOTE: this button is only rendered on flights with a departure date 24hrs later from the current time
query = 'DELETE FROM purchases WHERE email=“{{email}}” and ticket_ID=”{{ticket_id}}” and airline_name=”{{airline_name}}” and airplane_ID=”{{airplane_id}}” and flight_num=”{{flight_num}}” and departure_date_time=”{{departure_date_time}}” and first_name=”{{first_name}}” and last_name=”{{last_name}}”'

Rate/comment on past flights
NOTE: this button is only rendered on previous flights
ins = 'INSERT INTO reviews
VALUES(“{{email}}”,”{{ticket_id}}”,”{{airline_name}}”,”{{airplane_id}}”,”{{flight_num}}”,
“{{departure_date_time}}”, “{{rating}}”, “{{comment}}”)'

Track spending
Total spending:

```
query = f"'SELECT SUM(ticket_price) as total FROM purchases WHERE
                email="{email}" and
                purchase_date_time >= "{date_range_begin}" and
                purchase_date_time <= "{date_range_end}" "'
```

Spending by months

```
query = f"'SELECT DATE_FORMAT(purchase_date_time, '%m.%Y') as month,
                SUM(ticket_price) as total FROM purchases WHERE
                email="{email}" and
                purchase_date_time >= "{date_range_begin}" and
                purchase_date_time <= "{date_range_end}"
                GROUP BY DATE_FORMAT(purchase_date_time, '%m.%Y')"'
```

Airline staff use cases

View flights

gets matching flights in the date range, as well as average rating for each flight
NOTE: this query is executed in the filterFlights function
NOTE: some fields are not included if not specified

```
SELECT *, (
        SELECT AVG(rating) FROM reviews WHERE
        Flight.airline_name = airline_name and
        Flight.airplane_ID = airplane_ID and
        Flight.flight_num = flight_num and
        Flight.departure_date_time = departure_date_time
) as avg_rating
 FROM Flight WHERE
airline_name = "{airline_name}" and
departure_airport_code = "{departure_airport_code}" and
arrival_airport_code = "{arrival_airport_code}" and
DATE(departure_date_time) >= "{date_range_begin}" and
DATE(arrival_date_time) <= "{date_range_end}" and ' if date_range_end else
ORDER BY departure_date_time DESC
```

Create new flights

NOTE: checks for conflicting maintenance and flights before executing query

```
INSERT INTO Flight VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
cursor.execute(query, (airline_name, airplane_ID, flight_num, departure_date_time,
arrival_date_time, base_ticket_price, flight_status, departure_airport_code,
arrival_airport_code))
```

Change status of flights

```
UPDATE Flight SET flight_status="{new_status}" WHERE
        airline_name="{airline_name}" and
        airplane_ID="{airplane_ID}" and
        flight_num="{flight_num}" and
        departure_date_time="{departure_date_time}"
```

Add new airplane to system

```
INSERT INTO Airplane VALUES ("{airline_name}", "{airplane_ID}", {num_seats},
"{manufacturing_company}", "{model_num}", "{manufacture_date}")
```

Add new airport to system

```
INSERT INTO Airport VALUES ("{airport_code}", "{name}", "{city}", "{country}", {num_terminals},
"{airport_type}")
```

View flight ratings

NOTE: this is included under the staff-customers-on-flight page

```
SELECT * FROM reviews LEFT JOIN Customer on reviews.email = customer.email WHERE
        airline_name = "{airline_name}" and
        airplane_ID = "{airplane_ID}" and
        flight_num = "{flight_num}" and
        departure_date_time = "{departure_date_time}"
        ORDER BY rating DESC
```

Schedule maintenance

NOTE: checks for conflicting maintenance and flights before executing query

```
INSERT INTO Maintenance VALUES ("{airline_name}", "{airplane_ID}", "{start_date_time}",
"{end_date_time}")
```

View frequent customers

NOTE: this is included under the staff-customers-on-flight page

```
CREATE VIEW Customer_freq AS SELECT Customer.email as email, COUNT(*) as freq
        FROM Purchases JOIN Customer ON Purchases.email=Customer.email
        WHERE airline_name="{session["airline"]}" and
        departure_date_time >= "{today_minus_1yr}"
        GROUP BY email;
SELECT first_name, last_name, Customer.email, freq
        FROM Customer JOIN Customer_freq ON Customer.email=Customer_freq.email
```

```
        WHERE freq=(SELECT max(freq) FROM Customer_freq);
DROP VIEW Customer_freq;
```

View earned revenue

Total in past month

```
SELECT sum(ticket_price) as total FROM purchases
        WHERE airline_name="{session[‘airline’]" and
        purchase_date_time >= "{today_minus_1mo}";
```

Total in past year

```
SELECT sum(ticket_price) as total FROM purchases
                WHERE airline_name="session[‘airline’]" and
                purchase_date_time >= "{today_minus_1yr}";
```

Revenue by month

```
SELECT MONTH(purchase_date_time) as month, count(*) as tickets, sum(ticket_price) as
revenue
        FROM purchases WHERE
        airline_name="{session['airline']}" and
        purchase_date_time >= "{today_minus_1yr}"
        group by MONTHNAME(purchase_date_time);
```