Zhangyi Wang

zw1800

**SQL Queries**

**Part 1: General Features**

**View Public Information**

* + - SELECT airline\_name, flight\_num, departure\_airport, departure\_time, arrival\_airport, arrival\_time, price, status

FROM flight

ORDER BY departure\_time DESC

* + - Select and sort everything in flight (except the plane num)

**Public Search** (search for targeted public information, I = input)

* + - By airport
      * …WHERE departure/arrival\_airport = I …
    - By city
      * … FROM flight NATURAL JOIN airport

WHERE departure/arrival\_airport = airport\_name AND airport\_city = I …

* + - By status
      * …WHERE status = I …
    - By date
      * …WHERE departure/arrival\_time > I\_start and departure/arrival\_time < I\_end
      * I\_start and I\_end are in datetime format

**Registration** (customer, I = input)

Server checks if the email is taken

* + - SELECT \* FROM customer WHERE email = I[email]…

If the email is available:

* + - INSERT INTO customer VALUES (I…)
    - The password is hashed before insertion
    - Redirect to the login page

Else:

* + - Return the email, display an error message

**Login** (customer, I = input)

SELECT name FROM customer WHERE (email, password) = (I…)

If the returned data is not none, add session variables, display the customer home.

* + - If the email/username does not match up, all session variables are cleared, and the user is redirected to the home page

Else, display an error message

**Log Out**

Clear session variables, redirect to the home page

**Part 2: Customer features**

**Customer Home**

* + - First 10 Upcoming Flights:

SELECT airline\_name, flight\_num, departure\_airport, departure\_time, arrival\_airport, arrival\_time, price, status, ticket\_id, purchase\_date

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases WHERE customer\_email = I AND status = "Upcoming"

ORDER BY departure\_time DESC LIMIT 10

* + - Track Spending Data:

SELECT price, ticket\_id, purchase\_date

FROM purchases NATURAL JOIN ticket NATURAL JOIN flight WHERE customer\_email = I

ORDER BY purchase\_date ASC

**View my Flights**

* + - Fetch All Flights:

SELECT airline\_name, flight\_num, departure\_airport, departure\_time, arrival\_airport, arrival\_time, price, status, ticket\_id, purchase\_date

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases WHERE customer\_email = I

ORDER BY departure\_time DESC

**Purchase/View Available flights**

* + - Fetch All Available Flights:

SELECT airline\_name, flight\_num, departure\_airport, departure\_time, arrival\_airport, arrival\_time, price, seats, seats - count(ticket\_id) AS available\_seats

FROM flight NATURAL JOIN airplane NATURAL LEFT OUTER JOIN ticket

WHERE status = "Upcoming"

GROUP BY airline\_name, flight\_num

HAVING seats > count(ticket\_id)

ORDER BY departure\_time DESC

* + - At purchase:
      * Check if the Flight has Available Seats:

SELECT \*

FROM flight NATURAL JOIN ticket NATURAL JOIN airplane

WHERE status = "Upcoming" AND (airline\_name, flight\_num) = I

GROUP BY airline\_name, flight\_num

HAVING seats > count(ticket\_id)

* + - * For Auto-incrementing the Ticket Number:

SELECT MAX(ticket\_id) AS max\_id

FROM ticket

* + - * If Available to Insert:

INSERT INTO ticket VALUES (I)

INSERT INTO purchases(ticket\_id, customer\_email, purchase\_date) VALUES (I)

* + - * Else, display an error message.

**Part 3: Agent Features**

**Agent Home**

* + - * My Upcoming Flights:

Similar to the customer’s case

* + - * Fetch Commission Data:

SELECT price, customer\_email, purchase\_date

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases

WHERE booking\_agent\_id = I

ORDER BY purchase\_date DESC

**View my Flights**:

* + - * Similar to the customer’s case

**Purchase Tickets:**

* + - * Similar to the customer’s case
      * Check first whether agent works for the airline:

SELECT \*

FROM booking\_agent\_work\_for

WHERE email = “I” AND airline\_name = “I”

* + - * If not, display an error message. Purchase fails.

**Part 4: Staff Features**

**Staff Home**

* + - * Upcoming Flights:

SELECT \*

FROM flight

WHERE airline\_name = I AND status = "Upcoming" AND departure\_time > "I" AND departure\_time < "I"

ORDER BY departure\_time DESC LIMIT 10

**Top Booking Agents**

* + - * By Commission:

SELECT booking\_agent\_id, email, SUM(price) \* 0.1 AS commission

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases NATURAL JOIN booking\_agent

WHERE airline\_name = "I" AND purchase\_date > "I"

GROUP BY booking\_agent\_id

ORDER BY SUM(price) DESC LIMIT 5

* + - * By Sales:

SELECT booking\_agent\_id, email, COUNT(price) AS n\_sales

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases NATURAL JOIN booking\_agent

WHERE airline\_name = "I" AND purchase\_date > "I" GROUP BY booking\_agent\_id

ORDER BY COUNT(price) DESC LIMIT 5

**Top Customers**

* + - * SELECT email, name, COUNT(ticket\_id)

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases JOIN customer ON customer.email = purchases.customer\_email

WHERE airline\_name = "I" AND purchase\_date > "I"

GROUP BY email

ORDER BY COUNT(ticket\_id) DESC LIMIT 5

**Top Destinations**

* + - * SELECT airport\_city, COUNT(\*)

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases, airport

WHERE flight.arrival\_airport = airport.airport\_name AND airline\_name = "I" AND purchase\_date > "I"

GROUP BY airport\_city

ORDER BY COUNT(\*) DESC LIMIT 3

**For Sales Statics**

* + - * SELECT purchase\_date

FROM purchases NATURAL JOIN ticket NATURAL JOIN flight

WHERE airline\_name="I"

ORDER BY purchase\_date ASC

**For Sales Breakdown**

* + - * SELECT SUM(price)

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases

WHERE airline\_name = "I" AND purchase\_date > "I"

**View My Flights/View Customers/Change Status**

* + - SELECT \*

FROM flight

WHERE airline\_name = I

ORDER BY departure\_time DESC

* + - View Customers of a Specific Flight
      * SELECT email, name, ticket\_id

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases, customer

WHERE customer.email = purchases.customer\_email AND (airline\_name, flight\_num) = (I)

GROUP BY email

ORDER BY ticket\_id

* + - Change Status
      * SELECT email, name, ticket\_id

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases, customer

WHERE customer.email = purchases.customer\_email AND (airline\_name, flight\_num) = (I)

GROUP BY email

ORDER BY ticket\_id

**Add Flight**

* + - Fetch Available Airports, Planes and Add with Auto-increment Flight Number
      * SELECT DISTINCT airport\_name

FROM airport

* + - * SELECT DISTINCT airplane\_id

FROM airplane

WHERE airline\_name = I

* + - * SELECT MAX(flight\_num)

FROM flight

WHERE airline\_name = I

* + - Check for Time Conflicts
      * SELECT \*

FROM flight

WHERE (airline\_name = "I" AND airplane\_id = I AND ((departure\_time > 'I' AND departure\_time < 'I') OR (arrival\_time > 'I' AND arrival\_time < 'I') OR (departure\_time < 'I' AND arrival\_time > 'I))) OR 'I' >= 'I'

* + - Insert if Available
      * INSERT INTO flight VALUES (I)

**Add Plane**

* + - For Auto Incrementing the Plane id:

SELECT MAX(airplane\_id) AS MAX\_id

FROM airplane

WHERE airline\_name = I

* + - Insert if Available

INSERT INTO airplane VALUES (I)

* + - Fetch All Planes:

SELECT \*

FROM airplane

WHERE airline\_name = I

**Add Airport**

* + - Check whether staff has “Admin”permission:

SELECT permission\_type

FROM permission

WHERE username = “I”

If staff has permission:

* + - Fetch All Airports:

SELECT \*

FROM airport

* + - Insert if Available:

INSERT INTO airport VALUES (I)

Else, display an error message, return to staff home.

**View Agents**

* + - SELECT email, booking\_agent.booking\_agent\_id, COUNT(price), SUM(price)\*0.1

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases RIGHT JOIN booking\_agent ON purchases.booking\_agent\_id = booking\_agent.booking\_agent\_id

WHERE airline\_name = "{}"

GROUP BY booking\_agent.booking\_agent\_id

ORDER BY booking\_agent.booking\_agent\_id

**View Customers**

* + - SELECT email, name, COUNT(ticket\_id)

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases JOIN customer ON customer.email = purchases.customer\_email WHERE airline\_name = "I"

GROUP BY email

ORDER BY COUNT(ticket\_id) DESC

* + - Further Customer Details
      * SELECT flight\_num, ticket\_id

FROM flight NATURAL JOIN ticket NATURAL JOIN purchases

WHERE airline\_name = "I" AND customer\_email = "I"

**Add Permission to a new staff**

* + - Check whether the current staff has “Admin”permission:

SELECT permission\_type

FROM permission

WHERE username = “I”

If staff has permission:

* + - Fetch All “Admin”permissions:

SELECT \*

FROM permissions

* + - Check whether the new staff works for the same airline as the current staff (fetch the airline\_name of the new staff)

SELECT airline\_name

FROM airline\_staff

WHERE username = “I”

If so:

* + - Insert if Available:

INSERT INTO permission VALUES (I)

Else, display an error message, return to staff home.

**Add New Agents**

* + - Check whether the current staff has “Admin”permission:

SELECT permission\_type

FROM permission

WHERE username = “I”

If staff has permission:

* + - Fetch All Added booking agents:

SELECT \*

FROM booking\_agent\_work\_for

If adding the agent to the same airline the staff works for:

* + - Insert if Available:

INSERT INTO booking\_agent\_work\_for VALUES (I)

Else, display an error message, return to staff home.