Yukun Jiang (jy2363), Zheng Wang (zw1454)

CSCI-SHU 213 Databases (Fall 2020)
Final Project - User Case Query and Explanation

Please see below for the queries executed and their brief descriptions under each use case.

1. View Public Info

Find all search keys for user to select on a dropdown list:

```
SELECT DISTINCT arrival_airport FROM flight;
SELECT DISTINCT departure_airport FROM flight;
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
arrival_airport FROM flight);
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
departure_airport FROM flight);
```

Select the flights according to the search information specified by the user:

```
SELECT * FROM flight
WHERE departure_airport IN (SELECT airport_name FROM airport WHERE
airport_city=\"{}\") AND status = 'Upcoming' AND departure_airport = \"{}\"
AND arrival_airport IN (SELECT airport_name FROM airport WHERE airport_city=\"{}\"
AND  arrival_airport =\"{}\" AND DATE(departure_time) = DATE(\"{}\");
```

Select flight information and status according to the information specified by the user:

```
SELECT airline_name, flight_num, departure_time, arrival_time, status
FROM flight
WHERE flight_num = \'{}\' AND DATE(arrival_time) = DATE(\'{}\') AND
DATE(departure_time) = DATE(\'{}\') ;
```

2. Register

Check whether the account already exists:

```
SELECT * FROM customer WHERE email = \'{}\';
SELECT * FROM booking_agent WHERE email = \'{}\';
SELECT * FROM airline_staff WHERE username = \'{}\’;
```

Insert the registered account into the database:

```
INSERT INTO customer VALUES(\'{}\', \'{}\', MD5(\'{}\'), \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\',
\'{}\', \'{}\', \'{}\' );
INSERT INTO booking_agent VALUES(\'{}\', MD5(\'{}\'), \'{}\');
INSERT INTO airline_staff VALUES(\'{}\', MD5(\'{}\'), \'{}\', \'{}\', \'{}\', \'{}\');
```

3. Login

Check whether the login info matches the record of the database:

```
SELECT * FROM customer WHERE email = \'{}\' and password = MD5(\'{}\');
SELECT * FROM booking_agent WHERE email = \'{}\' and password = MD5(\'{}\');
SELECT * FROM airline_staff WHERE username = \'{}\' and password = MD5(\'{}\');
```

4. Logout

Customer use cases:

5. View my flights

Select all the upcoming flights of this user:

```
SELECT airline_name, flight_num, departure_airport, departure_time, arrival_airport,
arrival_time, status FROM purchases NATURAL JOIN (ticket NATURAL JOIN flight)
WHERE customer_email = \'{}\' AND status = 'Upcoming';
```

6. Purchase tickets

Check if duplicate purchase exists:

```
SELECT * FROM purchases NATURAL JOIN ticket
WHERE customer_email=\'{}\' AND airline_name=\'{}\' AND flight_num=\'{}\';
```

Insert purchased ticket into the database:

```
INSERT INTO ticket VALUES (\'{}\', \'{}\', \'{}\');
```

Insert purchase log into the database:

```
INSERT INTO purchases VALUES (\'{}\', \'{}\', NULL , DATE(NOW()) );
```

7. Search for flights

Find all search keys for user to select on a dropdown list:

```
SELECT DISTINCT arrival_airport FROM flight;
SELECT DISTINCT departure_airport FROM flight;
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
arrival_airport FROM flight);
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
departure_airport FROM flight);
```

Select the flights according to the search information specified by the user:

```
SELECT * FROM flight
WHERE departure_airport IN (SELECT airport_name FROM airport WHERE
airport_city=\"{}\") AND status = 'Upcoming' AND departure_airport = \"{}\"
AND arrival_airport IN (SELECT airport_name FROM airport WHERE airport_city=\"{}\"
AND  arrival_airport =\"{}\" AND DATE(departure_time) = DATE(\"{}\");
```

8. Track my spending

Check total spending in the past year:

```
SELECT SUM(price) FROM purchases NATURAL JOIN ticket NATURAL JOIN flight
WHERE customer_email= \'{}\' AND purchase_date BETWEEN date_sub(DATE(NOW()),
interval 1 year) AND DATE(NOW());
```

For generating a bar chart for the spending in the past 6 months:

```
SELECT YEAR(date_sub(DATE(NOW()), interval {} month)) as year ,
MONTH(date_sub(DATE(NOW()), interval {} month)) as month, SUM(price)
FROM ticket NATURAL JOIN purchases NATURAL JOIN flight
WHERE customer_email=\'{}\' AND YEAR(purchase_date)=YEAR(date_sub(DATE(NOW()),
interval {} month))  AND MONTH(purchase_date) = MONTH(date_sub(DATE(NOW()),
interval {} month));
```

Customize the time interval for spending tracking:

```
SELECT SUM(price)
FROM purchases NATURAL JOIN ticket NATURAL JOIN flight
WHERE customer_email= \'{}\' AND (YEAR(purchase_date) BETWEEN \'{}\' AND \'{}\')
AND (MONTH(purchase_date) BETWEEN \'{}\' AND \'{}\');
```

Booking agent use cases:

9. View my flights

Find all upcoming flights the agent ordered for customers:

```
SELECT f.airline_name, f.flight_num, f.departure_airport, f.departure_time, f.arrival_airport,
f.arrival_time, f.price, f.status, f.airplane_id, p.customer_email
FROM purchases p, ticket t, flight f
WHERE p.booking_agent_id = \'{}\' AND p.ticket_id = t.ticket_id AND t.airline_name =
f.airline_name AND t.flight_num = f.flight_num AND f.status = 'Upcoming' ;
```

Find all search keys for user to specify a range of search on a dropdown list:

```
SELECT DISTINCT arrival_airport FROM flight;
SELECT DISTINCT departure_airport FROM flight;
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
arrival_airport FROM flight);
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
departure_airport FROM flight);
SELECT DISTINCT flight_num FROM flight ;
```

Show all the flights the agent ordered for customers based on customized search:

```
SELECT * FROM flight f, ticket t, purchases p
```

```
WHERE f.departure_airport IN (SELECT airport_name FROM airport WHERE
airport_city=\'{}\') AND f.departure_airport = \"{}\" AND f.arrival_airport IN (SELECT
airport_name FROM airport WHERE airport_city=\'{}\') AND f.arrival_airport = \"{}\"
AND (DATE(f.departure_time) BETWEEN DATE(\'{}\') AND DATE(\'{}\') )
AND p.booking_agent_id = \'{}\' AND p.ticket_id = t.ticket_id AND t.airline_name =
f.airline_name AND t.flight_num = f.flight_num ;
```

10. Purchase tickets

Show all the possible customers’ email that the agent may purchase ticket for：

```
SELECT DISTINCT email FROM customer
```

Check if this customer has already held a ticket for this particular flight：

```
SELECT * FROM purchases NATURAL JOIN ticket WHERE customer_email=\'{}\' AND
airline_name=\'{}\' AND flight_num=\'{}\' ;
```

Create a ticket for this customer to purchase based on so far biggest ticket number:

```
SELECT MAX(ticket_id) FROM ticket;
INSERT INTO ticket VALUES (\'{}\', \'{}\', \'{}\') ;
```

Find the agent id for such representative purchase:

```
SELECT booking_agent_id FROM booking_agent WHERE email = \'{}\' ;
```

Finally, insert into the database such representative purchase records for this customer:

```
INSERT INTO purchases VALUES (\'{}\', \'{}\', \'{}\', DATE(NOW()) );
```

11. Search for flights

Extract all existing arrival_airport, departure_airport, arrival_city,departure_city to create a roll-down select menu:

```
SELECT DISTINCT arrival_airport FROM flight;
SELECT DISTINCT departure_airport FROM flight;
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
arrival_airport FROM flight);
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
departure_airport FROM flight);
```

Grab information based on customer's flight search selection:

```
SELECT * FROM flight WHERE status = 'Upcoming' AND departure_airport IN (SELECT
airport_name FROM airport WHERE airport_city=\'{}\')  AND departure_airport = \"{}\" AND
```

```
arrival_airport IN (SELECT airport_name FROM airport WHERE airport_city=\'{}\') AND
arrival_airport =\"{}\" AND DATE(departure_time) = DATE(\'{}\');
```

Show how many seats are available for this flight and to see if it’s still available to purchase:

(a) Get the airplane id for that flight:

```
SELECT airplane_id FROM flight WHERE airline_name = \'{}\' AND flight_num = \'{}\';
```

(b) Find the total available seats for that airplane:

```
SELECT seats FROM airplane WHERE airline_name = \'{}\' AND airplane_id = \'{}\';
```

(c) Find how many seats already sold for that flight:

```
SELECT COUNT(*) FROM ticket WHERE airline_name = \'{}\' AND  flight_num = \'{}\';
```

12. View my commission

Find total commission in the past 30 days:

```
SELECT 0.1*SUM(f.price) FROM purchases p, ticket t, flight f WHERE p.booking_agent_id =
\'{}\' AND p.ticket_id = t.ticket_id AND t.airline_name = f.airline_name AND t.flight_num =
f.flight_num AND p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -30 DAY);
```

Find total tickets sold in the past 30 days:

```
SELECT COUNT(*) FROM purchases p WHERE p.booking_agent_id = \'{}\' AND
p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -30 DAY);
```

Find total commission in selected customized period:

```
SELECT 0.1*SUM(f.price) FROM purchases p, ticket t, flight f WHERE p.booking_agent_id =
\'{}\' AND p.ticket_id = t.ticket_id AND t.airline_name = f.airline_name AND t.flight_num =
f.flight_num AND (p.purchase_date BETWEEN \'{}\' AND \'{}\' );
```

Find total tickets sold in selected customized period:

```
SELECT COUNT(*) FROM purchases p WHERE p.booking_agent_id = \'{}\' AND
(p.purchase_date BETWEEN \'{}\' AND \'{}\' );
```

13. View top customers

Find top customers in past 6 months based on tickets sold:

```
SELECT p.customer_email, COUNT(p.customer_email), c.name, c.phone_number, c.city,
c.state, c.date_of_birth FROM purchases p, customer c WHERE p.booking_agent_id = \'{}\'
AND p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -6 MONTH) AND
p.customer_email = c.email GROUP BY p.customer_email ORDER BY
COUNT(p.customer_email) DESC;
```

Find top customers in past 6 months based on commission earned:

```
SELECT p.customer_email, 0.1 * SUM(f.price), c.name, c.phone_number, c.city, c.state,
c.date_of_birth FROM purchases p, ticket t, flight f, customer c WHERE p.booking_agent_id =
\'{}\' AND p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -6 MONTH) AND
p.ticket_id = t.ticket_id AND t.airline_name = f.airline_name AND t.flight_num = f.flight_num
AND p.customer_email = c.email GROUP BY p.customer_email ORDER BY SUM(f.price)
DESC;
```

Airline staff use cases:

14. View my flights

Find all search keys for user to specify a range of search on a dropdown list:

```
SELECT airline_name FROM airline_staff WHERE username = \'{}\';
SELECT * FROM flight WHERE airline_name = \'{}\' AND (departure_time BETWEEN
NOW() AND ADDTIME(NOW(), '30 0:0:0'));
SELECT DISTINCT arrival_airport FROM flight;
SELECT DISTINCT departure_airport FROM flight;
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
arrival_airport FROM flight);
SELECT DISTINCT airport_city FROM airport WHERE airport_name IN (SELECT
departure_airport FROM flight);
SELECT DISTINCT flight_num FROM flight WHERE airline_name = \'{}\';
```

Display the flight search result after customized search:

```
SELECT * FROM flight
WHERE airline_name = \'{}\' AND departure_airport IN (SELECT airport_name FROM airport
WHERE airport_city=\'{}\') AND departure_airport = \"{}\" AND arrival_airport IN (SELECT
airport_name FROM airport WHERE airport_city=\'{}\') AND arrival_airport =\"{}\" AND
( DATE(departure_time) BETWEEN DATE(\'{}\') AND DATE(\'{}\') )
AND ( DATE(departure_time) >= DATE(\'{}\') );
```

See the customers of a particular flight:

```
SELECT c.email, c.name, c.phone_number, c.city, c.state, c.date_of_birth
FROM ticket t,  purchases p, customer c WHERE c.email = p.customer_email
AND t.ticket_id = p.ticket_id AND t.airline_name = \'{}\' AND t.flight_num = \'{}\';
```

15. Create new flights

Create drop-down list for the user to create a new flight

```
SELECT airline_name FROM airline_staff WHERE username = \'{}\'
SELECT * FROM flight WHERE airline_name = \'{}\' AND (departure_time BETWEEN
NOW() AND ADDTIME(NOW(), '30 0:0:0'));
SELECT DISTINCT airport_name FROM airport;
SELECT DISTINCT airport_name FROM airport;
SELECT DISTINCT airport_city FROM airport;
SELECT DISTINCT flight_num FROM flight WHERE airline_name = \'{}\';
SELECT DISTINCT airplane_id FROM airplane WHERE airline_name = \'{}\';
```

Check if the created flight already exists:

```
SELECT * FROM flight WHERE airline_name = \'{}\' AND flight_num = \'{}\';
```

Insert the new flight into the database

```
INSERT INTO flight VALUES(\'{}\', \'{}\', \"{}\", \'{}\', \"{}\", \'{}\', \'{}\', \'{}\', \'{}\');
```

16. Change status of flights

List all the existing flights for the user to change status:

```
SELECT airline_name FROM airline_staff WHERE username = \'{}\';
SELECT DISTINCT flight_num FROM flight WHERE airline_name = \'{}\';
```

Update the status of the selected flight in the database:

```
UPDATE flight SET status = \'{}\'
WHERE flight.airline_name = \'{}\' AND flight.flight_num = \'{}\' ;
```

17. Add airplane in the system

Insert the airplane into the system and assign an airplane id to it:

```
SELECT MAX(airplane_id) FROM airplane WHERE airline_name = \'{}\';
INSERT INTO airplane VALUES(\'{}\', \'{}\', \'{}\');
```

Display all the airplanes owned by the user’s airline:

```
SELECT * FROM airplane WHERE airline_name = \'{}\';
```

18. Add new airport in the system

Check if the added airport already exists:

```
SELECT EXISTS(SELECT * FROM airport WHERE airport_name = \"{}\");
```

Insert the airport into the database:

```
INSERT INTO airport VALUES(\"{}\", \'{}\');
```

19. View all the booking agents

Find top agents in past month based on tickets sold:

```
SELECT b.email, b.booking_agent_id, COUNT(p.ticket_id)
FROM booking_agent b, purchases p
WHERE p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -1 MONTH) AND
p.booking_agent_id = b.booking_agent_id
GROUP BY b.email, b.booking_agent_id ORDER BY COUNT(p.ticket_id) DESC;
```

Find top agents in past year based on tickets sold:

```
SELECT b.email, b.booking_agent_id, COUNT(p.ticket_id)
FROM booking_agent b, purchases p
WHERE p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -12 MONTH) AND
p.booking_agent_id = b.booking_agent_id
GROUP BY b.email, b.booking_agent_id ORDER BY COUNT(p.ticket_id) DESC;
```

Find top agents in past year based on commission earned:

```
SELECT b.email, b.booking_agent_id, 0.1*SUM(f.price)
FROM booking_agent b, purchases p, ticket t, flight f
WHERE p.purchase_date >= ADDDATE(DATE(NOW()), INTERVAL -12 MONTH) AND
p.booking_agent_id = b.booking_agent_id AND f.flight_num = t.flight_num AND
f.airline_name = t.airline_name AND t.ticket_id = p.ticket_id
GROUP BY b.email, b.booking_agent_id ORDER BY SUM(f.price) DESC;
```

20. View frequent customers

Create a view to store the frequency of customers:

```
SELECT airline_name FROM airline_staff WHERE username = \'{}\';
CREATE VIEW customer_frequency AS(
SELECT p.customer_email AS email, COUNT(p.ticket_id) AS frequency
FROM purchases p, ticket t
WHERE t.airline_name = \'{}\' AND t.ticket_id = p.ticket_id
GROUP BY p.customer_email);
```

Create another view to store the maximum frequency of customers:

```
CREATE VIEW max_frequency AS( SELECT MAX(frequency) as max_f FROM
customer_frequency);
```

Select the most frequent customer:

```
SELECT c.email, name, city, state, phone_number, date_of_birth
FROM customer c, customer_frequency f, max_frequency m
WHERE c.email = f.email AND f.frequency = m.max_f;
```

Drop the temporary views after selecting the most frequent customer:

```
DROP VIEW customer_frequency;
DROP VIEW max_frequency;
```

List all flights a particular customer has taken:

```
SELECT airline_name FROM airline_staff WHERE username = \'{}\';
SELECT f.airline_name, f.flight_num, f.departure_airport, f.departure_time, f.arrival_airport,
f.arrival_time, f.price, f.status, f.airplane_id FROM flight f, purchases p, ticket t
WHERE p.customer_email = \'{}\' AND p.ticket_id = t.ticket_id AND t.airline_name =
f.airline_name AND t.flight_num = f.flight_num AND f.airline_name = \'{}\' ;
```

21. View reports

Extract which airline does this staff work for：

```
SELECT airline_name FROM airline_staff WHERE username=\'{}\';
```

Retrieve the exact date of last month and last year:

```
SELECT DATE(NOW());
```

Find date of last month:

```
SELECT DATE_SUB(DATE(NOW()), interval 30 DAY);
```

Find date of last year:

```
SELECT DATE_SUB(DATE(NOW()), interval 365 DAY);
```

Search for how many ticket sold last month:

```
SELECT COUNT(*) FROM purchases NATURAL JOIN flight NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND DATE(purchase_date) BETWEEN \'{}\' AND \'{}\';
```

Search for how many ticket sold last year:

```
SELECT COUNT(*) FROM purchases NATURAL JOIN flight NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND DATE(purchase_date) BETWEEN \'{}\' AND \'{}\':
```

Generate the default past year ticket sales report bar chart:

```
SELECT YEAR(date_sub(DATE(NOW()), interval {} month)) as year,
MONTH(date_sub(DATE(NOW()), interval {} month)) as month, COUNT(*) FROM ticket
NATURAL JOIN purchases NATURAL JOIN flight WHERE airline_name=\'{}\' AND
YEAR(purchase_date)=YEAR(date_sub(DATE(NOW()), interval {} month)) AND
MONTH(purchase_date) = MONTH(date_sub(DATE(NOW()), interval {} month));
```

Generate the customized ticket sales report bar chart:

```
SELECT YEAR(date_sub(DATE(\'{}\'), interval {} month)) as year,
MONTH(date_sub(DATE(\'{}\'), interval {} month)) as month, COUNT(*) FROM ticket
NATURAL JOIN purchases NATURAL JOIN flight WHERE airline_name=\'{}\' AND
YEAR(purchase_date)=YEAR(date_sub(DATE(\'{}\'), interval {} month)) AND
MONTH(purchase_date) = MONTH(date_sub(DATE(\'{}\'), interval {} month));
```

22. Comparison of revenue earned

Search for non-agent last month revenue:

```
SELECT SUM(price) FROM flight NATURAL JOIN purchases NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND booking_agent_id is NULL AND (purchase_date BETWEEN
DATE_SUB(DATE(NOW()), interval 30 DAY) AND DATE(NOW()) );
```

Search for agent last month revenue:

```
SELECT SUM(price) FROM flight NATURAL JOIN purchases NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND booking_agent_id is NOT NULL AND (purchase_date
BETWEEN DATE_SUB(DATE(NOW()), interval 30 DAY) AND DATE(NOW()) );
```

Search for non-agent last year revenue:

```
SELECT SUM(price) FROM flight NATURAL JOIN purchases NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND booking_agent_id is NULL AND (purchase_date BETWEEN
DATE_SUB(DATE(NOW()), interval 365 DAY) AND DATE(NOW()) );
```

Search for agent last year revenue:

```
SELECT SUM(price) FROM flight NATURAL JOIN purchases NATURAL JOIN ticket
WHERE airline_name=\'{}\' AND booking_agent_id is NOT NULL AND (purchase_date
BETWEEN DATE_SUB(DATE(NOW()), interval 365 DAY) AND DATE(NOW()) );
```

23. View top destinations

Query for top destinations in the past 3 months:

```
SELECT f.arrival_airport, a.airport_city FROM purchases p, ticket t, flight f, airport a WHERE
p.ticket_id = t.ticket_id AND t.airline_name = f.airline_name AND t.flight_num = f.flight_num
AND f.airline_name = \'{}\' AND (f.arrival_time BETWEEN ADDDATE(NOW(), INTERVAL
-3 MONTH) AND NOW()) AND a.airport_name = f.arrival_airport GROUP BY arrival_airport
ORDER BY COUNT(p.customer_email) DESC;
```

Query for top destinations in the past year:

```
SELECT f.arrival_airport, a.airport_city FROM purchases p, ticket t, flight f, airport a WHERE
p.ticket_id = t.ticket_id AND t.airline_name = f.airline_name AND t.flight_num = f.flight_num
AND f.airline_name = \'{}\' AND (f.arrival_time BETWEEN ADDDATE(NOW(), INTERVAL
-12 MONTH) AND NOW()) AND a.airport_name = f.arrival_airport GROUP BY
arrival_airport ORDER BY COUNT(p.customer_email) DESC;
```