Application use cases and the queries:

Public section: In this section, anyone will be able to use these features.

1. View public information

1.1 Search for future flight(one way)

‘SELECT \* FROM flight WHERE d\_date = %s

and dep\_airport\_code = (select code from airport where airport.name = %s)

and arr\_airport\_code = (select code from airport where airport.name = %s) '

This query selects the flight based on the departure date, departure airport, and arrival airport entered by the user. (select code from airport where airport.name = %s), a subquery, helps to find the code of the airport the user enters.

1.2 Search for round trip

‘SELECT \* FROM flight WHERE d\_date = %s

and dep\_airport\_code = (select code from airport where airport.name = %s)

and arr\_airport\_code = (select code from airport where airport.name = %s)'

This is the first query that finds all the one way flights from the departure airport to the arrival airport. (select code from airport where airport.name = %s), a subquery, helps to find the code of the airport the user enters.

It is essentially the same query for finding all the one way flights from the arrival airport to the departure airport. All we need to do is to pass the arrival airport first, followed by the departure airport.

1.3 Search for flight status

'SELECT \* FROM flight where airline\_name = %s and flight\_num = %s and a\_date = %s and d\_date = %s'

This query finds the flight information, including the status of the flight, based on the criterias entered by the user.

2. Register

2.1 Register for staff

‘SELECT \* FROM staff WHERE username = md5(%s)’

Check whether the username exists.

'SELECT \* FROM staff\_phone WHERE phone\_num = %s'

Check whether the phone number is already registered or not.

'SELECT \* FROM airline WHERE name = %s'

Check whether the airline company the staff is going to register for exists or not.

'INSERT INTO staff VALUES(%s, md5(%s),%s, %s, %s, %s)'

If the username does not exist, he/she can register an account, and the username and password and other information will be inserted into the database.

'INSERT INTO staff\_phone VALUES(%s, %s)'

This query inserts the staff phone number.

2.2 register for customer

'SELECT \* FROM customer WHERE email = %s'

This query checks whether the email is already registered by a customer.

'INSERT INTO customer VALUES(%s, %s, md5(%s), %s,%s,%s, %s,%s,%s, %s,%s,%s)'

(username, name, password,buildingNum,street,city,state,phoneNum,passportNum,passportExp,passportCountry,birthday)

If not, he/she can register an account, and the username, email address, and password, and other attributes will be inserted into the database.

3.login

3.1 login for staff

'SELECT \* FROM customer WHERE email = %s and password = md5(%s)'

This query matches staff in our database with credentials entered by the current staff. If the query finds a match, we allow the current staff to log in.

3.2 login for customer

'SELECT \* FROM staff WHERE username = %s and password = md5(%s)'

This query matches customers in our database with credentials entered by the current customer. If the query finds a match, we allow the current customer to log in.

Customer section:

4. View my flights

'select \* from ticket where customer\_email = %s and d\_date > current\_date or (d\_date = current\_date and d\_time > current\_time)'

We want to find the ticket information. ‘customer\_email = %s’ means we locate the tickets of this particular customer. ‘d\_date > current\_date or (d\_date = current\_date and d\_time > current\_time’ means we find the flights that depart after current time and on.

5. Search for flights(one way and round trip)

This part is redirected to the public search part. The only difference is that when there is a session[‘username’], the user will be able to check the price of a ticket and purchase tickets.

‘SELECT \* FROM flight WHERE d\_date = %s

and dep\_airport\_code = (select code from airport where airport.name = %s)

and arr\_airport\_code = (select code from airport where airport.name = %s)'

This is the first query that finds all the one way flights from the departure airport to the arrival airport. (select code from airport where airport.name = %s), a subquery, helps to find the code of the airport the user enters.

It is essentially the same query for finding all the one way flights from the arrival airport to the departure airport. All we need to do is to pass the arrival airport first, followed by the departure airport.

(This is basically the same query as the flight search services in the public home.)

6. Purchase tickets

6.1 Price check

There is an additional use case inside the purchase ticket. The customer should be able to check the price of the ticket.

'select \* from flight where airline\_name = %s and d\_date = %s and d\_time = %s 'and flight\_num = %s'

This query is used to find out whether the customer’s information indicates a real flight. If there is no such information, we will give out an error message.

'select count(\*) as number\_seat from ticket natural join flight ' \

'where flight\_num = %s and airline\_name = %s and d\_date = %s and d\_time = %s '

query = 'select number\_seat from flight join airline ON flight.airline\_name = airline.name ' \

'join airplane on flight.airplane\_i\_num = airplane.iden\_num ' \

'where flight\_num = %s and flight.airline\_name = %s and d\_date = %s and d\_time = %s'

query = 'select base\_price from flight ' \

'where flight\_num = %s and airline\_name = %s and d\_date = %s and d\_time = %s '

These three queries are used together to calculate the final price of the ticket. Compared the sold ticket from the ticket and the number of seat in the flight table, and decide whether we will \*1.25 to the base price.

6.2 purchase ticket

There is a check query similar to the first query of 6.1.

'select count(\*) as number\_seat from ticket natural join flight ' \

'where flight\_num = %s and airline\_name = %s and d\_date = %s and d\_time = %s '

This query finds the number of tickets that have been already sold when the customer entered the particular flight.

query = 'select number\_seat from flight join airline ON flight.airline\_name = airline.name ' \

'join airplane on flight.airplane\_i\_num = airplane.iden\_num ' \

'where flight\_num = %s and flight.airline\_name = %s and d\_date = %s and d\_time = %s'

This query finds the total number of seats for the particular flight entered by the customer.

'select base\_price from flight

where flight\_num = %s and airline\_name = %s and d\_date = %s and d\_time = %s’

This query finds the base price of the ticket of the flight entered by the customer.

'INSERT INTO ticket VALUES ((select (MAX(tID)+1) from ticket as t),%s,%s,%s,%s, current\_date(),current\_time() ,%s,%s,%s,%s,%s)'

‘(select (MAX(tID)+1) from ticket as t)’ means we need to auto increment the current max ticket ID and set it as the ticket ID for the new ticket. Then we insert all the needed information for the new ticket into the ticket table.

7. Giving Ratings and Comment on previous flights

‘select \* from ticket where customer\_email = %s

and airline\_name = %s and d\_date = %s and d\_time = %s

and flight\_num = %s'

This query checks whether the customer purchased the ticket for the flight he/she is going to rate on. If yes, we could grant the customer the right rate on the flight.

'select \* from rate where customer\_email = %s

and airline\_name = %s and d\_date = %s and d\_time = %s

and flight\_num = %s'

This query checks whether the customer has already rated the flight or not. We try to locate the rating of this particular customer in our database.

'insert into rate VALUES(%s,%s,%s,%s,%s,%s,%s)'

If the customer purchased the ticket for the particular flight and has never rated on this flight, the customer could rate on this flight. We would insert the comment into our database.

8. Track My Spending

8.1 Spending in past year

‘select sum(sold\_price) as sumPrice from ticket

where customer\_email = %s and (purchase\_date > current\_date() - interval 1 year) '

This query finds all the tickets purchased by this customer in the last year, and summed up the price.

8.2 Spending in range

'select sum(sold\_price) as sumPrice from ticket

'where customer\_email = %s and purchase\_date >= %s and purchase\_date <= %s '

This query finds all the tickets purchased by this customer in the given range entered by the customer, and summed up the price.

8.3 Spending chart default

'select month(purchase\_date) as m, year(purchase\_date) as y, sum(sold\_price) as sumPrice

from ticket

where customer\_email = %s

group by month(purchase\_date), year(purchase\_date)'

This query categories the spendings into month and year for the customer, and shows the corresponding month spendings.

8.4 Spending chart in range

'select month(purchase\_date) as m, year(purchase\_date) as y, sum(sold\_price) as sumPrice

from ticket

where customer\_email = %s and purchase\_date >= %s and purchase\_date <= %s

group by month(purchase\_date), year(purchase\_date)'

This query categories the spendings into month and year for the customer, with the starting date and ending date defined by the customer, and shows the corresponding month spendings.

9. Logout

There is no query for this use case.

Staff section:

10. View flights

10.1 view all the future flights operated by the airline he/she works for the next 30 days

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT flight\_num, d\_date, d\_time from flight WHERE airline\_name = %s and d\_date > current\_date() and d\_date < current\_date () + interval 1 month ;

This query select future flights of his/her airline in a month

10.2 see all the current/future/past flights operated by the airline he/she works for based on the range of dates

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT flight\_num, d\_date, d\_time FROM flight WHERE airline\_name = %s

and d\_date > %s and d\_date < %s

This query selects flights in his/her airline based on range of date.

10.3 see all the current/future/past flights operated by the airline he/she works for based

source/destination airports

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT flight\_num, d\_date, d\_time FROM flight WHERE

airline\_name = %s and

dep\_airport\_code = (select distinct code from airport where airline\_name = %s)

and arr\_airport\_code = (select code from airport where airline\_name = %s);

This query selects flights based on departure and arrival airports.

10.4 see all the customers of a particular flight

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT DISTINCT name, email FROM customer, ticket WHERE airline\_name = %s and flight\_num = %s and d\_date = %s and d\_time = %s and

email = customer\_email

This query selects distinct names and emails of customers of a particular flight.

11. Create new flights

​​'select \* from staff where username = %s'

This is a sanity check to see if the current user is a staff or not.

check0 = 'select \* from airplane where iden\_num = %s and airline\_name = ' \

'(select airline\_name from staff where username = %s)'

check0 = 'select \* from airport where code = %s'

check0 = 'select \* from airport where code = %s'

These three check0 are sanity checks for whether the airplane exists and belongs to the airline, whether the arrival airport and departure airport exists.

'SELECT \* FROM flight Where airline\_name = %s and d\_date = %s and d\_time = %s and flight\_num = %s and status = %s'

This query checks whether the corresponding flight already exists, with the information provided by the staff.

'INSERT INTO flight VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)'

If the flight does not exist, we insert the flight information into the table. These attributes are username, d\_date, d\_time, flight\_num, a\_date, a\_time, int(base\_price), status, airplane\_i\_num, arr\_code, dep\_code.

12. Change Status of lights

'select \* from staff where username = %s'

This is a sanity check to see if the current user is a staff or not.

'SELECT \* FROM flight WHERE flight\_num = %s and d\_date = %s and d\_time = %s' \

' and airline\_name = (select airline\_name from staff where username = %s )'

This query checks whether the flight exists or not.

'UPDATE flight SET status = %s WHERE flight\_num = %s and d\_date = %s and d\_time = %s' \

' and airline\_name = (select airline\_name from staff where username = %s )'

If the flight exists, the staff could update the status of the flight.

13. Add airplane in the system

'select \* from staff where username = %s'

This is a sanity check to see if the current user is a staff or not.

'SELECT airline\_name FROM staff where username = %s'

This query finds the current staff’s airline name. So, we can prevent him/her from adding airplanes for other companies.

'SELECT \* FROM airplane where iden\_num = %s and airline\_name = %s'

This query checks whether the airplane already exists or not.

'INSERT into airplane VALUES(%s, %s, %s)'

If the airplane does not exist, we create this airplane and insert it into our database.

'SELECT \* FROM airplane where airline\_name = %s'

Once the airplane is created, the staff can see all the airplanes owned by his/her airplane company.

14. Add new airport in the system

'select \* from staff where username = %s'

This is a sanity check to see if the current user is a staff or not.

'SELECT \* FROM airport where code = %s'

This query checks whether the airport already exists or not.

'INSERT into airport VALUES(%s, %s, %s)'

If the airport does not exist, we can add the airport into our database.

15. View flight ratings

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT avg(star) as num FROM rate WHERE airline\_name = %s and flight\_num = %s

and d\_date = %s and d\_time = %s

This query finds the average rating of a particular flight.

SELECT star, comment FROM rate WHERE airline\_name = %s and flight\_num = %s

and d\_date = %s and d\_time = %s

This query selects all stars and comments of a particular flight.

16. View frequent customers

16.1 see the most frequent customer within the last year

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT distinct name, email from customer, ticket WHERE airline\_name = %s

and email = customer\_email and

purchase\_date > current\_date () - interval 1 year GROUP BY customer\_email

ORDER BY COUNT(\*) DESC LIMIT 1

This query finds the names and emails of the most frequent customer in the airline within 1 year.

16.2 see a list of all flights a particular Customer has taken only on that particular airline

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

SELECT flight\_num, d\_date, d\_time FROM ticket WHERE airline\_name = %s

and customer\_email = %s;

This query finds flights a particular customer has taken in this airline.

17. View reports

17.1 Ticket year report

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

'select count(\*) as num from ticket

where purchase\_date > current\_date() - interval 1 year and airline\_name =

(select airline\_name from staff where username = %s) '

This query finds the total number sold within the 1 year interval belongs to this company.

' select month(purchase\_date) as m, year(purchase\_date) as y, count(\*) as num ' \

'from ticket ' \

'where purchase\_date > current\_date() - interval 1 year and airline\_name = (select airline\_name from staff where username = %s)' \

'group by month(purchase\_date), year(purchase\_date)'

This query provides a table format for the report within one year. All data will be group by the combination of month and year, to give the table form.

17.2 Ticket month report

This part is similar to 17.1.

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

' select count(\*) as num from ticket ' \

'where purchase\_date > current\_date() - interval 1 month and airline\_name = ' \

'(select airline\_name from staff where username = %s)'

This query finds the total number sold within the 1 month interval belongs to this company.

' select month(purchase\_date) as m, year(purchase\_date) as y, count(\*) as num ' \

'from ticket ' \

'where purchase\_date > current\_date() - interval 1 month and airline\_name = (select airline\_name from staff where username = %s)' \

'group by month(purchase\_date), year(purchase\_date)'

This query provides a table format for the report within 1 month. All data will be grouped by the combination of month and year, to give the table form.

17.3 Ticket range report

This part is similar to 17.1.

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

' select count(\*) as num from ticket ' \

'where purchase\_date >= %s and purchase\_date <= %s and airline\_name = ' \

'(select airline\_name from staff where username = %s)'

This query finds the total number sold within the given time interval belongs to this company.

' select month(purchase\_date) as m, year(purchase\_date) as y, count(\*) as num ' \

'from ticket ' \

'where purchase\_date >= %s and purchase\_date <= %s and airline\_name = (select airline\_name from staff where username = %s)' \

'group by month(purchase\_date), year(purchase\_date)'

This query provides a table format for the report within the given time interval. All data will be grouped by the combination of month and year, to give the table form.

18. View Earned Revenues

18.1 Revenus last year

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

'select sum(sold\_price) as revenue from ticket ' \

'where purchase\_date > current\_date() - interval 1 year \

' and airline\_name = %s'

This query finds the sum of all sold prices as the revenue in the last year, which means 365 days from today.

To clarify, we use the purchase date to calculate the revenue, instead of the departure date.

18.2 Revenues last month

This part is similar to 18.1.

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

'select sum(sold\_price) as revenue from ticket ' \

'where purchase\_date > current\_date() - interval 1 month \

' and airline\_name = %s'

This query finds the sum of all sold prices as the revenue in the last month.

19. View Top destinations

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

SELECT airline\_name FROM staff WHERE username = %s

This query finds the current staff’s airline name.

'select dep\_airport\_code, (select city from airport where code = dep\_airport\_code) as city ' \

'from ticket natural join flight ' \

'where d\_date > current\_date() - interval 3 month and airline\_name = %s' \

'group by dep\_airport\_code ' \

'order by count(\*) desc limit 3'

This query will find out the most popular by ordering it and only shows the first 3 pieces of information.

'select sum(sold\_price) as revenue from ticket ' \

'where purchase\_date > current\_date() - interval 1 month' \

' and airline\_name = %s'

This query will find out the most popular by ordering it and only shows the first 3 pieces of information.

20. Logout

There is no query for this use case.

21. Add phone number

This is an extra use case for a staff member to add his or her phone number.

select \* from staff where username = %s

This is a sanity check to see if the current user is a staff or not.

query0 = 'select \* from staff\_phone where phone\_num = %s'

This is used to check whether the phone number is used.

query = 'insert into staff\_phone values (%s,%s)'

This is used to add the information.