REQUIRED Application Use Cases (aka features):

View Public Info:

#For future flights: Search for future flights based on source city/airport name, destination city/airport name, departure date for one way (departure and return dates for round trip)

—---For go trip:

```
SELECT * FROM flight WHERE (departure_airport, arrival_airport, date(departure_datetime)) =
(%s,%s,%s) AND departure_datetime>CURRENT_TIMESTAMP
```

—---For round trip:

```
SELECT * FROM flight WHERE (departure_airport, arrival_airport, date(departure_datetime)) =
(%s,%s,%s) AND departure_datetime>CURRENT_TIMESTAMP
```

#For flight status: see the flights status based on airline name, flight number, arrival/departure

```
SELECT flight_status FROM flight WHERE
(airline_name,flight_number,date(departure_datetime)) = (%s,%s,%s)
```

Register:

#Customer register:

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

#Staff register:

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

Login: 2 types of user login (Customer, and Airline Staff).

#Airline Staff login:

```
SELECT * FROM staff WHERE username = %s and userpassword = md5(%s)
```

#Customer login:

```
SELECT * FROM customer WHERE email = %s and customer_password = md5(%s)'
```

Customer use cases:

View My flights:

#View past flights:

```
SELECT flight.airline_name, flight.flight_number, flight.departure_datetime,
flight.departure_airport, flight.arrival_datetime, flight.arrival_airport, purchase.sold_price
FROM customer, purchase, ticket NATURAL JOIN flight WHERE customer.email =
purchase.email AND purchase.ticket_id = ticket.ticket_id AND flight.departure_datetime <=
CURRENT_TIMESTAMP AND customer.email = %s
```

#View future flights: The default should be showing for the future flights.

```
SELECT flight.airline_name, flight.flight_number, flight.departure_datetime,
flight.departure_airport, flight.arrival_datetime, flight.arrival_airport, purchase.sold_price
FROM customer, purchase, 'ticket NATURAL JOIN flight WHERE customer.email =
purchase.email AND purchase.ticket_id = ticket.ticket_id AND flight.departure_datetime >=
CURRENT_TIMESTAMP AND customer.email = %s
```

Search for flights:

#Search for go trip:

```
SELECT * FROM flight WHERE (departure_airport, arrival_airport, date(departure_datetime)) =
(%s,%s,%s) AND departure_datetime>CURRENT_TIMESTAMP
```

#Search for round trip:

```
SELECT * FROM flight WHERE (departure_airport, arrival_airport, date(departure_datetime)) =
(%s,%s,%s) AND departure_datetime>CURRENT_TIMESTAMP
```

Purchase tickets:

#Customer choose the tickets by showing the capacity, occupancy, ticket id andbase price of the ticket

```
SELECT COUNT(ticket_id) AS capacity FROM ticket WHERE
(airline_name,flight_number,departure_datetime) = (%s,%s,%s)
SELECT COUNT(ticket_id) AS occupancy FROM ticket NATURAL JOIN purchase WHERE
(airline_name,flight_number,departure_datetime) = (%s,%s,%s)
 SELECT ticket_id FROM ticket NATURAL LEFT OUTER JOIN purchase WHERE
(airline_name,flight_number,departure_datetime) = (%s,%s,%s) AND email IS null
```

```
SELECT base_price FROM flight WHERE (airline_name,flight_number,departure_datetime) =
(%s,%s,%s)
```

#Customer buy the tickets

```
 INSERT INTO purchase VALUES (%s,%s,%s,%s,%s,%s,%s,CURRENT_TIMESTAMP)
```

Give Ratings and Comment on previous flights:

```
SELECT * FROM customerrate WHERE email = %s
```

#delete the comment

```
DELETE FROM customerrate WHERE (email,airline_name,flight_number,departure_datetime)
= (%s,%s,%s,%s)
```

#add the comment

```
INSERT INTO customerrate VALUES (%s,%s,%s,%s,%s,%s)
```

Track My Spending:

#Track the total money spent for the past year

```
SELECT SUM(sold_price) AS yearSpending, CURRENT_DATE AS enddate,
date(CURRENT_DATE-10000) AS startdate FROM purchase WHERE email=%s AND
purchase_datetime >= date(CURRENT_DATE-10000) GROUP BY email
```

#Track the total money spent for the past 6 months

```
SELECT CURRENT_DATE, year(purchase_datetime) AS year, month(purchase_datetime) AS
month, SUM(sold_price) AS spending FROM purchase WHERE email=%s GROUP BY
year(purchase_datetime), month(purchase_datetime) HAVING
(year-year(CURRENT_DATE))*12+month >= month(CURRENT_DATE)-5
```

#search spending for a particular range:

```
SELECT year(purchase_datetime) AS year, month(purchase_datetime) AS month,
SUM(sold_price) AS spending FROM purchase WHERE date(purchase_datetime) >= %s AND
date(purchase_datetime) <= %s AND email = %s GROUP BY
year(purchase_datetime),month(purchase_datetime)
```

Airline Staff use cases:

View flights:

#Showing all the future flights operated by the airline he/she works for the next 30 days:

```
 SELECT * FROM flight WHERE departure_datetime <= CURRENT_TIMESTAMP + INTERVAL 30
day AND departure_datetime >= CURRENT_TIMESTAMP
```

#Search based on range of dates, source/destination airports/city.

```
SELECT * FROM flight WHERE date(departure_datetime) >= %s AND date(departure_datetime)
<= %s AND (departure_airport,arrival_airport) = (%s,%s)
```

#Search only based on range of dates

```
SELECT * FROM flight WHERE date(departure_datetime) >= %s AND date(departure_datetime)
<= %s
```

#Search only based on source/destination airports/city

```
SELECT * FROM flight WHERE (departure_airport,arrival_airport) = (%s,%s)
```

Create new flights:

#check if departure airport exist

```
SELECT * FROM airport WHERE airport_code = %s
```

#check if arrival airport exist

```
SELECT * FROM airport WHERE airport_code = %s
```

#check if airplane exist

```
SELECT * FROM airplane WHERE (airline_name,airplane_id) = (%s,%s)
```

#delete repeat information

```
DELETE FROM flight WHERE (airline_name,flight_number,departure_datetime) = (%s,%s,%s)
```

#Insert flight

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

#create ticket

```
INSERT INTO ticket VALUES (%s,%s,%s,%s)
```

Change Status of flights:

```
UPDATE flight SET flight_status = %s WHERE
(airline_name,flight_number,departure_datetime) = (%s,%s,%s)
```

Add airplane in the system:

#see all the airplanes owned by the airline he/she works for

```
SELECT * FROM airplane WHERE airline_name = %s
```

#Add airplane

```
INSERT INTO airplane VALUES (%s,%s,%s)
```

Add new airport in the system:

#see all the airports

```
SELECT * FROM airport
```

#add the airport

```
INSERT INTO airport VALUES (%s,%s,%s)
```

View flight ratings:

#See all the comments and ratings of that flight given by the customers

```
SELECT * FROM flight WHERE date(departure_datetime) >= %s AND date(departure_datetime)
<= %s AND (departure_airport,arrival_airport) = (%s,%s)
```

#See average ratings of that flight given by the customers

```
SELECT AVG(rate) AS avg_rating FROM customerrate WHERE
(airline_name,flight_number,departure_datetime) = (%s,%s,%s)
```

View frequent customers:

```
SELECT customer.email AS email, customer_name, COUNT(purchase.ticket_id) AS
ticketsBought, SUM(sold_price) AS totalSpending FROM customer,purchase,ticket WHERE
purchase.email = customer.email AND purchase_datetime >= CURRENT_TIMESTAMP -
INTERVAL 1 year AND purchase.ticket_id=ticket.ticket_id AND ticket.airline_name = %s
GROUP BY customer_name ORDER BY ticketsBought DESC LIMIT 10
```

#Seach for particular customer

```
SELECT DISTINCT airline_name, flight_number, departure_datetime, purchase.ticket_id
FROM purchase, ticket NATURAL JOIN flight WHERE purchase.ticket_id=ticket.ticket_id AND
purchase.email = %s ORDER BY departure_datetime DESC
```

View reports and View Earned Revenue:

#For amounts of ticket and the total revenue in the last month:

```
SELECT COUNT(ticket.ticket_id) AS ticketSold, SUM(purchase.sold_price) AS totalRevenue
FROM ticket,purchase WHERE ticket.ticket_id = purchase.ticket_id AND ticket.airline_name =
%s AND purchase.purchase_datetime >= CURRENT_TIMESTAMP - INTERVAL 1 month
```

#For amounts of ticket and the total revenue in the last year:

```
SELECT COUNT(ticket.ticket_id) AS ticketSold, SUM(purchase.sold_price) AS totalRevenue
FROM ticket,purchase WHERE ticket.ticket_id = purchase.ticket_id AND ticket.airline_name =
%s AND purchase.purchase_datetime >= CURRENT_TIMESTAMP - INTERVAL 1 year
```

#Search for the amounts of ticket and the total revenue in a time range:

```
SELECT year(purchase.purchase_datetime) AS theYear, month(purchase.purchase_datetime)
AS theMonth, COUNT(ticket.ticket_id) AS ticketSold, SUM(purchase.sold_price) AS
totalRevenue FROM ticket,purchase WHERE ticket.ticket_id = purchase.ticket_id AND
ticket.airline_name = %s AND date(purchase.purchase_datetime) >= %s and
date(purchase.purchase_datetime) <= %s GROUP BY year(purchase.purchase_datetime),
month(purchase.purchase_datetime)
```

View Top destinations:

#Find the top 3 most popular destinations for last 3 months:

```
SELECT airport.airport_code, airport.airport_name, airport.airport_city,
COUNT(ticket.ticket_id) AS ticket_num FROM purchase NATURAL JOIN ticket NATURAL JOIN
flight, airport WHERE flight.arrival_airport = airport.airport_code AND flight.airline_name =
%s AND purchase.purchase_datetime >= CURRENT_TIMESTAMP - INTERVAL 3 month GROUP
BY airport.airport_code ORDER BY ticket_num DESC LIMIT 3
```

#Find the top 3 most popular destinations for last year:

```
SELECT airport.airport_code, airport.airport_name, airport.airport_city,
COUNT(ticket.ticket_id) AS ticket_num FROM purchase NATURAL JOIN ticket NATURAL JOIN
flight, airport WHERE flight.arrival_airport = airport.airport_code AND flight.airline_name =
%s AND purchase.purchase_datetime >= CURRENT_TIMESTAMP - INTERVAL 1 yearGROUP BY
airport.airport_code ORDER BY ticket_num DESC LIMIT 3
```