**Queries for login function:**

- SELECT * FROM customer WHERE e_mail = %s and password = %s
  - This query gets all the information related to a customer based on their input (email, password)
- SELECT first_name FROM customer WHERE e_mail = %s and password = %s
  - This query gets the first name of a customer based on their input (email, password)

**Queries for stafflogin function:**

- SELECT * FROM airline_staff WHERE username = %s and password = %s
  - This query gets all the information related to staff based on their input (email, password)
- SELECT first_name FROM airline_staff WHERE username = %s and password = %s
  - This query gets the first name of a staff based on their input (email, password)

**Queries for register function:**

- SELECT * FROM customer WHERE e_mail = %s and first_name = %s and last_name = %s and password = %s and building_num = %s and street_name = %s and apt_num = %s and city = %s and state = %s and zip_code = %s
  - This query gets all the registration information of a customer. It is used to check if a customer already registered
- INSERT INTO customer VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
  - This query adds the customer information into the customer table

**Queries for staffregister function:**

- SELECT* FROM airline_staff WHERE username = %s and airline_name = %s and password = %s and first_name = %s and last_name = %s and date_of_birth = %s
  - This gets all the information of a staff member. It is used to check if a staff member already exists
  - 
- INSERT INTO airline_staff VALUES(%s, %s, %s, %s, %s, %s)
  - This query adds the customer information to the airline_staff table

**Queries for index function:**

- SELECT code FROM airport
  - This query gets all the airport code stored in the database. It is used to output all the departure/arrival airport available
- SELECT status, airline_name, flight_num, departure_date, arrival_date FROM flight WHERE departure_airport = %s and arrival_airport = %s and departure_date > NOW()
  - This query gets the future flights information based on the departure airport, arrival airport and departure date
- SELECT status, airline_name, flight_num, departure_date, arrival_date FROM flight WHERE departure_date > ALL (SELECT departure_date FROM flight WHERE departure_airport = %s and arrival_airport = %s)

- This query gets the corresponding return flight information based on the departure/arrival airport of the first flight.

**Queries for staffhome function:**

- SELECT airline_name FROM airline_staff WHERE username = %s
    - This query gets the airline name from the airline_staff where the staff member works for
    - It is used to check for cases to make sure everything is for that airline
- SELECT distinct flight_num, status, departure_date, departure_time, arrival_date, arrival_time, departure_airport, arrival_airport FROM flight natural join airline_staff WHERE airline_name = %s AND DATEDIFF(departure_date, CURDATE()) BETWEEN 0 AND 30;
    - This query gets the distinct flight number, status, departure date, departure time, arrival date, arrival time, departure airport, arrival airport where the departure date is between 30 days.
    - It is used for the staff members to view flights within the next 30 days.
- SELECT airplane_id FROM airplane WHERE airline_name = %s
    - Gets the airplane ID
    - Used for HTML purposes in to make sure the user selects the limited airplane IDs
- SELECT flight_num FROM flight WHERE airline_name = %s and arrival_date < NOW()
    - Gets the flight number from flights that already happened
    - This is also used for HTML purposes, in order to make sure that the staff member can only see reviews for past flights
- SELECT code from airport
    - Gets the airport codes from the airport table
    - This is use for HTML purposes, to makes sure the user can only see the added airports
- INSERT into flight VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    - Inserts the values received into flight table
    - This is for creating a flight
- UPDATE flight SET status = %s WHERE flight_num = %s
    - Updates the status of the flights
- INSERT into airplane VALUES(%s, %s, %s, %s, %s)
    - Inserts values into airplane table
    - Used to create an airplane
- INSERT into airport VALUES(%s, %s, %s, %s, %s)
    - Inserts values into airport table
    - Used to add an airport

**Queries for flight_rating function:**

- SELECT rate, comments FROM review WHERE flight_num = %s
    - Retrieves the ratings and comments from the review table
    - Used to insert into the table
- SELECT AVG(rate) FROM review WHERE flight_num = %s
    - Gets the average ratings
    - Used for the staff members to always see

**Queries for airplane_list function:**

- SELECT airline_name FROM airline_staff WHERE username = %s
    - This query gets the name of the airline the staff works at
- Select airplane_id, num_seats, Manufacturing_comp, Manufacturing_date from airplane where airline_name = %s
    - The output of the first query (airline name) is used here. This query gets the data of all the airplanes that correspond to that airline name

**Queries for frequent_customer function:**

- SELECT COUNT(ticket_id) as num_ticket, e_mail, first_name, last_name FROM customer natural join purchase natural join ticket natural join flight natural join airline WHERE airline_name = %s GROUP BY e_mail ORDER by COUNT(ticket_id) DESC
    - This query gets the most frequent customers and order them such that the customer who bought the most tickets is at the top of the table and the customer who bought the least tickets is at the bottom.

**Queries for report function:**

- SELECT COUNT(ticket_id) FROM purchase natural join ticket natural join flight WHERE airline_name = %s and purchase_date BETWEEN %s and NOW()"
    - It gets the number of ticket_id between a certain selected date and now
    - It is used to display the number of purchases

**Queries for revenue function:**

- SELECT SUM(ticket_price) FROM purchase natural join ticket natural join flight WHERE airline_name = %s and purchase_date BETWEEN DATE_SUB(NOW(), INTERVAL 30 DAY) AND NOW()
    - This query gets the total amount spent on flight tickets in a specific airline over the past 30 days (month)
- SELECT SUM(ticket_price) FROM purchase natural join ticket natural join flight WHERE airline_name = %s and purchase_date BETWEEN DATE_SUB(NOW(), INTERVAL 365 DAY) AND NOW()
    - This query gets the total amount spent on flight tickets in a specific airline over the past 365 days (year)

**Queries for home function:**

- SELECT flight_num FROM flight natural join ticket natural join purchase natural join customer WHERE customer.e_mail = %s"
    - Gets the flight number

    - Used to make sure a flight number exists in future cases
- SELECT ticket.flight_num, flight.airline_name, flight.departure_time, flight.departure_date, flight.departure_airport, flight.arrival_airport, ticket.ticket_id FROM flight natural join ticket natural join purchase natural join customer WHERE e_mail = %s"
    - Gets appropriate information about a purchased flight by the customer
    - It allows the customer to view his/her purchased flights
- SELECT code FROM airport
    - Gets the airport codes from airport
    - Used for HTML purposes in order from the user to view the limited added airports
- SELECT num_seats FROM airplane natural join flight where flight_num = %s
    - Gets the number of seats in an airplane
    - This is used to compare with the number of tickets in order to increase the price tickets by 25% when over 80% of the number of seats are bought
- SELECT COUNT(ticket_id) FROM ticket WHERE flight_num = %s
    - Gets the number of tickets for a certain flight
    - This is used to compare with the number of seats in order to increase the price tickets by 25% when over 80% of the number of seats are bought
- SELECT status, airline_name, flight_num, departure_date, arrival_date, base_price FROM flight WHERE departure_airport = %s and arrival_airport = %s and departure_date > NOW()"
    - Gets the appropriate information about a future flight
    - This is for the user to view future flights that they can buy
- SELECT num_seats FROM airplane natural join flight WHERE flight_num = %s
    - Gets the number of seats in an airplane (**this is for a return flight**)
    - This is used to compare with the number of tickets in order to increase the price tickets by 25% when over 80% of the number of seats are bought
- SELECT COUNT(ticket_id) FROM ticket WHERE flight_num = %s
    - Gets the number of tickets for a certain flight (**this is for return flights**)
    - This is used to compare with the number of seats in order to increase the price tickets by 25% when over 80% of the number of seats are bought
- SELECT purchase_date FROM purchase WHERE ticket_id = %s
    - Gets the purchased date of a purchased ticket.
    - This is use to make sure that the customer can only delete a flight within 1 day (24 hours)
    - Use to compare with today's date
- DELETE FROM purchase where ticket_id = %s
    - Deletes a purchase ticket from purchase table
- DELETE FROM ticket where ticket_id = %s
    - Deletes a purchased ticket from ticket table
- SELECT arrival_date FROM flight WHERE flight_num = %s

        - Gets the arrival date of a certain flight selected by the user
        - This to make sure the customer cannot rate a flight if he/she has not taken it yet
    - SELECT flight_num FROM review WHERE flight_num = %s and e_mail = %s
        - Gets the flight_num from the review table where the email of the logged in person is in.
        - This is to make sure the customer cannot review more than once
    - INSERT into review VALUES(%s, %s, %s, %s)
        - Inserts information such as e_mail, flight_num, rate, and comment into the review table.
        - It only inserts data if the flight has already occurred. This is to ensure that the customer cannot rate flights that he has not taken

**Queries for spending function:**

- SELECT SUM(ticket_price) FROM customer natural join purchase natural join ticket WHERE e_mail = %s and purchase_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 180 DAY) AND CURDATE()
    - This query outputs to the customers the total amount they spent on tickets over the past 6 months (180 days)
- SELECT SUM(ticket_price) FROM customer natural join purchase natural join ticket WHERE e_mail = %s and purchase_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 365 DAY) AND CURDATE()
    - This query outputs to the customers the total amount they spent on tickets over the past year (365 days)
- SELECT SUM(ticket_price) FROM customer natural join purchase natural join ticket WHERE e_mail = %s and purchase_date BETWEEN %s and %s
    - This query outputs to the customer the total amount they spent on a specific period of time. The user gets to choose between two dates and within those days, the total amount spent will be shown

**Queries for buy_flights function:**

- SELECT base_price FROM flight WHERE flight_num = %s
    - This query gets the base price of a specific flight based on the flight number
    - The output of this query is used to calculate the ticket price
- SELECT num_seats FROM airplane natural join flight where flight_num = %s
    - This query gets the number of all the seats in a flight
    - It is used to compare the total number of seats bought by customers
- SELECT COUNT(ticket_id) FROM ticket WHERE flight_num = %
    - This query gets the total number of seats bought by customers
- INSERT into ticket VALUES (%s, %s, %s, %s, %s, %s, %s)
    - This query inserts ticket information (ticket_id, flight_num, ticket_price, card_type, name_on_card, expiration_date, card_num) in the database only if the total number of tickets bought is less than the number of seats available.
- INSERT into purchase VALUES(%s, %s, %s, %s)
    - This query inserts purchase information (e_mail, ticket_id, current_time, current_date) in the purchase table only if the total number of tickets bought is less than the number of seats available.