1. Public search based on source city/airport name, destination city/airport name, date

```
select * from flight\
            where (departure_airport = %s \
                   or departure_airport in (select airport_name from airport where
airport_city = %s))\
                   and (arrival_airport = %s \
                   or arrival_airport in (select airport_name from airport where
airport_city = %s))\
                   and convert(departure_time,date) = %s\
                   and status = "upcoming"\
                   and (departure_time >= curtime() or arrival_time >= curtime())'
```

see the flights status based on flight number, arrival/departure date

```
select airline_name,flight_num, status, departure_time, arrival_time from flight\
            where flight_num=%s\
            and (convert(departure_time,date)=%s or convert(arrival_time,date)=%s)\
            and (departure_time >= curtime() or arrival_time >= curtime())'
```

2. Register: check whether user has already registered

```
query = "SELECT * FROM customer WHERE email = \'{}\'"
   cursor.execute(query.format(name))
```

Register: register into the system

```
ins = "INSERT INTO customer VALUES(\'{}\', \'{}\', md5(\'{}\'), \'{}\', \'{}\',
\'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\', \'{}\')"
       cursor.execute(ins.format(email, name, password, building_number, street, city,
state, phone_number, passport_number, passport_expiration, passport_country,
date_of_birth))
```

3. Login

```
if usrtype == 'customer':
       query = 'SELECT * FROM customer WHERE email = \'{}\' and password =
md5(\'{}\')'
   elif usrtype == 'agent':
       query = 'SELECT * FROM booking_agent WHERE email = \'{}\' and password =
md5(\'{}\')'
   else:
       query = 'SELECT * FROM airline_staff WHERE username = \'{}\' and password =
md5(\'{}\')'

   cursor.execute(query.format(username, password))
```

Customer

4. View my flights: see flights information which he/she purchased

```
query = 'SELECT purchases.ticket_id, ticket.airline_name, ticket.flight_num,
departure_airport, departure_time, arrival_airport, arrival_time \
               FROM purchases, ticket, flight \
               WHERE purchases.ticket_id = ticket.ticket_id \
               AND ticket.airline_name = flight.airline_name \
               AND ticket.flight_num = flight.flight_num \
               AND customer_email = \'{}\' AND departure_time > curdate()'
   cursor.execute(query.format(username))
```

specify a range of dates, specify destination and/or source airport name or city name

```
# Get flight information in the given period
   query = 'SELECT purchases.ticket_id, ticket.airline_name, ticket.flight_num,
departure_airport, departure_time, arrival_airport, arrival_time \
               FROM purchases, ticket, flight, airport \
               WHERE purchases.ticket_id = ticket.ticket_id \
               AND ticket.airline_name = flight.airline_name \
               AND ticket.flight_num = flight.flight_num \
               AND flight.departure_airport = airport.airport_name \
               AND purchases.customer_email = %s \
               AND flight.departure_time BETWEEN CAST(%s AS DATE) AND CAST(%s AS DATE)
\
               AND airport.airport_city = %s AND airport.airport_name = %s \
               AND (flight.airline_name, flight.flight_num) in \
                   (SELECT flight.airline_name, flight.flight_num FROM flight, airport
\
                   WHERE airport.airport_name=flight.arrival_airport \
                   AND airport.airport_city = %s \
                   AND airport.airport_name = %s)'

   cursor.execute(query, (username, fromdate, todate, fromcity, fromairport, tocity,
toairport))
```

5. Purchase tickets

```
query = 'SELECT * FROM flight, airport \
           WHERE airport.airport_name = flight.departure_airport \
           AND airport.airport_city = %s \
           AND airport.airport_name = %s \
           AND flight.status = "Upcoming"\
           AND %s BETWEEN DATE_SUB(flight.departure_time, INTERVAL 2 DAY) AND 
DATE_ADD(flight.departure_time, INTERVAL 2 DAY) \
           AND %s BETWEEN DATE_SUB(flight.arrival_time, INTERVAL 2 DAY) AND 
DATE_ADD(flight.arrival_time, INTERVAL 2 DAY) \
           AND (flight.airline_name, flight.flight_num) in \
               (SELECT flight.airline_name, flight.flight_num FROM flight, airport \
               WHERE airport.airport_name=flight.arrival_airport \
               AND airport.airport_city = %s \
               AND airport.airport_name = %s)'
   cursor.execute(query, (fromcity, fromairport, fromdate, todate, tocity, toairport))
```

```
# generate ticket id
   queryCount = 'SELECT COUNT(*) as count FROM ticket'
   cursor.execute(queryCount)
   ticketCount = cursor.fetchone()
   ticket_id = ticketCount[0] + 1
   # Create the new ticket
   queryNewTicket = 'INSERT INTO ticket VALUES(%s, %s, %s)'
   cursor.execute(queryNewTicket, (ticket_id, airline_name, flight_num))
   # Finalize the purchase
   queryPurchase = 'INSERT INTO purchases VALUES(%s, %s, %s, CURDATE())'
```

6. Search for flights:

```
query = 'SELECT * FROM flight, airport \
           WHERE airport.airport_name = flight.departure_airport \
           AND airport.airport_city = %s \
           AND airport.airport_name = %s \
           AND flight.status = "Upcoming"\
           AND %s BETWEEN DATE_SUB(flight.departure_time, INTERVAL 2 DAY) AND 
DATE_ADD(flight.departure_time, INTERVAL 2 DAY) \
           AND %s BETWEEN DATE_SUB(flight.arrival_time, INTERVAL 2 DAY) AND 
DATE_ADD(flight.arrival_time, INTERVAL 2 DAY) \
           AND (flight.airline_name, flight.flight_num) in \
               (SELECT flight.airline_name, flight.flight_num FROM flight, airport \
               WHERE airport.airport_name=flight.arrival_airport \
               AND airport.airport_city = %s \
               AND airport.airport_name = %s)'
```

7.Track My Spending

Total spending

```
query = 'SELECT sum(price) as total \
       FROM purchases, ticket, flight \
       WHERE purchases.ticket_id = ticket.ticket_id \
       AND ticket.airline_name = flight.airline_name AND ticket.flight_num =
flight.flight_num \
       AND purchases.purchase_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 1 YEAR) AND
CURDATE() \
       AND purchases.customer_email = %s'
   cursor.execute(query, (username))
```

Six months bar chart data

```
for i in range(6):
       query = 'SELECT sum(price) as monthlySpending \
           FROM purchases, ticket, flight \
           WHERE purchases.ticket_id = ticket.ticket_id \
           AND ticket.airline_name = flight.airline_name AND ticket.flight_num =
flight.flight_num \
           AND year(purchases.purchase_date) = year(CURDATE() - interval ' + str(i) +
' month)\
           AND month(purchases.purchase_date) = month(CURDATE() - interval ' + str(i)
+ ' month) \
           AND purchases.customer_email = %s'
       cursor.execute(query, (username))
```

specify a range of dates to view total amount of money spent within that range

```
queryGetTotal = 'SELECT sum(price) as total \
       FROM purchases, ticket, flight \
       WHERE purchases.ticket_id = ticket.ticket_id \
       AND ticket.airline_name = flight.airline_name AND ticket.flight_num =
flight.flight_num \
       AND purchases.purchase_date BETWEEN CAST(%s AS DATE) AND CAST(%s AS DATE) \
       AND purchases.customer_email = %s'
   cursor.execute(queryGetTotal, (fromdate, todate, username))
```

a bar chart showing month wise money spent within that range

```
for i in range(month_difference+1):
       query = 'SELECT sum(price) as monthlySpending \
           FROM purchases, ticket, flight \
           WHERE purchases.ticket_id = ticket.ticket_id \
           AND ticket.airline_name = flight.airline_name AND ticket.flight_num =
flight.flight_num \
           AND year(purchases.purchase_date) = year(%s - interval ' + str(i) + '
month)\
           AND month(purchases.purchase_date) = month(%s - interval ' + str(i) + '
month) \
           AND purchases.customer_email = %s'
       cursor.execute(query, (todate, todate, username))
```

Agent

4. View my flights: see flights information which he/she purchased

```
query = 'SELECT purchases.customer_email, purchases.ticket_id, ticket.airline_name,
ticket.flight_num, departure_airport, departure_time, arrival_airport, arrival_time \
               FROM purchases, ticket, flight, booking_agent \
               WHERE purchases.ticket_id = ticket.ticket_id \
               AND ticket.airline_name = flight.airline_name \
               AND ticket.flight_num = flight.flight_num \
               AND booking_agent.booking_agent_id = purchases.booking_agent_id \
               AND booking_agent.email = \'{}\'\
               AND departure_time > curdate() \
               ORDER BY customer_email'
   cursor.execute(query.format(username))
```

specify a range of dates, specify destination and/or source airport name or city name

```
query = 'SELECT purchases.customer_email, purchases.ticket_id, ticket.airline_name,
ticket.flight_num, departure_airport, departure_time, arrival_airport, arrival_time \
               FROM purchases, ticket, flight, airport, booking_agent \
               WHERE purchases.ticket_id = ticket.ticket_id \
               AND ticket.airline_name = flight.airline_name \
               AND ticket.flight_num = flight.flight_num \
               AND flight.departure_airport = airport.airport_name \
               AND booking_agent.booking_agent_id = purchases.booking_agent_id \
               AND booking_agent.email = %s\
               AND flight.departure_time BETWEEN CAST(%s AS DATE) AND CAST(%s AS DATE)
\
               AND airport.airport_city = %s AND airport.airport_name = %s \
               AND (flight.airline_name, flight.flight_num) in \
                   (SELECT flight.airline_name, flight.flight_num FROM flight, airport
\
                   WHERE airport.airport_name=flight.arrival_airport \
                   AND airport.airport_city = %s \
                   AND airport.airport_name = %s)'

   cursor.execute(query, (username, fromdate, todate, fromcity, fromairport, tocity,
toairport))
```

5. Purchase tickets

```
query = 'SELECT * FROM flight, airport \
           WHERE airport.airport_name = flight.departure_airport \
           AND airport.airport_city = %s \
           AND airport.airport_name = %s \
           AND flight.status = "Upcoming"\
           AND %s BETWEEN DATE_SUB(flight.departure_time, INTERVAL 2 DAY) AND
DATE_ADD(flight.departure_time, INTERVAL 2 DAY) \
           AND %s BETWEEN DATE_SUB(flight.arrival_time, INTERVAL 2 DAY) AND
DATE_ADD(flight.arrival_time, INTERVAL 2 DAY) \
           AND (flight.airline_name, flight.flight_num) in \
               (SELECT flight.airline_name, flight.flight_num FROM flight, airport \
               WHERE airport.airport_name=flight.arrival_airport \
               AND airport.airport_city = %s \
               AND airport.airport_name = %s)'
   cursor.execute(query, (fromcity, fromairport, fromdate, todate, tocity, toairport))
```

```
# Find the number of tickets to generate the next ticket_id
   queryCount = 'SELECT COUNT(*) as count FROM ticket'
   cursor.execute(queryCount)
   ticketCount = cursor.fetchone()
   ticket_id = ticketCount[0] + 1
   # Create the new ticket
   queryNewTicket = 'INSERT INTO ticket VALUES(%s, %s, %s)'
   cursor.execute(queryNewTicket, (ticket_id, airline_name, flight_num))
   # Get booking_agent_id
   queryGetID = 'SELECT booking_agent_id FROM booking_agent WHERE email=%s'
   cursor.execute(queryGetID, username)
   agentID = cursor.fetchone() # returns a dict
   # Finalize the purchase
   queryPurchase = 'INSERT INTO purchases VALUES(%s, %s, %s, CURDATE())'
   cursor.execute(queryPurchase, (ticket_id, customer_email, agentID[0]))
```

6. Search for flights

```
query = 'SELECT * FROM flight, airport \
           WHERE airport.airport_name = flight.departure_airport \
           AND airport.airport_city = %s \
           AND airport.airport_name = %s \
           AND flight.status = "Upcoming"\
           AND %s BETWEEN DATE_SUB(flight.departure_time, INTERVAL 2 DAY) AND
DATE_ADD(flight.departure_time, INTERVAL 2 DAY) \
           AND %s BETWEEN DATE_SUB(flight.arrival_time, INTERVAL 2 DAY) AND
DATE_ADD(flight.arrival_time, INTERVAL 2 DAY) \
           AND (flight.airline_name, flight.flight_num) in \
               (SELECT flight.airline_name, flight.flight_num FROM flight, airport \
               WHERE airport.airport_name=flight.arrival_airport \
               AND airport.airport_city = %s \
               AND airport.airport_name = %s)'
   cursor.execute(query, (fromcity, fromairport, fromdate, todate, tocity, toairport))
```

7. View my commission

```
# Get booking_agent_id
   queryGetID = 'SELECT booking_agent_id FROM booking_agent WHERE email=%s'
   cursor.execute(queryGetID, username)
   agentID = cursor.fetchone()
   # Get total commsion in the past 30 days
   queryGetCommission = 'SELECT sum(price)*.10 as totalComm FROM purchases, ticket,
flight \
                           WHERE purchases.ticket_id = ticket.ticket_id \
                           AND ticket.airline_name = flight.airline_name AND
ticket.flight_num = flight.flight_num \
                           AND purchases.purchase_date BETWEEN DATE_SUB(CURDATE(),
INTERVAL 30 DAY) AND CURDATE() \
                           AND purchases.booking_agent_id = %s'
   cursor.execute(queryGetCommission, agentID[0])
   totalComm = cursor.fetchone()
   totalCommVal = 0
   if totalComm[0] != None:
       totalCommVal = totalComm[0]
   # Get total tickets in the past 30 days
   queryGetTicketCount = 'SELECT count(*) as ticketCount FROM purchases, ticket,
flight \
                           WHERE purchases.ticket_id = ticket.ticket_id \
                           AND ticket.airline_name = flight.airline_name AND
ticket.flight_num = flight.flight_num \
                           AND purchases.purchase_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY) AND CURDATE() \
                           AND purchases.booking_agent_id = %s'
   cursor.execute(queryGetTicketCount, agentID[0])
```

specify a range of dates

```
queryGetID = 'SELECT booking_agent_id FROM booking_agent WHERE email=%s'
   cursor.execute(queryGetID, username)
   agentID = cursor.fetchone()
   # Get total commsion in the given period
   queryGetCommission = 'SELECT sum(price)*.10 as totalComm FROM purchases, ticket, flight \
                           WHERE purchases.ticket_id = ticket.ticket_id \
                           AND ticket.airline_name = flight.airline_name AND ticket.flight_num = flight.flight_num \
                           AND purchases.purchase_date BETWEEN CAST(%s AS DATE) AND CAST(%s AS DATE) \
                           AND purchases.booking_agent_id = %s'
   cursor.execute(queryGetCommission, (fromdate, todate, agentID[0]))
   totalComm = cursor.fetchone()
   totalCommVal = 0
   if totalComm[0] != None:
       totalCommVal = totalComm[0]
   # Get total tickets in the given period
   queryGetTicketCount = 'SELECT count(*) as ticketCount FROM purchases, ticket, flight \
                           WHERE purchases.ticket_id = ticket.ticket_id \
                           AND ticket.airline_name = flight.airline_name AND ticket.flight_num = flight.flight_num \
                           AND purchases.purchase_date BETWEEN CAST(%s AS DATE) AND CAST(%s AS DATE) \
                           AND purchases.booking_agent_id = %s'
   cursor.execute(queryGetTicketCount, (fromdate, todate, agentID[0]))
```

8. View Top Customers

Top 5 customers based on number of tickets bought

```
query1 = 'SELECT customer_email, count(ticket_id) as ticket_sales\
               FROM booking_agent NATURAL JOIN purchases\
               WHERE (purchase_date between date_sub(curdate(), interval 6 month) and curdate())\
                   and email = %s\
               group by customer_email\
               order by ticket_sales DESC\
               limit 5'
   cursor.execute(query1, (username))
   data1 = cursor.fetchall()
   ticketTop5 = ''
   for i in data1:
       ticketTop5 += str(i[1]) + " " + str(i[0]) + ","
```

Top 5 customers based on the amount of commission received in the last year

```
   query2 = 'SELECT customer_email, sum(flight.price) as commission\
               FROM booking_agent, purchases, ticket, flight\
               WHERE (purchase_date between date_sub(curdate(), interval 1 year) and
curdate())\
                   AND booking_agent.booking_agent_id = purchases.booking_agent_id \
                   AND purchases.ticket_id = ticket.ticket_id \
                   AND ticket.airline_name = flight.airline_name \
                   AND ticket.flight_num = flight.flight_num \
                   AND booking_agent.email = %s\
               group by purchases.customer_email\
               order by commission DESC\
               limit 5'
   cursor.execute(query2, (username))
```

Airline Staff use cases:

4. View My flights

```
query = 'select * from flight\
            where (departure_airport = %s \
                   or departure_airport in (select airport_name from airport where
airport_city = %s))\
                   and (arrival_airport = %s \
                   or arrival_airport in (select airport_name from airport where
airport_city = %s))\
                   and convert(departure_time,date) = %s\
                   and status = "upcoming"\
                   and (departure_time >= curtime() or arrival_time >= curtime())'
   cursor.execute(query,
(searchtext1,searchtext1,searchtext2,searchtext2,searchtext3))
```

status:

```
query = 'select airline_name,flight_num, status, departure_time, arrival_time from
flight\
            where flight_num=%s\
            and (convert(departure_time,date)=%s or convert(arrival_time,date)=%s)\
            and (departure_time >= curtime() or arrival_time >= curtime())'
   cursor.execute(query, (flightnumber,doradate,doradate))
```

Search flights:

```
query = 'select * from flight where airline_name = %s\
                and status = "upcoming"\
                and ((departure_time between curdate() and date_add(curdate(),
interval 30 day)) or (arrival_time between curdate() and date_add(curdate(), interval
30 day)))'
                #and departure_time between "2020-12-12" and "2021-01-12"'
       cursor.execute(query, (airline))
```

Search flights with specific criteria:

```
if not validateDates(fromdate, todate):
           error = 'Invalid date range'
           return redirect(url_for('staffHome', error=error))

       if dcity == 'None' and acity == 'None':
           query = 'select * from flight where airline_name = %s\
                   and ((convert(departure_time,date) between %s and %s)\
                       or (convert(arrival_time,date) between %s and %s))'
           cursor.execute(query, (airline,fromdate,todate,fromdate,todate))
           data = cursor.fetchall()
           cursor.close()
       elif dcity == 'None':
           query = 'select * from flight where airline_name = %s\
                   and (arrival_airport = %s \
                       or arrival_airport in (select airport_name from airport where
airport_city = %s))\
                   and ((convert(departure_time,date) between %s and %s)\
                       or (convert(arrival_time,date) between %s and %s))'
           cursor.execute(query,
(airline,acity,acity,fromdate,todate,fromdate,todate))
           data = cursor.fetchall()
           cursor.close()
       elif acity == 'None':
           query = 'select * from flight where airline_name = %s\
                   and (departure_airport = %s \
                       or departure_airport in (select airport_name from airport where
airport_city = %s))\
                   and ((convert(departure_time,date) between %s and %s)\
                       or (convert(arrival_time,date) between %s and %s))'
           cursor.execute(query,
(airline,dcity,dcity,fromdate,todate,fromdate,todate))
           data = cursor.fetchall()
           cursor.close()
       else:
           query = 'select * from flight where airline_name = %s\
                   and (departure_airport = %s \
                       or departure_airport in (select airport_name from airport where
airport_city = %s))\
                   and (arrival_airport = %s \
                       or arrival_airport in (select airport_name from airport where
airport_city = %s))\
                   and ((convert(departure_time,date) between %s and %s)\
                       or (convert(arrival_time,date) between %s and %s))'
           cursor.execute(query,
(airline,dcity,dcity,acity,acity,fromdate,todate,fromdate,todate))
```

Search customers

```
query = 'select customer_email from purchases natural join ticket where flight_num =
%s and airline_name=%s'
       cursor.execute(query, (flightnum, airline))
```

Search flights

```
query = 'select * from airplane where airplane_id = %s and airline_name = %s'
       cursor.execute(query, (airplaneid, airline))
query2 = 'select * from airport where airport_name=%s'
       cursor.execute(query2, (departport))
query3 = 'select * from airport where airport_name=%s'
       cursor.execute(query3, (arriveport))
query = 'insert into flight values (%s, %s, %s, %s, %s, %s, %s, %s, %s)'
       cursor.execute(query, (airline, flightnum, departport, departtime, arriveport,
arrivetime, price, status, airplaneid))
```

Update status

```
query = 'select * from flight where flight_num = %s and airline_name = %s'
       cursor.execute(query, (flightnum, airline))
query1 = 'update flight set status=%s where flight_num=%s and airline_name = %s'
```

```
        cursor.execute(query1, (status, flightnum, airline))
```

7. Add airplane in the system:

```
query = 'select * from airplane where airplane_id = %s'
        cursor.execute(query, (airplaneid))
query1 = 'insert into airplane values (%s, %s, %s)'
        cursor.execute(query1, (airline, airplaneid, seats))
        conn.commit()

        query2 = 'select * from airplane where airline_name = %s'
        cursor.execute(query2, (airline))
```

8. Add new airport in the system:

```
query = 'select * from airport where airport_name = %s'
query1 = 'insert into airport values (%s, %s)'
        cursor.execute(query1, (airport, city))
```

9. View all the booking agents:

```
query1 = 'select email, count(ticket_id) as ticket_sales\
                 from booking_agent natural join purchases natural join ticket\
                 where (purchase_date between date_sub(curdate(), interval 1 month)
and curdate())\
                        and airline_name = %s\
                 group by email\
                 order by ticket_sales DESC\
                 limit 5'
        cursor.execute(query1, (airline))
```

```
query2 = 'select email, count(ticket_id) as ticket_sales\
                 from booking_agent natural join purchases natural join ticket\
                 where (purchase_date between date_sub(curdate(), interval 1 year) and
curdate())\
                        and airline_name = %s\
                 group by email\
                 order by ticket_sales DESC\
                 limit 5'
        cursor.execute(query2, (airline))
```

```
query3 = 'select email, sum(price)*0.1 as totalcommission\
                 from booking_agent natural join purchases natural join ticket natural
join flight\
                 where (purchase_date between date_sub(curdate(), interval 1 year) and
curdate())\
                       and airline_name = %s\
                 group by email\
                 order by totalcommission DESC\
                 limit 5'
       cursor.execute(query3, (airline))
```

10. View frequent customers:

```
query = 'select customer.name, purchases.customer_email, count(ticket.ticket_id) as
ticket_purchased\
                from (purchases natural join ticket), customer\
                where customer.email = purchases.customer_email\
                      and ticket.airline_name = %s\
                      and (purchases.purchase_date between date_sub(curdate(),
interval 1 year) and curdate())\
                group by purchases.customer_email\
                order by ticket_purchased DESC\
                limit 1'

       cursor.execute(query, (airline))
```

particular:

```
query = 'select distinct flight_num from purchases natural join ticket where
airline_name = %s and customer_email=%s'
       cursor.execute(query, (airline, customer))
```

11. View reports:

```
query = 'select year,month,count(ticket_id)\
               from (select year(purchase_date) as year, month(purchase_date) as
month, ticket_id\
               from purchases natural join ticket\
               where (purchase_date between date_sub(curdate(), interval 1 year) and
curdate()) and airline_name = %s) as a\
               group by year, month'
       cursor.execute(query, (airline))
```

Specific range:

```
query = 'select count(ticket_id) as sales from purchases natural join ticket where
airline_name=%s and purchase_date between %s and %s'
       cursor.execute(query, (airline, begindate, enddate))
query = 'select year,month,count(ticket_id)\
               from (select year(purchase_date) as year, month(purchase_date) as
month, ticket_id\
               from purchases natural join ticket\
               where (purchase_date between %s and %s) and airline_name = %s) as a\
               group by year, month'
       cursor.execute(query, (begindate, enddate,airline))
```

Specific period:

```
query = 'select count(ticket_id) as sales from purchases natural join ticket where
airline_name=%s and (purchase_date between date_sub(curdate(), interval 1 ' + period +
') and curdate())'
       cursor.execute(query, (airline))
query = 'select year,month,count(ticket_id)\
               from (select year(purchase_date) as year, month(purchase_date) as
month, ticket_id\
               from purchases natural join ticket\
               where (purchase_date between date_sub(curdate(), interval 1 ' + period
+ ') and curdate()) and airline_name = %s) as a\
               group by year, month'
       cursor.execute(query, (airline))
```

12. Comparison of Revenue earned:

```
query1 = 'select sum(price)\
                 from flight natural join purchases natural join ticket\
                 where airline_name = %s and (purchase_date between
date_sub(curdate(), interval 1 month) and curdate())\
                 and booking_agent_id is null'
       cursor.execute(query1, (airline))
       data1 = cursor.fetchall()
       query2 = 'select sum(price)\
                 from flight natural join purchases natural join ticket\
                 where airline_name = %s and (purchase_date between
date_sub(curdate(), interval 1 month) and curdate())\
                 and booking_agent_id is not null'
       cursor.execute(query2, (airline))
       data2 = cursor.fetchall()
       query3 = 'select sum(price)\
                 from flight natural join purchases natural join ticket\
                 where airline_name = %s and (purchase_date between
date_sub(curdate(), interval 1 year) and curdate())\
                 and booking_agent_id is null'
       cursor.execute(query3, (airline))
       data3 = cursor.fetchall()
       query4 = 'select sum(price)\
                 from flight natural join purchases natural join ticket\
                 where airline_name = %s and (purchase_date between
date_sub(curdate(), interval 1 year) and curdate())\
                 and booking_agent_id is not null'
       cursor.execute(query4, (airline))
```

13. View Top destinations:

```
query1 = 'select flight.arrival_airport, airport.airport_city, count(*) as
total_purchase\
                from (flight natural join purchases natural join ticket),airport \
                where flight.arrival_airport = airport.airport_name and
ticket.airline_name = %s and purchases.purchase_date between date_sub(curdate(),
interval 3 month) and curdate()\
                group by flight.arrival_airport, airport.airport_city\
                order by count(*) DESC\
                limit 3'
       cursor.execute(query1, (airline))
       data1 = cursor.fetchall()
       query2 = 'select flight.arrival_airport, airport.airport_city, count(*) as
total_purchase\
                from (flight natural join purchases natural join ticket),airport \
                where flight.arrival_airport = airport.airport_name and
ticket.airline_name = %s and purchases.purchase_date between date_sub(curdate(),
interval 1 year) and curdate()\
                group by flight.arrival_airport, airport.airport_city\
                order by count(*) DESC\
                limit 3'
       cursor.execute(query2, (airline))
```