Booking Agents

|  |  |
| --- | --- |
| Use Case | Implementation |
| View My Flights | Get all the ticket ids where the booking agent help buy:  "Select ticket\_id from purchases where booking\_agent\_id = " + bookingID  Obtain the flight\_numbers from the ticket\_ids with a for loop:  query = "Select flight\_number from ticket where ticket\_id = "  for item in ticket\_ids:  query += str(item.get('ticket\_id'))  query += " or ticket\_id = "  query += " -1 "  Select all the details from the flight\_numbers that are in the future with for loop:  query = "Select \* from flight where (CURRENT\_DATE < flight.departure\_date OR (CURRENT\_DATE = flight.departure\_date AND CURRENT\_TIME < departure\_time)) and (flight\_number = "  for item in flight\_numbers:  query += str(item.get('flight\_number'))  query += " or flight\_number = "  query += " -1) " |
| Search for Flights | Same as view public as not logged in users |
| Purchase Flights |  |
| View my Commission | This gets the sum of commissions for this booking agent for the last 30 days:  query = "SELECT sum(`sold\_price`)/10 from purchases where (purchase\_date > ADDDATE(CURRENT\_DATE, INTERVAL -30 DAY)) and booking\_agent\_id =" + bookingID  cursor.execute(query)  commission = cursor.fetchall()[0].get("sum(`sold\_price`)/10")  This gets the amount of tickets for this booking agent for the last 30 days:  query = "SELECT COUNT(\*) from purchases WHERE (purchase\_date > ADDDATE(CURRENT\_DATE, INTERVAL -30 DAY)) and `booking\_agent\_id` =" + bookingID  cursor.execute(query)  tickets = cursor.fetchall()[0].get("COUNT(\*)")  If they search for a date range we modify both sum of commission and amount of tickets to accommodate that date range:  query = "SELECT sum(`sold\_price`)/10 from purchases where ((purchase\_date > \'" + request.form['begDate'] + "\') and ("purchase\_date < \'" + request.form['endDate']  query += "\')) and booking\_agent\_id =" + bookingID  query = "SELECT COUNT(\*) from purchases WHERE ((purchase\_date > \'" + request.form['begDate'] + "\') and ("purchase\_date < \'" + request.form['endDate'] query += "\')) and booking\_agent\_id =" + bookingID |
| View Top Customers | We query by last 6 months and booking agent id then group it by customer\_email then put it in order of the amount of purchases/tickets to get the top 5:  query = "SELECT `customer\_email`, count(\*) FROM purchases WHERE purchase\_date > ADDDATE(CURRENT\_DATE, INTERVAL -6 MONTH) and booking\_agent\_id = " + bookingID + " GROUP BY `customer\_email` ORDER BY COUNT(\*) DESC LIMIT 5 "  We query by last year and booking agent id then group it by customer\_email then put it in order of the amount of commissions to get the top 5:  query = "SELECT `customer\_email`, sum(sold\_price)/10 FROM purchases WHERE purchase\_date > ADDDATE(CURRENT\_DATE, INTERVAL -1 YEAR) and booking\_agent\_id = " + bookingID + " GROUP BY `customer\_email` ORDER BY sum(sold\_price)/10 DESC LIMIT 5" |
| Logout | Ends session |