

## Project Report (Team-15)

1. Missing Parts:
  - We were not able to implement frequentFlyer trigger. We were close to finishing it but there were some errors, so we give it as commented sql code in team15.sql
  - All other tasks and triggers have been implemented in the given java file and sql files.
2. Improvements:
  - More data can be populated to test top-k queries which need lot of data for proper testing of all the edge cases.
  - Some customer tasks had lot of edge cases but to test for them also test data we generated was not sufficient.
  - For submission we just used command line to show the output, but this project can be combined with the web development technology to get better interface.
  - In addition to high price and low price for every flight we can make more classifications like business class, economy class etc to take our project close to actual flight booking system.
  - The case of flight getting cancelled was not considered. For this we can have additional trigger or procedure which on cancellation of flight notify all the customers who reserved seat.
3. Important points or assumptions:
  - Whenever we must check if flight is full or not, we have also checked if plane type can be updated with the trigger written in the homework file. Due to this even if plane of current type is full but plane of more capacity is available system allows the reservation into that flight.
4. Working of driver code:
  - When you run the driver code you will see the simple menu which allows you to choose whether you are customer or are admin.
  - After populating database from text files provided, we insert customers and reservations manually for testing purpose.
  - For all the entries in the reservation table cost is computed by adding all the legs in the reservation.
  - Below we have mentioned some test cases for customer queries which will show you how we have defined customer tasks:
    1. For task 1 you can add user it will check first if customer with same name exists or not.
    2. Task 2 will again print the lists of customers in the system.
    3. For task-4 for direct routes you can give 'PIT' as departure city and 'JFK' as an arrival city and see the results. You can try many combinations of cities. For indirect routes if you give 'PIT' as departure city and 'LAX' as arrival city there will not be any results as difference between these 2 flights is not  $\geq 1$  hr. But if you give 'PIT' and 'BOS' it will show you one connected flight.

4. For task-5 now if you give 'PIT','BOS' as an input then you will not get any result as these flights belong to different airlines. But if you give 'IAH' and 'NYC' as an input it will show you results.
5. For task 6, If you give 'PIT' and 'POR' as input no results will be shown connecting flight going to POR is full. Instead if you try 'PIT' and 'ATL' you will get result as both connecting flights are not full.
6. For task 7 inputs from customer are customer\_id, flight\_number, current\_date, current\_time and departure\_date. After checking if flight is not full and available on given day inserts are made into reservation table and reservation\_detail table.

Test case for task -7:

- i. Customer\_id: 7, departure\_date: 2020-11-24 flight\_num:15, current date and time can be anything you wish.
- ii. This will make flight 15 full but upgrade of plane is possible so one more reservation will be allowed.
- iii. Customer\_id: 8, departure\_date: 2020-11-24 flight\_num:15, current date and time can be anything you wish.
- iv. After this if you try to reserve into same flight with different customer id it will throw error as flight is full.
7. For task 8,9 and 10 you can try any reservation number and see the output.
8. For task 11 and 12 you can give k as as input and see the difference. If you give k 1 or 2 you can see the difference in the output. (all these top k queries are just based on confirmed tickets i.e. reservations for which ticketed is true)
9. Task 13 will directly show the airlines as per rank.
10. Our inserts to customer and reservation table work properly but for easy testing we are running sql script which makes insert into customer and reservation tables.
11. If you run top k queries and task 9 after running other tasks it will be better.