Peer-Graded Assignment: Analyzing Big Data with SQL
Name: Theodor-Mihai Iliant
Date: January 6th 2021

Assignment

Recommend which pair of United States airports should be connected with a high-speed passenger rail tunnel. To do this, write and run a SELECT statement to return pairs of airports that are between **300** and **400** miles apart and that had at least **5,000** (five thousand) flights per year on average *in each direction* between them. Arrange the rows to identify which one of these pairs of airports has largest total number of seats on the planes that flew between them. Your SELECT statement must return all the information required to fill in the table below.

Recommendation

I recommend the following tunnel route:

	First Direction	Second Direction
Three-letter airport code for origin	SFO	LAX
Three-letter airport code for destination	LAX	SFO
Average flight distance in miles	337	337
Average number of flights per year	14712	14540
Average annual passenger capacity	1996597	1981059
Average arrival delay in minutes	10	14

Method

I identified this route by running the following SELECT statement using Impala on the VM:

```
select origin, dest, round(avg(distance)) as avg_dist, round(count(*) / 10) as avg_flights_yearly,
    round(sum(seats) / 10) as avg_capacity_yearly, round(avg(arr_delay)) as avg_delay
from fly.flights left outer join fly.planes
    on flights.tailnum = planes.tailnum
where distance between 300 and 400
group by origin, dest
having avg_flights_yearly > 5000
order by avg_capacity_yearly desc;
```

The following is the text of my query, so that it can be copied:

- - - please check photo attached on previous page for ease of reading the query

```
select origin, dest, round(avg(distance)) as avg_dist, round(count(*) / 10) as
avg_flights_yearly,
        round(sum(seats) / 10) as avg_capacity_yearly, round(avg(arr_delay)) as
avg_delay
    from fly.flights left outer join fly.planes
        on flights.tailnum = planes.tailnum
    where distance between 300 and 400
    group by origin, dest
    having avg_flights_yearly > 5000
    order by avg_capacity_yearly desc;
```

Notes

The question asks for ordering by total number of seats, but since average is calculated by using the common denominator 10, this is equivalent to ordering by the average annual capacity.
We want a left outer join because we must calculate the total number of flights for each pair, irrespective of whether we can find the flight's tail number in the other table. It's also true that the total number of seats may be higher in reality. What if the holy grail is hidden in the sense that we cannot exclude the possibility of the second ranked pair is the one that has NULLs in the tailnum field for all their flights? Just a hypothetical question, but it's not impossible, we just do not know. It would be such a pity though. If there are several planes that operate on the same pair of airports, then the missing value problem is not such a big issue, but if a pair of airports has a contract or something that stipulates that only one or two planes, say, operate on them and those few planes have a NULL in tailnum, we will omit all of those. We really want to be careful in reality about all these things. The assumptions we make about the missing values are very important. We want to be the best data-detectives!

These are the first 10 (out of 22) rows returned by my query:

	origin	dest	avg_dist	avg_flights_yearly	avg_capacity_yearly	avg_delay
1	SFO	LAX	337	14712	1996597	10
2	LAX	SFO	337	14540	1981059	14
3	PHX	LAX	370	8662	1219235	6
4	LAX	PHX	370	8650	1210173	6
5	PHX	SAN	304	6200	1067278	5
6	SAN	PHX	304	6216	1060204	4
7	SLC	DEN	391	8012	920919	4
8	DEN	SLC	391	7667	893437	6
9	BOS	DCA	399	8484	867688	1
10	DCA	BOS	399	8493	864009	4