Phase 3:

This is the last part of the Flight Reservation Project.
In this phase, you need to create a new schema named 'Flight_Reservations2020Ph3' and restore the provided backup. You are not allowed to modify any of the data unless it is required from this assignment. You are advised to solve the queries in ordered specified in this task. If your database has any extra or altered data, you will lose points! For this phase, you will have to implement some more complicated queries and return a backup of your database. Keep in mind that you may need to answer a few questions to the TA about your decisions.

Phase 3 -- Tasks:

Task 1a: **[8 points]**

In this task, we want you to identify the customers that used our system to book tickets. To do that, you need to create a new column to the USERS table, and name it as 'Active'; define this column as a small integer. Values for this column are '0' for non-active customers and '1' for active. You do not need to create an enumerated list. Consequently, for non-active customers, there is not any ticket information stored in our database. Your task is to update the 'Active' column with '1' for all active customers and with '0' for the non-active. This column must be left empty for the customer representatives and administrators.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.
(4) The select USERS typed command in your submission report file
(5) A screenshot with your query, result grid, and the action output.

Task 1b: **[3 points]**

For this task, you need to return the percentage of active customers.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 2: **[5 points]**

By an error to the system, the 'Type' attribute from table 'Flight' is set to '1'(One-way) for all trips. You need to write a query to fix this problem. Your task is to update the 'Type' attribute from table TICKET to the corresponding number {1: One-way, 2: Round-Trip}. An easy way to do that is by counting how many flights booked per ticket.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 3: [10 points]

Here you need to create a **view** named as 'viewTicketlInfo', that retrieves all useful information associated with a ticket. Specifically, this view should have the following attributes:

- ticketPurchaseDate *(Keep only the date part) – in an ascending order*
- ticketID
- ticketType *– if it is one-way or round trip. Your query needs to return the* ***description*** *of the type*
- custName
- custPhone
- flightID *– in one cell*
- travelDate
- flightClass
- flightFare *– You need to append the US dollar sign ($) before the amount*

Submit:

(1) Task description.
(2) Your 'Create view' command typed in your submission report file. Use only the above-provided names for your attributes. Do NOT alter any name.
(3) A screenshot, with your query, result grid, and the action output.
(4) Then your 'Select view' command typed in your submission report file.
(5) A screenshot with your query, result grid, and the action output.

Task 4: [5 points]

Write a query that retrieves data from your stored 'viewTicketlInfo' view. This query needs to summarize information about the flight fare per ticket. Order your results from the maximum amount to the minimum and then by the customer name. Your query needs to return the following attributes:

- ticketPurchaseDate *renamed to 'Purchase Date'*
- ticketID *as 'Ticket'*
- ticketType *as 'Ticket Type'*
- **custName** *as 'Customer'*
- custPhone *as 'Customer Number'*
- ticketFare *as 'Ticket fare'*

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 5: [10 points]

In this task, we want to count how many seats left for two specific flights. You need to write one query that returns:

- the fight number (merge in one cell the Airline Id + Flight Code),
- the travel date,
- Ids and names from the departure and destination airports,

- departure and arrival time,
- class description,
- the initially available number of seats,
- the total number of tickets sold, and
- the total number of available seats [Available = Initial-Sold].

In your query you need to consider the following two flights for the specific dates:

- UA589 [2020-04-24], and
- UA775 [2020-05-02].

Order your results by flight number and class. Do not forget to name your attributes to something meaningful!

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 6: [8 points]

For this task, you need to provide statistical information for our customers. Specifically, your query needs to return information for all customers' names and phones, the total number of tickets we issued along with the total number of flights per specific customer, and the total fare value. The CFO also requested to provide information about an average flight fare estimation per customer [total fare value by the number of flights]. Also, in the same query, you need to include customers with no travel history. Order your results based on fare value and customer name.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 7: [8 points]

Due to the current situation with COVID-19, our 'Online travel reservation system' CEO decided to grand to all upcoming round-trip flights (travel date after April 20, 2020) a 15% discount. You need to write a query to update only those ticket fares that fulfill those requirements.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot with your query, result grid, and the action output.

Task 8: [8 points]

Our table that associates airline companies with airports (ARLARP) is outdated. You need to make sure that the ARLARP table utilizes all the information existing on the flight's table. In this task, you are only required to insert new records based on the flights' table information, do not delete any airline airport association.

Submit:

(1) Task description.
(2) Your SQL command(s) typed in your submission report file.
(3) The screenshot(s), with your query, result grid, and the action output.

Task 9: [5 points]

For this task, you need to write a query that returns from the table 'Sequence' the routes (booked flights) that are operated from the biggest airplane, in terms of seat capacity.

Submit:

(1) Task description.
(2) Your SQL command typed in your submission report file.
(3) A screenshot(with your query, result grid, and the action output.

Task 10: [15 points]

American Airlines issued a new price policing, effective today, and we need to update our prices accordingly.
The policy dictates the following:

- an increase 10% to all 'Economy' fares round up to the nearest integer value,
- an increase of 20% to 'Extra Space' class compare to the new 'Economy' fare for that specific flight, round up to the nearest integer value
- an increase of 120% for the 'First' class compare to the 'Extra' fare for that specific flight, round up to the nearest integer value.

For a better understanding of the task requirements, let's consider the following example for flight AA2459. Currently, the 'Economy' fare is $60.00, with the new policy will be $66.00. Hence, there is no information for the 'Extra' class, the 'First' class fare should be ($66.00*20%) = $79.20 rounded to $79.00 multiplied with 120% resulting $173.8 rounded to $174.00. So, the first-class fare for flight AA2459 will be $174.00.

For this task you can write more than query, but make sure that you will not hardcode the flight or class info.

Submit:

(1) Task description.
(2) Your SQL command(s) typed in your submission report file.
(3) The screenshot(s) with your query, result grid, and the action output.

Task 11: [5 points]

Return a backup of your database.

Phase 3 -- Canvas Submission Guidelines:

(1) Create your submission report file.
- In the front-page, type a title for your project submission and specify your name, and your team member's name. If you changed teams include the names from your current team only.
- On the second page, include your honor code. Failing to do so will cost **[5 points].**
- Edit your submission file and place your phase 1 introduction, tasks and your proposed solutions, then phase 2 tasks and solutions. After phase 2, append your phase 3 requirements and your proposed solution. Your submission file should look as a professional report. **[5 points].**

(2) Create a folder with your team name, and
- Save your submission file as "Report.docx" or "Report.pdf". Make sure that all commands are editable!
- Save your database backup (self-contained file), named as backup.sql

(3) Zip your folder and submit it on Canvas any time before the deadline.

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Students are required to not share any of the project related documents and solution with others in any way or form even after the completion of the project. Students may, however, show their projects to interviewers.

Late penalty: -5 points per hour

- - - USEFULL- - -

How to **Take** a MySQL Workbench backup
https://www.youtube.com/watch?v=DiWfZy8exfo&feature=youtu.be&t=151

How to **Restore** a MySQL Workbench backup
https://www.youtube.com/watch?v=80XfCtw2MVc&feature=youtu.be&t=110