# **BTD210- Assignment 2**

## **Learning outcomes**

Upon successful completion of this assignment, you will have demonstrated the abilities to:

1. Convert tables into 3NF and infer the associated ERD
2. Implement the relational database model by creating tables and constraints on attributes.
3. Enter data into the database.
4. Query the database to retrieve information.

Please work in **groups** to complete this assignment. This assignment is **worth 10%** of the total course grade and will be evaluated through your written submission, as well as the assignment demo. During the assignment demo, group members are randomly selected to present the answers to each of the assignment questions. Group members who are not present during the assignment demo will lose the demo mark (up to 30% of assignment mark).

## **Submission**

Please submit the submission file(s) through Blackboard. Only one person must submit for the group and only the last submission will be marked.

* Assignment SQL file: Asg2\_grpGroupNumber.SQL
* Assignment document: Asg2\_grpGroupNumber.docx

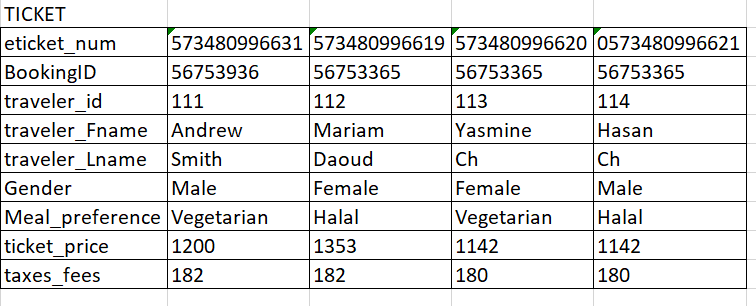
## **Assignment overview**

The CheapFare airline reservation company is a flight booking company of domestic and international airline tickets. The company wants to use a relational database for its operations. The manager has decided to refer to exemplar of bookings that are usually issued following a flight reservation by customers.

During the interview with the airline reservation company manager, the following information has been obtained:

1. Each booking may correspond to a one-way trip or a round-trip with several flights booked to lead to destination.
2. A flight is identified with a flight number (flight\_num). A flight number (e.g. AF393) determines the airline the flight belongs to (e.g. AF), and might be operated by a partnering airline (operatedBy is an airline\_code).
3. Each flight is always associated with the same aircraft, determined with aircraft\_code and description.
4. Flights determined with a flight\_num always operate from same departure airport and to the same arrival airport. Departure dates and time and arrival dates and time may vary.
5. A booking may generate several e-tickets, one per passenger. For example, booking\_id 56753365 for a mom and two kids generates 3 different e\_tickets.
6. Each ticket is identified with an eticket\_num and it belongs to one traveler. The database should also save the traveler meal preference (optional), the ticket price and the ticket tax fees.
7. A traveler is identified with a traveler id. The attributes are last name, first name and gender.

****



## **Part I. Database Normalization and Entity Relationship Model (40%)**

1. For each of the given tables, convert to 3NF, *showing all steps* (from 1NF to 3NF). Use the arrow notation. Do NOT include derived attributes.
2. Combine the tables obtained at the end of the normalization process into one logical design. Draw the *ERD* using Microsoft Visio. Note that your design must not have any many-to-many relationships or multi-valued attributes. PK and FK must be clearly stated. Strong and weak relationships must be labelled correctly. Export the ERD to .jpg and insert in your assignment document file.

## **Part II. Database IMPLEMENTATION (20%)**

In your SQL file, write the SQL statements to:

1. Create a database named BTD210\_Asg2\_YourGroupNo.
2. Create the tables of your relational database model. These must match your design in Part I. Specify the PK, FK, and other necessary constraints.
3. In your assignment document file, paste a screenshot of the object explorer displaying the table’s columns, keys and constraints. You can include many screenshots.

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

1. Enter the data from the samples shown in the given tables. Include all *insert* queries in you SQL file. Paste the output in your assignment document file.

A picture containing text

Description automatically generated

Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Text, table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

## **Part III. Data Manipulation (40%)**

For the following questions, include in your assignment document file,

1. The **SQL command** (both in the SQL file and the document file)
2. The **output** in text format,
3. **The number of rows** affected.
4. List the airline, aircraft and airport details for flight number AF393. Display the flight number, airline code and name, aircraft code and description, departure airport code and name and arrival airport code and name.

The **SQL command:**

SELECT FLIGHT\_NUM, A.AIRLINE\_CODE, B.AIRLINE\_NAME, A.AIRCRAFT\_CODE, C.AIRCRAFT\_DESC, A.DEP\_AIRPORT\_CODE, D.AIRPORT\_NAME, A.ARR\_AIRPORT\_CODE, E.AIRPORT\_NAME

FROM FLIGHT A JOIN AIRLINES B ON A.AIRLINE\_CODE = B.AIRLINE\_CODE

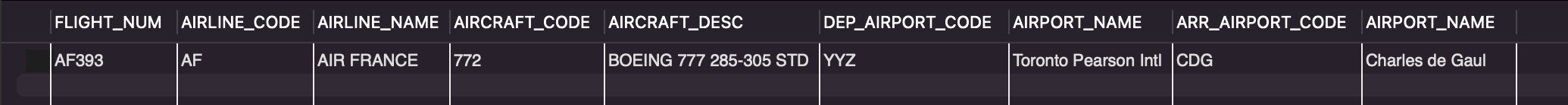
JOIN AIRCRAFT C ON A.AIRCRAFT\_CODE = C.AIRCRAFT\_CODE

JOIN AIRPORT D ON A.DEP\_AIRPORT\_CODE = D.AIRPORT\_CODE

JOIN AIRPORT E ON A.ARR\_AIRPORT\_CODE = E.AIRPORT\_CODE

WHERE FLIGHT\_NUM = 'AF393';

The **output** in text format: (not able to copy all the output in text format)



**The number of rows** affected: 1 row(s) returned

1. List the flight details for bookingId 56753365. Display booking id, booking date, the flight number (there should show several flights records for this booking), departure airport code, departure date and time, arrival airport code, arrival date and time. Order the rows by departure date and time.

The **SQL command:**

SELECT A.BOOKING\_ID, A.BOOKED\_ON, A.FLIGHT\_NUM, B.DEP\_AIRPORT\_CODE, C.DEP\_DATE, C.DEP\_TIME, B.ARR\_AIRPORT\_CODE, C.ARR\_DATE, C.ARR\_TIME

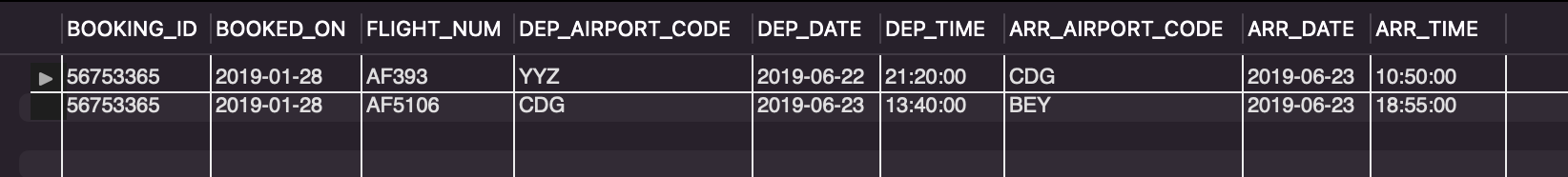
FROM BOOKING A JOIN FLIGHT B ON A.FLIGHT\_NUM = B.FLIGHT\_NUM

JOIN ROUTE C ON C.FLIGHT\_NUM = B.FLIGHT\_NUM

WHERE A.BOOKING\_ID = '56753365'

ORDER BY DEP\_DATE, DEP\_TIME;The **output** in text format: (not able to copy all the output in text format)

The **output** in text format:



**The number of rows** affected: 2 row(s) returned

1. List all passengers for bookingID 56753365. Display the bookingID, eticket number, traveler first and last name. This should show 3 passengers with 3 different etickets.

The **SQL command:**

SELECT B.BOOKING\_ID, B.ETICKET\_NUM, A.TRAVELER\_FNAME, A.TRAVELER\_LNAME

FROM TRAVELER A JOIN TICKET B ON A.TRAVELER\_ID = B.TRAVELER\_ID

WHERE B.BOOKING\_ID = '56753365'

The **output** in text format:

Table

Description automatically generated

**The number of rows** affected: 3 row(s) returned

1. Calculate the total fees for bookingID 56753365. This should be the sum of ticket price and tax fees for the three tickets associated with this booking. (Hint: this does not need a group by). Use a where clause to select only the tickets for that booking and use the sum() function.

The **SQL command:**

SELECT SUM(TICKET\_PRICE + TAXES\_FEES) AS TOTAL\_FEES FROM TICKET WHERE BOOKING\_ID = '56753365';

The **output** in text format:

Table

Description automatically generated

**The number of rows** affected: 1 row(s) returned

1. Show the list of all bookings and the number of tickets associated with each.

The **SQL command:**

SELECT BOOKING\_ID, COUNT(ETICKET\_NUM) AS NUMBER\_OF\_TICKETS

FROM TICKET

GROUP BY BOOKING\_ID;

The **output** in text format:

Table

Description automatically generated

**The number of rows** affected: 2 row(s) returned

1. Show the list of all bookings and the corresponding total fees. The result set should show the bookingID and the total fees.

The **SQL command:**

SELECT BOOKING\_ID, SUM(TICKET\_PRICE + TAXES\_FEES) AS TOTAL\_FEES

FROM TICKET

GROUP BY BOOKING\_ID;

The **output** in text format:

Table

Description automatically generated

**The number of rows** affected: 2 row(s) returned

1. Show the list of all booking whose total fees is larger than $1000.

The **SQL command:**

SELECT BOOKING\_ID, SUM(TICKET\_PRICE + TAXES\_FEES) AS TOTAL\_FEES

FROM TICKET

GROUP BY BOOKING\_ID

HAVING TOTAL\_FEES > 1000;

The **output** in text format:

Table

Description automatically generated

**The number of rows** affected: 2 row(s) returned

1. Create a view named “bookingFees” that saves query in question 6.

The **SQL command:**

CREATE VIEW BOOKINGFEES AS

SELECT BOOKING\_ID, SUM(TICKET\_PRICE + TAXES\_FEES) AS TOTAL\_FEES

FROM TICKET

GROUP BY BOOKING\_ID;

The **output** is the Table we got in view:

Table

Description automatically generated

**The number of rows** affected: 0 row(s) affected

1. Write a query that uses view bookingFees and to answer question 7.

The **SQL command:**

SELECT BOOKING\_ID, TOTAL\_FEES

FROM BOOKINGFEES

WHERE TOTAL\_FEES > 1000;

The **output** is the Table we got in view:

Table

Description automatically generated

**The number of rows** affected: 2 row(s) affected

## **Part IV: Group Work**

1. Add this declaration on the top of your file.

We, ------------(mention your names), declare that the attached assignment is our own work in accordance with the Seneca Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including web sites) **or distributed to other students.**

1. Specify what each member has done towards the completion of this work:

|  |  |  |
| --- | --- | --- |
|  | Name | Task(s) |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |