National University of Singapore
Department of Computing

BT2102 Data Management and Visualisation

Assignment 1A and 1B (Individual):
The Airline Seat-booking System (TASS)

Total marks: 1A = 20 marks; 1B = 20 marks

ASSIGNMENT 1B

BACKGROUND
You are provided with a conceptual and logical data model based on the narrative in Assignment 1A. They are included in **Appendix A** of this document.

WHAT ARE YOU REQUIRED TO DO?
1. **Implement the relational database (named "tass") on MySQL** (using the MySQL Workbench) based on the conceptual and logical data models provided and **by importing the data in the '.csv' files (these files are provided in the DataFiles.zip)**:
    a. Data representing the entities, attributes and relationships for TASS system are given in the following '.csv' files:
        i. **Airline**: Airline.csv
        ii. **Airport**: Airport.csv
        iii. **Booking**: Booking.csv
        iv. **BookingOffice**: BookingOffice.csv
        v. **City**: City.csv
        vi. **Class**: Class.csv
        vii. **Country**: Country.csv
        viii. **Currency**: Currency.csv
        ix. **Customer**: Customer.csv
        x. **CustomerEmail**: CustomerEmail.csv
        xi. **CustomerPhone**: CustomerPhone.csv
        xii. **Exchange**: Exchange.csv
        xiii. **Flight**: Flight.csv
        xiv. **FlightAvailability**: FlightAvailability.csv
        xv. **Payment**: Payment.csv
        xvi. **Status**: Status.csv
    b. You need to create the tables in MySQL workbench using the **logical data model specified in Appendix A** before you import the data. Declare the fields using MySQL data types "varchar(xx)", "int", "float", or "datetime" where appropriate. The size of the data structure is up to you e.g. CustomerID could be defined as varchar(30), FirstName as varchar(30), FlightPrice and TotalPrice as float, ArrivalDateTime and DepartureDateTime as datetime, TotalBusinessSeats and BookedEconomySeats as int, DestinationAirportCode as varchar(30), FlightNumber as varchar(30), etc. **DO NOT CHANGE THE TABLE OR COLUMN NAME.**
    c. Import the data into the newly created tables. **Where the name is a MySQL reserved word (e.g. names used to define table names or variables) such as "Exchange", "Date" and "Status", use a backtick "`" to capture the names e.g. `exchange`, `date`, or `status`.**
    d. When you import the data from the csv files into MySQL Workbench, take note that there is a certain sequence by which the tables are to be created and imported because of the dependency due to foreign keys. Follow this sequence:
        i. Country
        ii. City
        iii. Airline
        iv. Flight
        v. BookingOffice
        vi. Currency

       vii.   Airport
      viii.   FlightAvailability
        ix.   Class
         x.   Status
        xi.   Customer
       xii.   CustomerEmail
      xiii.   CustomerPhone
      xiv.   Exchange
       xv.   Booking
      xvi.   Payment

e. When the data are imported into MySQL Workbench, you will have the set of tables for the entities, attributes and relationships in the database, ready for executing the queries in **Appendix B**.

f. You may consider the database to be static i.e. the database is a snapshot or an extraction of data, which will not automatically update itself with any value.

g. You do not have to calculate any values by yourself.

h. **You also do not have to define any derived attributes when creating tables.**

i. **Assumptions:**
    i. All mentions of flight prices will be in Begonia dollars.
    ii. Airport tax is stored in local currencies where the airport is located.

2. **Implement the queries listed in Appendix B** directly in MySQL Workbench as MySQL scripts. Results tables for **some** of the queries are given.

3. **Include your final MySQL scripts** for the 19 queries in **Appendix B** into a MySQL script file. A sample template file for MySQL scripts is given in file "**BT2102-1B-MySQL.sql**" available in canvas. You are to use this template for your answers to the 19 queries.

DELIVERABLES

1. The MySQL script file containing a set of MySQL scripts to answer the 19 queries listed in **Appendix B**.
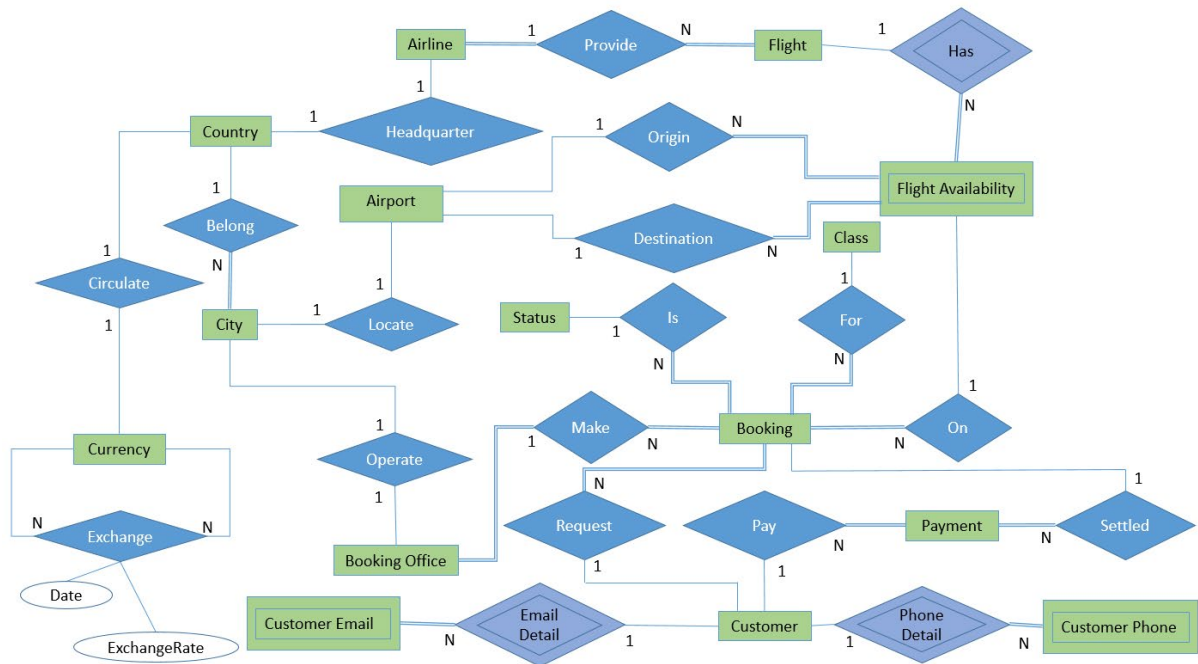

SUBMISSION

1. Name the MySQL script file as "**1B_A1234567B_TanGeNius.sql**" (where A1234567B is your student number and TanGeNius is your name). **Please follow this naming convention accordingly.**

2. Upload the MySQL script file to "**Assignment 1B**" in Canvas->Assignments by the assignment due date.

3. **THERE IS NO REQUIREMENT FOR YOU TO MAKE A PRESENTATION OR DEMONSTRATION OF THE MYSQL SCRIPT FILE.**

4. Please be reminded that **plagiarism is a serious offence**. Disciplinary actions will be taken against those caught cheating.

EVALUATION

1. This assignment will be evaluated as follows:
    a. **The correctness of the MySQL scripts** answering the 19 queries. (**20 marks**)

Appendix A

Conceptual Data Model

Logical Data Model

AIRLINE (AirlineCode, AirlineName, CountryCode) Primary Key AirlineCode Foreign Key CountryCode references Country (CountryCode)	FLIGHT (FlightNumber, BusinessClassIndicator, AirlineCode) Primary Key FlightNumber Foreign Key AirlineCode references Airline (AirlineCode)
COUNTRY (CountryCode, CountryName) Primary Key CountryCode	CITY (CityCode, CityName, CountryCode) Primary Key CityCode Foreign Key CountryCode references Country (CountryCode)
BOOKING OFFICE (OfficeID, CityCode) Primary Key OfficeID Foreign Key CityCode references City (CityCode)	CURRENCY (Currency, Abbreviation, CountryCode) Primary Key Currency Foreign Key CountryCode references Country (CountryCode)
EXCHANGE (FromCurrency, ToCurrency, Date, ExchangeRate) Primary Key FromCurrency, ToCurrency, Date Foreign Key FromCurrency references Currency (Currency) Foreign Key ToCurrency references Currency (Currency)	AIRPORT (AirportCode, AirportName, AirportTax, CityCode) Primary Key AirportCode Foreign Key CityCode references City (CityCode)
FLIGHT AVAILABILITY (FlightNumber, ArrivalDateTime, DepartureDateTime, TotalBusinessSeats, BookedBusinessSeats, TotalEconomySeats, BookedEconomySeats, OriginAirportCode, DestinationAirportCode,) Primary Key FlightNumber, ArrivalDateTime, DepartureDateTime Foreign Key FlightNumber references Flight (FlightNumber) Foreign Key OriginAirportCode references Airport (AirportCode) Foreign Key DestinationAirportCode references Airport (AirportCode)	BOOKING (BookingID, BookingDate, FlightPrice, TotalPrice, FlightNumber, ArrivalDateTime, DepartureDateTime, ClassID, StatusID, OfficeID, CustomerID) Primary Key BookingID Foreign Key FlightNumber references Flight Availability (FlightNumber) Foreign Key ArrivalDateTime references Flight Availability (ArrivalDateTime) Foreign Key DepartureDateTime references Flight Availability (DepartureDateTime) Foreign Key ClassID references Class (ClassID) Foreign Key StatusID references Status (StatusID) Foreign Key OfficeID references Booking Office (OfficeID) Foreign Key CustomerID references Customer (CustomerID) Derived Attribute TotalPrice = FlightPrice * ExchangeRate + OriginalAirportTax * ExchangeRate + DestinationAirportTax * ExchangeRate
CLASS (ClassID, class) Primary Key ClassID	STATUS (StatusID, Status) Primary Key StatusID
CUSTOMER (CustomerID, FirstName, LastName, Nationality, Street, City, Province, Country, PostalCode) Primary Key CustomerID	CUSTOMER PHONE (CustomerID, LocalNumber, CountryCode, AreaCode) Primary Key CustomerID, LocalNumber, CountryCode Foreign Key CustomerID references Customer (CustomerID)
CUSTOMER EMAIL (CustomerID, Email) Primary Key CustomerID, Email Foreign Key CustomerID references Customer (CustomerID)	PAYMENT (PaymentID, PaymentDate, PaidAmount, Balance, CustomerID, BookingID) Primary Key PaymentID Foreign Key CustomerID references Customer (CustomerID) Foreign Key BookingID references Booking (BookingID)

Appendix B: Queries

1. List all the customers whose mail addresses in Begonia. Display CustomerID, FirstName, and LastName. Sort results by CustomerID in ascending order. **(0.5 mark)**
    Columns in the result table:
    CustomerID, **FirstName**, **LastName**.
    Results table:

CustomerID	FirstName	LastName
CM018	Jean	Wilson
CM020	Jean	Thomas
CM023	Lilith	Evans
CM024	May	Jones
CM054	Olivia	Jones
CM055	Hannah	Jones
CM060	Lilith	Brown
NULL	NULL	NULL

2. List all different flight numbers that have been booked in November and will departure in 2023 January. Display FlightNumber only. **(0.5 mark)**
    Columns in the result table:
    FlightNumber.

3. List average currency exchange rates from BegoniaDollar to other currencies, e.g. from Begonia Dollar to Carnation Dollar. Display average exchange rate(s) that is (are) greater than 1. Sort them by average exchange rate in ascending order. Display FromCurrency, ToCurrency, and average exchange rate as AverageExchangeRate. **(0.5 mark)**
    Columns in the result table:
    FromCurrency, **ToCurrency**, **AverageExchangeRate**.

4. List number of available flights for all flight numbers between BorAirport (airport code is 'AP02') and RadAirport (airport code is 'AP09'). Display FlightNumber and number of flights as NumberOfFlights. You do not have to join "Airport", you may use "AP02" and "AP09" directly. **(0.5 mark)**
    Columns in the result table:
    FlightNumber, **NumberOfFlights**.

5. List number of flight numbers (FlightNumber) provided for each airline company. Display AirlineCode and number of flight numbers as NumberOfFlightNumbers. You do not have to consider available flights for each flight number, simply provides the number of flight numbers. **(0.5 mark)**
    Columns in the result table:
    AirlineCode, **NumberOfFlightNumbers**.

6. List airline companies containing "u" in their names. Show the number of flight numbers having business class and the number of flight numbers do not provide business class for each of them. Display AirlineName, BusinessClassIndicator, and number of flight numbers as NumberOfFlightNumbers. **(0.5 mark)**
    Columns in the result table:
    AirlineName, **BusinessClassIndicator**, **NumberOfFlightNumbers**.
    Results table:

AirlineName	BusinessClassIndicator	NumberOfFlightNumbers
GauAir	Yes	2
GauAir	No	6
TulAir	No	1
TulAir	Yes	3

7. List top 3 booking offices which have highest number of bookings. You do not need to consider the status of booking; bookings with any status do count. Display OfficeID and number of bookings as NumberOfBookings.
   (0.5 mark)

   Columns in the result table:
   OfficeID, **NumberOfBookings**.

8. List 5 flight numbers which have lowest average percentage of total booked seats (including economy and business) among their available flights. Display FlightNumber only.   **(0.5 mark)**
   Column in the result table:
   FlightNumber.
   Results table:

FlightNumber
FN056
FN012
FN052
FN022
FN006

9. For those booking status is "ST01", calculate the average number of payments to settle those bookings. Display the average number of payments as AvgNumberOfPayments. You do not have to join "Status", may use "ST01" directly.   **(1 mark)**
   Columns in the result table:
   AvgNumberOfPayments.

10. Calculate the percentage of foreigners in the customer list. Foreigner is identified as people whose country in mail address is not the same as his/her nationality. Display the percentage as ForeignerPercent. Round the percentage up to 2 decimal places, e.g. 50.99%.   **(1 mark)**
    Columns in the result table:
    ForeignerPercent.
    Results table:

ForeignerPercent
93.33%

11. List all Bookings placed between "2022-10-16" and "2022-11-15" which have not gotten any payments yet. "placed" refers to the booking that has been made; however, it does not mean this booking is confirmed. You also do not have to consider booking status. Display BookingID only.   **(1 mark)**
    Columns in the result table:
    BookingID.

12. List all customers come from Begonia (Nationality is "Begonia"). Find their latest payment dates. Display CustomerID, FirstName, LastName, and latest payment date as LatestPayment.   **(1 mark)**
    Columns in the result table:
    CustomerID, **FirstName**, **LastName**, **LatestPayment**.

13. Sort customers by CustomerID in ascending order. Select the top five customers. List number of bookings, number of emails, number of phones for each of them. Display CustomerID, number of bookings as NumberOfBookings, number of Emails as NumberOfEmails, number of phones as NumberOfPhones.

(1 mark)

Columns in the result table:
CustomerID, **NumberOfBookings**, **NumberOfEmails**, **NumberOfPhones**.
Results table:

CustomerID	NumberOfBookings	NumberOfEmails	NumberOfPhones
CM001	0	1	1
CM002	2	5	1
CM003	1	0	2
CM004	0	0	2
CM005	2	1	2

14. List top 5 flight numbers (FlightNumber) which have the most fluctuating flight price. Display FlightNumber, standard deviation of flight price as StdFlightPrice, average available business seats of all available flights for each flight number (including those not in the "booking" table) as AvgAvailableBusinessSeats, average available economy seats of all available flights for each flight number (including those not in the "booking" table) as AvgAvailableEconomySeats. You do not have to calculate flight price for business class yourself; the value shown in "FlightPrice" column is already the final flight price for business or economy class. "Including those not in the "booking" table" means including those available flights even if they have not yet been booked. **(1 mark)**
Columns in the result table:
FlightNumber, **StdFlightPrice**, **AvgAvailableBusinessSeats**, **AvgAvailableEconomySeats**.

15. Find customers who do not have email address but have phone number. List them who only has one phone number. Display CustomerID, full name (format as "FirstName LastName", e.g. "Jean Smith"), and phone number (format as "CountryCode-AreaCode-LocalNumber", e.g. "11-111-123456"). Sort them by CustomerID in ascending order. **(2 marks)**
Columns in the result table:
CustomerID, **FullName**, **PhoneNo**.
Results table:

CustomerID	FullName	PhoneNo
CM019	Vera Thomas	19-191-286618
CM020	Jean Thomas	13-131-205776
CM023	Lilith Evans	14-141-236795
CM031	Erica Smith	12-121-599234
CM058	Nicole Wilson	16-161-129040

16. Among cities in each country, list city with the highest sum of TotalPrice. Exclude canceled bookings. Display CountryName, CityName and sum of total price as TotalSales. Sort records by CountryName in ascending order. Assume you do not know the StatusID for "canceled". That means, you need to JOIN tables to exclude canceled bookings. **(2 marks)**
Columns in the result table:
CountryName, **CityName**, **TotalSales**.
Results table:

CountryName	CityName	TotalSales
Begonia	Boron	7852.670104980469
Carnation	Cesium	45322.049560546875
Gaura	Germanium	10438.485076904297
Ipomoea	Iridium	3464.7450256347656
Rose	Rhodium	75970.4296875
Tulip	Thorium	90332.939453125

17. List top 5 canceled bookings which have the highest FlightPrice. Display BookingID, FlightNumber, ArrivalDateTime, DepartureDateTime, origin city code as OriginCityCode, destination city code as DestinationCityCode. Assuming you do not know the StatusID for "canceled" booking; you will have to JOIN tables to select canceled bookings. You do not have to calculate the flight price yourself. Use the value shown in the column "FlightPrice" directly. **(2 marks)**
Columns in the result table:
BookingID, **FlightNumber**, **ArrivalDateTime**, **DepartureDateTime**, **OriginCityCode**, **DestinationCityCode**.

18. If the exchange rate from RoseDollar to CarnationDollar change (exchange rate from CarnationDollar to RoseDollar will not change), how many different flight numbers (FlightNumber) will be affected based on current booking situation. Display number of flight numbers as NumberOfPotentialAffected. You do not have to consider status of booking. Include all bookings with any status. **(2 marks)**
Column in the result table:
NumberOfPotentialAffected.
Results table:

NumberOfPotentialAffected
5

19. Rank airports based on total number of flights arrival and departure in weekdays. Display AirportCode and rank. (Airports with the same number of flights are treated as having the same rank. For example, if there are 50 flights arrival and departure during weekdays in Airport01 and Airport 02. 48 flights in Airport 03. The ranking is as shown in the table below. Note that Airport 01 and Airport 02 are in the same rank 1, while the Airport 03 is rank 2.) (Hint: consider using the DENSE_RANK() function). You have to consider all available flights. You do not have to consider whether this flight has been booked or not. **(2 marks)**
Columns in the result table:
AirportCode, **Rank**.

Airport	Number of Flights	Rank
Airport 01	50	1
Airport 02	50	1
Airport 03	48	2

Results table:

AirportCode	Rank
AP05	1
AP09	2
AP11	2
AP03	3
AP07	3
AP06	4
AP12	5
AP02	6
AP04	7
AP10	7
AP08	8
AP01	9