FIT3003 Group Assignment - Sem 2/2018 (Weight = 25%)
Due date: Week 12, Monday 15-Oct-2018, 12 noon

A. General Information and Submission

- This is a group assignment. One group consists of 2 students only. You need to report your group composition to your tutor as soon as possible.
- *Submission method*: Submission is online through Moodle
- *Penalty for late submission*: 10% deduction for each day
- *Oracle account details*: You will need to supply with this assignment an Oracle username and password, used for this assignment.
- *Assignment Coversheet*: You will need to sign the assignment coversheet

B. Problem Description – Open Flights & Travel Data Warehouse

OpenFlights is a tool that let you map your flights around the world, search and filter them in all sorts of interesting ways, calculate statistics automatically, and share your flights and trips with friends and the entire world (if you wish). It's also the name of the open-source project to build the tool.

Our Open Flights & Travel (OFT) database maintains flights information around the globe for all airports and their inbound and outbound routes. The routes and flights are collected from OpenFlights website as of January 2012. The database also stores booking transactions for passengers during the period 01-01-2006 and 31-12-2009, and the records of membership who joined during 01-01-2005 and 31-12-2014.The captured booking transaction data is mainly for flights arriving and departing from Australia besides some records for other destinations. The database tables can be found at **opfl**. You can, for example, execute the following query:

select * from opfl.<table_name>;

The data definition of each table in opfl is as follows:

Table Name (PK/FK)	**Attributes and Data Types**		**Notes**
Airports PRIMARY KEY: AirportID	AirportID	NUMERIC	This table stores airports information. It contains around 7,000 airports spanning the globe. IATA_FAA is a 3-letter. ICAO is a 4-letter code. Latitude and longitude are in degrees. Altitude in feet. Time zone stores hours offset from UTC. DST is daylight savings time and take one of these characters: E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None), or U (Unknown). *The ICAO: airport code or location indicator is a four-character alphanumeric code designating each airport around the world.*
	Name	VARCHAR	
	City	VARCHAR	
	Country	VARCHAR	
	IATA_FAACode	VARCHAR	
	ICAO	VARCHAR	
	Latitude	NUMERIC	
	Longitude	NUMERIC	
	Altitude	NUMERIC	
	Timezone	NUMERIC	
	DST	CHAR	

			The IATA or FAA: Federal Aviation Administration identifier is a three- or four-letter alphanumeric code identifying United States airports
Airlines PRIMARY KEY: AirlineID	AirlineID Name Alias IATACode ICAO CallSign Country Active	NUMERIC VARCHAR VARCHAR CHAR CHAR VARCHAR VARCHAR CHAR	This table stores the information of around 6,000 airlines. IATA is 2-letter code. ICAO is 3-letter code. Active indicates whether the airline has been recently operational. *The International Air Transport Association uses sets of 3-letter IATA identifiers, which are used for airline operations, baggage routing, and ticketing. The ICAO airline designator is a code assigned to aircraft operating agencies, aeronautical authorities, and services related to international aviation.*
Routes PRIMARY KEY: RouteID FOREIGN KEYS: AirlineID, SourceAirportID, DestAirportID	RouteID AirlineID SourceAirportID DestAirportID Stops Equipment Distance ServiceCost NewlyOpened	NUMERIC NUMERIC NUMERIC NUMERIC NUMERIC VARCHAR NUMERIC NUMERIC CHAR	This table stores the information of around 59,000 routes between 3,209 airports on 531 airlines spanning the globe. Equipment contains 3-letter codes separated by spaces for aircraft types used on this route. Distance is in Km. ServiceCost is the cost to run all airline services offered by the airline in this route. NewlyOpened indicates whether this route is opened in the last three years.
Aircrafts PRIMARY KEY: IATACode	IATACode ICAOCode Manufacturer Model WakeCategory	CHAR CHAR VARCHAR VARCHAR CHAR	It stores aircrafts types and manufacturers information. Wake category indicates the weight of the aircraft (Heavy, Medium, and Light). IATACode is 3-charater code. The code UKN denotes unknown aircraft model details. *IATA aircraft type designators are trigram letter/digit codes used for aircraft models, like "J41" (British Aerospace Jetstream 41) and "744" (Boeing 747-400).*
Flights PRIMARY KEY: FlightID FOREIGN KEYS: RouteID, AircraftID	FlightID FlightDate DepartTime ArrivalTime Fare RouteID AircraftID	CHAR DATE DATE DATE NUMERIC NUMERIC CHAR	It stores the information of around 50,000 flights. DepartTime and Arrival Time are in local time of source and destination airports. Fare is the minimum charge for an adult on this flight and it is in USD.
Passengers PRIMARY KEY: PassID	PassID FirstName LastName EmailAddress Age Nationality	NUMERIC VARCHAR VARCHAR VARCHAR NUMERIC VARCHAR	It stores the details of around 10,000 passengers. They are mostly from Australia.

Transactions PRIMARY KEY: PassID, FlightID FOREIGN KEYS: PassID, FlightID	PassID	NUMERIC	It stores around 25,000 booking transactions at different dates between 2006 and 2009. TotalPaid is the total amount of money paid by the passenger for this trip and it is in USD. Discount is applied on some transactions.
	FlightID	VARCHAR	
	BookingDate	DATE	
	Discount	NUMERIC	
	TotalPaid	NUMERIC	
Airline_Services PRIMARY KEY: ServiceID	ServiceID	NUMERIC	It stores 11 standard airline services offered by all airlines. However, each airline may choose to offer only some of them based on different routes and their business.
	Name	VARCHAR	
	Description	VARCHAR	
Provides PRIMARY KEY/ FOREIGN KEY: AirlineID, ServiceID	AirlineID	NUMERIC	It records the information about which airline offers which services.
	ServiceID	NUMERIC	
Promotion PRIMARY KEY： PromotionID	PromotionID	VARCHAR	It stores the information for promotion that applied to the membership joining, each promotion has a discount (e.g. 0.1 means 10% off). The start date and end date represent the period of this promotion.
	Discount	NUMERIC	
	StartDate	DATE	
	EndDate	DATE	
MembershipType PRIMARY KEY： MembershipTypeID	MembershipTypeID	CHAR	It stores 4 different membership, each type has a corresponding membership joining fee. The “period” is the period that can last once a membership has been joined. The unit for the period is “year”.
	MembershipName	VARCHAR	
	MembershipFee	NUMERIC	
	Period	NUMERIC	
MembershipJoin Records PRIMARY KEY: PassID, JoinDate FOREIGN KEYS: PassID, MembershipTypeid, Promotion	Passid	NUMERIC	It stores the records of passengers who joined different types of membership during 01-01-2005 and 31-12-2014. The enddate indicates when the membership is expired.
	MembershipTypeid	CHAR	
	JoinDate	DATE	
	EndDate	DATE	
	Promotion	VARCHAR	

C. Tasks

The assignment is divided into **FIVE** main tasks:

1. **Design a data warehouse for the above OFT database.**
   You are required to create a data warehouse for the OFT database.
   The management is especially interested in the following fact measures:
   - Average Total Paid for tickets
   - Average Agent Profit (total paid – flight fare)
   - Average Passenger Age
   - Total Number of Routes
   - Average Route Distance
   - Total Service Cost
   - Average Membership Sales

   The following show some possible dimension attributes that you may need in your data warehouse:
   - Source Airport
   - Source Airport City
   - Source Airport Country
   - Destination Airport
   - Destination Airport City
   - Destination Airport Country
   - Airline Services
   - Flight Date (Day[Sat, Sun, … Fri], Month[Jan-Dec], Year)
   - Flight Type (Domestic, International)
   - Flight Class (First Class [TotalPaid >= 2*Flight Fare], Business Class [1.5 * Fare =< Total Paid < 2 * Fare], Economy Class [Total Paid < 1.5 * Fare])
   - Passenger Nationality
   - Passenger Type(Children [Age<11], Teenager [11<=Age<=17], Adult [18<=Age<=60], [Elder (>60)])
   - Membership Type
   - Membership Join Date (Month,Year)

   For each attribute, you may apply your own design decisions on specifying a range or a group, but make sure to specify them in your submission.

- **Preparation stage.**
  Before you design the data warehouse, you need to make sure that you have explored the operational database and have done data cleaning when necessary. If you have done the data cleaning process, you need to explain what strategies you have taken to explore and clean the data.

  The outputs of this task are:

  a) The E/R diagram of the operational database,

  b) If you have done the data cleaning, explain what kind of data cleaning process that you have done (you need to show the SQL to explore the operational database, and SQL of the data cleaning, as well as the screenshot of data *before* and *after* data cleaning),

- **Designing the data warehouse by drawing star/snowflake schema.**

  The star schema for this data warehouse consists of three fact tables: one is related to the **Route**, another is related to the **Transaction**, and the last one is related to **Membership_Sales**. You need to identify the fact measures and dimensions. The following queries might help you to identify the fact measures and dimensions:

  - What is the total number of passengers who departed from "Melbourne Intl" airport in 2008?
  - What is the average paid ticket price between Melbourne and Sydney?
  - What is the top 3 average ages of passengers traveling on business class from an Australian airport?
  - What is the total number of incoming routes to a particular city?
  - What is the average route distance between airports in Europe?
  - What is the average cost of each airline service?
  - What is the average sales of gold membership for Adult passengers joined in each year?

  You also need to pay attention to the granularity of your fact tables, so you have to create **two versions** of star/snowflake based on different level of aggregation.
  The two versions of the star/snowflake represent different level of aggregation. Version-1 should be in level 2, which means high level of aggregation. Version-2 should be in level 0, which means no aggregation.

Version Name	Level
Version-1	Level 2 (High aggregation)
Version-2	Level 0 (No aggregation)

  The star/snowflake schema of both versions you created must contain **Bridge Table** and **Temporal**. You can choose to use Hierarchy or Non-Hierarchy, but you need to provide the reason why do you use Hierarchy or Non-Hierarchy. If there is any Determinant Dimension, you need to denote the Determinant Dimension clearly with a "*" besides your dimension's name (e.g. "*ABC_DIM"). You can use different temporal data warehousing technique (you can use any SCD type except for SCD type-0 and SCD type-1) for the temporal dimension, and provide the reasons of your choice.

The outputs of this task are:

c) Two versions of star/snowflake schema diagrams,

d) A short explanation of why you chose hierarchy or non-hierarchy,

e) The reasons of the choice of SCD type for temporal dimension,

f) A short explanation of the difference among the two versions.

2. **Implement the two versions of star/snowflake schema using SQL.**
You need to implement the star/snowflake schema for the two versions that you have drawn in Task 1 above. It means that you need to create the different fact and dimension tables for two versions, and populate these tables accordingly.

When naming the fact tables and dimension tables, you need to give the identical name for the two versions and end with the version number to differentiate them.
For example, "Transaction_fact_v1" for version-1 and "Transaction_fact_v2" for version-2.

The output is a series of SQL statements to perform this task. You will also need to show that this task has been carried out successfully.

If your account is full, you will need to drop all of the tables that you have previously created during the tutorials.

The outputs of this task are:

a) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-1

b) SQL statements (e.g. create table, insert into, etc) to create the star/snowflake schema Version-2

c) Screen shots of the tables that you have created; this includes the contents of each table that you have created. If the table is very big, you can only show the first part of the data.

3. **Basic Reports (Two reports in this task).**

You are required to generate the following reports using both data warehouse versions, **version-1 (Level 2)** and **version-2 (Level 0 no aggregation)**, that you have implemented in Task 2. For each report, you ought to produce the SQL command and sample report output.

Note: Since you need to use the no aggregation data warehouse to generate the following reports which have the aggregated values, several steps of pre-processing will be needed, which means you might need multiple SQL statements to generate these reports.

Report 1:

What is the top 3 average ages of passengers traveling on business class from an Australian airport?

Report 2:

What is the total number of newly joined gold membership for Adults passenger in each month?

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands, and

(*c*) The screenshots of the query results (or part of the query results).

4. **Create the following advanced reports using OLAP queries (Six reports in this task).**

You are required to use the data warehouse that you implemented, based on **star schema version-1 (Level 2)** and **star schema version-2 (Level 0 no aggregation)**, to generate the following reports. For each report, you need to produce the OLAP SQL command and sample report output.

Report 3: Transactions' Report

Produce the following report (Note that the figures in the sample reports below are not accurate. However, your reports must contain all the columns shown in the sample reports below).

Flight Year	Flight Type	Flight Class	Source Country	Destination Country	Number of Transactions	Average Agent Profit (USD)
2006	Domestic	First Class	Any Country	Australia	200	400
2007	Any Types	Business	Greece	Australia	2000	300
2008	International	Economy	Australia	Syria	50	500
2008	Any Types	Any Class	Any Country	Any Country	4000	200
……	……	……	……		……	

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands, and

(*c*) The screenshots of the query results (or part of the query results).

Report 4: Report with proper sub-totals:

Produce one report using the Cube or Roll-up operator.

What are the sub-total and total agent profits of airports and airlines?

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands that include sub-totals, using the Cube or Roll-up or Partial Cube/Roll-up operators, and

(*c*) The screenshots of the query results (or part of the query results).

Report 5: Reports with moving and cumulative aggregates

Produce one report containing moving and cumulative aggregates.

What are the total and cumulative monthly total sales of Gold membership in 2009?

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands that contains moving and cumulative aggregates, and

(*c*) The screenshots of the query results (or part of the query results).

Report 6: Reports with Partitions

Produce **one** report that contain partitions.

What are the city ranks by total number of incoming routes in each country?

(*a*) The query questions written in English,

(*b*) The SQL commands that contains partitions, and

(*c*) The screenshots of the query results (or part of the query results).

Report 7: City-to-City Routes' Report

Produce the following report (Note that the figures in the sample reports below are not accurate. However, your reports must contain all the columns shown in the sample reports below).

Departure City	Departure Country	Arrival City	Arrival Country	Number of Routes	Average Distance (km)
Melbourne	Australia	Singapore	Singapore	10	10000
Sydney	Any Country	Melbourne	Australia	3	400
Any City	Australia	Any City	Malaysia	20	9987
Any City	Any Country	Any City	Any Country	9999	99879
……	……	……	……		

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands, and

(*c*) The screenshots of the query results (or part of the query results).

Report 8: Passengers' Report

Produce the following report (Note that the figures in the sample reports below are not accurate. However, your reports must contain all the columns shown in the sample reports below).

Nationality	Passenger Type	Source Country	Destination Country	Average Passenger Age (Integer)
Australian	Adult	Australia	Singapore	25
Australian	Elder	Italy	Any	35
Australian	Children	Any	Any	23
French	Any	Any	Any	40
……	……		……	……

Passenger type: Children (Age<11), Teenager (11<=Age<=17), Adult (18<=Age<=60), Elder (>60) instead

The outputs of this task are:

(*a*) The query questions written in English,

(*b*) The SQL commands, and

(*c*) The screenshots of the query results (or part of the query results).

D. Submission Checklist

1. One **combined pdf file** containing all tasks mentioned above:
   - ☐ Cover page
   - ☐ A signed coversheet
   - ☐ Details of your ORACLE accounts
   - ☐ A contribution declaration form:

     Each student must state the parts of the assignment that he/she did. An example is as follows:

     Percentage of contribution:
     1. Name: Adam, ID: 210008, Contribution: 60%
     2. Name: Ben, ID: 230933, Contribution: 40%

     List of parts that each student did:
     1. Adam: list the parts that Adam did
     2. Ben: list the parts that Ben did

   - ☐ Task C.1 (outputs *a, b, c, d, e, f*)
   - ☐ Task C.2 (outputs *a, b, c*)
   - ☐ Task C.3 Report 1 (outputs *a, b, c*)
   - ☐ Task C.3 Report 2 (outputs *a, b, c*)
   - ☐ Task C.4 Report 3: Transactions' report (outputs *a, b, c*)

- ☐ Task C.4 Report 4: Report with proper sub-totals (outputs *a, b, c*)
- ☐ Task C.4 Report 5: Report with moving and cumulative aggregates (outputs *a, b, c*)
- ☐ Task C.4 Report 6: Report with partition (outputs *a, b, c*)
- ☐ Task C.4 Report 7: City-to-city routes' report (outputs *a, b, c*)
- ☐ Task C.4 Report 8: Passengers' report (outputs *a, b, c*)

2. **txt files** from the following tasks:
   - ☐ Task C.1 (SQL command as required by output *b*)
   - ☐ Task C.2 Implement Star Schemas (SQL command as required by output a and b)
   - ☐ Task C.3 Basic Reports (SQL command as required by output *b*)
   - ☐ Task C.4 Report 3: Transactions' report (SQL command as required by output *b*)
   - ☐ Task C.4 Report 4: Report with proper sub-totals (SQL command as required by output *b*)
   - ☐ Task C.4 Report 5: Report with moving and cumulative aggregates (SQL command as required by output *b*)
   - ☐ Task C.4 Report 6: Report with partition (SQL command as required by output *b*)
   - ☐ Task C.4 Report 7: City-to-city routes' report (SQL command as required by output *b*)
   - ☐ Task C.4 Report 8: Passengers' report (SQL command as required by output *b*)

   All of the above txt files must be run-able in Oracle.

3. Zip all the files above (pdf from #1 above, and txt files from #2 above), and upload this zip file to Moodle.

THE END