FIT3003 Major Assignment - Sem 2 2022 (Weight = 20%)
Due date: Week 11, Monday 10-October-2022, 11:55pm

Version: 1.0 – 26/08/2022

Learning Outcomes:

LO1. Design multidimensional databases and data warehouses.
LO2. Use fact and dimensional modelling.
LO3. Implement online analytical processing (OLAP) queries.
LO4. Explain the roles of data warehousing architecture and the concepts of granularity in data warehousing.
LO5. Create business intelligence reports using data warehouses and OLAP.

A. General Information and Submission

- This is a group assignment. One group consists of **two students from the same lab**. You need to register your group composition through the FIT3003 Group Selection Form as soon as possible.
- *Submission method*: Submission is online through Moodle.
- *Penalty for late submission*: 10% deduction for each day (applies for the whole group).
- *Assignment coversheet*: You will need to sign the assignment coversheet.
- *Contribution form*: The contribution form needs to be completed by all members and signed (e-signature is acceptable) as an agreement between members. The contribution declaration template is shown on the Major Assignment FAQ page on the EdStem forum.
- *Assignment FAQ*: There is a Major Assignment FAQ page set up on the EdStem forum.

B. Problem Description

MonCity is one of the largest smart cities in the world, and was developed by Monash University for education and research purposes. In MonCity, there are four different zone areas (ZoneA, ZoneB, ZoneC, ZoneD). A wide range of self-driving cars is provided for Monash students and staff members to travel between different zone areas or within the same zone area.

MonCity has an existing operational database that maintains and stores all of the self-driving car-related information, such as the booking, maintenance, and accident records required for management's daily operation. However, in order to improve work efficiency, management at MonCity has decided to hire your team of data warehouse engineers to design and develop a data warehouse that can quickly generate reports based on their needs. Management at MonCity wants to generate reports to keep track of the **bookings, accidents** and **maintenance information**, such as calculating statistics of booking records and accident records, which can be used for

self-driving car analyses later. For the accident details, they are particularly interested in the **number of accidents per self-driving car** (e.g., the number of accidents caused by Car01).

MonCity's operational database tables can be found in the **MonCity** account. You can, for example, execute the following query:

Select * From MonCity.<table_name>;

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

Table Name	Attributes and Data Types		Notes
FACULTY	FACULTYID	VARCHAR	This table stores faculty information. One faculty can have multiple passengers.
	FACULTYNAME	VARCHAR	
	ZONE	VARCHAR	
RESEARCHCENTER	CENTERID	VARCHAR	This table stores information about research centres. One research centre may have several maintenance teams.
	CENTERNAME	VARCHAR	
	PHONE	NUMBER	
	OPENINGHOUR	VARCHAR	
PASSENGER	PASSENGERID	VARCHAR	This table stores passenger information. Each passenger is from a certain faculty and can have one or multiple booking records.
	PASSENGERNAME	VARCHAR	
	PASSENGERROLE	VARCHAR	
	PASSENGERGENDER	VARCHAR	
	PASSENGERAGE	NUMBER	
	FACULTYID	VARCHAR	
ERROR	ERRORCODE	VARCHAR	This table stores accident error information. One error code may involve more than one accident.
	ERRORMESSAGE	VARCHAR	

MAINTENANCETYPE	MAINTENANCETYPE	VARCHAR	This table stores maintenance type information. One maintenance type may involve more than one maintenance record.
	MAINTENANCEDESCRIPTION	VARCHAR	
BOOKING	BOOKINGID	VARCHAR	This table stores the information of booking records. One booking record belongs to a certain passenger and car.
	REGISTRATIONNO	VARCHAR	
	BOOKINGDATE	DATE	
	DEPARTUREZONE	VARCHAR	
	DESTINATIONZONE	VARCHAR	
	PASSENGERID	VARCHAR	
CAR	REGISTRATIONNO	VARCHAR	This table stores the information of self-driving cars. One car may have more than one booking, one accident, and one maintenance record.
	CARMODEL	VARCHAR	
	MANUFACTURINGYEAR	NUMBER	
	CARBODYTYPE	VARCHAR	
	NUMSEATS	NUMBER	
MAINTENANCE	MAINTENANCEID	VARCHAR	This table stores each car's maintenance history. Each maintenance record belongs to a certain car, maintenance type and maintenance team.
	REGISTRATIONNO	VARCHAR	
	MAINTENANCEDATE	DATE	
	MAINTENANCETYPE	VARCHAR	
	MAINTENANCECOST	NUMBER	
	TEAMID	VARCHAR	
ACCIDENTINFO	ACCIDENTID	VARCHAR	This table stores the information of each accident, including accident ID, accident zone, etc. Each accident can be categorised into a certain error code.
	ACCIDENTZONE	VARCHAR	
	CAR_DAMAGE_SEVERITY	VARCHAR	

	ERRORCODE	VARCHAR	
CARACCIDENT	REGISTRATIONNO	VARCHAR	This table stores information about accidents and all related cars. One accident can involve more than one car.
	ACCIDENTID	VARCHAR	
MAINTENANCETEAM	TEAMID	VARCHAR	This table stores information about maintenance teams. One maintenance team may have more than one maintenance record. Each maintenance team may be from more than one research center.
	TEAMLEADER	VARCHAR	
	PHONE	NUMBER	
BELONGTO	TEAMID	VARCHAR	This table stores the relationship between research centres and maintenance teams.
	CENTERID	VARCHAR	

C. Tasks

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

1. **Design a data warehouse for the above MonCity database.**
   You are required to create a data warehouse for the **MonCity** database. The management is interested in the following fact measures:
   - Number of booking records
   - Total maintenance cost
   - Number of maintenance records
   - Number of accident records

   The following are some possible dimension attributes that you may need in your data warehouse:
   - Booking time period: by month
   - Faculty
   - Passenger age group: Young adults (18-35 years old); middle-aged adults (36-59 years old); old-aged adults (over 60 years old)

- Car body type (including number of seats)
- Research centre
- Maintenance type
- Accident Zone (based on accident zone in the ACCIDENTINFO table)
- Error code
- Car damage severity

For each attribute, you may apply your own design decisions on specifying a range or group where this is not already detailed above, but make sure to specify them in your submission.

- **Preparation stage.**

  Before you start designing the data warehouse, you have to ensure that you have explored the operational database and have done sufficient data cleaning. Once you have done the data cleaning process, you are required 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 process, explain the strategies you used in this process (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 may contain multi-fact(s). You need to identify the fact measures, dimensions, and attributes of the star/snowflake schema. The following queries might help you to identify the fact measures and dimensions:

  - How many booking records were made by passengers from the IT faculty in July by using bus?
  - How many booking records were made by each passenger age group?
  - How many maintenance records were listed that include at least one maintenance team from research centre 4 (CE04)?
  - List the total maintenance cost of each maintenance type for mini bus.
  - How many accidents were recorded in the ZoneA for Car01?
  - List the number of accidents for each error code at ZoneB for Car06.
  - List the number of accidents with severe damage inflicted on Car05.

FACT:
- Booking records
- Maintenance records
- Maintenance cost
- # of Accident

DIM:
Record By PASSENGER AGE
Record By RESERACHCENTRE
Cost By Maintainence type for minibus
# of Acc By ACCIDENT ZONE (Acc table)

You should pay attention to the granularity of your fact tables. You are required to create **two versions** of star/snowflake schemas based on different levels of aggregation. Version-1 should be in the highest level of aggregation. Version-2 should be in level 0, which means it contains no aggregation.

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

The star/snowflake schema of both versions you created might contain **bridge tables, determinant dimensions, and/or temporal dimensions**. If you are using bridge tables, make sure to include the weight factor and list aggregate attributes. If you are using determinant dimensions, make sure the correct notation has been used (broken lines for the table boundary) and you must provide the reasons for your choice(s). You can use different temporal data warehousing techniques for the temporal dimensions. If there are any, you must provide the reasons for your choice(s).

The outputs of this task are:

c) Two versions of star/snowflake schema diagrams.

d) 1. The reasons for the choice of determinant dimension(s) in your star schema, or the reason for its absence.

2. The reasons for the choice of SCD type(s) for any temporal dimensions in your star schema, or the reason for its absence.

e) An explanation of the differences between the two versions of star/snowflake schemas.

Note: *The above explanation must be consistent with your star schema diagram and based on the assignment scenario. Please have a maximum of 300 words for each explanation.*

2. Implement the two versions of the star/snowflake schemas using SQL.

You are required to implement the star/snowflake schemas for the two versions that you have drawn in Task 1. That is, you may need to create the different fact and dimension tables for the two versions in SQL, and populate these tables accordingly.

When naming the fact tables and dimension tables, you are required to give identical names for the two versions and suffix the version number to differentiate them. For example, you can use “MonCity_fact_v1” for version-1 and “MonCity_fact_v2” for

version-2. If the dimension is the same between the two versions, you do not need to create them twice.

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.

Note:

- If your account is full, you will need to drop all of the tables that you have previously created during labs.
- If you have dopped all tables in you account and you still encounter the `ORA-01536: space quota exceeded for tablesace 'TABLE_NAME'`, please check your SQL code whether you have properly joined all tables. This issue was mainly caused when you did not do the table join properly as the number of records multiplied during the process.

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.

***Note:** The SQL statements for both levels of star schema must be presented in the PDF file.*

3. Create the following reports using OLAP queries.

You are required to generate several reports using the **data warehouse version-1 (high aggregation)** that you have implemented in Task 2. For each report, you should produce the SQL command and sample report output.

Note: the table snapshots of the following questions are only for reference purposes.

3.1. OLAP Queries

REPORT 1: MonCity's cumulative number of booking records of each month for Faculty of IT

Create an OLAP SQL command that will produce the following report (including the same output order):

FacultyID	Month	Total bookings	Cumulative number of booking records
FIT	January	260	260
FIT	February	230	490
FIT	March	234	724
FIT	April	228	952
FIT	May	245	1,197
FIT	June	252	1,449
FIT	July	249	1,698
FIT	August	245	1,943
FIT	September	274	2,217
FIT	October	256	2,473
FIT	November	251	2,724
FIT	December	251	2,975

The outputs of this task are:

(a) The SQL command, and

(b) The screenshot of the query results (or part of the query results), including all attribute names.

REPORT 2: MonCity's maintenance report

Create an OLAP SQL command that will produce the following report (including the same output order):

Team ID	Car body type	Total number of maintenance	Total maintenance cost
All Teams	All Car Body Types	399	125,300
All Teams	Bus	136	44,900
All Teams	Mini Bus	113	34,000

All Teams	People Mover	150	46,400
T002	All Car Body Types	197	62,700
T002	Bus	58	18,400
T002	Mini Bus	62	19,300
T002	People Mover	77	25,000
T003	All Car Body Types	202	62,600
T003	Bus	78	26,500
T003	Mini Bus	51	14,700
T003	People Mover	73	21,400

The outputs of this task are:

(a) The SQL command, and

(b) The screenshot of the query results (or part of the query results), including all attribute names.

REPORT 3: MonCity's rank analysis for the number of accidents

Create an OLAP SQL command that will produce the following report (including the same output order):

Error Code	Registration No.	Car body type	Total number of accidents	Rank
Error001	Car01	Bus	13	1
Error001	Car12	Mini Bus	12	2
Error001	Car19	Mini Bus	12	2
Error001	Car04	Bus	12	2
Error001	Car08	Bus	11	3
Error001	Car20	Mini Bus	11	3
Error002	Car22	People Mover	45	1
Error002	Car27	People Mover	42	2
Error002	Car23	People Mover	39	3
Error002	Car30	People Mover	39	3
Error003	Car06	Bus	12	1
Error003	Car14	Mini Bus	12	1
Error003	Car10	Bus	11	2
Error003	Car01	Bus	11	2
Error003	Car12	Mini Bus	10	3
Error003	Car09	Bus	10	3

Error004	Car12	Mini Bus	13	1
Error004	Car15	Mini Bus	10	2
Error004	Car20	Mini Bus	9	3
Error004	Car18	Mini Bus	9	3
Error004	Car04	Bus	9	3
Error005	Car16	Mini Bus	11	1
Error005	Car08	Bus	10	2
Error005	Car20	Mini Bus	10	2
Error005	Car19	Mini Bus	9	3
Error005	Car05	Bus	9	3
Error005	Car01	Bus	9	3
Error005	Car12	Mini Bus	9	3

The outputs of this task are:

(a) The SQL command, and

(b) The screenshot of the query results (or part of the query results), including all attribute names.

REPORT 4: MonCity's booking report

Create an OLAP SQL command that will produce the following report (including the same output order):

Car body type	Age group	Faculty ID	Total number of bookings
People Mover	All Age Groups	All Faculties	3,396
People Mover	All Age Groups	ART	453
People Mover	All Age Groups	BUS	314
People Mover	All Age Groups	ENG	841
People Mover	All Age Groups	FIT	1,009
People Mover	All Age Groups	SCI	779
People Mover	Group1	All Faculties	1,380
People Mover	Group1	ART	169
People Mover	Group1	BUS	121
People Mover	Group1	ENG	382
People Mover	Group1	FIT	390
People Mover	Group1	SCI	318
People Mover	Group2	All Faculties	1,722
People Mover	Group2	ART	284

People Mover	Group2	BUS	193
People Mover	Group2	ENG	429
People Mover	Group2	FIT	497
People Mover	Group2	SCI	319
People Mover	Group3	All Faculties	294
People Mover	Group3	ENG	30
People Mover	Group3	FIT	122

The outputs of this task are:

(a) The SQL command, and

(b) The screenshot of the query results (or part of the query results), including all attribute names.

3.2. Reports with rollup and partial rollup

Produce **two** reports that contain subtotals and one fact measure, using rollup and partial rollup.

REPORT 5: Produce one booking-related report that is useful for management that uses rollup.

REPORT 6: Produce one booking-related report that is useful for management that uses partial rollup.

The outputs of this task are:

(a) The query questions written in English,

(b) An explanation of the differences between rollup and partial rollup,

(c) The SQL commands that contain rollup and partial rollup, and

(d) The screenshots of the query results (or part of the query results).

3.3. Report with moving and cumulative aggregates

Produce **two** reports containing moving and cumulative aggregates and one fact measure.

REPORT 7: Produce one moving aggregate report that relates to the Booking information.

The report must contain or use the month information and number of Bookings in the output.

REPORT 8: Produce one cumulative aggregate report that relates to the maintenance information.

The report must contain or use the number of maintenance records or the total maintenance cost in the output.

The outputs of this task are:

(a) The query questions written in English,

(b) Your explanation of why such a query is necessary or valuable for management,

(c) The SQL commands that contain moving and cumulative aggregates, and

(d) The screenshots of the query results (or part of the query results).

4. Business Intelligence (BI) Reports

Choose any **four** reports from Task 3 and change the presentation of these reports in a BI report format. Create **one** dashboard based on your chosen reports. This new representation should be appealing to management. Additionally, in these new reports, you might want to include some selection buttons, which may give users options on what criteria to include so that the graph report will be more dynamic.

R2; R4 - my chart; R5 stack barchart;

5. Final Recommendations/Suggestions

Due to the successful operation of the MonCity project, Monash University decided to invest further in the self-driving car project. As a data warehouse engineer:

- You are required to provide **a suggestion with valid supporting data** to help management decide on how to improve current self-driving car projects, such as which car body type should be further invested in based on the Booking, maintenance, and accident records, or which error code has contributed to the most number of accidents, etc.; or
- If you have a different opinion on the self-driving car project, provide a reason why the project is not worth further investment.

Note: The above suggestion must contain **valid supporting data from the MonCity database and data warehouse**. This could be tables, graphs, or charts. Please use a maximum of 500 words for your suggestions.

D. Checkpoints

Checkpoint	Weight	Assessment	Due date
Checkpoint 1	1%	ER Diagram Group contribution check	Week 7 (during lab)
Checkpoint 2	1%	Star Schema Version-1 Group contribution check	Week 9 (during lab)

The Checkpoints will only be assessed during your allocated lab. Your group is required to complete the assessment for each checkpoint in order to obtain the allocated mark. There are associated mark penalties for **not** meeting the checkpoint assessment on time to a satisfactory state.

The member contribution will be checked by your allocated tutor regularly. Your contribution will be based on team member reviews if you have been absent from one of the checkpoint assessments, and the Major Assignment grade will be adjusted accordingly.

Note that the Final Report and Implementation are worth 18%.

E. Submission Checklist

1. One **combined .pdf file** containing all tasks mentioned above:
    - ☐ Cover page
    - ☐ A signed coversheet
    - ☐ A contribution declaration form:
        - ☐ Each student must state the parts of the assignment that they completed. 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 completed:
          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)
    - ☐ Task C.2 (outputs a, b)
    - ☐ Task C.3 Reports 1-4 (outputs a, b)
    - ☐ Task C.3 Reports 5-8 (outputs a, b, c, d)
    - ☐ Task C.4 (screenshot of a dashboard with four reports)

2. **.sql files** for the following task:
    - ☐ Task C.1 (SQL command as required by output b)
    - ☐ Task C.2 Implementation of the Star Schemas (SQL command as required by output a and b)
    - ☐ Task C.3 Reports 1-4 (SQL command as required by output a)
    - ☐ Task C.3 Reports 5-8 (SQL command as required by output c)

    All of the above SQL files must be runnable in Oracle. Zero marks will be given for the implementation if:
    - SQL files are not runnable, or
    - SQL files are missing, or
    - Wrong SQL files are submitted, including showing inconsistent SQL statements between pdf and SQL files.

3. **.pbix file** for the following task:
    - ☐ Task C.4 (a dashboard with four reports)

4. Zip all the SQL files from #2 and the file from #3, and name the ZIP folder as **MA_SQL_BI.zip**.

Submission Method:

1. Upload the **PDF file** from Checklist #1 and the **ZIP file** from Checklist #4 to Moodle by the due date: **Monday, 10 October 2022, 11:55pm.**
    - The submission of this assignment must be in the form of **a single PDF file AND a single ZIP file**. No other forms will be accepted.
    - One member of your group can upload the submission. However, please note that **all group members must click the submit button and accept the submission statement** (failure to do so will cause your assignment submission to be in draft mode and will incur late penalties).
    - You must ensure that you have all the files listed in this checklist before submitting your assignment to Moodle. Failure to submit a complete list of files will lead to mark penalties.

2. Penalty for late submission: 10% deduction for each day, including weekends

3. Submission cut-off time: **Monday, 17 October 2022, 11:55 pm**. The submission link will be unavailable after this time.

Getting help and support:

What can you get help for?

- ***Consultations with the Teaching Team***
  Talk to the Teaching Team:
  https://lms.monash.edu/course/view.php?id=140811&section=2
- ***English language skills***
  Talk to English Connect: https://www.monash.edu/english-connect
- ***Study skills***
  Talk to a learning skills advisor: https://www.monash.edu/library/skills/contacts
- ***Counselling***
  Talk to a counsellor: https://www.monash.edu/health/counselling/appointments

Extensions:

If you are experiencing difficulties that you think will impact your ability to meet this deadline, you may apply for an assignment extension. You must apply **no later than two university working days after the due date** of this assignment (before Wednesday, 19 October 2022, 11:55 PM).

The extension application can be found on *Moodle > Assessments > How to Apply for an Extension*. Please allow **two business days** for your application to be processed.

Please ensure your application is supported by appropriate documentation. You can find more information about assignment extensions at the Special Consideration website.

Special Consideration:

Students should carefully read the Special Consideration website, especially the details about the formal documentation required.

All special consideration requests should be made using the Special Consideration Application.

Please do not assume that submission of a Special Consideration application guarantees that it will be granted – you must receive an official confirmation that it has been granted.

Late Penalty:

Late assignments submitted without an approved extension may be accepted up to a maximum of **seven days** with the approval of the Chief Examiner and/or Lecturer but will be **penalised at the rate of 10% per day (including weekends and public holidays)**. Assignments submitted more than seven days after the due date will receive a zero mark for that assignment and may **not receive any feedback**.

Please note (late penalties and extensions):

1. An inability to manage your time or computing resources will not be accepted as a valid excuse. (Several assignments being due at the same time are a fact of university life.)
2. Group issues, hardware failures, whether of personal or university equipment, are not normally recognised as valid excuses. Failure to back up assignment files is also not recognised as a valid excuse.

Plagiarism and Collusion:

Monash University is committed to upholding standards and academic integrity and honesty. Please take the time to view these links.

Academic Integrity Module

Student Academic Integrity Policy

Test your knowledge, collusion (FIT No Collusion Module)

END OF MAJOR ASSIGNMENT