

FIT5195 Major Assignment - Sem 1/2020 (Weight = 25%)**Due date: ~~Week 11, Friday 5-Jun-2020,~~****Week 12, Wednesday, 10-Jun-2020, 11:55pm****A. General Information and Submission**

- This is a group assignment. One group consists of 2-3 students. You need to register your group composition through the [form](#) 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
- *Contribution Form*: The contribution needs to be completed by all members and please sign the form as an agreement between members.

B. Problem Description – Monash Real Estate (MonRE) Data Warehouse

Monash Real Estate (MonRE) is an agency run by Monash University, that helps individuals within the University's community (such as the university's staff and students) to find real estate properties around Australia. The management at MonRE frequently generates reports to keep track of the real estate property related information, e.g. calculating statistics of rents and sales automatically which can later be used for forecasting various trends and making predictions about the real estate.

At present, MonRE has an existing operational database which maintains and stores all of the real estate related information (such as rents, sales, properties etc.) required for the agency's daily operation. However, since the staff at MonRE have limited database knowledge and the operational database is quite large, MonRE has decided to hire your team of Data Warehouse Engineers to design, develop and quickly generate reports from a Data Warehouse.

MonRE's operational database tables can be found at **MonRE**. You can, for example, execute the following query:

```
select * from MonRE.<table_name>;
```

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

| Table Name (PK/FK) | Attributes and Data Types | | Notes |
|--------------------|---------------------------|---------|---|
| Address | ADDRESS_ID | NUMERIC | This table stores the property address information. It contains |
| | STREET | VARCHAR | |
| | SUBURB | VARCHAR | |
| | POSTCODE | NUMBER | |

| | | | |
|----------------------|---------------------|---------|--|
| | | | around 13,000 addresses in Australia. |
| Advertisement | ADVERT_ID | NUMERIC | This table stores the information (id and name) of advertisements. |
| | ADVERT_NAME | VARCHAR | |
| Agent | PERSON_ID | NUMERIC | This table stores the information of around 2,000 agents (according to the agent ID) and the agent's salary. |
| | SALARY | NUMERIC | |
| Agent_Office | PERSON_ID | NUMERIC | This table stores the information of which agent belongs to which agency office. |
| | OFFICE_ID | NUMERIC | |
| Client | PERSON_ID | NUMERIC | A client in the client table is a person who either has bought/rented/inspected (i.e visited)/created a wishlist for one or more properties. Currently, there are approximately over 3000 entries in this table including the client's person id, the maximum budget and minimum budget that the client is willing to spend. |
| | MIN_BUDGET | NUMERIC | |
| | MAX_BUDGET | NUMERIC | |
| Client_Wish | FEATURE_CODE | NUMERIC | This table stores the client's wishlist details for some certain features that clients wish to have in their interested properties. |
| | PERSON_ID | NUMERIC | |
| Feature | FEATURE_CODE | NUMERIC | This table stores approximately over 700 features of the properties e.g. Ensuite, Dishwasher etc. The id and description of the features are stored in this table. |
| | FEATURE_DESCRIPTION | VARCHAR | |
| Office | OFFICE_ID | NUMERIC | The office table stores the details (id and name) of the agency's office. There are over 1000 offices. |
| | OFFICE_NAME | VARCHAR | |
| Person | PERSON_ID | NUMERIC | The person table has almost 7000 entries and each entry contains the details such as id, name, gender, contact details etc. about all individuals associated with |
| | TITLE | VARCHAR | |
| | FIRST_NAME | VARCHAR | |
| | LAST_NAME | VARCHAR | |
| | GENDER | VARCHAR | |

| | | | |
|-------------------------|--------------------------|---------|---|
| | ADDRESS_ID | NUMERIC | MonRE. A person in the person table could be a client or agent. |
| | PHONE_NO | VARCHAR | |
| | EMAIL | VARCHAR | |
| Postcode | POSTCODE | NUMERIC | The postcode table has over 600 entries and contains the postcode information such as the postcode and the state (e.g. 3145 VIC) |
| | STATE_CODE | CHAR | |
| Property | PROPERTY_ID | NUMERIC | The property table stores various details (such as id, address, type etc) about the properties. The property date added is the date in which the property was added to MonRE's database. There are over 6000 entries in the property table. |
| | PROPERTY_DATE_ADDED | DATE | |
| | ADDRESS_ID | NUMERIC | |
| | PROPERTY_TYPE | VARCHAR | |
| | PROPERTY_NO_OF_BEDROOMS | NUMERIC | |
| | PROPERTY_NO_OF_BATHROOMS | NUMERIC | |
| | PROPERTY_NO_OF_GARAGES | NUMERIC | |
| | PROPERTY_SIZE | NUMERIC | |
| | PROPERTY_DESCRIPTION | VARCHAR | |
| Property_Advert | PROPERTY_ID | NUMERIC | The property_advert table stores information about advertisements made by agents about properties. Each advertisement for a property lists the cost for either renting or buying the property. |
| | ADVERT_ID | NUMERIC | |
| | AGENT_PERSON_ID | NUMERIC | |
| | COST | NUMERIC | |
| Property_Feature | PROPERTY_ID | NUMERIC | This table records which feature belongs to which property. |
| | FEATURE_CODE | NUMERIC | |
| Rent | RENT_ID | NUMERIC | This table records details of rents (such as property, client, agent, price etc.) |
| | AGENT_PERSON_ID | NUMERIC | |
| | CLIENT_PERSON_ID | NUMERIC | |
| | PROPERTY_ID | NUMERIC | |
| | RENT_START_DATE | DATE | |
| | RENT_END_DATE | DATE | |
| | PRICE | NUMERIC | |
| Sale | SALE_ID | NUMERIC | This table records details of sales (such as property, client, agent, price etc.) |
| | AGENT_PERSON_ID | NUMERIC | |
| | CLIENT_PERSON_ID | NUMERIC | |

| | | | |
|--------------|------------------|---------|---|
| | SALE_DATE | DATE | |
| | PROPERTY_ID | NUMERIC | |
| | PRICE | NUMERIC | |
| State | STATE_CODE | CHAR | The state table contains the code and name of the states. |
| | STATE_NAME | VARCHAR | |
| Visit | CLIENT_PERSON_ID | NUMERIC | This table records which details of clients visiting properties during inspections and the agent in charge of the inspection. |
| | AGENT_PERSON_ID | NUMERIC | |
| | PROPERTY_ID | NUMERIC | |
| | VISIT_DATE | DATE | |
| | DURATION | NUMERIC | |

C. Tasks

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

1. Design a data warehouse for the above MonRE database.

You are required to create a data warehouse for the MonRE database.

The management is especially interested in the following fact measures:

- Average Rental Fees
- Average Agent Earning
- Average Sales
- Total Number of Rent
- Total Number of Sale
- Total Number of Agents
- Total Number of Clients
- Total Number of Properties
- Average Number of Property Visit

The following show some possible dimension attributes that you may need in your data warehouse:

- Property features
- Property type (house, apartment, etc.)
- Property sales/rental time according to season
- Property location

- Maximum client budget range for buying/renting properties (Low [0 to 1000], Medium [1001 to 100000], High [100001 to 10000000])
- Rental period (Short: < 6 months, Medium: 6-12 months, Long: > 12 months)
- Agent/Client's gender
- Visit date (Day[Sat, Sun, ... Fri], Month[Jan-Dec], Year)
- Client Wishlist
- Agent salary
- Agent office size (Small: < 4 employees, Medium: 4-12 employees, Big: > 12 employees)
- Property scale (Extra Small: <= 1 bedroom, Small: 2-3 bedrooms, Medium: 3-6 bedrooms, Large: 6-10 bedrooms, Extra Large: > 10 bedrooms)
- Property feature category (Very basic: < 10 features, Standard: 10-20 features, Luxurious: > 20 features)

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 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/snowflake schema for this data warehouse contains multi-facts. 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:

- What is the average rental fee of apartments around South Yarra, VIC in 2019?
- What is the average earning for all Ray White agents?
- What is the average sales for houses in VIC compared to NSW?
- Who are the top 3 agents in Melbourne?
- What is the total number of rent for clients who stay in small scale properties with very basic features?
- What is the total number of sales for Townhouses with Air Conditioning and Security?

- What is the total number of female agents who work in a medium agent office?
- What is the total number of clients with a high budget?
- What is the total number of properties being advertised for sale in April 2020?
- What is the average number of property visits during summer?
- Which day is the most popular visit day?

You should pay attention to the granularity of your fact tables. You are required to create **two versions** of star/snowflake based on different levels of aggregation.

The two versions of the star/snowflake represent different levels of aggregation. Version-1 should be in the highest level of aggregation. Version-2 should be in level 0, which means no aggregation.

| Version Name | Level |
|--------------|---------------------------------|
| Version-1 | High aggregation (e.g. Level 2) |
| Version-2 | No aggregation (Level 0) |

The star/snowflake schema of both versions you created should contain **Bridge Table** and **Temporal**. You can choose to use **Hierarchy or Non-Hierarchy**, but you need to provide the reason why you use Hierarchy or Non-Hierarchy. If there is any **Determinant Dimension**, you are required to denote the Determinant Dimension clearly with a “*” besides your dimension’s name (e.g. “*ABC_DIM”). You can use different temporal data warehousing techniques for the temporal dimension and provide the reasons of your choice.

The outputs of this task are:

- Two versions of star/snowflake schema diagrams,
- A short explanation of why you chose hierarchy or non-hierarchy,
- The reasons of the choice of SCD type for temporal dimension,
- A short explanation of the difference among the two versions of star/snowflake schema.

2. Implement the **two versions** 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, “MonRE_fact_v1” for version-1 and “MonRE_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) Screenshots 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. Create the following reports using OLAP queries.

You are required to generate the reports using both data warehouse versions, **version-1** (e.g. **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.

a. *Simple reports:*

Produce **three** reports. Each report contains two attributes from two different dimensions, and one fact measurement.

For the report itself, the first report must be about **Top k**, the second report is **Top n%**, and the third report is **Show All**.

The outputs of this task are:

- (a) The query questions written in English,
- (b) Your explanation on why such a query is necessary or useful for the management,
- (c) The SQL commands, and
- (d) The screenshots of the query results (or part of the query results), including all attribute names.

b. *Reports with proper sub-totals:*

Produce **four** reports. These reports must include sub-totals, using the Cube or Roll-up or Partial Cube/Roll-up operators.

REPORT 4 and REPORT 5: What are the sub-total and total rental fees from each suburb, time period, and property type? (You must use the Cube and Partial Cube operator)

REPORT 6 and REPORT 7: Produce 2 other sub-totals reports that are useful for management using Roll-up and Partial Roll-up.

The outputs of this task are:

- (a) The query questions written in English,
- (b) Your explanation on why such a query is necessary or useful for the management,
- (c) The SQL commands that include sub-totals, using the Cube or Roll-up or Partial Cube/Roll-up operators, and
- (d) The screenshots of the query results (or part of the query results).

c. *Reports with moving and cumulative aggregates:*

Produce **three** reports containing moving and cumulative aggregates.

REPORT 8: What is the total number of clients and cumulative number of clients with a high budget in each year?

REPORT 9 and REPORT 10: Produce 2 other moving/cumulative aggregate reports that are useful for management.

The outputs of this task are:

- (a) The query questions written in English,
- (b) Your explanation on why such a query is necessary or useful for the management,
- (c) The SQL commands that contains moving and cumulative aggregates, and
- (d) The screenshots of the query results (or part of the query results).

d. *Reports with Partitions:*

Produce two reports that contain partitions.

REPORT 11: Show ranking of each property type based on the yearly total number of sales and the ranking of each state based on the yearly total number of sales.

REPORT 12: Produce another partitioning report that is useful for management.

The outputs of this task are:

- (a) The query questions written in English,
- (b) Your explanation on why such a query is necessary or useful for the management,
- (c) The SQL commands that contains partitions, and
- (d) The screenshots of the query results (or part of the query results).

4. Business Intelligence (BI) Reports.

Choose any **five** reports from Task 3, and change the presentation of these reports by representing these in a graph format. This new presentation should be more appealing to the management. You can use any graph software to show the graph reports. Additionally, in these new reports, you might want to include some selection buttons (for illustrative purposes), which may give users options on what criteria to choose, so that the graph report will be more dynamic.

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 Simple Reports (outputs *a, b, c, d*)
- ☐ Task C.3 More Reports with Sub-Totals (outputs *a, b, c, d*)
- ☐ Task C.3 Reports with Moving and Cumulative Aggregates (outputs *a, b, c, d*)
- ☐ Task C.3 Reports with Partitions (outputs *a, b, c, d*)
- ☐ Task C.4 (five graphs)

2. **.sql 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 Simple Reports (SQL command as required by output *c*)

- Task C.3 More Reports with Sub-Totals (SQL command as required by output c)
- Task C.3 Reports with Moving and Cumulative Aggregates (SQL command as required by output c)
- Task C.3 Reports with Partitions (SQL command as required by output c)

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

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

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.

THE END