> # FIT5137 Group Assignment - Sem 2/2020 (Weight: 20%)
> ## Due date: Week 8, Wednesday, 7-October-2020, 11:55pm

A. General Information and Submission

- o This is a group assignment. One group consists of **2 students** only.
- o *Submission method*: Submission is online through Moodle.
- o *Penalty for late submission*: 10% deduction for each day.
- o *Assignment Cover Sheet*: You will need to sign the assignment cover sheet.
- o *Contribution Form*: The contribution needs to be completed by all members and please sign (e-signature is acceptable) the form as an agreement between members.

B. Problem Description – MonashMRDB

MonR is Monash University's restaurant management team that manages all restaurants on all Monash campuses. In recent times, due to the popularity of Mexican Food, MonR has decided to analyze Mexican Food related data. To maintain authenticity, MonR has acquired some sample data from about Mexican restaurants (or places) and their Customers (or users) from the originating place, Mexico.

The sample data, consists of:
- ● userProfile.json
- ● placeProfiles.json
- ● openingHours.csv
- ● user_ratings.csv
- ● place_ratings.csv

Note: These data are raw data and do not follow any particular schema.

For the data management, MonR has hired your team of Advanced Database Experts to perform the following tasks.

C. Tasks

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

C.1. Creating the Databases and Importing Data.
Since MonR has heard about both MongoDB and Cassandra, therefore, they wish to use a combination of both technologies to analyse the data. They prefer the profile information related data for both users and places to be stored in MongoDB, and the reviews related data to be stored in Cassandra.

Task Requirement:

MongoDB (MongoDB Compass and Mongo Shell):

1. Using the **Mongo Shell** and appropriate **Mongo Shell** command(s), create a database called:
   FIT5137A1MRDB

2. Using the **Mongo Shell** and appropriate **Mongo Shell** command(s), create the following 2 collections:
   userProfiles
   placeProfiles

3. Using **MongoDB Compass** and appropriate MongoDB Data Types, import the data in
   a. **userProfile.json** into the **userProfiles** collection.
   b. **placeProfiles.json** into the **placeProfiles** collection.

4. In MongoDB, we understand that there are two data modelling methods, which are **embedding** and **referencing**.
   a. In your own words, explain what you understand by embedding and referencing data models.

   b. Import the data in **openingHours.csv** into your **FIT5137A1MRDB** database using appropriate data models.

   c. Include in your report, an explanation behind the selection of the data model.

Cassandra (Cassandra Shell):
5. Create a keyspace called **FIT5137A1_MRDB** for the Cassandra database, with SimpleStrategy and replication factor of 1.
6. Create the following column families using appropriate data types :
   a. user_ratings,
   b. place_ratings
7. Using the cassandra **COPY** command import the following data into the tables:
   a. **user_ratings.csv** into the **user_ratings** table.
   b. **place_ratings.csv** into the **place_ratings** table.

NOTE:
- For optimization purposes, you must not use any loops and/or additional variables (unless specified in the requirements) while performing this task.
- You must also take into account that more data may be added in the future.
- If required, you may create additional collection, data types, and/or tables. However, please ensure that you indicate those additional collection/data types/tables in your report.

Task Outputs:

❏ Task C.1.1. and C.1.2. MongoDB code and screenshots of results.
❏ Task C.1.3. screenshot from MongoDB Compass of at least one entire document after importing.
❏ Task C.1.4.
  ❏ C.1.4.a. and C.1.4.c. required explanations.
  ❏ C.1.4.b. screenshot(s) from MongoDB Compass of at least one entire document after importing both data files.
❏ Task C.1.5., Task C.1.6. and Task C.1.7. Cassandra commands and screenshots of results.

C.2. Modifying the Databases.

This task requires you to modify the appropriate database(s) for each question using Mongo Shell and Cassandra Shell. As part of the task you are required to identify which database (MongoDB or Cassandra) is required to be modified.

1. MonR has gained some new information about a trendy new place. Therefore, **without creating any new fields**, insert all of the information provided in Table 1.

Table 1 : New place information	
place id	70000
place name	Taco Jacks
street address	Carretera Central Sn
city	San Luis Potosi
state	SLP
country	Mexico
alcohol	No_Alcohol_Served
smoking area	not permitted
dress code	informal
accessibility	completely
price	medium
franchise	TRUE
area	open
other services	Internet
parking arrangements	none
accepted payment modes	any
cuisines	Mexican, Burgers
opening hours	Mon;Tue;Wed;Thu;Fri; 09:00-20:00
	Sat; Sun; 12:00-18:00

2. They have also realised that the user with user_id 1108, no longer prefers Fast_Food and also prefers to pay using debit_cards instead of cash. Therefore, **without looking up the existing values or adding any new fields,** update user 1108's favorite cuisines and favorite payment methods.

3. The management has realised that the user with user_id 1063 was an error. Therefore delete the user 1063 from the database.

4. To be consistent with the changes made in Task C.2 (1), (2), and (3), the management has also requested to update the reviews information to reflect the changes made to the users 1108 and remove the user 1063's reviews. They have looked up the data in the reviews table and have provided the information in table 2.

Table 2 : User ratings information	
user id	review id for reviews made by user
1063	137, 138, 139, 140, 141
1108	65, 66, 67, 68, 69, 70, 71, 72, 73, 74

5. It was also seen that user 1060 has reviewed Taco Jacks (ie. the new place with place id 70000), therefore using the information from table 3, insert the following data: (for this insert **only** you may look up the details of user 1060).

Table 3 : User 1060's rating for place 70000	
rating_id	7777
rating_place	2
food_rating	1
service_rating	2

Task Requirement:
- Create the necessary modifications given in Task C.2. in the appropriate databases.
- MongoDB Compass must not be used for Task C.2.
- Ensure that data which is not required to be modified is not modified in any way.

Task Outputs:
- ❏ Mongo Shell and Cassandra Shell code and screenshots of results for all questions in Task C2. To show that your database modifications were successful you may create queries and display the before and after changes.

C.3. Querying the Management.

This task requires you to query the appropriate database(s) for each question. As part of the task you are required to identify which database (MongoDB or Cassandra) is required to be modified.

Note: if required you may create indexes however, you must provide appropriate reasoning where any index/indices are created.

1. How many users are there in the database?
2. How many places are there in the database?
3. How many reviews were made in the database?
4. How many reviews were created by places having public parking arrangements?
5. Display the user id, rating of places and the user's personality for all users who are interested in technology, thrifty protector types of workers, with blue as their favorite colour and are casual drinkers.
   (Note that the number of rows returned should be greater than zero)
6. What are the place ids and ratings for food for places serving only Pizzeria cuisine? (Note the result should include cuisine).
7. Display all users who are students and prefer a medium budget restaurant.
8. Display all users who like Bakery cuisines and combine your output with all places having Bakery cuisines.
9. Display International restaurants that are open on Sunday.
10. Display the average place rating, average food rating, and average service rating for puesto de tacos. Show the average values in decimal points.
11. Display the average age according to each drinker level.
12. For each user whose favourite cuisine is Family, display the place ID, the place rating, the food rating and the user's budget.
13. What are the top 3 most popular ambiences (friends/ family/ solitary) for a single when going to a Japanese restaurant?
14. List the names of unique cuisines in the database.
15. Display all of the restaurants and indicate using a separate field/column whether the restaurant includes mexican cuisines. For instance, you can display if the restaurant serves mexican cuisine then the result should show the restaurant name followed by "serves mexican food" in the next field/column, or if the restaurant does not serves mexican cuisine then the result should show the restaurant name followed by "doesn't serves mexican food" in the next field/column.

You are required to provide **five additional queries** that you consider to be useful to the MonR management team's operation.

Task Requirement:

- Create at least **two indices** including compound indexes for queries that are frequently used and Justify why you have chosen the fields to be the index.
- Provide the necessary appropriate read operation code for each query. For your answer you may follow where needed the following format similar to that shown in the tutorials.

- When creating each query you have to take into consideration the efficiency of the query operation.

Task Outputs:

- ❏ Mongo Shell Aggregation and Cassandra Shell code and screenshots of results for all questions in Task C3.1. C3.1.11
- ❏ The explanation for Task C3.11.c

C.4. Summary Reports.

As a summary you are required to provide a comparison between the three databases i.e. Relational Database, Document-Oriented Database and Column-Oriented Database

Task Requirement:

- In your own words provide an explanation on how each type of database works.
- Provide a comparison in a tabular format with details on the main strengths and weaknesses of each database.
- Include a real-world example where each database is used.

MonR is happy with your work. However, they have to cut down their budget and they have decided to use only one database instead of two.

Task Requirement:

- Decide which database to use (either MongoDB or Cassandra).
- Provide a list of the steps (including an explanation and a flowchart) on how the merging process could be implemented.
- Provide a valid explanation on why you have selected one database over the other.

Note: You are NOT required to create the merged database, but your explanations must be valid for MonR.

Task Outputs:

- ❏ A report specifying the requirements, flow chart(s) and any references made from any sources. Note that this section does not require any code submissions.

C.5. Connecting to Drivers (Optional Bonus Section - up to 10 marks)

MonR has come to know that both MongoDB and Cassandra are NoSQL databases that can be incorporated into applications and be used as backend database stores. In order to incorporate MongoDB and Cassandra databases into applications (e.g. a Python application)

we can use drivers. For details about how to use the MongoDB and Cassandra drivers you can refer to the steps given in the driver tutorials (under Week 8 Tutorial Resources).

Therefore, as a bonus requirement MonR has requested you to create a runnable driver script incorporating the code from Task C1, C2 and C3. You may separate the code into two driver script files, i.e. one for MongoDB and one for Cassandra.

Task Requirement:

- Provide in your report a list of steps you have taken to connect the MongoDB and Cassandra drivers
- Provide two driver script files (i.e. one for MongoDB and another for Cassandra)

Note: This is a bonus section and therefore the marks awarded in this section are additional to what receive for the Task C1, C2, C3 and C4. If you choose not to attempt this section then you can still receive full marks based on Task C1, C2, C3 and C4. However, to attain full marks for Task C5, both of your scripts must be runnable in the MacOS/Windows terminal/command prompt.

Task Outputs:

- ❏ List of steps to take in order to connect MongoDB and Cassandra drivers.
- ❏ A script for MongoDB driver and another script for Cassandra driver.

D. Submission Checklist

1. One **combined pdf file** containing all tasks mentioned above:

   ☐ Cover page

   ☐ A signed cover sheet

   ☐ A contribution declaration form, e.g.:

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

   > Percentage of contribution:
   > I.   Name: Adam, ID: 210008, Contribution: 60%
   > II. Name: Ben, ID: 230933, Contribution: 40%

   > List of parts that each student did:
   > I.   Adam: list the parts that Adam did
   >   ☐ Task C.1 (tasks *1.1, 1.2,...*)
   >
   >   ☐ Task C.2 (tasks *2.1, 2.2,...*)
   >
   >   ☐ Task C.3 (tasks *3.1, 3.2,...*)
   >
   > II. Ben: list the parts that Ben did
   >   ☐ Task C.3 (tasks *3.11, 3.12...*)
   >
   >   ☐ Task C.4
   >
   >   ☐ Task C.5

   ☐ A **report** that combines all the tasks (including explanations, code and screenshots):

   ☐ Task C.1. (*MongoDB Tasks 1-4, Cassandra Tasks 5-7*)

   ☐ Task C.2. (*Tasks 1-5*)

   ☐ Task C.3. (*Tasks 1-15, additional queries 1-5*)

   ☐ Task C.4. (*An original report specifying the requirements, flow chart(s) and any references made from any sources. Note that this section does not require any code submissions*)

   ☐ [OPTIONAL] Task C.5. (*List of steps and scripts for MongoDB and Cassandra Drivers*)

Note: It is expected that this PDF document is quite large due to incorporating the above components. Please note that the report readability also contributes towards your assignment mark.

2. **.js and .cql files** from the following tasks:

   ☐ Task C.1 (Create MongoDB and Cassandra databases and provide the commands and screenshots as required)

   ☐ Task C.2 (MongoDB and Cassandra commands and screenshots of any outputs for the required tasks)

   ☐ Task C.3 (MongoDB and Cassandra commands and screenshots of query outputs for all queries)

   All of the above files must be run-able in MongoDB and Cassandra format. You should also clearly indicate each task using comments.

Note: Penalties may apply for any inconsistencies between the code and screenshots in the PDF report and that in the script files.

3. **[OPTIONAL]** **Driver script files (i.e. .py file)** from the following tasks:

   ☐ Task C.5 (MongoDB and Cassandra driver scripts and provide screenshots as required)

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

References

Restaurant & Consumer (RC) Data Set
https://archive.ics.uci.edu/ml/datasets/Restaurant+%26+consumer+data

THE END