

158.337 Group Project Instructions: Part B

You will continue to work in **pairs** for this assignment. Your group does not need to register again but in case you change your group partner please let us know via email (i.b.sofat@massey.ac.nz). You will lose significant marks if you do not work as a pair. You can use pairing forum for this but do this early on; in fact, if required sort this first. Use the following instructions to guide you in completing **Four Sections (Section A – Section D)** of the project. You should turn the work in for grading by the due date (**Tuesday, 2nd June 4.00 pm**).

Note: Make sure you have altered the password of your (Group) Oracle a/c and keep it safe within your group. You do not want other people accessing your account and code. We have seen this in past that students have been able to access other accounts, as the default passwords are not changed. Remember it is your responsibility to protect your work.

PART B

Queries (SQL/LINQ), Triggers, Procedures, Function, Cursor (PL SQL) & MongoDB (NoSQL)

Download following two files provided in a compressed folder (PartB.zip) under the Assignment heading from the Stream site:

University database file (UniversityDatabase.sql) - for Sections A, B and C.

The file provides code to create the required database in your account. Seven tables created are – LOCATION, FACULTY, STUDENT, TERM, COURSE, COURSE_SECTION, and ENROLLMENT.

Collection file (NoSQL.txt) - for Section D.

Use a new script file using some text editor (e.g. Notepad++) for writing your solution code. Name your file correctly (e.g. SectionA.sql). Include query number (e.g. a) along with the given query question (e.g. Write a query that will list...) as part of the comment before writing solution code (SQL / PL/SQL / LINQ) in this file. Annotate your code with suitable comments wherever necessary.

Use a word file for your project report (this can be later converted into a pdf should you wish to submit a pdf report). Report will be mainly used for grading so will need to transfer your final working code in this report as well. Here you also need to include the output results of your code. The work included in the report should be well formatted (proper alignment, appropriate column names, etc.) for readability purposes. The code file will be used wherever necessary (for example may not use for simple queries as can see the output but may use for the other code to ensure that it works in different simulated situations).

For PL/SQL (triggers, procedures, etc.) besides the creation code, the output produced from creation (e.g. procedure successfully created, etc.) also include the testing

examples (e.g. execution of the procedure using some quality test data example(s) to prove that procedure/trigger works the way it is intended to work).

All (SQL/LINQ) queries must **only** be based on the information provided in the assignment question itself (e.g. Write a query that will list...). **Do not** use a different criterion to arrive at the expected equivalent query result set. Also, **do not** manipulate the query results by using **ROWID** to sort and get the first record, etc.

Please make sure the query results display the asked information and the columns that are necessary to easily evaluate your query results. That doesn't mean use * to display everything – use your judgement and fulfil the requirements. Your code should work correctly even when database is modified (e.g. added more records, deleted some, modified some values, etc.).

Note: For **all exercises** asking you to **display names**, list first name and last name as a single column instead of separate columns.

Remember to include all three the **asked question** (as comment), the **code** (SQL /PL SQL /LINQ/ MapReduce) and its **output** in your report. Certain questions can have other additional requirements, so meet those as well.

Section A (SQL Queries)

(20 marks)

Note: For each of the questions (a)-(j) in Section A, write code using a SINGLE statement ONLY (i.e. you cannot write two different select statements to arrive at an answer; however, select can be used more than once in a statement i.e. **only one semi-colon** in your code for one question). Use only the **given** criteria. We will either give full or zero marks for each of the questions in this section so make sure your answer is fully correct.

- a. Write a query that will list all the students who have had their 25th birthday (i.e. aged 25 or over). Display students' ages in complete years (e.g. 27) and in the order of decreasing age. (1 mark)
- b. Write a query that will list the total building capacity of various buildings. The rooms with a capacity of less than five must be excluded when generating a building's total capacity. This list (in the increasing order of the total capacity) should only contain the buildings with a total building capacity of 150 or over. (1 mark)
- c. Write a query that will list all faculty supervisors and their respective students. Arrange your list in the order of faculty supervisor's names. (1 mark)
- d. Write a query that will list all the faculty members (along with the building code and room number) who are located in the (BUS)iness building. (1.5 marks)

- e. Write a query that will list students who enrolled in the courses offered in either the Fall term of 2017 or the Fall term of 2018. Do not display the duplicate student names in the output. (1.5 marks)
- f. Write a query that will list all the students (along with their grade and course details) who got at least B or better grade (i.e. B or A) in any of their courses. The list should be in the order of student id. (2 marks)
- g. Write a query listing the details of the faculty member(s) who supervise(s) the highest number of students. The result should also display the number of students. The query should also work in situations when more than one supervisor has highest students (e.g. 2 supervisors each having 10 students, 10 being highest). (3 marks)
- h. Write a query that will list students enrolled with a total of 15 or more course credit points. List your results in the decreasing order of total credit points. Do not assume or hard code the value of the course credits (e.g. 3, 6, etc.). (3 marks)
- i. Write a query that will list student(s) enrolled with the highest total course credit points. The result should also display the number of courses a student is enrolled for along with the highest total credit points. (3 marks)
- j. Write a query that lists all the courses (with their course names) and the course sections that are offered either on a (M)onday or at least four times a week. Also, display the number of the days that the courses are offered (e.g. 5 days). (3 marks)

Note: In table Course_Section, the attribute c_sec_day lists weekdays, where the first letter represents a weekday (e.g. M-Monday,.....F-Friday; for Thursday R is used). The course section weekdays are listed in the order M(onday) to (F)riday i.e. Monday being the first.

Section B (PL SQL)

(26 marks)

Comprehensive testing examples needed for **all** PL/SQL (triggers, procedures, cursor and function). You need to capture and show that you tested your code using good test cases. May include some exception handling if deemed necessary.

- k. Think of some **useful** business rules or situations where it would be appropriate for your triggers to fire. **Do not** write triggers to do something that could be done using other database design constraints (e.g., simple referential integrity checking, default values to attributes, or simply saying **record is being inserted, or an attribute has a null value, etc.**). Provide sensible and useful trigger examples and **do not use the already given or similar triggers** for this question.

Start with first clearly explaining the context, purpose (what they will do) of your triggers. Then provide the PL SQL code and the results. Triggers should be based on the tables already provided. Do not unnecessarily create too many and/or similar tables. Adding one or two tables may be okay – but justification needed. Altering a table (adding a field) is fine.

Write **two** triggers **one statement level and another row level**. Display the successful creation and running of the triggers. Please ensure that you also display the relevant tables before and after (results of the trigger) the trigger is fired. Remember to provide the purpose of your triggers (as stated in question l. below, this helps us to evaluate your work against the stated requirements). (10 marks)

- l. Write a trigger that does not allow more than two 'Full' ranked professors as part of the faculty (For example, trigger should fire if a new (third) Full professor is added or rank of one of the existing Associate professors is updated to Full). Provide comprehensive test data and results to confirm that the trigger works. (4 marks)
- m. Write a procedure to insert a new faculty record. The procedure should also automatically calculate the faculty salary value. This calculated salary is 15% less than the average salary of the existing faculty members.
Provide rest of the attribute values as input parameters. Execute your procedure to insert at least one faculty record. (3 marks)
- n. Write a trigger to check that when salary is updated for an existing faculty the raise is not over 4%. (4 marks)
- o. Write a cursor to list course sections for all the MIS courses (along with their courses names and credits). (3 marks)
- p. Write a function, which can be used to format faculty member's salary to \$9,999,999.99. Do not hard code the exact salary datatype (i.e. your function should work even if in future some minor changes are made to the salary data type/size). Call this function in a SQL statement for displaying a faculty member's salary. (2 marks)

Section C (LINQ Queries)

(6 marks)

For (q) – (u) below, write queries using LINQ.

- q. List all faculty who earn 80,000 or over. (1 mark)
- r. List all courses that have MIS in their course number. (1 mark)
- s. List all faculty and their location details. (1 mark)
- t. Display the total number of rooms in each building. (1.5 marks)
- u. Display total number of students supervised by each faculty in the order of faculty last name. (1.5 marks)

Section D (NoSQL)

(8 marks)

Use the code provided in NoSQL.txt to create a collection of eleven rows called dragons.

Now write **MapReduce code** to generate a report based on:

- the **gender-wise average weight**.

(e.g. – for the female gender the average weight will be say 99.99 (some value))

Include both code and the report generated in your report (printed work). Place the code in script file as well.

Assignment Submission

Organise your **final report** (print friendly – though hard copy not required) to include complete requirements. Make sure the report contents are also in the order of the laid assignment requirements. There are **three** parts to this assignment submission, a soft copy report (word/pdf) and two other resources - **scripts (code file) and database objects (in your Oracle Group account)**.

You must put your final report and code file(s) into a “single” compressed file and submit via the Stream Assignment link. Make sure there are not too many separate/unnecessary duplicated code files (e.g. all section A queries in a single file and not separate files).

Checklist for **final report**:

Readable format of SQL, PL/SQL, LINQ commands, and Map Reduce code. Remember to include the related **outputs** along with the code (not in appendix). For some questions, there are additional specific requirements (e.g. describe the purpose of triggers using plain English statements to, test data, etc.).

Checklist for **submission under your Group Oracle a/c**:

All objects - database tables, triggers, procedures, etc.

Checklist for **Stream submission**:

Check your project work and make sure that all the scripts run without any errors. Also, check that appropriate names have been given to all file(s). Upload your report and code on Stream as a single compressed file on the Assignment link.

Note:

Enter your names, student ids on the project marking sheet (**Appendix B**). **The project report should include this marking sheet (at the front i.e. 1st page). Make sure to provide the correct Oracle username in your report. Remember to include the script file(s).**

- Include both the report and script(s) in the compressed file (e.g. zip) and upload on the Stream assignment link.
- Include your objects (procedures, etc.) under your Oracle a/c.

Appendix B

158.337 Project Marking Sheet

(Attach this page at the front of your project report.)

(Please make sure you provide all the necessary details)

Oracle Account: Group_____

Group Member 1 - ID number, Name

Group Member 2 - ID number, Name

(Grader's section, please do not write below this)

PART B: _____/60marks

Part B Comments:

Project Part B marks: /15