158.337 Group Project Instructions: Part B

You will continue to work in **pairs** for this assignment. You do not need to register again but in case you change your group partner please let me know via email. Use the following instructions to guide you in completing the project. You should turn the work in for grading by the due date **(Tuesday, 24th May 4.00 pm).**

Note: Make sure you have altered the password of your Group Oracle a/c and please keep it safe within your group. You do not want other people accessing your account. Remember it is your responsibility to protect your work.

PART B

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

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

University database file (UniversityDatabse.sql) - for Sections A, B and C.
The file provides code to create the required database for Part B. Seven tables created are – LOCATION, FACULTY, STUDENT, TERM, COURSE, COURSE_SECTION, and ENROLLMENT.

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

Create a script file using some text editor (e.g. notepad) for writing your solution code. Include query number (e.g. a) and the given Query statement (provided in plain English) as part of the comment before each query's SQL / PL/SQL / LINQ code and output results. Make sure the output results in the project report are reasonably formatted (proper alignment, appropriate column names, etc.) for readability purposes. Questions (within a section) are not of same value.

For triggers, procedures and function include the creation code, the output produced from creation (e.g. procedure successfully created) and testing of each trigger/procedure/function (e.g. execution of the procedure using some quality test examples).

All the queries (SQL/LINQ) must be based **only** on the information provided in the query (English) statement. **Do not** use a different criterion to arrive at the expected equivalent query result (set). Also, do not use ROWID etc. to manipulate the query results. Please make sure the query results display all the columns that are necessary to easily evaluate the query results.

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 the **BOTH** the **code (SQL /PL SQL /LINQ/ MapReduce)** and its **output** in your printed report.

Section A (SQL Queries) (20 marks)

Note: For each of the questions (a)-(j) in Section A, write only a single query (i.e. you cannot write two select statements to arrive at an answer and use only the given criteria). We will either give full or zero marks for questions in this section.

a. Write a query that will list the students who have completed 26 years. Display their ages in completed years and in the order of decreasing age.

b. Write a query that will list the total building capacity for every building. This list (in the increasing order of the total capacity) should only contain the buildings with a total building capacity of 100 or over.

c. Write a query that will list all the students and their respective (faculty) supervisors. Arrange your list in the order of faculty supervisor's names.

d. Write a query that will list the faculty members who are located in the (LIB)rary building.

e. Write a query that will list 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.

f. Write a query that will list students who enrolled in the courses offered in the terms Spring 2013 or Spring 2014. Do not display the duplicate student names.

g. Write a query listing the details of the faculty member(s) who supervises the highest number of students. Also display the highest number of supervised students.

h. Write a query that will list students enrolled with a total of 12 or more credit points (in the decreasing order). Do not assume or hard code the value of the course credits.

i. Write a query that will list student(s) enrolled with the maximum total credit points.

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).
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 Thu**r**sday 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)

For (k) below, provide the **equivalent plain English statements** (that clearly explain the purpose of your triggers), **PL SQL code** and **the results.** Provide **sensible and useful** examples and **do not use** the already given or similar triggers as your examples for this question.

Note: For Statement and row level Generic Trigger question (k) – triggers shouldn't do what could be easily done within the create table commands (e.g. if something is null or not and then display this has a null/not null value – no credit for such triggers will be given as the null/not null constraint could achieve the same).
Triggers should be mainly based on the tables already provided and not unnecessarily create too many and /or similar tables. Altering table (adding a field) is fine. Adding one or two tables may be okay – but justification needed.
Comprehensive testing needed for all triggers and procedures.

k. 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 triggers are fired. Remember to provide the equivalent English statements about your triggers' purpose.

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 is updated to Full for one of the existing Associate professors.). Provide comprehensive test data and results to confirm that the trigger works.

m. Write a procedure to insert a new faculty record. The procedure should also automatically calculate the faculty salary value. This calculated salary is 20% 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.

n. Write a trigger to check that when salary is updated (for an existing faculty) the raise is not over 10%.

o. Write a function to format a number (9, 2) to $9,999,999.99. Use this function in a SQL statement for displaying a faculty member's salary.

Section C (LINQ Queries) (6 marks)

For (p) – (t) below, write queries using LINQ.

p. List all faculty who earn 90,000 or over.
q. List all courses that have MIS in their course number.
r. List all faculty and their location details.
s. Display the total number of rooms in each building.
t. Display total number of students supervised by each faculty in the order of faculty name.

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** and **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 printed work. Place the code in script file as well.

Assignment Submission

Organize your **final report** to include complete requirements. Make sure the report contents are also in the order of the laid requirements. There are **three** parts to this assignment submission, a physical printed report and two electronic resources - **database objects and scripts. You must put all the files you used as solution to Part B exercises into a "single" zip file and should submit it via the Stream Assignment link.**

Checklist for physical report submission

Readable printout of SQL, PL/SQL, LINQ commands, and MapReduce code. Remember to include the related outputs as well. For some questions there are additional specific requirements (e.g. plain English statements to describe the purpose of triggers, test data, etc.).

Checklist for submission under your Group Oracle a/c
All objects - database tables, triggers, procedures, functions, etc.

Checklist for Stream submission

All the code should also be uploaded on Stream as a single zip file under the Assignment link. 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).

Note:

Enter your names, student id numbers on the project marking sheet **(Appendix B). Submit the completed project report with a marking sheet as a bound document. Make sure to provide the Oracle username in your report. Submit the script files as a single .zip file to the Stream assignment link. Include your objects (procedures, etc.) under your Oracle a/c.**

Appendix B

158.337 Project Marking Sheet

(Print and attach this page to your project report BEFORE you turn it in.)

(Please make sure you provide all the necessary details)

Group Number: _____

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:

Total Marks for the Project

_____/100 is equivalent to _____/25 marks