THE UNIVERSITY OF SYDNEY

School of Information Technologies
Bryn Jeffries

COMP9120/INFO2x20: Database Systems **Semester 1, 2017**

Assignment 3 Brief: Database Application Programming

1 Introduction

This assignment is about the programming of database interaction code within an application, building on the Olympic database scenario that you used in Assignments 1 and 2. The objectives are to gain practical experience interacting with a relational database using an Application Programming Interface (API) and transaction programming, and to understand the importance and application of basic security measures. There are also opportunities to use more advanced database application programming techniques, such as stored procedures, indexes and access control privileges.

This is a group assignment for teams of about 3 members, and it is assumed that you will continue in your Assignment 2 group. You should inform your tutor as soon as possible if you wish to change groups.

Please also keep an eye on the discussion forum and further announcements in Ed STEM.

2 Submission Details

A final version of your database model should be submitted via PASTA (linked from e-learning) by 11:59PM Sunday of Week 12 (June 4, 2017)

2.1 Submission Items

Please submit your solution in the 'Assignment' section of the unit e-learning site by the deadline, including the following items.

2.1.1 Source code

All source code items should be submitted as a single zip file. To allow automated testing of some parts of your submission it is important to follow specific conventions about the names and paths of files in your submitted zip archive.

Client Source Code The zip file should have a folder `client` containing all the source-code files you have written for your client application. This source code can be in your choice of language, such as Python or Java, and may use external libraries, but all database functionality (query construction, transaction handling, etc.) must be written without the use of automated tools or object-relational wrappers. All database functionality must be provided through a single file. Conventions will vary by language for the name and path of this file, so please check on Ed for further details.

Database Schema DDL/DCL If you have done any extensions that modify the database you should include all such additions (`ALTER TABLE` statements, views, stored procedures, triggers, indexes or grant statements) as a plain text file `sql/ddl_dcl.sql`. *You should ensure that this file runs on a clean version of the original schema on a Oracle 12c (COMP9120 students) or PostgreSQL 9.5 (INFO2x20 students) database without errors.*

2.2 Feedback

You have the opportunity to consult with the tutor each week and make use of feedback from them. Where resources allow we will also provide limited testing of your submission in PASTA, and you can submit multiple times prior to the deadline to identify some areas of improvement.

2.3 Academic Honesty

By uploading your submission to PASTA your group implicitly agrees to abide by the University policies regarding academic honesty, and in particular that all the work is original and not plagiarised from the work of others. If you believe that part of your submission is not the work of your group members you must bring this to the attention of your tutor or lecturer immediately. See the policy slides released in Week 1 for further details.

In assessing a piece of submitted work, the School of IT may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program. A copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.

2.4 Late submissions

If no draft is submitted by the deadline you will not receive a draft contribution score or feedback from your tutor. You still have an opportunity to submit a final version. Late submissions will be penalised 20% per day late.

3 Deliverables

Your submission should support the functionality detailed below.

3.1 Core Functionality

Member login Users must enter a member ID and password that matches a valid record in the database in order to access the rest of the application

Member details After logging in, members should be presented with details such as their Member ID, accommodation location, and their number of bookings.

Browse bookings Users can access a list of all their bookings, listed chronologically by the trip start time (newest first).

Search journeys Users can enter a day and start and end locations for a trip and see details of all matching journeys.

Make booking *Staff only* can book a trip on behalf on another member. This operation must be done **as a transaction**

Browse events All members can retrieve details of all events for a specific sport.

Browse event results Members can retrieve details for a given event, including the medallists for that event

3.1.1 COMP9120 - specific task

Indexing Identify a frequently used query in the application, and add appropriate indexed to improve its performance. You should be able to justify your choice in terms of its applicability to the given query, and the resulting change to the execution plan.

3.1.2 INFO2820 - specific tasks

Recursive location queries Extend the Search Journeys functionality by using recursive SQL to find retrieve all journeys between any locations that are children of the specified start and end locations.

Security Layer All access to the database should be as a public user (`info2120public`) through a security layer of views and stored procedures

3.2 Extensions

For full marks you should attempt two of the following:

Paged Results Limit the number of rows retrieved for a query by specifying a number of rows required and an offset, and support these in the GUI.

Analytical query Present statistical data for results that might be used in a dashboard panel, using OLAP SQL functions .

Choose your own extension You can discuss other options for extensions with your tutor, including one of the above options from another stream's specific tasks.

4 Marking

This assignment is worth 10% of your final grade for the unit of study.

4.1 Rubric

Your submissions will be marked according to the following rubric, with a maximum possible score of 10 points. These marks are based on how you present your project in the technical report, supported by inspection of your actual code and discussion with your tutor.

	Novice (0 pts)	Competent (1 pt)	Proficient (2 pts)
Core Functionality	Core functionality not satisfactory implemented.	Good attempt at most functionality but some significant omissions or mistakes .	All functionality implemented with no or only a few minor mistakes.
General SQL Usage	SQL not used appropriately in the system.	Sound use of basic SQL statements throughout.	Excellent use of SQL throughout client, including appropriate use of complex SQL (e.g., GROUP BY, OUTER JOIN).
Transaction Usage	No transactions used.	Demonstrated understanding and implementation of transactions for at least one major function.	All functions implemented with excellent understanding of transactions, with evidence of handling of failed commits and consideration of appropriate isolation levels.
Database Security	No protection against SQL injection or use of stored procedures.	Good attempt of a (non-trivial) stored procedure and/or protection against SQL injection.	Comprehensive protection against SQL injection. **INFO2820:** Security layer completely implemented with no direct access to tables.
Extensions	No extensions beyond the core functionality.	One extension was attempted, but either with mistakes or not very substantial.	At least one substantial extension to the given scenario was included, described and implemented correctly.

4.2 Group member participation

If members of your group do not contribute sufficiently you should alert your tutor as soon as possible. The tutor has the discretion to scale the group's mark for each member as follows:

Level of contribution	Proportion of final grade received
No participation.	0%
Full understanding of the submitted work.	50%
Minor contributor to the group's submission.	75%
Major contributor to the group's submission.	100%