Monash University
FIT5148 – Distributed and Big Data Processing, Semester 1, 2016

Practical Assignment

Individual Assignment

Due Date: Week 5, Thursday by 3pm

This assignment is **worth 30% of the total mark** for this unit.

Demos/Interviews will be held during the tutorial time in week 6 and 7.
Make sure you arrange an interview/demo time with your tutor by Week 5.

An overview:
This assignment includes designing and implementation of **Monash Conference Management (MCM) system** that will utilise a distributed database model. The application will be used by conference organisers to allow paper submission from authors, allocate reviewers to submitted papers, and make decisions on the paper status through using a Java-based GUI client.

In this assignment, you will create an Oracle Database from NetBeans IDE and use Java code to manage the database through a user friendly interface. The client application and operations will be also implemented in NetBeans.

The Report
This assignment requires you to submit a report **in a Word document** that includes all the SQL and PL/SQL codes, your Java code (not automatically generated code), diagrams, fragmentation explanations and screenshots according to each task's requirement. The report should follow **the right order** (e.g. Task 1 part one, Task 1 part two, etc), use **headings and subheadings**, and **one or two sentences explaining** each content such as code, screenshots, or diagrams.

The assignment consists of 6 Tasks. To receive the full mark for each task, you need to complete all the requirements of a task and **meet the nine additional marking criteria** as well.

Task 1- Database (8 marks)

The potential tables are described here.

The Monash Conference Management system will store data in two locations. The MCM system stores information about a number of conferences organised throughout the year.

1. The database will store the data about main entities including conference, conference track, paper, author, submission, PC member, review, and the best paper award:

a) A **conference** table will store information about name, year, starting date, ending date, country, city, venue, and contact email. **(0.5 mark)**

b) **Each conference can have many tracks**. **Each track can accept many papers.** A **track** table will store information about the conference, the track name and description. **(0.5 mark)**

c) **Each track can have many PC members.** A **PC member** table will store information about first name, surname, title (e.g. Dr, Prof, or none), position (e.g. lecturer, researcher, student etc.), affiliation (e.g. Monash University) and email. **(0.5 mark)**

d) **Each PC (Program Committee) member can review many papers.** The **review** table will store information about the PC member, the paper, due date, reviewed date, recommendation and comment. The recommendation can be only: *accept, weak accept, borderline, weak reject, and reject*. **(0.5 mark)**
Each paper will be **reviewed by exactly 3 PC members. (0.5 mark)**

e) A **paper** table will store information about title, abstract, type and submission date. The paper type can be only: *full paper, Research-in-Progress papers and posters.* **(0.5 mark)**

f) Each author can write many papers. Each paper can be written by many authors. A **submission table** will store information about the authors, paper, and the order of authors (e.g. if 3 authors, order values will be 1, 2, 3 allocated to each author). **(0.5 mark)**
One of the authors of a paper will be assigned as **the corresponding author**. **(0.5 mark)**

g) An **author** table will store information about first name, surname, affiliation, country, email, and contact number. **(0.5 mark)**

h) One paper from each track will receive the best paper award. The **best paper award** table will store information about the track, paper title, and award price. **(0.5 mark)**

Based on the above-mentioned descriptions, you will **create tables,** decide on **attribute names and data types,** and **establish proper relationships** between them**.** You need to make decisions on using **appropriate PKs and FKs.** You will add **integrity constraints, check constraints or triggers** to achieve the above-mentioned requirements and conditions.

a. Provide your SQL code for all the parts (a) to (h)

2. Tables will be implemented and then distributed at FIT5148A and FIT5148B.
Conference table will be implemented on **FIT5148A** and **all the other tables on FIT5148B. (0.5 mark)**

(Note: those tables on two different locations that have relationships need to be dealt with in a different manner and this is specified in Task 3)

3. Draw and provide **an ER diagram** for your designed schema**.** **(0.5mark)**

a. Provide an ER diagram

4. You will create at least **one index** for an appropriate attribute and **sequences** where necessary and increment them by 1. **(1 mark)**

a. Provide your SQL code

5. Population of **meaningful data** in the tables (**minimum 5 records per table**) **(1 mark)**

a. Provide your SQL code

Task 2 – Oracle Stored Procedure (1.5 mark)

- Create an **Oracle Stored Procedure** for **updating the venue attribute** in the conference table. Show with screenshots that you call it and it works. The screenshots should **show an example of updating code** and **the results** that are displayed when it is executed. **(1.5 mark)**
    a. **Provide your screenshot and SQL code.**

Task 3 – Triggers (3 marks):

- **Oracle Triggers** are used to **implement referential integrity** across those tables that have relationships but are located **across FIT5148A and FIT5148B**. These triggers need to consider **all the possible operations** (**insert, update, and delete**). **(3 marks)**
    a. **Provide your SQL code.**

Task 4 Fragmentation (6.5 marks):

1. From all the tables you created, you will select ONE table for **Vertical fragmentation** and ONE for **Horizontal fragmentation**. These two tables should be **different**. The fragmentation will split a table into two subrelations/fragments. To do this task, you will make your own **assumptions** regarding **user applications** and MUST **provide them** (refer to lecture 2 and 3).

a) For the horizontal fragmentation, you also need to **provide the lists of predicates** and **minterm predicates** that you use. **(3 marks)**

b) For the vertical fragmentation, you need to provide four **queries, Usage Matrix**, **Access Frequency Matrix** for two sites of FIT5148A and FIT5148B, and **Attribute Affinity Matrix** of all the attributes. **(3 marks)**

c) Implement both types of fragments on the database such that they are populated with original tables' data (refer to Tutorial 1 and how we created tables from other existing tables). Use different and meaningful names for each fragment. **Two implemented fragments CANNOT be in the same location.** Populate your fragments with data. **(0.5 mark)**

a.Provide your assumptions, user applications and queries, lists of predicates and minterm predicates, specified matrices and your SQL code to create fragments on two locations.

(Note: this task's purpose is to practise the theory of fragmentation based on the lectures. Fragments will not be used after they are created)

Task 5 – User Interface (9 marks):

You will creating a Java-based GUI client in NetBeans that will consist of the following screens. You need to use **jTable** in all the screens except the Main Screen to view records.

1. **A main screen** that allows user to view other frames and navigate between frames. **(1 mark)**
2. **Conference Frame** which enables four operations to **view, insert, update and delete** records from the conference table. This frame enables to **search for conferences by city**. **(1 mark)**

3. **Track Frame** which has four operations to **view, insert, update and delete** records from the track table. This frame enables to **search for tracks by a conference** name. **(1 mark)**
4. **Author Frame** which has four operations to **view, insert, update and delete** records from the author table. This frame enables to **search for authors by country**.(**1 mark)**
5. **PC Member Frame** which has four operations to **view, insert, update and delete** records from the PC member table. This frame enables to **search for PC members by affiliation**. **(1 mark)**
6. **Review Frame** which has four operations to **view, insert, update and delete** records from the review table. This frame enables to **search for reviews by the recommendation**. **(1 mark)**
7. **Submission Frame** which has four operations to **view, insert, update and delete** records from the submission table. This frame enables to **search by conference. (1 mark)**
8. **Paper Frame** which has four operations to **view, insert, update and delete** records from the paper table**.** This frame enables to **search by conference. (1 mark)**
9. **Best Paper Award Frame** which has four operations to **view, insert, update and delete** records from the best paper award table. This frame enables to **search for awards by conference**. **(1 mark)**

a. Provide only your written code (DO NOT include the GUI code), screenshots of frames and operations.

Note: **DO NOT create a database connection in each frame**. There is **a mark deduction** if you make multiple connections to the same database concurrently and if you do not close the connections properly.

Task 6 – Web based Client (2 marks)

A web-based Client is implemented using any platform like jsp or JavaScript. This includes four operations to **view, insert, update and delete** records for **the author table**. **(2 marks)**

a.Provide your written codes of the operations and the screenshots.

Additional Marking Criteria:

Just completing a task **does not yield in a HD mark**. You need to meet these marking criteria:

1. **the completion of all the tasks according to the specifications,**
2. **quality of the design and code,**
3. **quality of the GUI interface, this includes the layout, design, spacing, components used, ease of accessing and viewing information, feedback to user actions using messages (e.g. Update was successful),**
4. **The navigation between the frames should be easy and straightforward,**
5. **the level of effort and creativity,**
6. **functionality of all the operations,**
7. **handling of all the input errors and exceptions properly, and**
8. **answering questions during the interview and showing deep understanding of underlying concepts**.
9. **Quality of reports i**.e., headings, order, readability

Demos

The students will demonstrate their programs against tables in the FIT5148A and FIT5148B databases on the HIPPO.ITS.MONASH.EDU.AU server during the tutorial time in Weeks 6-7. Make sure you arrange an interview/demo time with your tutor by week 4. ***Students who do not attend an interview will receive zero marks for the assignment.***

Submission Requirements:

All the following files should be uploaded to Moodle as a zip file and use the following naming convention: FIT5148-A1-[StudentID].zip. There is a mark deduction for any missing document.

1. An Individual Assessment **Cover Sheet**
2. **Table of Contents**
3. One page that briefly lists which tasks you have completed and any additional functionality or features that you have done.
4. Providing **SQL code, ER diagram, java code and screenshots in the Task order with numbered headings** as specified in each requirement

Late Submission:

Late Assignments or extensions will not be accepted unless you submit a special consideration form and provide valid documentation such as a medical certificate prior to the submission deadline (NOT after). Otherwise, there will be **5% penalty per day including weekends.**

PLEASE NOTE.

Before submitting your assignment, please make sure that you haven't breached the University plagiarism and cheating policy. It is the student's responsibility to make themselves familiar with the contents of these documents.

Please also note the following from the Plagiarism Procedures of Monash, available at http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-procedures.html

Plagiarism occurs when students **fail to acknowledge** that the ideas of others are being used. Specifically it occurs when:

- **other people's work** and/or ideas are paraphrased and presented without a reference;
- **other students' work is copied or partly copied**;
- **other people's designs, codes or images are presented as the student's own work**;
- phrases and passages are used verbatim without quotation marks and/or without a reference to the author or a web page;
- lecture notes are reproduced without due acknowledgement.