



Software
Engineering
The University of Auckland

SOFTENG 351 S1 C – Assignment 2

Due Date: Thursday 30 April 2020 at 5pm

50 marks in total = 5% of the final grade

Q1. The COMPANY database

(30 Marks)

This questions makes use of the COMPANY database taught in the class. Here are the steps for creating the COMPANY database. Firstly, select the databased named “**stu_upi_SOFTENG_351_C_S1_2020_A2_Q1**” in PhpMyAdmin for this question.

Please note that the “**stu_upi**” is used as an example of the UPI. You will have your **own UPI** instead for the database name and used it in this assignment. Secondly, execute the SQL statements **given** in the “**COMPANY_DATA.sql**” file to create the tables and dump data into the database.

With data populated into the COMPANY database, we can now execute SQL queries to retrieve useful information from it. For example, the following query *retrieves the name and address of all employees who work for the “Research” department.*

```
SELECT Fname, Lname, Address
FROM EMPLOYEE, DEPARTMENT
WHERE Dname = 'Research' AND Dnumber = Dno;
```

The result of the query is shown in the figure below, which was generated by clicking the “**Print view**” link on the SQL page after the query execution.

SQL result

Host: Student mysql database

Database: teaching_jsun062_351_C_S1_2020_A2_Q1

Generation Time: Apr 15, 2020 at 03:49 PM

Generated by: phpMyAdmin 4.0.10.7 / MySQL 5.1.73-log

SQL query: SELECT Fname, Lname, Address FROM EMPLOYEE,
DEPARTMENT WHERE Dname = 'Research' AND Dnumber = Dno
LIMIT 0, 30 ;

Rows: 4

This table does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Fname	Lname	Address
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Joyce	English	5631 Rice, Houston, TX
Ramesh	Narayan	971 Fire Oak, Humble, TX

Print

Note that the above output not only shows the SQL statement and the result of the query, but also lists the working database name which contains your own UPI in it. You **should use this layout** (i.e., screenshots generated by the “Print view”) to answer the questions in the assignment.

IMPORTANT: *You need to populate the given data into the COMPANY database before answering the following questions. Please **DO NOT CHANGE** any data (states) in the tables, because these data are used to generate correct output in order to answer questions of the assignment.*

- 1) Use the set UNION operation to define a query that list all the projects (show project names, project number and location information) that the employee(s) with the last name ‘James’ who is either as a worker or as a manager of the department that controls the project.
[3 marks]
- 2) List all the employees (show the employee ssn, first and last names) who live in “Houston” and work on projects that are located in the same city.
[3 marks]
- 3) Retrieve a list of employees and the projects they are working on of the "Hardware" department (show project name, hours, employee ssn, first and last names), ordered by the project names in ascending, then by the hours worked in descending, and then alphabetically by the employees' last name in ascending.
[3 marks]
- 4) Retrieve the names of all employees (show first, last names and the department name as “HoD”) who do not have supervisors and manage a department.
[3 marks]
- 5) Use nested query to retrieve the managers (show employee ssn, first and last names and the gender) who have at least two dependents.
[3 marks]
- 6) Use the JOIN operation to define the following query. For each department, retrieve the department number, department name, the number of employees (as “No_of_Employees”) in the department, sum of the salaries (as “Total_Sal”) of all employees, the maximum salary (as “Highest_Sal”), the minimum salary (as “Lowest_Sal”), and the average salary (as “Average_Sal”).
[3 marks]
- 7) For each project on which has more than three employees working, retrieve the project number, the project name, and the number of employees who work on the project (as “Team_Size”), and the total number of hours allocated to the project (as “Total_Hours”).
[3 marks]

- 8) Specify the following views in SQL and display the data states of the views.
- a) A view (“dept_managers”) that has the department number, department name, manager’s first and last names, and the manager salary for every department, with the restriction that the manager must work for the same department.
[1.5 marks]
 - b) A view (“proj_summary”) that has the project number, project name, controlling department name, number of employees (as “Team_Size”), and total hours worked per week on the project (as “Total_Hours”) for each project.
[1.5 marks]
- 9) Find the top 5 highest paid employees (show all the attributes) in the company. (*Hint:* the LIMIT clause at the end of the SQL statement can be used to select a fix number of tuples from the query).
[3 marks]
- 10) Find the top 3 hardworking employees in the company. In this option, the query retrieves the employees (with employee ssn, first and last names) who are ranked in top 3 of number of hours worked per week (as “Work_load”) together with their salary and department name (as “Department”).
[3 marks]

Q2. The STUDENT database**(20 Marks)**

This question uses a simplified student registration database, which consists of the following tables:

- **Students:** This table records information about students. The gradAssistant attribute records whether the student is a graduate assistant or not. The graduate assistants automatically qualify for a full tuition waiver (they still have to pay the fixed fee). The instate attribute records whether the student is an in-state student or not. Again, this has an impact on the fees the student would pay.
- **Staff:** This table records information about staff users of the system. There are two categories of staff: “Registrar” and “Department”. These users would have different capabilities and functions in the application to be developed. Note: A view called lunarUsers is created to provide a simple way to authenticate users of the system.
- **Courses:** This table records information about courses in the university catalogue which includes course number, title and credit hours. The credit hours value will be used in calculating the GPA in the student’s transcript.
- **Sections:** This table records the course offerings for each term and includes the term, year, and course record number (crn), a unique number assigned to course offerings for a specific term and year. The table also includes start time and end time and meeting days as well as the name of the instructor. Finally, this table records a Boolean value (yes or no) called auth to indicate if the registration for this course is open to all or is done only by authorization.
- **Enrolls:** This table records information about which student has registered for which course offering.

- **Authorizations:** This table records authorizations given to students for specific course offerings. Two types of authorizations are given: OVFL for overflow, i.e. allows students to register in a course offering that does not have any open seats, and AUTH for authorization to register in a course offering that is designated as an authorization only course offering.
- **FixedFee:** This table records information about all fixed fees a student is required to pay each term they register.
- **VariableFeeRate:** This table records per credit hour fee rate for different categories of students (graduate vs undergraduate students and in-state vs out-of-state students).

There are 3 kinds of database users in the system:

- **Registrar Staff:** These users will have the ability to load the database tables, make changes to courses, sections, fee details etc.
- **Department Staff:** These users will have the ability to authorize students into sections, overflow students into sections, add assistantship information to the system, generate class lists etc.
- **Student:** These users will be able to register for classes, see their schedules, see fee detail, see transcripts etc.

The following real-world constraints need to be enforced by the system:

- Undergraduate students are not allowed to register for graduate courses numbered 6000 and above.
- Students should not be allowed to register for a class which is FULL unless they have an overflow.
- Students should not be allowed to register for a class which is listed as AUTHORIZATION ONLY unless they have an authorization.
- Undergraduate students are not allowed to register for more than 20 hours in a semester and the limit for graduate students is 15.
- Students cannot register for two classes that overlap in meeting time.

Select the database named “**stu_upi_SOFTENG_351_C_S1_2020_A2_Q2**” in the PhpMyAdmin tool for this question, and execute the SQL statements given in the “**STUDENT_DATA.sql**” file for creating the tables and dump data into the database.

After the data has been populated into the database, we can now execute SQL queries to retrieve useful information from it for this question.

IMPORTANT: You need to populate the given data into the *STUDENT* database before answering the following questions. Please **DO NOT CHANGE** any data (states) in the tables, because these data are used to generate correct output in order to answer questions of the assignment.

- a) A department staff can perform the following functions, e.g., authorize students into sections, overflow students into sections, add assistantship on system and generate class lists. Write a SQL query to generate the class list for the section “10101” in term ‘FA’ of ‘2002’. The query should show attributes crn, ctitle, cprefix, cno, sid, fname and lname.

[5 marks]

- b) A student can perform the following functions, e.g., add a section, drop a section, see schedule for a term, see fee detail, and see transcript information. Write a SQL query to generate the timetable schedule for the student “1111” of the term “FA” in “2002”. The query should show attributes `crn`, `ctitle`, `cprefix`, `cno`, `days`, `startTime`, `endTime`, `room`, and `instructor`.

[5 marks]

- c) A registration staff can perform the following functions, e.g., load sections from file, load grades from file, increase section capacity, display term schedule, display student transcript, and display student schedule and fee detail. Write a SQL query to display the transcript records of the student “1111”. The query should show attributes `year`, `term`, `cprefix`, `cno`, `ctitle`, and `grade`.

[5 marks]

- d) Some real-world constraints can be defined as SQL assertions and enforced onto the database state. For example, the restriction of “*Undergraduate students are not allowed to register for graduate courses numbered 6000 and above.*” Can be expressed with the following SQL assertion.

```
CREATE ASSERTION undergraduate_enrolment_restriction
CHECK (NOT EXISTS ( SELECT *
                    FROM students AS S, courses AS C,
                    sections AS SE, enrolls AS E
                    WHERE S.sType = 'UGRAD' AND S.sid =
                    E.sid AND C.cno = SE.cno AND SE.crn
                    = E.crn AND C.cno >= 6000 )
);
```

Write a SQL assertion definition (‘`workload_restriction`’) that specifies the restriction of “*Undergraduate students are not allowed to register for more than 20 hours in a semester and the limit for graduate students is 15.*”.

Unfortunately, the MySQL DBMS used with the PhpMyAdmin tool does not support assertion definitions. However, you can still partially check the syntax of the `SELECT` statements used in the assertion definition. To answer this question, you don’t need to provide the screenshot of the query result, but to include the correct SQL assertion definition as the answer.

[5 marks]

Remark: You are required to use the PhpMyAdmin tool to complete the assignment. You should submit the assignment in a single PDF file that contains all your answers to the above questions. The answers (except for question 2(d)) should be generated as screenshots by the “**Print view**” function of the SQL window after the query executions, as shown by the example output. Note that the screenshots of the answers must contain **your UPI** in the correct database names. As for question 2(d), please just include the SQL assertion statement of the answer in the submission file.