



MACQUARIE UNIVERSITY
Faculty of Science and Engineering
Department of Computing

ISYS224/ITEC624 Database Systems 2019 (Semester 2)

Assignment 2

Database Programming and Implementation (worth 15%)

Check-in: 6 pm, Wednesday, October 23rd, 2019 (Week 11)

Final Due: 6 pm, Wednesday, October 30th, 2019 (Week 12)

Student Code of Conduct

Macquarie University students have a responsibility to be familiar with the Student Code of Conduct: <https://students.mq.edu.au/study/getting-started/student-conduct>

Student Support

Macquarie University provides a range of support services for students. For details, visit <http://students.mq.edu.au/support/>

Important Note:

This assignment will be marked out of 15 and will contribute 15% towards your final grade.

You are required to electronically submit your assignment as per specification by the due date.

1 Problem Domain

The problem domain is a simplified portion of what you came across earlier in Assignment 1.

Advertising agency Shine manages its client companies' advertising campaigns. It has a number of staff members with specialized skills. A record is kept of each client company, and of every campaign. Each campaign is based on a unique theme. A Shine staff may work on zero, or one or more campaigns at a time.

When a campaign starts, an estimate of the cost needs to be set and agreed upon. The actual cost of a campaign is calculated from staff time devoted to it. Staff members are paid (and clients charged) based on the hourly rate associated with their pay grade.

When the campaign is completed, an actual completion date and the actual cost are recorded and a single invoice for the whole campaign is sent to the client. The issue date and payment status of the invoice are also recorded. When the client pays, the date paid is recorded.

A relational data model for this domain is provided below to facilitate your understanding of the domain. A DDL script (`create_DB.sql`) for creating the corresponding database, and a DML script (`populate_DB.sql`) for populating this database with some sample data are being provided in the Assignment 2 folder. Note that the DDL script will also create another table (not included in the diagram), called `alerts`, which is meant to contain messages as to which invoices are overdue.

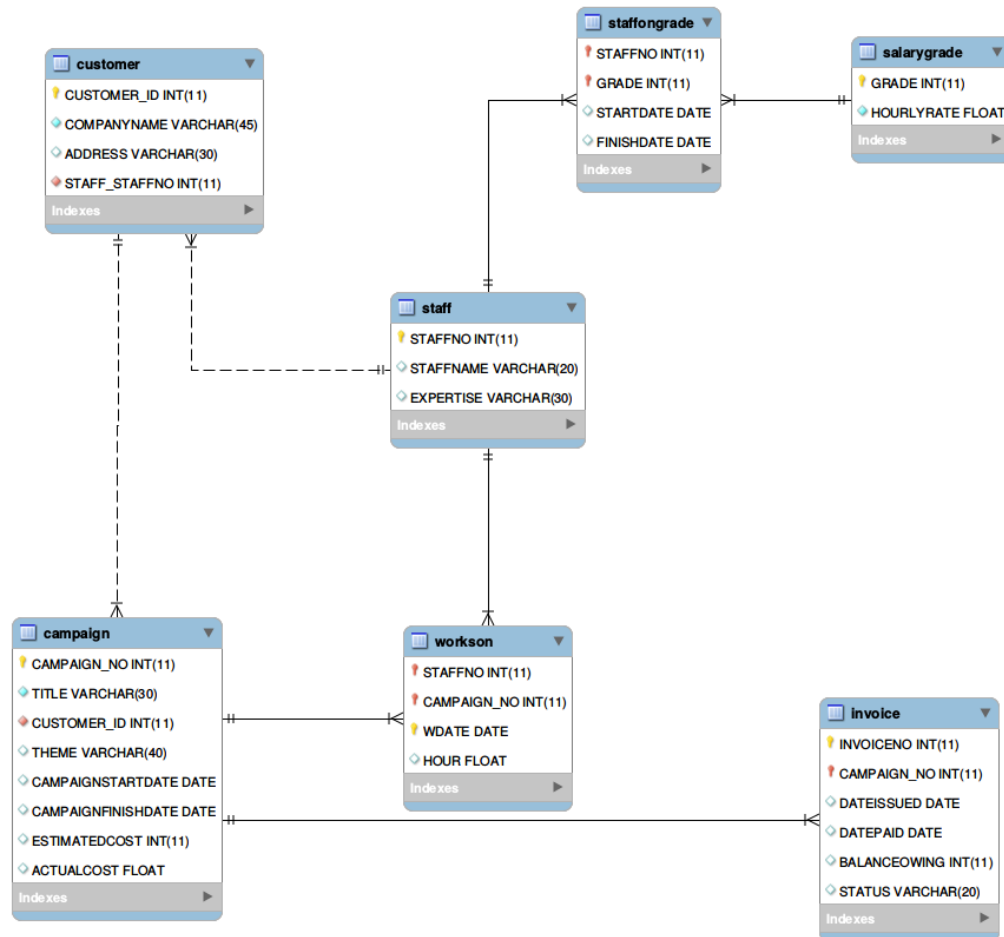


Figure 1 : Relational Data Model

2 Task Specifications

Task One [2 marks]

Create the tables listed in the relational data model as shown in Figure 1, as well as the **alerts** table, by running the provided DDL script provided in the 'Assignment 2' folder. Then insert sample records to the tables by running the provided DML script provided. Verify that the tables are created and populated as intended.

Task Two and Three [10 marks]

A template (`stored_programs_template.sql`) is provided for you to write the required programs (one trigger and two procedures). You must not change the “signature” (name, parameters with their types, orders, etc.) these programs. It is best to take this template and fill out the required codes. Remember to put your name and student number at the top as required. You

should save this file with name of the following format, and submit it among other files: `<yourLastname_yourFirstname>_programs.sql`.

Task Two [4 marks]

Shine considers any invoice that remains unpaid for more than 30 days after it has been issued as overdue. Write a trigger called `tr_overdue` that fires whenever the `invoice` table is updated. If the `STATUS` of any invoice is updated to "OVERDUE" from something else, then a new row is inserted to the `alerts` table with the current date as the `message_date`, user as the `origin`, and 'Invoice with number: `x` is now overdue!' as the message, where `x` is the invoice number in question.

Task Three [6 marks]

As part of this task you will write two procedures:

1. Procedure `sp_finish_campaign (in c_title varchar(30))` that takes as input the title of a campaign, and records that the corresponding campaign has been completed by (a) updating the `CAMPAIGNFINISHDATE` to the current date, and (b) updating the `ACTUALCOST`. The actual cost of a campaign is calculated from the number of hours different staff put into it on different dates, and the salary grade they were on such days. The procedure should signal an error with the message 'ERROR! Campaign title does not exist' if the input does not match any existing campaign.
2. Procedure `sync_invoice()` that identifies the invoices which are still recorded as UNPAID (as value of `STATUS`) more than 30 days after the invoice was issued. It updates those records, and sets the status to OVERDUE.

Task Four (Testing) [3 marks]

This task involves testing the code developed in Task Two and Task Three.

- First you are required to test the programs you wrote against the sample data provided. The tests to be carried out, in the order given, is provided in the provided file: `test_script.sql`. These tests constitute a minimal test against a very small number of records. Look at this file carefully to understand what is being tested and what should be the expected result. Note the roles of setting `autocommit off` and the `rollback` statements.
- Next you carry out a more extensive test by testing the programs against a larger set of records that are designed to easily expose any flaws in your programs. You do that by deleting records, adding records, or modifying the records in other ways, and then calling different procedures. You accordingly modify the file `test_script.sql`, and save it as `<yourLastname_yourFirstname>_test_script.sql`, and submit among other files.

3. Report Preparation

You will also prepare and submit a report (in the PDF format). A word file template for this purpose will be provided which you will complete, **convert to pdf**, and submit. The file you submit would be named: `<yourLastname_yourFirstname>_report.pdf`.

Your report should have the following sections:

1. Initial State of the database. Paste to the word file the screen shots showing the provided sample data in the tables.
2. Stored Programs. Paste into this section the programs you wrote (the sql file `<yourLastname_yourFirstname>_programs.sql` that you prepared).
3. Required Testing against Sample Database. Paste into this section the initial tests you ran (one by one) and then the corresponding results as screenshots. The provided script file lists what tests to run, how, and in which order.
4. More Extensive Testing. Explain what sort of changes you are going to make to which tables, what tests you are going to run, and why. Copy and paste from your file `<yourLastname_yourFirstname>_test_script.sql` the DML statements you used to modify the tables followed by the screenshots of the records in those tables; then copy and paste the procedure calls you made, and the screenshots of the records in the relevant tables (or the error messages).
5. Notes (optional). In this optional section, you might wish to note anything, such as whether you faced any particular difficulty in completing any of these tasks, the nature and extent of any help you received from anyone, and why.

Remember to convert this file to pdf and submit only the pdf file.

4. What to Submit, and When?

You will submit three files:

1. `<yourLastname_yourFirstname>_programs.sql`
2. `<yourLastname_yourFirstname>_test_script.sql`.
3. `<yourLastname_yourFirstname>_report.pdf`.

You will submit the files in two stages. In the first stage, you must submit two draft files by 6:00pm, Wednesday Week 11:

- a) of the program file `<yourLastname_yourFirstname>_programs.sql` including at least the trigger `tr_overdue` in it, and
- b) of your report file `<yourLastname_yourFirstname>_report.pdf`, with complete Section (1) and partially complete sections (2) and (3).

You can modify these files while preparing your final version. However, **failure to submit the two draft files by the required date (week 11 Wednesday) will attract a penalty of 3 marks.**

The final version of these three files must be submitted by 6:00pm, Wednesday Week 12.