CANTERBURY CHRIST CHURCH UNIVERSITY

### Computing, Digital Forensics and Cybersecurity

#### **YEAR TWO COMPUTING 2018-19**

**Developing Database Systems with SQL (MCOMD2DDS) *SEMESTER 1***

# ASSIGNMENT 1 (GROUP)

**16 October 2018**

**Title: Titanium Fabrications Ltd (TFL)**

**Weighting (as a percentage of whole module): 50%**

**Set By:** Gerald Stock (Gerald.Stock@canterbury.ac.uk)

**Checked By:** Vijay Sahota (Vijay.Sahota@canterbury.ac.uk)

Dates of Submission:

**Assignment 1**:

**19 November 2018 at 14:00** via Turnitin and Blackboard (see below).

Any late submissions of the assignment must be submitted via Turnitin and Blackboard (see below).

***One* submission per group is required.**

**Individual Report**:

**26 November 2018 at 14:00** (i.e. *after* the submission of Assignment 1) via Turnitin.

Any late submissions of the Individual Report must be submitted via Blackboard.

***Each* group member *must* submit an Individual Report and *failure to do so will result in a mark of zero being awarded* to the *individual concerned.***

***If you are allocating marks equally amongst all of your group members then the contents of the "Analysis of Own Progress", "Analysis of Group's Progress", "Analysis of How Well the Group Worked as a Team" and "Other" sections of your Individual Report can all be left blank.***

##### Assignment Feedback

* your mark will be available on Blackboard;
* feedback, where appropriate, and your mark will be provided on a bespoke feedback sheet handed out during one of the time‑tabled sessions;
* feedback of a general nature will be provided during one of the time‑tabled sessions;
* feedback in the form of a model answer will be presented during one of the time‑tabled sessions.

##### General Instructions

* This is a group assignment. The groups will be listed on Blackboard.
* *Each group will consist of three members who will work together as a team in order to carry out all the required tasks. If you choose to form a group of only two members then you should attempt to find a third member of your choice otherwise you risk having someone who is not yet in a group assigned to your group and that someone might not suit you!  
  If you are in a group of two or have not yet found a group to join, do not let this stop you beginning work on this assignment.  
  If you can’t find a group to join then please get in touch with the module leader. Groups of four will not normally be allowed.*
* IT IS YOUR RESPONSIBILITY TO CONTACT OTHER MEMBERS OF YOUR GROUP
* Some helpful notes for guidance can be found on Blackboard.
* Please check on the Computing All Years General Blackboard within the Assessment Info folder which itself is in the Course Information folder for general advice on how to present your work.
* Late submission of coursework will be penalised.
* You will, from time to time, be contacted by e-mail regarding this assignment. It will be assumed that any e-mail message sent to you (including that sent after you have completed this module) is read, understood and acted upon. It is therefore essential that you check your e-mail regularly.
* You may, as a group and/or individually, be asked to attend a formal meeting where any aspect of this assignment may be subject to discussion/demonstration.
* Students are expected to be available at university until the last day of each semester. While the intention is not to prevent students from going home before the end of semester (for example, after their last examination) such action must not interfere with the smooth running of this module. If you plan to absent yourself from university before the end of any semester, please obtain prior written permission from the module leader as early as possible prior to your proposed absence.

# Learning Outcomes

By the end of the module students should be able to:

1. analyse effectively and model the data for an existing/required system;
2. conceptually design a database (after affective fact finding);
3. logically design a database;
4. effectively implement and test a small relational database application system using the tools provided within a modern multi-user database management system environment.

**Scenario**

**Title: Titanium Fabrications Ltd (TFL)**

As a result of a strategic study TFL, has decided to develop a Personnel and Costing system. The main aim of the system is to improve control over the way that staff are allocated to projects and to obtain more up-to-date information about costs.

The Requirements Specification includes an Entity Relationship diagram that is given in Appendix A. Within the Requirements Specification the following have been identified as the *only* data items necessary for the new system:

Employee number, employee name, address-house number, address-road, address‑UK‑post code (which, of course, has a specific format), date of birth, status (Pensioner or Non‑pensioner, for ex-employees, or Current employee) and national insurance number (which has a specific format) are always held for every employee/ex-employee, while:

1. current monthly salary and office number are always held for current employees only;
2. current monthly pension is always held for ex‑employees who are Pensioners only;
3. leaving date is always held for all ex-employees only.

Job code (one upper case alphabetic character (excluding I, O, Q, V, Y and Z) followed by one numeric character), which represents a particular job title, and job appointment date are both stored for every job. Note that job titles themselves are *not* stored. There are currently several employees with a job code of K7.

Project code, project name, project target completion date and project budget (person-hours) are always stored for all projects while project actual completion date is only stored for completed projects.

Sub-project number (two digits) and sub-project name are always stored for all sub‑projects.

Assignment description and running total of actual person‑hours worked to date (on a particular assignment by a particular employee) are always stored for all assignments while actual start date and actual finish date are only stored when available.

The Requirements Specification also includes the following business rules:

details are held of both the current job, if any, and *every* previous job held by all current and ex-employees;

all project codes begin with the characters 'GBR', followed by 3 numeric characters noting that all six characters are always stored;

all sub-project numbers are unique only within a specific project;

an employee may be concurrently assigned to more than one *different* sub‑project;

an employee is only ever assigned to any specific sub-project once;

an employee will only ever hold a particular job, identified by job code, once;

an employee is never promoted more than once per day;

when a new employee is appointed to his/her first job the job appointment date for that job is effectively the date upon which the employee starts working for TFL.

It is planned to implement a replacement system for the above scenario. To this end, a Senior Systems Analyst has produced an ER (Entity Relationship) diagram (see Appendix A) that models the basic data requirements for the new system.

1. Requirements

You have been asked to design and implement a relational database for the scenario given above.

* 1. **Check Your Understanding**Check your understanding of the scenario in conjunction with the material provided in the appendices. Perceived (and possibly actual) ambiguities will be resolved in timetabled classes.
  2. **Extend the Conceptual Data Model**Extend the Conceptual Data Model given in Appendix A by:

1.2.1 assigning attributes to the *appropriate* entities.

1.2.2 mapping the ER diagram onto a set of relations.

1.2.3 describing, in an implementation independent fashion (i.e. independent of any specific (R)DBMS), a set of tables that unambiguously includes ***all***:

* primary keys and, if any, alternate keys;
* foreign keys;
* attributes and their data types (including their length/precision) and whether they are mandatory or not;
* defaults, if any;
* user-defined semantic constraints implied by the scenario and the appendices;
* additional constraints that you consider appropriate.

It is strongly suggested that the above is developed in tabular format.

**Notes:**

* **do not** introduce any attributes other than those mentioned in the scenario;
* **do not** change the ER diagram provided (except to remove many to many relationships, if any);
* **do not** introduce any additional entities or add any subtypes[[1]](#footnote-1).

***Whilst the work done here in this section is an essential prerequisite for that in section 1.3 below you are not required to submit it as part of your assignment and no marks will be awarded for it.***

***DO NOT SUBMIT ANYTHING FROM THIS SECTION AS PART OF THIS ASSIGNMENT.***

* 1. **Implementing the Logical Model**

1.3.1 Write a command file to create a set of tables, based on the work that you have done in section 1.2 above, using the Data Definition Language (DDL) subset of Oracle's SQL\*Plus covered in this module.

***The only tools that you are permitted to use to create the required command file are Notepad or an alternative text editor.***

You are required to use version 11.0.x.x.x of the Oracle SQL\*Plus environment.   
If you produce solutions using different versions of the software then you ***must*** ensure that they work using the software available on the University's network.  
  
Your command files should be **neat, easy to comprehend and *appropriately* commented**, and must be useable to re‑create your tables with the minimum of effort.

Ensure that you make ***appropriate*** use of the commenting features of Oracle SQL and Oracle SQL\*Plus in preparing your command file.

Failure to provide a neat, easy to comprehend and appropriately commented command file will result in loss of marks.

***You must not include any regular expressions, packages, procedures, functions or triggers in your answer[[2]](#footnote-2).***

***Marks will be awarded for this section.***

1.3.2 Detail all the user-defined constraints (developed in section 1.2), if any, that you have not implemented, explaining why you have not done so.

***Marks will be awarded for this section.***

**1.4 Testing Constraints**

You should, of course, test all the constraints that you have used in your table definitions. Note, however, that the testing of 'standard' constraints (such as primary key, foreign key (including on delete cascade), not null and unique), data types, defaults and Oracle’s standard functions would normally be cursory. It would, however, be advantageous for you to thoroughly test any constraints (standard or otherwise), data types or defaults where you do not fully understand their implications.

You should, however, thoroughly and exhaustively test all the user-defined (semantic, i.e. check) constraints. The documentation supporting the testing of each user-defined/semantic constraint should include all those items discussed/detailed in class.

You may be asked to show details of your testing at a formal meeting where any aspect of this assignment may be subject to discussion.

***Whilst testing is essential to prove that what you have produced in section 1.3 above actually works you are not required to submit it as part of your assignment and no marks will be awarded for it.***

***DO NOT SUBMIT YOUR TESTING AS PART OF THIS ASSIGNMENT.***

**2. Submission Details**

Your *assignment* must be submitted as specified on the front page of this assignment and as detailed below:

* a well laid out and *appropriately* commented single Oracle **SQL\*Plus command file** named S1\_2DDS\_2018-19\_<Your-Full-Name>\_tables.sql that can be used to create ***all*** of the required tables submitted via Blackboard.  
  Your command file must include, at the front, your group number and group members as a comment. Your code must be easily understandable and easily maintainable.  
  You must ensure that your command file can be run by typing   
  start S1\_2DDS 2018-19 <Your-Full-Name> tables.sql at the SQL> prompt using version 11.0.x.x.x of the Oracle SQL\*Plus environment[[3]](#footnote-3).
* a single report (.docx) which ***must*** be submitted via Turnitin[[4]](#footnote-4) with the following descriptors:  
   Assignment Title: S1\_2DDS <Your-Full-Name>  
   Document Name: S1\_2DDS\_2018-19\_<Your-Full-Name>  
  The report ***must*** contain:
  + a title/front page (including your group number and group members);
  + a contents page;
  + an introduction section;
  + a section entitled “**Implementing the Logical Model”** containing an ***exact*** copy[[5]](#footnote-5) of the fileS1**\_**2DDS 2018-19 <Your-Full-Name> tables.sql and details of all the user‑defined constraints (developed in section 1.2), if any, that you have not implemented;
  + a conclusion section;
  + a references section which ***must*** include reference to, amongst others, ***all*** code written by others that you have included in your submission;
  + an appendixwhich should hold the minutes from your group meetings.

and **must conform to the Assignment & Report Layout & Presentation Requirements** **detailed on the Computing All Years General Blackboard within the Assessment** Info folder which itself is in the Course Information folder.

Please do not submit .zip, .rar, .7z, .pdf, etc. files.

Your *Individual Report* must be submitted as specified on the front page of this assignment via Blackboard.

**Before** you submit your work check that:

* **all** required items are included;
* items not required are **not** included.

Now ***re-read*** all the requirements regarding this assignment to ensure that you are aware of what is required and thus do not:

* spend time producing material that does not need to be submitted;
* omit material that is required.

**3. Marking Guidelines**

Assignment 1 will be marked out of 100.

|  |  |
| --- | --- |
| Section 1.3  Marks will be awarded for:   * logical design * neatness of code * appropriately commented code * appropriate data typing * omission of redundant data, if any * appropriate constraints * appropriate defaults * details of all user‑defined constraints, if any, not implemented | 90 marks |
| Presentation[[6]](#footnote-6)  See the **Assignment & Report Layout & Presentation Requirements** detailed on the Computing All Years General Blackboard within the Assessment Info folder which itself is in the Course Information folder. | 10 marks |

**Each student within a group may be awarded a different mark.**

**APPENDIX A: Entity Relationship Diagram**

****

**NOTE: The above diagram *must* be printed at 300 dpi (dots per inch) otherwise the dotted lines, if any, will *not* be visible on the printed copy.**

APPENDIX B: Group Work Individual Report

Note that:

* ***one*** Individual Report is required for ***each*** group member;
* 1500 words should be regarded as the absolute ***maximum*** for an Individual Report;
* **If you are allocating marks *equally* amongst all of your group members then the contents of the "Analysis of Own Progress", "Analysis of Group's Progress", "Analysis of How Well the Group Worked as a Team" and "Other" sections of your individual report can all be left blank.**
* Individual Reports should be submitted separately from the assignment;
* a template for the Individual Report can be found on the Computing All Years General Blackboard within the Assessment Info folder which itself is in the Course Information folder. Individual Reports in any other format are unacceptable;
* failure to submit the Individual Report, by the specified date without good reason, will result in a mark of zero being awarded to the individual concerned for the assignment as a whole.

**APPENDIX C: Example Minutes of Meeting**

An example for the minutes of a meeting can be found on the Computing All Years General Blackboard within the Assessment Info folder which itself is in the Course Information folder.

1. Subtypes have ***not*** been covered on this module. [↑](#footnote-ref-1)
2. All these have ***not*** been covered on this module. [↑](#footnote-ref-2)
3. If you produce solutions using different versions of the software, then you ***must*** ensure that they work using the software available on the University's network. [↑](#footnote-ref-3)
4. TURNITIN will be used for originality/plagiarism checking: you will be allowed to upload your document as many times as you like until the submission date to perform your own plagiarism check in addition to the checking that will be performed by tutors marking your assignment.

   For further help and guidance on TURNITIN submissions go to the Blackboard HELP tab. The guidance includes an overview of TURNITIN, a guide to using TURNITIN through blackboard and information relating to copyright and data protection which you MUST read so that you understand your rights.

   For further help and guidance on Blackboard submissions go to the Blackboard HELP tab.

   **Important notes regarding the submission:**

   * When you submit your work via TURNITIN/Bb you will receive an automatic email receipt.
   * You will be allowed to submit your project after the submission deadline. Note, however, that the usual late penalties will apply.

   [↑](#footnote-ref-4)
5. A screen dump is *not* acceptable. [↑](#footnote-ref-5)
6. This refers to the quality of the non-technical aspects of your submission. It does *not* mean that you are required to give a presentation. [↑](#footnote-ref-6)