University of Brighton

School of Computing, Engineering and Mathematics

Coursework assignment 2018-19

Module Title:	**Databases II**
Module Code:	**CI204**
Author(s)/Marker(s) of Assignment	Goran Soldar, Jennie Harding
Assignment No:	1
Assignment Title:	Individual Coursework
Assignment weighting:	50% of module mark
Module Learning Outcome/s Covered (Refer to module syllabus)	LO1 design and build a relational database system LO3 access and manipulate data using the standard language SQL LO4 develop stored procedures and triggers LO5 make use of DBMS facilities to ensure the integrity and security of a database and make use of utilities
Assignment Brief and Assessment Criteria:	
See attached sheets for tasks and Assessment Criteria and Case Study	
Date of issue:	w/b 16th October 2018
Deadline for submission:	**The submission is in two parts. Each part is equally weighted.** **Part 1: HAND IN DATE:** 8/2/19 at 15:00h via StudentCentral (PDF format) **Part 2: HAND IN DATE:** 30/4/19 at 1500h hours via StudentCentral (PDF format) *Note - Students are allowed to submit work within two weeks of the published deadline or the last working day immediately prior to the feedback date if this is shorter than two weeks – late work is capped at the pass mark.*
Date feedback will be provided	Feedback will be provided electronically through Turnitin within 4 weeks of the hand-in date

1. A copy of your coursework submission may be made as part of the University of Brighton's and School of Computing, Engineering & Mathematics procedures which aim to monitor and improve quality of teaching. If a copy is made, it will be kept only for this purpose and will be destroyed once this purpose has been fulfilled. You should refer to your student handbook for details.
2. All work submitted must be your own (or your team's for an assignment which has been specified as a group submission) and all sources which do not fall into that category must be correctly attributed. The markers may submit the whole set of submissions to the JISC Plagiarism Detection Service.

Assignment Tasks

Part 1: Design (50%)

Write a report which documents the logical design of a database to support the Information System requirements described in Appendix A. Your logical design should be carried out using the top-down and bottom-up techniques that you have been taught in the module.

1) *Evidence of normalization.*

Normalise the supplied booking document (removing repeating groups, resolving functional dependencies). See Case Study. Two instances of the booking form are supplied. The outcome of the normalisation should inform your ERD.

2) *Definition of tables / Data Dictionary*

For each table, list the table name, a description of the entity that is implemented, the primary key of the table (name and data type) and other identified attributes (names, data types and all other constraints).

3) *Entity-Relationship Diagram to complete the Logical Design*

Produce a key-only Entity-Relationship diagram showing primary and foreign keys to meet the requirements, showing all entities and relationships (with relationship name and cardinality).

4) *Evaluate your design*

Write a short (250 word) evaluation of your database design, discussing the requirements that you have included and any problematic issues that were difficult to resolve.

Part 2: Implementation (50%)

You will be supplied with a partial ERD after the feedback date for part 1, in case you choose to revisit your design before full implementation.

Write a report which documents the implementation of the database to support the Information System that you designed in part 1. Your report must contain a short description of each stage of the implementation and explain the logic of the any procedural code that has been written. You must also provide the code itself and screen-shots to demonstrate that each part of your design works correctly**. You may be required to demonstrate these and will need to keep a copy of your implemented database until marks have been returned.**

5) *Create tables using SQL DDL*

Write SQL statements to create the tables that implement the database you designed in part 1. Your report must show the SQL CREATE TABLE statements and images of the tables created in SQL Server or mySQL or another DBMS approved by the module tutor.

6) *Populate tables using SQL INSERT*

Write SQL statements to populate all the tables you have previously created, with at least five records each. Your report must show the SQL INSERT statements and images of the tables when they have been populated. *NOTE: you should consider creating a formal test data set where you know what the outcome of a given SQL query will be.*

7) *Retrieve information using SQL queries*

Write SQL queries to demonstrate that your system can address the requirements in Appendix A. Your report must show the SQL code and the result of executing each query. These should be clearly labelled with the requirement number.

8) *Implementation of DBMS functionality*

Design appropriate stored procedures, functions, triggers and application to implement the functionality defined in Appendix A. Your report must show the SQL code and the result of execution.

End of tasks

Part 1: Design (50% of total coursework mark)

Marking criteria *(Weighting for PtA)*	*Factors contributing to a good grade*	*Minimum for a pass mark*
Normalization (25%)	All repeating groups have been removed. All functional dependencies have been resolved. Tables are in 3NF.	An attempt has been made to normalize the tables that shows understanding of the principles
Tables (25%)	All necessary tables are defined. Appropriate primary key is identified. Other attributes are appropriate, well-named and with any other constraints appropriate to the system	Entities described as tables with primary keys and other useful attributes
ERD (25%)	Diagram properly drawn using a CASE tool. Diagram matches the tables and relationships identified in other parts of the documentation. Multiplicity and optionality indicated on relationships.	ER diagram produced using a standard notation
Evaluation (25%)	A well-written discussion to evaluate the design decisions, recognising valid alternatives and justifying choices made.	Report discusses the principles of databases design and applies them to the case study

Part 2: Implementation (50% of total coursework mark)

Marking criteria *(Weighting for PtB)*	*Factors contributing to a good grade*	*Minimum for a pass mark*
Creation of Tables in SQL (15%)	SQL syntax correctly used in each statement with evidence of successful execution. Proper use of a full range of constraints (data types, primary key, foreign key, not null, unique and check, where appropriate), default values and generated keys	SQL syntax correctly used in some statements. Attributes have reasonable data types and primary keys are defined
Population of tables (15%)	SQL syntax correctly used in each statement with evidence of successful execution. A range of data values to test any constraints on attributes.	SQL syntax correctly used in some tables inserting data to each table created
SQL queries (30%)	Correct SQL syntax to answer the query with evidence of successful execution. Partial answers may gain some marks if there is evidence of a reasonable attempt.	SQL syntax correctly used mainly correctly to attempt all MUST queries
Stored procedures and triggers (30%)	Correct SQL syntax to perform the defined actions with evidence of successful execution. Partial answers may gain some marks if there is evidence of a reasonable attempt.	Correct type of procedural program selected for the defined functionality. Some attempt to write the code as demonstrated in lab sessions.
Application (10%)	A simple application is created that will run the select queries and procedures.	No attempt necessary to pass assignment

A more detailed marking rubric will be made available via StudentCentral.