

**Business Computing undergraduate Assessment Brief**

**Module Name:** Databases 2

**Module Number:** *CSY2038 PR1 Build*

**Student Declaration:**

Submission of this piece certifies that the attached work is my (or in the case of group work) our own. All external references and sources are clearly acknowledged and identified within the contents. Referencing is in accordance with the Universities Guidelines. I am/we are aware of the University’s regulations concerning plagiarism and collusion.

**Assessment Aim(s)**

The aim of this assessment is to:

1. Design, create, test and document methods for building an object relational database with useful data abstraction and automatizing useful processes in PL/SQL

**Assessment Requirements**

**You should base your designs on the ERD (separate file).**

**Tasks**

1. **Based on the ERD design a schema to include a number of procedures, functions and triggers**

* Select an appropriate **sub section** of the ERD **approx. 5 tables** to implement and add objects to, presented in a **schema** **and skeleton tables**
* Determine extraction methods for querying and processing simple and complex data
* Identify relevant, useful activities that can be automated using PL/SQL
* Briefly describe the purpose of each procedure, function or trigger
* List the naming conventions you will be adopting

10 Marks

1. **Implement your designs from TASK 1**

* Organise your files appropriately
* You should include tables, udts, collection types, queries, procedures, functions, triggers and cursors

60 Marks

1. **Testing** - *to be presented in the documentation*

* Draw up an appropriate functional test plan **for the PL/SQL**
* Establish appropriate test cases
* Test your programme, *you do not need to test the DDL SQL*

15 Marks

1. **Documentation**

* Document your project including
  + Proposed automation strategy eg task 1
  + Evidence of additional research eg bulk binds, switch
  + Commented code
  + The test plan

10 Marks

1. **Video Demo – All group members** 5 Marks

**Deliverables**

This assessment requires you to submit the following:

* ***1* *script file***, called CSY2038PR1 and initials with comments at the top of the script file which include student names and numbers
* The script file should have the extension .txt or .sql
* ***1 word document*** for the documentation including the design (task 1) and tests (task 3). The code should be included as **text** appendices not images (task 2). The word document should have the same name as your script file
* ***1 video demo***. The recorded demo should be no longer than 10 mins and should demonstrate all the SQL and PL/SQL and tests included in your script files
* All submissions should be through the module Blackboard ‘Submit Your Work’ section

Students are responsible for keeping backup copies of their own work. Loss of work will not be accepted as an excuse for late submission

**Assessment Learning Outcomes**

The learning outcomes for this assessment are: (from module spec)

**Knowledge and Understanding**

1. Demonstrate an understanding of data definition and data manipulation in SQL databases
2. Exhibit a knowledge of procedural SQL (PL/SQL)

**Subject Specific Skills**

1. Develop a series of script files to implement a complex multi-table object relational database using a commercial database management system (DBMS)
2. Design and develop a series of extraction methods to provide useful intelligence

**Key Skills**

1. Problem Solving: Construct strategies to locate, access and extract relevant information

**Feedback Methods**

Grades and feedback methods for this module are outlined in the Module Blackboard under the ‘Feedback and Grades’ sections

**Marking Criteria**

(see Student Handbook for generic criteria)

**Task 1 – Design**

1. Well considered and comprehensive schema with appropriate notation and naming conventions. Useful and relevant procedural and data extraction methods that include UDTs and collection types and built in functions.
2. Logical and complete schema with appropriate notation and naming conventions. Examples of some useful and relevant procedural and data extraction methods that include UDTs and collection types and some built in functions.
3. Logical and partially complete schema with mostly appropriate notation and naming conventions. Examples of some procedural and data extraction methods that may or may not be useful or relevant include some UDTs and possibly collection types and some built in functions.
4. Limited schema may be illogical and flawed, with some appropriate notation and naming conventions. Examples of some procedural and data extraction methods may have been omitted, and may or may not be useful or relevant include some UDTs and possibly collection types and some built in functions. Missing elements.
5. Minimal attempt, weak notation, errors and omissions, limited consideration given to procedural elements.

**Task2 – SQL and PL/SQL**

1. Well-constructed tables, with correct datatypes, **some complex object types, well-structured** dummy data. Entity and referential integrity correct. **Includes all drops**

A **broad** range of named sub routines working. Small well-constructed procedures and functions designed to achieve a single task. Includes **examples of most** items such as cursors, built in functions, loops. Appropriate naming conventions, exception handling

1. Well-constructed tables, with correct datatypes, **some object types**. Test data. Entity and referential integrity correct. **Includes all drops**

A **range** of named sub routines working. Small well-constructed **working** procedures and functions designed to achieve a single task. Includes **a range of** examples of items such as cursors, built in functions, loops. Appropriate naming conventions.

1. Well-constructed tables, with correct datatypes, **some object types, mainly working**. Test data. Grants. Entity and referential integrity correct. **Includes all drops**

**Some** named sub routines **working** procedures and functions. Includes **some** examples of items such as cursors, built in functions, loops. Appropriate naming conventions.

1. Poorly structured tables, **some object types attempted** test data, ill-considered datatypes, little or no entity or referential integrity. Errors and omissions. DCL contains errors

**Some** named/**anonymous** sub routines working, **mostly** **working** procedures and functions. Includes **some** examples of items such as cursors, built in functions, loops. Appropriate naming conventions.

1. Poorly structured tables, ill-considered datatypes, little or no entity or referential integrity. Errors and omissions.

No or not working sub routines, limited examples of items such as cursors, built in functions, loops. Some appropriate naming conventions. May or may not be working

**Task 3 – Testing**

1. **Comprehensive** test schedule, specific and detailed, implemented, tests include syntax and functionality testing as well as user access levels. Has used and referenced a range of methods
2. **Most** tests included in schedule, specific and detailed, **most** implemented, tests include syntax and functionality testing as well as user access levels.
3. **A number** of test included in schedule, maybe **woolly** or **lacks detailed**, most implemented, tests include syntax and functionality testing as well as user access levels. May or may not be referenced
4. **Some** tests included, woolly and generic
5. Limited or no testing

**Task 4 – Documentation**

1. **Consistently** well written using a **professional** language, presented and structured, structured. Evidence of **research**, includes, plan/overview, testing, code
2. Tidy structure, **mainly** professional language some evidence of research, includes, plan/overview, testing, code
3. **Headings included,** missing components, may or may not use professional language
4. Missing items, maybe poorly structured, lacks references
5. Limited, missing items, or just a code dump

**Video Demo**

1. Ran smoothly from **run commands**. Clearly prepared, includes **test statements** to illustrate code. **Able to explain**, what and why code has been written.
2. Ran smoothly from run commands. Clearly prepared, includes **test statements** to illustrate code. **Able to explain**, what and why code has been written. With a few elements that **could have gone better**
3. **Mostly ran from run commands**, some statements manually copied and pasted. **Minor changes** of data and statements during the demo an attempt to get it working. Can explain why they have implemented most of the elements.
4. Few ran from run commands, some statements manually copied and pasted. **Changing data and statements** during the demo in an attempt to get it working. **Struggles to explain** why they have implemented some elements.
5. No show