Assessment Task – General

National Unit Details			
Code(s)	ICTDBS403 ICTDBS502 ICTWEB425	**Title(s)**	Create Basic Databases Design a Database Apply SQL to extract and manipulate data
Assessment Number	AT 1	**Assessment Title**	Build Query Challenge

Section 1 – General Assessment Information	
Decision Making Rules	Every task must be completed satisfactorily to be assessed as competent in the unit. * *For graded units, competence must be demonstrated before a mark can be given.*
Reasonable Adjustment	Students may request reasonable adjustment for assessment tasks. Reasonable adjustment usually involves varying: ▪ the processes for conducting the assessment (eg: allowing additional time, varying the venue) ▪ the evidence gathering techniques (eg: oral rather than written questioning, use of a scribe, modifications to equipment) However, the evidence collected must allow the student to demonstrate all requirements of the unit.
Special Consideration	Students can apply for Special Consideration where personal circumstances have adversely affected their task result or ability to undertake an assessment. A Special Consideration form can be completed prior to, but no later than 3 days after, the date of the assessment and submitted to the relevant Manager.
Re-submission *(where tasks are not satisfactorily completed)*	Assessment tasks that are not satisfactory can be resubmitted up until the end of the unit as scheduled on the Unit Outline. The timing on this may depend on the equipment required for this assessment task. **NOTE**: Assessment tasks submitted for the first time after the end of the unit as scheduled on the Unit Outline will not be assessed and student should be told to re-enrol in the unit.
Plagiarism	There are serious penalties for plagiarism. Students must ensure that all assessments are their own work (or group work). Please refer to https://www.swinburne.edu.au/current-students/manage-course/exams-results-assessment/plagiarism-academic-integrity/

Assessment Task – General

National Unit Details			
Code(s)	ICTDBS403 ICTDBS502 ICTWEB425	**Title(s)**	Create Basic Databases Design a Database Apply SQL to extract and manipulate data
Assessment Number	AT 1	**Assessment Title**	Build Query Challenge

Section 2 – Student and Assessor Instructions

Conditions

This assessment is:

- *Open book*
- *Individual*

You will need access to:

- *Computer with MS Office*
- *Access to an SQL Server database.*

Task Overview and/or Description

Download the following files/documents:

- Build_Query_Challenge_ERD.pdf
- Build_Query_Challenge_Data.xlsx
- Build_Query_Challenge_Data_Dictionary.pdf
- Build_Query_Challenge_Submission_Template.docx

Setup:

- Create a (public) git repo for your database & clone it into a suitable local dev environment.
- Add the provided Submission Template to your git repo.
- Create a SINGLE .sql file in your git repo, put ALL the SQL you write in this .sql script
- Add your name and student id to the top of BOTH the submission template and .sql script (it will need to be COMMENTED in the sql script.)
- If you are able to use Azure / AWS / Cloud Service:
  Create (and configure as necessary) an SQL Server database on a suitable cloud service.
  Capture a screenshot showing the database you created inside the cloud service admin view.
- If you are unable to use Azure / AWS / Cloud Service:
  Clean out your existing practice database (i.e. drop all the table from practice exercises) and use that.

Task 1.

- Convert the provided ERD to a Relational Schema.
- Follow the design as provided. Do NOT make design changes of any kind.
- Add your Relational schema in a COMMENT block inside your .sql script

***Commit** your work in your Git repo with the **commit message "Task 1 Complete"** & push it to origin.*

Task 2.

- Based **directly** on your **Relational Schema** from task 1 and the provided **Data Dictionary**, write and execute the DDL to create your database.
- Using an **SQL Query** (not the GUI) verify that all tables have been successfully created. Capture screenshot/s of this query and its result set & add it to the submission template.

***Commit** your work in your Git repo with the **commit message "Task 2 Complete"** & push it to origin.*

Assessment Task – General

National Unit Details			
Code(s)	ICTDBS403 ICTDBS502 ICTWEB425	**Title(s)**	Create Basic Databases Design a Database Apply SQL to extract and manipulate data
Assessment Number	AT 1	**Assessment Title**	Build Query Challenge

Section 2 – Student and Assessor Instructions

Task 3.

- Write and execute the DML to add the test data provided to your database

  Additional Data:
  You MUST also add a yourself as a client. Use your name and student id, make up any other required data.

- In the query editor on the cloud service your database is deployed on & run the query:

  Select * from client

- *Add a screenshot of the result set from the above query to the submission template*

***Commit** your work in your Git repo with the **commit message "Task 3 Complete"** & push it to origin.*

Task 4.

Write and run queries to meet each of the following requirements.

Query 1:

- Write a query that shows the client first name and surname, the tour name and description, the tour event year, month, day and fee, the booking date and the fee paid for the booking.

Query 2:

- Write a query which shows the number of bookings for each (tour event) month, for each tour in the following example format. For example:

EventMonth	TourName	Num Bookings
Jan	North	1
Jan	South	7
Jan	West	4
Feb	North	5

(The actual results will vary. This demonstrates format only)

Query 3:

- Write a query which lists all bookings which have a payment amount greater than the average payment amount. *(This query must use a sub-query.)*

***Commit** your work in your Git repo with the **commit message "Task 4 Complete"** & push it to origin.*

Assessment Task – General

National Unit Details			
Code(s)	ICTDBS403 ICTDBS502 ICTWEB425	**Title(s)**	Create Basic Databases Design a Database Apply SQL to extract and manipulate data
Assessment Number	AT 1	**Assessment Title**	Build Query Challenge

Section 2 – Student and Assessor Instructions

Task 5:

- Create a View based on Query 1 from Task 4

***Commit** your work in your Git repo with the **commit message "Task 5 Complete"** & push it to origin.*

Task 6:

- Write queries to prove your responses to task 4 are returning the correct/sensible results.
- E.g. to test that **select * from client** is returning the correct number of rows you could use **select count(*) from client** and check that the number in the count query is the same as the number of rows returned by the select * query.
- Provide a (short) written explanation of how each of your 'test' queries verifies that the related task 3 query is correct. Add these explanations as COMMENTS after each test query in your .sql script.

***Commit** your work in your Git repo with the **commit message "Task 6 Complete"** & push it to origin.*

Submit the link to your git repository in the Assessment submission portal on the Swinburne LMS

How the Assessment will be Conducted

You will be provided with

a) Build_Query_Challenge_ERD.pdf
b) Build_Query_Challenge_Data.xlsx
c) Build_Query_Challenge_Data_Dictionary.pdf
d) Build_Query_Challenge_Submission_Template.docx

Submission Details

Submit the link to your git repository in the Assessment submission portal on the Swinburne LMS

Your git repository must contain a submission template and a .sql script in which you have added your responses to each of the tasks.

Section 3 – Assessment Criteria (Evidence to be Provided by the Student)

Submit the link to your git repository in the Assessment submission portal on the Swinburne LMS

Your git repository must contain a submission template and a .sql script in which you have added your responses to each of the tasks.

Assessment Task – General

National Unit Details			
Code(s)	ICTDBS403 ICTDBS502 ICTWEB425	**Title(s)**	Create Basic Databases Design a Database Apply SQL to extract and manipulate data
Assessment Number	AT 1	**Assessment Title**	Build Query Challenge

Notes for the teacher

List in the Required Evidence column below all aspects of the task that are required to be demonstrated by the student for satisfactory completion of the task.

Ensure that it is very clear to the student the evidence they are being asked to provide. This should also ensure that different assessors would give a similar result when assessing the evidence of a student. Depending on the assessment task, the evidence listed may be a repeat of the Task Overview and/or Description but more often it is a clear list of only what is being assessed.

There must be room for Student Feedback in required evidence section.

* *For graded units, ensure the marks allocated to each requirement are not too broad a range. You could use a rubric in Bb.*

Please note: A suggestion for graded tasks is to list the required evidence to be assessed as Satisfactory, Unsatisfactory, Not submitted. Once the student has satisfactorily completed all required evidence, allocate the marks using a rubric.

Marking Guide				
Required Evidence		**Satisfactory**	**Not Submitted**	**Unsatisfactory**
1	Challenge Completed			
If the unit is graded, add rubric or how marks allocated here.				

Feedback to Student		
Task Result	☐ **Satisfactory**	☐ **Unsatisfactory**

	Name	Signature	Date
Assessor			