Assignment Specification -V1

Learning objectives

1. Apply the knowledge related to design, implementation and querying a relational database to solve a real-world problem.
2. Design and implement advanced SQL features (such as stored procedures, triggers etc.)
3. Implement a programming language interface to connect to a database.
4. Reflect on the design and implementation decisions, identify challenges, and suggest ways to improve based on modern database concepts.

1. Introduction

This assignment expects you to apply the knowledge you have gained in the unit via lectures, practical classes, and practical tests to design, implement and query a real-world database system. You will (a) design and implement a database, (b) fill it with sample data, query the database to get some meaningful information effectively, and (c) document your database design, implementation, query designs, results you have obtained and reflection of your work.
Your work will be assessed based on the code/ scripts and the documentation you produced, as well as a demonstration of your work.
The mark of the assignment is given out of 100, and the assessment is worth 50% of your final mark (overall mark of the unit) as specified in the unit outline.

2. Scenario

The Olympic Games 2024 has concluded recently. The Olympic Games is a global event held every four years where individuals and teams from hundreds of countries worldwide participate. The Paralympics also happen in the same year as the Olympics.

Assume you or your friends are interested in the Olympic Games (either the Olympics or the Paralympics) and need to know the details of the games, participants, winners and all the interesting things about this event.

You can assume you are interested in the 2024 Olympics or several years of games, including 2024.

3. Detailed description

Considering the given scenarios, you are expected to do the following tasks in this assignment (Marks are given out of 100):

Part 1. Designing a database and the relational schema based on it. [18 marks]

a. ER Modelling: [8 marks]
Identify entities and attributes, relationships, cardinality, and participation constraints and design an ER diagram. Think about different forms of entities, attributes and relationships that can be useful in your design so that you can show your ability to model diverse scenarios properly. The ER diagram should be drawn following Chen's notation used in lectures.

b. Define the relational schema: [6 marks]
Convert your ER model to a relational schema by doing suitable relational mapping. You may do this iteratively, starting first with basic tables, attributes, and relationships and then refining it to convert more complex relationships in the design.
All your tables should be in at least the first normal form, but if you have done the design and the relational mapping correctly, your tables would be in the third normal form.
You have to think about the constraints, such as primary and foreign keys. You may also add any other constraints deemed required. You are expected to improve the preliminary work by adding suitable referential integrity constraints.

c. Data description: [4 marks]
Then, decide on suitable data types and attribute level constraints (such as NOT NULL) for attributes. Show this information in a tabular format. For each attribute, you must define at least the attribute name, selected data type, description of the attribute and any constraints on attribute values. Any important business rules you assume also must be defined here.

Part 2. Implementing the database you have designed and load it with sample data. [16 marks]

a. Implementation: [12 marks]
Looking at the relational schema and the data description resulting from part 1 above, implement your database with suitable tables and constraints, including proper referential integrity constraints. First, create a sample database with <suitable name> _<your student ID>, then implement all tables there.

b. Loading sample data: [4 marks]
Insert sample values into the implemented database and demonstrate that the integrity constraints are met when entering data.
NOTE: Some web links to obtain sample data relating to the given scenario will be mentioned on the assignment page of Blackboard. You may use these sources or any other suitable source to obtain sample data. You have to enter a reasonable amount of data into the database so that meaningful results for the queries you will design in the next sections can be obtained.

Part 3. Designing and implementing queries: [16 marks]

When your database is up and running, it is time to retrieve data to answer some reasonable queries. First, think yourself and derive some meaningful and useful questions (around 6 -10 questions) regarding the data of your database, which can be converted to SQL queries to get answers. Make sure you think about using a single table or several tables, obtaining data based on conditions, string manipulation, etc. (there are many other aspects you can think about).
Then, for each of your questions, design and implement an appropriate SQL query to produce the required answer in an effective manner.
You should demonstrate that you can,
Level1: [8 marks]

Use basic SQL SELECT statements with the use of numeric data, date-time functions, string comparison and manipulation, and other related basic methods with suitable WHERE clauses.
Level 2: [8 marks]
Use joins and sub-queries with GROUP BY, ORDER BY, aggregate functions and related clauses.

Part 4. Increase the database functionality with advanced concepts: [16 marks]

Design and implement database programming concepts such as stored procedures, triggers, views, and indexes to improve the capabilities of your database. You are expected to design and implement at least *two* categories of the above-advanced features, including stored procedures and show at least two uses. For example, you can implement two useful stored procedures and two useful triggers. You are expected to demonstrate your ability to meaningfully use different concepts such as parameters, user-defined variables, loops/ if-else, different forms of triggers, complex views, etc.

Part 5. Connect to the database using a suitable programming language and show sample query results: [10 marks]

Demonstrate your ability to connect to a MySQL database via Python3 and use it in a Python3 environment/ Java environment.
First, you are expected to connect your database to Python3 or Java in a proper way and call your already defined queries via the Python programming constructs to show that you can do some useful database activities. Then, you are expected to demonstrate your ability to perform other operations, such as inserting/ updating/ deleting data via the programming environment. (Alternatively, you may use Java connectivity to answer this sub-section).

For parts 2 to 5 above, SQL statements are to be written. Overall use of proper SQL (comments, readability of code, styling, good practices, proper use of scripts, etc.) is also expected [10 Marks]

You may note that the total mark, as per the above-detailed description, is 88. The remaining marks will be allocated solely for the documentation (introduction, reflection, user guide etc.).

Note: You can re-use the SQL queries and approaches from the lectures and practicals, but as the scenario is different, you cannot use them directly. Remember to cite and cite your sources, if any. If you submit work that you have already submitted for a previous assessment (in this unit or any other), you have to state this specifically.

4. Documentation

You need to document what you have done in each stage of the assessment so that another person can get a clear idea about what you have done. You are expected to produce two short documents in this assessment.

1. User guide to implement and use your database [5 Marks]
   In this document, you are expected to describe clearly how the database you have designed/implemented can be created in a MySQL server and then use it to run the queries you have developed. You must clearly indicate how any MySQL scripts you have produced to create the database, create tables, etc., are to be executed, with relevant information about your MySQL version information, operating system, etc. You must provide the commands and may include screenshots also. (This is not your report.)

By following what you have written in the guide, another person should be able to implement and use your database. Use suitable headings and organize your document. Include a suitable sub-section/cover page indicating the assessment name, report title, your name, your Curtin student ID, and lab group. The user guide would be 2-5 pages.

2. Report on your database [35 Marks, including the design section's 18 marks]
   Your report should include the following sections:

   a. Cover page [1 mark]
      Include the assessment name, your name, Curtin student ID, and practical class (date/time/ lab number)
   b. Introduction [1 mark]
      Short overview of your work, including the selected scenario and activities you have done.
   c. Design of the database [18 marks, part 1 of the assessment task]
      i. Explain why you have selected the entities, relationships, data types, etc.
      ii. ER diagram, Relational schema, data description and any other material you have produced in the design stage i.e., part 1 of the assignment
      iii. Any assumption you made during the design of the database.
   d. Implementation of the database and adding sample data
      [3 marks out of 16 marks for the part 2 of the assessment task]
      Briefly describe how you have implemented the database with evidence of implementation.
      Briefly describe your sample data, data sources, and how you have insert data to your database with evidence.
   e. Use of the database
      [7 marks out of 44 marks of the part 3-part 5 assessment task]
      i. Design and implement queries.
      Briefly describe what you wish to know by using the query, why the results are important, and evidence of use of them (query implementation and sample outputs).
      ii. Design and implementation of advanced features
      Briefly describe the advanced features you have implemented, their use and evidence of your implementations/ outputs.
      iii. Database connectivity and Python implementation
      Briefly describe the database connectivity implementation and evidence of your implementation.
   f. Discussion [5 marks]
      Reflect on your own work, including a summary of what you have achieved, challenges you have faced, limitations and ways to improve your work with other features you have not considered, and any other information you wish to present.

Your report would be around 10-20 pages.

5. What you will be submitting

Your submission will be done in three steps.

Submission step 1:
You must submit a signed and dated assignment cover sheet (Declaration Of Originality sheet) in PDF format to the "**Assignment submission: cover sheet"** link provided in the assignment folder. A

blank assessment coversheet (Declaration of Originality) is available under the Assessments page of Blackboard. You can sign a hard copy and scan it or fill in a soft copy and digitally sign it.

Submission step 2:
Your documentation,

1. Your user guide (refer to section 4.1)
2. Your report on your database (refer to section 4.2),

should be submitted to the **Turnitin links (User Guide Submission: Step 2(a) and Report Submission: step 2(b))** provided in the assignment folder, respectively. Your documents submitted to **Turnitin links** and other links should be in PDF format.

Your report name must follow the following format:
"<Your Last Name as per Blackboard>< Your First Name as per Balckboard>_<your student ID>_DbSReport2024"

Submission step 3:

You should submit a single zip file of all the work produced in this assessment to the "**Assignment submission: step 3**" link provided in the assignment folder. First, create a folder with the name "<Your Last Name as per Blackboard>< Your First Name as per Balckboard>_<your student ID>_DbS2024"

Then, place all your work inside this folder. Example: *SuttonLiam_12134567_DbS2024*

The folder should contain:

1. Your SQL/database programming/ data files: You must submit all your .sql files or any other file resulting in parts 2-5 of the detailed description section. Name your files in an appropriate manner and they must be referred to in your user guide correctly.
2. Your user guide (refer to section 4.1) that has already been submitted to the Turnitin link.
3. Your report on your database (refer to section 4.2) has already been submitted to the Turnitin link.

You can organize your files in sub-folders so that it is well organized and easy to use.

Zip this folder and submit to the "**Assignment submission: step 3**" provided in the assignment page. Make sure that your zip file contains what is required. Anything not included in your submission may not be marked, even if you attempt to provide them later. You are responsible for ensuring your submission is complete and correct.

6. Demonstration

After submitting your assignment, a short demonstration (15-20 minutes) will be held, and the schedule will be announced later.

7. Marking guide

Your work will be assessed based on (a) code submission, (b) Report, and (c) Demonstration of your work. Look at section 5 of this document for submission instructions.

Different evaluation components will have marks allocated as per table 1 below. Detailed mark allocation for sub-sections can be seen in Table 2.

Table 1: Mark allocation for different submission/evaluation components

Evaluation component	Total mark
Design and implementation of SQL / programming parts (Part 2-5 of the detailed description), based on the work you have submitted to the "**Assignment submission: step 3**" link	40 marks
Documentation, based on the work you have submitted to the **"Assignment submission: step 2(a) and 2(b)**" links	40 marks
Demonstration of your work	20 marks

Table 2: Mark allocation for assessment tasks and how they are assessed.

Assessment task	Allocated Mark	Mark allocation for evaluation components			
		Demons tration	Code submission (Submission step 3)	Documentation (Submission step 2)	
				User Guide	Report
Designing a database and converting it to a relational schema.	18	-	-	-	18
Implement the database you have designed and load it with sample data.	16	5	8		3
Designing and implementing queries.	16	5	8		3
Increase the database functionality with advanced concepts.	16	5	8		3
Connect to the database using a suitable programming language and show sample query results.	12	5	6		1
Quality of the scripts (naming, professional code, comments, use of scripts, etc.)	10	-	10	-	-
Instruction to use database/queries.	5	-	-	5	-
Introducing your work and reflection	7	-	-	-	7
Total	100	20	40	5	35

8. Requirement to pass the unit.

As specified in the unit outline, you should score at least 40% of the final assessment to pass the unit. This assignment is your final assessment; therefore, you need to get at least 40% of the marks of this assignment to pass the unit.

Exact mark breakdown in Sections 3 - 4 and Section 7 of this document represent maximums, achieved only if you completely satisfy the requirements of the relevant section.

Plagiarism is a serious offence. This assignment has many correct solutions so plagiarism will be easy for us to detect (and we will). Please read the *Coding and Academic Guidelines* on Blackboard and for information about plagiarism, please refer to http://academicintegrity.curtin.edu.au

In summary, this is an assessment task. If you use someone else's work or someone else's assistance to help you complete the part of the assessment, where it's intended that you complete it yourself, you will have compromised the assessment. **You are prohibited from obtaining the support of AI tools when attempting this assessment**. Refer unit outline for more information on this aspect. You will not receive any marks for any parts of your submission that are not your original work. In the case of doubt, you may be asked to explain your code and the reason for choices that you have made as part of coding to the unit coordinator. A failure to adequately display knowledge required to have produced the code will most likely result in being formally accused of cheating. Finally, be sure to secure your code. If someone else gets access to your assignment for any reason (including because you left it on a lab machine, lost a USB drive containing the code or put it on a public repository) you will be held partially responsible for any plagiarism that results.

9. Late submissions

You must submit the assignment on the due date. Acceptance of late submissions is not automatic and will require supporting documentation proving that the late submission was due to unexpected factors outside your control.

Note that external pre-scheduled commitments including, but not limited to, work, travel, scheduled medical, sporting, family or community engagements, are not considered for unexpected factors outside your control. If you know you have, or are likely to have such engagements and that they may affect your ability to complete the assignment, you will be expected to have planned your work accordingly. This may mean that you need to start and/or complete your assignment early to make sure that you are able to hand it in on time. Also, note that IT-related issues are almost never a valid excuse.

Refer to the unit outline to know how the penalty will be applied in case of late submissions. As per the university policy, any submission made after seven (7) calendar days of the due date will not be marked, and you will automatically fail the unit.

Note that if you are granted an extension, you will be able to submit your work up to the extended time without penalty – this is different from submitting late.

10. Clarifications and amendments

This assignment specification may be clarified and/or amended at any time. Such clarifications and amendments will be announced in the lecture and on the unit's Blackboard page (not necessarily at the same time and not necessarily in that order). These clarifications and amendments form part of the assignment specification and may include things that affect mark allocations or specific tasks. It is your responsibility to be aware of these, either by attending the lectures, watching the iLecture and/or monitoring the Blackboard page.

11. General instructions

- **Only MYSQL must be used as the DBMS when implementing the assignment tasks.**
- **All coding/ MySQL coding must be done in a Linux environment using a command prompt.**
- Remember to start small and build upon what you have already done. If you spend more time thinking and designing what you would do rather than quickly trying to implement something, you will be able to do much better in this assignment.
- This assignment is open to be expanded and includes complex concepts. However, it would be a good practice not to make your scenario too complex (or too simple). Think about the total mark allocation, the time you would spend on each section and mark allocation for each section very carefully.
- **You may include more useful functionality than mentioned in the detailed description (section 3) above to your database for additional (bonus) marks**. For example, you may add or modify table columns using queries or use Python3 to create the database and retrieve data, etc. If you add such additional functionality, make sure they make sense and discuss them in your report also.
- **All your SQL code/ programs should be commented on to explain what each query/ section does and how the section works.**
- Doing the design of the database on paper would be much easier than doing it in a computer screen. It will help you to “think aloud”, make different decisions and then refine your diagrams in an iterative manner. You can convert your ER diagram to a computer-drawn diagram using MS Visio or another software if you wish, only after refining it to make it better.

End of the Assignment.