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

In this unit, you have learned about how to design a database system using ER modelling, convert the design to a relational schema, and implement your schema in a database by writing SQL DDL statements. Further, you have learned about various SQL DML statements to insert and update data, retrieve data fulfilling different needs using basic queries, joins and sub-queries. You are introduced to advanced features such as stored procedures. You will soon learn how to use SQL statements inside a programming language environment and transactions also.

This assignment expects you to apply the all 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 in effective manner, and(c) document your database design, implementation and query designs, results you have obtained and discuss your work.

2. Scenario

Scenario A: Cricket is a sport popular in many parts of the world, mostly in commonwealth countries. Different international cricket tournaments are held annually with matches between countries or other forms of teams. Test matches, ODIs and T20 are some broad forms of cricket. Assume you or your friends are interested in Cricket matches and wish to know about different international tournaments (series), teams, players, scores, winners, stadiums different matches held etc. and all interesting things about cricket matches. You may concern about one form of game like T20 series or several forms, in either case covering many years.

Scenario B: Film festivals around the world are a platform to showcase cinematic and popular films and appreciate the people in the film industry. There are many film festivals happening around the world. Cannes Film Festival and Cairo International Film Festival are some examples. Film festivals recognize the people in the industry with several categories of awards including actors, directors and technical people as well as the films itself. Assume you or your friends are interested in film festivals around the world (either one large festival or several festivals) and wish to know about festival location, awards given in each year, who got awards, award winning film's details etc. and all interesting things about film festivals.

3. Detailed description

Considering ONLY ONE of the given scenarios, you are expected to do 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 Model: Identify entities and attributes, relationships, cardinality, and participation constraints and design the ER diagram. The ER diagram should be drawn following the Chen's notation used in lectures. Think about different forms of entities, attributes and relationships which can be useful in your design so that you can show your ability to model diverse scenarios properly.
b. Define the relational schema:
Convert your ER model to a relational schema by doing suitable relational mapping. You may do this iteratively, starting with first identify basic tables, attributes and then refining it to convert more complex relationships in the design.
All your tables should be in at least first normal form but if you have done the design and mapping correctly, your tables would be in third normal form.
You have to think about the constraints such as primary and foreign keys. You may add any other constraints deemed required also. You are expected to improve the preliminary work by adding suitable referential integrity constraints.
c. Data description: Then, decide on suitable data types and any attribute level constraints (such as NOT NULL) for attributes and show them 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 business rules you assume important also must be defined here.

Part 2. Implementing the database, you have designed: [16 marks]

a. Looking at the relational schema and the data description resulted 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>, and then implement all tables there.
b. Insert sample values to the 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 in 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 to the database so that the results of the queries you will design in next sections can be obtained.

Part 3. Designing and implementing queries: [12 marks]

a. 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, several tables, obtaining data based on conditions, string manipulation etc. (there are many other aspects you can think about).

Then, for each of your question, design and implement an appropriate SQL query to produce the required answer in effective manner.
You should demonstrate that you can,
Level1:
Use basic SQL SELECT statements, with use of numeric data, date-time functions, string comparison and manipulation, and other related basic methods with suitable WHERE clauses.
Level 2:
Use joins and sub-queries, with GROUP BY, ORDER BY, aggregate functions and related clauses.

Part 4. Increase the database functionality with advanced concepts: [10 marks]

Use stored-procedures, triggers, views and indexes to improve the capabilities of your database. Design and implement at least *two* categories of above advanced features and show several uses of them. For example, you can have several stored procedures and few triggers.

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

In this section of the assignment, you are expected to demonstrate your ability to connect to a MySQL database with Python3 and use it in Python3 environment.
You are expected to connect your database to Python3 and call your already defined queries via the python programming constructs to show that you can do some useful database activities.

For part 2- part 5 above, SQL statements are to be written. Overall use of proper SQL (comments, readability of code, styling, good practices etc.) is also expected [12 Marks]

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 self-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 specifically state this.

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 [6 Marks]
   In this document you are expected to describe clearly how can the database you have designed / implemented can be implemented in a MySQL server and then use it to run the queries you have developed. You have to 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 have to provide the commands and may use screen shots also.
   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

cover page indicating assessment name, report title, your name, your Curtin student ID, lab group. User guide would be 3-10 pages.

2. Report on your database [34 Marks including Design section's 18 marks]
   Your report should include following sections:

   a. Cover page
      Include the assessment name, your name, Curtin student ID, practical class (date / time/ lab number)
   b. Introduction
      Short overview of the wok you have done, including the selected scenario and activities you have done.
   c. Design of the database
      i. Explanation on 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
      Briefly describe of how you have implemented the database with evidence of implementation.
      Briefly describe your sample data and how you have insert data to your database.
   e. Use of the database
      i. Design and implementing of queries
      Briefly describe what do you wish to know by using the query, why the results are important, 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, evidence of your implementations/ outputs.
      iii. Database connectivity and Python implementation
      Briefly describe the database connectivity implementation, and evidence of your implementation.
   f. Discussion
      Reflect on your own work including summary of what you have achieved, challenges you have faced, limitation 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 **two** steps.

Submission step 1: Your documentation,

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

should be submitted to the **Turnitin links (User Guide Submission: Step 1(a) and Report Submission: step 1(b)** provided in the assignment folder respectively. Your documents submitting to **Turnitin links** and other links should be in PDF format.

Submission step 2: You should submit single zip file of all the work produced in this assessment to the "**Assignment submission: step 2**" link provided in the assignment folder. First, create a folder with name < YourFirstName><YourLastName>_<your student ID>. Then place all your work inside this folder. Example: *JohnWhite_12134567*

The folder should contain:

1. Your SQL/database programming/ data files: You must submit all your .sql files or any other file resulted in part 2-5 of the detailed description section. Name your files in appropriate manner and they must be referred in your user guide correctly.
2. Your user guide (refer section 4.1) already submitted to the Turnitin link.
3. Your report on your database (refer section 4.2) already submitted to the Turnitin link.
4. A signed and dated assignment cover sheet in PDF format. Assessment coversheet is available under Assessments page of Blackboard. You can sign a hard copy and scan it in or you can fill in a soft copy and digitally sign it.

You may 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 2**" 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. It is your responsibility to make sure that your submission is complete and correct.

6. Marking rubric

Marks will be given out of 100 and will contributed to 50% of the unit's total assessment marks as specified in the unit outline. Marks will be given for your code submission, demonstration of your work and for the documentation.

	Total	Description
Design and implementation of SQL / programming parts (Part 2-5 of the detailed description), based on the work you have submitted to "**Assignment submission: step 2**" link	40 marks	Part 2 - Part5: 28 Marks, Overall (comments, readability, proper database concepts, professional code etc.): 12 Marks
Demonstration of your work	20 marks	
Report and the user guide	40 marks	User guide: 6 Marks Report: Design section of the detailed description (Part 1) 18 Marks, Other sections: 16 Marks

7. 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.

Marks of the assignment is given out of 100 and the assessment is worth 50% of your final mark (overall mark of the unit).

Exact mark breakdown in Section 3 - 4 and Section 6 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 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 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.

8. 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.

In the event that you submit your assignment late and you will be penalised with a 5% penalty for 1 day late, then an extra 10% penalty for each calendar day late after that up to a maximum of seven (7) calendar days, as per the university policy. Any work submitted after this time 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.

9. 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.

10. General instructions

- Only MYSQL must be used as the database when implementing the assignment tasks.
- All coding/ MySQL work must be done in a Linux environment using command prompt.
- Remember to start small and build upon what you have already done. If you spend more time in thinking and designing what you would do, rather than quickly try to implement something, you would be able to do much better in this assignment.
- This assignment is open to be expanded and include complex concepts, however, it would be a good practice to not make you scenario too complex (or too simple). Think about the total mark allocation, 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 for creating the database itself and then 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 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 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