Database Systems – ISYS1001/ISYS5008 2021 semester 2

Assignment Specification -V1

Learning objectives

1. Apply the knowledge related to design, implementation and usage of 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 if any based on modern database concepts.

1. Introduction

You have learned about designing a database system with the use of ER modelling, and then how to convert the design to a relational schema. You have learned how to add suitable constraints to make the tables you have designed to work as intended, while maintaining the integrity of the database. Similarly, you have learned about how to implement the tables you have created by writing SQL DDL statements and, insert and update and retrieve data fulfilling different needs using SQL DML statements such as basic queries, joins and sub-queries. You have learned about advanced features such as stored procedures and you will soon learn how to use SQL statements inside a programming language environment and transactions also.

This assignment expects you to apply the knowledge you have gained via lectures, practical classes and practical tests throughout the semester 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 an effective manner, and (c) document your database design, implementation, and query designs, and the results you have obtained, reflecting on your own work.

2. Scenario

The Olympic Games is a global event, held every four years, where individuals and teams from hundreds of countries around the world would participate. Since 1960, the Paralympics have also been held in the same year as the Olympics. The 2020 Olympics recently concluded. Assume that you or your friends are interested in Olympic Games (either Olympics or Paralympics) and need to know about the details of games, participants, winners, and all interesting things about this event.

3. Detailed description

Considering the given scenario, you are expected to do the following tasks in this assignment:

1. **Design a database and the relational schema based on real world scenario: [20 marks]**
    a. You are expected to identify entities and attributes, relationships, multiplicity and participation constraints and design the ER diagram. The ER diagram should be drawn following the Chen's notation used in the lectures OR IE notation shown in the lecture slides.
    b. Then, you need to decide on suitable data types and any attribute-level constraints (such as NOT NULL). You have to define your attributes in a suitable tabular format with at least the name of the attribute, selected data type, description of the attribute and any constraints on attribute values.
    c. Define the relational schema:
    You may do this iteratively, starting with first identifying basic tables and attributes, 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 also may add any other constraints deemed required. Then you may improve the preliminary work by adding referential integrity constraints.

2. **Implementing the database: [10 marks]**
    a. Looking at the relational schema and the data description resulting from part 1 above, you will implement the database with suitable tables and constraints.
    First, create a sample database with the name <suitable name>_<your studentID>, and implement all tables there.
    b. Insert sample values to the database and demonstrate that the integrity constraints are met when entering data.
    NOTE: Web links to obtain sample data relating to the given scenario will be mentioned in the assignment page. You have to enter a reasonable amount of data only to the database so that the query results can be obtained in next sections.

3. **Designing and implementing queries: [10 marks]**

    When your database is up and running, it is time to retrieve data to answer some reasonable queries. First, think through the real-world scenario and derive some meaningful questions (around 5-10 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 of).

    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:

        Level 1: 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.

4. **Increase the database functionality with advanced concepts: [10 marks]**
   You can use stored procedures, triggers, views and indexes to improve the capabilities of your database. Design and implement at least *two* categories of the above advanced features. For example, you can have several stored procedures and a few triggers.

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 abilities to connect to a MySQL database with Python3 and use it in Python3 environment.
   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.

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, either in the code and/or in the report.

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**
   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. 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 organise your document. User guide would be 3-10 pages.

2. **Report on your database**
   Your report should include sections on:

   a. Introduction
   b. Design of the database
      i. Explaining why you have selected the entities, relationships, data types etc.
      ii. ER diagram, relational schema 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.
   c. Implementation of the database
   d. Design of queries and use of the database
      i. Design and implementation of queries, why they are used, sample outputs.

   ii. Design and implementation of advanced features you have selected to implement and why they are important, and sample outputs
   iii. Database connectivity and any Python implementation
e. Discussion of your work including challenges, limitation and future directions (ways to improve your work with other features you have not considered).
f. Any other things you wish to present in addition to above.

Include a suitable cover page indicating your name, student ID, lab group (i.e. day and time) and the document title to both of the documents. Your report should 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 4.2)
2. Your report on your database (refer section 4.1),

should be submitted to the **Turnitin link** provided in the assignment folder. Your documents 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 *Assignment_<your student ID>.* Then place all your work inside this folder.

The folder should contain:

1. Your SQL/database programming files: You have to submit all your .sql files or any other file resulting from part 2-5. Name your files in an appropriate manner and ensure they are correctly referred to in your user guide.
2. Your user guide (refer section 4.2) and your report on your database (refer section 4.1) already submitted to the Turnitin link.
3. A signed and dated assignment cover sheet. The Assessment cover sheet is available under the 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.

Zip this folder and submit to the ***"Assignment submission: step 2"*** provided on 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 it later. **It <u>is your responsibility</u> 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.

Design and implementation of SQL / programming parts (Parts 2-5 of the detailed description), based on the work you have submitted	40 marks
Demonstration of your work	20 marks
Report and the user guide	40 marks **Part 1 design documents (20 marks) will be part of the report**

7. Requirement to pass the unit

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

The assignment is marked out of a total of 100 marks and is worth 50% of your final semester mark (overall mark of the unit).

The exact mark breakdown in Section 3 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). For information about plagiarism, please refer to http://academicintegrity.curtin.edu.au

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 are deemed to have a valid excuse, you will be penalised 10% (that is, 10% out of 100%, not out of what you would have received) per calendar day that you are late, up to a maximum of seven (7) calendar days. 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. Piazza has also been made available to you if you wish to ask questions about this specification or do not understand what is required in a section.

10. General instructions

- 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 your scenario too complex (or too simple). Think very carefully about the total mark allocation, the time you would spend on each section and the mark allocation for each section.
- **You can include more useful functionality than mentioned in the detailed description (section 3) above to your database for additional (bonus) marks**. For example, you can 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 also discuss them in your report.
- All of your SQL code / programs should be commented to an appropriate standard to explain what each query or 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, after refining it to make it better.

—— End of the Assignment ——