DAT602 – Assessment One
Journal

Daniel Waller : 158 758 567

Contents

Journal One – Course Introduction

Today's class covered the topic overview, schedule, and assessments. The learnings in class are to be demonstrated in the project assessment and journal. The journal is to document your observations and learnings during the course. It should cover reflections on how we are designing and learning how to develop database applications.

We have been working in teams to produce slides documenting our understanding of the project and how it will be assessed. Below is a general overview that was discussed in class.

ASSESSMENT OVERVIEW

- **Assessment One**
  - Journal
  - Insight and learnings documented each week - 10%
  - **DUE**: NOW, Feedback throughout the semester and due at the end of the course

- **Assessment Two**
  - A project, three phases with reports - 90%
- **Phase One – 25%**
  - Analysis and Design
  - Storyboards, CRUD
  - Phase One **DUE**: 24 March 2021
- **Phase Two – 30%**
  - Database Procedures
  - Create a database and procedures with console for the application
  - Phase Two **DUE**: 19 May 2021
- **Phase Three – 35%**
  - Front End
  - Create a prototype GUI with .NET data grids
  - Phase Three **DUE**: 23 June 2021

MY GAME OVERVIEW

The game is Wizard Quest. Players move around the map; as they do so, more of the world will be uncovered. The main aim is to collect gold rings to gain points. The map has several other items for players that can assist in their quest. Players can set up an account, play, and communicate with other players in real-time.

Additionally, players will be able to save their games, delete their accounts, and administrators will be able to assist players with account and game issues.

LESSON TWO REVIEW

Today the tutor focused on assignment two—specifically, the game description and what it might do. Additionally, we were provided with a game example and a sample report to assist with expected deliverables and presentation for grading. These elements will be essential to ensure a logical and systematic approach to each assignment phase previously discussed.

An overview of storyboarding was discussed using the example report. The focus here is to ensure an organised, well-illustrated sequence. Ensuring well labelled and communicated process to ensure

the readers understanding. Storyboarding is an integral part of phase one, a variety of techniques and software solutions were discussed to achieve the results required. I am exploring Adobe XD as a possible solution.

CRUD was introduced. Key points included how CRUD relates to SQL:

- Create = Insert
- Retriever = Select
- Update = Update
- Delete = Delete

The tutor demonstrated two different ways to design a CRUD table and work with assignment two. A CRUD table and storyboarding can be attempted as part of phase one in assignment two.

Finally, we briefly explored MySQL, working through a tutor example and creating procedures to run specific statements. Procedures will form the basis of many of the component in our game construction. Further exploration and work are to be completed on MySQL and procedures to assist with our knowledge and practical application towards completing assignment two.

Journal Two – Relational Databases

Today's lesson revied relational databases. Relational databases are data sets laid out in a logical, systematic manner not to be repeated or doubled in anyways.

- Each record is a mathematical tuple. Consider a row in an excel spreadsheet.
- Parts of the record are fields that are the columns.
- Logical sets are related to each other through **key** values.
- Each data value is held only once for data independence, storage efficiency, processing, user-friendliness, and industry standards.

SQL - DDL

Additionally, we covered revision on SQL DDL, data definition language and covered examples on creating tables in MySQL via a PowerPoint presentation. During this time, we discussed datatypes and which types would be appropriate in a range of scenarios. When using DDL to create SQL data structures, it is essential to define constraints via primary, composite, and any foreign keys to ensure that data values are still only held once per the industry standard and efficiency metrics.

To practice DDL scripting in MySQL, I successfully recreated the student access database structure in the MySQL workbench application.

SQL-DML

Furthermore, we discussed SQL DML, data manipulation language and reviewed examples of input data through SQL insert statements. First, via a single line insert script and then a multiline model.

Additional DML scripts were reviewed, specifically the select statement and how it can be used to compile specific sets of data within a database by specifying what you would like to select, where you want it from, and conditions for that selection.

I have completed an SQL procedural script in MySQL Workbench that replaces the personnel access database we were given to replicate for practice.

ENTITY RELATIONSHIP DIAGRAMS
This week, we examined ERD's and how they relate to our game assessment in the second lesson this week. This will be useful in developing our diagrams. Exploring the concept of logical sets related via key values, we discussed primary and foreign key references.

JOINS
Following on from ERD's, we explored joins and how the primary and foreign keys are used in conjunction with the SQL join statement. Several different joins were then discussed, the most common being in the inner join the standard type when using the SQL join statement followed by left and right joins, respectively. Several examples we discussed in class will be the starting point for implementing procedures from our CRUD analysis and tables derived for our game.

ASSESSMENT PROGRESS
So far, I have completed a general description, and I am refining it whilst building the storyboards. I hope to have storyboarding finished over the weekend to begin working on an ERD and CRUD table.

Journal Three – SQL Stored Procedures, Flow Control, and Update

- Flow control allows the construction of functions and procedures within the MySQL database.
- Benefits such as IF statements can be leveraged and integrated directly into your database.

We covered an initial overview of stored procedures and how flow control would work a simple login procedure in class. This initial procedure can be adapted to the login process required for our game in project one.

Additional code such as IF statements to handle username validity, password, and login count will initially be considered. This learning will form the foundation in which procedural code will develop for our game with additional and more complex functionality.

UPDATE
In our second lesson, we investigated update statements and how they can be integrated into our procedural code.

Developing a more sophisticated procedure for login has started my thought process around how this can be achieved. While I have not been entirely successful at this stage, I am progressing towards a solution. Sharing our code with our classmates has demonstrated several different solutions can be utilised to achieve similar outcomes.

I can see how update statements will allow further database functionality. In our game, I can see update statements being used in several ways. I have a better understanding of how my CRUD table should look, and I will be reviewing it to ensure it is correct.

I now understand how an update statement could be used in a procedure to unlock an account or update a user record in the administration portal of my game. This learning has given excellent insight. I look forward to seeing how it works with C# integration.

PROJECT UPDATE
Storyboarding has been completed and written up for the final document. I am progressing well with my CRUD table and ERD. I have started a few test DDL statements in SQL to deploy my first iteration of the proposed solution efficiently.

Journal Four – Delete and C#

In our first lesson, the tutor introduced the delete statement. This statement can be used simply to delete all records from a table, utilising the delete from clause. Or with a specific type of record target with the addition of the where clause.

The concept is a simple one, and I have tried integrating it into a few test code lines in my DML statements for assessment two. I did have several issues with foreign key constraints and referential integrity errors.

Seeing how my DDL statement could be modified to ensure that delete statements remove all referenced data from any child tables as well via the cascade on delete as part of a referential key statement.

I have made minor modifications to my DDL to assist with delete statements in my database's first iteration.

C# AND MYSQL INTEGRATION
Our second lesson this week introduced MySQL and C# integration. We used a tutor-provided database and C# console code to connect our local instance of our database on our computers and visual studio. An essential critical step to ensure a connection is to make sure your database is loaded on your local instance and available in MySQL workbench. MySQL server extensions are loaded in visual studio.

In MySQL, we discussed adding a user with a password to the database and granting access to the entire database via the grant all statement. Additionally, the safe mode needs to be off for advanced functionality to work correctly.

Visual studio access MySQL via MySQL.Data.MySQLClient. Access to your specific database can be achieved via a DataSet method. The method contains the connection string and user details to connect and pass data between C# and your database. This is achieved by using the MySQLHelper statement in C#.

We achieved a successful connection in class between the test database and the visual studio console app. We were able to implement the AddUserName procedure and see how it functioned via debug using breakpoints. This allowed visibility of data being passed through the application. Seeing this made it easier to understand what each of the DataSet method lines was achieving at what time.

Trying to implement additional procedures in the database in C# has helped improve my understanding further.

Journal Five – Milestone Review, Introduction, and Joins

This week's focus has been on completing milestone one assessment requirements. We have conducted an in-class review of the assessment and looked over an example of what the requirement report should look like and contain. Milestone two has been introduced, and requirements have been discussed, and there will be future sessions dedicated to the learnings required to complete it.

My focus has been to understand how joins work more proficiently as they are required throughout my game to function currently. I am now experimenting with my game player inventory, specifically joining it with an asset table to get descriptions and values of the player's assets and calculate their current score.

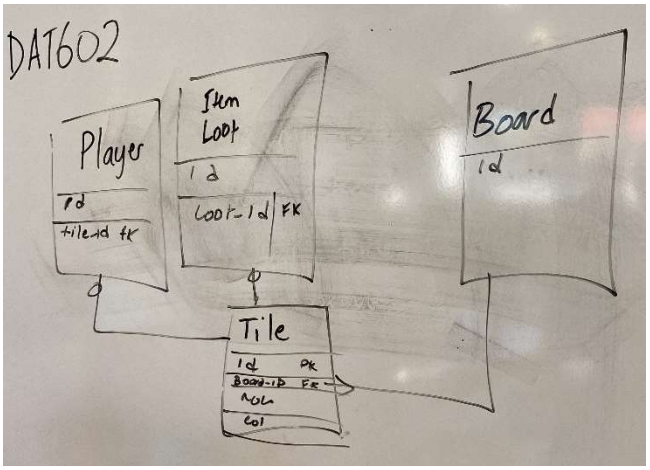
Journal Six – Group By Having

This week the concept of group by having was introduced. The tutor gave an overview in class and encouraged us to investigate this further in our own time. I completed the group by having author exercise provided and read that there are serval differences between MySQL and SQL Server in terms of functionality.

The author exercise provided insight into selecting logical sets of data and complete mathematical functions to extract relevant data. Understanding how these functions work in their entirety is beneficial as many businesses will have large databases with information stored on them. The data isn't helpful until functions such as group by having are applied to give relevant business information.

Journal Seven – Subqueries

This week, we reviewed an ERD of a simplified tile game like what it may look like in our game development. We focused on player, item, tile, and board tables.



In our first lesson, we reviewed how we could find a player location on a board using a select statement to find the row and column tile ID by joining the player and tile table where the specified player id is true.

Using this initial query, we can now extend upon the learning and devised a select statement that will find all the neighbouring tiles. Essentially, we can use the code from the previous solution and create another copy of the tile table, joining them together where each tile does not equal the player ID. Meaning, there is now a table of board tiles that are only tiles that the player is not currently on. Further selection constraints via a where statement means the result will only show tiles that a valid to move to. The tiles directly above, below, and to each side in a three-by-three grid except the tile currently occupied by the player.

In our second lesson, we extended upon our previously created statements to create a statement that would check to ensure the new tile location selected by the player is a valid location. That is, the tile selected in a neighbouring tile. This was achieved by creating an initial select query with the proposed tile row and column selected. Then through the use of a subquery within the statement, the code previously developed for selected neighbouring tiles is used. If the proposed move and the neighbouring tile data match, the move is a valid one. If not, a null result would be achieved.

Lastly, we developed an update statement to updated the player's tile location. This was achieved by nesting the previous move checking statement as a subquery inside an update statement for the player's tile ID.

It brought all the learning together and developed a procedure that encompasses all the previous statements to form a move procedure for potential use in our game development—learning subqueries while revisiting ERD, and select statements have assisted with further understanding and valuable learning while developing statements for milestone two.

The tutor has also introduced common table expressions via an updated SQL query script shared later in the week. I am currently reviewing and can see that assigning aliases to subqueries makes the code easier to understand and work with. This could be beneficial to also integrate into milestone two work.

Journal Eight – More Subqueries, ACID, Milestone Review

This week the tutor discussed subqueries further, specifically, subqueries with multiple conditions such as ANY, IN, SOME, ALL, ROW, EXISTS, and NOT EXISTS. Additionally, correlated subqueries and derived tables were also discussed.

While I have started to use EXISTS in some of my current select statements in my work, I can see how some of these other conditions could be highly beneficial in data manipulation.

Milestone Two was discussed in terms of requirements, and the tutor provided a demonstration of a console app. We will go over the code in the demonstrated data access class next week to have some class time to experiment with it.

The concept of ACID (Atomicity, Consistency, Isolation, Durability) was discussed. We have been encouraged to do our reading on the topic, and documenting our ACID knowledge is a requirement for milestone two. ACID is the concept that each transaction should be atomic, meaning the whole thing together. This is important in databases when multiple users are involved as you do not want data that each user retrieves or adds to be invalid or cause corruption to your data.

In terms of our game development, we need to consider if the procedures we have developed are atomic and complete the expected result.

The tutor also gave a short lecture on locking select statements with the use of LOCK and SHARE. Locking is essential when the integrity of data needs to be preserved during the transaction. So other uses can access the locked data until the current transaction is completed. This offers another level of atomic protection. However, it is crucial to note locking in some cases can lead to your database being jammed as a lock procedure that isn't appropriately designed could cause significant impacts to access and usability. It is essential to plan and use locking in the right circumstances.

MySQL has a transaction management system that handles all the requests sent to the database. A demonstration video of how the transaction management system processes data via the isolation level. Isolation levels read committed, read committed, repeatable read, and serializable.

Different isolation levels directly affect database efficiency and data integrity. The more integrity placed on the database reduces the efficiency. My initial reading research on the topic suggests an isolation level of read committed will give the best ratio of consistency versus efficiency for my assessment.

Journal Nine – LINQ

The tutor introduced the LINQ (Language Integrated Query) queries in C#. LINQ operates with the IEnumerable interface and can manipulate data in a similar way MySQL does. It is essential to use LINQ in the right circumstances when manipulating data from another source. The logic should be centrally contained in the database, whereas LINQ can extend the logic created. Essentially, LINQ bridges the gap between application and database.

Reading the provided Microsoft Docs shows how LINQ can complete complex calculations and obtain specific data. LINQ uses a combination of C# variable code and statements similar to MySQL, such as SELECT, FROM, and WHERE. Importantly the application of these statements is slightly different to that of MySQL.

The tutor provided an example of how LINQ works with a MySQL database and an add user and get all user list in C#. This is extremely useful as it will assist with creating the console application required. We have been given time next lesson to experiment with LINQ and try it with our databases. I have had several issues with my code in my initial attempts. I have found that specific code varies from each release of .NET. I switched to the .NET framework rather than the new .NET core as I was having issues getting LINQ and my database to work together in my specific environment.

Journal Ten – Error Handling and Rollback

The tutor gave a short review lecture summarising the importance of transaction isolation before introducing error handling, auto-commit settings and rollback. Rollback means if a transaction fails before it completes, it will roll back the data to a specified rollback location. For example, if the transaction includes some update to the database but fails after the update, it will roll back the changes. Rollback is extremely useful for data integrity as none of the data is changed if there is a problem during the transaction.

The tutor provided an example of a rollback procedure in MySQL where an add user transaction would fail as the username was not unique. The addition of the start transaction followed by commit at the end indicates the defined rollback zone for which the preceding rollback statement would function.

Additionally, error handling was explored. Error handling is highly beneficial in development as it allows you to understand issues your current procedures have when manipulating data. The tutor demonstrated the application of diagnostic conditions and RETURNED_SQLSTATE to provide better information about the error and assist in debugging. I was able to use the code provided to develop a diagnostic procedure for error handling. This would help understand errors efficiently in the future.

Journal Eleven – Data Grids

The tutorial on data grids brings all the learning previously about LINQ and C# together and allows completion of the milestone three GUI game application. Again, issues with consistent code mean I have opted not the .NET framework rather than the new .NET core release. As previously discussed, I have had functionality issues.

The tutor provided code and a demonstration of a small data grid where it obtains users and allows editing on another form. Important to note that the current refresh method is not functioning correctly and only refreshes data contained in the C# user class rather than the database. The tutor left us with the issue to explore how we might achieve this.

I used the tutor's code to understand how data grids and populate. In my development, I decided to forgo the update of the user's details in the user class in C# but instead wait for a response from the database as to the transaction result. If successful, the edit user form closes and refreshed the data grid with the new data directly from the database.

The tutor also reminded us of the importance of user privileges, specifically for the procedure and function access control.