IFB105 Database Management
Semester 1, 2022
Assessment task 3B – SQL

Name ……………………….	Assessment 2 - SQL
Due ………………………..	Sunday 12 June 11:59pm
Weight …………………….	30%
Type ………………………	Individual
Submit …………………….	ZIP file containing 2 SQL scripts, via Blackboard

Rationale and Description

This assessment will give you an opportunity to apply the concepts and skills you acquire in the unit to a 'realistic' database design scenario and reflect on the data requirements of an organisation. You will use your knowledge from the lectures together with the techniques practiced in the tutorial sessions and apply both to a set of tasks that refer to some business concern.

Learning Outcomes

A successful completion of this task will demonstrate:

1. Knowledge of database design principles and SQL programming to understand and create a SQL database script based on a given relational map.
2. Knowledge of database query principles and security practices.

Instructions

This assessment must be completed individually. You will solve **two tasks**:

1. Create a database from a relational map
2. Query the Treasure Hunter database (including security tasks).

You must submit 2 SQL scripts within a ZIP file using the submission link in Blackboard:

3. SQL script (file extension sql) containing your solution to task 1
4. SQL script containing your solutions for tasks 2 and 3.

If you do not follow these instructions, you will be penalised 5 points!

Feedback

Feedback will be provided on **specific** questions related to the assessment during the tutorial times, and via MSTeams. Please note, that the teaching team will **NOT** respond to general "is this good enough" style questions or requests to pre-assess your work.

Late Submission

According to QUT policies, submissions past the deadline without an approved extension attached, will NOT be marked, and will thus achieve a mark of 0. Therefore, you are strongly advised to submit your report, even if it is only partially complete, by the due date.

Extensions

According to QUT policies, an extension to the due date of the assignment may only be granted on medical/compassionate grounds. Please see the Late assignments and extensions information on the unit's Blackboard Assessment page.

Reviews

We cannot remark assessments. All assessments will be returned with feedback explaining the reasons for the marks allocated. If you require additional feedback or clarification, discuss it with your tutor. If you believe that there is a component that has not been marked in accordance with the criteria sheet, you must identify it in writing to your tutor.

Academic Honesty

Any action or practice on your part which would defeat the purposes of assessment is regarded as academic dishonesty. The penalties for academic dishonesty are provided in the Student Rules. For more information consult the QUT Library resources for avoiding plagiarism.

Resources

The following resources may assist with the completion of this task:

- Refer to tutorial materials, Microsoft Teams, and any lecture videos.
- Refer to the recommended book.

Questions

Questions related to the assessment should be directed to the teaching team during the workshops or drop-in sessions, and also via MS Teams. The teaching team will not be available to answer questions outside business hours, nor in the hours immediately before the assessment is due.

Assessment tasks

Task 1 [7 marks]

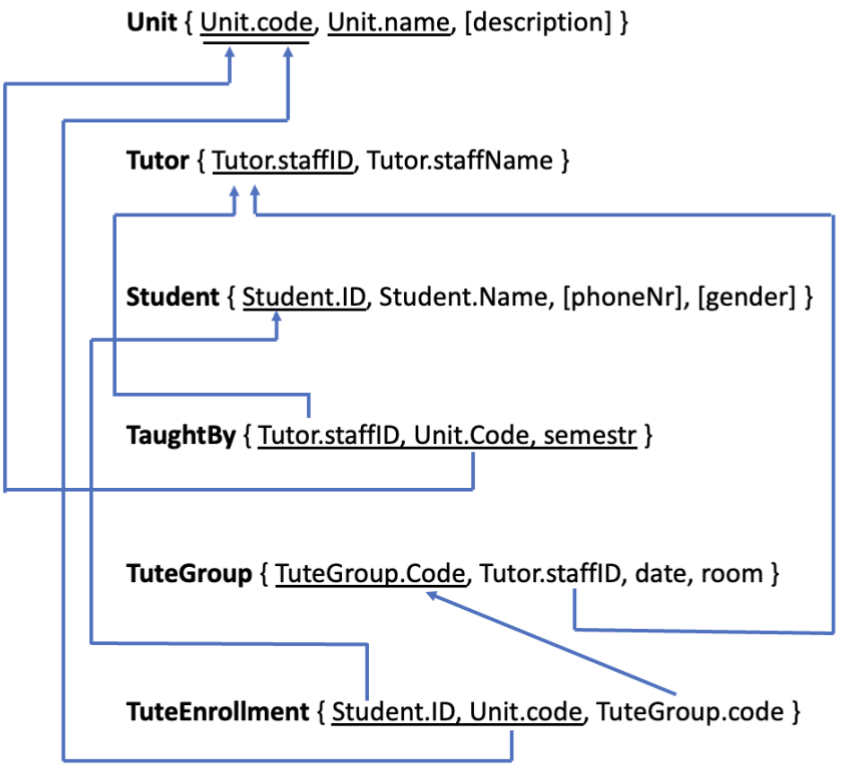
An SQL script is a set of SQL commands saved as an SQL file. If you are already working with MySQL, you can either execute an SQL script file using the source command or import it in Workbench.

Write an SQL script to create a database to match the Rmap provided below. ***Your script MUST execute in MySQL Workbench without errors to receive full marks.***

Marks will be awarded for the following:

1. Creating the database without error (1 mark)
2. Successfully creating new tables, including all attributes (2 marks)
3. Including constraints (1 mark)
4. Correctly creating Primary Keys (1.5 marks)
5. Correctly creating Foreign Keys (1.5 marks)

Rmap of a given relational schema to create a database for Task 1



Unit { Unit.code, Unit.name, [description] }

Tutor { Tutor.staffID, Tutor.staffName }

Student { Student.ID, Student.Name, [phoneNr], [gender] }

TaughtBy { Tutor.staffID, Unit.Code, semestr }

TuteGroup { TuteGroup.Code, Tutor.staffID, date, room }

TuteEnrollment { Student.ID, Unit.code, TuteGroup.code }

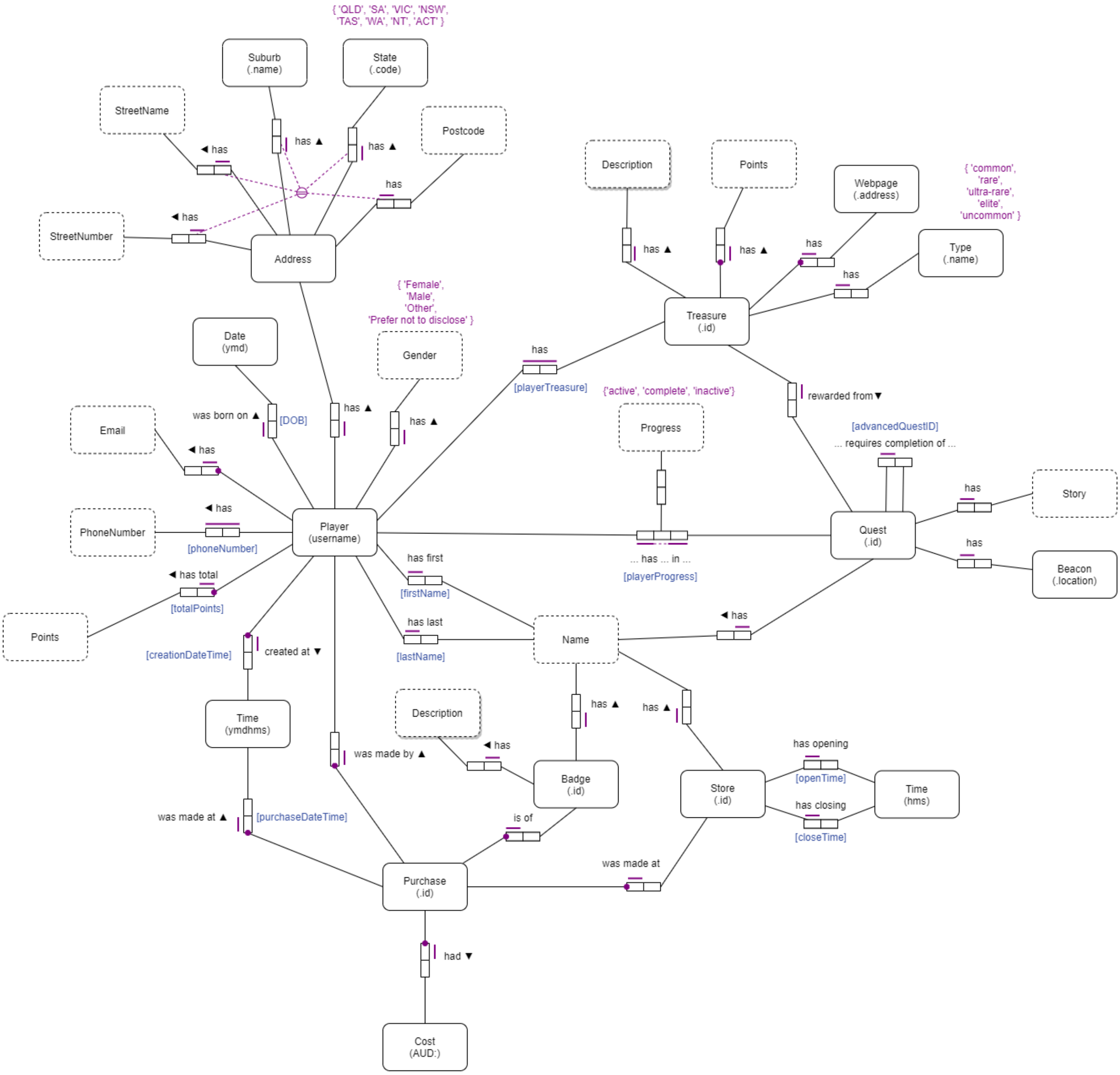
Notes

- Units are taught between in semester 1, 2, or summer
- Staff ID is a 7 digit auto incrementing number
- Student ID is a 7 digit auto incrementing number
- Unit codes are 3 numbers and then 3 letters
- Student and staff names include both first name and surname
- Day hour code is the full date and start time
- Tute group codes are the combination of numbers and letters
- The default room number for tute groups is 1 (which is the location of student services)
- Gender can be female, male, or other

Task 2 [15 marks] - Treasure Hunter database

For task 2, we have provided you with the creation script for the Treasure Hunter database. Run this script in MySQL Workbench to create the database. You should execute your query solutions to extract the necessary information.

The script is based on the following ORM:



Query 1 (1 mark)

Write a query to list the treasureID, description, points and types of treasures that contain either 'brick' or 'map' in their description.

Query 2 (1.5 marks)

Write a query to list the total number of treasures available for each treasure type. Your output should contain the type and the total of each type in ascending order of the number of types.

Query 3 (1.5 marks)

Write a query that displays the name, badgeID, and the cost of the most expensive badge.

Query 4 (3.5 marks)

Write a query that lists all badge sales. Your output should show the name of the badge together with first name, last name, and email address of the player(s) that made the purchase. Sort the list based on the badge name followed by first name then last name in ascending order.

Query 5 (3.5 marks)

Write a query that provides the players' name (first and last), username, and how many advanced quests they have completed. If a player did not complete any advanced quests, do not include them in your output.

Query 6 (4 marks)

Write a query to produce a report for each store, including stores without any sales.

Your result-set should include the following information:

- the storeID
- the store name
- the number of unique players that have purchased a badge from the store
- the number of unique players that have not purchased a badge from the store
- the total money spent at the store
- the most expensive badge a player has purchased at the store
- the cheapest badge a player has purchased at the store
- the average price of the items that have been purchased at the store.

Task 3 [8 marks] - Treasure Hunter database

Insert (1 mark)

Write an INSERT command to insert a row into badge table. The badge is called '*Summer Rain*' and the description should be '*Beach, sun and holidays*'.

Delete (1 mark)

Write a DELETE command to remove all the rows from the player progress table for which progress is complete.

Update (1 mark)

Write an UPDATE comment to change the address of all players with the last name '*Halpin*' who live at '*1800 Zelda Street, Linkburb*' to '*72 Evergreen Terrace, Springfield*'.

Create Index (1 mark)

Currently the database only contains a small number of records. However, the data contained within it is expected to grow significantly in the future. Creating indexes on commonly searched columns is a way performance issues can be minimized. Write a command to create an index on story column of the quest table.

Create view – 2 marks

Write a command to create a view which lists the firstname, lastname, and account creation date of all players that have started a quest but are currently inactive.

Task 4 [2 marks] - Security

Question 1 – 2 marks

Working as a Database Administrator for the MySQL Treasure Hunter database, write the following commands for two employees namely Catie and Manav to achieve the following database security requirements:

A. User Catie is no longer allowed to add data to the Player table (0.25 marks)
B. User Catie is no longer allowed to delete records from the Player table (0.25 marks)
C. User Manav must be able to add records to the Quest table (0.25 marks)
D. User Manav must be able to remove records from the Quest table (0.25 marks)

Assume usernames of employees namely Catie and Manav are *catie* and *manav* respectively.