IFB130 Databases
Project Part B

Project Overview
This IFB130 project gives 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.

The submission is divided into two parts due at different times during the semester. These parts will cover:

A. Design of a Database
B. Creation and Use of Databases

The Tasks for Part B
For Assessment 2B you will be required to:

1. Create of a database for the fictitious book store *Oktomook*
2. Query the *Treasure Hunter's* Database

Weighting
Part B is worth 25 marks, for **25%** of the unit.

Groups
You can complete this assignment individually or in pairs. You do not have to work in the same pairs as you did for Part A.

If you choose to work in a pair, only one student should submit the assignment. Please provide the name and student number of the person you worked with in the README.txt file. No consideration will be given to students who claim they did more work in their pair than the other student because this assignment can be done individually.

Doing the assignment in undeclared groups, or groups larger than two students, will be treated as plagiarism. Pairs that work together and then split due to difficulties must not submit any of the same work, or it will be treated as plagiarism.

Due Date
Week 12: Monday 16th of October at 11:59 pm

Submission
You need to submit 3 files for this assignment in a ZIP file:

1. SQL script (a text file with the file extension changed to sql) containing your solution to task 1 only.
2. SQL script containing your solutions for tasks 2, 3, 4 and 5.
3. README.txt containing your Full name, student number (and partners' details if applicable) and a list of any queries you have attempted but were not able to successfully run in Workbench.

Scripts in different file types will not be accepted. This needs to be uploaded to the submission link on Blackboard.

Late Submission
Assessment work submitted after the due date will be marked only with an approved extension (MOPP E/6.8.2). Assessment work submitted after the due date without an approved extension or, where an extension has been granted, after the extended due date, will not be marked and a grade of 1 or 0% will be awarded against the assessment item.

Administrative Issues
The unit outline sets out the requirements surrounding:

- Extensions (including for disabilities)
- Penalties for late submission
- Appeals.

Assessment 2B Tasks

Assessment 2B requires you to complete a number of tasks to fulfill the requirements of the scenarios. For this assessment, you will:

1. Build a script that will create a database for a given relational schema; (Week 6)
2. Provide the SQL commands needed to retrieve the required data using assessment 1a extended schema; (Week 7 to 9)
3. Provide the commands to modify (INSERT, UPDATE & DELETE) the data using assessment 1a extended schema; (Week 6)
4. Provide the commands needed to create appropriate indexes and views; (Week 6 to 9)
5. Provide advice on the basic security measures that should be implemented. (Week 10)

Task 1 [6 marks] Oktomook Book Store Database

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

Write an SQL script that builds a database to match the relational model provided to you. These SQL statements in the script must be provided in the correct order.

Marks will be awarded for the following:

1. Creating the database (1 mark)
2. Successfully creating new tables (1 mark)
3. Including all attributes (1 mark)
4. Including constraints (1 mark)
5. Correctly creating Primary Keys (1 mark)
6. Correctly creating Foreign Keys (1 mark)

You are required to create a database for the fictitious book store *Oktomook* for Task 1. The database is based on the model below:

OKTOMOOK RELATIONAL MODEL

Branch (branchNumber, branchName, streetNo, streetName, branchCity, branchState, numberEmployees)

Publisher (publisherCode, publisherName, publisherCity, publisherState)

Author (authorID, firstName, lastName)

Book (ISBN, title, publisherCode, genre, retailPrice, paperback)

Wrote (ISBN, authorID, sequenceNumber)

Inventory (ISBN, branchNumber, quantityInStock)

FOREIGN KEYS

- Book(publisherCode) is dependent on Publisher (publisherCode)
- Wrote (ISBN) is dependent on Book (ISBN)
- Wrote (authorID) is dependent on Author (authorID)
- Inventory (ISBN) is dependent on Book (ISBN)
- Inventory (branchNumber) is dependent on Branch (branchNumber)

OTHER CONSTRAINTS

- The domain of Publisher(state) is [QLD, VIC, NSW, WA, TAS, NT, SA].
- The domain of Book(genre) is [Non-Fiction, Science Fiction, Fantasy, Crime, Mystery, Young Adult, Romance, General Fiction]
- ISBN must be a 13-digit number and may begin with a zero.
- The publisher name and book title are both mandatory.
- Paperback must be either True or False.
- The default quantity in stock is 0.

Task 2 [11 marks] using the Treasure Hunter's database

For task 2, we have provided you with the creation script for the Treasure Hunter's database. You should run this script in MySQL Workbench and use this database to extract the necessary information.

The script is based on the following schematic:

TREASURE HUNTER'S RELATIONAL MODEL

Player (username, firstName, lastName, gender, DOB, email, streetNo, streetName, suburb, state, postcode, creationDateTime, totalPoints)

PhoneNumber (phoneNumber, username)

Treasure (treasureID, description, points, webpage, type, questID)

Quest (questID, questName, story, beacon, advancedQuestID)

Store (storeID, storeName, openTime, closeTime)

Badge (badgeID, badgeName, badgeDescription)

PlayerProgress (questID, username, progress)

PlayerTreasure (username, treasureID)

Purchase (purchaseID, storeID, username, badgeID, purchaseDateTime, cost)

FOREIGN KEYS

- PhoneNumber (username) is dependent on Player(username)
- Quest (advancedQuestID) is dependent on Quest(questID)
- Treasure (questID) is dependent on Quest (questID)
- PlayerProgress (questID) is dependent on Quest (questID)
- PlayerProgress (username) is dependent on Player (username)
- PlayerTreasure (username) is dependent on Player (username)
- PlayerTreasure (treasureID) is dependent on Treasure (treasureID)
- Purchase (storeID) is dependent on Store (storeID)
- Purchase (username) is dependent on Player (username)
- Purchase (badgeID) is dependent on Badge (badgeID)

OTHER CONSTRAINTS

- Player (gender) must be female, male, other or prefer not to disclose.
- Player (state) domain is [QLD, SA, TAS, NSW, WA, NT or ACT].
- Treasure (type) domain is [common, uncommon, rare, ultra-rare or elite].
- Players may enter up to three phone numbers.
- Players must enter at least one phone number.
- PlayerProgress (progress) domain is [active, inactive or complete].
- Player (email) is mandatory.

Query 1 (1 mark)

Write a query to list the name (first and last), date of birth, gender and email of players who live in Sunnybank or Sunnybank Hills. Note that you can assume these are the only suburbs starting with 'Sunnybank'.

Query 2 (2 mark)
Write a query to find out how much each player has spent at the stores. Your output should be sorted by username in descending order.

Query 3 (2 marks)
Write a query that lists the username and phone number of the oldest player. Note that you must use subqueries for this query.

Query 4 (3 marks)

Write a query that lists all of the badges. If the badge has been purchased include the first name, last name and email address of the player(s). Sort the list based on the badge name followed by first name then last name in ascending order.

Query 5 (3 marks)

List the number of quests embarked upon by each player with progress status 'complete' for the treasures that are common.

Task 3 [3 marks]

Insert (1 mark)
Write an INSERT command to insert a row into badge table. The badge is called 'Fools Gold' and the description should be 'Trickiest trickster in all the seas'.

Delete (1 marks)
Write a DELETE command to remove all the rows from the player progress table for which progress is inactive.

Update (1 mark)
Write an UPDATE comment to change the address of all players with the last name 'Smith' who live at '180 Zelda Street, Linkburb' to '72 Evergreen Terrace, Springfield'.

Task 4 [3 marks]

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 webpage of the treasure table.

Create view – 2 marks
Write a command to create a view to list the firstname, lastname and account creation date of any players that haven't completed any quests.

Task 5 [2 marks]

Treasure Hunters Company have two employees, Nikki and Phil, to work with the MySQL database. Working as MySQL database administrator, provide the commands required to grant or revoke access so the following security requirements are met:

A. User Nikki must be able to add records to the PLAYER table (0.5 Marks)
B. User Nikki must be able to remove records from the PLAYER table (0.5 Marks)
C. User Phil is no longer allowed to add data to the PLAYER table (0.5 Marks)
D. User Phil is no longer allowed to delete records from the PLAYER table (0.5 Marks)

Assume usernames of employees Nikki and Phil are *nikki* and *phil* respectively.