Lab 1 Part 2

Database Design

Overview

Database Design for a dynamic web application

Relational Data Modelling,

Learning objectives

- Describe how objects and their properties are represented in a relational database
- Depict a relational database schema in an Entity-Relationship (ER) diagram
- Identify and select appropriate field types for storing data in a MySQL database

Context

In the year 2 Dynamic Web Applications, you were working on a 'Book shop' application called Berties Books. You creates a database table to hold your list of books. Here's a reminder of the SQL you ran to create this table:

```
CREATE DATABASE myBookshop;

USE myBookshop;

CREATE TABLE books (id INT AUTO_INCREMENT,name VARCHAR(50),price DECIMAL(5, 2)
unsigned,PRIMARY KEY(id));

INSERT INTO books (name, price)VALUES('database book', 40.25),('Node.js book',
25.00), ('Express book', 31.99) ;
```

Let us suppose we wanted to develop the app functionality further and we have a very large book database...

New functional requirements:

- We want users of the app to be able to search for books, publishers, authors containing specific words.
- We want users of the app to be able to search for books based on publication date.
- We'd like the most recent published books to appear highest in the list by default, but for the user to be able to change the sort criteria.
- We'd like at least two types of users for the app, customers to search for books reading data related to books from the database and admin users to create, update and delete data as well as reading data from the database.

- We'd like a user to be able to click on a book and see all data relevant to that book as well as checking the stock availability of a book.
- We'd like admin users to be able to update stock availability when new books arrive at the store or when books are sold to customers.

How readily does the current data storage solution facilitate these basic functional requirements?

After completing Tasks 1-6 you need to **upload your design documents** or notes on the VLE as explained in Task 7.

Tasks

Task 1: The application concept

List all functionalities of the app you have developed so far and the functionalities listed above in addition to any new functionalities you would like to add to your app.

Task 2: List all basic functions the app will perform

Make your basic functions very specific, i.e.:

Basic function 1: Display a list of available books in alphabetic order. The author's name will appear next to the name and price of the book and ...

Refer to the functional requirement list above to list basic functions, Each functional requirement listed above may contain more than one basic function.

You may find sketching a simple wireframe helps with this part of the task.

Task 3: Object Identification

In a relational database schema, every type of object or entity is represented by its own distinct table. Every record (row) in the table will hold details about a specific instance of that type of thing.

What are the objects or entities that need to be represented in your database? For example, if the Twitter app were to use a relational database, a `Tweet' might be considered an entity (a type of thing), and have its own table. Each row within the Tweet table would then represent a specific Tweet.

List the tables required for your application to meet the basic requirements identified in Task 2.

Task 4: Object Properties

Each column or field in the tables will represent a property of that class of object. For example, the properties of a Tweet might include 'user_account', 'timestamp', and 'tweet_text'.

What information about an object do you think would be needed for your application to meet its functional requirements?

List the fields you think should be associated with each entity in your schema.

Task 5: Field Types

When designing a relational database schema, we should also consider the type of data that each column or field in the database will contain.

Here are some of its commonly used field types:

- INT (whole numbers)
- DECIMAL (decimal numbers)
- VARCHAR(L) (variable length string, where L is the maximum length)
- CHAR(L) (fixed-length string, where L is the length)
- TEXT (for long strings of text)
- BLOB (binary object, for binary encoded data such as images)
- TIMESTAMP (for date and time values stored in UTC)

You can refer to the MySQL Documentation for other field types.

Beside each field on your list, specify which field type you would choose to represent its data.

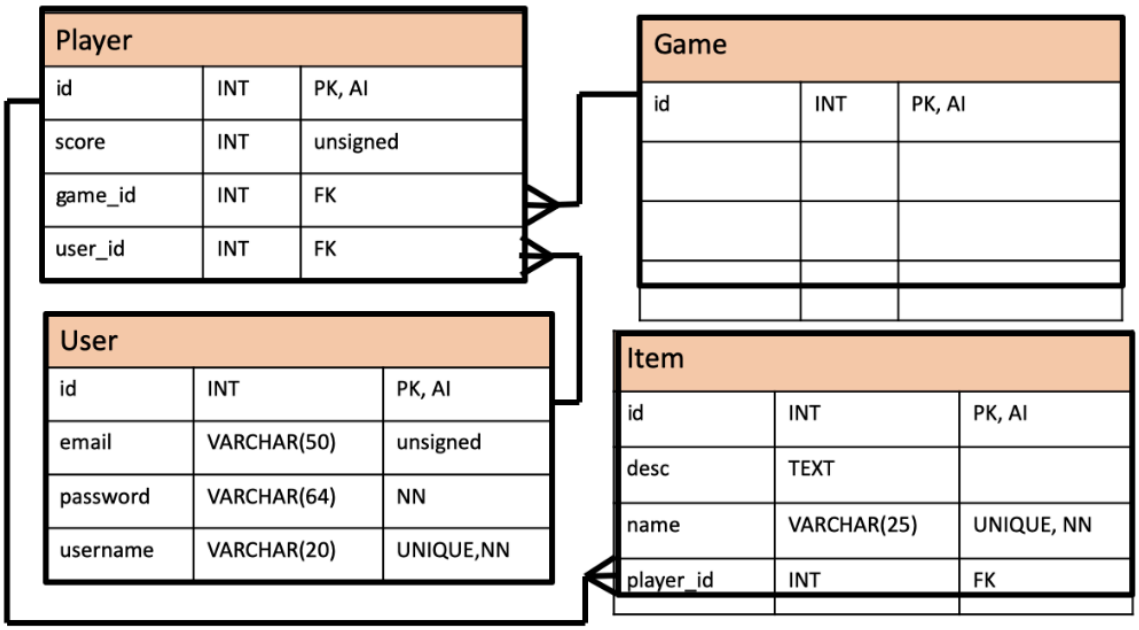
Task 6: Depict your schema in an ER diagram

A standard format for depicting relational database schemas is with an Entity-Relationship (ER) diagram. Some examples (Example ER Diagrams) have been shared with you on the VLE (it's a good idea to have a look!). Produce an ER diagram for your application's database design. **Include the table names, field names, field types, primary and foreign keys (PKs and FKs), and connecting lines (one-to-one, one-to-many, or many-to-many)** to indicate which tables are related.

When designing your database, remember different types of SQL queries you practiced in the previous lab; basic and advanced queries including aggregate functions joins and nested SELECT. Thinking of potential future queries or functionalities could help you to identify entities, properties for each entity, and PKs, FKs, and association types.

Answer formative quiz lab5-2, the quiz will provide you with some hints to improve your design.

There are different formats to draw an ER diagram, as you saw in the Example ER Diagrams. Please use the example below as a guideline of how your ER diagram must look like:



Task 7: Upload your design documents

Upload your database design including the list of basic functions (tasks 1 and 2) and Entity-Relationship Diagram (tasks 3-6) in form of pictures of any hand or computer-drawn documents or notes on the VLE.

Submit your documents on the **lab 5 assessment page**.

Your Entity-relationship diagram must include table names, field names, field types, primary and foreign keys (PKs and FKs), and connecting lines (one-to-one, one-to-many, or many-to-many), refer to the example above as an example.

Task 8: Questions?

Take a few minutes to reflect on what you have done today. What did you learn, and what did you find confusing?

Extension Task: Dig more

Is there any many-to-many association in your ER diagram?

Draw a second Entity-Relationship Diagram including a **junction table** to resolve the issue of many-to-many association.

Leave a comment if you have done the extension task.

Marking of this assignment:

- List of basic functions 20
- Entity-relationship diagram 50
- Extension task 30 marks

END