COMP 636: Python and Database Assessment

Due: **5pm Monday 19th December 2022**

Worth **30%** of COMP636 grade

Submit via Akoraka | Learn

Instructions

This code is library management system for Waikirikiri Library. It is used to manage books, borrowers and loans. You are provided with an outline of the program to complete.

50 marks available in total.

Add the following features to the system:

1. **Menu enhancements (3 marks):** Modify the main menu so that:
    a. All letter inputs are treated as upper-case, e.g., 'q' treated as 'Q' in the main menu.
    b. The menu can be repeated (without an error message) by pressing 'R' (or 'r').
2. **Book list (5 marks):** List all books, sorted by Year of Publication (descending), Category (A->Z) and then Title (A->Z). Each book in your list must also display the number of copies of that book. A basic, but incomplete, function has been provided.
3. **Borrower list (2 marks):** List all borrowers sorted by family name, then first name.
4. **Edit borrower details (12 marks):** List borrowers alphabetically by family name, then first name, also showing BorrowerID. Allow a user to select a borrower (by ID) and then update any values for that borrower (except ID).
5. **Add Loan (6 marks):** Add a new book loan to the system, setting the borrowed date to today. User should input the ID of the borrower and the ID of the book copy.
6. **Overdue books report (8 marks):** Overdue books are those that have been on loan for longer than 35 days (5 weeks). Produce a (on-screen) report that shows overdue book information: book title, borrower name, number of days on loan. No specific ordering is required.
7. **Most loaned Books report (8 marks):** Produce a (on-screen) report that shows the books that have been loaned the most. List books in descending (highest to lowest) order of number of times borrowed.
8. **Overall quality (6 marks):** Tidy output presentation, code structure and commenting across all questions.

File Download and Submission Instructions:

Download the following files from Akoraka | Learn on the COMP636 Assessment page:

- `library_py_db_[your_name].py` – initial code to begin from.
- `connect_library.py` – to be updated with your MySQL connection details.
- `library_create_schema.sql` – run this to create and populate your database.

Submit (upload) only your main Python (.py) file: `library_py_db_[your_name].py`

- Include your name in the filename and your name and student ID in a comment at the start of the file
- Submit your file via the submission link on the COMP636 Assessment page

Mark Allocation:

50 marks available:

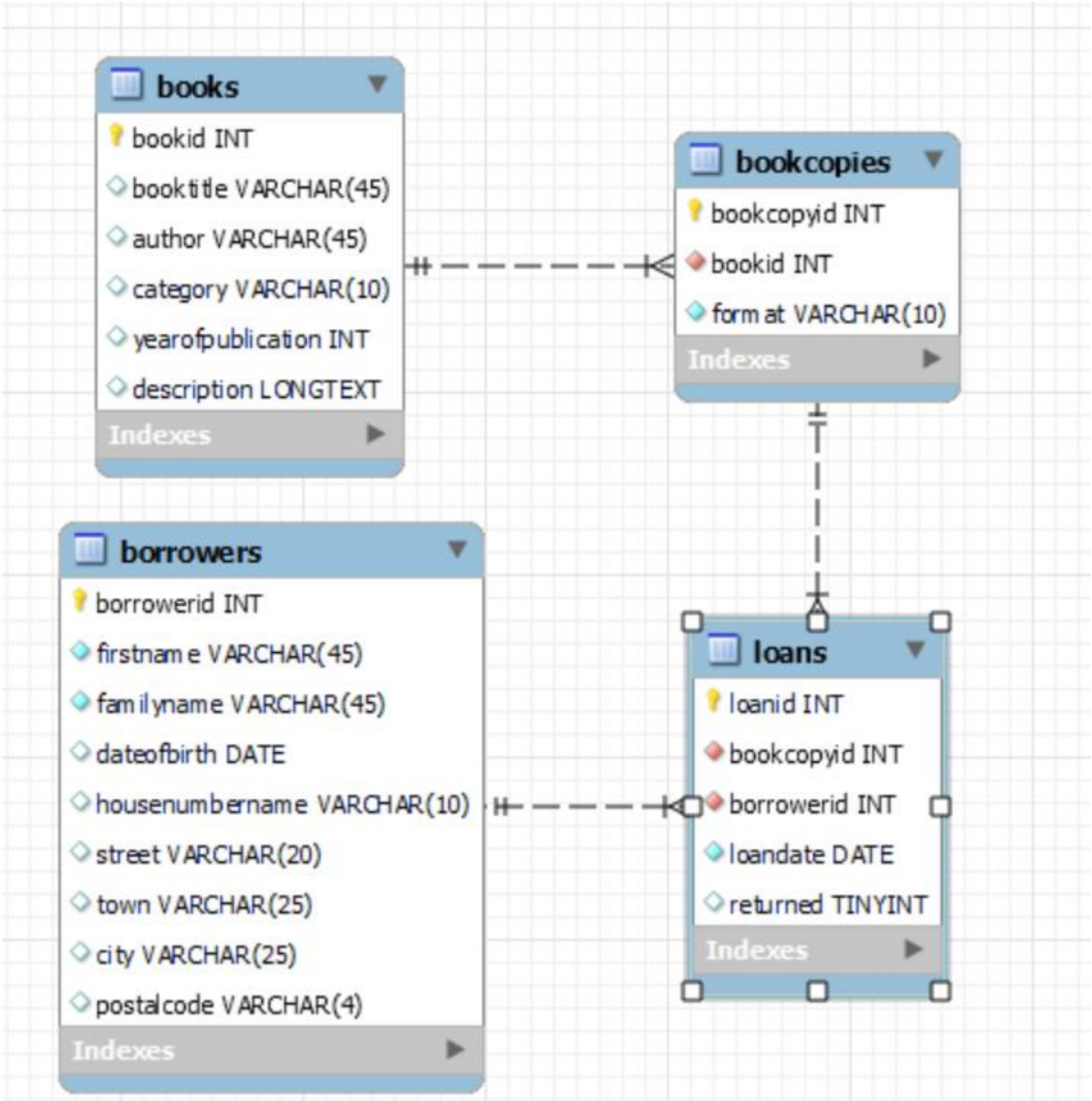
Item	**Marks available**
Menu enhancements	3
Book list	5
Borrower list	2
Update Borrower	12
Add Loan	6
Overdue Books Report	8
Most Loaned Books Report	8
Overall quality	6
TOTAL	**50**

Additional notes:

- **The quality of the user experience will be taken into account for each assessment item** as well as in the Overall Quality mark in the table above. Full marks for any item will require validation of data types (if required) and details in the interface that demonstrate some consideration of what would work well for the user (within the limitations of the terminal window output in VS Code).
- A diagram of the database tables, field names and their relationships, is provided at the end of this document.
- The provided Python file `library_py_db_[your_name].py` contains a menu structure and partially completed functions, these must not be deleted or renamed, but you may add arguments/parameters to these functions. You may also add additional functions of your own if you wish. Rename the file to include your name.
- You will need to update `connect_library.py` with your own connection details to your database. These details are visible on the first screen of MySQL Workbench in the grey box.
- To populate your database, open `library_create_schema.sql`. Run all of the queries in the file in MySQL Workbench, connected to your local MySQL Server database (press ⚡ with nothing selected). This will create the tables and then populate them with data. You will need to refresh your Schemas pane in the sidebar to see the new schema and double click `library` to activate it. These queries can be re-run at any time to reset your data. The structure of the database is shown on the final page of this document.
- The `columnOutput()` function is available to produce nicely formatted output in columns. Instructions for use are in the comments for the function. You are encouraged to use it, but it is not essential for all menu items.
- Add comments to your code. The existing comments give some hints about the behaviour expectations of each function.

Database structure

Note: The lines shown only indicate the relationships between the tables. They do not indicate which fields are included in the table relationships. See the table below the diagram for the field details.



books

- bookid INT
- booktitle VARCHAR(45)
- author VARCHAR(45)
- category VARCHAR(10)
- yearofpublication INT
- description LONGTEXT
- Indexes

bookcopies

- bookcopyid INT
- bookid INT
- format VARCHAR(10)
- Indexes

borrowers

- borrowerid INT
- firstname VARCHAR(45)
- familyname VARCHAR(45)
- dateofbirth DATE
- housenumbername VARCHAR(10)
- street VARCHAR(20)
- town VARCHAR(25)
- city VARCHAR(25)
- postalcode VARCHAR(4)
- Indexes

loans

- loanid INT
- bookcopyid INT
- borrowerid INT
- loandate DATE
- returned TINYINT
- Indexes

Foreign keys

Foreign keys enforce ‘referential integrity’ – which means that a value in the child table field must match an entry in the parent table field, e.g., a **bookid** cannot be entered into the **bookid** field of the **bookcopies** table if it is not in the **bookid** field of the book table.

Parent table and field(s)		Child table and field(s)
books.bookid	—<	bookcopies.bookid
bookcopies.bookcopyid	—<	loans.bookcopyid
borrowers.borrowerid	—<	loans.borrowerid

Loans that have been returned have the value 1 in the returned field.