Cognizant GenC Case Study

Library Management System

Candidate ID: 29854750
Name: B Jamin Enock

Section 2: SQL Queries

Basic Queries:

```
CREATE DATABASE library;
```

The command "CREATE DATABASE library;" sets up a digital storage space to organize and manage information for a library, facilitating efficient data handling and retrieval.

```
USE library;
```

The command "USE library;" selects the "library" database for subsequent operations, allowing access and modification of its stored data.

```
CREATE TABLE books(book_id INT,title VARCHAR(100),author
VARCHAR(50),genre VARCHAR(50),price DECIMAL(10,2),stock INT,PRIMARY
KEY(book_id));
```

The command "CREATE TABLE books" defines a structured format within the "library" database to store book details such as ID, title, author, genre, price, and stock, with "book_id" as the primary key for uniquely identifying each book.

```
CREATE TABLE borrowers(borrower_id INT,name VARCHAR(100),contact
VARCHAR(50),PRIMARY KEY(borrower_id));
```

The command `CREATE TABLE borrowers` sets up a table to store borrower information with columns for borrower ID, name, and contact details. The `borrower_id` is designated as the primary key to uniquely identify each borrower.

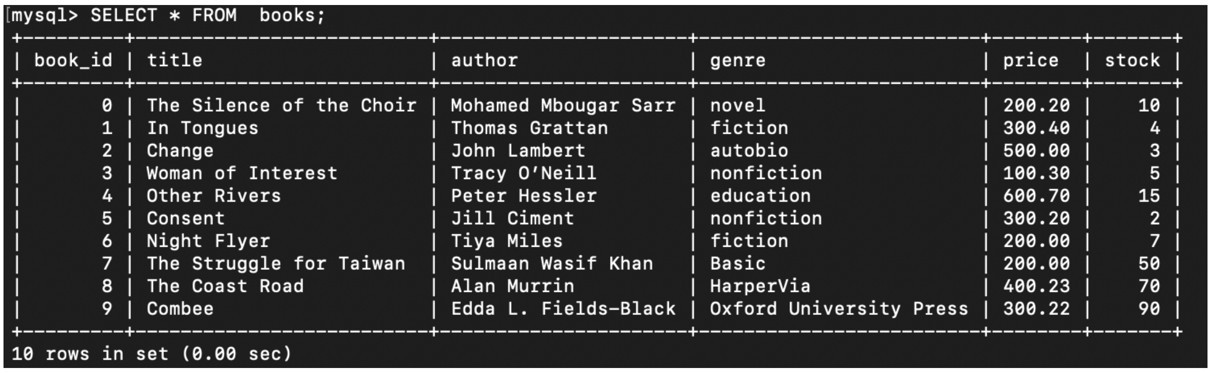
```
CREATE TABLE borrowing(borrowing_id INT,book_id INT,borrower_id
INT,quantity_borrowed INT,borrowing_date DATE,FOREIGN KEY(book_id)
REFERENCES books(book_id),FOREIGN KEY(borrower_id) REFERENCES
borrowers(borrower_id));
```

The command "CREATE TABLE borrowing" sets up a table in the database to manage borrowing records, including details such as borrowing ID, book ID, borrower ID, quantity borrowed, and borrowing date. It establishes foreign key constraints linking "book_id" to the "books" table and "borrower_id" to a hypothetical "borrowers" table for relational integrity.

```
INSERT INTO books(book_id,title,author,genre,price,stock) VALUES
(0,'The Silence of the Choir','Mohamed Mbougar
Sarr','novel',200.20,10),
(1,'In Tongues','Thomas Grattan','fiction',300.40,4),
(2,'Change','John Lambert','autobio',500,3),
(3,'Woman of Interest','Tracy O’Neill','nonfiction',100.30,5),
(4,'Other Rivers','Peter Hessler','education',600.70,15);
```

The command `INSERT INTO books` adds multiple rows of book data into the "books" table, including book ID, title, author, genre, price, and stock for each entry.

```
SELECT * FROM  books;
```



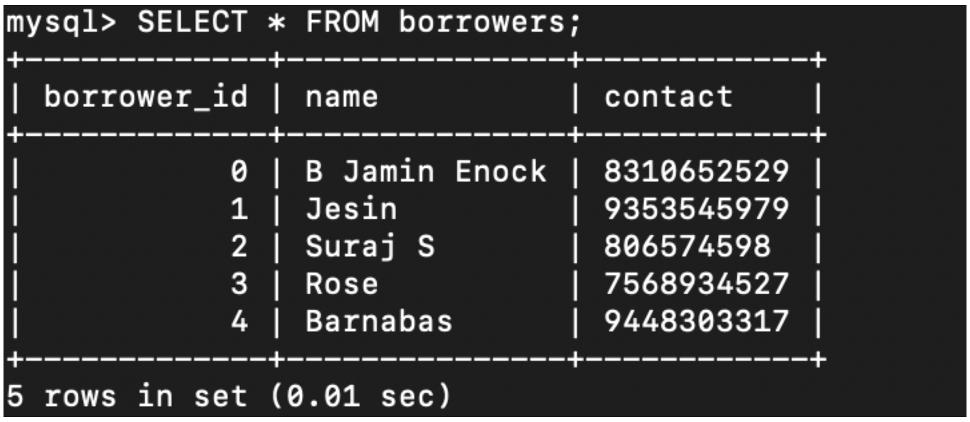
```
mysql> SELECT * FROM  books;
+---------+-------------------------+---------------------+-------------------------+--------+-------+
| book_id | title                   | author              | genre                   | price  | stock |
+---------+-------------------------+---------------------+-------------------------+--------+-------+
|       0 | The Silence of the Choir | Mohamed Mbougar Sarr | novel                  | 200.20 |    10 |
|       1 | In Tongues              | Thomas Grattan      | fiction                 | 300.40 |     4 |
|       2 | Change                  | John Lambert        | autobio                 | 500.00 |     3 |
|       3 | Woman of Interest       | Tracy O’Neill       | nonfiction              | 100.30 |     5 |
|       4 | Other Rivers            | Peter Hessler       | education               | 600.70 |    15 |
|       5 | Consent                 | Jill Ciment         | nonfiction              | 300.20 |     2 |
|       6 | Night Flyer             | Tiya Miles          | fiction                 | 200.00 |     7 |
|       7 | The Struggle for Taiwan | Sulmaan Wasif Khan  | Basic                   | 200.00 |    50 |
|       8 | The Coast Road          | Alan Murrin         | HarperVia               | 400.23 |    70 |
|       9 | Combee                  | Edda L. Fields-Black | Oxford University Press | 300.22 |    90 |
+---------+-------------------------+---------------------+-------------------------+--------+-------+
10 rows in set (0.00 sec)
```

The query `SELECT * FROM books;` retrieves all rows and columns from the "books" table, displaying details such as book ID, title, author, genre, price, and stock for each book stored in the database.

```
INSERT INTO borrowers(borrower_id,name,contact) VALUES
(0,'B Jamin Enock','8310652529'),
(1,'Jesin','9353545979'),
(2,'Suraj S','806574598'),
(3,'Rose','7568934527'),
(4,'Barnabas','9448303317');
```

The SQL command `INSERT INTO borrowers` adds new records into the "borrowers" table, specifying each borrower's ID, name, and contact information. This operation is essential for maintaining a database of library users, allowing efficient management and retrieval of borrower details for transactions and communication purposes.

```
SELECT * FROM borrowers;
```



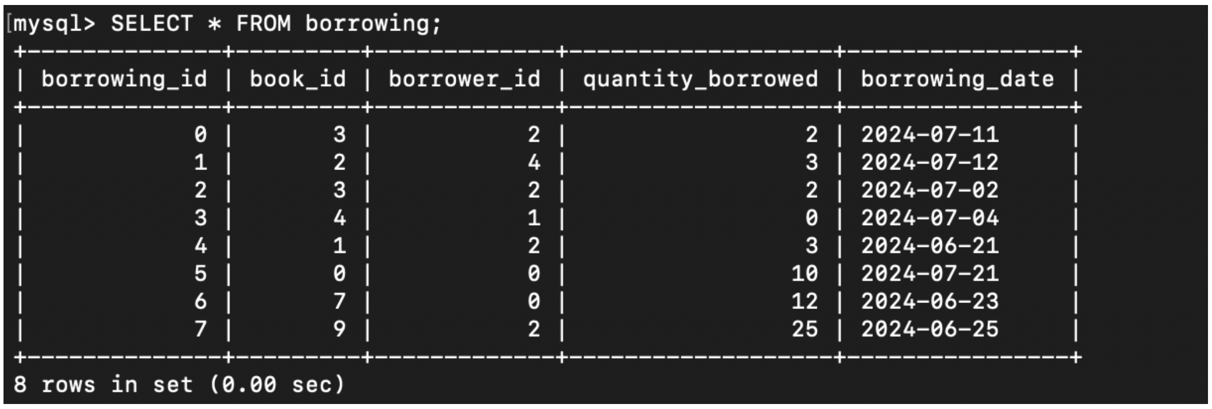
```
mysql> SELECT * FROM borrowers;
+-------------+---------------+------------+
| borrower_id | name          | contact    |
+-------------+---------------+------------+
|           0 | B Jamin Enock | 8310652529 |
|           1 | Jesin         | 9353545979 |
|           2 | Suraj S       | 806574598  |
|           3 | Rose          | 7568934527 |
|           4 | Barnabas      | 9448303317 |
+-------------+---------------+------------+
5 rows in set (0.01 sec)
```

The query `SELECT * FROM borrowers;` retrieves all rows and columns from the "borrowers" table, displaying the borrower ID, name, and contact information for each borrower stored in the database.

```
INSERT INTO
borrowing(borrowing_id,book_id,borrower_id,quantity_borrowed,borrowi
ng_date) VALUES
(0,3,2,2,'2024-07-11'),
(1,2,4,3,'2024-07-12'),
(2,3,2,2,'2024-07-02'),
(3,4,1,0,'2024-07-04'),
(4,1,2,3,'2024-06-21');
```

The SQL command `INSERT INTO borrowing` is used to add records into the "borrowing" table, specifying details such as borrowing ID, book ID, borrower ID, quantity borrowed, and borrowing date for each transaction. This operation allows the database to track and manage borrowing activities, facilitating efficient monitoring of book loans and associated borrower information.

```
SELECT * FROM borrowing;
```



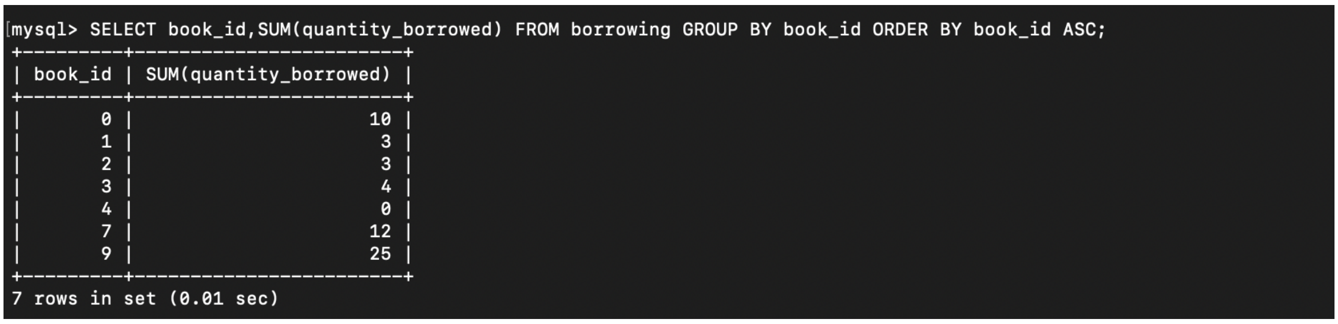
```
mysql> SELECT * FROM borrowing;
+--------------+---------+-------------+-------------------+----------------+
| borrowing_id | book_id | borrower_id | quantity_borrowed | borrowing_date |
+--------------+---------+-------------+-------------------+----------------+
|            0 |       3 |           2 |                 2 | 2024-07-11     |
|            1 |       2 |           4 |                 3 | 2024-07-12     |
|            2 |       3 |           2 |                 2 | 2024-07-02     |
|            3 |       4 |           1 |                 0 | 2024-07-04     |
|            4 |       1 |           2 |                 3 | 2024-06-21     |
|            5 |       0 |           0 |                10 | 2024-07-21     |
|            6 |       7 |           0 |                12 | 2024-06-23     |
|            7 |       9 |           2 |                25 | 2024-06-25     |
+--------------+---------+-------------+-------------------+----------------+
8 rows in set (0.00 sec)
```

The query `SELECT * FROM borrowing;` retrieves all rows and columns from the "borrowing" table, displaying details such as borrowing ID, book ID, borrower ID, quantity borrowed, and borrowing date for each borrowing transaction recorded in the database.

Questions:

1. Write a query to find the total quantity of each book borrowed.

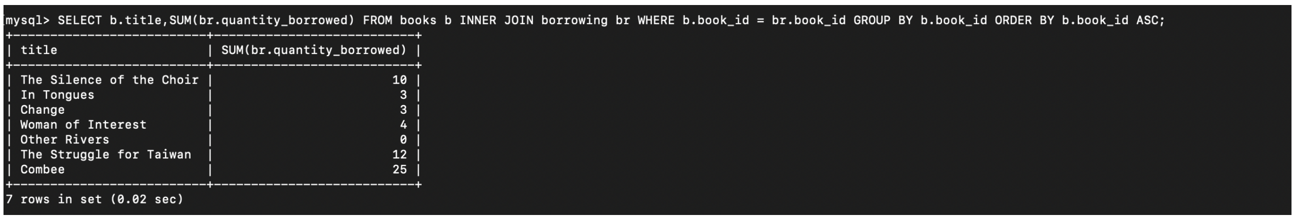
```
SELECT book_id,SUM(quantity_borrowed) FROM borrowing GROUP BY
book_id ORDER BY book_id ASC;
```



```
mysql> SELECT book_id,SUM(quantity_borrowed) FROM borrowing GROUP BY book_id ORDER BY book_id ASC;
+---------+------------------------+
| book_id | SUM(quantity_borrowed) |
+---------+------------------------+
|       0 |                     10 |
|       1 |                      3 |
|       2 |                      3 |
|       3 |                      4 |
|       4 |                      0 |
|       7 |                     12 |
|       9 |                     25 |
+---------+------------------------+
7 rows in set (0.01 sec)
```

2. Write a query to find the book title and total quantity borrowed for each book.

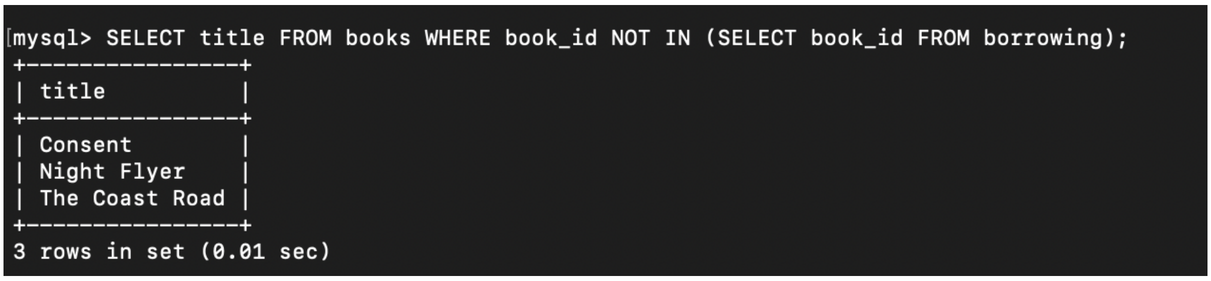
```sql
SELECT b.title,SUM(br.quantity_borrowed) FROM books b INNER JOIN
borrowing br WHERE b.book_id = br.book_id GROUP BY b.book_id ORDER
BY b.book_id ASC;
```



```
mysql> SELECT b.title,SUM(br.quantity_borrowed) FROM books b INNER JOIN borrowing br WHERE b.book_id = br.book_id GROUP BY b.book_id ORDER BY b.book_id ASC;
+--------------------------+--------------------------+
| title                    | SUM(br.quantity_borrowed) |
+--------------------------+--------------------------+
| The Silence of the Choir |                       10 |
| In Tongues               |                        3 |
| Change                   |                        3 |
| Woman of Interest        |                        4 |
| Other Rivers             |                        0 |
| The Struggle for Taiwan  |                       12 |
| Combee                   |                       25 |
+--------------------------+--------------------------+
7 rows in set (0.02 sec)
```

3. Write a query to find the titles of books that have never been borrowed.

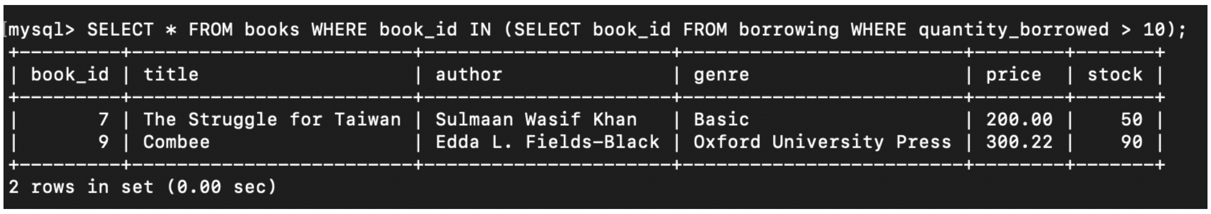
```sql
SELECT title FROM books WHERE book_id NOT IN (SELECT book_id FROM
borrowing);
```



```
mysql> SELECT title FROM books WHERE book_id NOT IN (SELECT book_id FROM borrowing);
+----------------+
| title          |
+----------------+
| Consent        |
| Night Flyer    |
| The Coast Road |
+----------------+
3 rows in set (0.01 sec)
```

4. Write a query to find the books that have been borrowed more than 10 times.

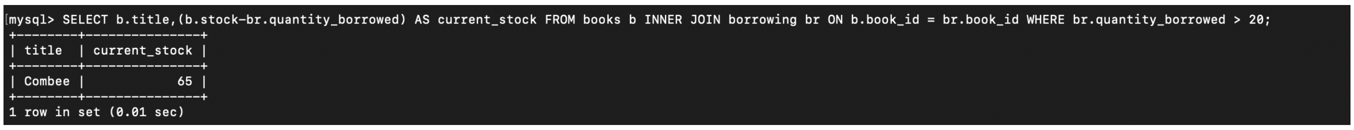
```sql
SELECT * FROM books WHERE book_id IN (SELECT book_id FROM borrowing
WHERE quantity_borrowed > 10);
```



```
mysql> SELECT * FROM books WHERE book_id IN (SELECT book_id FROM borrowing WHERE quantity_borrowed > 10);
+---------+-------------------------+---------------------+-------------------------+--------+-------+
| book_id | title                   | author              | genre                   | price  | stock |
+---------+-------------------------+---------------------+-------------------------+--------+-------+
|       7 | The Struggle for Taiwan | Sulmaan Wasif Khan  | Basic                   | 200.00 |    50 |
|       9 | Combee                  | Edda L. Fields-Black | Oxford University Press | 300.22 |    90 |
+---------+-------------------------+---------------------+-------------------------+--------+-------+
2 rows in set (0.00 sec)
```

5. Write a query to find the book titles and their current stock levels for books that have been borrowed more than 20 times.

```sql
SELECT b.title,(b.stock-br.quantity_borrowed) AS current_stock FROM
books b INNER JOIN borrowing br ON b.book_id = br.book_id WHERE
br.quantity_borrowed > 20;
```



```
mysql> SELECT b.title,(b.stock-br.quantity_borrowed) AS current_stock FROM books b INNER JOIN borrowing br ON b.book_id = br.book_id WHERE br.quantity_borrowed > 20;
+--------+---------------+
| title  | current_stock |
+--------+---------------+
| Combee |            65 |
+--------+---------------+
1 row in set (0.01 sec)
```