SQL Project: Library Database

This project demonstrates SQL functions including:

- Table Creation
- Data Insertion
- Joins
- Sub-queries
- CTE (Common Table Expressions)

---------- Table Definitions----------

- Create the Authors table

```
CREATE TABLE Authors (
    AuthorID INT PRIMARY KEY,
    Name TEXT
);
```

- Create the Books table

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title TEXT,
    AuthorID INT,
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);
```

- Create the Checkouts table

```
CREATE TABLE Checkouts (
    CheckoutID INT PRIMARY KEY,
    BookID INT,
    CheckoutDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (BookID) REFERENCES Books(BookID)
```

);

----------Sample Data Insertion ----------

- Insert sample data into Authors table

```
INSERT INTO Authors (AuthorID, Name) VALUES
(1, 'J.K. Rowling'),
(2, 'George Orwell'),
(3, 'J.R.R. Tolkien');
```

- Insert sample data into Books table

```
INSERT INTO Books (BookID, Title, AuthorID) VALUES
(1, 'Harry Potter', 1),
(2, '1984', 2),
(3, 'The Hobbit', 3);
```

- Insert sample data into Checkouts table

```
INSERT INTO Checkouts (CheckoutID, BookID, CheckoutDate, ReturnDate) VALUES
(1, 1, '2023-01-01', '2023-01-10'),
(2, 2, '2023-01-05', NULL),
(3, 3, '2023-01-10', '2023-01-20'),
(4, 1, '2023-02-01', NULL);
```

---SQL Queries---

-CTE: Calculate the number of times each book has been checked out

```
WITH BookCheckoutCounts AS (
    SELECT BookID, COUNT(*) AS CheckoutCount
    FROM Checkouts
    GROUP BY BookID
)
```

```
-- Query to join all tables and include checkout counts
SELECT
    b.BookID,
    b.Title,
    a.Name AS Author,
    COALESCE(c.CheckoutCount, 0) AS CheckoutCount
FROM
    Books b
JOIN Authors a ON b.AuthorID = a.AuthorID
LEFT JOIN BookCheckoutCounts c ON b.BookID = c.BookID;

- CTE: Find books that are currently checked out
WITH CurrentlyCheckedOut AS (
    SELECT BookID
    FROM Checkouts
    WHERE ReturnDate IS NULL
)
- Query to list all books and indicate whether they are currently checked out
SELECT
    b.BookID,
    b.Title,
    CASE
        WHEN c.BookID IS NOT NULL THEN 'Checked Out'
        ELSE 'Available'
    END AS Status
FROM
    Books b
LEFT JOIN CurrentlyCheckedOut c ON b.BookID = c.BookID;
```