Learning Outcome 3 exam

**-- Create the database**

CREATE DATABASE IF NOT EXISTS library\_db;

USE library\_db;

**-- Create the books table**

CREATE TABLE IF NOT EXISTS books (

book\_id INT PRIMARY KEY,

title VARCHAR(100),

author\_id INT,

genre\_id INT,

copies\_available INT,

FOREIGN KEY (author\_id) REFERENCES authors(author\_id),

FOREIGN KEY (genre\_id) REFERENCES genres(genre\_id)

);

**-- Create the authors table**

CREATE TABLE IF NOT EXISTS authors (

author\_id INT PRIMARY KEY,

author\_name VARCHAR(50)

);

**-- Create the genres table**

CREATE TABLE IF NOT EXISTS genres (

genre\_id INT PRIMARY KEY,

genre\_name VARCHAR(50)

);

**-- Create the borrowers table**

CREATE TABLE IF NOT EXISTS borrowers (

borrower\_id INT PRIMARY KEY,

borrower\_name VARCHAR(50)

);

**-- Create the loans table**

CREATE TABLE IF NOT EXISTS loans (

loan\_id INT PRIMARY KEY,

book\_id INT,

borrower\_id INT,

due\_date DATE,

FOREIGN KEY (book\_id) REFERENCES books(book\_id),

FOREIGN KEY (borrower\_id) REFERENCES borrowers(borrower\_id)

);

**-- Insert sample data into the tables**

INSERT INTO authors VALUES (1, 'J.K. Rowling');

INSERT INTO authors VALUES (2, 'George Orwell');

INSERT INTO genres VALUES (1, 'Fantasy');

INSERT INTO genres VALUES (2, 'Dystopian');

INSERT INTO books VALUES (1, 'Harry Potter and the Sorcerer\'s Stone', 1, 1, 5);

INSERT INTO books VALUES (2, '1984', 2, 2, 3);

INSERT INTO borrowers VALUES (1, 'Alice Johnson');

INSERT INTO borrowers VALUES (2, 'Bob Smith');

INSERT INTO loans VALUES (1, 1, 1, '2024-03-15');

INSERT INTO loans VALUES (2, 2, 2, '2024-04-01');

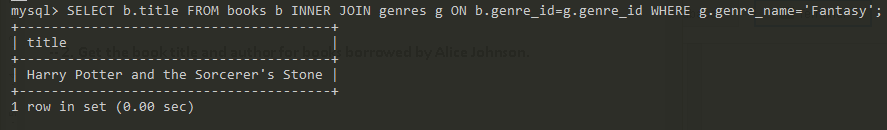
Diagrama

Descripción generada automáticamente

DESCRIBE authors;DESCRIBE books;DESCRIBE loans;DESCRIBE borrowers;DESCRIBE genres;

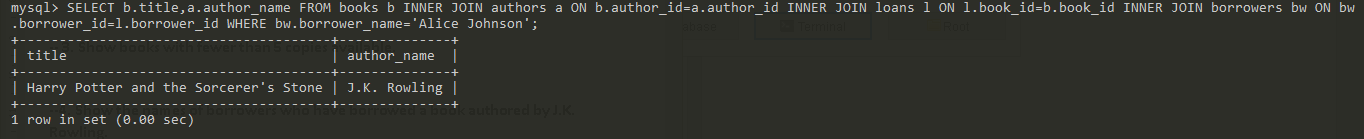
**- 1. Show all books in the Fantasy genre.**

SELECT b.title FROM books b INNER JOIN genres g ON b.genre\_id=g.genre\_id WHERE g.genre\_name=’Fantasy’;



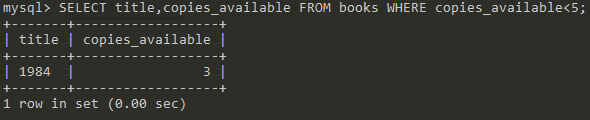
**-- 2. Get the book title and author for books borrowed by Alice Johnson.**

SELECT b.title,a.author\_name FROM books b INNER JOIN authors a ON b.author\_id=a.author\_id INNER JOIN loans l ON l.book\_id=b.book\_id INNER JOIN borrowers bw ON bw.borrower\_id=l.borrower\_id WHERE bw.borrower\_name=’Alice Johnson’;



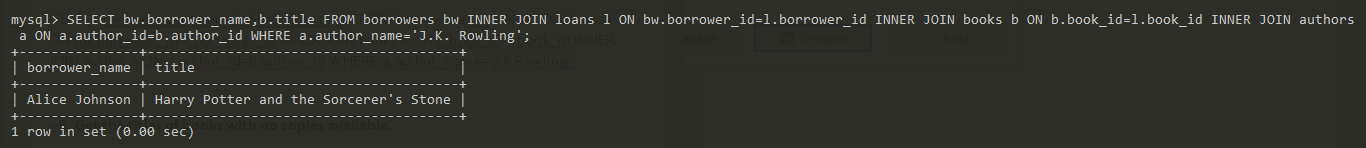
**-- 3. Show books with fewer than 5 copies available.**

SELECT title,copies\_available FROM books WHERE copies\_available<5;



**--4. Show the names of borrowers who have borrowed a book authored by J.K. Rowling.**

SELECT bw.borrower\_name,b.title,l.due\_date FROM borrowers bw INNER JOIN loans l ON bw.borrower\_id=l.borrower\_id INNER JOIN books b ON b.book\_id=l.book\_id INNER JOIN author a ON a.author\_id=b.author\_id WHERE a.author\_name=’J.K. Rowling’;



**--5. Get the titles of books with no copies available.**

SELECT title FROM books WHERE copies\_available=0;



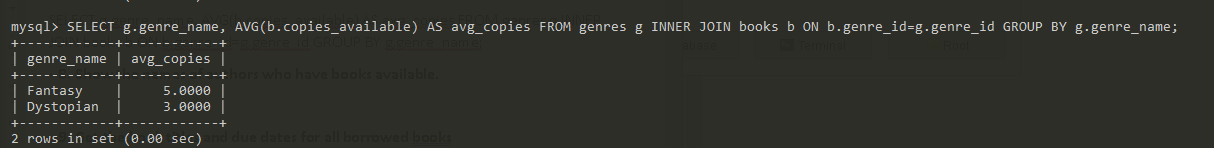
**--6. Show the borrowers who have not borrowed any books.**

SELECT bw.borrower\_name FROM borrowers bw INNER JOIN loans l ON bw.borrower\_id=l.borrower\_id WHERE l.book\_id=NULL;



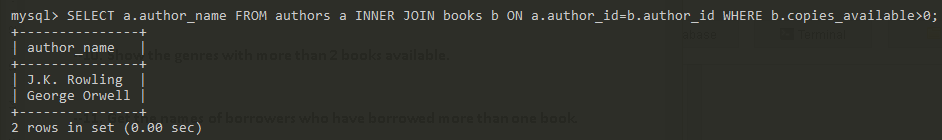
**--7. Get the average number of copies available for books in each genre.**

SELECT g.genre\_name, AVG(b.copies\_available) AS avg\_copies FROM genres g INNER JOIN books b ON b.genre\_id=g.genre\_id GROUP BY g.genre\_name;



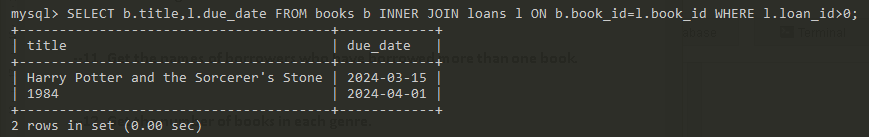
**--8. Show the names of authors who have books available.**

SELECT a.author\_name FROM authors a INNER JOIN books b ON a.author\_id=b.author\_id WHERE b.copies\_available>0;



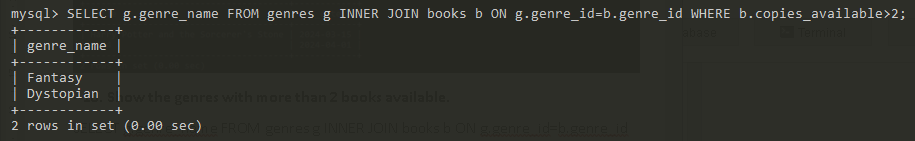
**--9. Get the book titles and due dates for all borrowed books**

SELECT b.title,l.due\_date FROM books b INNER JOIN loans l ON b.book\_id=l.book\_id WHERE l.loan\_id>0;



**--10. Show the genres with more than 2 books available.**

SELECT g.genre\_name FROM genres g INNER JOIN books b ON g.genre\_id=b.genre\_id WHERE b.copies\_available>2;



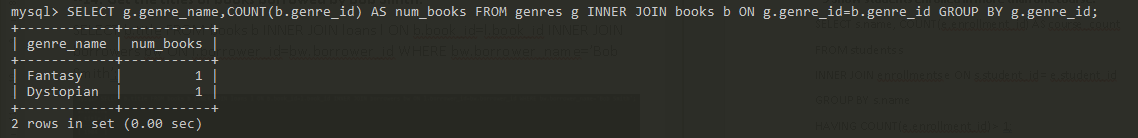
**--11. Get the names of borrowers who have borrowed more than one book.**

SELECT bw.borrower\_name FROM borrowers bw INNER JOIN loans l ON bw.borrower\_id=l.borrower\_id GROUP BY l.borrower\_id HAVING COUNT(l.borrower)>1;



**--12. Get the number of books in each genre.**

SELECT g.genre\_name,COUNT(b.genre\_id) AS num\_books FROM genres g INNER JOIN books b ON g.genre\_id=b.genre\_id GROUP BY g.genre\_id;



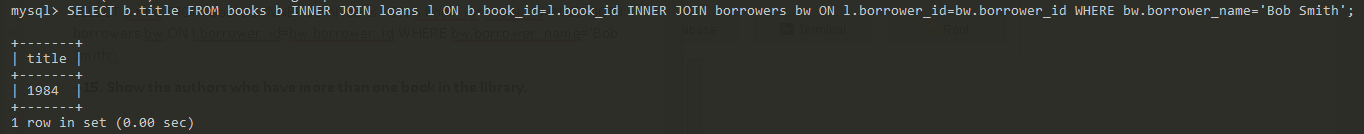
**--13. Show the names of borrowers who have overdue books.**

SELECT bw.borrower\_name FROM borrowers bw INNER JOIN loans l ON bw.borrower\_id=l.borrower\_id WHERE l.due\_date<curtime();



**--14. Get the titles of books borrowed by Bob Smith.**

SELECT b.title FROM books b INNER JOIN loans l ON b.book\_id=l.book\_id INNER JOIN borrowers bw ON l.borrower\_id=bw.borrower\_id WHERE bw.borrower\_name=’Bob Smith’;



**--15. Show the authors who have more than one book in the library.**

SELECT a.author\_name FROM authors a INNER JOIN books b ON a.author\_id=b.author\_id GROUP BY a.author\_id HAVING COUNT(b.book\_id)>1;



**--16. Get the total number of copies available for all books.**

SELECT SUM(copies\_available) AS total\_num\_books FROM books;

