

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);
```

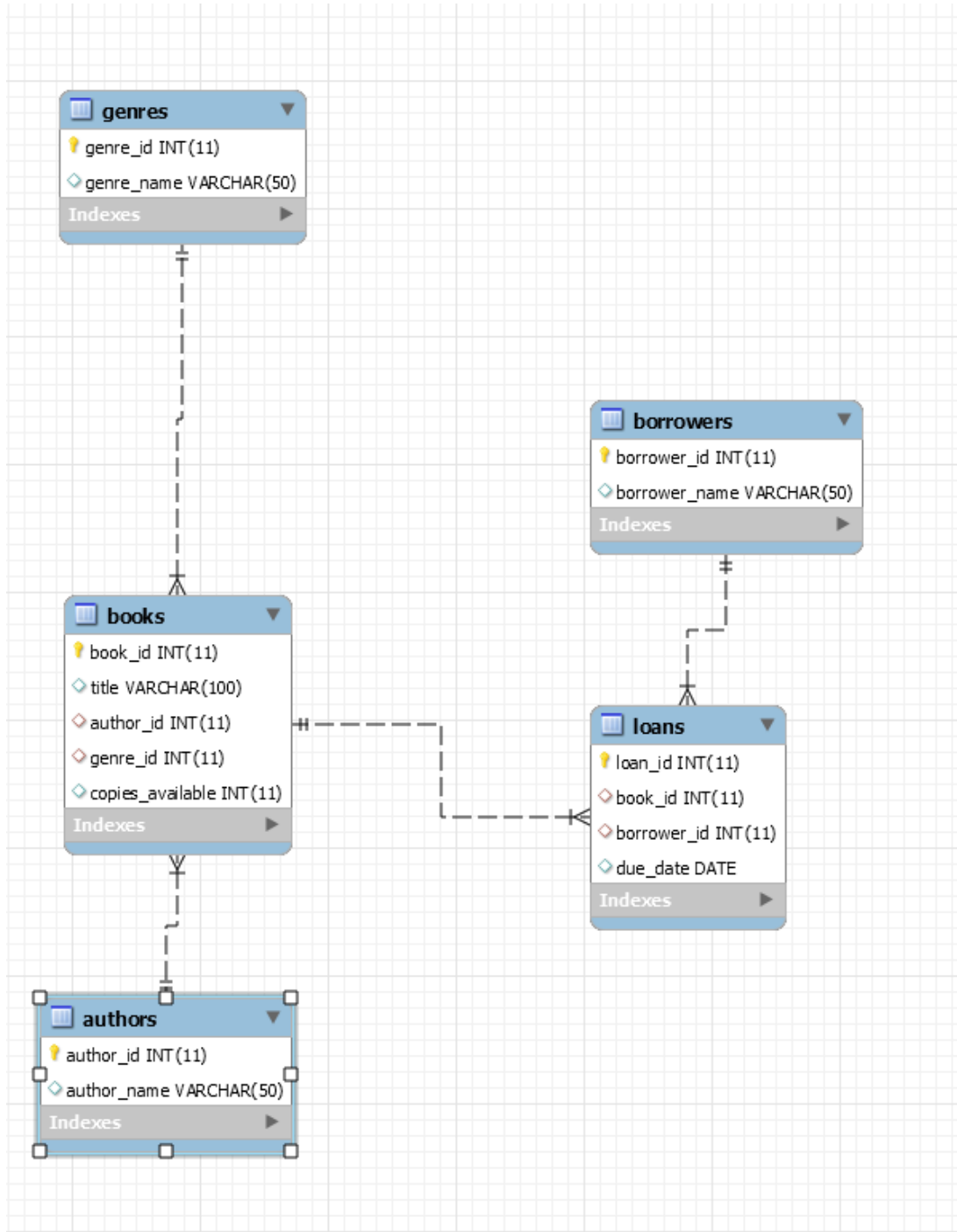
```
INSERT INTO books VALUES (2, '1984', 2, 2, 3);
```

```
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');
```



```
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';
```

```
mysql> SELECT b.title FROM books b INNER JOIN genres g ON b.genre_id=g.genre_id WHERE g.genre_name='Fantasy';
+-----+-----+
| title |
+-----+-----+
| Harry Potter and the Sorcerer's Stone |
+-----+-----+
1 row in set (0.00 sec)
```

-- 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';
```

```
mysql> 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';
+-----+-----+
| title | author_name |
+-----+-----+
| Harry Potter and the Sorcerer's Stone | J.K. Rowling |
+-----+-----+
1 row in set (0.00 sec)
```

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

```
SELECT title,copies_available FROM books WHERE copies_available<5;
```

```
mysql> SELECT title,copies_available FROM books WHERE copies_available<5;
+-----+-----+
| title | copies_available |
+-----+-----+
| 1984 | 3 |
+-----+-----+
1 row in set (0.00 sec)
```

--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';
```

```
mysql> SELECT bw.borrower_name,b.title 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 authors a ON a.author_id=b.author_id WHERE a.author_name='J.K. Rowling';
+-----+-----+
| borrower_name | title |
+-----+-----+
| Alice Johnson | Harry Potter and the Sorcerer's Stone |
+-----+-----+
1 row in set (0.00 sec)
```

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

```
SELECT title FROM books WHERE copies_available=0;
```

```
mysql> SELECT title FROM books WHERE copies_available=0;
Empty set (0.00 sec)
```

--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;
```

```
mysql> SELECT bw.borrower_name FROM borrowers bw INNER JOIN loans l ON bw.borrower_id=l.borrower_id WHERE l.book_id=NULL;  
Empty set (0.00 sec)
```

--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;
```

```
mysql> 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;  
+-----+-----+  
| genre_name | avg_copies |  
+-----+-----+  
| Fantasy    | 5.0000     |  
| Dystopian  | 3.0000     |  
+-----+-----+  
2 rows in set (0.00 sec)
```

--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;
```

```
mysql> SELECT a.author_name FROM authors a INNER JOIN books b ON a.author_id=b.author_id WHERE b.copies_available>0;  
+-----+  
| author_name |  
+-----+  
| J.K. Rowling |  
| George Orwell |  
+-----+  
2 rows in set (0.00 sec)
```

--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;
```

```
mysql> 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;  
+-----+-----+  
| title | due_date |  
+-----+-----+  
| Harry Potter and the Sorcerer's Stone | 2024-03-15 |  
| 1984 | 2024-04-01 |  
+-----+-----+  
2 rows in set (0.00 sec)
```

--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;
```

```
mysql> SELECT g.genre_name FROM genres g INNER JOIN books b ON g.genre_id=b.genre_id WHERE b.copies_available>2;  
+-----+  
| genre_name |  
+-----+  
| Fantasy    |  
| Dystopian  |  
+-----+  
2 rows in set (0.00 sec)
```

--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_id)>1;
```

```
mysql> 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_id)>1;  
Empty set (0.00 sec)
```

--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;
```

```
mysql> 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;  
+-----+-----+  
| genre_name | num_books |  
+-----+-----+  
| Fantasy | 1 |  
| Dystopian | 1 |  
+-----+-----+  
2 rows in set (0.00 sec)
```

--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();
```

```
mysql> SELECT bw.borrower_name FROM borrowers bw INNER JOIN loans l ON bw.borrower_id=l.borrower_id WHERE l.due_date<curtime();  
Empty set (0.00 sec)
```

--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';
```

```
mysql> 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';  
+-----+  
| title |  
+-----+  
| 1984 |  
+-----+  
1 row in set (0.00 sec)
```

--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;
```

```
mysql> 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;  
Empty set (0.00 sec)
```

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

```
SELECT SUM(copies_available) AS total_num_books FROM books;
```

```
mysql> SELECT SUM(copies_available) AS total_num_books FROM books;  
+-----+  
| total_num_books |  
+-----+  
| 8 |  
+-----+  
1 row in set (0.00 sec)
```