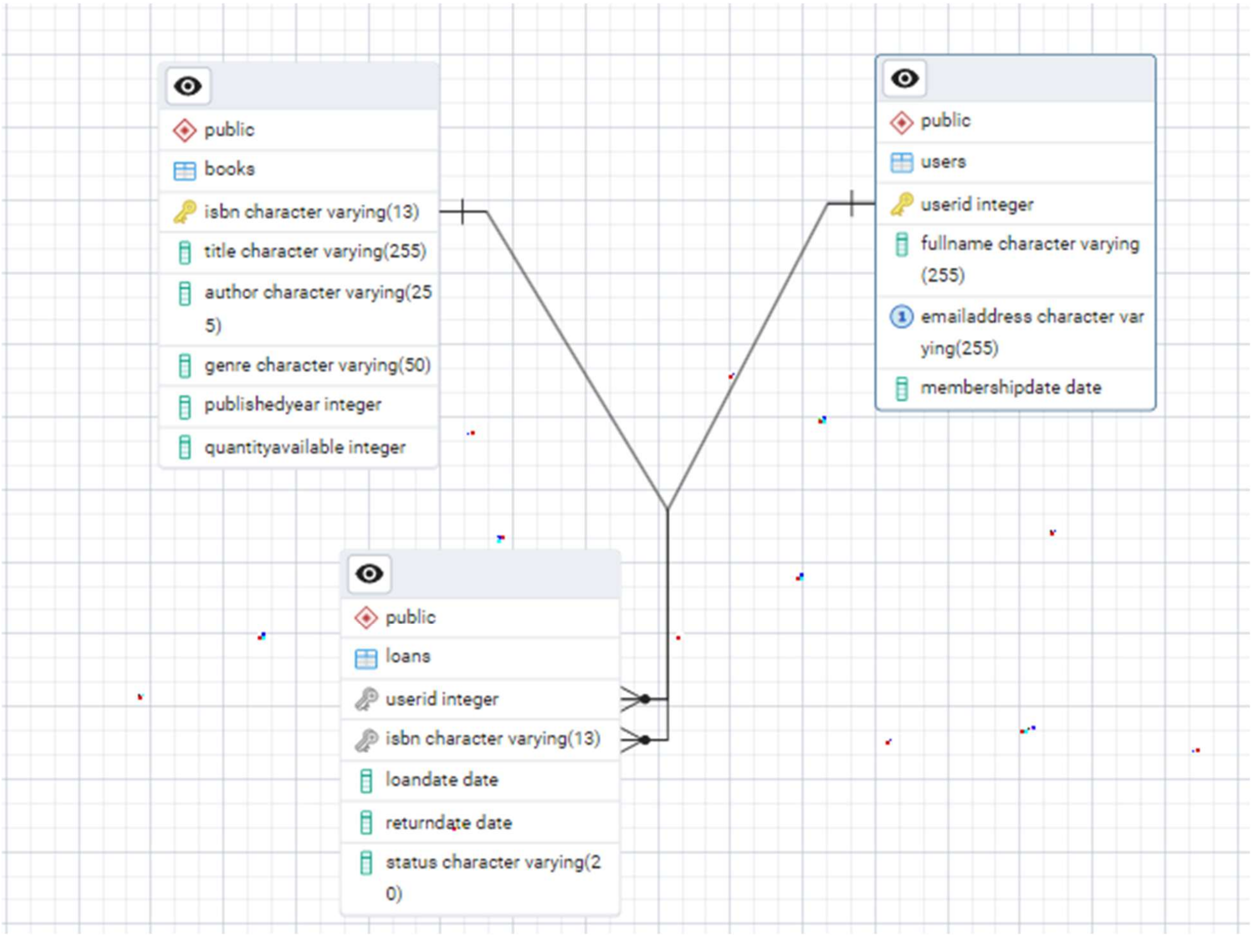
Krystal Jane Siaotong BSSE – 2

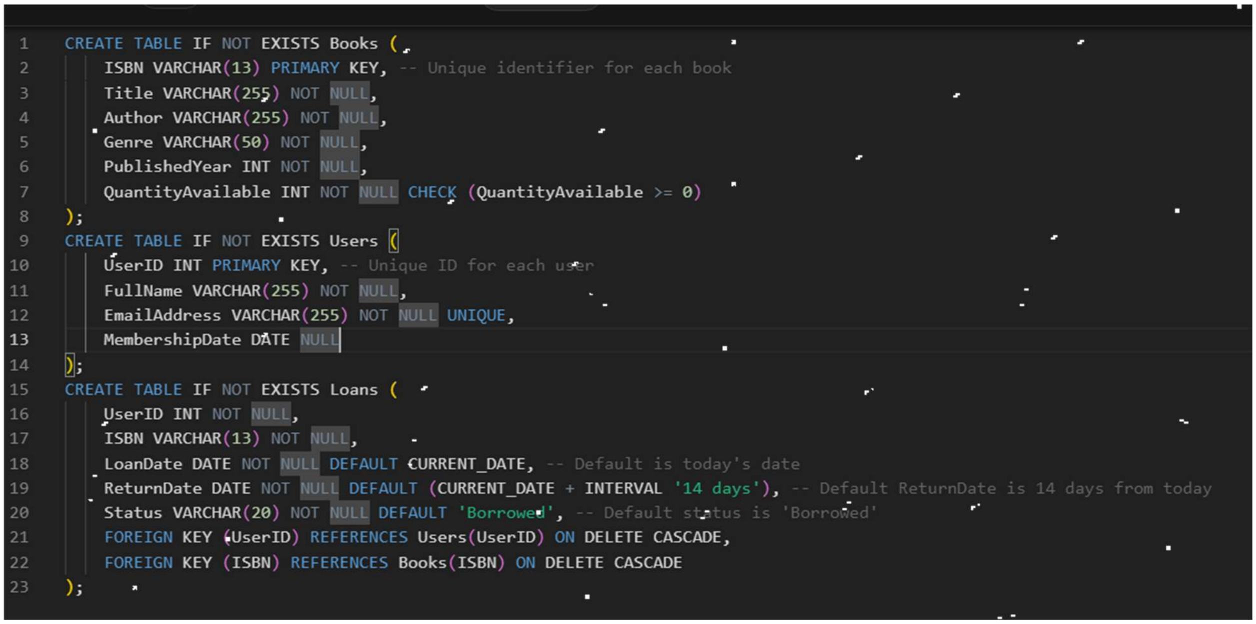
SE 2141

Online Library Management System

Part 1: Conceptual Design


public
books
isbn character varying(13)
title character varying(255)
author character varying(255)
genre character varying(50)
publishedyear integer
quantityavailable integer
public
users
userid integer
fullname character varying (255)
emailaddress character varying(255)
membershipdate date
public
loans
userid integer
isbn character varying(13)
loandate date
returndate date
status character varying(20)

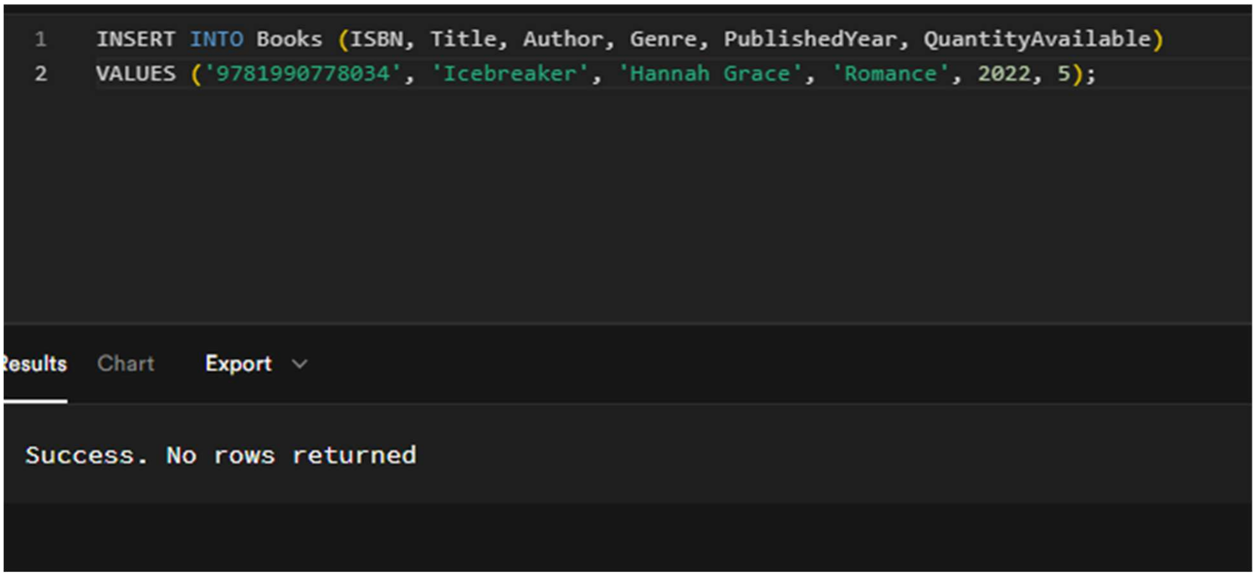
Part 2: Logical Design



```sql
CREATE TABLE IF NOT EXISTS Books (
    ISBN VARCHAR(13) PRIMARY KEY, -- Unique identifier for each book
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255) NOT NULL,
    Genre VARCHAR(50) NOT NULL,
    PublishedYear INT NOT NULL,
    QuantityAvailable INT NOT NULL CHECK (QuantityAvailable >= 0)
);
CREATE TABLE IF NOT EXISTS Users (
    UserID INT PRIMARY KEY, -- Unique ID for each user
    FullName VARCHAR(255) NOT NULL,
    EmailAddress VARCHAR(255) NOT NULL UNIQUE,
    MembershipDate DATE NULL
);
CREATE TABLE IF NOT EXISTS Loans (
    UserID INT NOT NULL,
    ISBN VARCHAR(13) NOT NULL,
    LoanDate DATE NOT NULL DEFAULT CURRENT_DATE, -- Default is today's date
    ReturnDate DATE NOT NULL DEFAULT (CURRENT_DATE + INTERVAL '14 days'), -- Default ReturnDate is 14 days from today
    Status VARCHAR(20) NOT NULL DEFAULT 'Borrowed', -- Default status is 'Borrowed'
    FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,
    FOREIGN KEY (ISBN) REFERENCES Books(ISBN) ON DELETE CASCADE
);
```

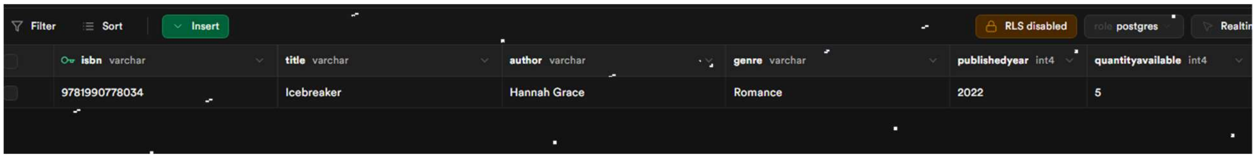
Part 3: SQL Queries

a. Insert a new book.



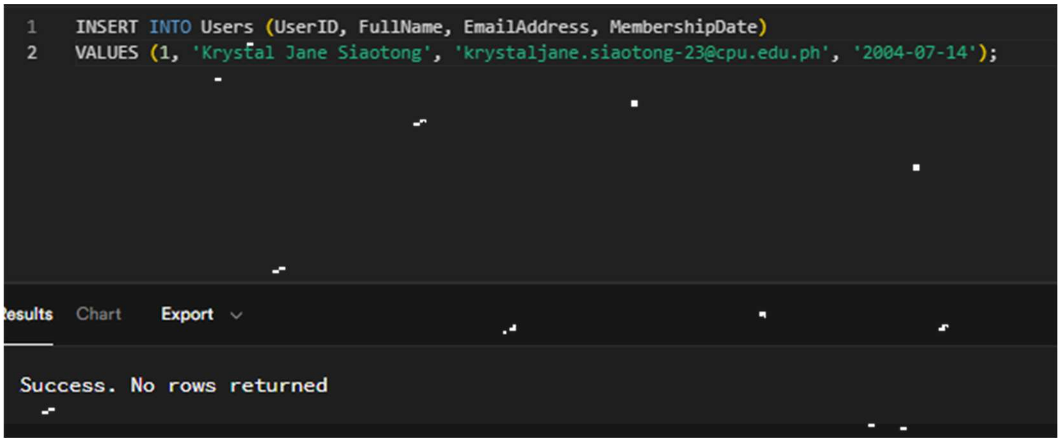
```sql
INSERT INTO Books (ISBN, Title, Author, Genre, PublishedYear, QuantityAvailable)
VALUES ('9781990778034', 'Icebreaker', 'Hannah Grace', 'Romance', 2022, 5);
```

```
Success. No rows returned
```



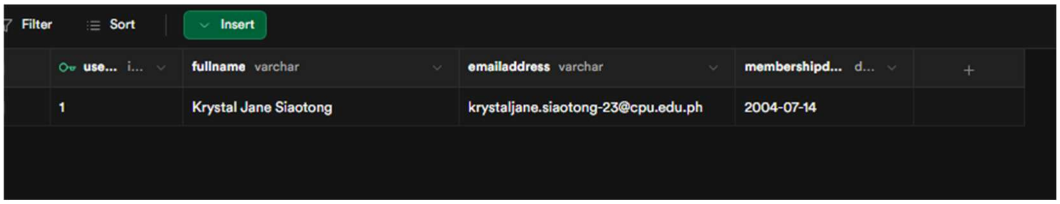
isbn varchar	title varchar	author varchar	genre varchar	publishedyear int4	quantityavailable int4
9781990778034	Icebreaker	Hannah Grace	Romance	2022	5

b.



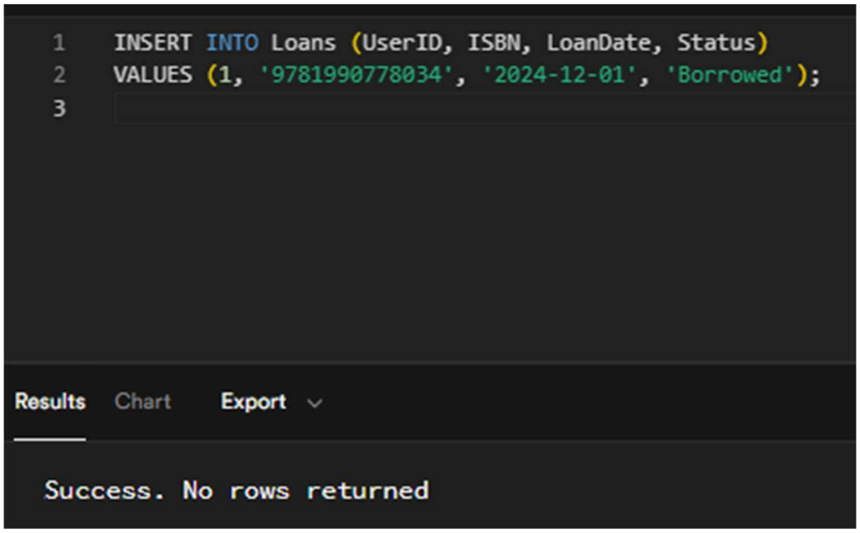
```sql
INSERT INTO Users (UserID, FullName, EmailAddress, MembershipDate)
VALUES (1, 'Krystal Jane Siaotong', 'krystaljane.siaotong-23@cpu.edu.ph', '2004-07-14');
```

```
Success. No rows returned
```



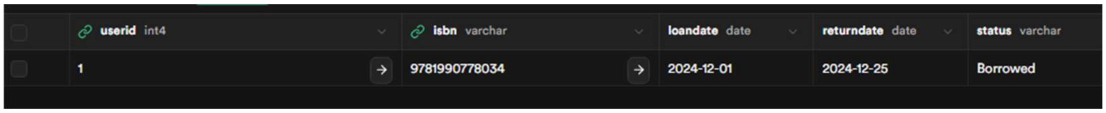
use... i...	fullname varchar	emailaddress varchar	membershipd... d...
1	Krystal Jane Siaotong	krystaljane.siaotong-23@cpu.edu.ph	2004-07-14

c. Record a book loan for a user.



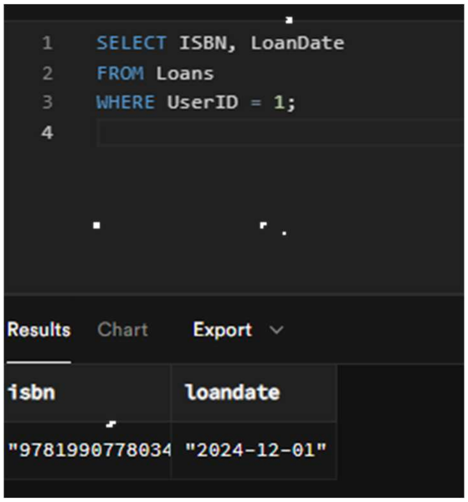
```sql
INSERT INTO Loans (UserID, ISBN, LoanDate, Status)
VALUES (1, '9781990778034', '2024-12-01', 'Borrowed');
```

```
Success. No rows returned
```



userid int4	isbn varchar	loandate date	returndate date	status varchar
1	9781990778034	2024-12-01	2024-12-25	Borrowed

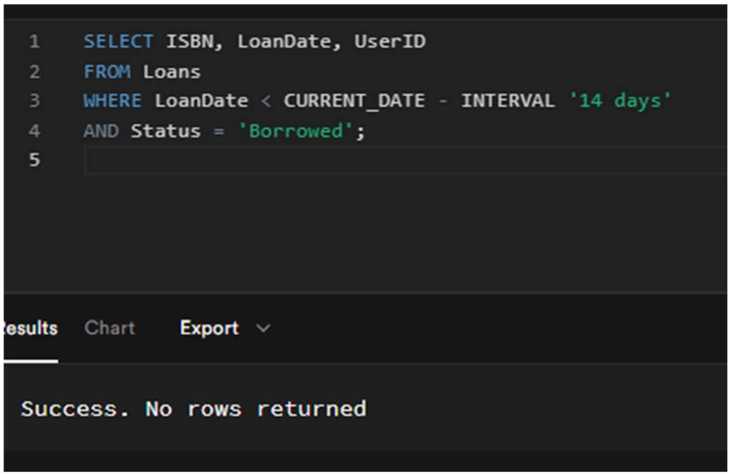
d. Find all books borrowed by a specific user.



```sql
SELECT ISBN, LoanDate
FROM Loans
WHERE UserID = 1;
```

isbn	loandate
"9781990778034	"2024-12-01"

e. List all overdue.

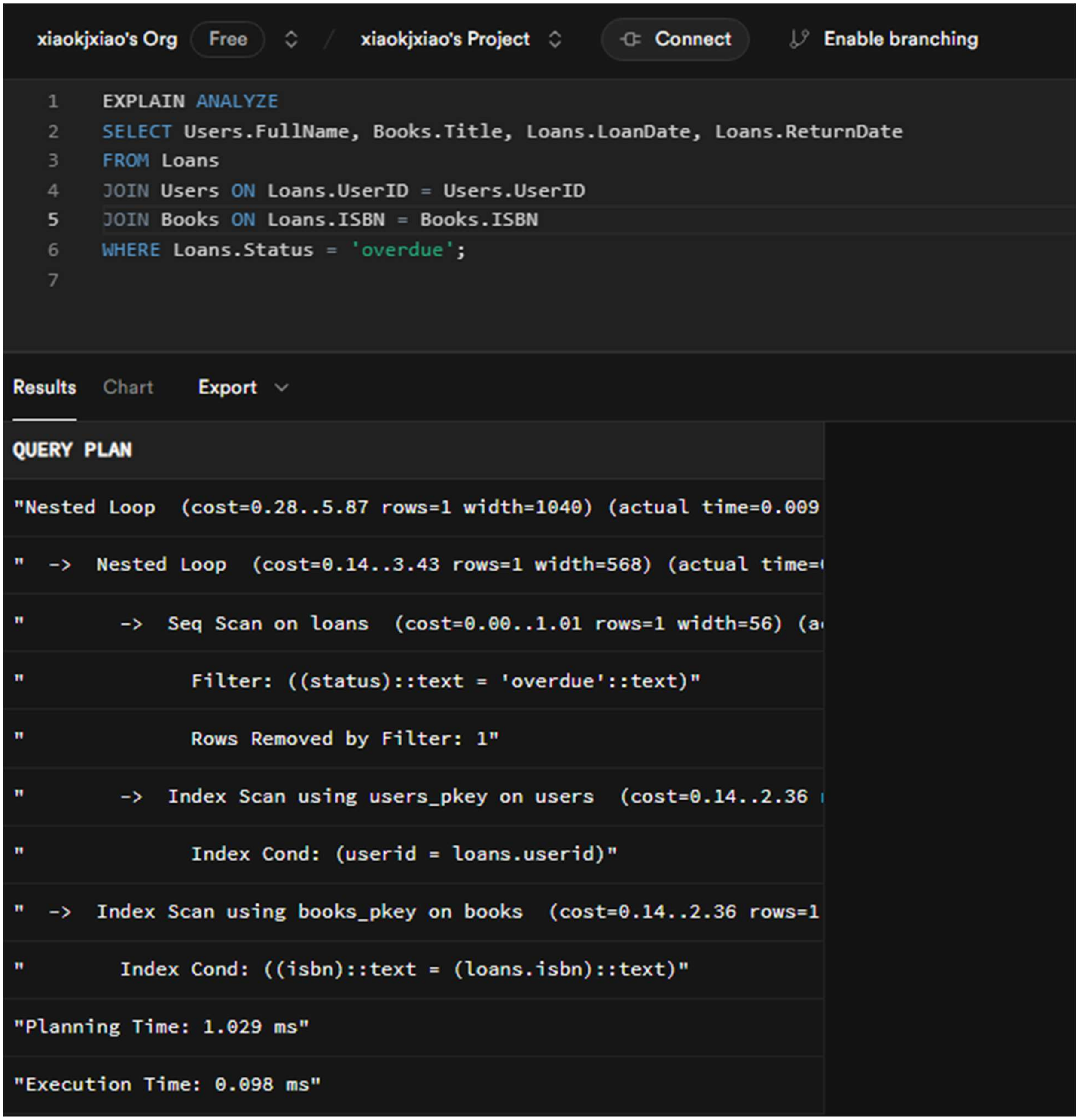


```sql
SELECT ISBN, LoanDate, UserID
FROM Loans
WHERE LoanDate < CURRENT_DATE - INTERVAL '14 days'
AND Status = 'Borrowed';
```

```
Success. No rows returned
```

4. Data Integrity and Optimization

Libraries use technology and systematic management to ensure that books are available before allowing them to be borrowed, preventing the disappointment of finding no copies left. They maintain a database to track the number of copies available, check availability before lending, and manage multiple requests with transactions to avoid confusion. Automatic rules, known as triggers, instantly notify patrons if a book is unavailable, while program logic within their systems verifies book availability before permitting a loan. These strategies ensure that library patrons can only borrow books that are currently in stock.p



```
EXPLAIN ANALYZE
SELECT Users.FullName, Books.Title, Loans.LoanDate, Loans.ReturnDate
FROM Loans
JOIN Users ON Loans.UserID = Users.UserID
JOIN Books ON Loans.ISBN = Books.ISBN
WHERE Loans.Status = 'overdue';
```

```
QUERY PLAN
"Nested Loop  (cost=0.28..5.87 rows=1 width=1040) (actual time=0.009
"  ->  Nested Loop  (cost=0.14..3.43 rows=1 width=568) (actual time=
"        ->  Seq Scan on loans  (cost=0.00..1.01 rows=1 width=56) (a
"              Filter: ((status)::text = 'overdue'::text)"
"              Rows Removed by Filter: 1"
"        ->  Index Scan using users_pkey on users  (cost=0.14..2.36
"              Index Cond: (userid = loans.userid)"
"  ->  Index Scan using books_pkey on books  (cost=0.14..2.36 rows=1
"        Index Cond: ((isbn)::text = (loans.isbn)::text)"
"Planning Time: 1.029 ms"
"Execution Time: 0.098 ms"
```

Part 5: Reflection

As a library management system scales to millions of users and books, it faces challenges like performance issues, data consistency, and backup needs. To enhance efficiency, indexing on key columns (ISBN, UserID, LoanDate) can speed up queries, and database sharding helps manage data loads. To maintain consistency during simultaneous transactions, solutions like transactions, optimistic locking, and row-level locks are essential. Foreign keys and triggers ensure data integrity, automating adjustments for book quantities. Incremental backups and replication safeguard against data loss, while read replicas and load balancing improve performance under increased operations. Careful planning for data migration and upgrades prevents downtime, and utilizing cloud services offers cost-effective scaling solutions. By implementing these strategies, a library management system can effectively accommodate a growing user base.