LIBRARY DBMS

CMPE343 REPORT

0. Project Members and Contributions

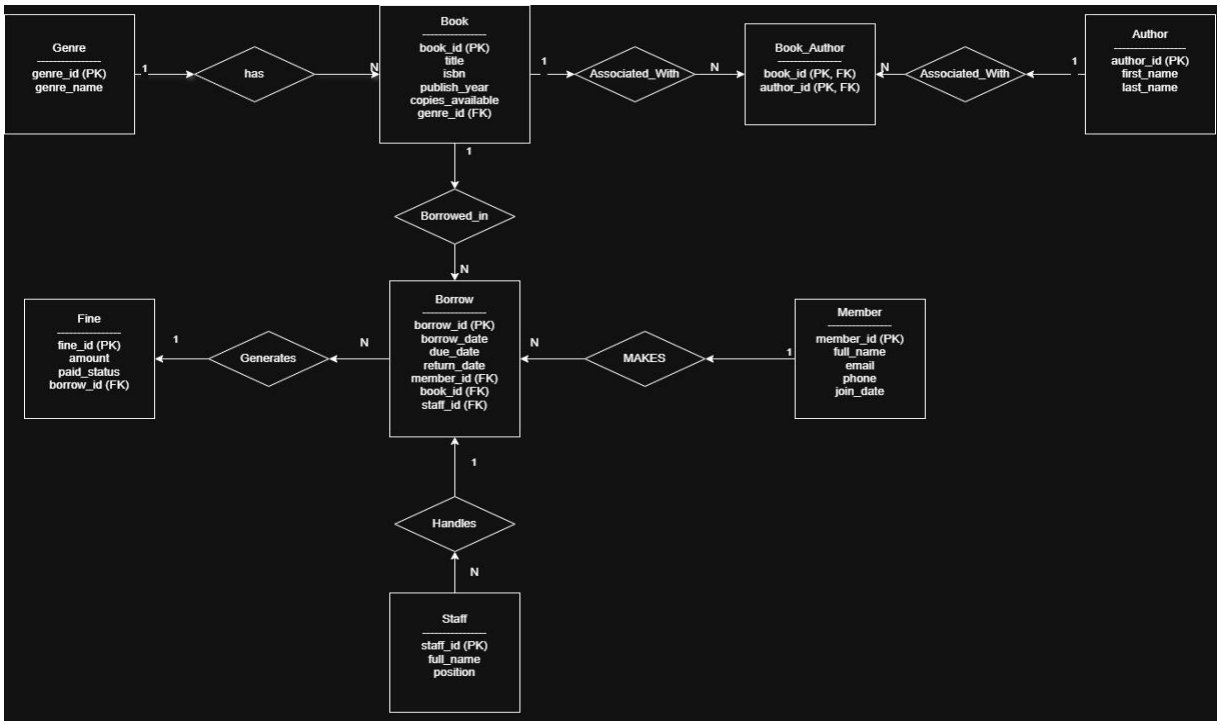
- TRESOR KALOMBO KALALA 22512651 : DDL, DML,QUERIES(1-5)
- Sultan Makhmut 22305640: ERD,DML,QUERIES(6-10)
- Dzaan Zachariah Allam 22120765: Report, DML,QUERIES(11-15)

1. Introduction

This Library Database Management System (DBMS) was developed using PostgreSQL for the CMPE343 course. The system manages authors, books, genres, members, staff, borrow transactions, and fines. Assumptions include unique IDs for each entity, proper constraints for data integrity, and realistic sample data.

2. Database

- ### ER Diagram



Data Definition Language (DDL)

```
-- TABLE 1: AUTHOR : Stores book authors

CREATE TABLE Author (

author_id INT PRIMARY KEY, ----------Unique author ID

first_name VARCHAR(50) NOT NULL, ----------Author first name

last_name VARCHAR(50) NOT NULL ----------Author last name

);

----------

-- TABLE 2: BOOK: Stores books in the library

CREATE TABLE Book (

book_id INT PRIMARY KEY, ----------Unique book ID

title VARCHAR(100) NOT NULL, ----------Book title

isbn VARCHAR(20) UNIQUE, ----------ISBN number (unique)

publish_year INT, ----------Year of publication

copies_available INT DEFAULT 1 ----------Available copies

CHECK (copies_available >= 0) ----------Cannot be negative

);

----------

-- TABLE 3: MEMBER: Stores library members

CREATE TABLE Member (

member_id INT PRIMARY KEY, ----------Unique member ID

full_name VARCHAR(100) NOT NULL, ----------Member name

email VARCHAR(100) UNIQUE, ----------Email (unique)

phone VARCHAR(20), ----------Phone number

join_date DATE DEFAULT CURRENT_DATE ----------Date joined

);
```

```
----------------------------
--TABLE 4: STAFF: Stores library staff
CREATE TABLE Staff (
staff_id INT PRIMARY KEY, ----------Unique staff ID
full_name VARCHAR(100) NOT NULL, ----------Staff name
position VARCHAR(50) -- Job position
);
----------------------------
-- TABLE 5: BOOK_AUTHOR : Connects books and authors (many-to-many)
CREATE TABLE Book_Author (
book_id INT, ----------Book ID
author_id INT, ----------Author ID
PRIMARY KEY (book_id, author_id), ----------Composite primary key
FOREIGN KEY (book_id) REFERENCES Book(book_id),
FOREIGN KEY (author_id) REFERENCES Author(author_id)
);
------------------------------
--TABLE 6: BORROW: Stores borrow transactions
CREATE TABLE Borrow (
borrow_id INT PRIMARY KEY, ----------Borrow ID
member_id INT, ----------Member borrowing
book_id INT, ----------Borrowed book
staff_id INT, ----------Staff handling
borrow_date DATE NOT NULL, ----------Borrow date
due_date DATE NOT NULL, ----------Due date
return_date DATE, ----------Return date
```

```
FOREIGN KEY (member_id) REFERENCES Member(member_id),
FOREIGN KEY (book_id) REFERENCES Book(book_id),
FOREIGN KEY (staff_id) REFERENCES Staff(staff_id)
);
----------------------------------
--TABLE 7: FINE : Stores fines for late returns
CREATE TABLE Fine (
fine_id INT PRIMARY KEY, --------------------Fine ID
borrow_id INT UNIQUE, --------------------Related borrow
amount DECIMAL(6,2), --------------------Fine amount
paid_status VARCHAR(10) DEFAULT 'UNPAID',
FOREIGN KEY (borrow_id) REFERENCES Borrow(borrow_id)
);
---------------------------------------
-- TABLE 8: GENRE : Stores book genres
CREATE TABLE Genre (
   genre_id INT PRIMARY KEY,         -- Unique genre ID
   genre_name VARCHAR(50) NOT NULL    -- Genre name
);
```

- **Data Manipulation Language (DML)**

```
-- Authors
INSERT INTO Author VALUES (1, 'Tresor', 'Kalombo');
INSERT INTO Author VALUES (2, 'Fyodor', 'Dostoevsky');
INSERT INTO Author VALUES (3, 'Friedrich', 'Nietzsche');
INSERT INTO Author VALUES (4, 'Jane', 'Austen');
```

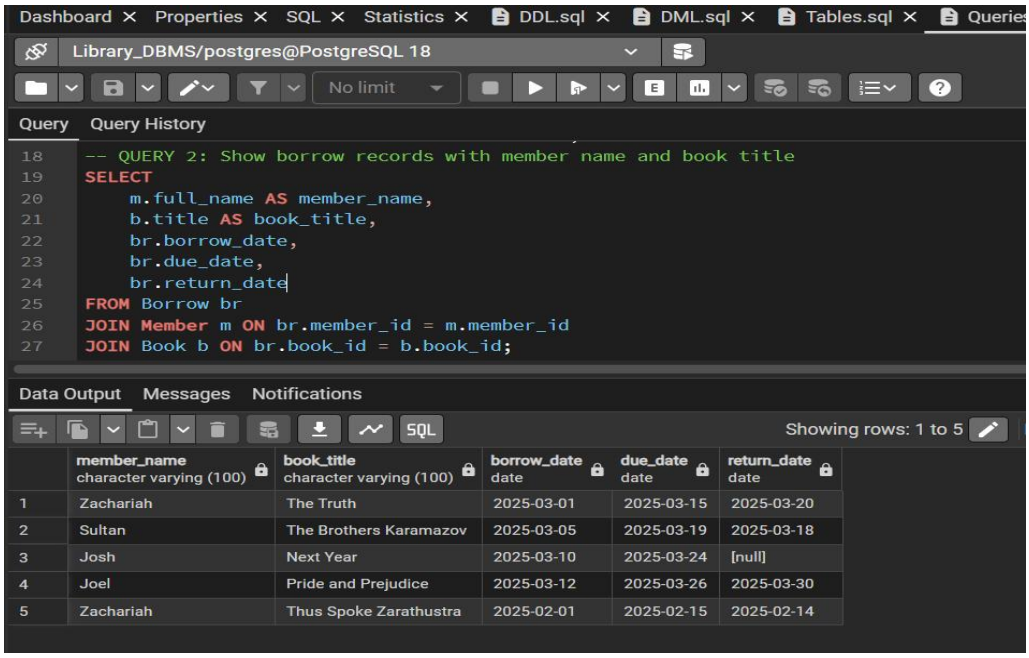
```
INSERT INTO Author VALUES (5, 'Albert', 'Camus');


-- Books

INSERT INTO Book VALUES (1, 'The Truth', '000077770000', 2057, 7);

INSERT INTO Book VALUES (2, 'The Brothers Karamazov', '000077770001', 1880, 5);

INSERT INTO Book VALUES (3, 'Next Year', '000077770002', 2059, 4);

INSERT INTO Book VALUES (4, 'Pride and Prejudice', '000077770003', 1813, 6);

INSERT INTO Book VALUES (5, 'The Stranger', '000077770004', 1942, 0); -- no copies
available

INSERT INTO Book VALUES (6, 'Thus Spoke Zarathustra', '000077770005', 1883, 1);

INSERT INTO Book VALUES (7, 'Notes from Underground', '000077770006', 1864, 0); -- no
copies available


-- Connect books with authors

-- Author 1 wrote two books

INSERT INTO Book_Author VALUES (1, 1);

INSERT INTO Book_Author VALUES (3, 1);

-- Author 2 wrote two books

INSERT INTO Book_Author VALUES (2, 2);

INSERT INTO Book_Author VALUES (7, 2);

-- Other authors

INSERT INTO Book_Author VALUES (6, 3); -- Nietzsche

INSERT INTO Book_Author VALUES (4, 4); -- Austen

INSERT INTO Book_Author VALUES (5, 5); -- Camus


---Genre

-- Add genre reference to Book table
```

```
ALTER TABLE Book

ADD COLUMN genre_id INT;

-- Add foreign key constraint

ALTER TABLE Book

ADD CONSTRAINT fk_book_genre

FOREIGN KEY (genre_id) REFERENCES Genre(genre_id);

-- Insert genres

INSERT INTO Genre VALUES (1, 'Philosophy');

INSERT INTO Genre VALUES (2, 'Classic Literature');

INSERT INTO Genre VALUES (3, 'Fiction');

INSERT INTO Genre VALUES (4, 'Romance');

INSERT INTO Genre VALUES (5, 'Psychological');

-- Assign genres to books

UPDATE Book SET genre_id = 3 WHERE book_id = 1; -- The Truth (Fiction)

UPDATE Book SET genre_id = 2 WHERE book_id = 2; -- The Brothers Karamazov (Classic)

UPDATE Book SET genre_id = 3 WHERE book_id = 3; -- Next Year (Fiction)

UPDATE Book SET genre_id = 4 WHERE book_id = 4; -- Pride and Prejudice (Romance)

UPDATE Book SET genre_id = 5 WHERE book_id = 5; -- The Stranger (Psychological)

UPDATE Book SET genre_id = 1 WHERE book_id = 6; -- Thus Spoke Zarathustra (Philosophy)

UPDATE Book SET genre_id = 2 WHERE book_id = 7; -- Notes from Underground (Classic)



-- Members

INSERT INTO Member VALUES (1, 'Zachariah', '22120765@student.ciu.edu.tr',
'+2347061614885', '2025-10-07');
```

```
INSERT INTO Member VALUES (2, 'Sultan', '22305640@student.ciu.edu.tr',
'+77052175021', '2025-09-07');

INSERT INTO Member VALUES (3, 'Josh', 'josh@student.ciu.edu.tr', '+9000000000', '2025-
08-01');

INSERT INTO Member VALUES (4, 'Joel', 'joel@student.ciu.edu.tr', '+904444444446', '2024-
07-15');

-- staff members

INSERT INTO Staff VALUES (1, 'Jacob Bush', 'Assistant');

INSERT INTO Staff VALUES (2, 'Aline Koj', 'Librarian');

INSERT INTO Staff VALUES (3, 'Mark Koop', 'Supervisor');

INSERT INTO Staff VALUES (4, 'Jil Umar', 'Manager');


-- Borrow

INSERT INTO Borrow VALUES (1, 1, 1, 1, '2025-03-01', '2025-03-15', '2025-03-20');

INSERT INTO Borrow VALUES (2, 2, 2, 2, '2025-03-05', '2025-03-19', '2025-03-18');

INSERT INTO Borrow VALUES (3, 3, 3, 3, '2025-03-10', '2025-03-24', NULL);

INSERT INTO Borrow VALUES (4, 4, 4, 4, '2025-03-12', '2025-03-26', '2025-03-30');

INSERT INTO Borrow VALUES (5, 1, 6, 2, '2025-02-01', '2025-02-15', '2025-02-14');


-- Fines

INSERT INTO Fine VALUES (1, 1, 5.00, 'UNPAID');

INSERT INTO Fine VALUES (2, 4, 7.00, 'PAID');

INSERT INTO Fine VALUES (3, 3, 3.50, 'UNPAID');
```

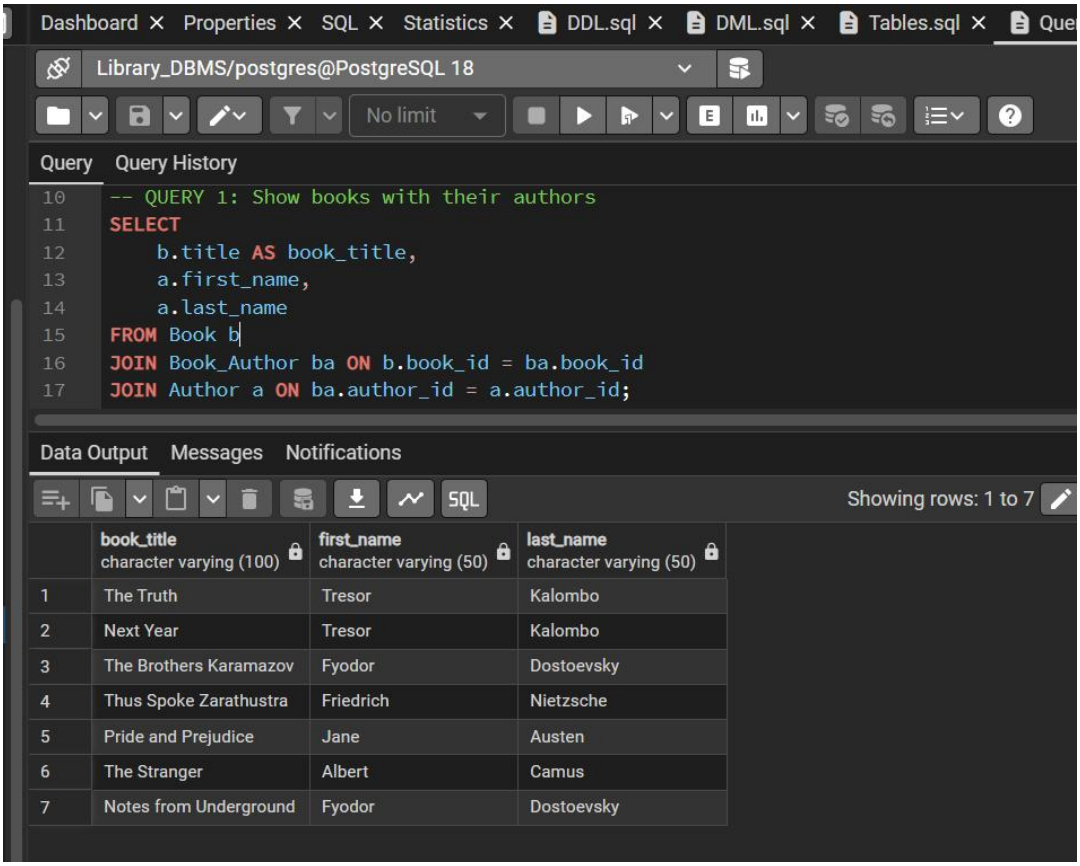
3. Queries

- ### Query 1



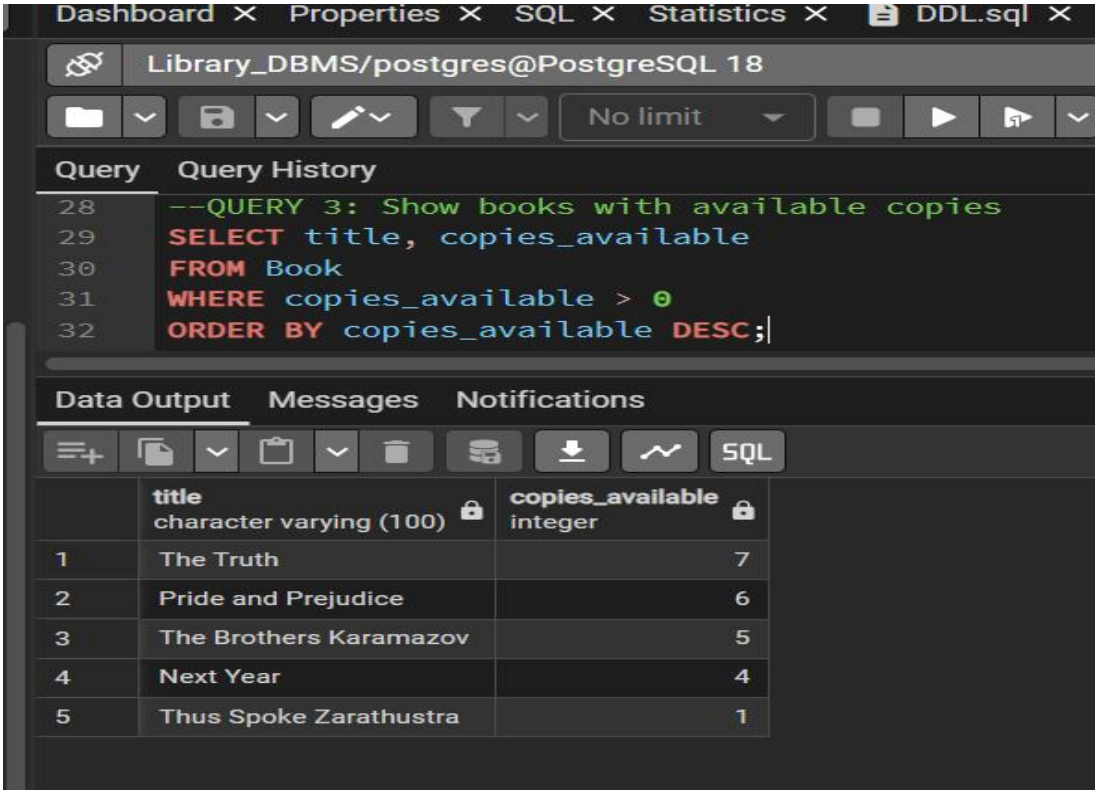
	member_name character varying (100)	book_title character varying (100)	borrow_date date	due_date date	return_date date
1	Zachariah	The Truth	2025-03-01	2025-03-15	2025-03-20
2	Sultan	The Brothers Karamazov	2025-03-05	2025-03-19	2025-03-18
3	Josh	Next Year	2025-03-10	2025-03-24	[null]
4	Joel	Pride and Prejudice	2025-03-12	2025-03-26	2025-03-30
5	Zachariah	Thus Spoke Zarathustra	2025-02-01	2025-02-15	2025-02-14

- ### Query 2

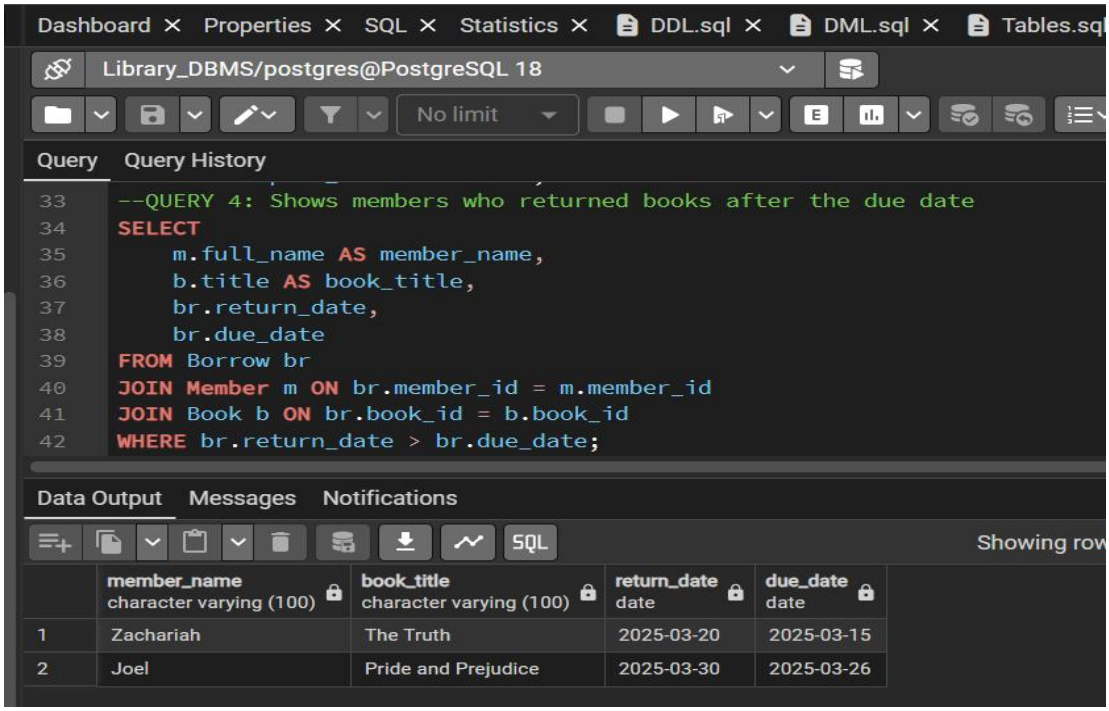


	book_title character varying (100)	first_name character varying (50)	last_name character varying (50)
1	The Truth	Tresor	Kalombo
2	Next Year	Tresor	Kalombo
3	The Brothers Karamazov	Fyodor	Dostoevsky
4	Thus Spoke Zarathustra	Friedrich	Nietzsche
5	Pride and Prejudice	Jane	Austen
6	The Stranger	Albert	Camus
7	Notes from Underground	Fyodor	Dostoevsky

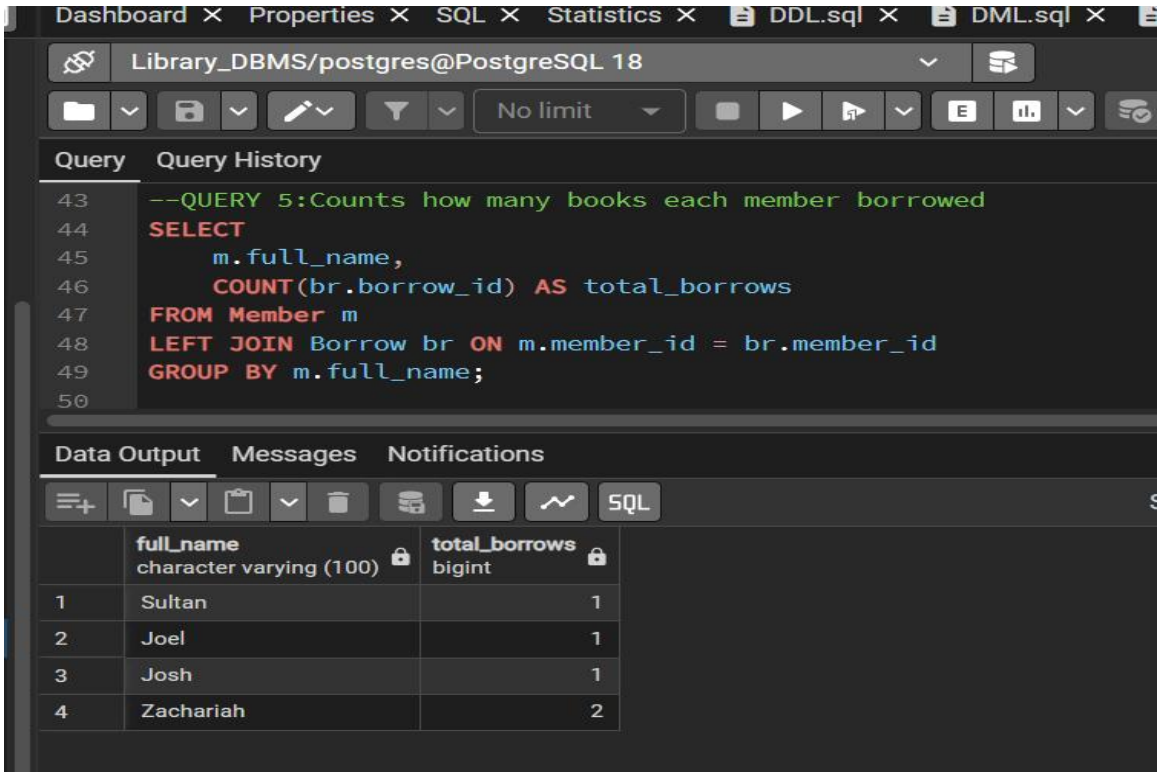
Query 3



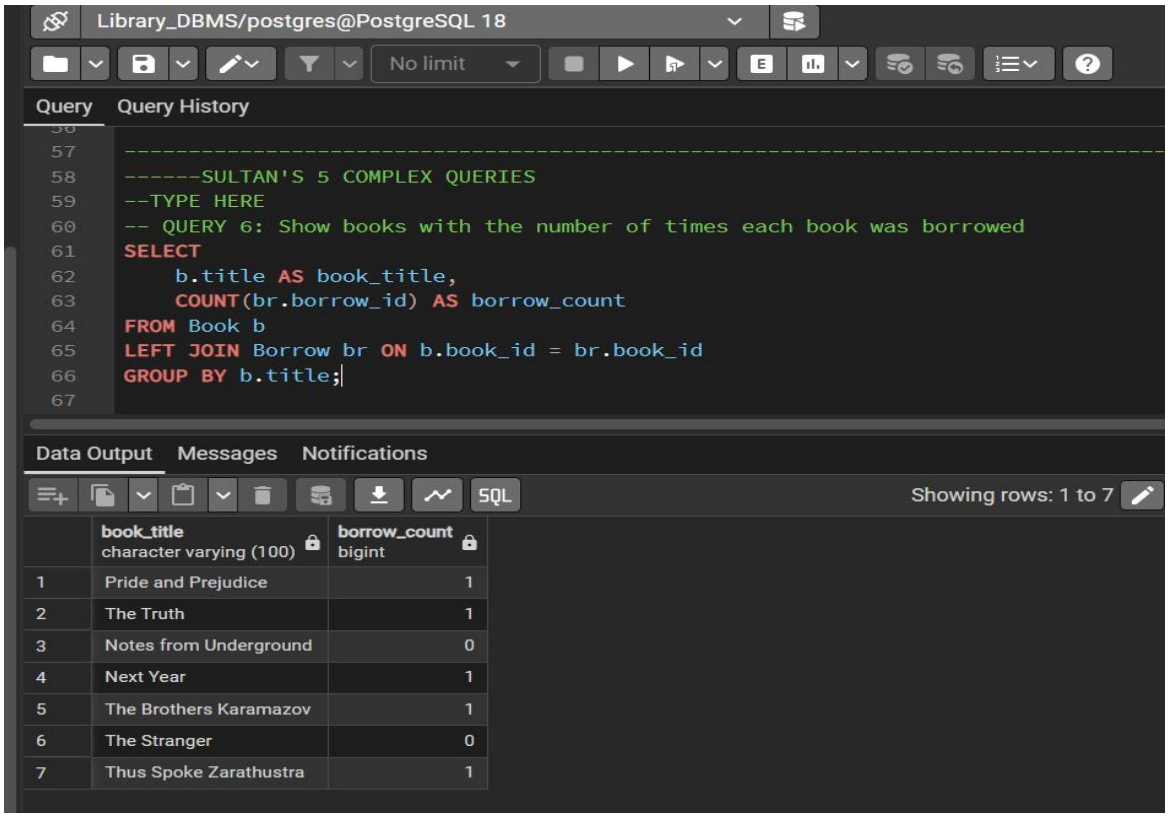
Query 4



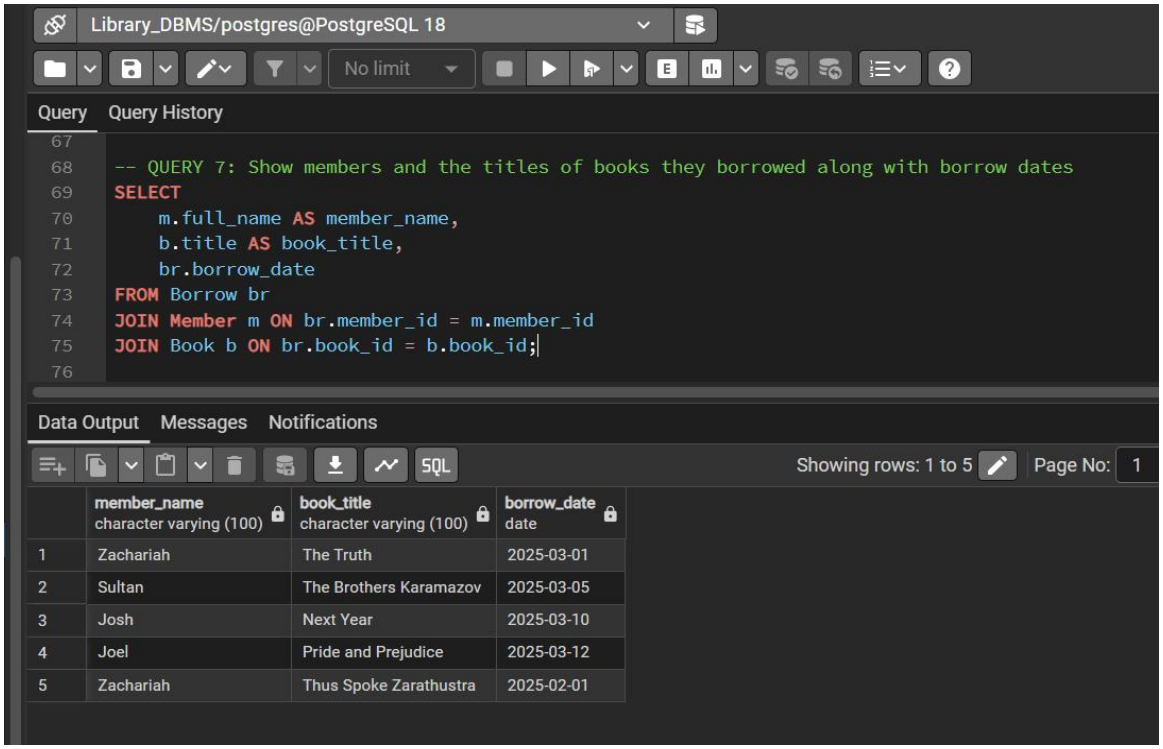
Query 5



Query 6

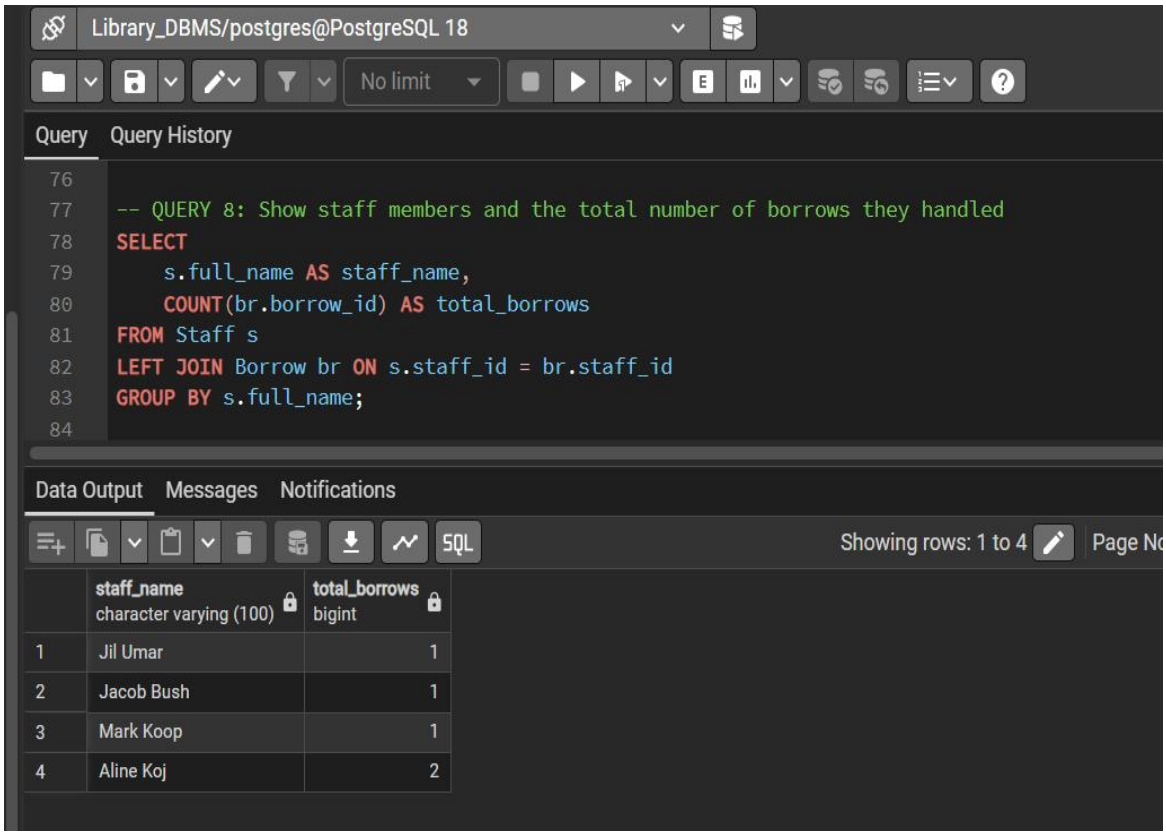


Query 7



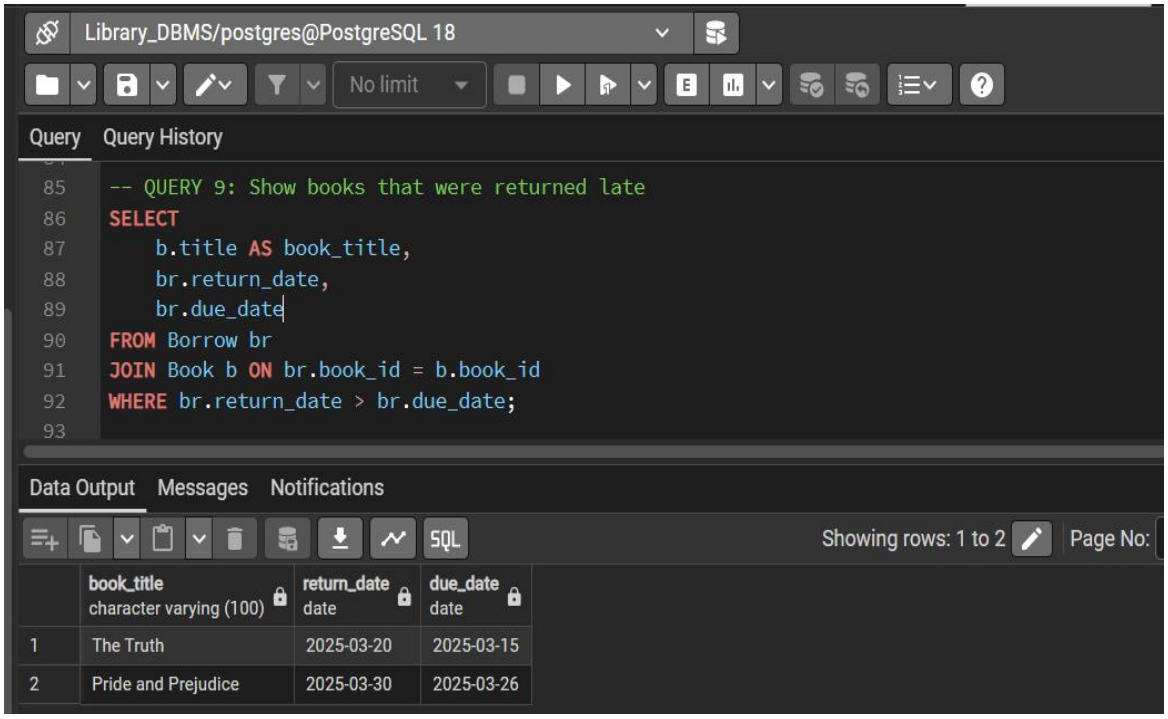
	member_name character varying (100)	book_title character varying (100)	borrow_date date
1	Zachariah	The Truth	2025-03-01
2	Sultan	The Brothers Karamazov	2025-03-05
3	Josh	Next Year	2025-03-10
4	Joel	Pride and Prejudice	2025-03-12
5	Zachariah	Thus Spoke Zarathustra	2025-02-01

Query 8

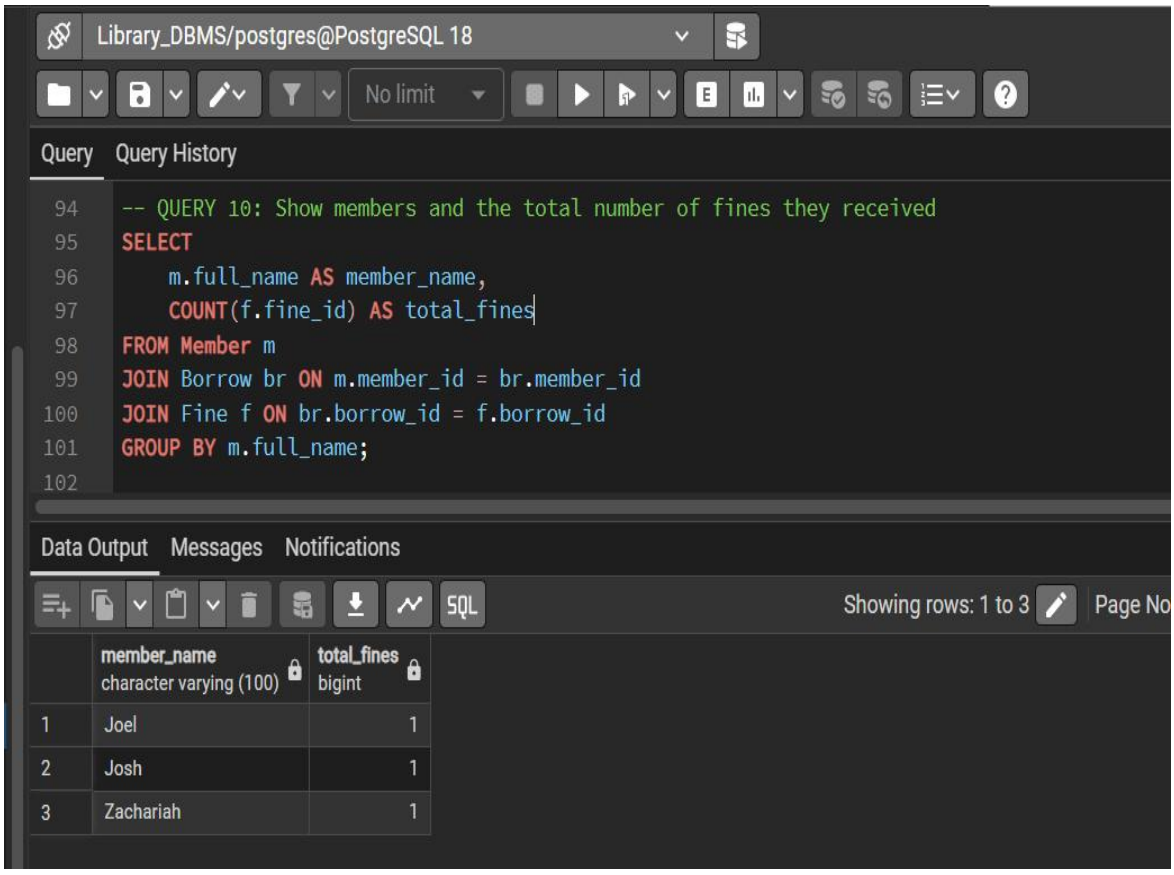


	staff_name character varying (100)	total_borrows bigint
1	Jil Umar	1
2	Jacob Bush	1
3	Mark Koop	1
4	Aline Koj	2

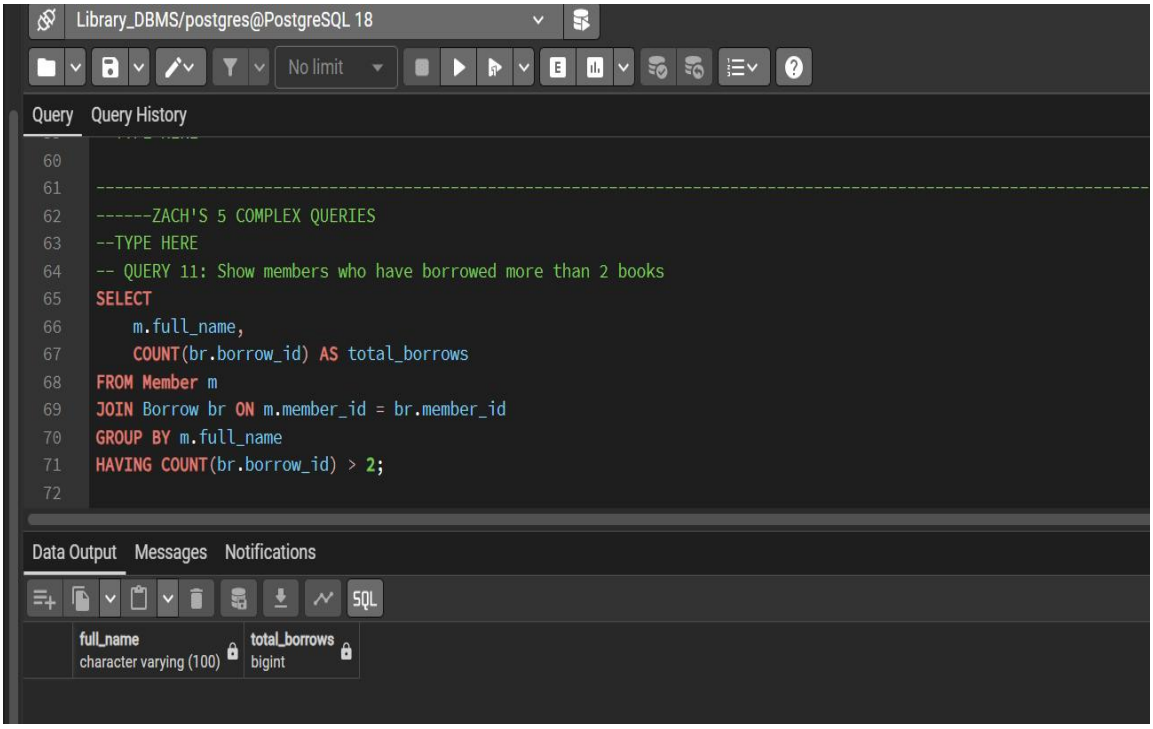
Query 9



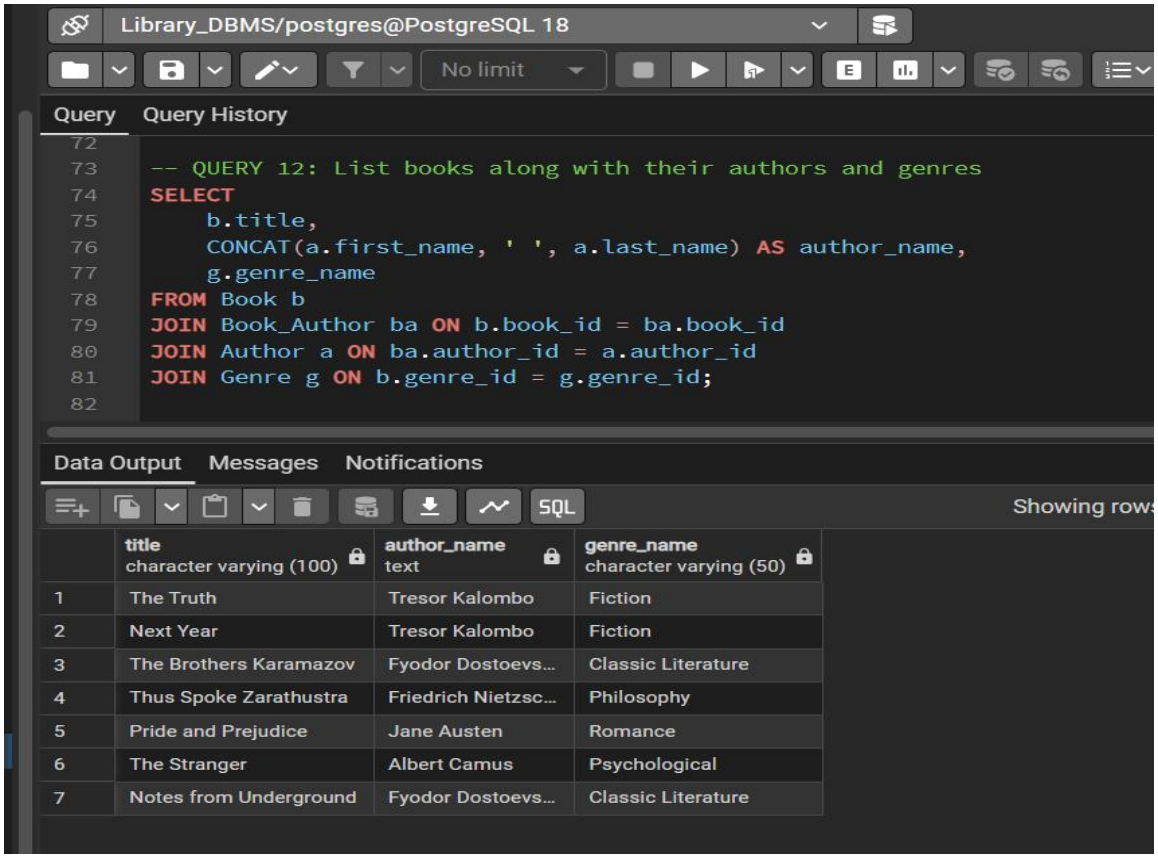
Query 10



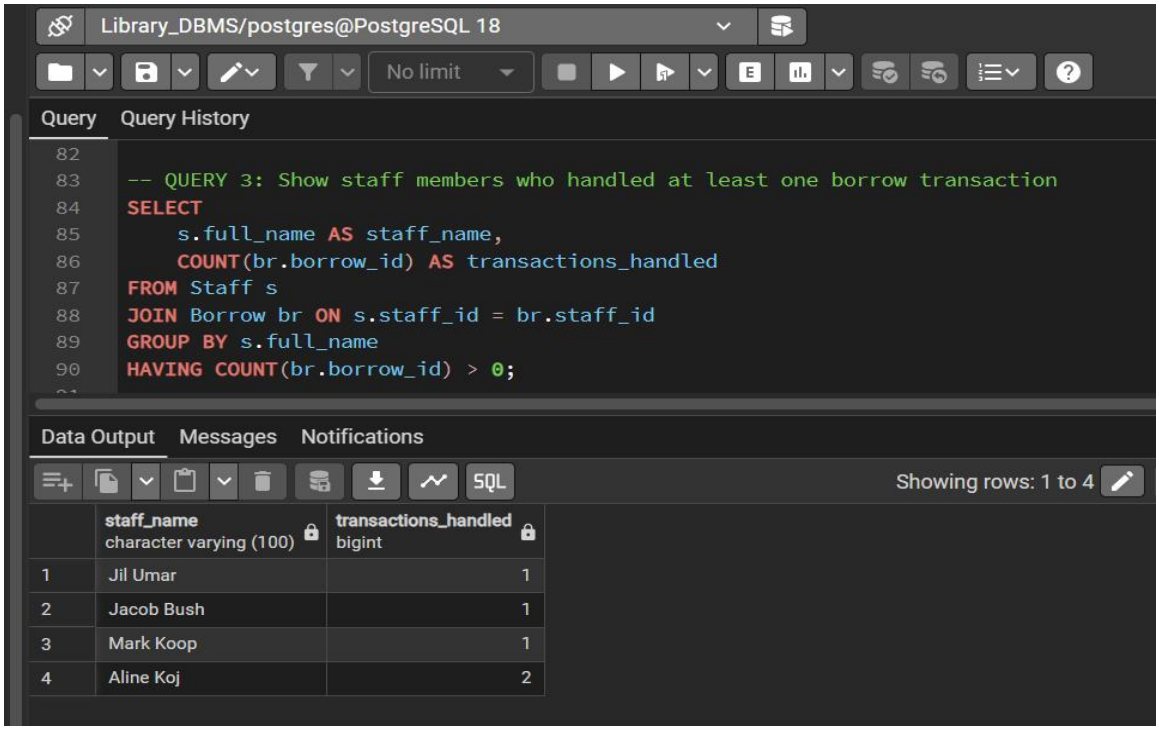
Query 11



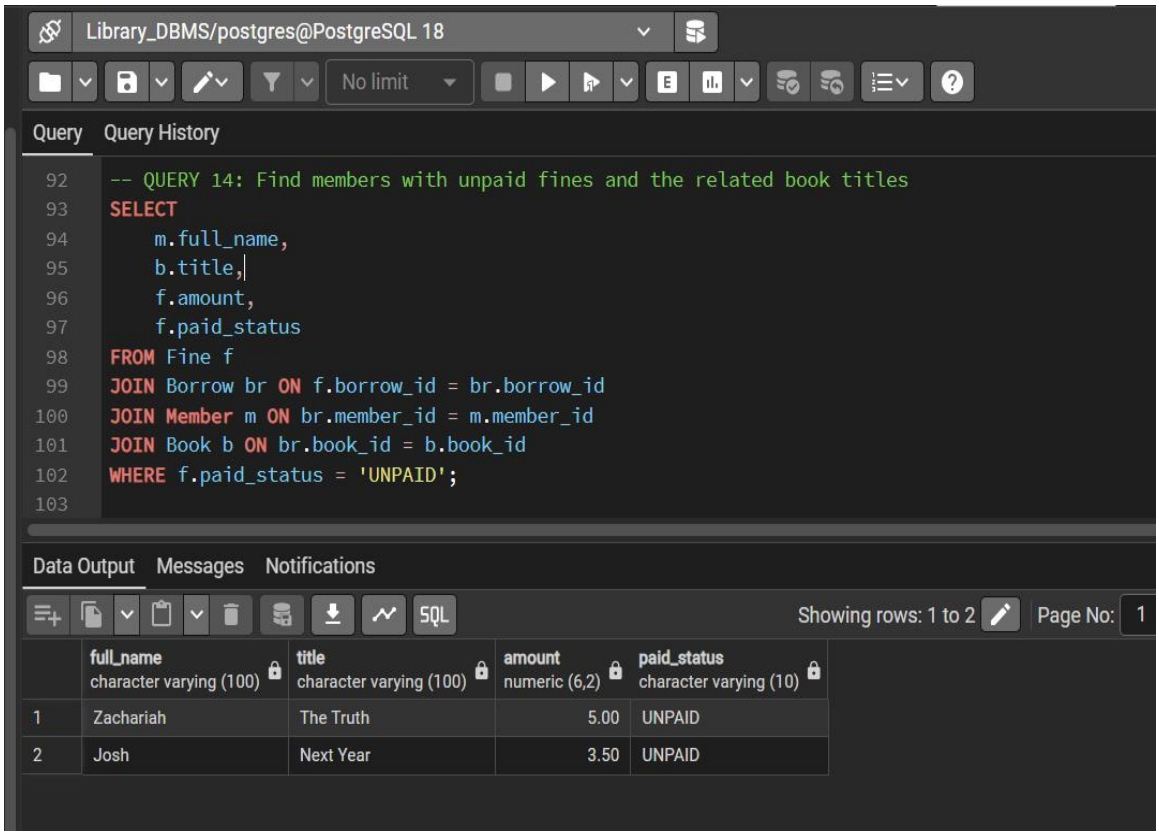
Query 12



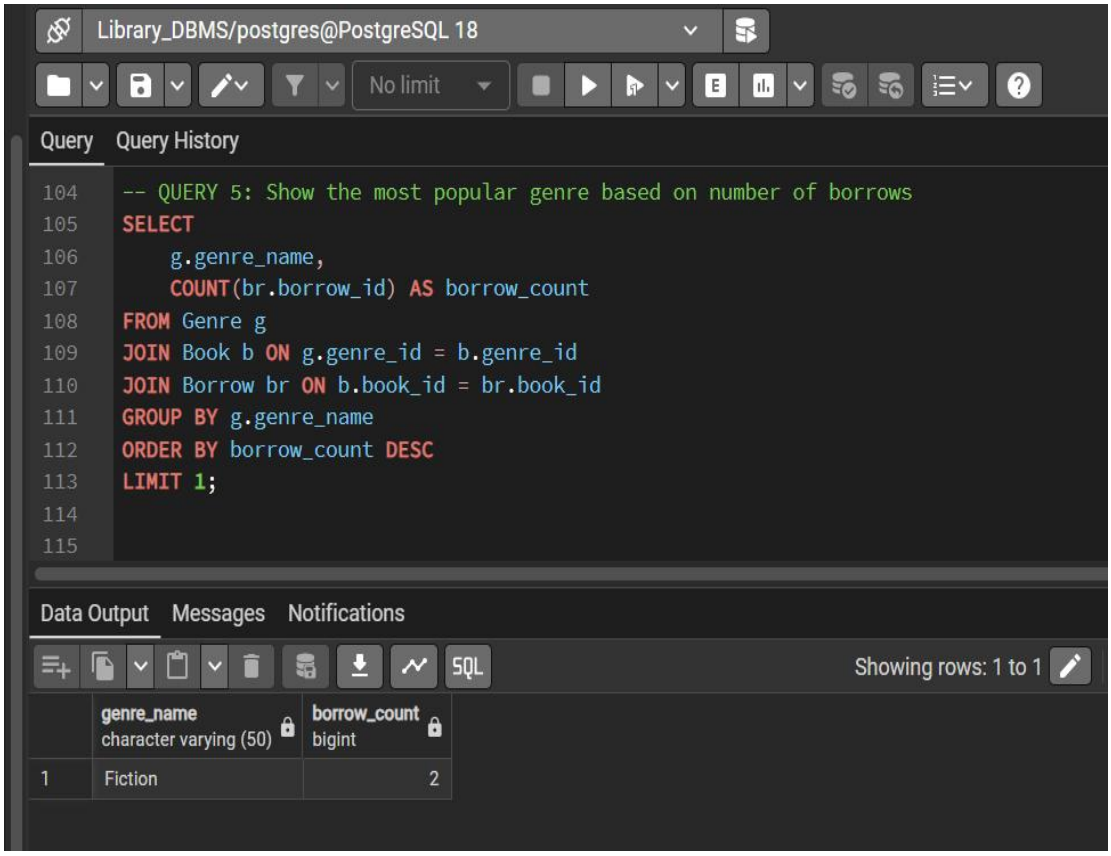
Query 13



Query 14



Query 15


Library_DBMS/postgres@PostgreSQL 18
No limit
Query
Query History
104 -- QUERY 5: Show the most popular genre based on number of borrows
105 SELECT
106 g.genre_name,
107 COUNT(br.borrow_id) AS borrow_count
108 FROM Genre g
109 JOIN Book b ON g.genre_id = b.genre_id
110 JOIN Borrow br ON b.book_id = br.book_id
111 GROUP BY g.genre_name
112 ORDER BY borrow_count DESC
113 LIMIT 1;
114
115
Data Output
Messages
Notifications
SQL
Showing rows: 1 to 1
genre_name
character varying (50)
borrow_count
bigint
1
Fiction
2