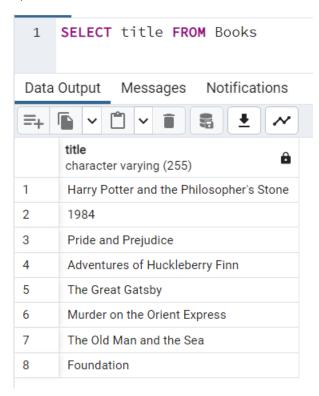
2.

a)



b)

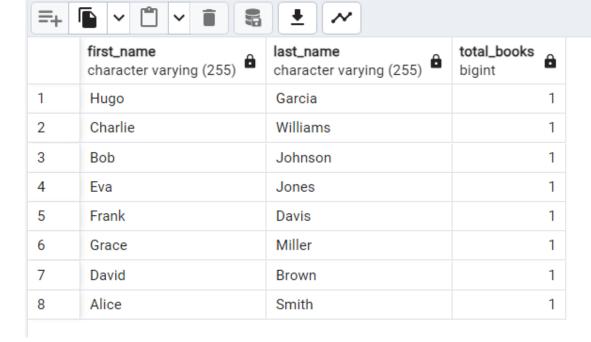
1 SELECT first\_name, last\_name FROM Members WHERE join\_date > '2023-01-01' Messages Data Output Notifications <u>+</u> last\_name first\_name character varying (255) character varying (255) 1 Alice Smith 2 Bob Johnson 3 Charlie Williams David 4 Brown 5 Eva Jones 6 Frank Davis 7 Miller Grace 8 Hugo Garcia

c)

## Query Query History

```
--coalesce function will handle the NULL if a member does
1
2
   SELECT
       m.first_name,
3
       m.last_name,
4
5
       COALESCE(COUNT(b.borrow_id), 0) AS total_books
6
   FROM Members m
   LEFT JOIN Borrowing b ON m.member_id = b.member_id
7
   GROUP BY m.first_name, m.last_name
8
```

## Data Output Messages Notifications

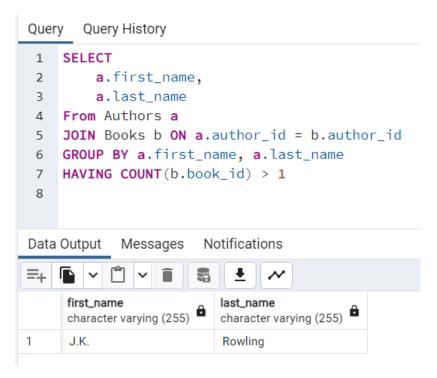


d)

inserted another book for jk rowling for testing purposes

```
Query Query History

1 INSERT INTO Books (title, author_id, publisher_id, published_date, isbn, available_
2 VALUES
3 ('Harry Potter and the big chungus', 1, 1, '1997-06-26', '97805632699', 10)
```

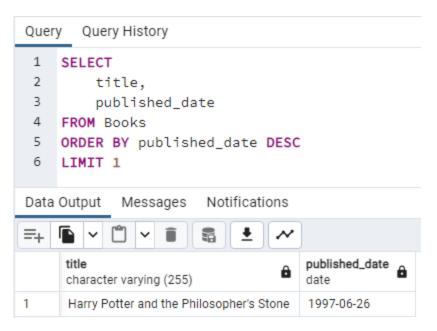


e)

Inserted a new member who doesn't have any book borrowed

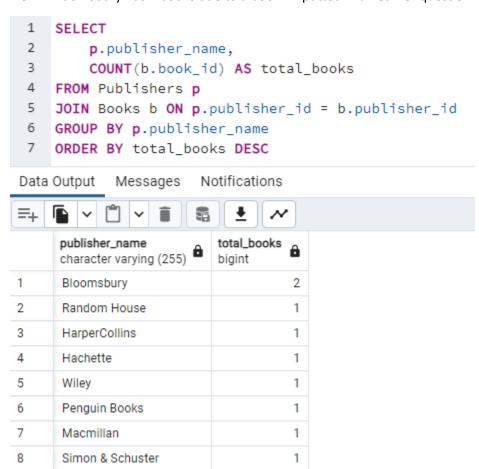
```
INSERT INTO Members (first_name, last_name, email, phone, join_date)
1
2
   ('Ian', 'Hectox', 'Smosh@email.com', '123-456-2580', '2024-08-25')
3
Query Query History
1
   SELECT
2
       m.first_name,
3
       m.last_name
4 FROM Members m
   LEFT JOIN Borrowing b ON m.member_id = b.member_id
   WHERE borrow_id is NULL
Data Output Messages Notifications
character varying (255)
                    character varying (255)
                     Hectox
```

f)



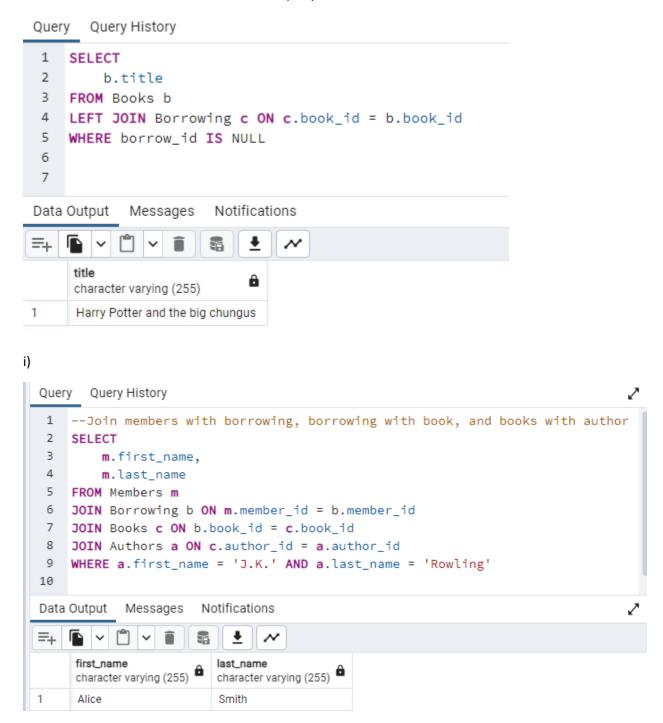
g)

NOTE: Bloomsbury has 2 books due to a book I inputted in an earlier question



## h)

NOTE: book is from new value inserted in a query above



j)

Inserted 3 more values into borrowing so we can test

```
Query Query History
 1 INSERT INTO Borrowing (book_id, member_id, borrow_date, return_date)
 2 VALUES
 3 (1, 5, '2023-09-01', NULL),
 4 (1, 4, '2023-09-01', NULL),
 5 (1, 3, '2023-09-01', NULL),
 6 (1, 2, '2023-09-01', NULL)
Query Query History
 1
    SELECT
 2
         a.first_name,
 3
         a.last name
 4 FROM Authors a
 5 LEFT JOIN Books b ON a.author_id = b.author_id
 6 LEFT JOIN Borrowing c ON b.book_id = c.book_id
 7 GROUP BY a.first_name, a.last_name
    HAVING COUNT(c.borrow_id) > 3
 9
10
Data Output
           Messages
                      Notifications
=+
                         last_name
     first_name
                         character varying (255)
     character varying (255)
1
     J.K.
                         Rowling
```