Library Management

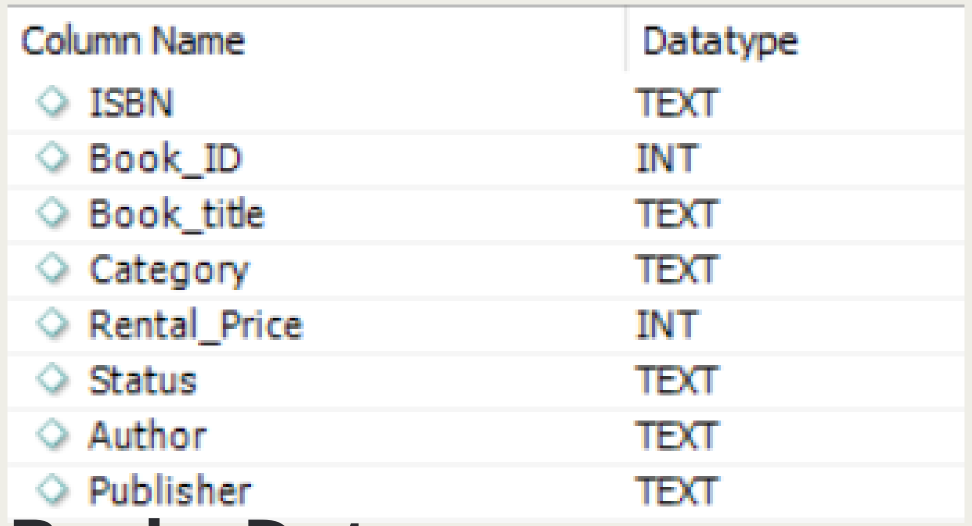
library Managment report using sql

In this project, I have utilized SQL queries to analyze various aspects of library management. This includes tracking inventory data and Book returns, calculating average return times, identifying the most active customers, and determining the most common issue dates. By leveraging SQL, I was able to extract meaningful insights from the library's data, which can help improve operational efficiency and enhance user experience.

AGENDA

- Dataset Overview
- Inventory Management
- Branch/Employee Management
- Book Management
- Library Overview

DATASET OVERVIEW



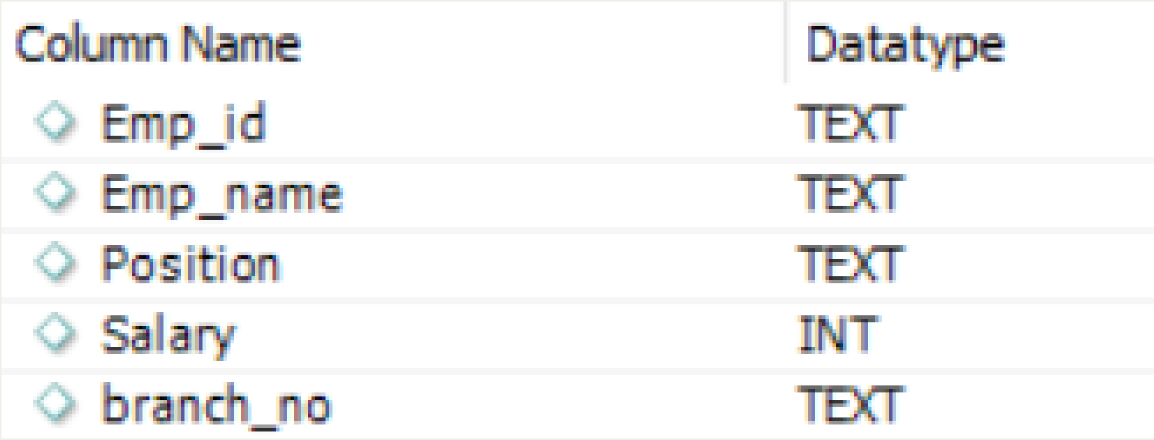
Column Name	Datatype
ISBN	TEXT
Book_ID	INT
Book_title	TEXT
Category	TEXT
Rental_Price	INT
Status	TEXT
Author	TEXT
Publisher	TEXT

Books Data

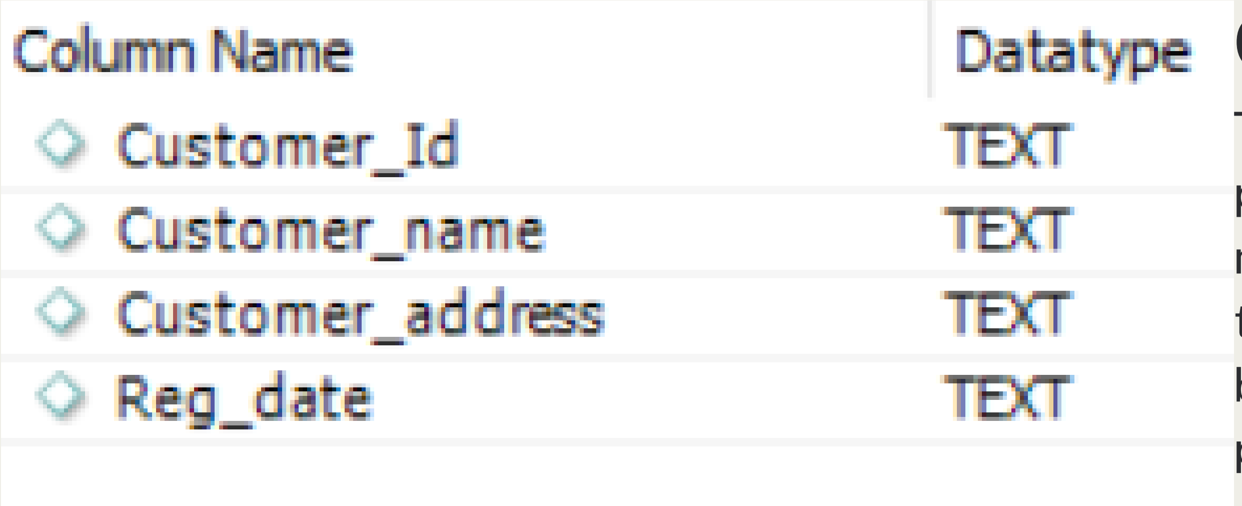
This dataset is used to manage and analyze the collection of books in the library, including details about each book’s title, author, genre, and publication year.

Employee Data

The **employee** dataset is essential for managing and analyzing the workforce within the library system



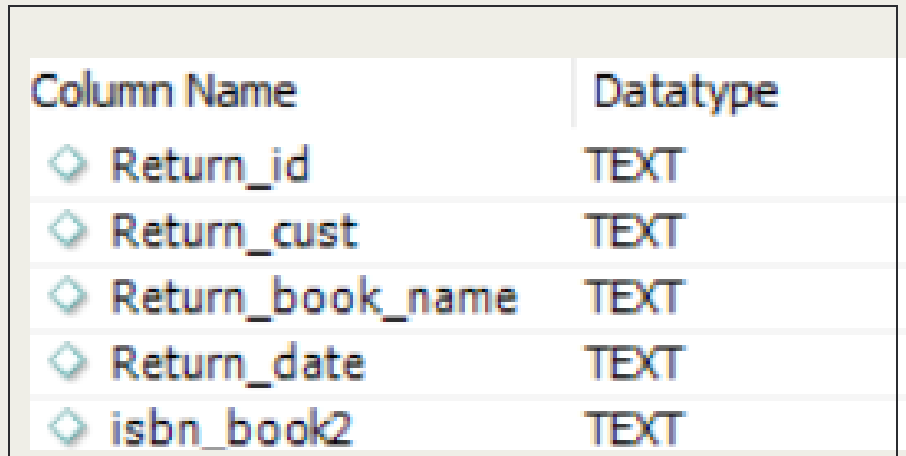
Column Name	Datatype
Emp_id	TEXT
Emp_name	TEXT
Position	TEXT
Salary	INT
branch_no	TEXT



Column Name	Datatype
Customer_Id	TEXT
Customer_name	TEXT
Customer_address	TEXT
Reg_date	TEXT

Customer Data

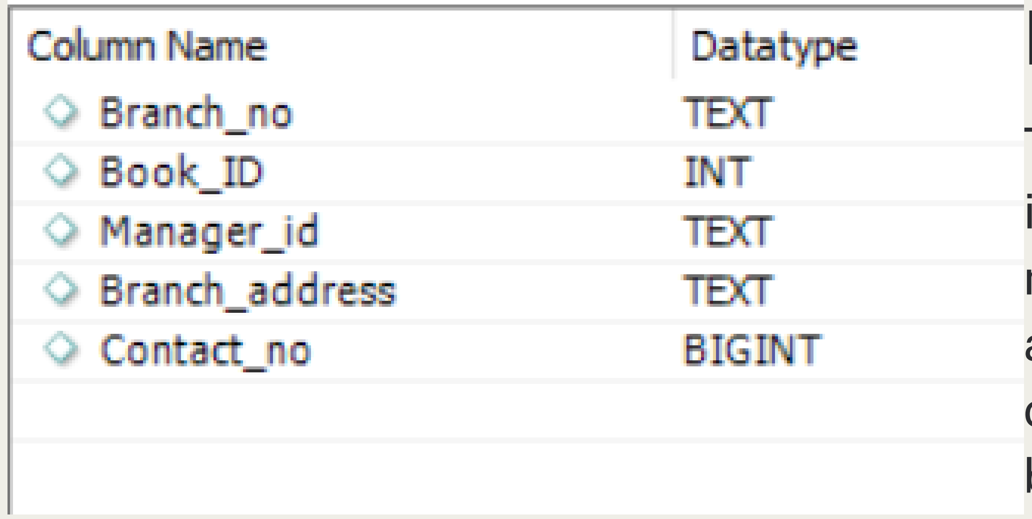
The **customer** dataset plays a vital role in managing and analyzing the interactions and behaviors of library patrons.



Column Name	Datatype
Return_id	TEXT
Return_cust	TEXT
Return_book_name	TEXT
Return_date	TEXT
isbn_book2	TEXT

Issue_status Data

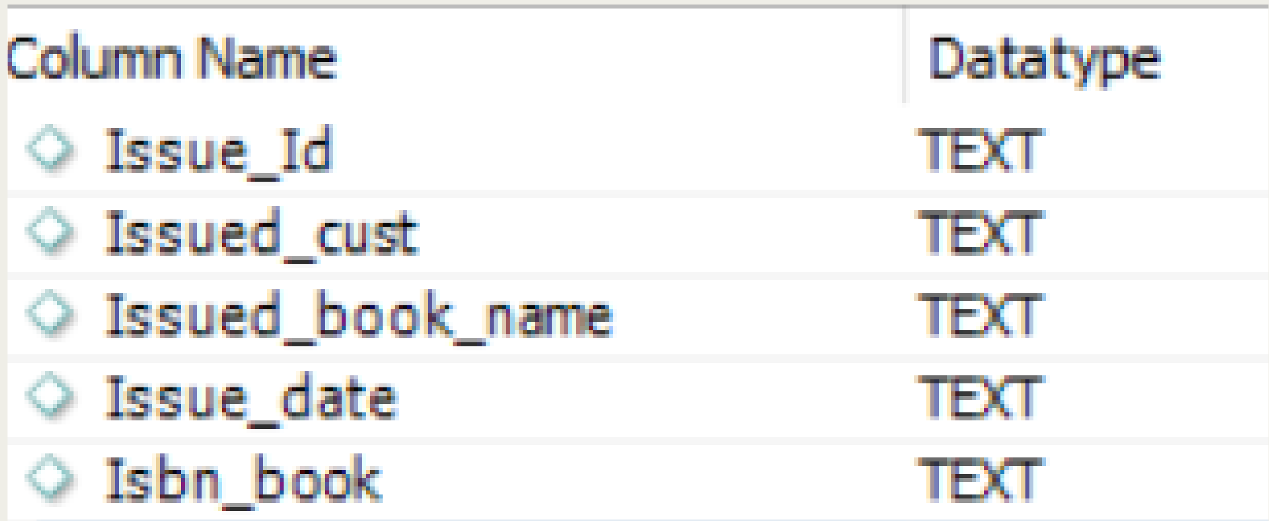
The **return_status** dataset is essential for tracking and managing the return of borrowed books within the library system.



Column Name	Datatype
Branch_no	TEXT
Book_ID	INT
Manager_id	TEXT
Branch_address	TEXT
Contact_no	BIGINT

Branch Data

The **branch** dataset is crucial for managing and analyzing the operations of library branches



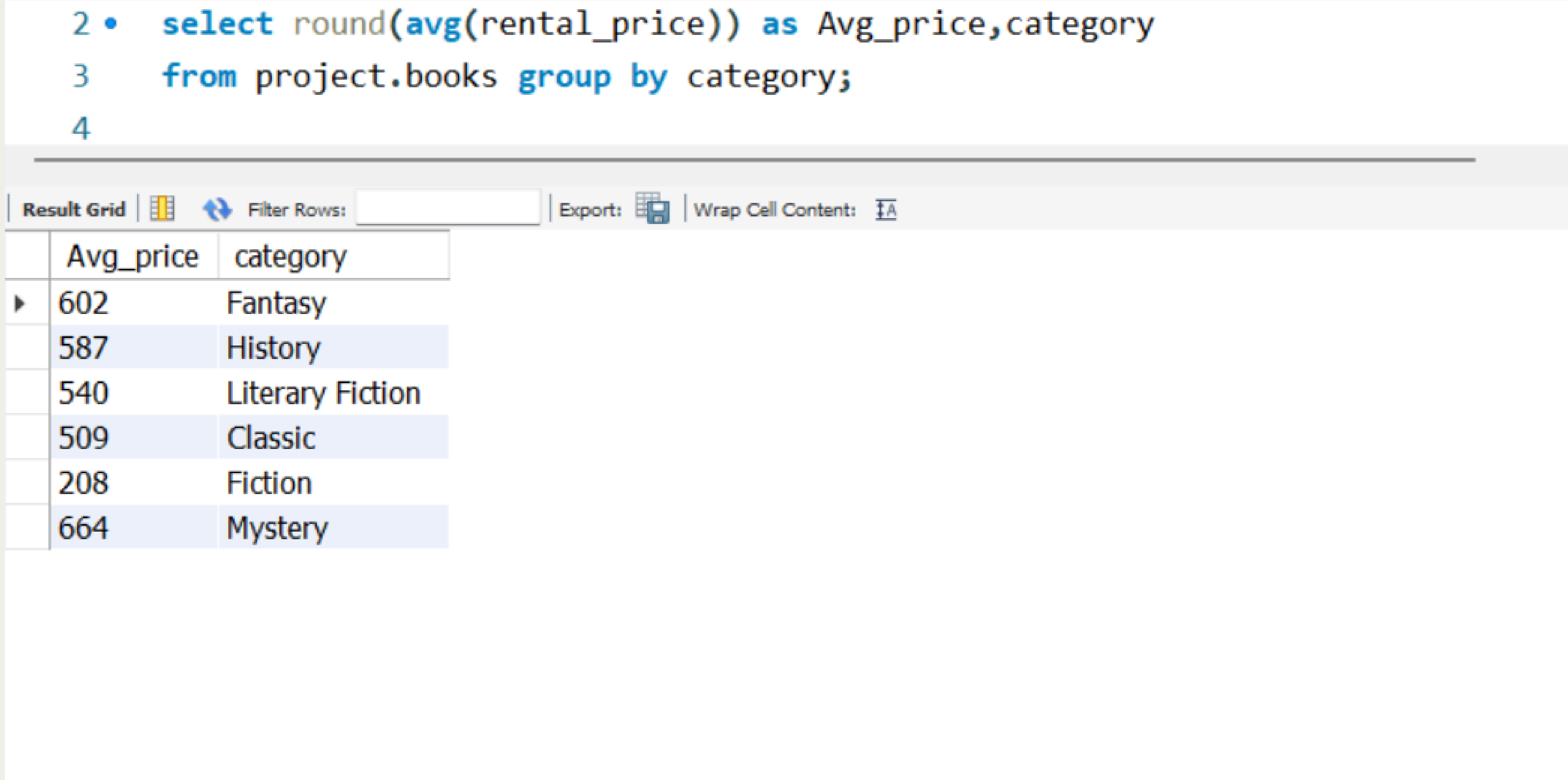
Column Name	Datatype
Issue_Id	TEXT
Issued_cust	TEXT
Issued_book_name	TEXT
Issue_date	TEXT
Isbn_book	TEXT

Issue_status Data

The issue_status dataset is crucial for tracking and managing the borrowing activities within the library.

INVENTORY MANGEMENT

Find the average book price based on genre

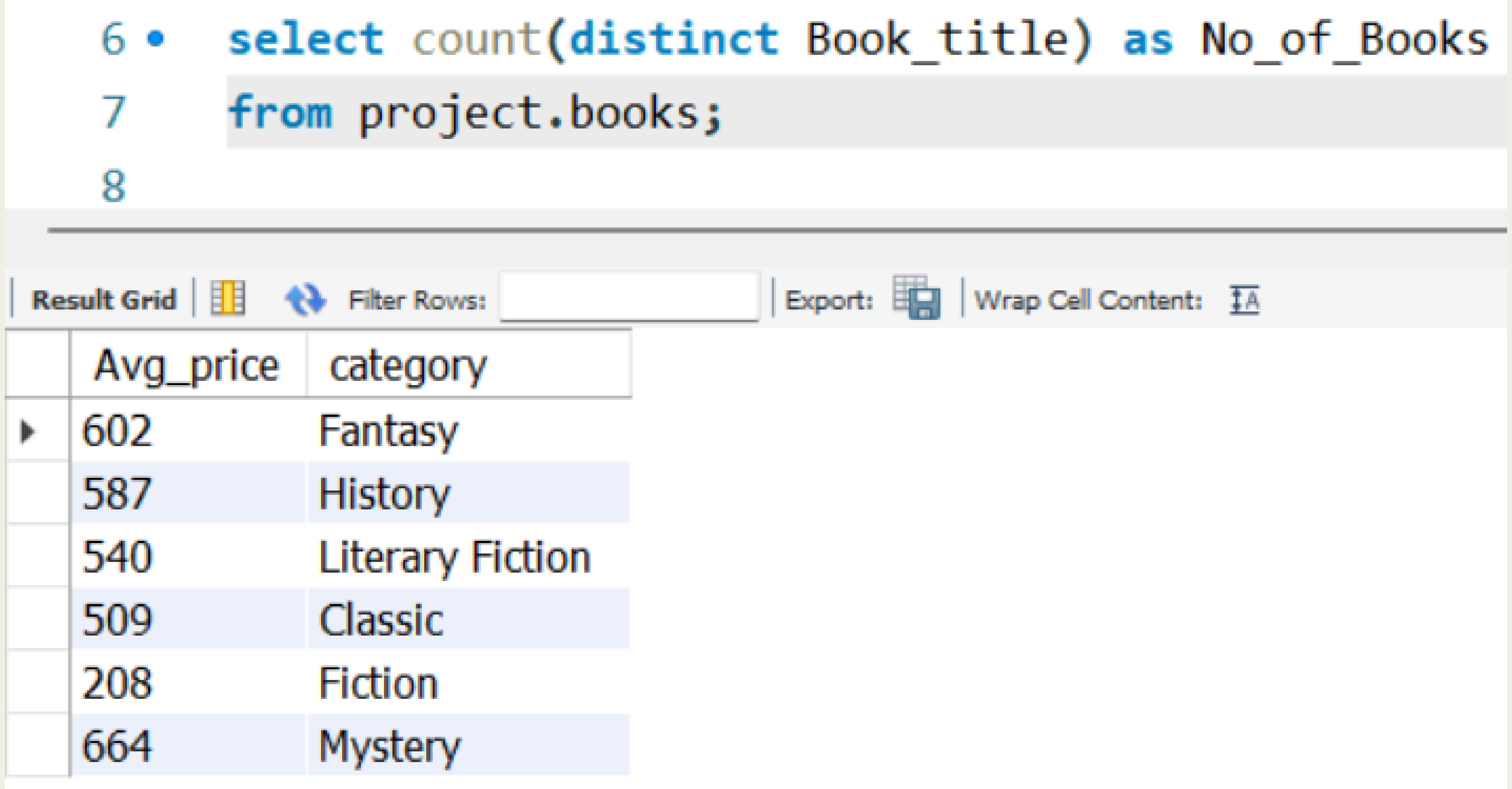


```sql
select round(avg(rental_price)) as Avg_price,category
from project.books group by category;

```

Avg_price	category
602	Fantasy
587	History
540	Literary Fiction
509	Classic
208	Fiction
664	Mystery

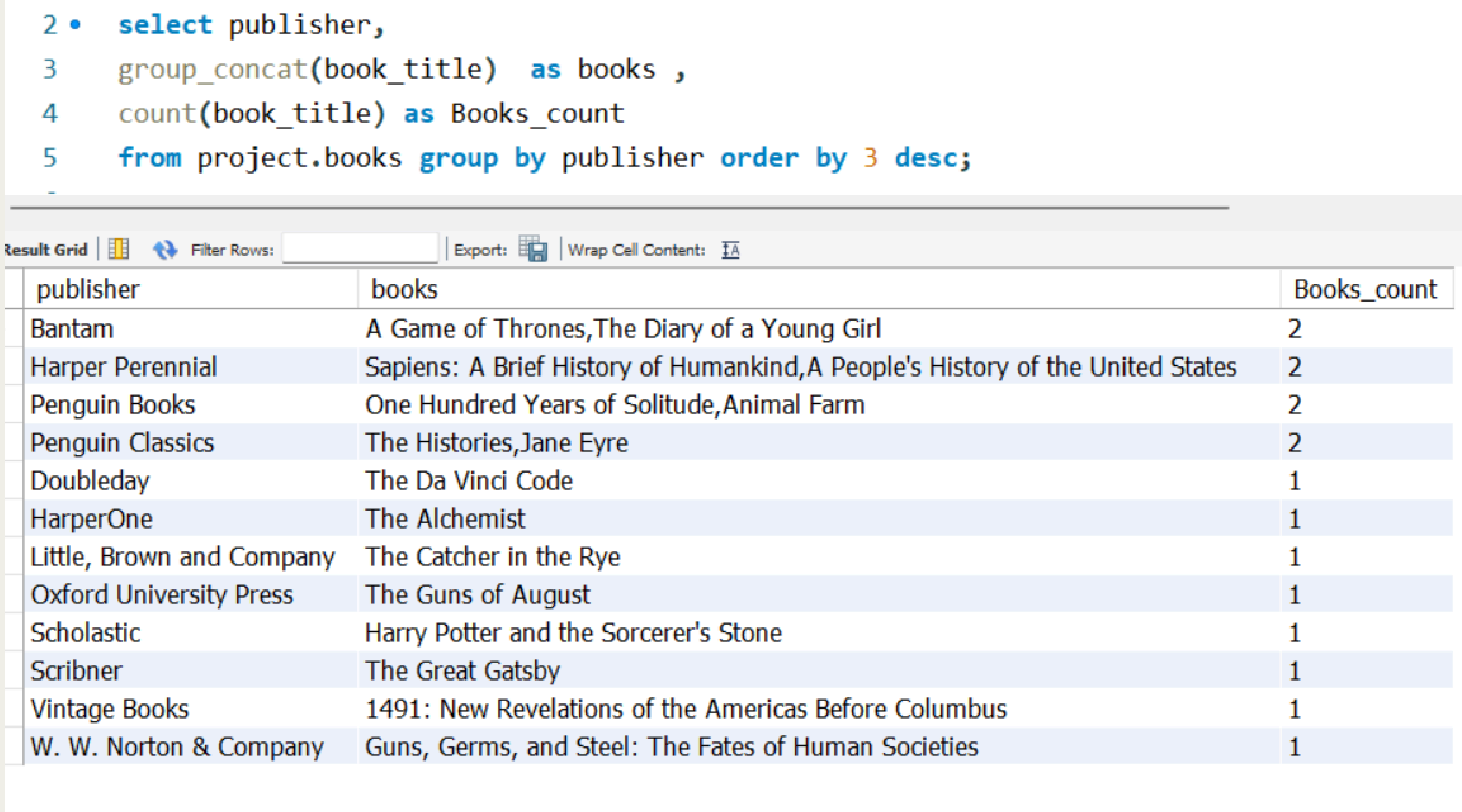
Number of books available in the library



```sql
select count(distinct Book_title) as No_of_Books
from project.books;
```

Avg_price	category
602	Fantasy
587	History
540	Literary Fiction
509	Classic
208	Fiction
664	Mystery

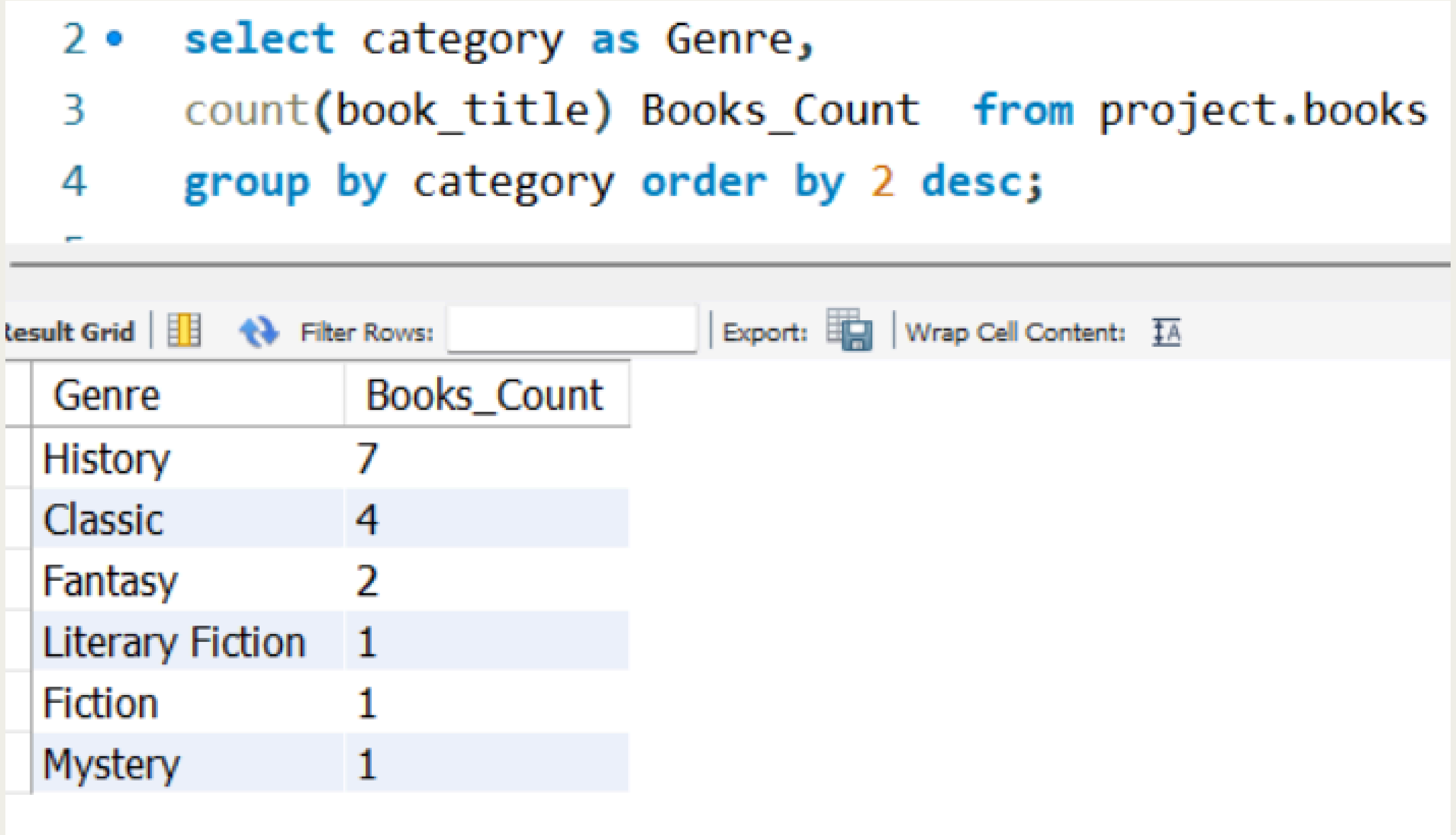
Find out how many books have been published by each publisher



```sql
select publisher,
group_concat(book_title)  as books ,
count(book_title) as Books_count
from project.books group by publisher order by 3 desc;
```

publisher	books	Books_count
Bantam	A Game of Thrones,The Diary of a Young Girl	2
Harper Perennial	Sapiens: A Brief History of Humankind,A People's History of the United States	2
Penguin Books	One Hundred Years of Solitude,Animal Farm	2
Penguin Classics	The Histories,Jane Eyre	2
Doubleday	The Da Vinci Code	1
HarperOne	The Alchemist	1
Little, Brown and Company	The Catcher in the Rye	1
Oxford University Press	The Guns of August	1
Scholastic	Harry Potter and the Sorcerer's Stone	1
Scribner	The Great Gatsby	1
Vintage Books	1491: New Revelations of the Americas Before Columbus	1
W. W. Norton & Company	Guns, Germs, and Steel: The Fates of Human Societies	1

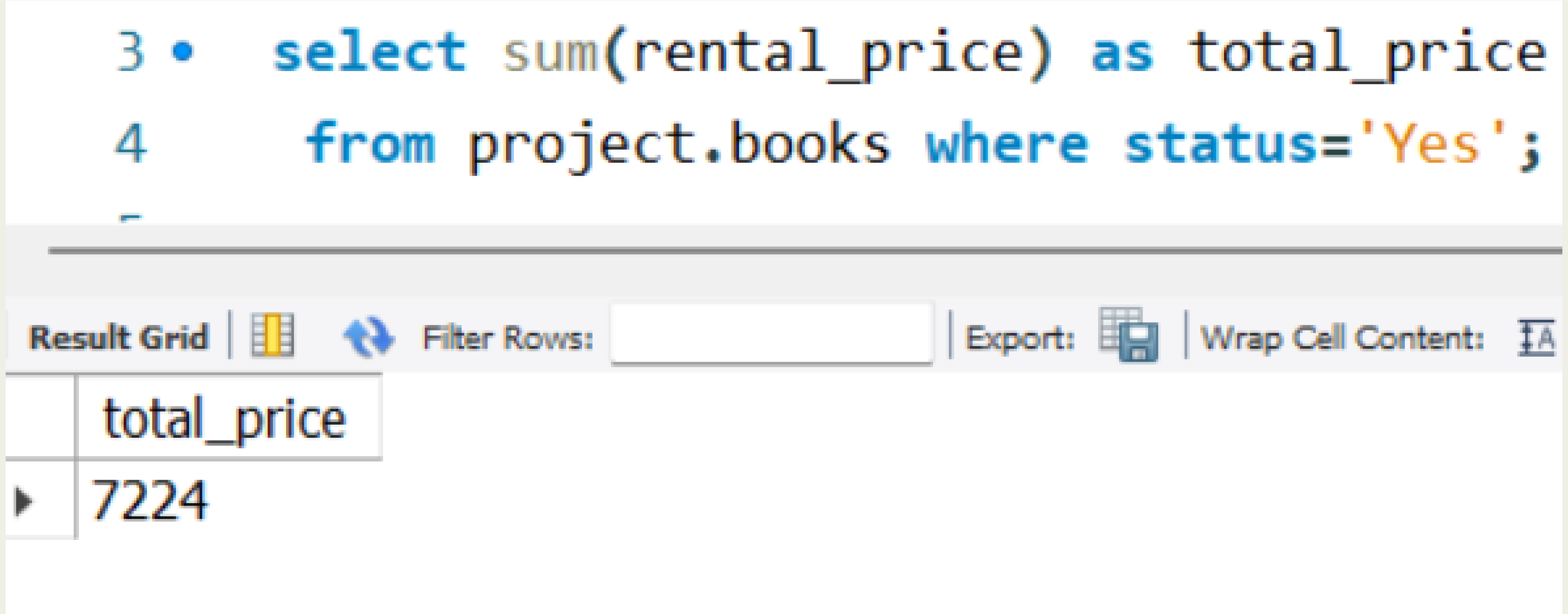
Available books in each genre



```
select category as Genre,
count(book_title) Books_Count  from project.books
group by category order by 2 desc;
```

Genre	Books_Count
History	7
Classic	4
Fantasy	2
Literary Fiction	1
Fiction	1
Mystery	1

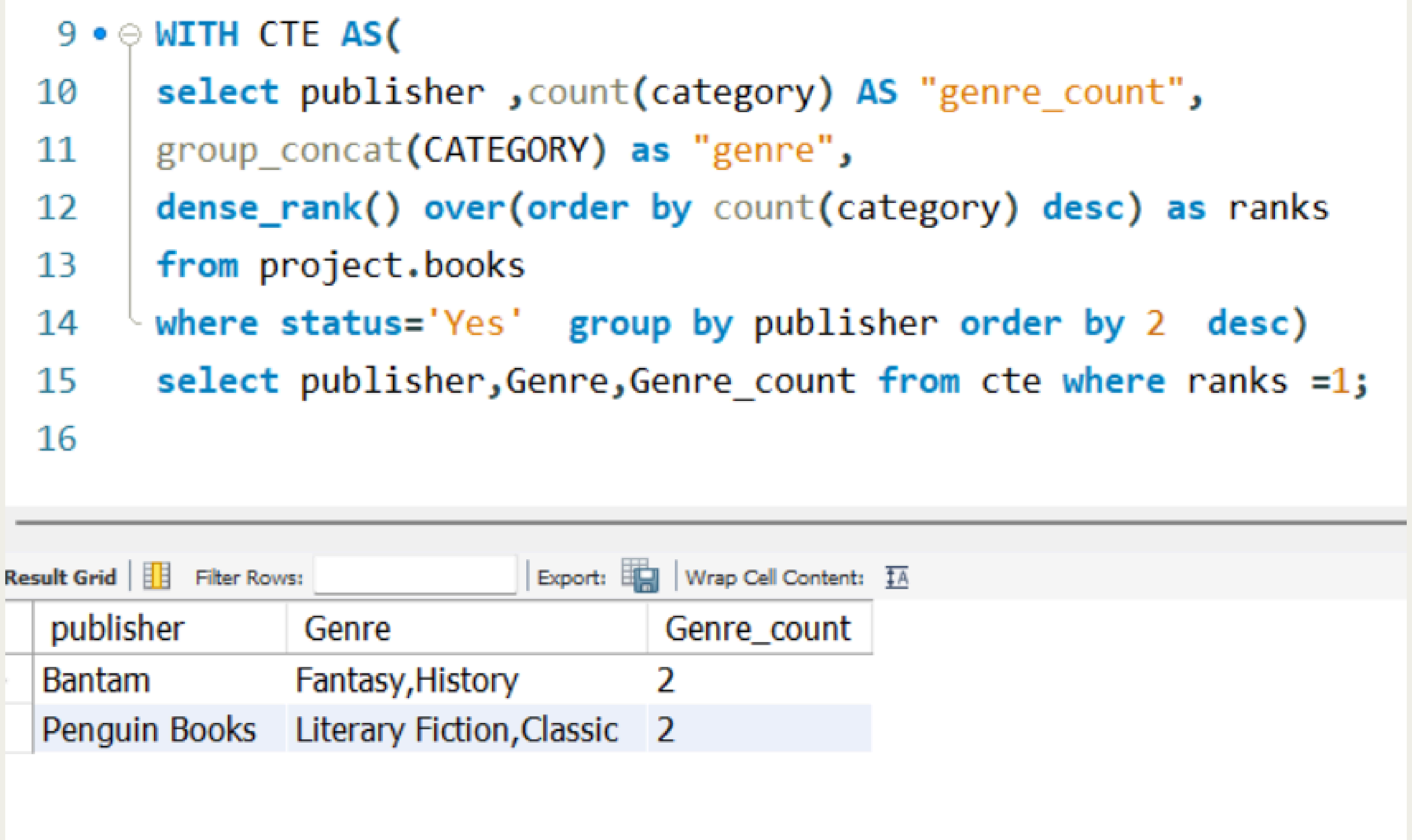
Total value of all books in inventory



```sql
select sum(rental_price) as total_price
 from project.books where status='Yes';
```

```
total_price
7224
```

Find the publishers who have published books available in the library in the most genres and list the genres

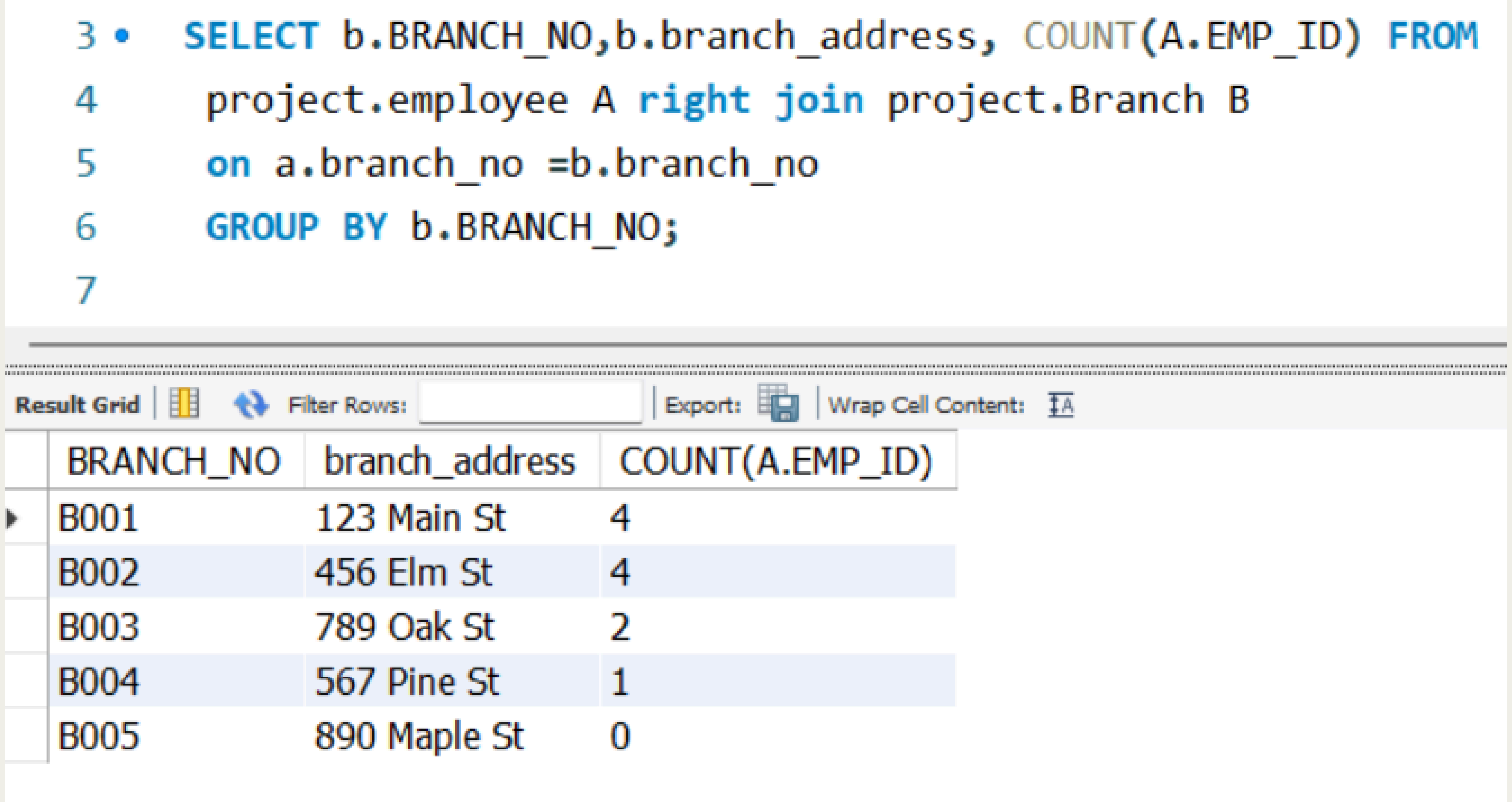


```sql
WITH CTE AS(
select publisher ,count(category) AS "genre_count",
group_concat(CATEGORY) as "genre",
dense_rank() over(order by count(category) desc) as ranks
from project.books
where status='Yes'  group by publisher order by 2  desc)
select publisher,Genre,Genre_count from cte where ranks =1;
```

publisher	Genre	Genre_count
Bantam	Fantasy,History	2
Penguin Books	Literary Fiction,Classic	2

BRANCH/EMPLOYEE MANGEMENT

Total number of employees working in each branch

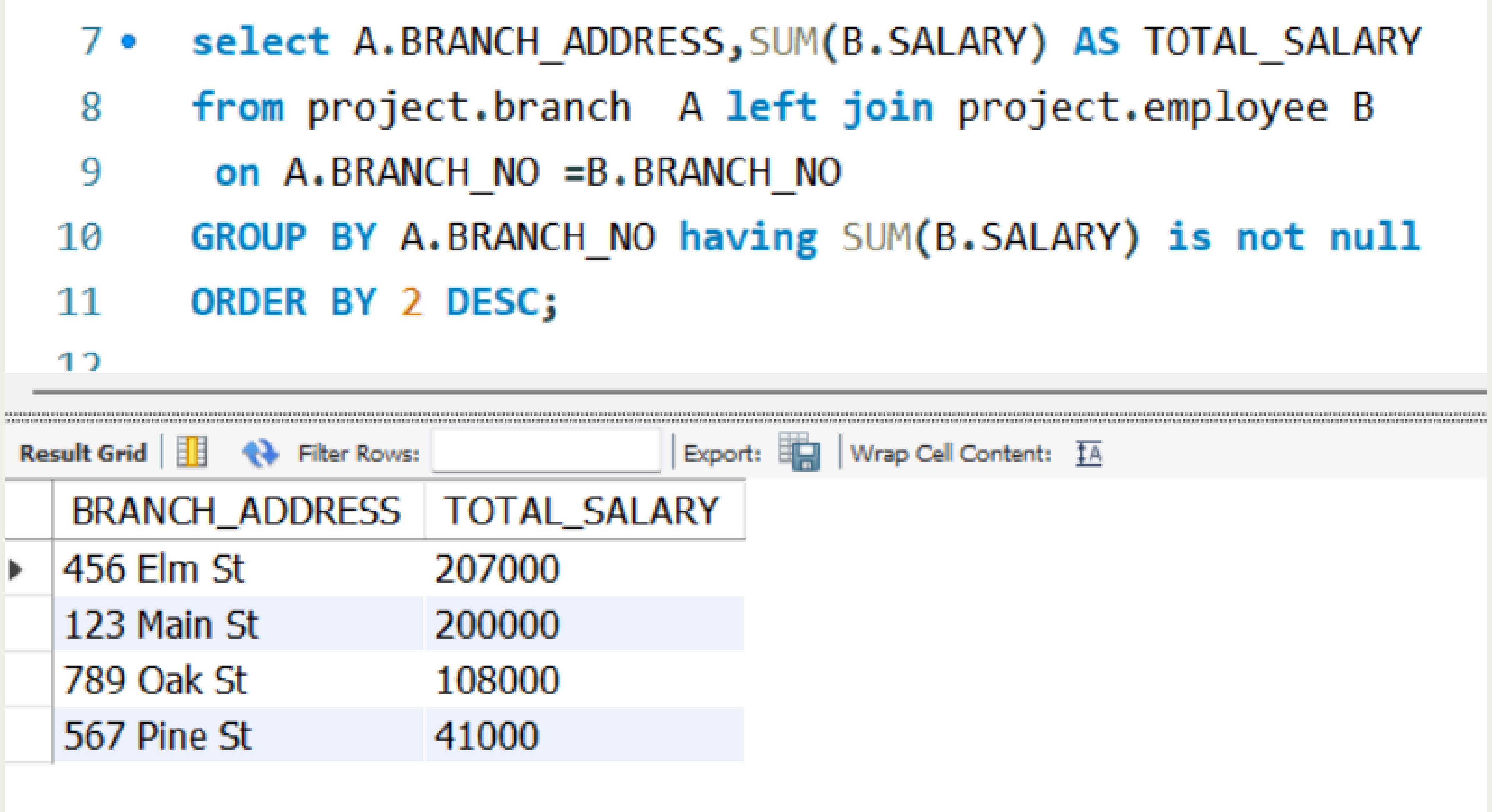


```
SELECT b.BRANCH_NO,b.branch_address, COUNT(A.EMP_ID) FROM
 project.employee A right join project.Branch B
 on a.branch_no =b.branch_no
 GROUP BY b.BRANCH_NO;

```

BRANCH_NO	branch_address	COUNT(A.EMP_ID)
B001	123 Main St	4
B002	456 Elm St	4
B003	789 Oak St	2
B004	567 Pine St	1
B005	890 Maple St	0

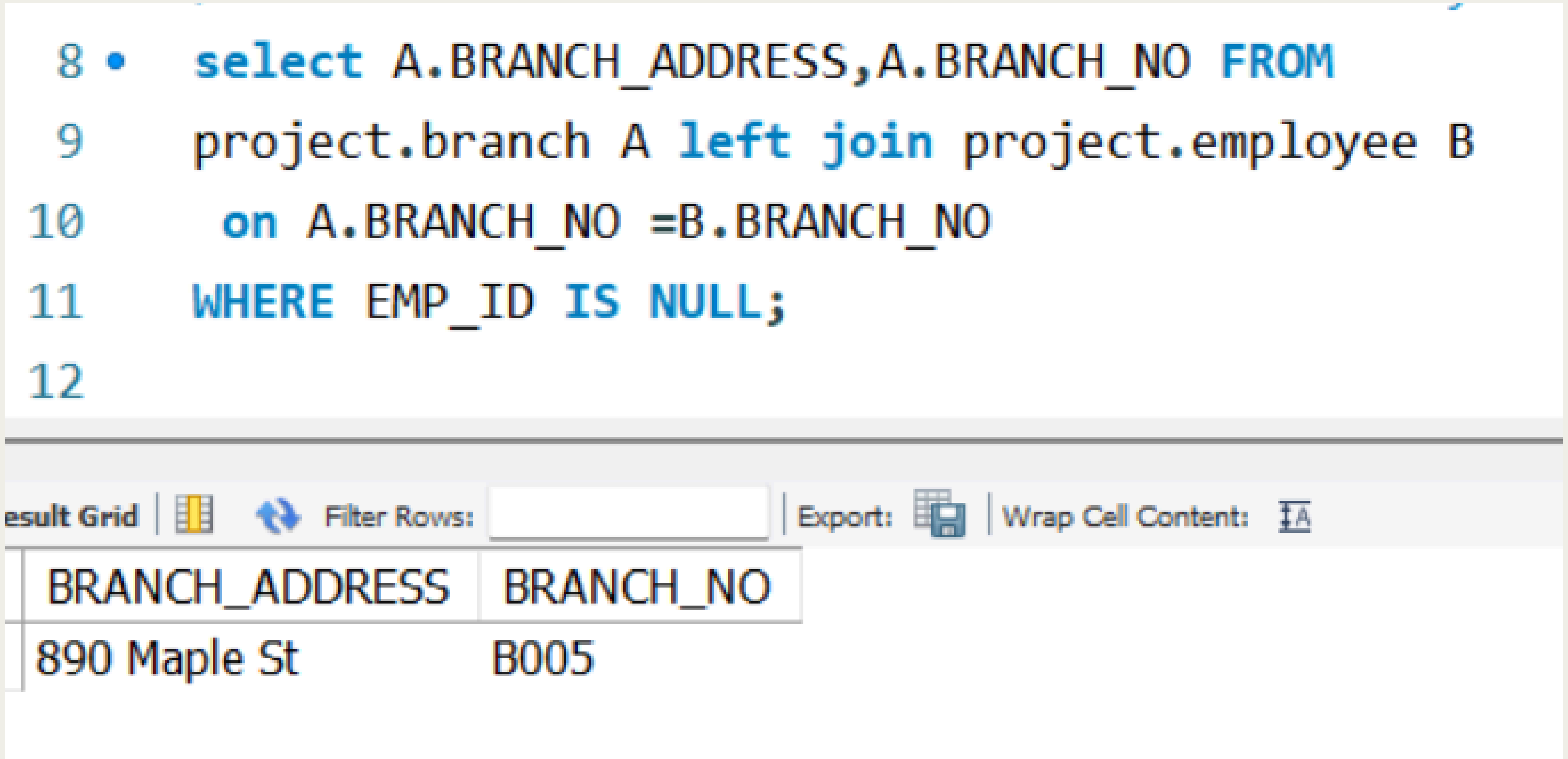
List all branches along with the total salary expenditure for each branch



```sql
select A.BRANCH_ADDRESS,SUM(B.SALARY) AS TOTAL_SALARY
from project.branch  A left join project.employee B
 on A.BRANCH_NO =B.BRANCH_NO
GROUP BY A.BRANCH_NO having SUM(B.SALARY) is not null
ORDER BY 2 DESC;
```

BRANCH_ADDRESS	TOTAL_SALARY
456 Elm St	207000
123 Main St	200000
789 Oak St	108000
567 Pine St	41000

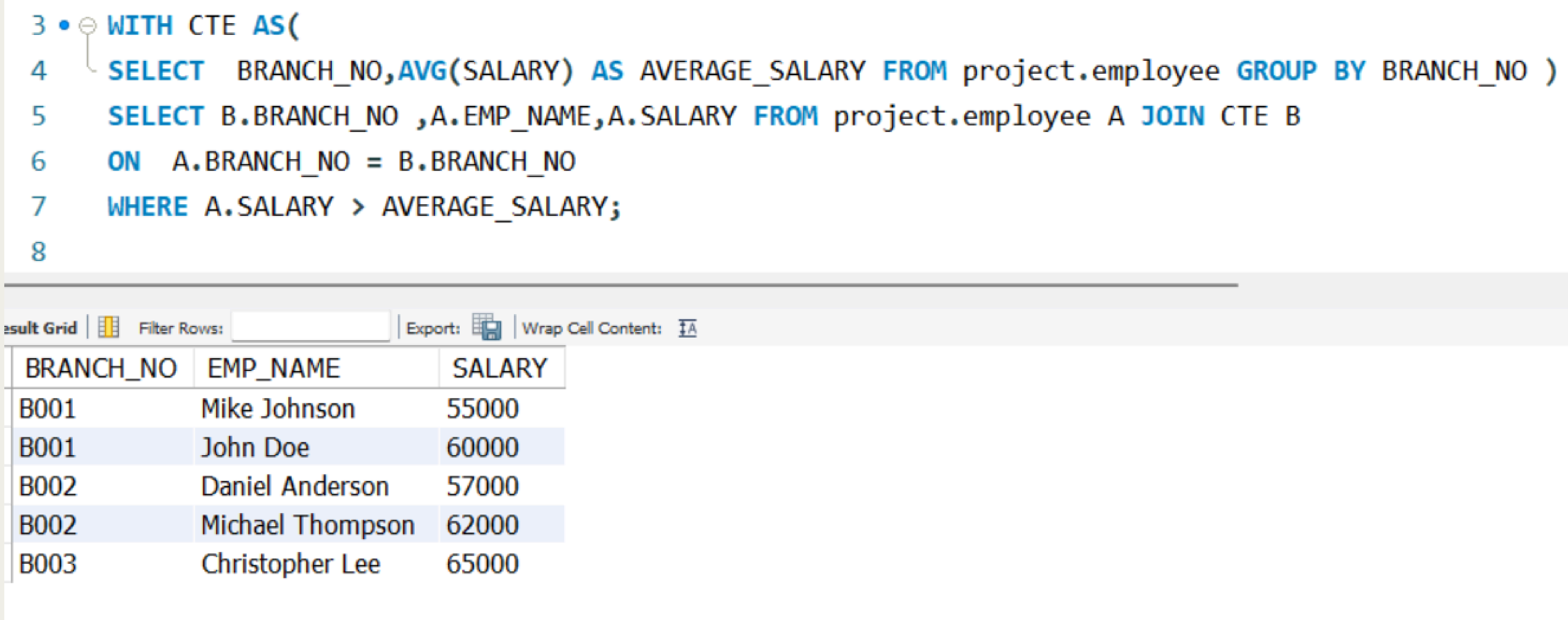
Retrieve the branches that do not have any employees



```sql
select A.BRANCH_ADDRESS,A.BRANCH_NO FROM
project.branch A left join project.employee B
 on A.BRANCH_NO =B.BRANCH_NO
WHERE EMP_ID IS NULL;
```

BRANCH_ADDRESS	BRANCH_NO
890 Maple St	B005

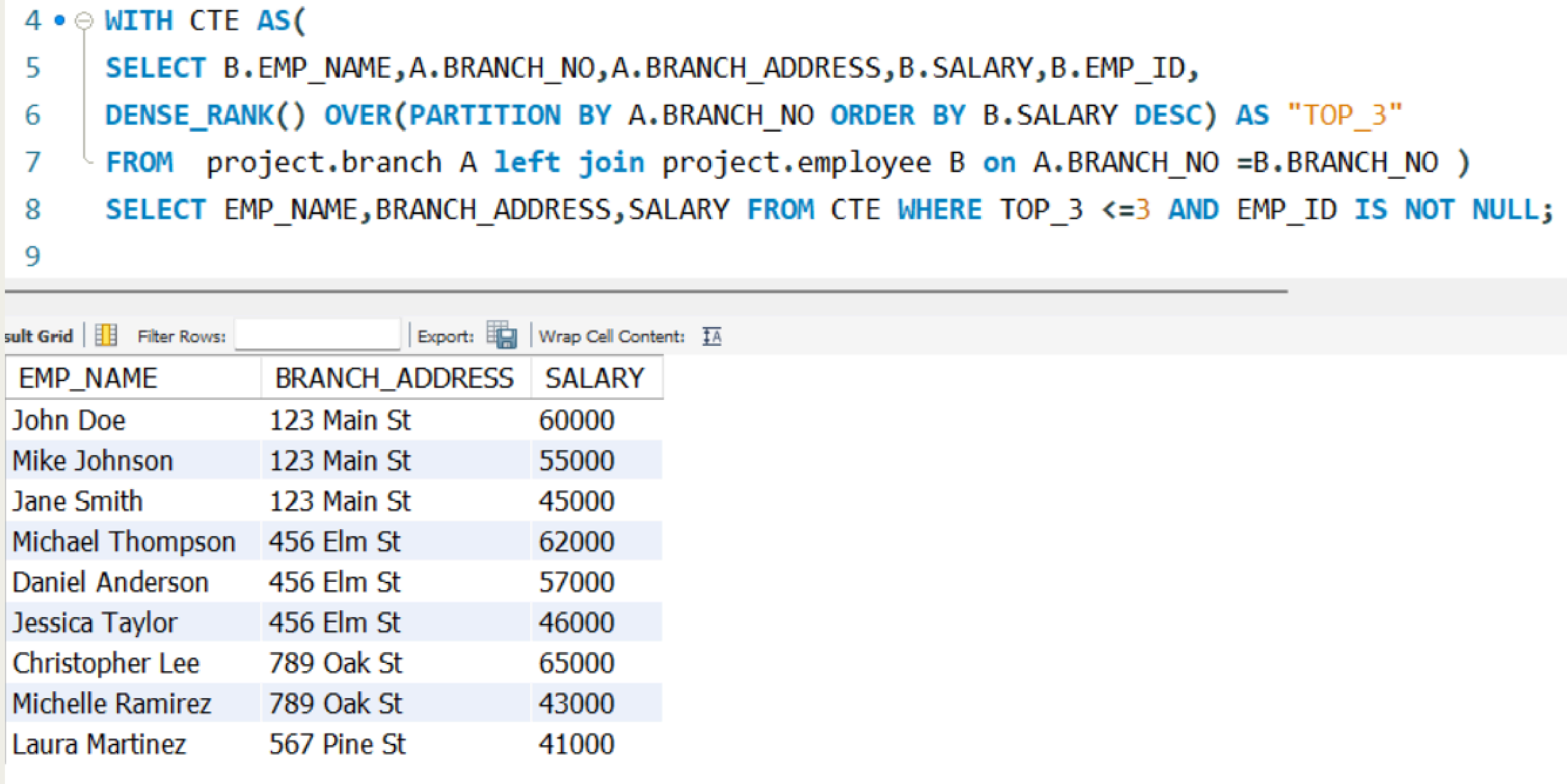
Find the employees who earn more than the average salary of their respective branches



```
WITH CTE AS(
SELECT  BRANCH_NO,AVG(SALARY) AS AVERAGE_SALARY FROM project.employee GROUP BY BRANCH_NO )
SELECT B.BRANCH_NO ,A.EMP_NAME,A.SALARY FROM project.employee A JOIN CTE B
ON  A.BRANCH_NO = B.BRANCH_NO
WHERE A.SALARY > AVERAGE_SALARY;
```

BRANCH_NO	EMP_NAME	SALARY
B001	Mike Johnson	55000
B001	John Doe	60000
B002	Daniel Anderson	57000
B002	Michael Thompson	62000
B003	Christopher Lee	65000

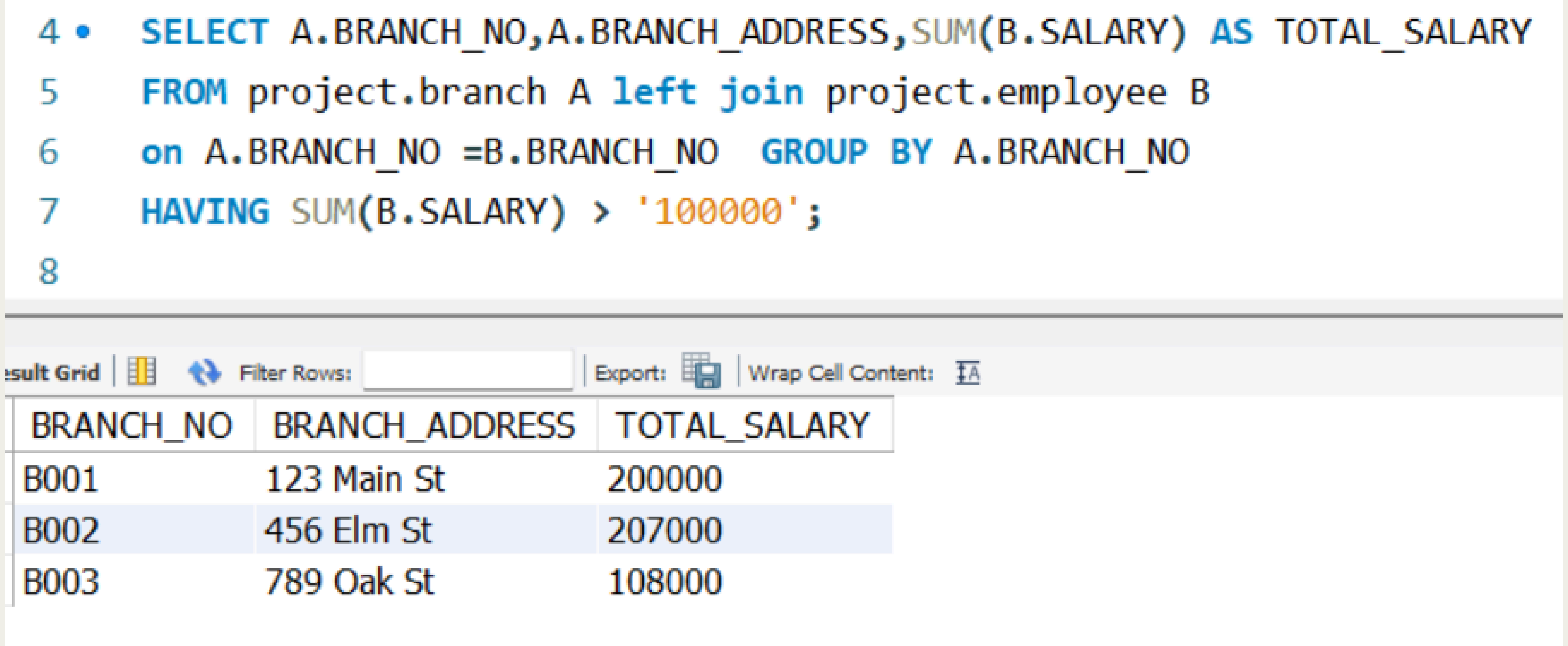
List the top 3 highest-paid employees in each branch



```sql
WITH CTE AS(
SELECT B.EMP_NAME,A.BRANCH_NO,A.BRANCH_ADDRESS,B.SALARY,B.EMP_ID,
DENSE_RANK() OVER(PARTITION BY A.BRANCH_NO ORDER BY B.SALARY DESC) AS "TOP_3"
FROM  project.branch A left join project.employee B on A.BRANCH_NO =B.BRANCH_NO )
SELECT EMP_NAME,BRANCH_ADDRESS,SALARY FROM CTE WHERE TOP_3 <=3 AND EMP_ID IS NOT NULL;
```

EMP_NAME	BRANCH_ADDRESS	SALARY
John Doe	123 Main St	60000
Mike Johnson	123 Main St	55000
Jane Smith	123 Main St	45000
Michael Thompson	456 Elm St	62000
Daniel Anderson	456 Elm St	57000
Jessica Taylor	456 Elm St	46000
Christopher Lee	789 Oak St	65000
Michelle Ramirez	789 Oak St	43000
Laura Martinez	567 Pine St	41000

Find the branches where the total salary expenditure exceeds a 1 Lakh

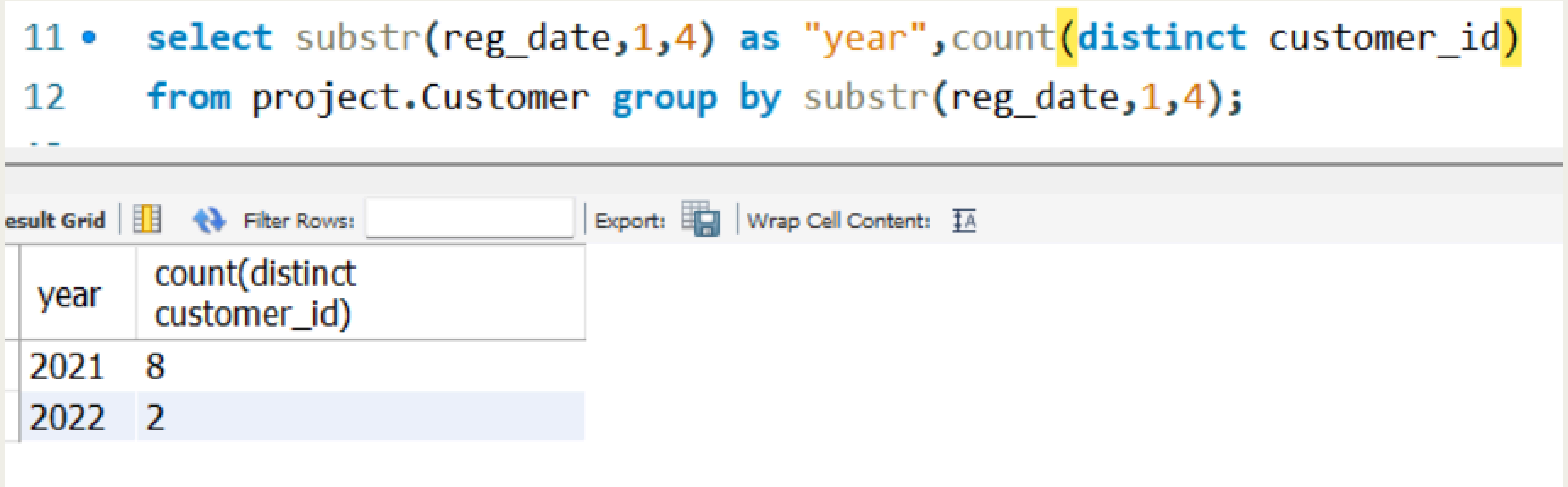


```
SELECT A.BRANCH_NO,A.BRANCH_ADDRESS,SUM(B.SALARY) AS TOTAL_SALARY
FROM project.branch A left join project.employee B
on A.BRANCH_NO =B.BRANCH_NO  GROUP BY A.BRANCH_NO
HAVING SUM(B.SALARY) > '100000';
```

BRANCH_NO	BRANCH_ADDRESS	TOTAL_SALARY
B001	123 Main St	200000
B002	456 Elm St	207000
B003	789 Oak St	108000

BOOK MANGEMENT

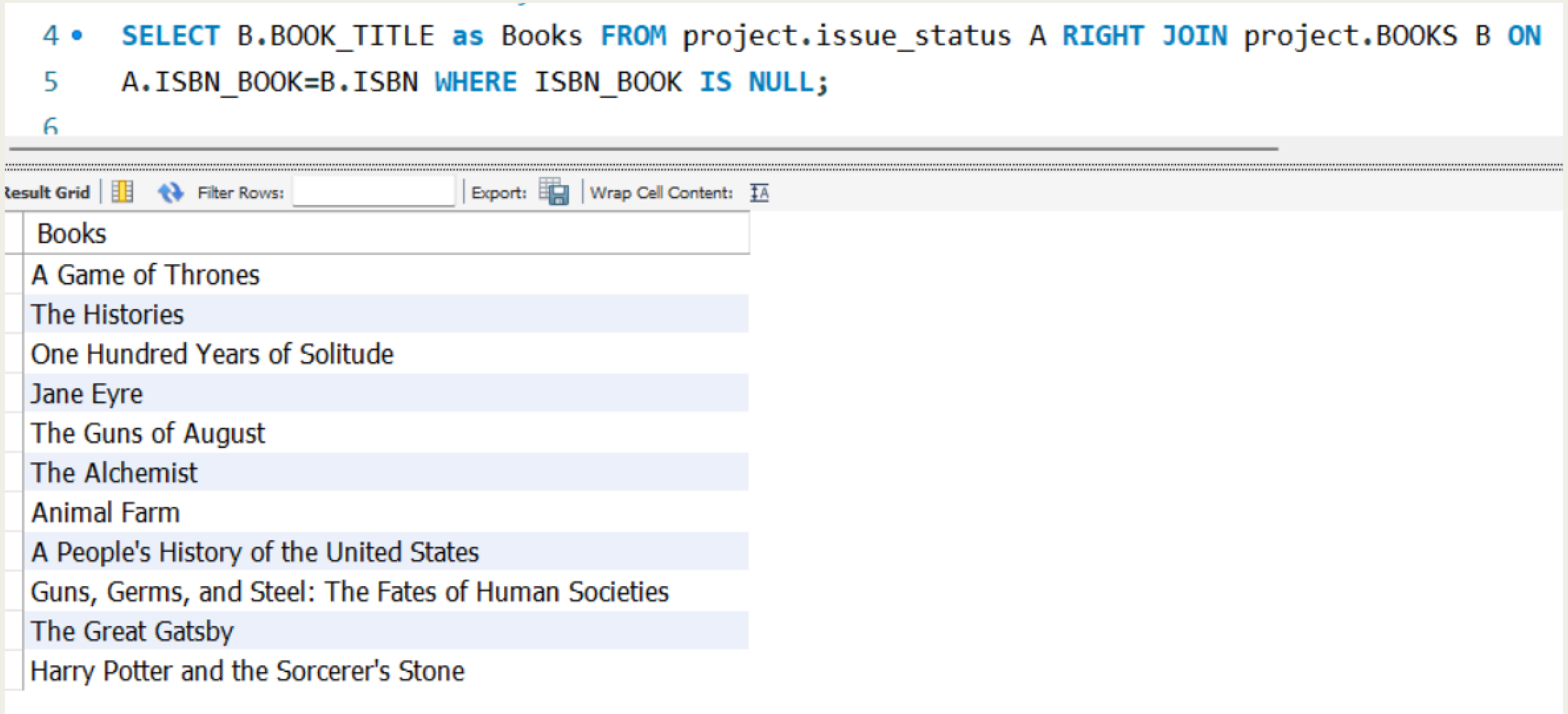
customer count based on year



```
select substr(reg_date,1,4) as "year",count(distinct customer_id)
from project.Customer group by substr(reg_date,1,4);
```

year	count(distinct customer_id)
2021	8
2022	2

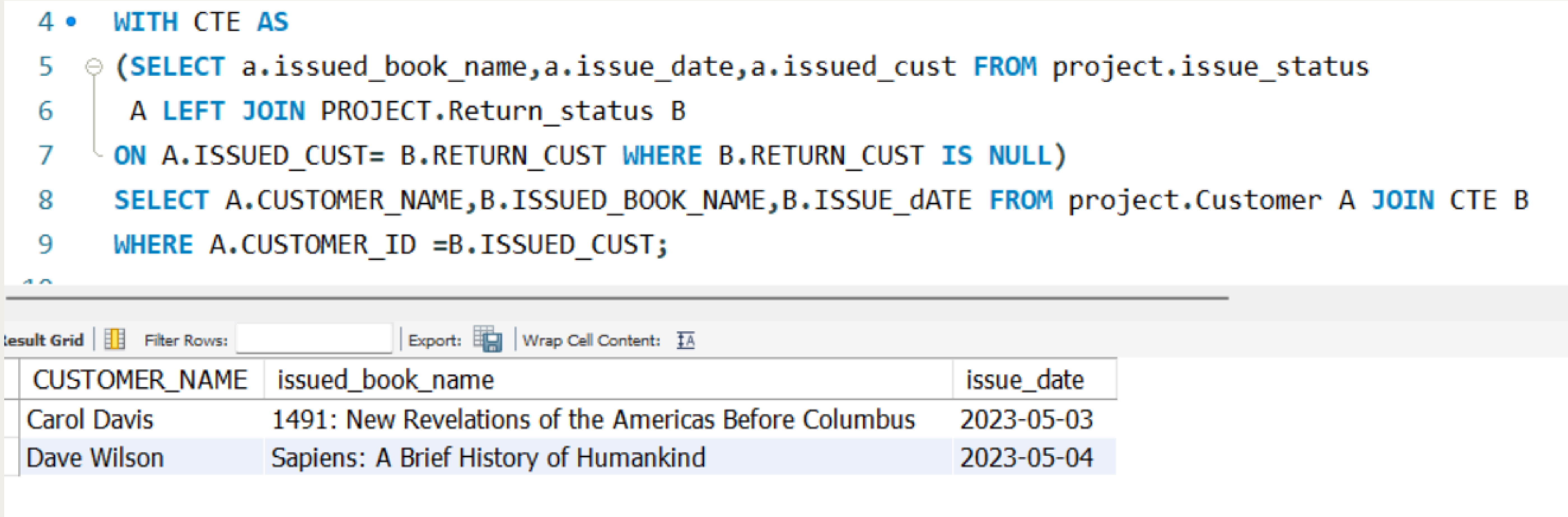
Books that have not been borrowed by customers



```sql
SELECT B.BOOK_TITLE as Books FROM project.issue_status A RIGHT JOIN project.BOOKS B ON
A.ISBN_BOOK=B.ISBN WHERE ISBN_BOOK IS NULL;
```

Books
A Game of Thrones
The Histories
One Hundred Years of Solitude
Jane Eyre
The Guns of August
The Alchemist
Animal Farm
A People's History of the United States
Guns, Germs, and Steel: The Fates of Human Societies
The Great Gatsby
Harry Potter and the Sorcerer's Stone

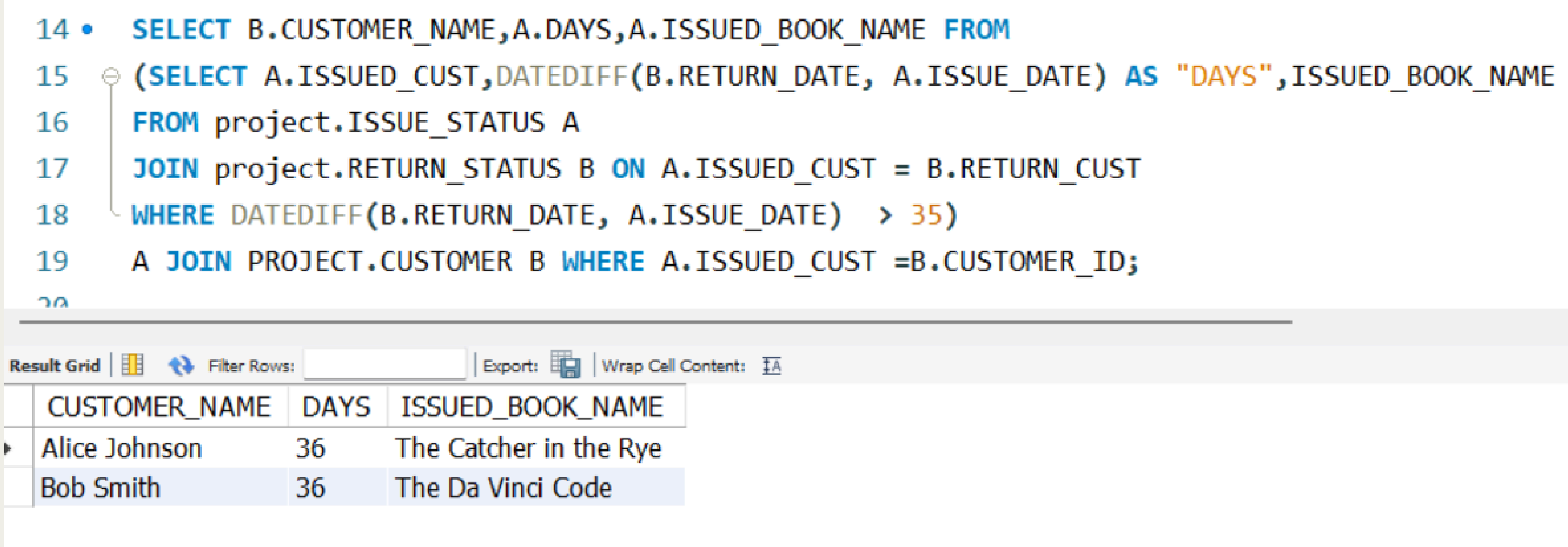
List all books that have been issued but never returned



```sql
WITH CTE AS
(SELECT a.issued_book_name,a.issue_date,a.issued_cust FROM project.issue_status
 A LEFT JOIN PROJECT.Return_status B
ON A.ISSUED_CUST= B.RETURN_CUST WHERE B.RETURN_CUST IS NULL)
SELECT A.CUSTOMER_NAME,B.ISSUED_BOOK_NAME,B.ISSUE_dATE FROM project.Customer A JOIN CTE B
WHERE A.CUSTOMER_ID =B.ISSUED_CUST;
```

CUSTOMER_NAME	issued_book_name	issue_date
Carol Davis	1491: New Revelations of the Americas Before Columbus	2023-05-03
Dave Wilson	Sapiens: A Brief History of Humankind	2023-05-04

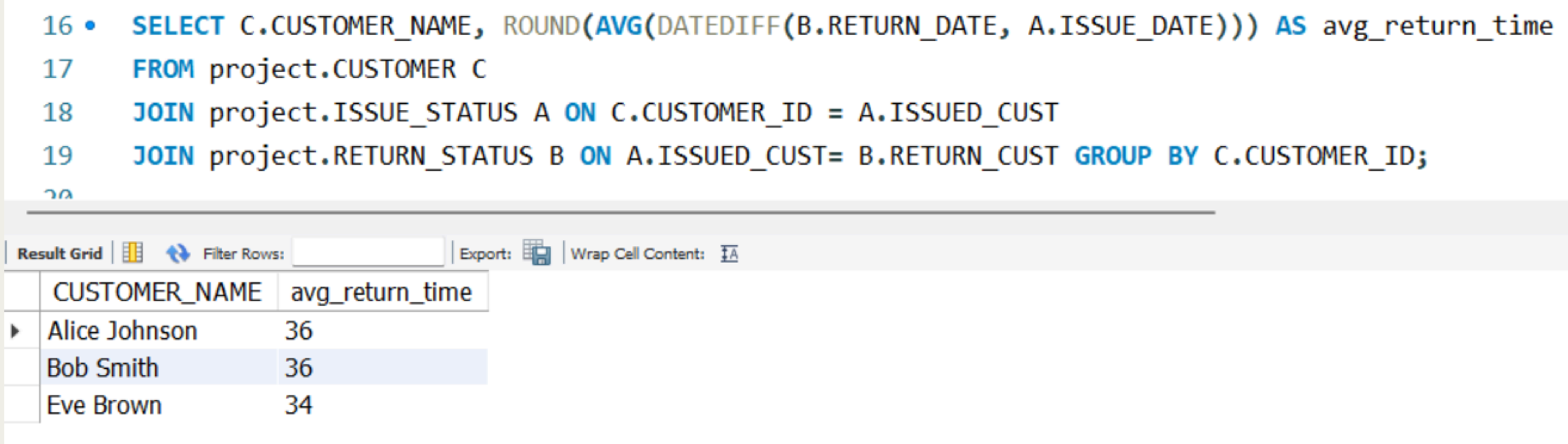
Customers who borrowed books for more than 35 days



```sql
SELECT B.CUSTOMER_NAME,A.DAYS,A.ISSUED_BOOK_NAME FROM
(SELECT A.ISSUED_CUST,DATEDIFF(B.RETURN_DATE, A.ISSUE_DATE) AS "DAYS",ISSUED_BOOK_NAME
FROM project.ISSUE_STATUS A
JOIN project.RETURN_STATUS B ON A.ISSUED_CUST = B.RETURN_CUST
WHERE DATEDIFF(B.RETURN_DATE, A.ISSUE_DATE)  > 35)
A JOIN PROJECT.CUSTOMER B WHERE A.ISSUED_CUST =B.CUSTOMER_ID;
```

CUSTOMER_NAME	DAYS	ISSUED_BOOK_NAME
Alice Johnson	36	The Catcher in the Rye
Bob Smith	36	The Da Vinci Code

Calculate the average time taken by each customer to return books



```sql
SELECT C.CUSTOMER_NAME, ROUND(AVG(DATEDIFF(B.RETURN_DATE, A.ISSUE_DATE))) AS avg_return_time
FROM project.CUSTOMER C
JOIN project.ISSUE_STATUS A ON C.CUSTOMER_ID = A.ISSUED_CUST
JOIN project.RETURN_STATUS B ON A.ISSUED_CUST= B.RETURN_CUST GROUP BY C.CUSTOMER_ID;
```

CUSTOMER_NAME	avg_return_time
Alice Johnson	36
Bob Smith	36
Eve Brown	34

LIBRARAY OVERVIEW

16
Total Collections

13
Availble Books

10
Customers

11
employees

5
No of Branches

600 INR
Max avergae book cost

Thank you!

CREATED BY
VISHNUPRIYA KATHIRVEL