Name: Eric Niyikiza

ID: 24666

Course: PL/SQL group: D

SQL Library System Project Documentation

1. Overview

Problem Definition & Success Criteria

Problem Definition

Managing library data often requires not only storing information about users, books, and borrow records, but also analyzing it to answer deeper questions:

- Who are the top borrowers?
- How do borrowing trends change over time?
- How can we compare borrowing activity across different users or periods?
- How can we segment users into meaningful groups for reporting or decision-making?

Traditional SQL queries (like `SELECT` with joins) provide basic answers, but they lack advanced analytical capabilities. To solve this, we implemented **SQL window functions** (Ranking, Aggregate, Navigation, Distribution) on top of a relational schema (Users, BookTable, Borrowers).

Success Criteria

The project is considered successful if the following outcomes are achieved:

1. **Database Schema Setup**
    - Tables (`Users`, `BookTable`, `Borrowers`) created with proper primary and foreign keys.
    - Sample data inserted (15 users, 15 books, 10+ borrow records).
2. **Joins Implemented**
    - Queries using `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, and `FULL OUTER JOIN` correctly combine data across tables.
    - Results clearly show relationships between users, books, and borrow records.
3. **Window Functions Demonstrated**
    - **Ranking Functions** (`ROW_NUMBER`, `RANK`, `DENSE_RANK`, `PERCENT_RANK`) used to identify top borrowers.
    - **Aggregate Functions** (`SUM`, `AVG`, `MIN`, `MAX` with ROWS and RANGE) used to calculate running totals and trends.
    - **Navigation Functions** (`LAG`, `LEAD`) used to compare borrowing activity across time.

   - **Distribution Functions** (`NTILE(4)`, `CUME_DIST`) used to segment users into quartiles and cumulative distribution.

4. **Interpretation Provided**

- Each query includes a short explanation of what it does and why it’s useful.
- Results demonstrate practical insights (e.g., top N borrowers, borrowing trends, customer segmentation).

This project demonstrates SQL fundamentals and advanced analytics using a **library management system**.
It includes:

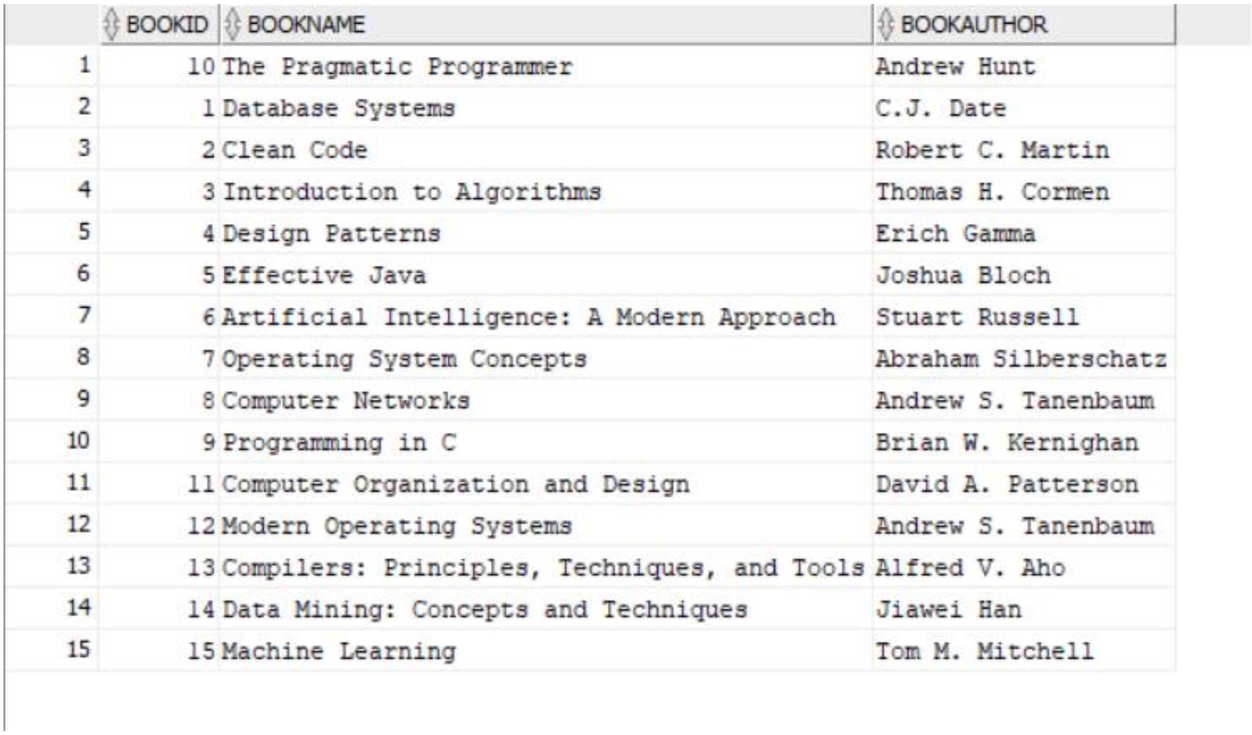
1. **Table creation** (Users, BookTable, Borrowers)

   ```
   CREATE TABLE BookTable (
      BookID INT PRIMARY KEY,
      BookName VARCHAR2(100) NOT NULL,
      BookAuthor VARCHAR2(100)
   );
   INSERT INTO BookTable (BookID, BookName, BookAuthor)
   VALUES (1, 'Database Systems', 'C.J. Date');

   INSERT INTO BookTable (BookID, BookName, BookAuthor)
   VALUES (2, 'Clean Code', 'Robert C. Martin');

   INSERT INTO BookTable (BookID, BookName, BookAuthor)
   VALUES (3, 'Introduction to Algorithms', 'Thomas H. Cormen');

   INSERT INTO BookTable (BookID, BookName, BookAuthor)
   ```

```
VALUES (4, 'Design Patterns', 'Erich Gamma');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (5, 'Effective Java', 'Joshua Bloch');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (6, 'Artificial Intelligence: A Modern Approach', 'Stuart Russell');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (7, 'Operating System Concepts', 'Abraham Silberschatz');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (8, 'Computer Networks', 'Andrew S. Tanenbaum');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (9, 'Programming in C', 'Brian W. Kernighan');

INSERT INTO BookTable (BookID, BookName, BookAuthor)
VALUES (10, 'The Pragmatic Programmer', 'Andrew Hunt');


-- Insert 5 more books
INSERT INTO BookTable (BookID, BookName, BookAuthor)
```

```
VALUES (11, 'Computer Organization and Design', 'David A. Patterson');


INSERT INTO BookTable (BookID, BookName, BookAuthor)

VALUES (12, 'Modern Operating Systems', 'Andrew S. Tanenbaum');


INSERT INTO BookTable (BookID, BookName, BookAuthor)

VALUES (13, 'Compilers: Principles, Techniques, and Tools', 'Alfred V. Aho');


INSERT INTO BookTable (BookID, BookName, BookAuthor)

VALUES (14, 'Data Mining: Concepts and Techniques', 'Jiawei Han');


INSERT INTO BookTable (BookID, BookName, BookAuthor)

VALUES (15, 'Machine Learning', 'Tom M. Mitchell');
```

2. **Data insertion** (sample records)



	BOOKID	BOOKNAME	BOOKAUTHOR
1	10	The Pragmatic Programmer	Andrew Hunt
2	1	Database Systems	C.J. Date
3	2	Clean Code	Robert C. Martin
4	3	Introduction to Algorithms	Thomas H. Cormen
5	4	Design Patterns	Erich Gamma
6	5	Effective Java	Joshua Bloch
7	6	Artificial Intelligence: A Modern Approach	Stuart Russell
8	7	Operating System Concepts	Abraham Silberschatz
9	8	Computer Networks	Andrew S. Tanenbaum
10	9	Programming in C	Brian W. Kernighan
11	11	Computer Organization and Design	David A. Patterson
12	12	Modern Operating Systems	Andrew S. Tanenbaum
13	13	Compilers: Principles, Techniques, and Tools	Alfred V. Aho
14	14	Data Mining: Concepts and Techniques	Jiawei Han
15	15	Machine Learning	Tom M. Mitchell

3. **Joins** (combining tables)

INNER JOIN

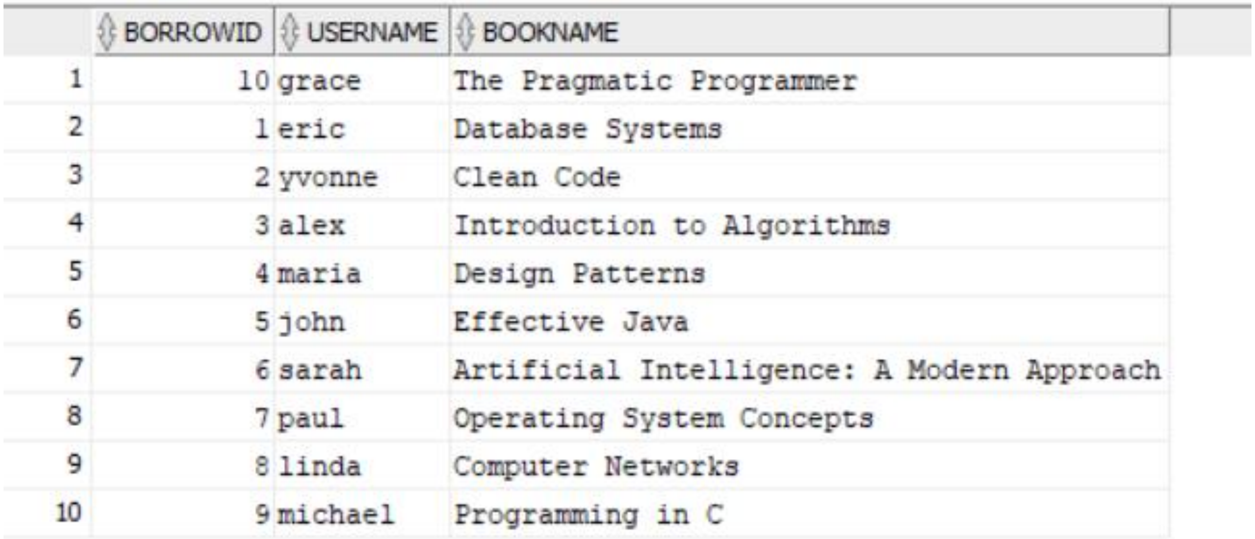
SELECT b.BorrowID, u.Username, bk.BookName

FROM Borrowers b

INNER JOIN Users u ON b.UserID = u.UserID

INNER JOIN BookTable bk ON b.BookID = bk.BookID;

Results



	BORROWID	USERNAME	BOOKNAME
1	10	grace	The Pragmatic Programmer
2	1	eric	Database Systems
3	2	yvonne	Clean Code
4	3	alex	Introduction to Algorithms
5	4	maria	Design Patterns
6	5	john	Effective Java
7	6	sarah	Artificial Intelligence: A Modern Approach
8	7	paul	Operating System Concepts
9	8	linda	Computer Networks
10	9	michael	Programming in C

LEFT JOIN

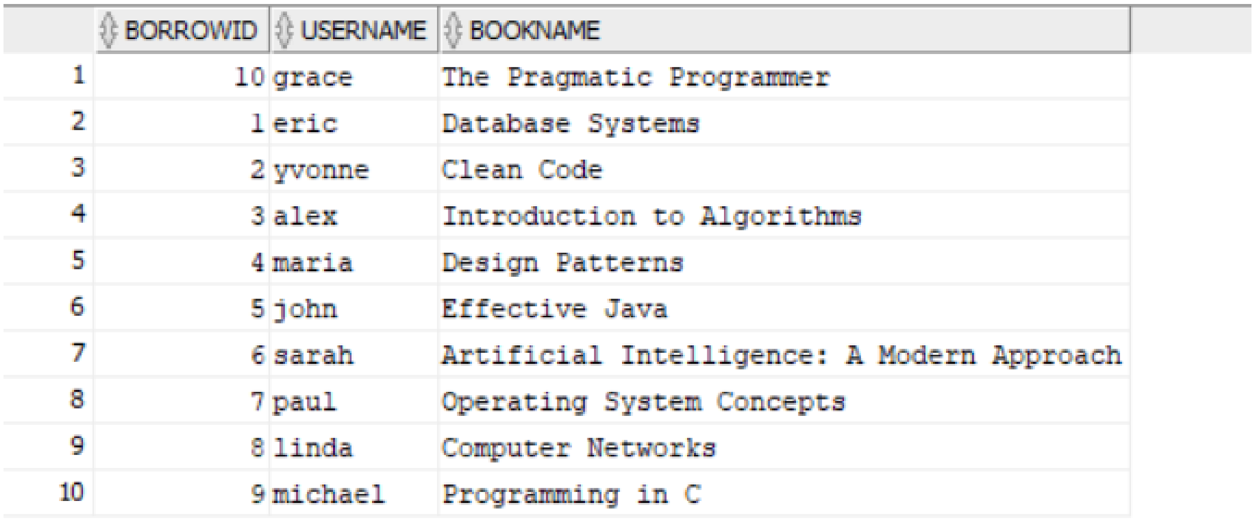
SELECT b.BorrowID, u.Username, bk.BookName

FROM Borrowers b

LEFT JOIN Users u ON b.UserID = u.UserID

LEFT JOIN BookTable bk ON b.BookID = bk.BookID;

Results



	BORROWID	USERNAME	BOOKNAME
1	10	grace	The Pragmatic Programmer
2	1	eric	Database Systems
3	2	yvonne	Clean Code
4	3	alex	Introduction to Algorithms
5	4	maria	Design Patterns
6	5	john	Effective Java
7	6	sarah	Artificial Intelligence: A Modern Approach
8	7	paul	Operating System Concepts
9	8	linda	Computer Networks
10	9	michael	Programming in C

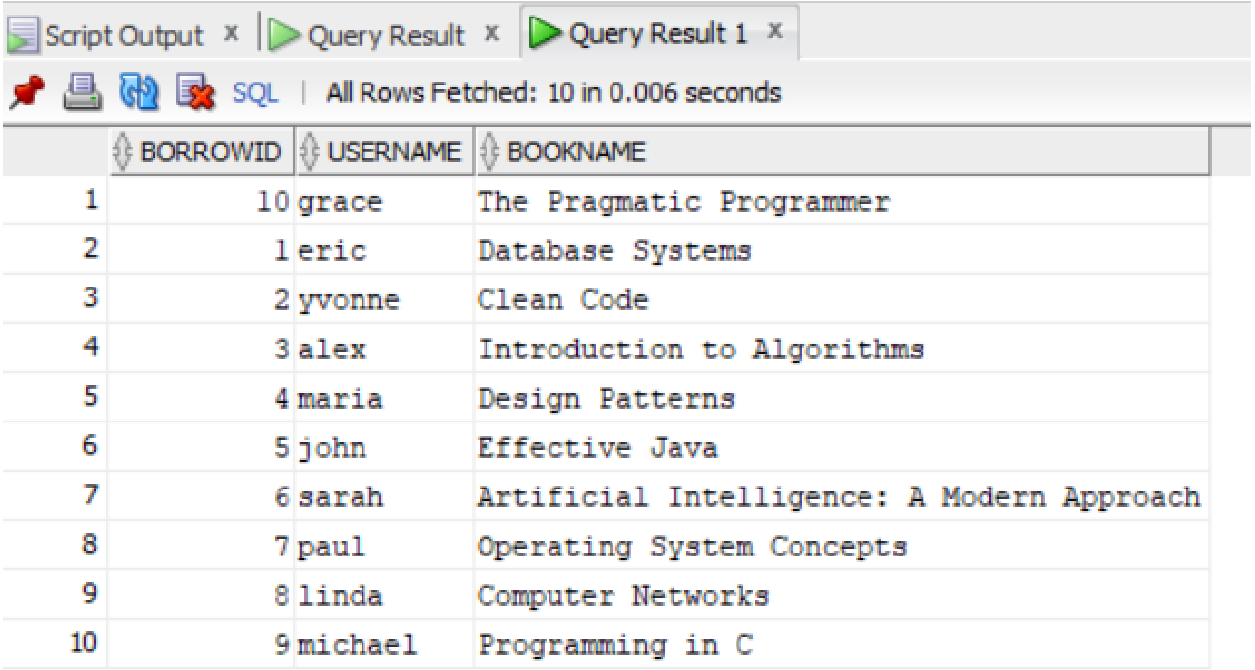
RIGHT JOIN

SELECT b.BorrowID, u.Username, bk.BookName

FROM Borrowers b

RIGHT JOIN Users u ON b.UserID = u.UserID

JOIN BookTable bk ON b.BookID = bk.BookID;



All Rows Fetched: 10 in 0.006 seconds

	BORROWID	USERNAME	BOOKNAME
1	10	grace	The Pragmatic Programmer
2	1	eric	Database Systems
3	2	yvonne	Clean Code
4	3	alex	Introduction to Algorithms
5	4	maria	Design Patterns
6	5	john	Effective Java
7	6	sarah	Artificial Intelligence: A Modern Approach
8	7	paul	Operating System Concepts
9	8	linda	Computer Networks
10	9	michael	Programming in C

FULL OUTER JOIN

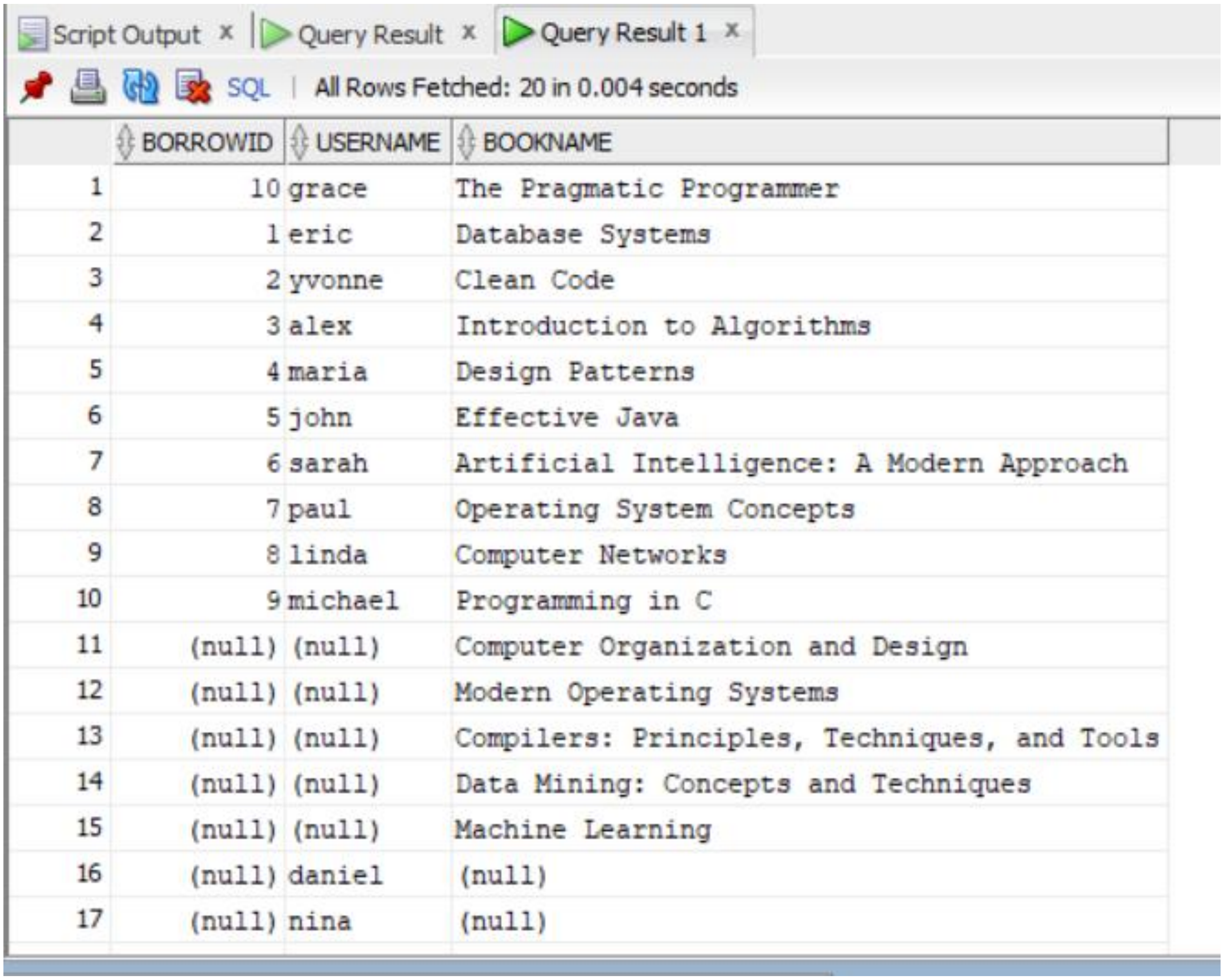
SELECT b.BorrowID, u.Username, bk.BookName

FROM Borrowers b

FULL OUTER JOIN Users u ON b.UserID = u.UserID

FULL OUTER JOIN BookTable bk ON b.BookID = bk.BookID;

Result



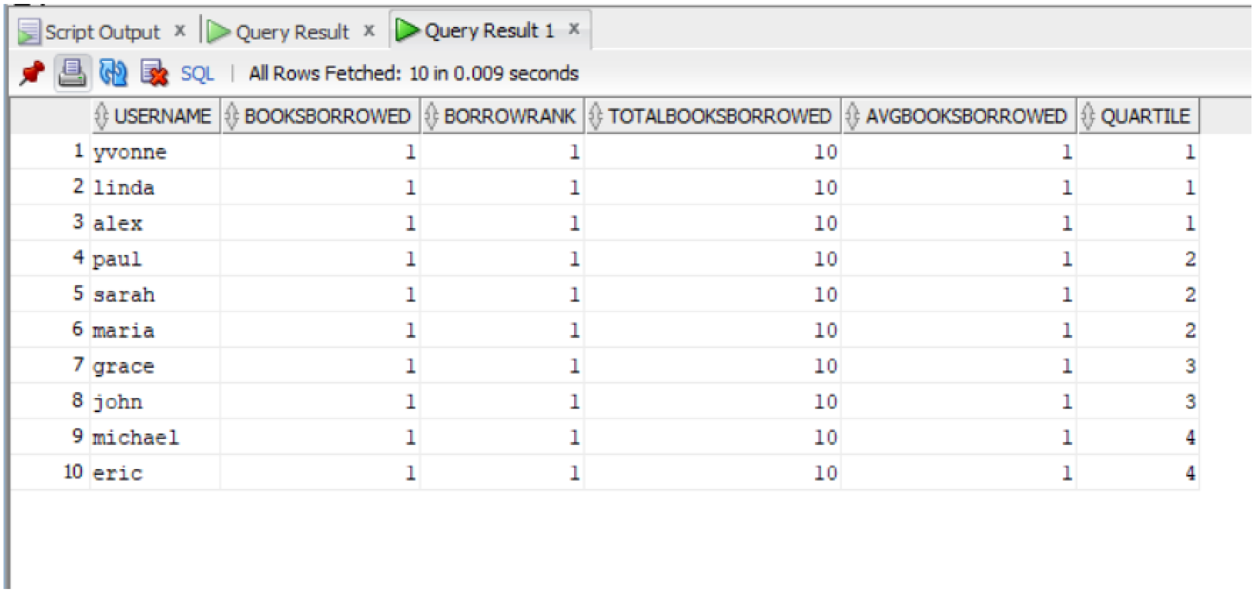
All Rows Fetched: 20 in 0.004 seconds

	BORROWID	USERNAME	BOOKNAME
1	10	grace	The Pragmatic Programmer
2	1	eric	Database Systems
3	2	yvonne	Clean Code
4	3	alex	Introduction to Algorithms
5	4	maria	Design Patterns
6	5	john	Effective Java
7	6	sarah	Artificial Intelligence: A Modern Approach
8	7	paul	Operating System Concepts
9	8	linda	Computer Networks
10	9	michael	Programming in C
11	(null)	(null)	Computer Organization and Design
12	(null)	(null)	Modern Operating Systems
13	(null)	(null)	Compilers: Principles, Techniques, and Tools
14	(null)	(null)	Data Mining: Concepts and Techniques
15	(null)	(null)	Machine Learning
16	(null)	daniel	(null)
17	(null)	nina	(null)

4. **Window functions** (Ranking, Aggregate, Navigation, Distribution)

```
SELECT

   u.Username,

   COUNT(b.BorrowID) AS BooksBorrowed,



   -- Rank users by number of books borrowed

   RANK() OVER (ORDER BY COUNT(b.BorrowID) DESC) AS BorrowRank,
    -- Running total of books borrowed across all users
    SUM(COUNT(b.BorrowID)) OVER () AS TotalBooksBorrowed,

    -- Average number of books borrowed across all users
    AVG(COUNT(b.BorrowID)) OVER () AS AvgBooksBorrowed,

    -- Divide users into 4 quartiles based on borrow count
    NTILE(4) OVER (ORDER BY COUNT(b.BorrowID) DESC) AS Quartile
FROM Borrowers b
JOIN Users u ON b.UserID = u.UserID
GROUP BY u.Username
ORDER BY BorrowRank;
```

Result



All Rows Fetched: 10 in 0.009 seconds

	USERNAME	BOOKSBORROWED	BORROWRANK	TOTALBOOKSBORROWED	AVGBOOKSBORROWED	QUARTILE
1	yvonne	1	1	10	1	1
2	linda	1	1	10	1	1
3	alex	1	1	10	1	1
4	paul	1	1	10	1	2
5	sarah	1	1	10	1	2
6	maria	1	1	10	1	2
7	grace	1	1	10	1	3
8	john	1	1	10	1	3
9	michael	1	1	10	1	4
10	eric	1	1	10	1	4