# Automated Solutions To Library Operations

An  Advanced Analytics in SQL

By Loretta Chimezie

Data Analyst

# Project Summary

This project analyzes operational datasets and introduces automated solutions to optimize performance, improve service delivery, and reduce operational costs across library processes. The solution improves process accuracy, reduces processing time and enhances organizational effectiveness for strategic decision-making.

# Objectives

To create SQL queries or procedures that addresses each of the following issues:

1. Identify the scope of overdue book problem – how many members with unreturned books and how much losses
2. Instantly updates books status
3. Shows performance metrics across all branches
4. Identify active and inactive members
5. To track staff productivity besides manager’s observations
6. Identifies members with damaged book(s) history in order to reduce annual replacement cost

# Datasets Description

- Branch.csv
- Books.csv
- Employees.csv
- Members.csv
- Issued_status.csv
- Return_status.csv
- Key fields — member_id, reg_date, branch_id, emp_id, issued_date, return_date, book_name, book_quality, book_title, category, status

# Data Cleaning

- Converted string dates using STR_TO_DATE()

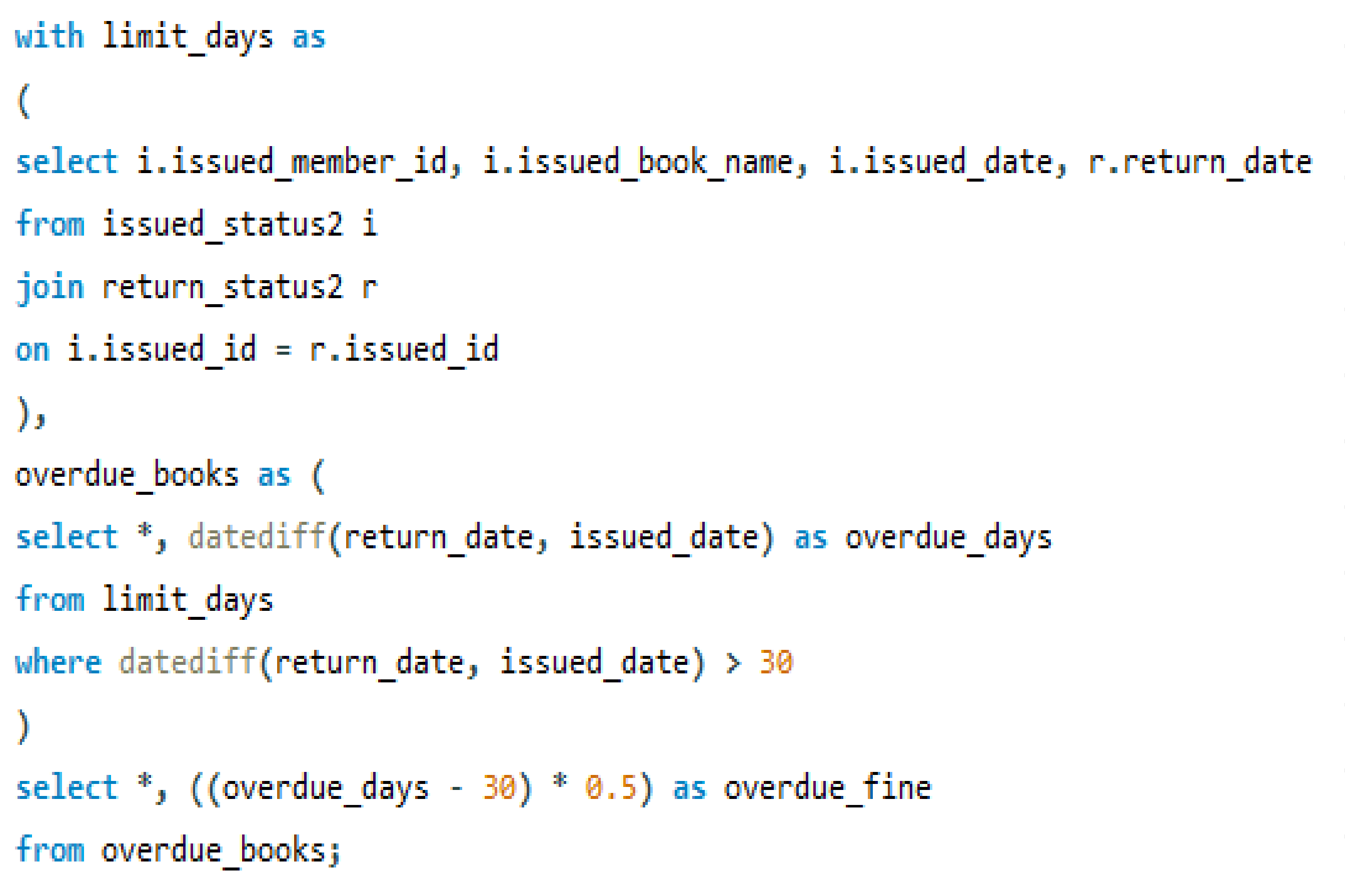
# Key KPIs

- Total books issued - 420
- Total books returned - 150
- Most borrowed category - History
- Damaged books count - 22
- Active members - 32
- Overdue books count - 72

# Key insight 1:

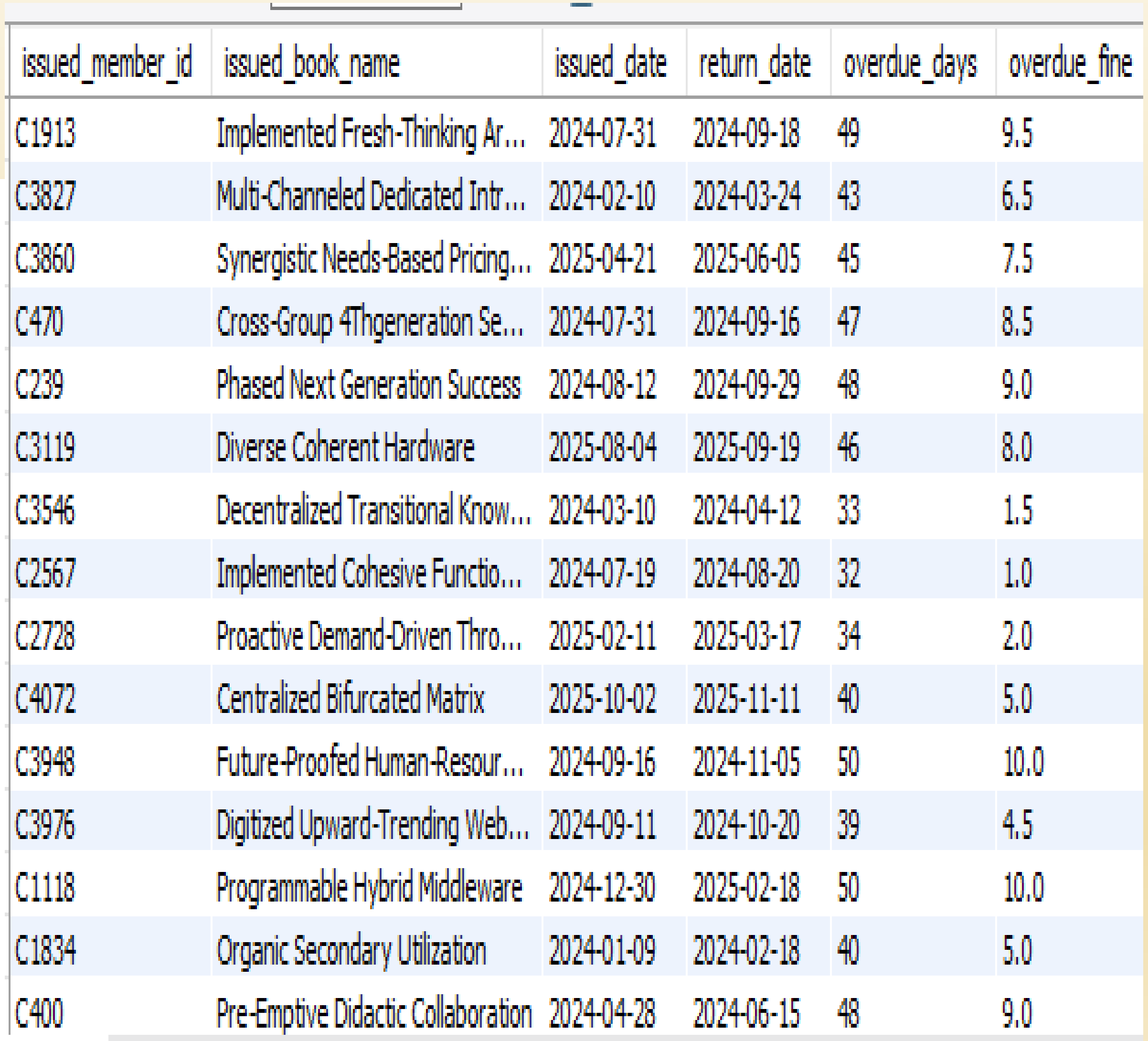
## Query

The SQL statement identifies all members with overdue books, calculates how many days overdue, and computes the appropriate fines ($0.50 per day)



```
with limit_days as
(
select i.issued_member_id, i.issued_book_name, i.issued_date, r.return_date
from issued_status2 i
join return_status2 r
on i.issued_id = r.issued_id
),
overdue_books as (
select *, datediff(return_date, issued_date) as overdue_days
from limit_days
where datediff(return_date, issued_date) > 30
)
select *, ((overdue_days - 30) * 0.5) as overdue_fine
from overdue_books;
```

## Output

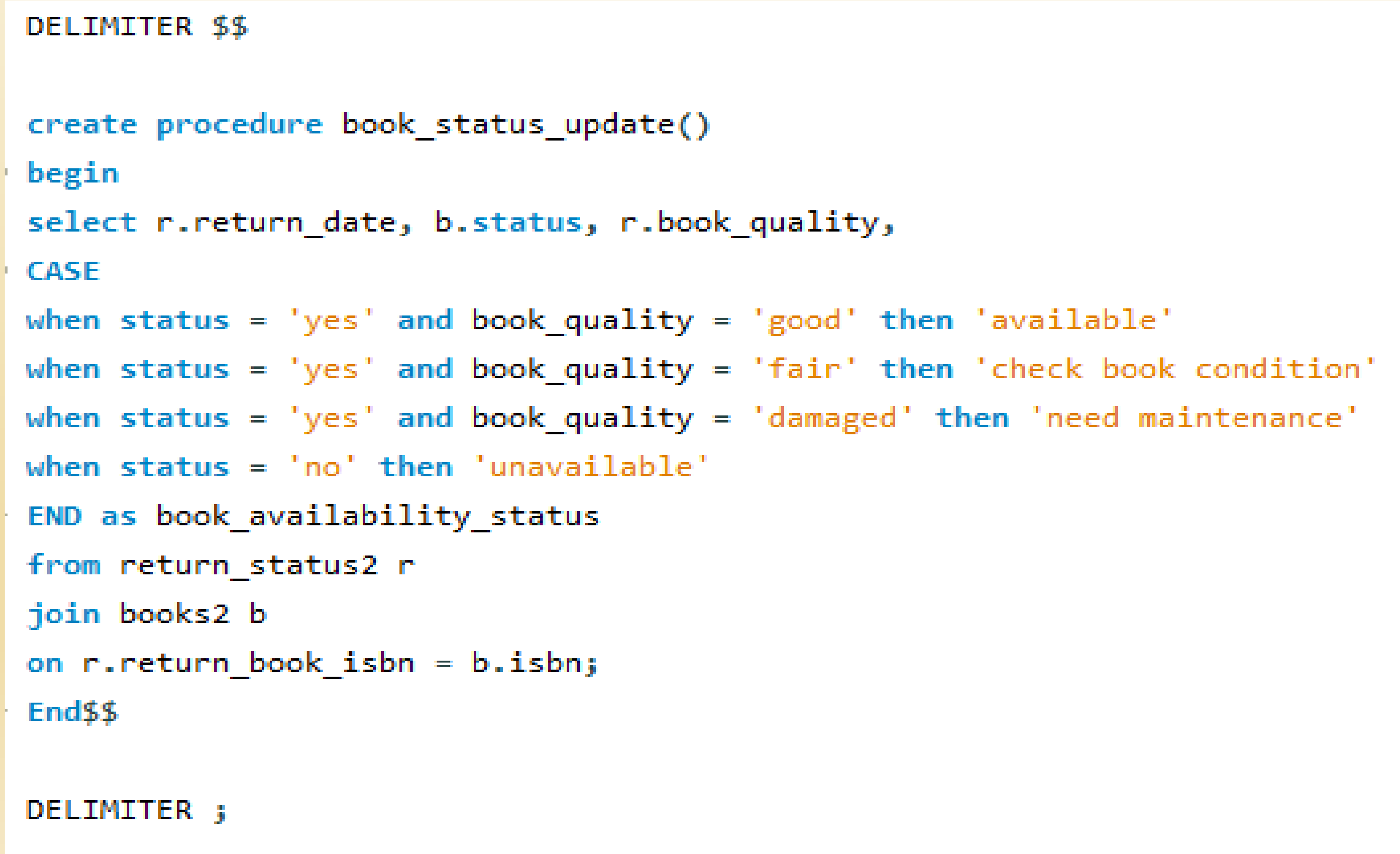


| issued_member_id | issued_book_name | issued_date | return_date | overdue_days | overdue_fine |
|---|---|---|---|---|---|
| C1913 | Implemented Fresh-Thinking Ar... | 2024-07-31 | 2024-09-18 | 49 | 9.5 |
| C3827 | Multi-Channeled Dedicated Intr... | 2024-02-10 | 2024-03-24 | 43 | 6.5 |
| C3860 | Synergistic Needs-Based Pricing... | 2025-04-21 | 2025-06-05 | 45 | 7.5 |
| C470 | Cross-Group 4Thgeneration Se... | 2024-07-31 | 2024-09-16 | 47 | 8.5 |
| C239 | Phased Next Generation Success | 2024-08-12 | 2024-09-29 | 48 | 9.0 |
| C3119 | Diverse Coherent Hardware | 2025-08-04 | 2025-09-19 | 46 | 8.0 |
| C3546 | Decentralized Transitional Know... | 2024-03-10 | 2024-04-12 | 33 | 1.5 |
| C2567 | Implemented Cohesive Functio... | 2024-07-19 | 2024-08-20 | 32 | 1.0 |
| C2728 | Proactive Demand-Driven Thro... | 2025-02-11 | 2025-03-17 | 34 | 2.0 |
| C4072 | Centralized Bifurcated Matrix | 2025-10-02 | 2025-11-11 | 40 | 5.0 |
| C3948 | Future-Proofed Human-Resour... | 2024-09-16 | 2024-11-05 | 50 | 10.0 |
| C3976 | Digitized Upward-Trending Web... | 2024-09-11 | 2024-10-20 | 39 | 4.5 |
| C1118 | Programmable Hybrid Middleware | 2024-12-30 | 2025-02-18 | 50 | 10.0 |
| C1834 | Organic Secondary Utilization | 2024-01-09 | 2024-02-18 | 40 | 5.0 |
| C400 | Pre-Emptive Didactic Collaboration | 2024-04-28 | 2024-06-15 | 48 | 9.0 |

# Key Insight 2: An automated procedure

## Procedure

- This stored procedure instantly updates book status to "available" when the return is logged/book is in good condition, and trigger the necessary follow-up actions when book condition says otherwise.

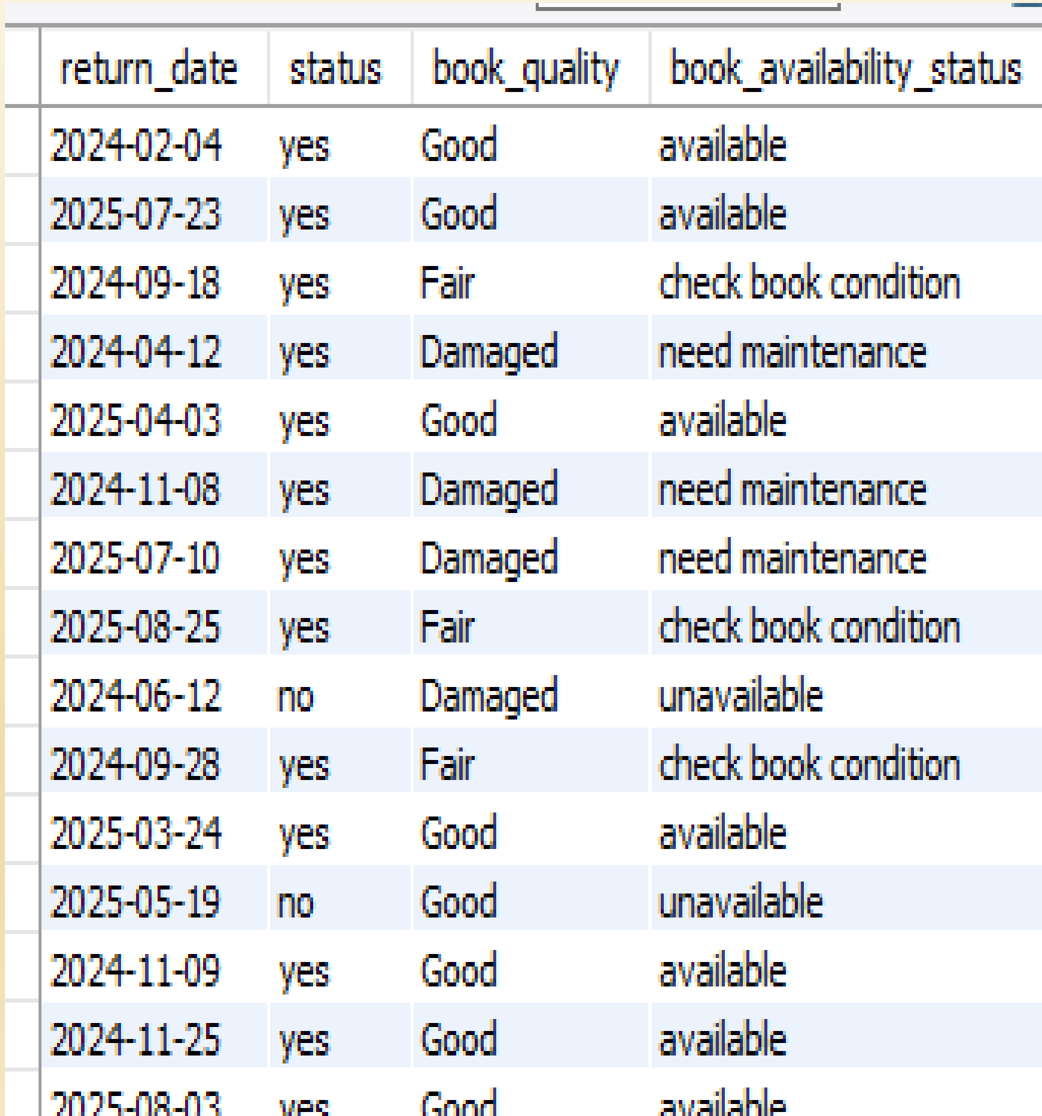


```
DELIMITER $$

create procedure book_status_update()
begin
select r.return_date, b.status, r.book_quality,
CASE
when status = 'yes' and book_quality = 'good' then 'available'
when status = 'yes' and book_quality = 'fair' then 'check book condition'
when status = 'yes' and book_quality = 'damaged' then 'need maintenance'
when status = 'no' then 'unavailable'
END as book_availability_status
from return_status2 r
join books2 b
on r.return_book_isbn = b.isbn;
End$$

DELIMITER ;
```

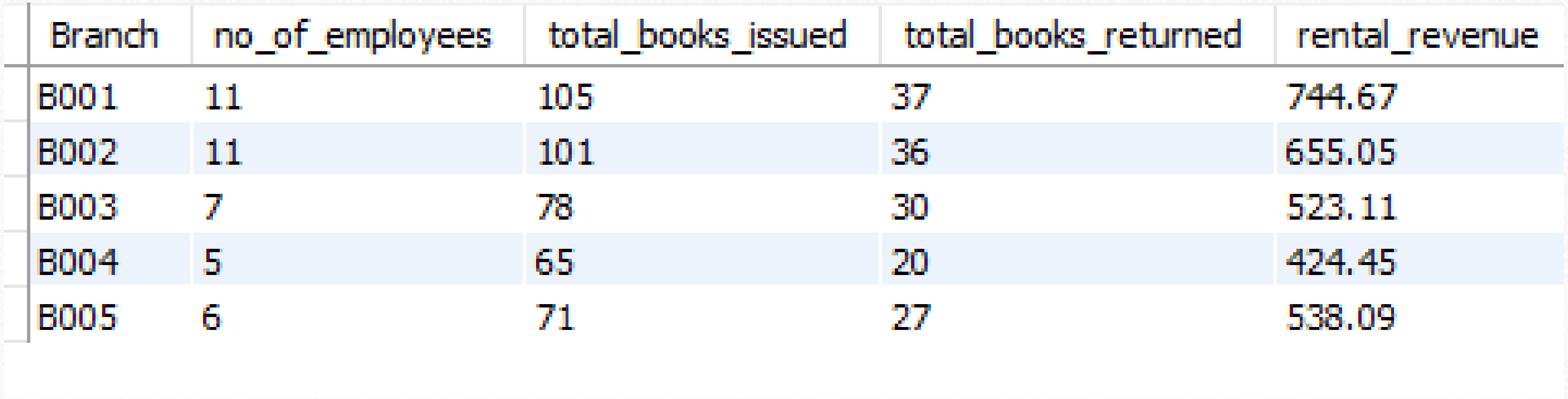
## Output



| return_date | status | book_quality | book_availability_status |
|---|---|---|---|
| 2024-02-04 | yes | Good | available |
| 2025-07-23 | yes | Good | available |
| 2024-09-18 | yes | Fair | check book condition |
| 2024-04-12 | yes | Damaged | need maintenance |
| 2025-04-03 | yes | Good | available |
| 2024-11-08 | yes | Damaged | need maintenance |
| 2025-07-10 | yes | Damaged | need maintenance |
| 2025-08-25 | yes | Fair | check book condition |
| 2024-06-12 | no | Damaged | unavailable |
| 2024-09-28 | yes | Fair | check book condition |
| 2025-03-24 | yes | Good | available |
| 2025-05-19 | no | Good | unavailable |
| 2024-11-09 | yes | Good | available |
| 2024-11-25 | yes | Good | available |
| 2025-08-03 | yes | Good | available |

# Key Insight 3: Performance report across branches

- B005 generates more revenue than the two previous branches before it. B001 generates the highest revenue

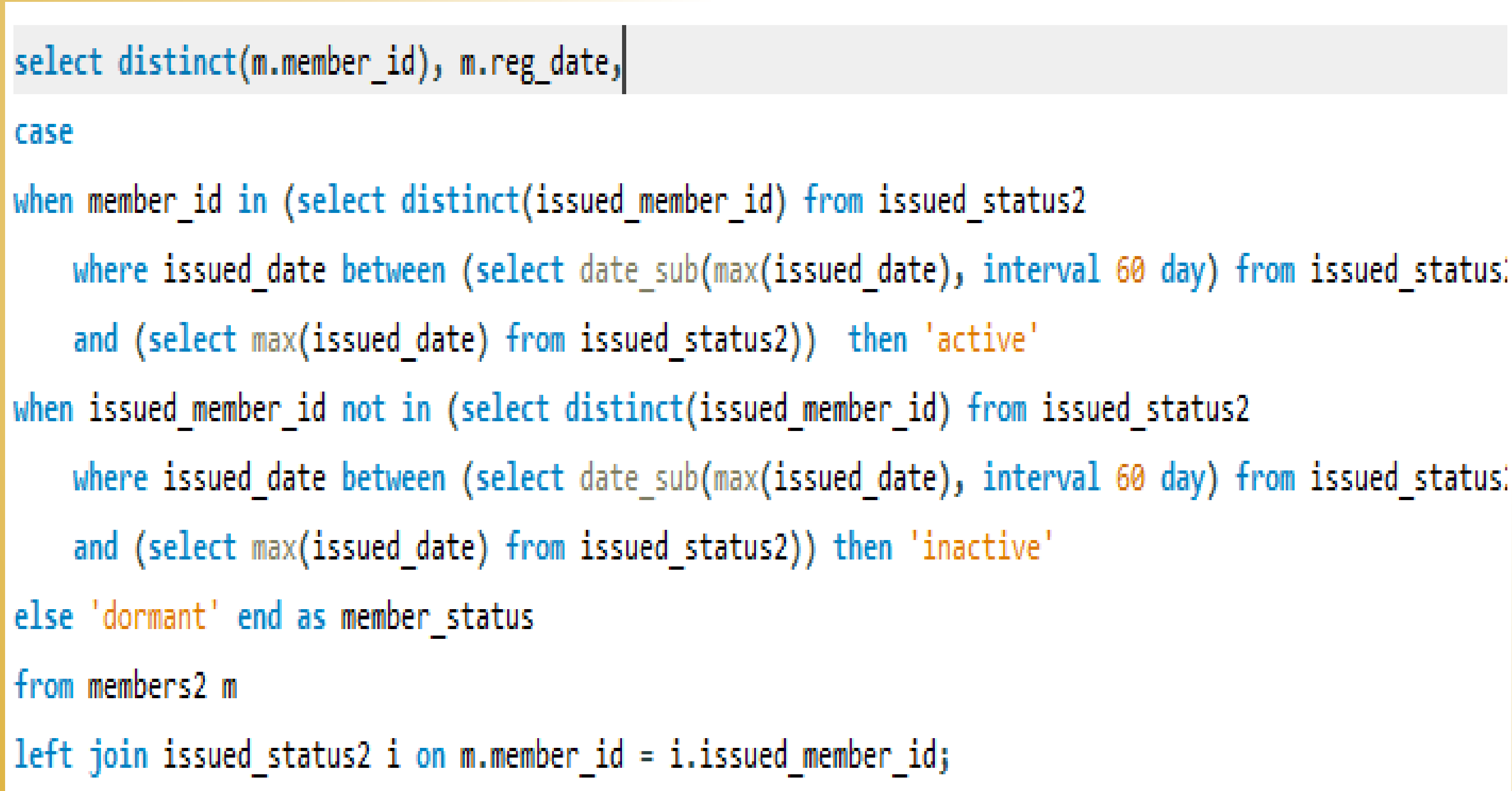


| Branch | no_of_employees | total_books_issued | total_books_returned | rental_revenue |
|---|---|---|---|---|
| B001 | 11 | 105 | 37 | 744.67 |
| B002 | 11 | 101 | 36 | 655.05 |
| B003 | 7 | 78 | 30 | 523.11 |
| B004 | 5 | 65 | 20 | 424.45 |
| B005 | 6 | 71 | 27 | 538.09 |

# Key Insight 4: A query that identify active and inactive members

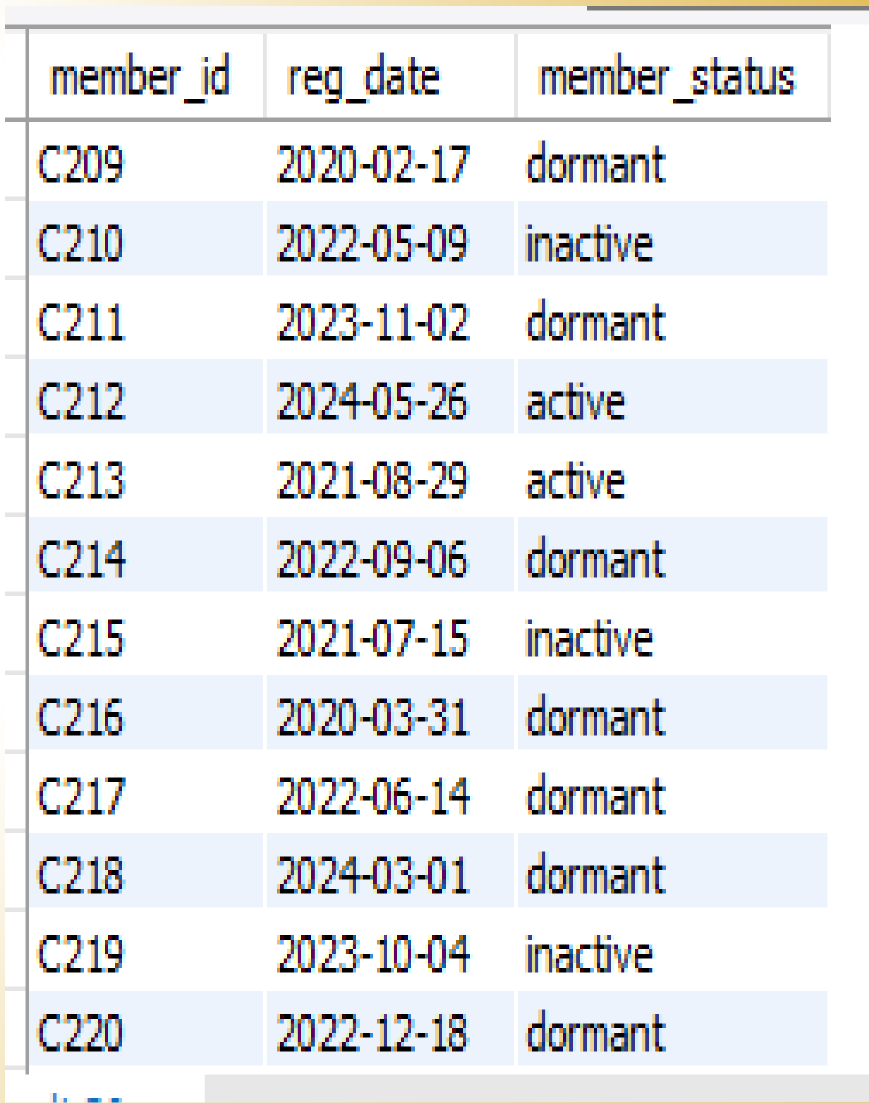
## Query

- The SQL statement groups the member status into ‘active’- those who have borrowed books in the last 60days, ‘inactive’- members who have borrowed books but not in the last 60days, and ‘dormant’- registered members who have never borrowed any book



```
select distinct(m.member_id), m.reg_date,
case
when member_id in (select distinct(issued_member_id) from issued_status2
    where issued_date between (select date_sub(max(issued_date), interval 60 day) from issued_status
    and (select max(issued_date) from issued_status2))  then 'active'
when issued_member_id not in (select distinct(issued_member_id) from issued_status2
    where issued_date between (select date_sub(max(issued_date), interval 60 day) from issued_status
    and (select max(issued_date) from issued_status2)) then 'inactive'
else 'dormant' end as member_status
from members2 m
left join issued_status2 i on m.member_id = i.issued_member_id;
```

## Output

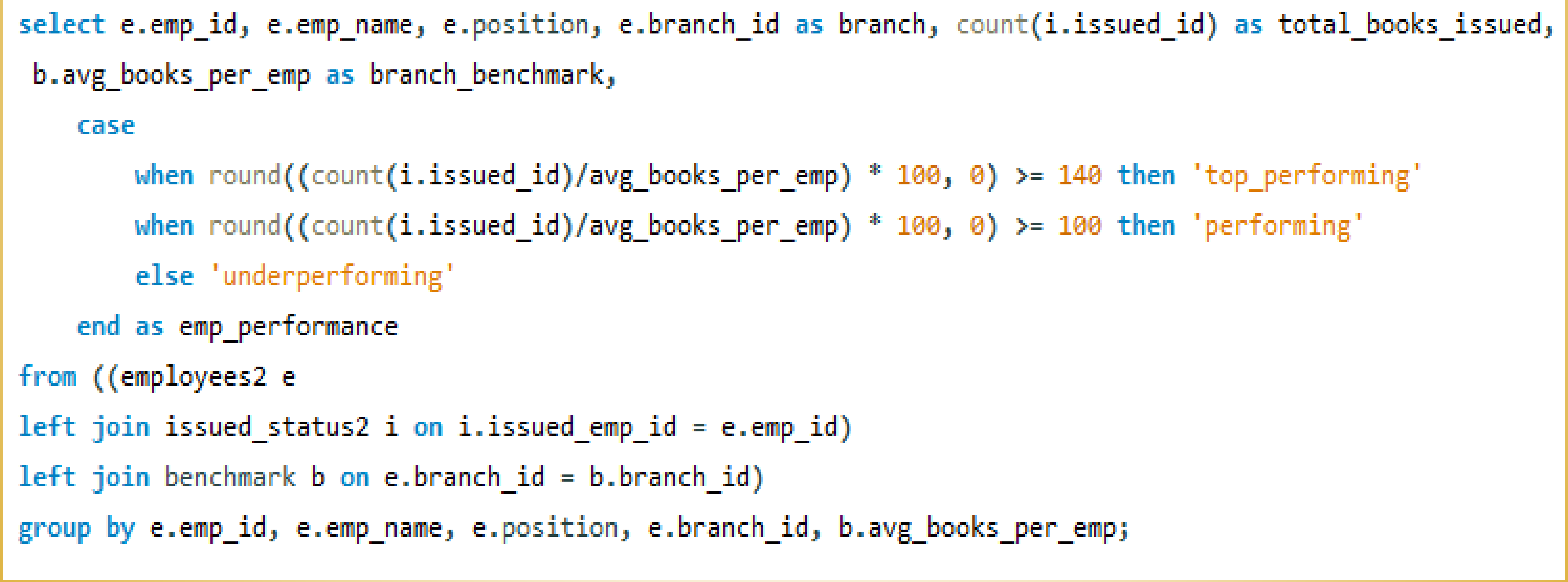


| member_id | reg_date | member_status |
|---|---|---|
| C209 | 2020-02-17 | dormant |
| C210 | 2022-05-09 | inactive |
| C211 | 2023-11-02 | dormant |
| C212 | 2024-05-26 | active |
| C213 | 2021-08-29 | active |
| C214 | 2022-09-06 | dormant |
| C215 | 2021-07-15 | inactive |
| C216 | 2020-03-31 | dormant |
| C217 | 2022-06-14 | dormant |
| C218 | 2024-03-01 | dormant |
| C219 | 2023-10-04 | inactive |
| C220 | 2022-12-18 | dormant |

# Key insight 5: staff productivity

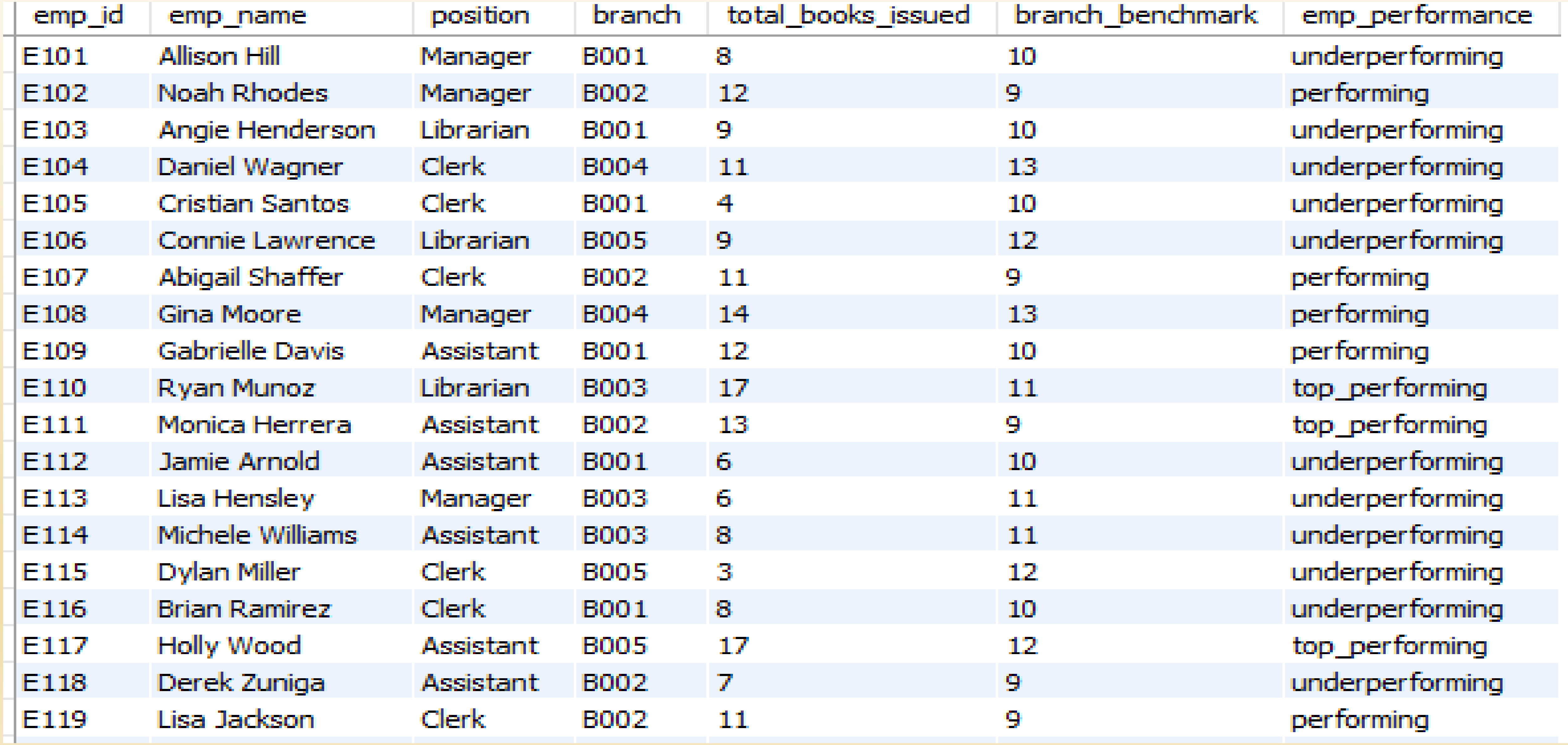
## Query

The SQL statement groups employees performance into 3: ‘top performing’ (employees that processed 140% above their branch’s benchmark of the number of books issue), ‘performing’ (those that have processed 100% and above), and ‘underperforming’ (as employees that processed below their branch benchmark).



```sql
select e.emp_id, e.emp_name, e.position, e.branch_id as branch, count(i.issued_id) as total_books_issued,
 b.avg_books_per_emp as branch_benchmark,
    case
        when round((count(i.issued_id)/avg_books_per_emp) * 100, 0) >= 140 then 'top_performing'
        when round((count(i.issued_id)/avg_books_per_emp) * 100, 0) >= 100 then 'performing'
        else 'underperforming'
    end as emp_performance
from ((employees2 e
left join issued_status2 i on i.issued_emp_id = e.emp_id)
left join benchmark b on e.branch_id = b.branch_id)
group by e.emp_id, e.emp_name, e.position, e.branch_id, b.avg_books_per_emp;
```

# Key insight 5: Output

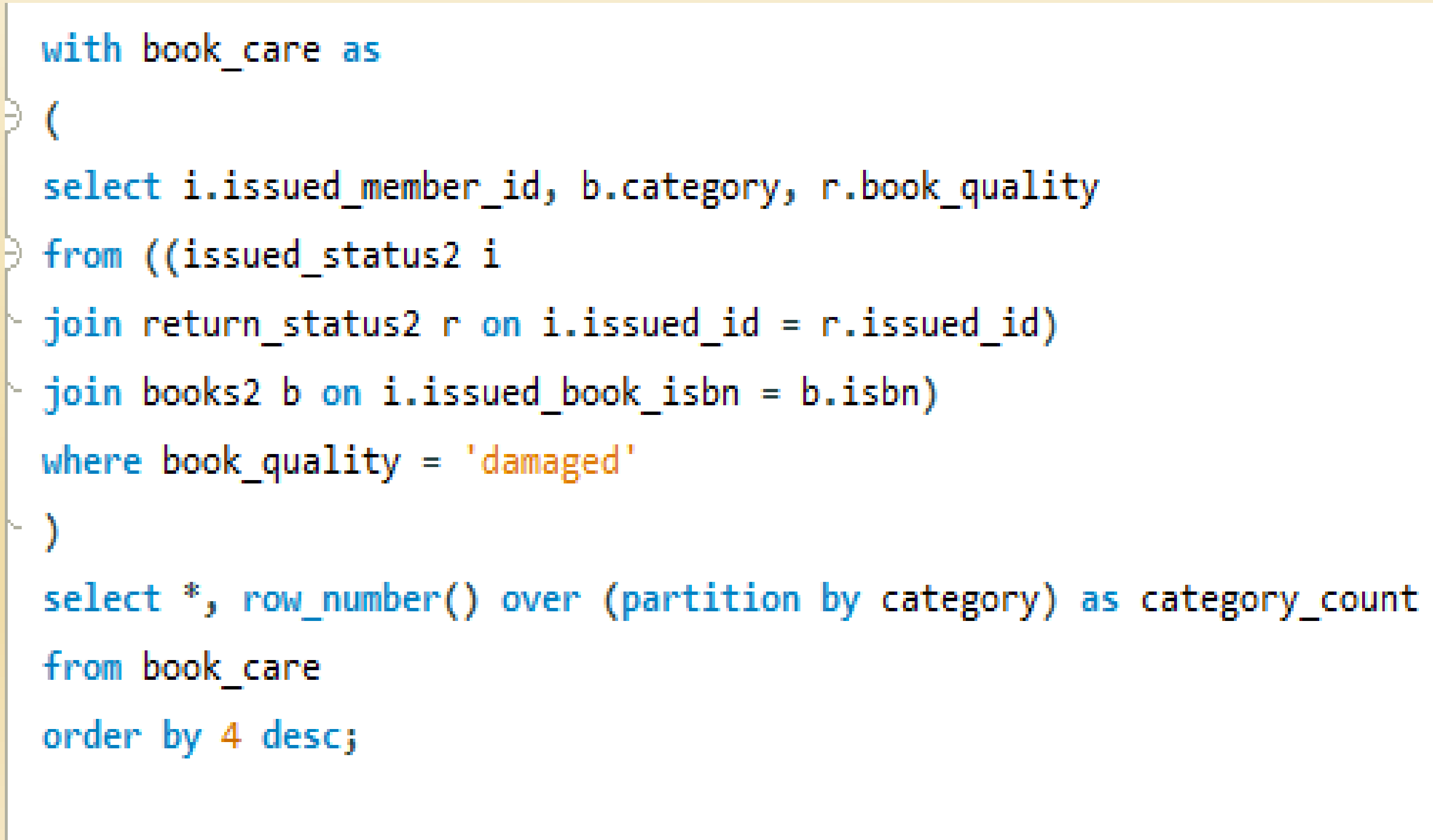


| emp_id | emp_name | position | branch | total_books_issued | branch_benchmark | emp_performance |
|---|---|---|---|---|---|---|
| E101 | Allison Hill | Manager | B001 | 8 | 10 | underperforming |
| E102 | Noah Rhodes | Manager | B002 | 12 | 9 | performing |
| E103 | Angie Henderson | Librarian | B001 | 9 | 10 | underperforming |
| E104 | Daniel Wagner | Clerk | B004 | 11 | 13 | underperforming |
| E105 | Cristian Santos | Clerk | B001 | 4 | 10 | underperforming |
| E106 | Connie Lawrence | Librarian | B005 | 9 | 12 | underperforming |
| E107 | Abigail Shaffer | Clerk | B002 | 11 | 9 | performing |
| E108 | Gina Moore | Manager | B004 | 14 | 13 | performing |
| E109 | Gabrielle Davis | Assistant | B001 | 12 | 10 | performing |
| E110 | Ryan Munoz | Librarian | B003 | 17 | 11 | top_performing |
| E111 | Monica Herrera | Assistant | B002 | 13 | 9 | top_performing |
| E112 | Jamie Arnold | Assistant | B001 | 6 | 10 | underperforming |
| E113 | Lisa Hensley | Manager | B003 | 6 | 11 | underperforming |
| E114 | Michele Williams | Assistant | B003 | 8 | 11 | underperforming |
| E115 | Dylan Miller | Clerk | B005 | 3 | 12 | underperforming |
| E116 | Brian Ramirez | Clerk | B001 | 8 | 10 | underperforming |
| E117 | Holly Wood | Assistant | B005 | 17 | 12 | top_performing |
| E118 | Derek Zuniga | Assistant | B002 | 7 | 9 | underperforming |
| E119 | Lisa Jackson | Clerk | B002 | 11 | 9 | performing |

# Key Insight 6: A query that identifies members who have issued books marked as "damaged"

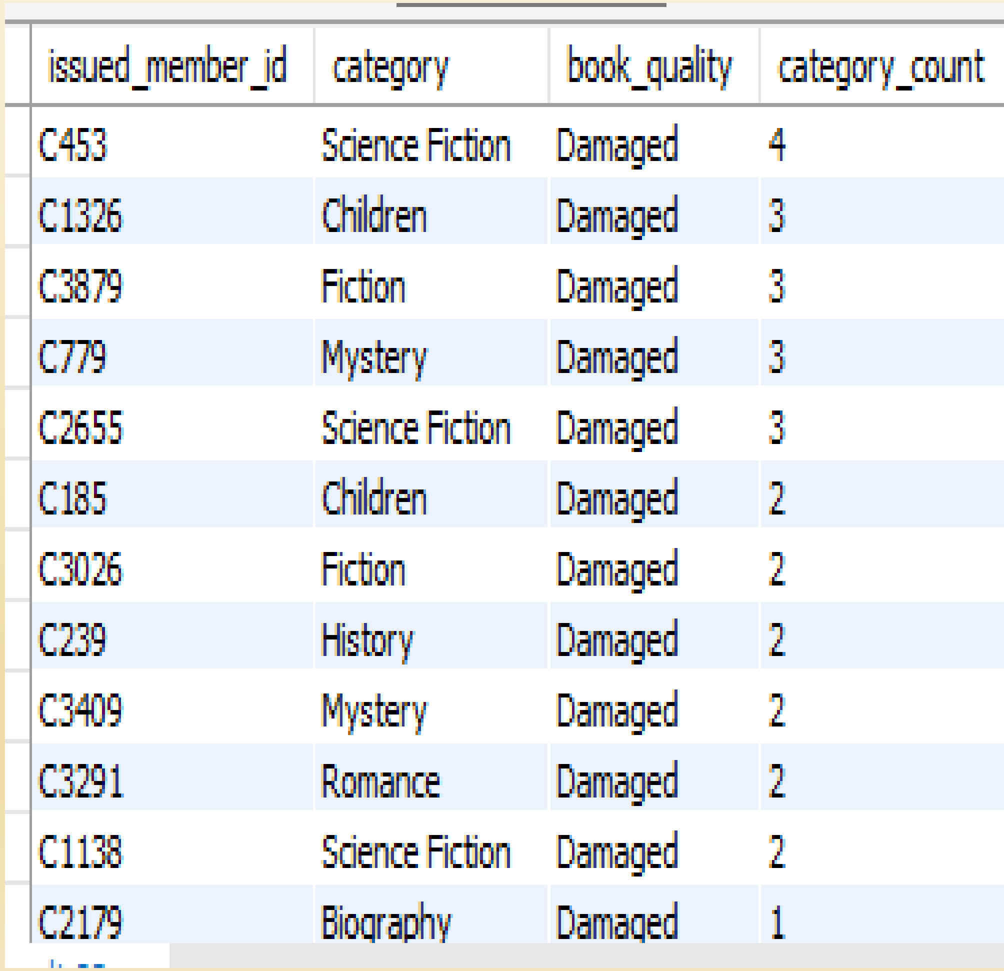
## Query

The SQL statement identifies members that have return books as damaged and showing the category of books as such



```
with book_care as
(
select i.issued_member_id, b.category, r.book_quality
from ((issued_status2 i
join return_status2 r on i.issued_id = r.issued_id)
join books2 b on i.issued_book_isbn = b.isbn)
where book_quality = 'damaged'
)
select *, row_number() over (partition by category) as category_count
from book_care
order by 4 desc;
```

## Output



| issued_member_id | category | book_quality | category_count |
|---|---|---|---|
| C453 | Science Fiction | Damaged | 4 |
| C1326 | Children | Damaged | 3 |
| C3879 | Fiction | Damaged | 3 |
| C779 | Mystery | Damaged | 3 |
| C2655 | Science Fiction | Damaged | 3 |
| C185 | Children | Damaged | 2 |
| C3026 | Fiction | Damaged | 2 |
| C239 | History | Damaged | 2 |
| C3409 | Mystery | Damaged | 2 |
| C3291 | Romance | Damaged | 2 |
| C1138 | Science Fiction | Damaged | 2 |
| C2179 | Biography | Damaged | 1 |

# Summary of findings

- B005 generates more revenue than the two previous branches before it. B001 generates the highest revenue
- Each employee makes an average of 89 in revenue in branch B005 while average of 59 in revenue in branch B002
- 420 members have borrowed books out of which only 32 are active (have borrowed books in the last 60days)
- Over 3000 members are registered but have never borrowed books
- On the average, each staff is suppose to issue at least 9books across branches.
- Identified five staffs who are considered as top performers and twenty staffs as under-performing
- Book category that are returned damaged are usually the science fiction and children books
- Most borrowed books are in the History category, followed by Fiction and then Thriller

# Recommendations

- I suggest we re-introduce the "Summer Reading Challenge" promotion or any other promotions to encourage more numbers to be active
- Increase overdue fines to encourage members to return books on or before due date
- Branch B005 is doing well in terms of performance and revenue compared to other branches
- Target re-engagement campaigns for dormant members
- Training program for underperforming staff
- Do a staff reshuffling across branches if possible
- Have informed conversations with members who usually borrow science fiction and children books about book care
- Carry out another member satisfaction ratings related to service speed in the nearest future to check for improved rate

# Conclusion

- In conclusion, implementing targeted promotions, stricter compliance measures, staff development and redistribution, member re-engagement initiatives, and follow-up satisfaction assessments—while leveraging high-performing branches as benchmarks—will collectively improve operational efficiency, member activity, service quality, and overall organizational performance.

Thank you