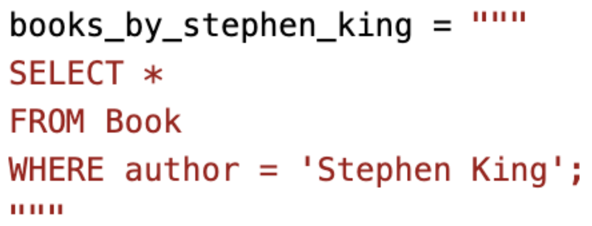
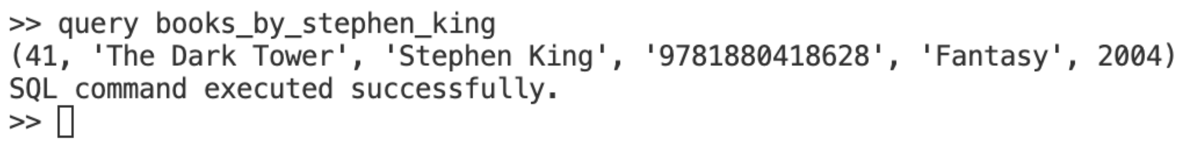
Pre-Defined Queries:

1. **List all books by a specific author**
   Display all books in the library collection written by a particular author.



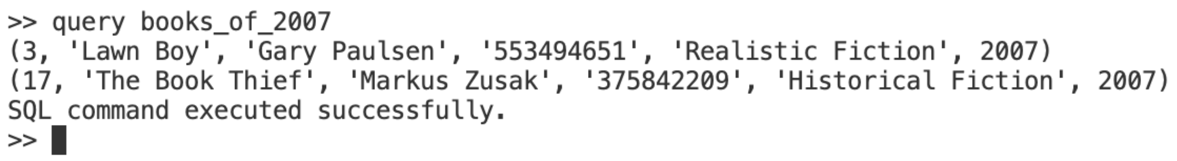
```
books_by_stephen_king = """
SELECT *
FROM Book
WHERE author = 'Stephen King';
"""
```



```
>> query books_by_stephen_king
(41, 'The Dark Tower', 'Stephen King', '9781880418628', 'Fantasy', 2004)
SQL command executed successfully.
>> 
```

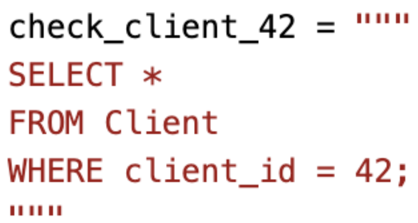
2. **Find books by publication year**
   Retrieve a list of books published in a specific year.

```
books_of_2007_query = """
SELECT *
FROM Book
WHERE publication_year = 2007;
"""
```

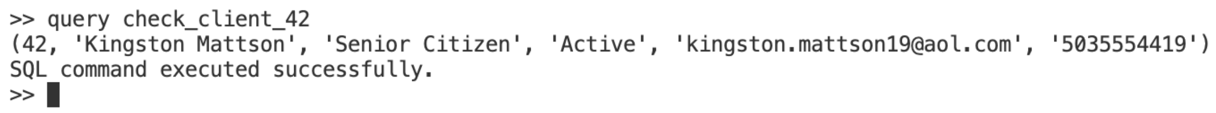


```
>> query books_of_2007
(3, 'Lawn Boy', 'Gary Paulsen', '553494651', 'Realistic Fiction', 2007)
(17, 'The Book Thief', 'Markus Zusak', '375842209', 'Historical Fiction', 2007)
SQL command executed successfully.
>> 
```

3. **Check membership status**
   Display the current status and account information for a specific client based on their unique ID.



```
check_client_42 = """
SELECT *
FROM Client
WHERE client_id = 42;
"""
```

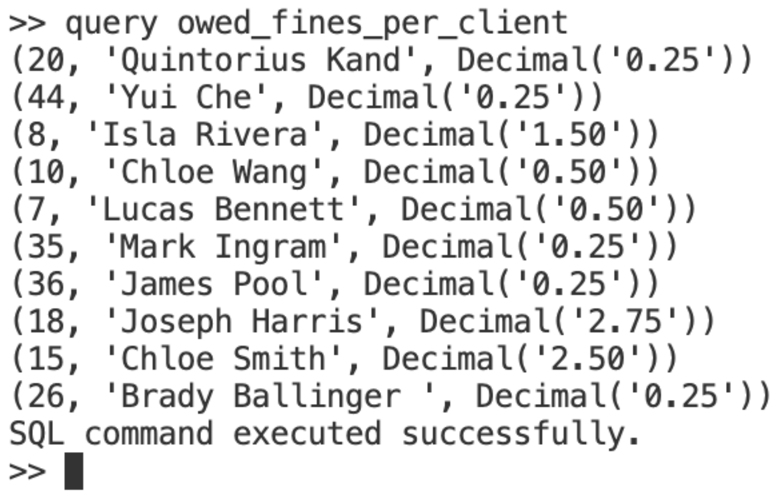


```
>> query check_client_42
(42, 'Kingston Mattson', 'Senior Citizen', 'Active', 'kingston.mattson19@aol.com', '5035554419')
SQL command executed successfully.
>> █
```

4. **Fine calculation**

Calculate the total fines owed by each member, considering overdue books and a daily fine rate (e.g., $0.25 per day).

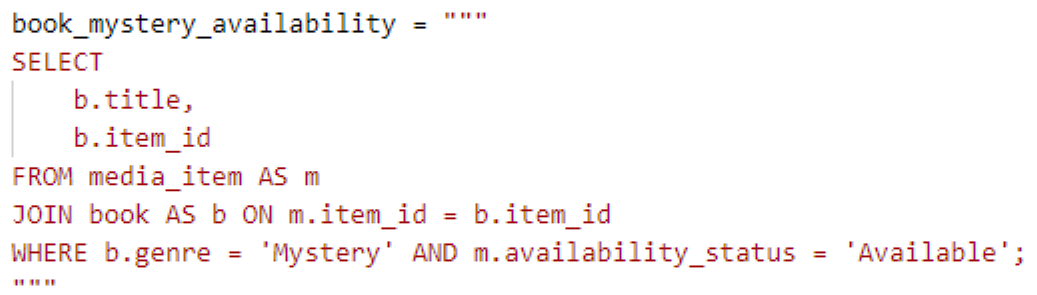
```
owed_fines_per_client = """
SELECT
  c.client_id,
  c.name,
  SUM(
    GREATEST(
      (t.returned_date::date - t.expected_return_date::date),
      0
    ) * 0.25
  ) AS total_owed
FROM transaction AS t
JOIN client AS c ON t.client_id = c.client_id
WHERE t.returned_date IS NOT NULL -- Only consider returned items
  AND t.returned_date > t.expected_return_date
GROUP BY c.client_id, c.name
HAVING SUM(GREATEST((t.returned_date::date - t.expected_return_date::date), 0) * 0.25) > 0;
"""
```



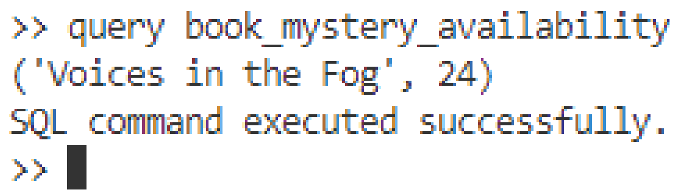
```
>> query owed_fines_per_client
(20, 'Quintorius Kand', Decimal('0.25'))
(44, 'Yui Che', Decimal('0.25'))
(8, 'Isla Rivera', Decimal('1.50'))
(10, 'Chloe Wang', Decimal('0.50'))
(7, 'Lucas Bennett', Decimal('0.50'))
(35, 'Mark Ingram', Decimal('0.25'))
(36, 'James Pool', Decimal('0.25'))
(18, 'Joseph Harris', Decimal('2.75'))
(15, 'Chloe Smith', Decimal('2.50'))
(26, 'Brady Ballinger ', Decimal('0.25'))
SQL command executed successfully.
>> 
```

5. Book availability

Display a list of all available books (not currently borrowed) within a specific genre (e.g.) mystery).



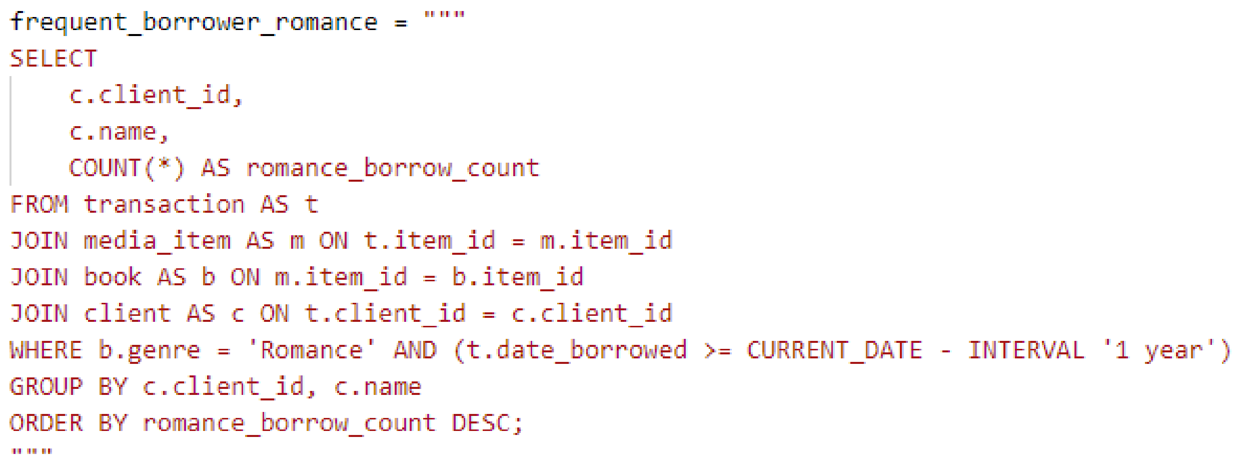
```
book_mystery_availability = """
SELECT
    b.title,
    b.item_id
FROM media_item AS m
JOIN book AS b ON m.item_id = b.item_id
WHERE b.genre = 'Mystery' AND m.availability_status = 'Available';
"""
```



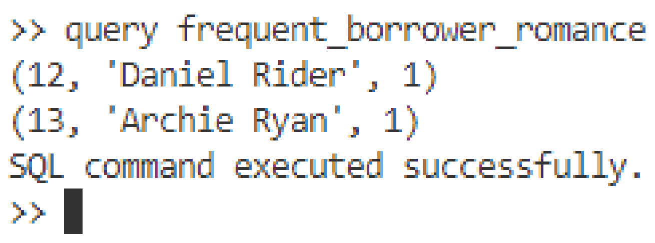
```
>> query book_mystery_availability
('Voices in the Fog', 24)
SQL command executed successfully.
>> 
```

6. Frequent borrowers of a specific genre

Identify the members who have borrowed the most books in a particular genre (e.g., "Romance") in the last year



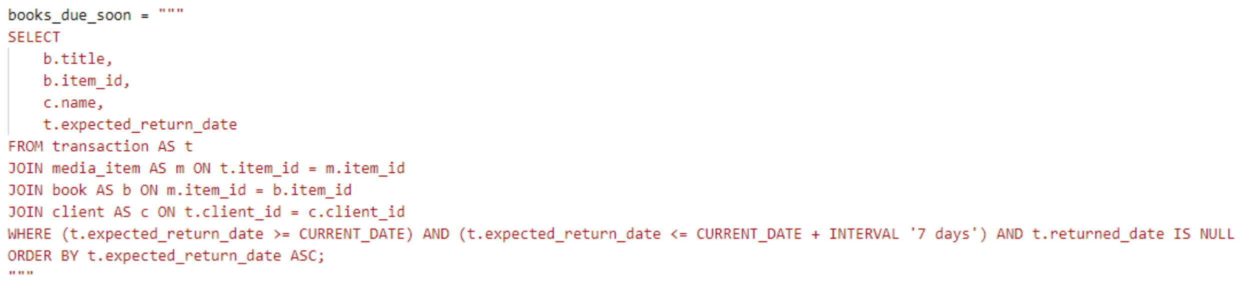
```
frequent_borrower_romance = """
SELECT
    c.client_id,
    c.name,
    COUNT(*) AS romance_borrow_count
FROM transaction AS t
JOIN media_item AS m ON t.item_id = m.item_id
JOIN book AS b ON m.item_id = b.item_id
JOIN client AS c ON t.client_id = c.client_id
WHERE b.genre = 'Romance' AND (t.date_borrowed >= CURRENT_DATE - INTERVAL '1 year')
GROUP BY c.client_id, c.name
ORDER BY romance_borrow_count DESC;
"""
```



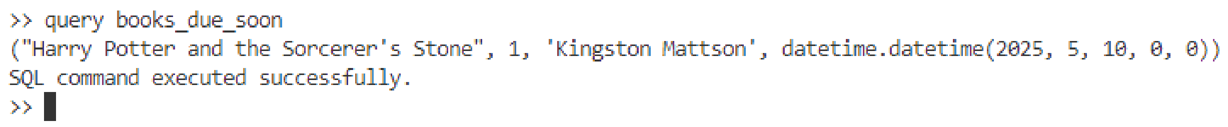
```
>> query frequent_borrower_romance
(12, 'Daniel Rider', 1)
(13, 'Archie Ryan', 1)
SQL command executed successfully.
>> 
```

7. Books due soon

Generate a report of all books due within the next week, sorted by due date.



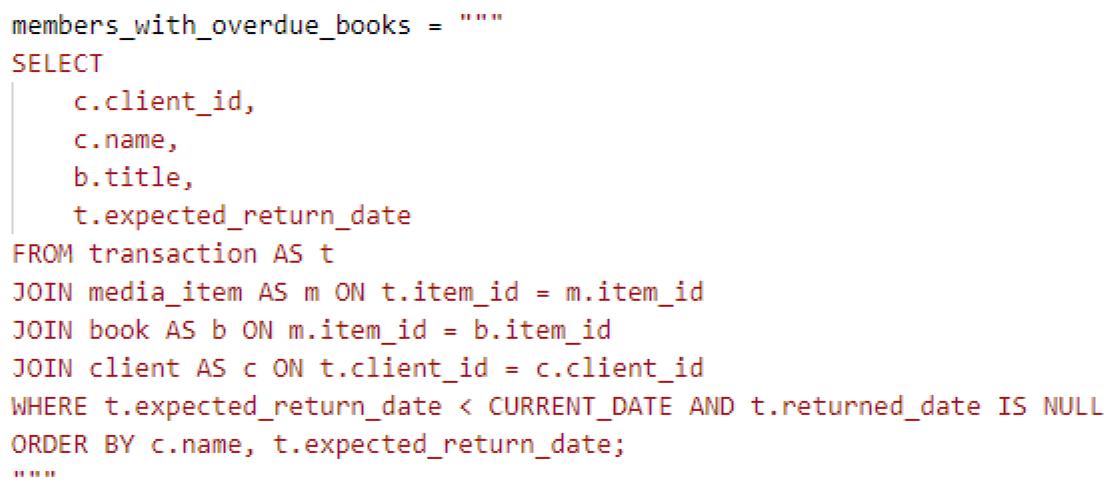
```
books_due_soon = """
SELECT
    b.title,
    b.item_id,
    c.name,
    t.expected_return_date
FROM transaction AS t
JOIN media_item AS m ON t.item_id = m.item_id
JOIN book AS b ON m.item_id = b.item_id
JOIN client AS c ON t.client_id = c.client_id
WHERE (t.expected_return_date >= CURRENT_DATE) AND (t.expected_return_date <= CURRENT_DATE + INTERVAL '7 days') AND t.returned_date IS NULL
ORDER BY t.expected_return_date ASC;
"""
```



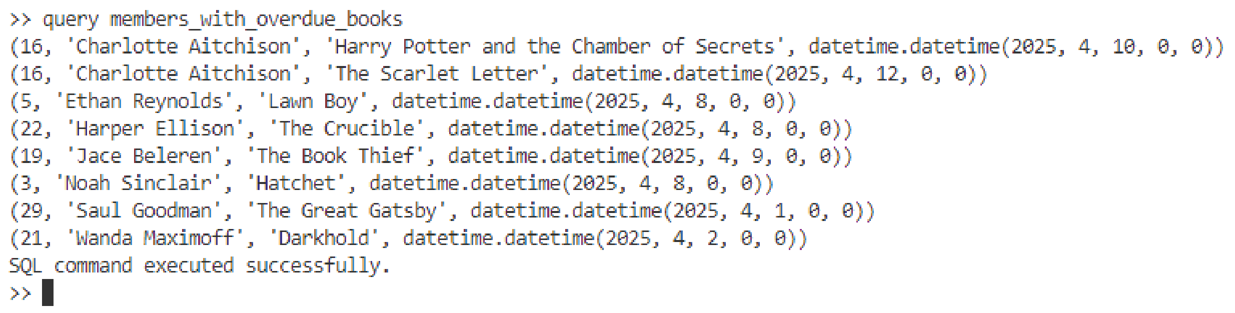
```
>> query books_due_soon
("Harry Potter and the Sorcerer's Stone", 1, 'Kingston Mattson', datetime.datetime(2025, 5, 10, 0, 0))
SQL command executed successfully.
>>
```

8. Members with overdue books

List all members who currently have at least one overdue book, along with the titles of the overdue books.



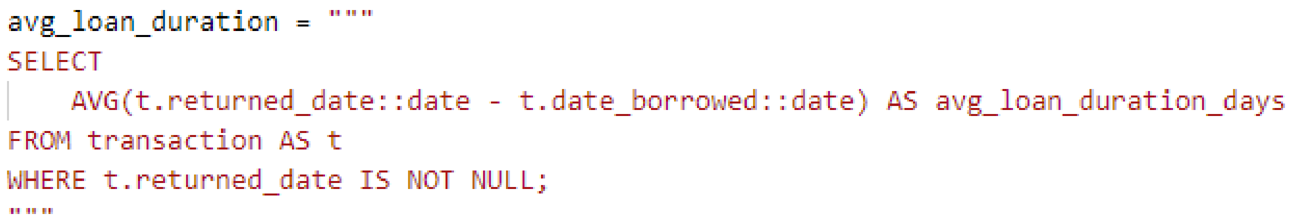
```
members_with_overdue_books = """
SELECT
    c.client_id,
    c.name,
    b.title,
    t.expected_return_date
FROM transaction AS t
JOIN media_item AS m ON t.item_id = m.item_id
JOIN book AS b ON m.item_id = b.item_id
JOIN client AS c ON t.client_id = c.client_id
WHERE t.expected_return_date < CURRENT_DATE AND t.returned_date IS NULL
ORDER BY c.name, t.expected_return_date;
"""
```



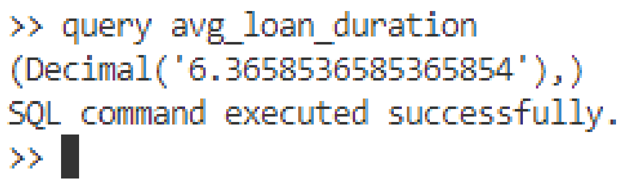
```
>> query members_with_overdue_books
(16, 'Charlotte Aitchison', 'Harry Potter and the Chamber of Secrets', datetime.datetime(2025, 4, 10, 0, 0))
(16, 'Charlotte Aitchison', 'The Scarlet Letter', datetime.datetime(2025, 4, 12, 0, 0))
(5, 'Ethan Reynolds', 'Lawn Boy', datetime.datetime(2025, 4, 8, 0, 0))
(22, 'Harper Ellison', 'The Crucible', datetime.datetime(2025, 4, 8, 0, 0))
(19, 'Jace Beleren', 'The Book Thief', datetime.datetime(2025, 4, 9, 0, 0))
(3, 'Noah Sinclair', 'Hatchet', datetime.datetime(2025, 4, 8, 0, 0))
(29, 'Saul Goodman', 'The Great Gatsby', datetime.datetime(2025, 4, 1, 0, 0))
(21, 'Wanda Maximoff', 'Darkhold', datetime.datetime(2025, 4, 2, 0, 0))
SQL command executed successfully.
>> 
```

9. Average borrowing time

Calculate the average number of days members borrow books for a specific genre.



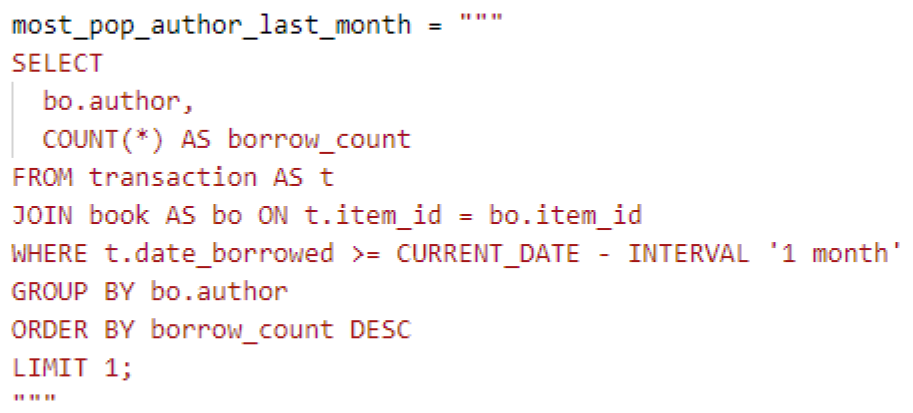
```
avg_loan_duration = """
SELECT
    AVG(t.returned_date::date - t.date_borrowed::date) AS avg_loan_duration_days
FROM transaction AS t
WHERE t.returned_date IS NOT NULL;
"""
```



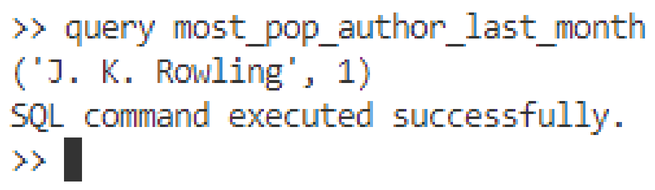
```
>> query avg_loan_duration
(Decimal('6.3658536585365854'),)
SQL command executed successfully.
>>
```

10. Most popular author in the last month

Determine the author whose books have been borrowed the most in the last month.



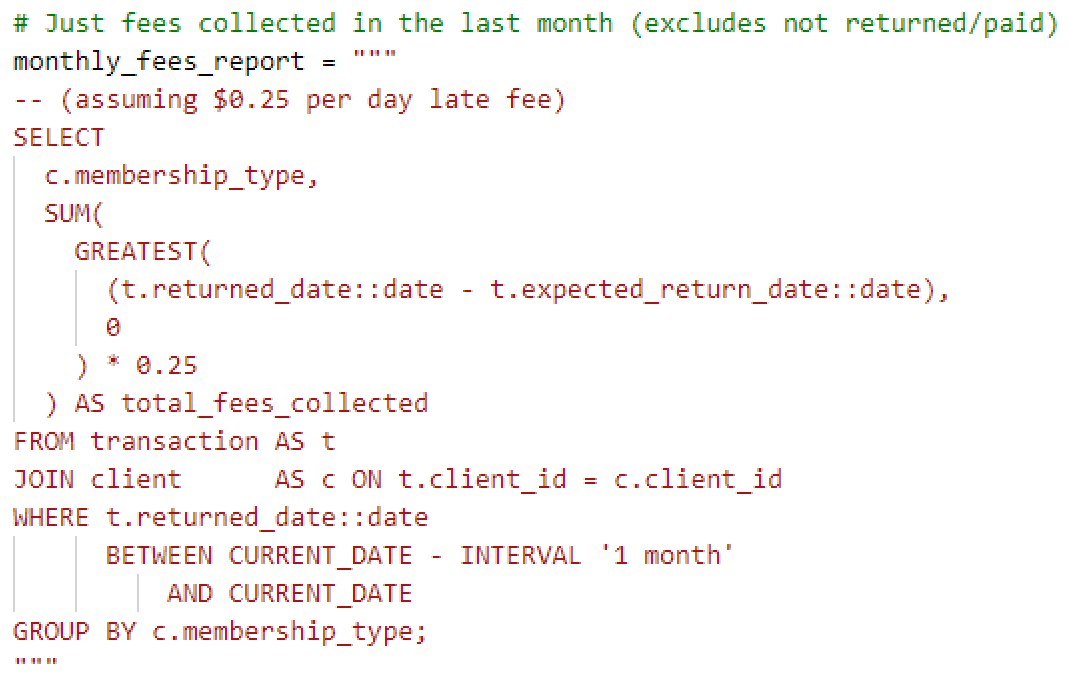
```
most_pop_author_last_month = """
SELECT
    bo.author,
    COUNT(*) AS borrow_count
FROM transaction AS t
JOIN book AS bo ON t.item_id = bo.item_id
WHERE t.date_borrowed >= CURRENT_DATE - INTERVAL '1 month'
GROUP BY bo.author
ORDER BY borrow_count DESC
LIMIT 1;
"""
```



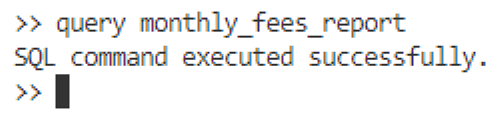
```
>> query most_pop_author_last_month
('J. K. Rowling', 1)
SQL command executed successfully.
>> 
```

11. Monthly fees report

Generate a report of total fees collected within the last month, broken down by membership type.



```
# Just fees collected in the last month (excludes not returned/paid)
monthly_fees_report = """
-- (assuming $0.25 per day late fee)
SELECT
  c.membership_type,
  SUM(
    GREATEST(
      (t.returned_date::date - t.expected_return_date::date),
      0
    ) * 0.25
  ) AS total_fees_collected
FROM transaction AS t
JOIN client      AS c ON t.client_id = c.client_id
WHERE t.returned_date::date
      BETWEEN CURRENT_DATE - INTERVAL '1 month'
          AND CURRENT_DATE
GROUP BY c.membership_type;
"""
```

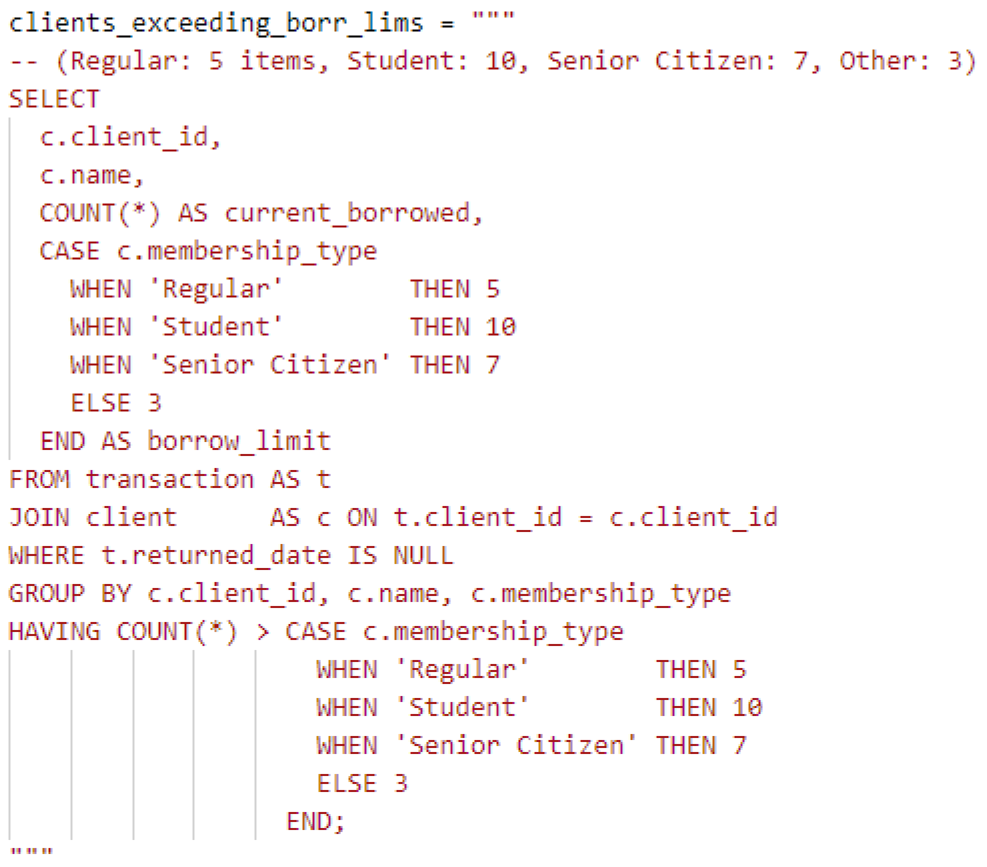


```
>> query monthly_fees_report
SQL command executed successfully.
>>
```

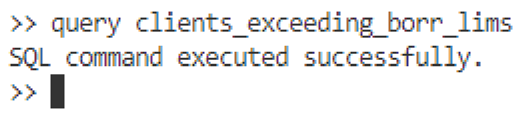
Note: The output is empty since none of our books have been returned late in the last month,

12. Exceeded borrowing limits

Produce a list of clients who have exceeded their borrowing limits.



```
clients_exceeding_borr_lims = """
-- (Regular: 5 items, Student: 10, Senior Citizen: 7, Other: 3)
SELECT
  c.client_id,
  c.name,
  COUNT(*) AS current_borrowed,
  CASE c.membership_type
    WHEN 'Regular'        THEN 5
    WHEN 'Student'        THEN 10
    WHEN 'Senior Citizen' THEN 7
    ELSE 3
  END AS borrow_limit
FROM transaction AS t
JOIN client      AS c ON t.client_id = c.client_id
WHERE t.returned_date IS NULL
GROUP BY c.client_id, c.name, c.membership_type
HAVING COUNT(*) > CASE c.membership_type
                    WHEN 'Regular'        THEN 5
                    WHEN 'Student'        THEN 10
                    WHEN 'Senior Citizen' THEN 7
                    ELSE 3
                  END;
"""
```

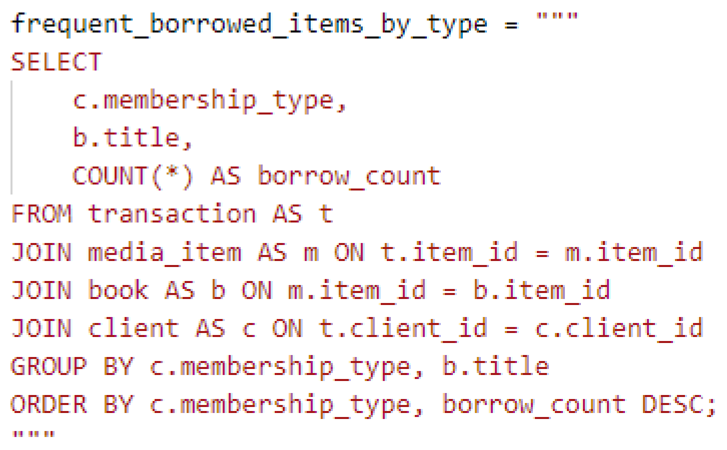


```
>> query clients_exceeding_borr_lims
SQL command executed successfully.
>>
```

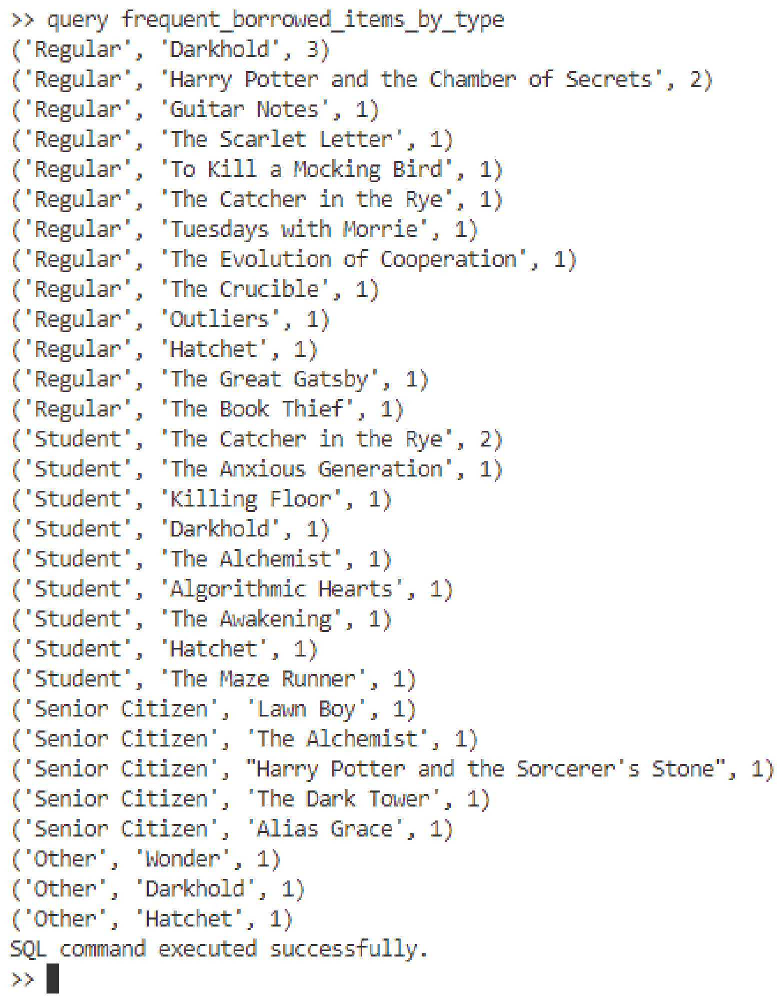
Note: Returns no clients since there are no clients exceeding their borrowing limits, as verified by the member engagement report results showing no client having more than 3 total transactions.

13. Frequent borrowed items by client type

Determine the most frequently borrowed items by each client type.



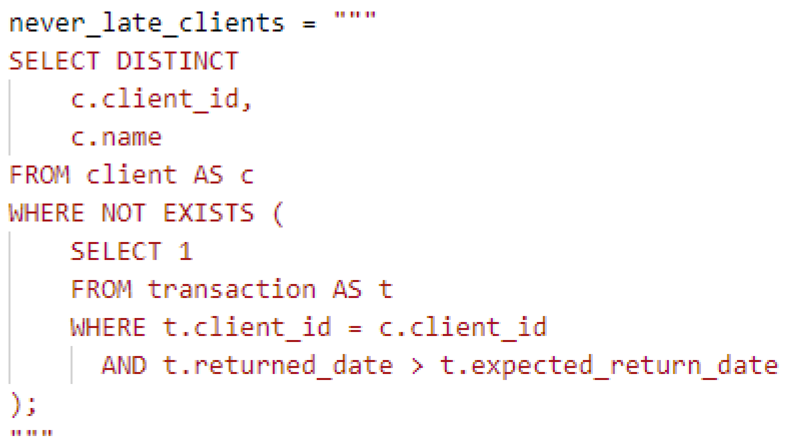
```
frequent_borrowed_items_by_type = """
SELECT
    c.membership_type,
    b.title,
    COUNT(*) AS borrow_count
FROM transaction AS t
JOIN media_item AS m ON t.item_id = m.item_id
JOIN book AS b ON m.item_id = b.item_id
JOIN client AS c ON t.client_id = c.client_id
GROUP BY c.membership_type, b.title
ORDER BY c.membership_type, borrow_count DESC;
"""
```



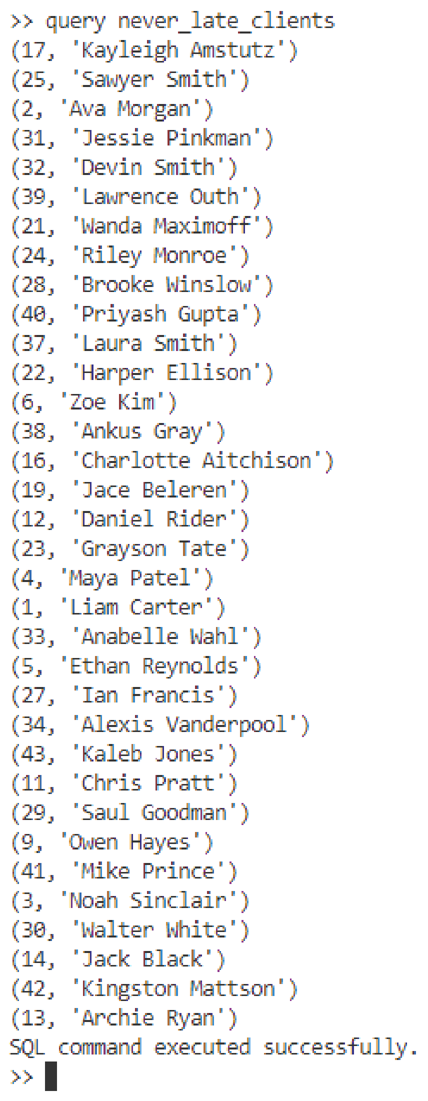
```
>> query frequent_borrowed_items_by_type
('Regular', 'Darkhold', 3)
('Regular', 'Harry Potter and the Chamber of Secrets', 2)
('Regular', 'Guitar Notes', 1)
('Regular', 'The Scarlet Letter', 1)
('Regular', 'To Kill a Mocking Bird', 1)
('Regular', 'The Catcher in the Rye', 1)
('Regular', 'Tuesdays with Morrie', 1)
('Regular', 'The Evolution of Cooperation', 1)
('Regular', 'The Crucible', 1)
('Regular', 'Outliers', 1)
('Regular', 'Hatchet', 1)
('Regular', 'The Great Gatsby', 1)
('Regular', 'The Book Thief', 1)
('Student', 'The Catcher in the Rye', 2)
('Student', 'The Anxious Generation', 1)
('Student', 'Killing Floor', 1)
('Student', 'Darkhold', 1)
('Student', 'The Alchemist', 1)
('Student', 'Algorithmic Hearts', 1)
('Student', 'The Awakening', 1)
('Student', 'Hatchet', 1)
('Student', 'The Maze Runner', 1)
('Senior Citizen', 'Lawn Boy', 1)
('Senior Citizen', 'The Alchemist', 1)
('Senior Citizen', "Harry Potter and the Sorcerer's Stone", 1)
('Senior Citizen', 'The Dark Tower', 1)
('Senior Citizen', 'Alias Grace', 1)
('Other', 'Wonder', 1)
('Other', 'Darkhold', 1)
('Other', 'Hatchet', 1)
SQL command executed successfully.
>>
```

14. Never late returns

Find out which clients have never returned an item late.



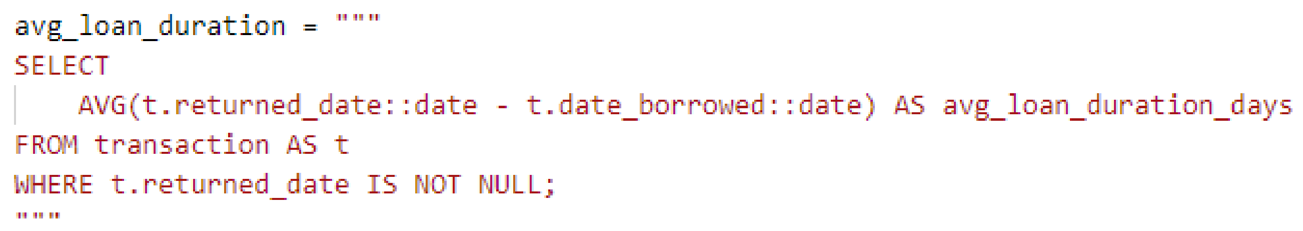
```
never_late_clients = """
SELECT DISTINCT
    c.client_id,
    c.name
FROM client AS c
WHERE NOT EXISTS (
    SELECT 1
    FROM transaction AS t
    WHERE t.client_id = c.client_id
      AND t.returned_date > t.expected_return_date
);
"""
```



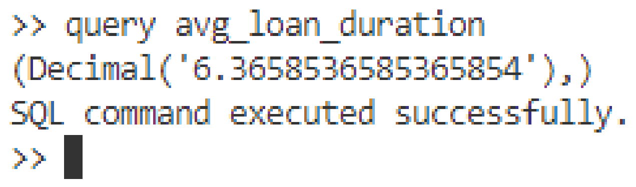
```
>> query never_late_clients
(17, 'Kayleigh Amstutz')
(25, 'Sawyer Smith')
(2, 'Ava Morgan')
(31, 'Jessie Pinkman')
(32, 'Devin Smith')
(39, 'Lawrence Outh')
(21, 'Wanda Maximoff')
(24, 'Riley Monroe')
(28, 'Brooke Winslow')
(40, 'Priyash Gupta')
(37, 'Laura Smith')
(22, 'Harper Ellison')
(6, 'Zoe Kim')
(38, 'Ankus Gray')
(16, 'Charlotte Aitchison')
(19, 'Jace Beleren')
(12, 'Daniel Rider')
(23, 'Grayson Tate')
(4, 'Maya Patel')
(1, 'Liam Carter')
(33, 'Anabelle Wahl')
(5, 'Ethan Reynolds')
(27, 'Ian Francis')
(34, 'Alexis Vanderpool')
(43, 'Kaleb Jones')
(11, 'Chris Pratt')
(29, 'Saul Goodman')
(9, 'Owen Hayes')
(41, 'Mike Prince')
(3, 'Noah Sinclair')
(30, 'Walter White')
(14, 'Jack Black')
(42, 'Kingston Mattson')
(13, 'Archie Ryan')
SQL command executed successfully.
>> █
```

15. Average loan duration

Calculate the average time an item stays on loan before being returned.



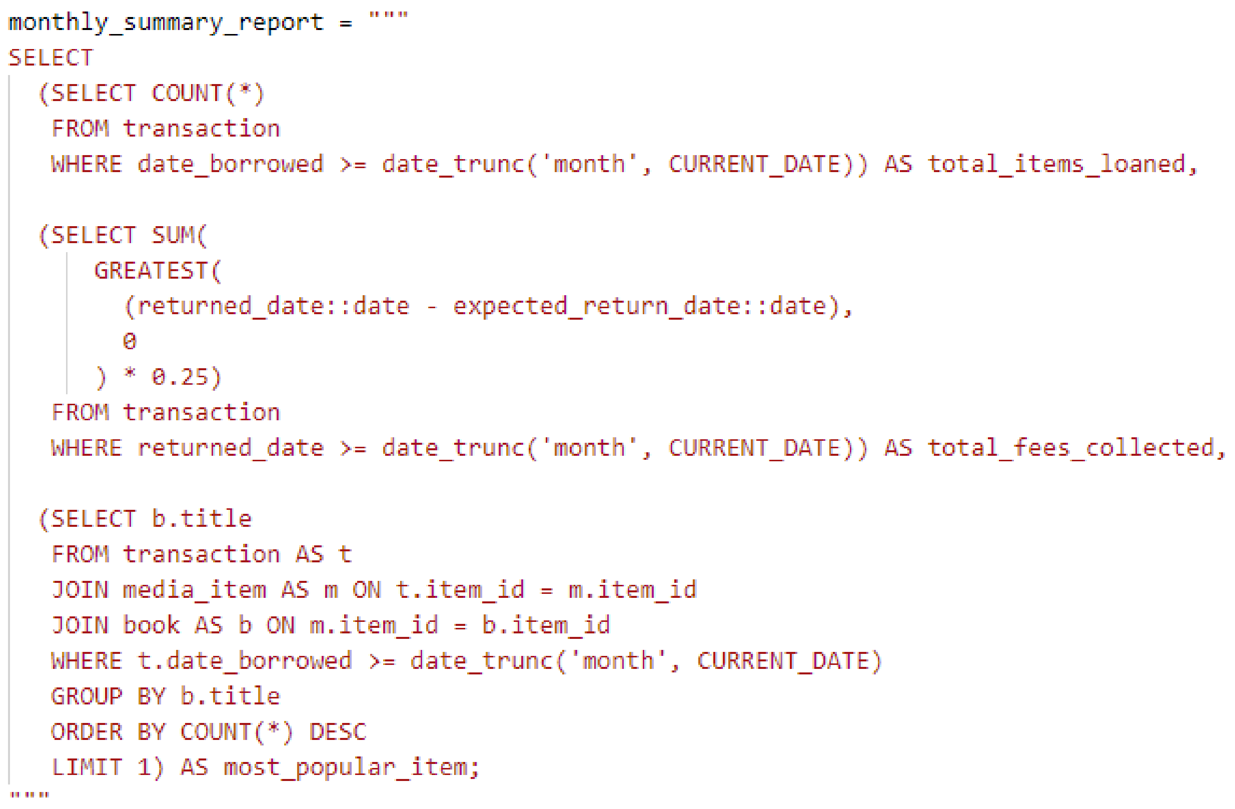
```
avg_loan_duration = """
SELECT
    AVG(t.returned_date::date - t.date_borrowed::date) AS avg_loan_duration_days
FROM transaction AS t
WHERE t.returned_date IS NOT NULL;
"""
```



```
>> query avg_loan_duration
(Decimal('6.3658536585365854'),)
SQL command executed successfully.
>>
```

16. Monthly summary report

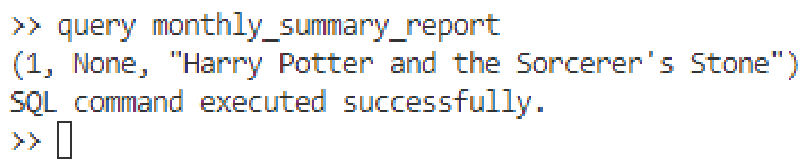
Generate a report summarizing the total number of items loaned, total fees collected, and most popular items for the month.



```
monthly_summary_report = """
SELECT
  (SELECT COUNT(*)
   FROM transaction
   WHERE date_borrowed >= date_trunc('month', CURRENT_DATE)) AS total_items_loaned,

  (SELECT SUM(
      GREATEST(
        (returned_date::date - expected_return_date::date),
        0
      ) * 0.25)
   FROM transaction
   WHERE returned_date >= date_trunc('month', CURRENT_DATE)) AS total_fees_collected,

  (SELECT b.title
   FROM transaction AS t
   JOIN media_item AS m ON t.item_id = m.item_id
   JOIN book AS b ON m.item_id = b.item_id
   WHERE t.date_borrowed >= date_trunc('month', CURRENT_DATE)
   GROUP BY b.title
   ORDER BY COUNT(*) DESC
   LIMIT 1) AS most_popular_item;
"""
```



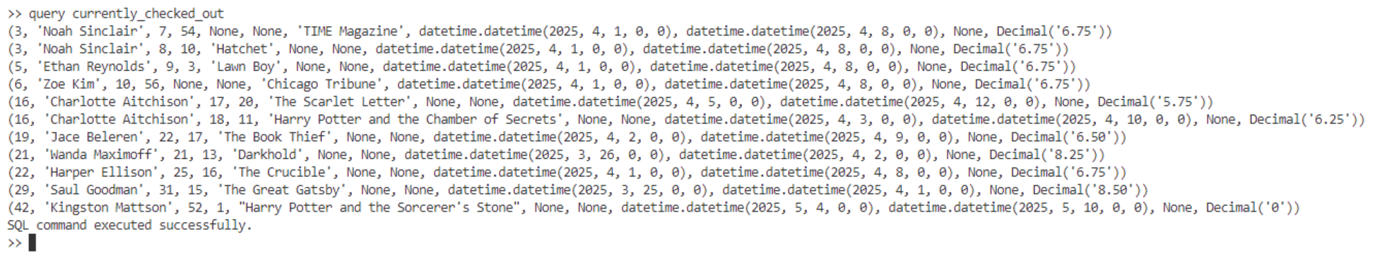
```
>> query monthly_summary_report
(1, None, "Harry Potter and the Sorcerer's Stone")
SQL command executed successfully.
>>
```

Note: This only has one item since there is only a single transaction in our database that has a book checked out in the past month. This is confirmed as only 1 book has been borrowed (verified by the results of the books due soon query), no fees have been collected (verified by the results from the monthly fees report query), and “Harry Potter and the Sorcerer’s Stone” is the most popular book (verified by the results of the most popular item in the last month query).

17. Currently checked out items

Returns client,

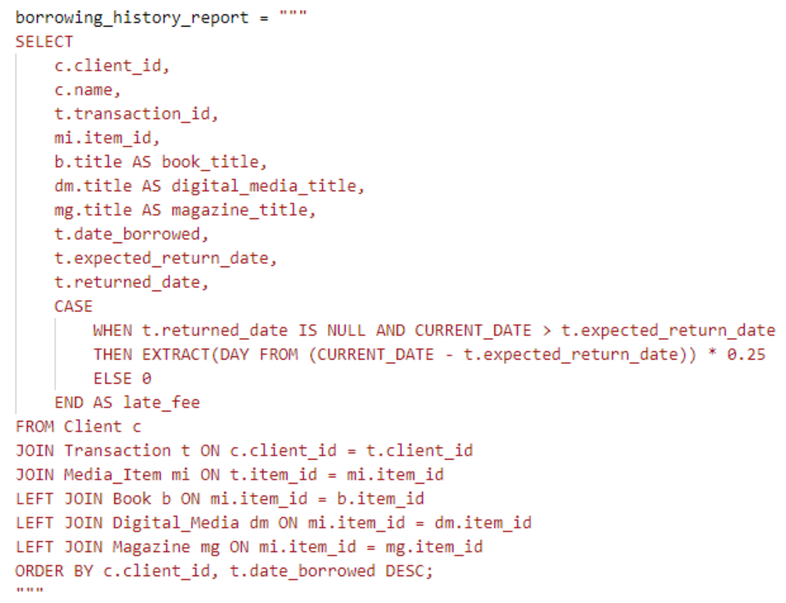
```
currently_checked_out = """
SELECT DISTINCT
    c.client_id,
    c.name,
    t.transaction_id,
    mi.item_id,
    b.title AS book_title,
    dm.title AS digital_media_title,
    mg.title AS magazine_title,
    t.date_borrowed,
    t.expected_return_date,
    t.returned_date,
    CASE
        WHEN t.returned_date IS NULL AND CURRENT_DATE > t.expected_return_date
        THEN EXTRACT(DAY FROM (CURRENT_DATE - t.expected_return_date)) * 0.25
        ELSE 0
    END AS late_fee
FROM Client c
JOIN Transaction t ON c.client_id = t.client_id
JOIN Media_Item mi ON t.item_id = mi.item_id
LEFT JOIN Book b ON mi.item_id = b.item_id
LEFT JOIN Digital_Media dm ON mi.item_id = dm.item_id
LEFT JOIN Magazine mg ON mi.item_id = mg.item_id
WHERE t.returned_date IS NULL
ORDER BY c.client_id, t.date_borrowed DESC;
"""
```



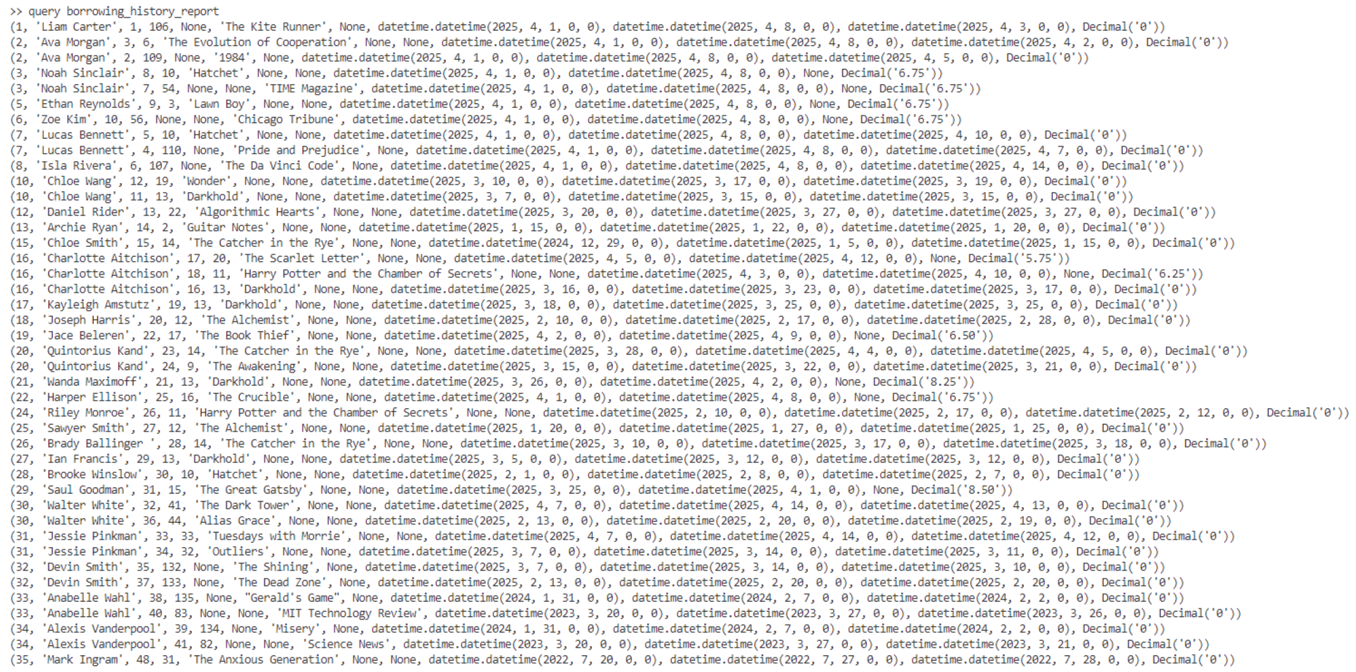
```
>> query currently_checked_out
(3, 'Noah Sinclair', 7, 54, None, None, 'TIME Magazine', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(3, 'Noah Sinclair', 8, 10, 'Hatchet', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(5, 'Ethan Reynolds', 9, 3, 'Lawn Boy', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(6, 'Zoe Kim', 10, 56, None, None, 'Chicago Tribune', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(16, 'Charlotte Aitchison', 17, 20, 'The Scarlet Letter', None, None, datetime.datetime(2025, 4, 5, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), None, Decimal('5.75'))
(16, 'Charlotte Aitchison', 18, 11, 'Harry Potter and the Chamber of Secrets', None, None, datetime.datetime(2025, 4, 3, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), None, Decimal('6.25'))
(19, 'Jace Beleren', 22, 17, 'The Book Thief', None, None, datetime.datetime(2025, 4, 2, 0, 0), datetime.datetime(2025, 4, 9, 0, 0), None, Decimal('6.50'))
(21, 'Wanda Maximoff', 21, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 26, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), None, Decimal('8.25'))
(22, 'Harper Ellison', 25, 16, 'The Crucible', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(29, 'Saul Goodman', 31, 15, 'The Great Gatsby', None, None, datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 4, 1, 0, 0), None, Decimal('8.50'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
SQL command executed successfully.
>> 
```

18. Client borrowing report

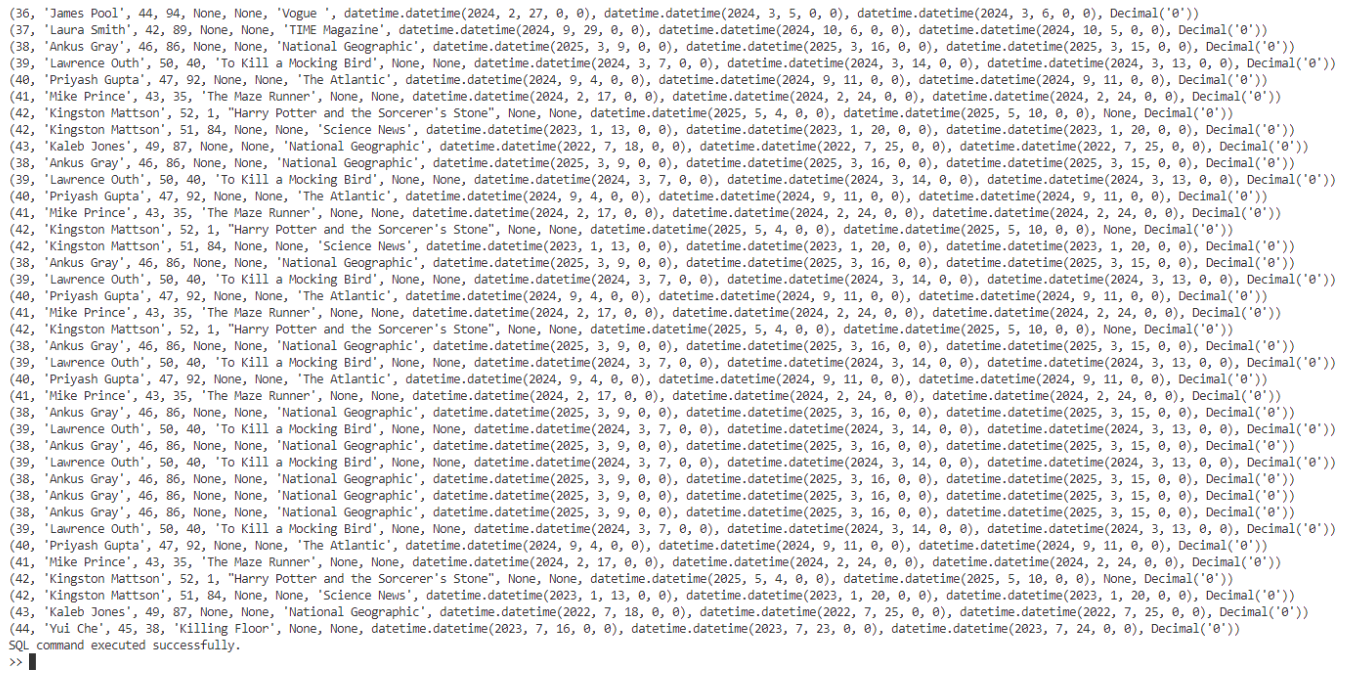
Produce an individual report for each client showing their borrowing history, outstanding fees, and any reserved items.



```
borrowing_history_report = """
SELECT
    c.client_id,
    c.name,
    t.transaction_id,
    mi.item_id,
    b.title AS book_title,
    dm.title AS digital_media_title,
    mg.title AS magazine_title,
    t.date_borrowed,
    t.expected_return_date,
    t.returned_date,
    CASE
        WHEN t.returned_date IS NULL AND CURRENT_DATE > t.expected_return_date
        THEN EXTRACT(DAY FROM (CURRENT_DATE - t.expected_return_date)) * 0.25
        ELSE 0
    END AS late_fee
FROM Client c
JOIN Transaction t ON c.client_id = t.client_id
JOIN Media_Item mi ON t.item_id = mi.item_id
LEFT JOIN Book b ON mi.item_id = b.item_id
LEFT JOIN Digital_Media dm ON mi.item_id = dm.item_id
LEFT JOIN Magazine mg ON mi.item_id = mg.item_id
ORDER BY c.client_id, t.date_borrowed DESC;
"""
```



```
>> query borrowing_history_report
(1, 'Liam Carter', 1, 106, None, 'The Kite Runner', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 3, 0, 0), Decimal('0'))
(2, 'Ava Morgan', 3, 6, 'The Evolution of Cooperation', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), Decimal('0'))
(2, 'Ava Morgan', 2, 109, None, '1984', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 5, 0, 0), Decimal('0'))
(3, 'Noah Sinclair', 8, 10, 'Hatchet', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(3, 'Noah Sinclair', 7, 54, None, None, 'TIME Magazine', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(5, 'Ethan Reynolds', 9, 3, 'Lawn Boy', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(6, 'Zoe Kim', 10, 56, None, None, 'Chicago Tribune', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(7, 'Lucas Bennett', 5, 10, 'Hatchet', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), Decimal('0'))
(7, 'Lucas Bennett', 4, 110, None, 'Pride and Prejudice', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 7, 0, 0), Decimal('0'))
(8, 'Isla Rivera', 6, 107, None, 'The Da Vinci Code', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), Decimal('0'))
(10, 'Chloe Wang', 12, 19, 'Wonder', None, None, datetime.datetime(2025, 3, 10, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), datetime.datetime(2025, 3, 19, 0, 0), Decimal('0'))
(10, 'Chloe Wang', 11, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(12, 'Daniel Rider', 13, 22, 'Algorithmic Hearts', None, None, datetime.datetime(2025, 3, 20, 0, 0), datetime.datetime(2025, 3, 27, 0, 0), datetime.datetime(2025, 3, 27, 0, 0), Decimal('0'))
(13, 'Archie Ryan', 14, 2, 'Guitar Notes', None, None, datetime.datetime(2025, 1, 15, 0, 0), datetime.datetime(2025, 1, 22, 0, 0), datetime.datetime(2025, 1, 20, 0, 0), Decimal('0'))
(15, 'Chloe Smith', 15, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2024, 12, 29, 0, 0), datetime.datetime(2025, 1, 5, 0, 0), datetime.datetime(2025, 1, 15, 0, 0), Decimal('0'))
(16, 'Charlotte Aitchison', 17, 20, 'The Scarlet Letter', None, None, datetime.datetime(2025, 4, 5, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), None, Decimal('5.75'))
(16, 'Charlotte Aitchison', 18, 11, 'Harry Potter and the Chamber of Secrets', None, None, datetime.datetime(2025, 4, 3, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), None, Decimal('6.25'))
(16, 'Charlotte Aitchison', 16, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 23, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), Decimal('0'))
(17, 'Kayleigh Amstutz', 19, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 18, 0, 0), datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 3, 25, 0, 0), Decimal('0'))
(18, 'Joseph Harris', 20, 12, 'The Alchemist', None, None, datetime.datetime(2025, 2, 10, 0, 0), datetime.datetime(2025, 2, 17, 0, 0), datetime.datetime(2025, 2, 28, 0, 0), Decimal('0'))
(19, 'Jace Beleren', 22, 17, 'The Book Thief', None, None, datetime.datetime(2025, 4, 2, 0, 0), datetime.datetime(2025, 4, 9, 0, 0), None, Decimal('6.50'))
(20, 'Quintorius Kand', 23, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2025, 3, 28, 0, 0), datetime.datetime(2025, 4, 4, 0, 0), datetime.datetime(2025, 4, 5, 0, 0), Decimal('0'))
(20, 'Quintorius Kand', 24, 9, 'The Awakening', None, None, datetime.datetime(2025, 3, 15, 0, 0), datetime.datetime(2025, 3, 22, 0, 0), datetime.datetime(2025, 3, 21, 0, 0), Decimal('0'))
(21, 'Wanda Maximoff', 21, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 26, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), None, Decimal('8.25'))
(22, 'Harper Ellison', 25, 16, 'The Crucible', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(24, 'Riley Monroe', 26, 11, 'Harry Potter and the Chamber of Secrets', None, None, datetime.datetime(2025, 2, 10, 0, 0), datetime.datetime(2025, 2, 17, 0, 0), datetime.datetime(2025, 2, 12, 0, 0), Decimal('0'))
(25, 'Sawyer Smith', 27, 12, 'The Alchemist', None, None, datetime.datetime(2025, 1, 20, 0, 0), datetime.datetime(2025, 1, 27, 0, 0), datetime.datetime(2025, 1, 25, 0, 0), Decimal('0'))
(26, 'Brady Ballinger ', 28, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2025, 3, 10, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), datetime.datetime(2025, 3, 18, 0, 0), Decimal('0'))
(27, 'Ian Francis', 29, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 5, 0, 0), datetime.datetime(2025, 3, 12, 0, 0), datetime.datetime(2025, 3, 12, 0, 0), Decimal('0'))
(28, 'Brooke Winslow', 30, 10, 'Hatchet', None, None, datetime.datetime(2025, 2, 1, 0, 0), datetime.datetime(2025, 2, 8, 0, 0), datetime.datetime(2025, 2, 7, 0, 0), Decimal('0'))
(29, 'Saul Goodman', 31, 15, 'The Great Gatsby', None, None, datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 4, 1, 0, 0), None, Decimal('8.50'))
(30, 'Walter White', 32, 41, 'The Dark Tower', None, None, datetime.datetime(2025, 4, 7, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), datetime.datetime(2025, 4, 13, 0, 0), Decimal('0'))
(30, 'Walter White', 36, 44, 'Alias Grace', None, None, datetime.datetime(2025, 2, 13, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), datetime.datetime(2025, 2, 19, 0, 0), Decimal('0'))
(31, 'Jessie Pinkman', 33, 33, 'Tuesdays with Morrie', None, None, datetime.datetime(2025, 4, 7, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), Decimal('0'))
(31, 'Jessie Pinkman', 34, 32, 'Outliers', None, None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 14, 0, 0), datetime.datetime(2025, 3, 11, 0, 0), Decimal('0'))
(32, 'Devin Smith', 35, 132, None, 'The Shining', None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 14, 0, 0), datetime.datetime(2025, 3, 10, 0, 0), Decimal('0'))
(32, 'Devin Smith', 37, 133, None, 'The Dead Zone', None, datetime.datetime(2025, 2, 13, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), Decimal('0'))
(33, 'Anabelle Wahl', 38, 135, None, "Gerald's Game", None, datetime.datetime(2024, 1, 31, 0, 0), datetime.datetime(2024, 2, 7, 0, 0), datetime.datetime(2024, 2, 2, 0, 0), Decimal('0'))
(33, 'Anabelle Wahl', 40, 83, None, None, 'MIT Technology Review', datetime.datetime(2023, 3, 20, 0, 0), datetime.datetime(2023, 3, 27, 0, 0), datetime.datetime(2023, 3, 26, 0, 0), Decimal('0'))
(34, 'Alexis Vanderpool', 39, 134, None, 'Misery', None, datetime.datetime(2024, 1, 31, 0, 0), datetime.datetime(2024, 2, 7, 0, 0), datetime.datetime(2024, 2, 2, 0, 0), Decimal('0'))
(34, 'Alexis Vanderpool', 41, 82, None, None, 'Science News', datetime.datetime(2023, 3, 20, 0, 0), datetime.datetime(2023, 3, 27, 0, 0), datetime.datetime(2023, 3, 21, 0, 0), Decimal('0'))
(35, 'Mark Ingram', 48, 31, 'The Anxious Generation', None, None, datetime.datetime(2022, 7, 20, 0, 0), datetime.datetime(2022, 7, 27, 0, 0), datetime.datetime(2022, 7, 28, 0, 0), Decimal('0'))
```

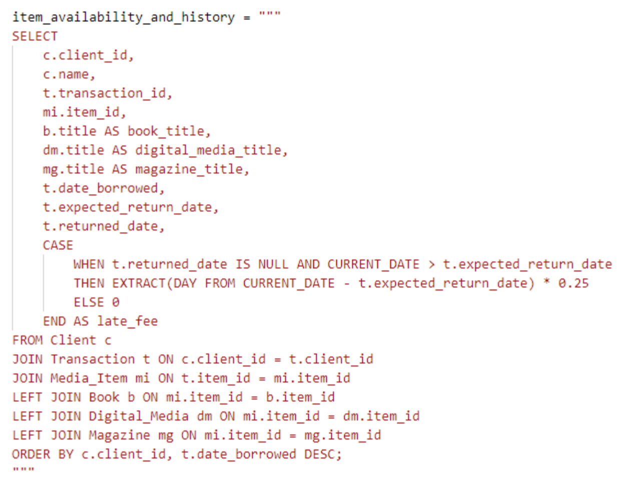


```
(36, 'James Pool', 44, 94, None, None, 'Vogue ', datetime.datetime(2024, 2, 27, 0, 0), datetime.datetime(2024, 3, 5, 0, 0), datetime.datetime(2024, 3, 6, 0, 0), Decimal('0'))
(37, 'Laura Smith', 42, 89, None, None, 'TIME Magazine', datetime.datetime(2024, 9, 29, 0, 0), datetime.datetime(2024, 10, 6, 0, 0), datetime.datetime(2024, 10, 5, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
(42, 'Kingston Mattson', 51, 84, None, None, 'Science News', datetime.datetime(2023, 1, 13, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), Decimal('0'))
(43, 'Kaleb Jones', 49, 87, None, None, 'National Geographic', datetime.datetime(2022, 7, 18, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
(42, 'Kingston Mattson', 51, 84, None, None, 'Science News', datetime.datetime(2023, 1, 13, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0'))
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
(42, 'Kingston Mattson', 51, 84, None, None, 'Science News', datetime.datetime(2023, 1, 13, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), Decimal('0'))
(43, 'Kaleb Jones', 49, 87, None, None, 'National Geographic', datetime.datetime(2022, 7, 18, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), Decimal('0'))
(44, 'Yui Che', 45, 38, 'Killing Floor', None, None, datetime.datetime(2023, 7, 16, 0, 0), datetime.datetime(2023, 7, 23, 0, 0), datetime.datetime(2023, 7, 24, 0, 0), Decimal('0'))
SQL command executed successfully.
>> 
```

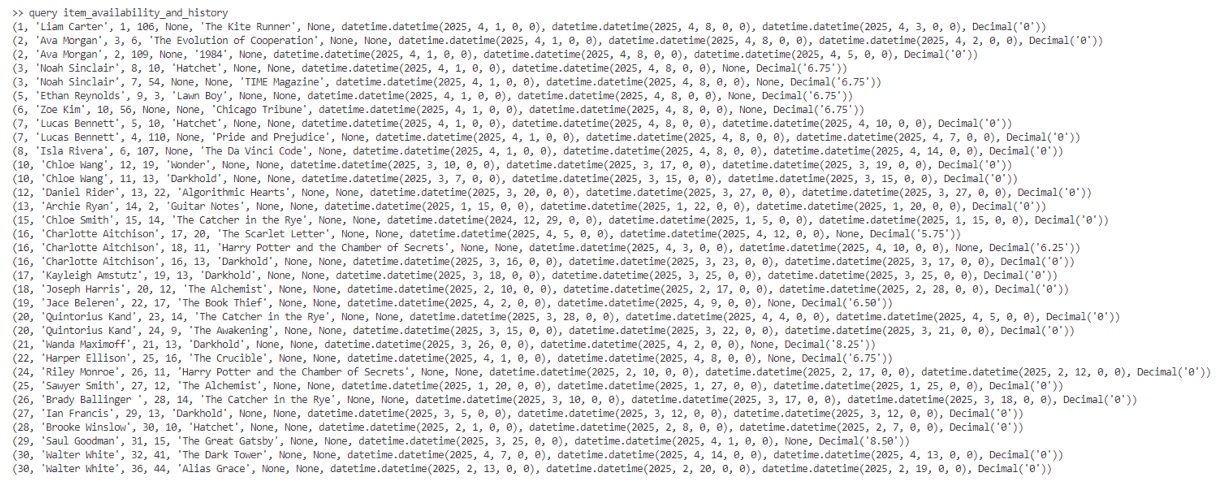
Note: The result is in 2 images since the query results are too large to screenshot in a single image. There is a single tuple returned for each item that has been checked out. The amount shown for each client match the number of transactions returned in the member engagement report.

19. Item availability and history

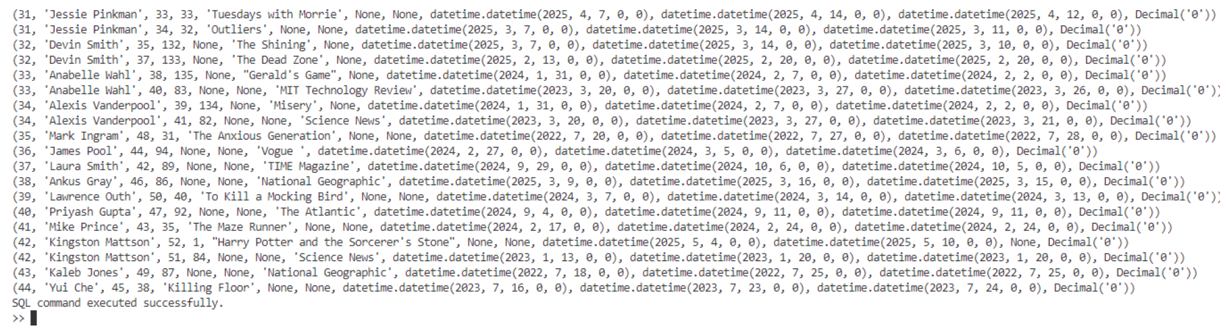
List all items, their current availability status, and their last borrowed date. Highlight items that have not been borrowed in the past six months.



```
item_availability_and_history = """
SELECT
    c.client_id,
    c.name,
    t.transaction_id,
    mi.item_id,
    b.title AS book_title,
    dm.title AS digital_media_title,
    mg.title AS magazine_title,
    t.date_borrowed,
    t.expected_return_date,
    t.returned_date,
    CASE
        WHEN t.returned_date IS NULL AND CURRENT_DATE > t.expected_return_date
        THEN EXTRACT(DAY FROM CURRENT_DATE - t.expected_return_date) * 0.25
        ELSE 0
    END AS late_fee
FROM Client c
JOIN Transaction t ON c.client_id = t.client_id
JOIN Media_Item mi ON t.item_id = mi.item_id
LEFT JOIN Book b ON mi.item_id = b.item_id
LEFT JOIN Digital_Media dm ON mi.item_id = dm.item_id
LEFT JOIN Magazine mg ON mi.item_id = mg.item_id
ORDER BY c.client_id, t.date_borrowed DESC;
"""
```



```
>> query item_availability_and_history
(1, 'Liam Carter', 1, 106, None, 'The Kite Runner', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 3, 0, 0), Decimal('0'))
(2, 'Ava Morgan', 3, 6, 'The Evolution of Cooperation', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), Decimal('0'))
(2, 'Ava Morgan', 2, 109, None, '1984', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 5, 0, 0), Decimal('0'))
(3, 'Noah Sinclair', 8, 10, 'Hatchet', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(3, 'Noah Sinclair', 7, 54, None, None, 'TIME Magazine', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(5, 'Ethan Reynolds', 9, 3, 'Lawn Boy', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(6, 'Zoe Kim', 10, 56, None, None, 'Chicago Tribune', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(7, 'Lucas Bennett', 5, 10, 'Hatchet', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), Decimal('0'))
(7, 'Lucas Bennett', 4, 110, None, 'Pride and Prejudice', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 7, 0, 0), Decimal('0'))
(8, 'Isla Rivera', 6, 107, None, 'The Da Vinci Code', None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), Decimal('0'))
(10, 'Chloe Wang', 12, 19, 'Wonder', None, None, datetime.datetime(2025, 3, 10, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), datetime.datetime(2025, 3, 19, 0, 0), Decimal('0'))
(10, 'Chloe Wang', 11, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(12, 'Daniel Rider', 13, 22, 'Algorithmic Hearts', None, None, datetime.datetime(2025, 3, 20, 0, 0), datetime.datetime(2025, 3, 27, 0, 0), datetime.datetime(2025, 3, 27, 0, 0), Decimal('0'))
(13, 'Archie Ryan', 14, 2, 'Guitar Notes', None, None, datetime.datetime(2025, 1, 15, 0, 0), datetime.datetime(2025, 1, 22, 0, 0), datetime.datetime(2025, 1, 20, 0, 0), Decimal('0'))
(15, 'Chloe Smith', 15, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2024, 12, 29, 0, 0), datetime.datetime(2025, 1, 5, 0, 0), datetime.datetime(2025, 1, 15, 0, 0), Decimal('0'))
(16, 'Charlotte Aitchison', 17, 20, 'The Scarlet Letter', None, None, datetime.datetime(2025, 4, 5, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), None, Decimal('5.75'))
(16, 'Charlotte Aitchison', 18, 11, 'Harry Potter and the Chamber of Secrets', None, None, datetime.datetime(2025, 4, 3, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), None, Decimal('6.25'))
(16, 'Charlotte Aitchison', 16, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 23, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), Decimal('0'))
(17, 'Kayleigh Amstutz', 19, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 18, 0, 0), datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 3, 25, 0, 0), Decimal('0'))
(18, 'Joseph Harris', 20, 12, 'The Alchemist', None, None, datetime.datetime(2025, 2, 10, 0, 0), datetime.datetime(2025, 2, 17, 0, 0), datetime.datetime(2025, 2, 28, 0, 0), Decimal('0'))
(19, 'Jace Beleren', 22, 17, 'The Book Thief', None, None, datetime.datetime(2025, 4, 2, 0, 0), datetime.datetime(2025, 4, 9, 0, 0), None, Decimal('6.50'))
(20, 'Quintorius Kand', 23, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2025, 3, 28, 0, 0), datetime.datetime(2025, 4, 4, 0, 0), datetime.datetime(2025, 4, 5, 0, 0), Decimal('0'))
(20, 'Quintorius Kand', 24, 9, 'The Awakening', None, None, datetime.datetime(2025, 3, 15, 0, 0), datetime.datetime(2025, 3, 22, 0, 0), datetime.datetime(2025, 3, 21, 0, 0), Decimal('0'))
(21, 'Wanda Maximoff', 21, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 26, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), None, Decimal('8.25'))
(22, 'Harper Ellison', 25, 16, 'The Crucible', None, None, datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), None, Decimal('6.75'))
(24, 'Riley Monroe', 26, 11, 'Harry Potter and the Chamber of Secrets', None, None, datetime.datetime(2025, 2, 10, 0, 0), datetime.datetime(2025, 2, 17, 0, 0), datetime.datetime(2025, 2, 12, 0, 0), Decimal('0'))
(25, 'Sawyer Smith', 27, 12, 'The Alchemist', None, None, datetime.datetime(2025, 1, 20, 0, 0), datetime.datetime(2025, 1, 27, 0, 0), datetime.datetime(2025, 1, 25, 0, 0), Decimal('0'))
(26, 'Brady Ballinger ', 28, 14, 'The Catcher in the Rye', None, None, datetime.datetime(2025, 3, 10, 0, 0), datetime.datetime(2025, 3, 17, 0, 0), datetime.datetime(2025, 3, 18, 0, 0), Decimal('0'))
(27, 'Ian Francis', 29, 13, 'Darkhold', None, None, datetime.datetime(2025, 3, 5, 0, 0), datetime.datetime(2025, 3, 12, 0, 0), datetime.datetime(2025, 3, 12, 0, 0), Decimal('0'))
(28, 'Brooke Winslow', 30, 10, 'Hatchet', None, None, datetime.datetime(2025, 2, 1, 0, 0), datetime.datetime(2025, 2, 8, 0, 0), datetime.datetime(2025, 2, 7, 0, 0), Decimal('0'))
(29, 'Saul Goodman', 31, 15, 'The Great Gatsby', None, None, datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 4, 1, 0, 0), None, Decimal('8.50'))
(30, 'Walter White', 32, 41, 'The Dark Tower', None, None, datetime.datetime(2025, 4, 7, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), datetime.datetime(2025, 4, 13, 0, 0), Decimal('0'))
(30, 'Walter White', 36, 44, 'Alias Grace', None, None, datetime.datetime(2025, 2, 13, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), datetime.datetime(2025, 2, 19, 0, 0), Decimal('0'))
```



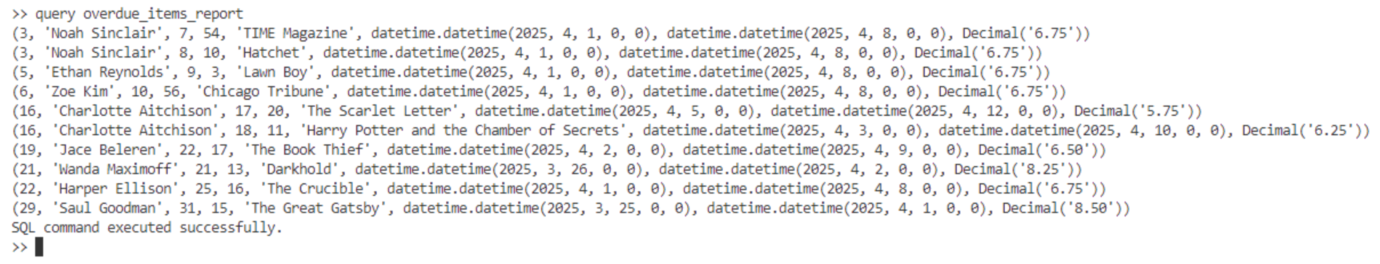
```
(31, 'Jessie Pinkman', 33, 33, 'Tuesdays with Morrie', None, None, datetime.datetime(2025, 4, 7, 0, 0), datetime.datetime(2025, 4, 14, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), Decimal('0'))
(31, 'Jessie Pinkman', 34, 32, 'Outliers', None, None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 14, 0, 0), datetime.datetime(2025, 3, 11, 0, 0), Decimal('0'))
(32, 'Devin Smith', 35, 132, None, 'The Shining', None, datetime.datetime(2025, 3, 7, 0, 0), datetime.datetime(2025, 3, 14, 0, 0), datetime.datetime(2025, 3, 10, 0, 0), Decimal('0'))
(32, 'Devin Smith', 37, 133, None, 'The Dead Zone', None, datetime.datetime(2025, 2, 13, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), datetime.datetime(2025, 2, 20, 0, 0), Decimal('0'))
(33, 'Anabelle Wahl', 38, 135, None, "Gerald's Game", None, datetime.datetime(2024, 1, 31, 0, 0), datetime.datetime(2024, 2, 7, 0, 0), datetime.datetime(2024, 2, 2, 0, 0), Decimal('0'))
(33, 'Anabelle Wahl', 40, 83, None, None, 'MIT Technology Review', datetime.datetime(2023, 3, 20, 0, 0), datetime.datetime(2023, 3, 27, 0, 0), datetime.datetime(2023, 3, 26, 0, 0), Decimal('0')
(34, 'Alexis Vanderpool', 39, 134, None, 'Misery', None, datetime.datetime(2024, 1, 31, 0, 0), datetime.datetime(2024, 2, 7, 0, 0), datetime.datetime(2024, 2, 2, 0, 0), Decimal('0'))
(34, 'Alexis Vanderpool', 41, 82, None, None, 'Science News', datetime.datetime(2023, 3, 20, 0, 0), datetime.datetime(2023, 3, 27, 0, 0), datetime.datetime(2023, 3, 21, 0, 0), Decimal('0'))
(35, 'Mark Ingram', 48, 31, 'The Anxious Generation', None, None, datetime.datetime(2022, 7, 20, 0, 0), datetime.datetime(2022, 7, 27, 0, 0), datetime.datetime(2022, 7, 28, 0, 0), Decimal('0'))
(36, 'James Pool', 44, 94, None, None, 'Vogue ', datetime.datetime(2024, 2, 27, 0, 0), datetime.datetime(2024, 3, 5, 0, 0), datetime.datetime(2024, 3, 6, 0, 0), Decimal('0'))
(37, 'Laura Smith', 42, 89, None, None, 'TIME Magazine', datetime.datetime(2024, 9, 29, 0, 0), datetime.datetime(2024, 10, 6, 0, 0), datetime.datetime(2024, 10, 5, 0, 0), Decimal('0'))
(38, 'Ankus Gray', 46, 86, None, None, 'National Geographic', datetime.datetime(2025, 3, 9, 0, 0), datetime.datetime(2025, 3, 16, 0, 0), datetime.datetime(2025, 3, 15, 0, 0), Decimal('0'))
(39, 'Lawrence Outh', 50, 40, 'To Kill a Mocking Bird', None, None, datetime.datetime(2024, 3, 7, 0, 0), datetime.datetime(2024, 3, 14, 0, 0), datetime.datetime(2024, 3, 13, 0, 0), Decimal('0')
(40, 'Priyash Gupta', 47, 92, None, None, 'The Atlantic', datetime.datetime(2024, 9, 4, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), datetime.datetime(2024, 9, 11, 0, 0), Decimal('0'))
(41, 'Mike Prince', 43, 35, 'The Maze Runner', None, None, datetime.datetime(2024, 2, 17, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), datetime.datetime(2024, 2, 24, 0, 0), Decimal('0'))
(42, 'Kingston Mattson', 52, 1, "Harry Potter and the Sorcerer's Stone", None, None, datetime.datetime(2025, 5, 4, 0, 0), datetime.datetime(2025, 5, 10, 0, 0), None, Decimal('0'))
(42, 'Kingston Mattson', 51, 84, None, None, 'Science News', datetime.datetime(2023, 1, 13, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), datetime.datetime(2023, 1, 20, 0, 0), Decimal('0'))
(43, 'Kaleb Jones', 49, 87, None, None, 'National Geographic', datetime.datetime(2022, 7, 18, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), datetime.datetime(2022, 7, 25, 0, 0), Decimal('0'))
(44, 'Yui Che', 45, 38, 'Killing Floor', None, None, datetime.datetime(2023, 7, 16, 0, 0), datetime.datetime(2023, 7, 23, 0, 0), datetime.datetime(2023, 7, 24, 0, 0), Decimal('0'))
SQL command executed successfully.
>>
```

Note: The result is in 2 images since the query results are too large to screenshot in a single image.

20. Overdue items report

Generate a report listing all overdue items, the client responsible, and the calculated late fees.

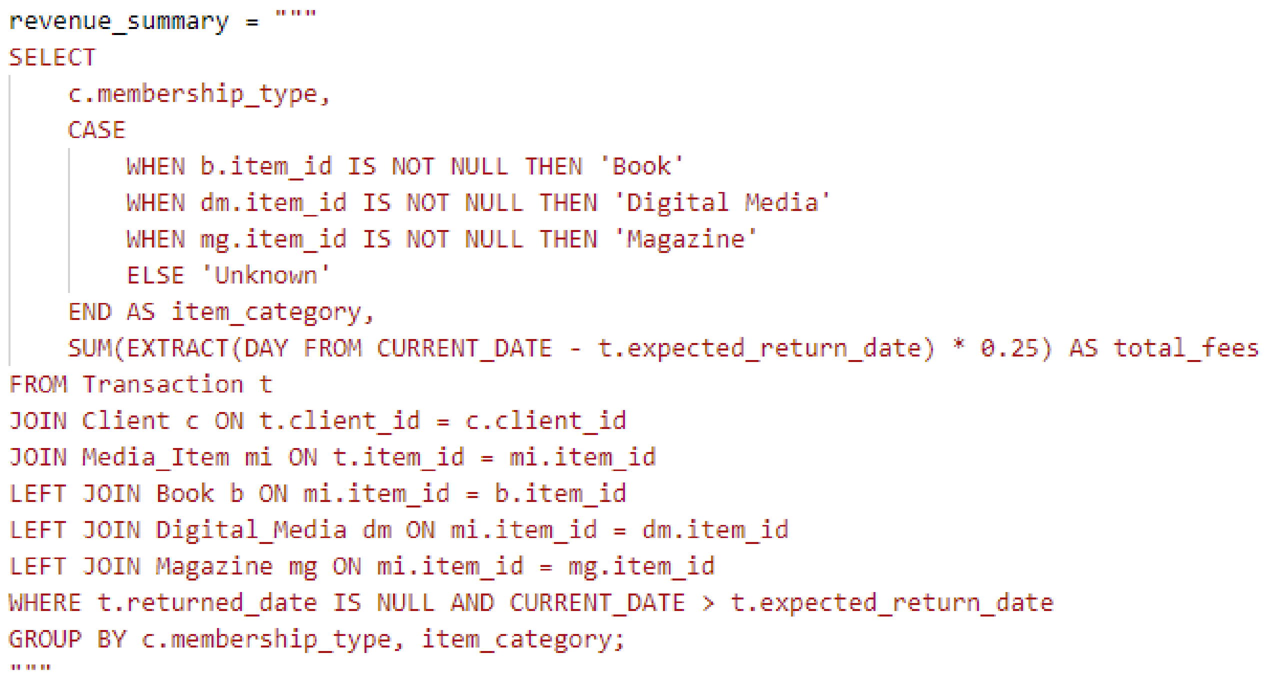
```
overdue_items_report = """
SELECT
    c.client_id,
    c.name,
    t.transaction_id,
    mi.item_id,
    CASE
        WHEN b.title IS NOT NULL THEN b.title
        WHEN dm.title IS NOT NULL THEN dm.title
        WHEN mg.title IS NOT NULL THEN mg.title
        ELSE 'Unknown'
    END AS title,
    t.date_borrowed,
    t.expected_return_date,
    EXTRACT(DAY FROM CURRENT_DATE - t.expected_return_date) * 0.25 AS late_fee
FROM Transaction t
JOIN Client c ON t.client_id = c.client_id
JOIN Media_Item mi ON t.item_id = mi.item_id
LEFT JOIN Book b ON mi.item_id = b.item_id
LEFT JOIN Digital_Media dm ON mi.item_id = dm.item_id
LEFT JOIN Magazine mg ON mi.item_id = mg.item_id
WHERE t.returned_date IS NULL AND CURRENT_DATE > t.expected_return_date;
"""
```



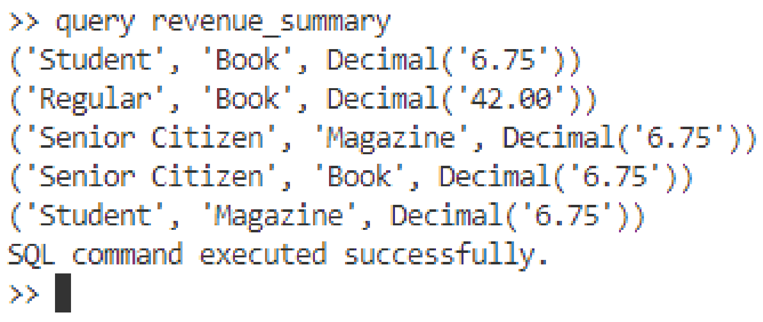
```
>> query overdue_items_report
(3, 'Noah Sinclair', 7, 54, 'TIME Magazine', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), Decimal('6.75'))
(3, 'Noah Sinclair', 8, 10, 'Hatchet', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), Decimal('6.75'))
(5, 'Ethan Reynolds', 9, 3, 'Lawn Boy', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), Decimal('6.75'))
(6, 'Zoe Kim', 10, 56, 'Chicago Tribune', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), Decimal('6.75'))
(16, 'Charlotte Aitchison', 17, 20, 'The Scarlet Letter', datetime.datetime(2025, 4, 5, 0, 0), datetime.datetime(2025, 4, 12, 0, 0), Decimal('5.75'))
(16, 'Charlotte Aitchison', 18, 11, 'Harry Potter and the Chamber of Secrets', datetime.datetime(2025, 4, 3, 0, 0), datetime.datetime(2025, 4, 10, 0, 0), Decimal('6.25'))
(19, 'Jace Beleren', 22, 17, 'The Book Thief', datetime.datetime(2025, 4, 2, 0, 0), datetime.datetime(2025, 4, 9, 0, 0), Decimal('6.50'))
(21, 'Wanda Maximoff', 21, 13, 'Darkhold', datetime.datetime(2025, 3, 26, 0, 0), datetime.datetime(2025, 4, 2, 0, 0), Decimal('8.25'))
(22, 'Harper Ellison', 25, 16, 'The Crucible', datetime.datetime(2025, 4, 1, 0, 0), datetime.datetime(2025, 4, 8, 0, 0), Decimal('6.75'))
(29, 'Saul Goodman', 31, 15, 'The Great Gatsby', datetime.datetime(2025, 3, 25, 0, 0), datetime.datetime(2025, 4, 1, 0, 0), Decimal('8.50'))
SQL command executed successfully.
>>
```

21. Revenue summary

Summarize the library's revenue from fees, showing the breakdown by membership type and item category.



```
revenue_summary = """
SELECT
    c.membership_type,
    CASE
        WHEN b.item_id IS NOT NULL THEN 'Book'
        WHEN dm.item_id IS NOT NULL THEN 'Digital Media'
        WHEN mg.item_id IS NOT NULL THEN 'Magazine'
        ELSE 'Unknown'
    END AS item_category,
    SUM(EXTRACT(DAY FROM CURRENT_DATE - t.expected_return_date) * 0.25) AS total_fees
FROM Transaction t
JOIN Client c ON t.client_id = c.client_id
JOIN Media_Item mi ON t.item_id = mi.item_id
LEFT JOIN Book b ON mi.item_id = b.item_id
LEFT JOIN Digital_Media dm ON mi.item_id = dm.item_id
LEFT JOIN Magazine mg ON mi.item_id = mg.item_id
WHERE t.returned_date IS NULL AND CURRENT_DATE > t.expected_return_date
GROUP BY c.membership_type, item_category;
"""
```



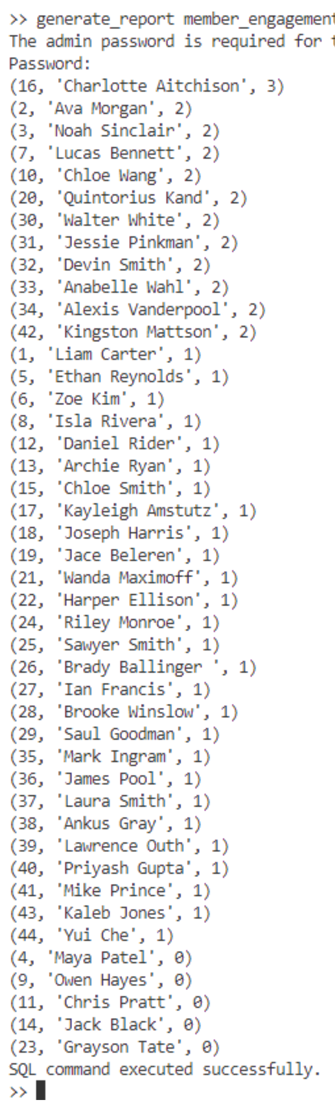
```
>> query revenue_summary
('Student', 'Book', Decimal('6.75'))
('Regular', 'Book', Decimal('42.00'))
('Senior Citizen', 'Magazine', Decimal('6.75'))
('Senior Citizen', 'Book', Decimal('6.75'))
('Student', 'Magazine', Decimal('6.75'))
SQL command executed successfully.
>>
```

Report:

Member Engagement Report

A report is generated which, for every client, returns the client's id, name, and number of transactions, sorted from the most transactions to the least transactions.

```
member_engagement_report = """
SELECT DISTINCT c.client_id, c.name, COUNT(t.transaction_id) AS total_transactions
FROM Client c
LEFT JOIN Transaction t ON c.client_id = t.client_id
GROUP BY c.client_id, c.name
ORDER BY total_transactions DESC;
"""
```



```
>> generate_report member_engagement
The admin password is required for that action.
Password:
(16, 'Charlotte Aitchison', 3)
(2, 'Ava Morgan', 2)
(3, 'Noah Sinclair', 2)
(7, 'Lucas Bennett', 2)
(10, 'Chloe Wang', 2)
(20, 'Quintorius Kand', 2)
(30, 'Walter White', 2)
(31, 'Jessie Pinkman', 2)
(32, 'Devin Smith', 2)
(33, 'Anabelle Wahl', 2)
(34, 'Alexis Vanderpool', 2)
(42, 'Kingston Mattson', 2)
(1, 'Liam Carter', 1)
(5, 'Ethan Reynolds', 1)
(6, 'Zoe Kim', 1)
(8, 'Isla Rivera', 1)
(12, 'Daniel Rider', 1)
(13, 'Archie Ryan', 1)
(15, 'Chloe Smith', 1)
(17, 'Kayleigh Amstutz', 1)
(18, 'Joseph Harris', 1)
(19, 'Jace Beleren', 1)
(21, 'Wanda Maximoff', 1)
(22, 'Harper Ellison', 1)
(24, 'Riley Monroe', 1)
(25, 'Sawyer Smith', 1)
(26, 'Brady Ballinger ', 1)
(27, 'Ian Francis', 1)
(28, 'Brooke Winslow', 1)
(29, 'Saul Goodman', 1)
(35, 'Mark Ingram', 1)
(36, 'James Pool', 1)
(37, 'Laura Smith', 1)
(38, 'Ankus Gray', 1)
(39, 'Lawrence Outh', 1)
(40, 'Priyash Gupta', 1)
(41, 'Mike Prince', 1)
(43, 'Kaleb Jones', 1)
(44, 'Yui Che', 1)
(4, 'Maya Patel', 0)
(9, 'Owen Hayes', 0)
(11, 'Chris Pratt', 0)
(14, 'Jack Black', 0)
(23, 'Grayson Tate', 0)
SQL command executed successfully.
>>
```