University of Mila
2nd undergraduate year
Academic year : 2025-2026

Department of Computer Science
Database Course
Teacher : F.Benabderrahmane

Practical Work 2 — SQL Query Language (SELECT)

Context

We consider again the Library database introduced in Practical Work 1, already created and populated. The objective of this session is to progressively master the SQL SELECT statement, from basic retrieval to advanced querying techniques.

Part 1 — Basic Queries : *Single-table queries with simple predicates*Retrieve all available books.

1. Retrieve books published between 1980 and 2024.
2. List members whose first name starts with 'A'.
3. List books whose category is 'Computer Science' or 'Novel'.
4. Retrieve books whose title contains the keyword 'Networks'.
5. List members who registered after March 1, 2024.
6. Retrieve books published after 2000.
7. List authors of Algerian nationality.

Part 2 — Intermediate Queries : Complex Selection Conditions and Join Operations

8. List books that are not available and belong to the category 'Novel'.
9. Retrieve members registered in 2024 whose phone number starts with '05'.
10. List books whose title contains 'DB' and whose publication year is greater than 2000.
11. Retrieve books that are available and not in the category 'History'.
12. Retrieve each book's title, author, and publication year.
13. List the names of members who have performed at least one loan.
14. Retrieve books borrowed by the member 'Sherif Aicha'.
15. List all reservations with the member's full name.
16. Retrieve books along with their author and category.
17. List all loans with member name, book title, and loan date.

Part 3 — Advanced Queries : Aggregate Functions, GROUP BY and HAVING, set operations, Subqueries

18. Count the total number of books.
19. Count members who registered in 2024.
20. Find the earliest publication year.
21. Compute the average publication year of Computer Science books.
22. Compute the number of available books per category.
23. For each category, display the number of books, only if this number is greater than 1.
24. List authors who have written at least two books.
25. List members who have made more than one loan.
26. List titles of books that have been borrowed OR reserved. *(UNION)*
27. List books that have been both borrowed AND reserved. *(INTERSECT)*
28. List books that have been borrowed but NOT reserved. *(EXCEPT)*
29. Retrieve books whose author is over 80 years old (assumed in 2024).
30. Retrieve books written by Algerian authors.
31. List members who have performed at least one loan.
32. Retrieve books that have been reserved.

33. Retrieve books whose publication year is greater than any book written by Victor Hugo.
34. Retrieve loans that occurred before any reservation.
35. Retrieve books published after all books in the 'History' category.
36. Retrieve loans whose return date is later than all loan dates.
37. Retrieve books published before all books in the 'Computer Science' category.
38. Retrieve books published in the same year as any book in the 'Novel' category.
39. Retrieve books whose publication year is different from all books in the 'Poetry' category.

Part 4 — Additional Advanced Queries

40. Retrieve titles of books written by French authors.
41. Retrieve books that have never been borrowed.
42. List members who have never made a reservation.
43. Retrieve books borrowed only by members registered in 2024.
44. Retrieve books written by the same author as 'The Little Prince'.
45. Retrieve books older than all books in the 'Science' category.
46. Retrieve books borrowed by members who have also made reservations.
47. Retrieve books that have been reserved but never borrowed.
48. Display all books sorted by publication year in descending order.
49. List members sorted by last name in ascending order, then by first name in ascending order.
50. Display loans sorted by loan date from the most recent to the oldest.
51. List books sorted first by category, then by title.
52. List members who have made at least one loan using EXISTS.
53. List books that have never been borrowed using NOT EXISTS.
54. List members who have made at least one reservation using EXISTS.
55. List authors who have no books recorded in the library using NOT EXISTS.
56. List books that have been borrowed at least once using EXISTS.
57. List members who have never made any reservation using NOT EXISTS.
58. List books whose publication year is greater than the average publication year of books in the same category.
59. List members whose number of loans is greater than the average number of loans per member.
60. List authors who have written more books than the average number of books per author.
61. Display books that have been borrowed more times than any other book in the same category.
62. List members who made a loan after their own registration date.
63. List books whose publication year is the oldest within their category.
64. Display members who have made a reservation for a book they also borrowed later.
65. List books that are borrowed by members who have never reserved any book.
66. Display authors for whom all recorded books have been borrowed at least once.
67. List categories in which at least one book has never been borrowed.

Solution

Part 1 — Basic Queries SOLUTIONS :

1) SELECT *

```
  FROM book
  WHERE publication_year BETWEEN 1980 AND 2024;
2) SELECT *
  FROM member
  WHERE first_name LIKE 'A%';

3) SELECT *
  FROM book
  WHERE category IN ('Computer Science', 'Novel');
4) SELECT *
  FROM book
  WHERE title ILIKE '%Networks%';

5) SELECT *
  FROM member
  WHERE registration_date > DATE '2024-03-01';

6) SELECT *
  FROM book
  WHERE publication_year > 2000;

7) SELECT *
  FROM author
  WHERE nationality = 'Algerian';
```

8) List books that are not available and belong to the category 'Novel'.

```
  SELECT *
  FROM book
  WHERE available = FALSE
    AND category = 'Novel';
```

9) Retrieve members registered in 2024 whose phone number starts with '05'.

```
 SELECT *
  FROM member
  WHERE EXTRACT(YEAR FROM registration_date) = 2024
    AND phone LIKE '05%';
```

10) List books whose title contains 'DB' and whose publication year is greater than 2000.

```
  SELECT *
  FROM book
  WHERE title ILIKE '%DB%'
```

```
AND publication_year > 2000;
```

11) Retrieve books that are available and not in the category 'History'.

```
SELECT *
FROM book
WHERE available = TRUE
AND category <> 'History';
```

12) Retrieve each book's title, author, and publication year.

```
SELECT b.title, a.first_name,  a.last_name, b.publication_year
FROM book b
JOIN author a ON b.author_id = a.author_id;
```

13) List the names of members who have performed at least one loan

Solution 1 — JOIN :

```
SELECT DISTINCT m.last_name, m.first_name
FROM member m
JOIN loan l ON m.member_id = l.member_id;
```

Solution 2 — Subquery :

```
SELECT last_name, first_name
FROM member
WHERE member_id IN (
                        SELECT member_id
                          FROM loan
                       );
```

14) Retrieve books borrowed by the member 'Sherif Aicha'

Solution 1 — JOIN :

```
SELECT b.*
FROM book b
JOIN loan l ON b.book_id = l.book_id
JOIN member m ON l.member_id = m.member_id
WHERE m.last_name = 'Sherif'
AND m.first_name = 'Aicha';
```

Solution 2 — Subquery :

```
SELECT *
FROM book
WHERE book_id IN (
   SELECT l.book_id
   FROM loan l
```

```
    JOIN member m ON l.member_id = m.member_id
    WHERE m.last_name = 'Sherif'
    AND m.first_name = 'Aicha'
);
```

15) List all reservations with the member's full name

```
SELECT r.reservation_id, m.last_name,  m.first_name,  r.book_id, r.reservation_date
FROM reservation r
JOIN member m ON r.member_id = m.member_id;
```

16) Retrieve books along with their author and category

```
SELECT b.title, b.category, a.first_name, a.last_name
FROM book b
JOIN author a ON b.author_id = a.author_id;
```

17) List all loans with member name, book title, and loan date

```
SELECT m.last_name, m.first_name,  b.title, l.loan_date
FROM loan l
JOIN member m ON l.member_id = m.member_id
JOIN book b ON l.book_id = b.book_id;
```

Part 3 — Advanced Queries :

18) Count the total number of books

```
SELECT COUNT(*) AS total_books
FROM book;
```

19) Count members who registered in 2024

```
SELECT COUNT(*) AS members_2024
FROM member
WHERE EXTRACT(YEAR FROM registration_date) = 2024;
```

20) Find the earliest publication year

```
SELECT MIN(publication_year) AS earliest_publication_year
FROM book;
```

21) Compute the average publication year of Computer Science books

```
SELECT AVG(publication_year) AS avg_publication_year
FROM book
WHERE category = 'Computer Science';
```

22) Compute the number of available books per category

```
SELECT category, COUNT(*) AS available_books
FROM book
WHERE available = TRUE
GROUP BY category;
```

23) For each category, display the number of books, only if this number is greater than 1

```
SELECT category, COUNT(*) AS nb_books
FROM book
GROUP BY category
HAVING COUNT(*) > 1;
```

24) List authors who have written at least two books

```
SELECT a.author_id, a.first_name, a.last_name, COUNT(*) AS nb_books
FROM author a
JOIN book b ON a.author_id = b.author_id
GROUP BY a.author_id, a.first_name, a.last_name
HAVING COUNT(*) >= 2;
```

25) List members who have made more than one loan

```
SELECT m.member_id, m.first_name, m.last_name, COUNT(*) AS nb_loans
FROM member m
JOIN loan l ON m.member_id = l.member_id
GROUP BY m.member_id, m.first_name, m.last_name
HAVING COUNT(*) > 1;
```

26) List titles of books that have been borrowed OR reserved (UNION)

```
SELECT b.title
FROM book b
JOIN loan l ON b.book_id = l.book_id
UNION
SELECT b.title
FROM book b
JOIN reservation r ON b.book_id = r.book_id;
```

27) List books that have been both borrowed AND reserved (INTERSECT)

```
SELECT b.title
FROM book b
JOIN loan l ON b.book_id = l.book_id
INTERSECT
SELECT b.title
```

```
FROM book b
JOIN reservation r ON b.book_id = r.book_id;
```

28) List books that have been borrowed but NOT reserved (EXCEPT)

```
SELECT b.title
FROM book b
JOIN loan l ON b.book_id = l.book_id
EXCEPT
SELECT b.title
FROM book b
JOIN reservation r ON b.book_id = r.book_id;
```

29) Retrieve books whose author is over 80 years old (assumed in 2024)

Solution 1 — with birth_date

```
SELECT b.*
FROM book b
JOIN author a ON b.author_id = a.author_id
WHERE DATE '2024-12-31' - a.birth_date >= INTERVAL '80 years';
```

Solution 2 — by birth year

```
SELECT b.*
FROM book b
JOIN author a ON b.author_id = a.author_id
WHERE EXTRACT(YEAR FROM AGE(DATE '2024-12-31', a.birth_date)) > 80;
```

30) Retrieve books written by Algerian authors

Solution 1 — JOIN

```
SELECT b.*
FROM book b
JOIN author a ON b.author_id = a.author_id
WHERE a.nationality = 'Algerian';
```

Solution 2 — Subquery

```
SELECT *
FROM book
WHERE author_id IN (
  SELECT author_id
  FROM author
  WHERE nationality = 'Algerian'
);
```

31) List members who have performed at least one loan

Solution 1 — IN

```
SELECT *
FROM member
WHERE member_id IN (
  SELECT member_id
  FROM loan
);
```

Solution 2 — EXISTS

```
SELECT *
FROM member m
WHERE EXISTS (
  SELECT 1
  FROM loan l
  WHERE l.member_id = m.member_id
);
```

32) Retrieve books that have been reserved

Solution 1 — JOIN

```
SELECT DISTINCT b.*
FROM book b
JOIN reservation r ON b.book_id = r.book_id;
```

Solution 2 — Subquery

```
SELECT *
FROM book
WHERE book_id IN (
  SELECT book_id
  FROM reservation
);
```

33. Retrieve books whose publication year is greater than any book written by Victor Hugo

```
SELECT *
FROM book
WHERE publication_year > ANY (
  SELECT b.publication_year
  FROM book b
  JOIN author a ON b.author_id = a.author_id
  WHERE a.first_name = 'Victor'
    AND a.last_name = 'Hugo'
);
```

34. Retrieve loans that occurred before any reservation

```
SELECT *
FROM loan
WHERE loan_date < ANY (
```

```
    SELECT reservation_date
    FROM reservation
);
```

35. Retrieve books published after all books in the 'History' category

```
SELECT *
FROM book
WHERE publication_year > ALL (
    SELECT publication_year
    FROM book
    WHERE category = 'History'
);
```

36. Retrieve loans whose return date is later than all loan dates

```
SELECT *
FROM loan
WHERE return_date > ALL (
    SELECT loan_date
    FROM loan
);
```

37. Retrieve books published before all books in the 'Computer Science' category

```
SELECT *
FROM book
WHERE publication_year < ALL (
    SELECT publication_year
    FROM book
    WHERE category = 'Computer Science'
);
```

38. Retrieve books published in the same year as any book in the 'Novel' category

```
SELECT *
FROM book
WHERE publication_year = ANY (
    SELECT publication_year
    FROM book
    WHERE category = 'Novel'
);
```

39. Retrieve books whose publication year is different from all books in the 'Poetry' category

```
SELECT *
FROM book
WHERE publication_year <> ALL (
    SELECT publication_year
    FROM book
```

```
    WHERE category = 'Poetry'
);
```

Part 4 — Additional Advanced Queries

40. Retrieve titles of books written by French authors

```
SELECT b.title
FROM book b
JOIN author a ON b.author_id = a.author_id
WHERE a.nationality = 'French';
```

41. Retrieve books that have never been borrowed

Solution 1 — NOT IN

```
SELECT *
FROM book
WHERE book_id NOT IN (
    SELECT book_id
    FROM loan
);
```

Solution 2 — NOT EXISTS

```
SELECT *
FROM book b
WHERE NOT EXISTS (
    SELECT 1
    FROM loan l
    WHERE l.book_id = b.book_id
);
```

42. List members who have never made a reservation

Solution 1 — NOT IN

```
SELECT *
FROM member
WHERE member_id NOT IN (
    SELECT member_id
    FROM reservation
);
```

Solution 2 — NOT EXISTS

```
SELECT *
FROM member m
WHERE NOT EXISTS (
    SELECT 1
    FROM reservation r
```

```
    WHERE r.member_id = m.member_id
);
```

43. Retrieve books borrowed only by members registered in 2024

```
SELECT b.*
FROM book b
WHERE EXISTS (
    SELECT 1
    FROM loan l
    WHERE l.book_id = b.book_id
)
AND NOT EXISTS (
    SELECT 1
    FROM loan l
    JOIN member m ON l.member_id = m.member_id
    WHERE l.book_id = b.book_id
      AND EXTRACT(YEAR FROM m.registration_date) <> 2024
);
```

44. Retrieve books written by the same author as 'The Little Prince'

```
SELECT *
FROM book
WHERE author_id = (
    SELECT author_id
    FROM book
    WHERE title = 'The Little Prince'
);
```

45. Retrieve books older than all books in the 'Science' category

```
SELECT *
FROM book
WHERE publication_year < ALL (
    SELECT publication_year
    FROM book
    WHERE category = 'Science'
);
```

46. Retrieve books borrowed by members who have also made reservations

```
SELECT DISTINCT b.*
FROM book b
JOIN loan l ON b.book_id = l.book_id
WHERE l.member_id IN (
    SELECT member_id
```

```
    FROM reservation
);
```

47. Retrieve books that have been reserved but never borrowed

Solution 1 — EXCEPT

```
SELECT b.*
FROM book b
JOIN reservation r ON b.book_id = r.book_id
EXCEPT
SELECT b.*
FROM book b
JOIN loan l ON b.book_id = l.book_id;
```

Solution 2 — EXISTS / NOT EXISTS

```
SELECT *
FROM book b
WHERE EXISTS (
   SELECT 1
   FROM reservation r
   WHERE r.book_id = b.book_id
)
AND NOT EXISTS (
   SELECT 1
   FROM loan l
   WHERE l.book_id = b.book_id
);
```

48. Display all books sorted by publication year in descending order

```
SELECT *
FROM book
ORDER BY publication_year DESC;
```

49. List members sorted by last name in ascending order, then by first name in ascending order

```
SELECT *
FROM member
ORDER BY last_name ASC, first_name ASC;
```

50. Display loans sorted by loan date from the most recent to the oldest

```
SELECT *
FROM loan
ORDER BY loan_date DESC;
```

51. List books sorted first by category, then by title

```
SELECT *
FROM book
ORDER BY category ASC, title ASC;
```

52. List members who have made at least one loan (using EXISTS)

```
SELECT *
FROM member m
WHERE EXISTS (
  SELECT 1
  FROM loan l
  WHERE l.member_id = m.member_id
);
```

53. List books that have never been borrowed using NOT EXISTS

```
SELECT *
FROM book b
WHERE NOT EXISTS (
  SELECT 1
  FROM loan l
  WHERE l.book_id = b.book_id
);
```

54. List members who have made at least one reservation using EXISTS

```
SELECT *
FROM member m
WHERE EXISTS (
  SELECT 1
  FROM reservation r
  WHERE r.member_id = m.member_id
);
```

55. List authors who have no books recorded in the library using NOT EXISTS

```
SELECT *
FROM author a
WHERE NOT EXISTS (
  SELECT 1
  FROM book b
  WHERE b.author_id = a.author_id
);
```

56. List books that have been borrowed at least once using EXISTS

```
SELECT *
FROM book b
```

```
WHERE EXISTS (
  SELECT 1
  FROM loan l
  WHERE l.book_id = b.book_id
);
```

57. List members who have never made any reservation using NOT EXISTS

```
SELECT *
FROM member m
WHERE NOT EXISTS (
  SELECT 1
  FROM reservation r
  WHERE r.member_id = m.member_id
);
```

58. List books whose publication year is greater than the average publication year of books in the same category

```
SELECT *
FROM book b1
WHERE publication_year > (
  SELECT AVG(b2.publication_year)
  FROM book b2
  WHERE b2.category = b1.category
);
```

59. List members whose number of loans is greater than the average number of loans per member

```
SELECT m.member_id, m.first_name, m.last_name, COUNT(l.loan_id) AS nb_loans
FROM member m
JOIN loan l ON m.member_id = l.member_id
GROUP BY m.member_id, m.first_name, m.last_name
HAVING COUNT(l.loan_id) > (
  SELECT AVG(nb_loans)
  FROM (
    SELECT COUNT(*) AS nb_loans
    FROM loan
    GROUP BY member_id
  ) t
);
```

60. List authors who have written more books than the average number of books per author

```
SELECT a.author_id, a.first_name, a.last_name, COUNT(b.book_id) AS nb_books
FROM author a
JOIN book b ON a.author_id = b.author_id
```

```
GROUP BY a.author_id, a.first_name, a.last_name
HAVING COUNT(b.book_id) > (
  SELECT AVG(nb_books)
  FROM (
    SELECT COUNT(*) AS nb_books
    FROM book
    GROUP BY author_id
  ) t
);
```

61. Display books that have been borrowed more times than any other book in the same category

```
SELECT b.book_id, b.title, b.category, COUNT(l.loan_id) AS nb_loans
FROM book b
JOIN loan l ON b.book_id = l.book_id
GROUP BY b.book_id, b.title, b.category
HAVING COUNT(l.loan_id) >= ALL (
  SELECT COUNT(l2.loan_id)
  FROM book b2
  JOIN loan l2 ON b2.book_id = l2.book_id
  WHERE b2.category = b.category
  GROUP BY b2.book_id
);
```

62. List members who made a loan after their own registration date

```
SELECT DISTINCT m.*
FROM member m
JOIN loan l ON m.member_id = l.member_id
WHERE l.loan_date > m.registration_date;
```

63. List books whose publication year is the oldest within their category

```
SELECT *
FROM book b
WHERE publication_year = (
  SELECT MIN(b2.publication_year)
  FROM book b2
  WHERE b2.category = b.category
);
```

64. Display members who have made a reservation for a book they also borrowed later

```
SELECT DISTINCT m.*
FROM member m
JOIN reservation r ON m.member_id = r.member_id
JOIN loan l ON m.member_id = l.member_id
```

```
        AND r.book_id = l.book_id
WHERE l.loan_date > r.reservation_date;
```

65. List books that are borrowed by members who have never reserved any book

```
SELECT DISTINCT b.*
FROM book b
JOIN loan l ON b.book_id = l.book_id
JOIN member m ON l.member_id = m.member_id
WHERE NOT EXISTS (
  SELECT 1
  FROM reservation r
  WHERE r.member_id = m.member_id
);
```

66. Display authors for whom all recorded books have been borrowed at least once

```
SELECT a.*
FROM author a
WHERE EXISTS (
  SELECT 1
  FROM book b
  WHERE b.author_id = a.author_id
)
AND NOT EXISTS (
  SELECT 1
  FROM book b
  WHERE b.author_id = a.author_id
   AND NOT EXISTS (
     SELECT 1
     FROM loan l
     WHERE l.book_id = b.book_id
   )
);
```

67. List categories in which at least one book has never been borrowed

```
SELECT DISTINCT b.category
FROM book b
WHERE NOT EXISTS (
  SELECT 1
  FROM loan l
  WHERE l.book_id = b.book_id
);
```