Atividade Prática – Consultas SQL (JOINs e Subconsultas) – Biblioteca Universitária

Objetivo: Executar consultas SQL para responder 20 questões situacionais, utilizando diferentes formas de JOINs e subconsultas. Ambiente sugerido: PostgreSQL no db-fiddle.com.

Diagrama do Banco de Dados:
https://bioparkedu.brightspace.com/content/enforced/10521-IA2502022/DER_Biblioteca_vetorial.pdf

Script do Banco de Dados:
https://bioparkedu.brightspace.com/content/enforced/10521-IA2502022/bd.sql

Sumário

1. Relatório de itens de empréstimo (aluno, título, data prevista, situação)

```
SELECT
  s.full_name        AS aluno,
  b.title            AS titulo_livro,
  li.expected_return_date AS data_prevista,
  CASE WHEN li.return_date IS NULL THEN 'Ativo' ELSE 'Devolvido' END AS situacao
FROM loan_item li
JOIN loan      l ON l.loan_id = li.loan_id
JOIN student   s ON s.student_id = l.student_id
JOIN exemplar  e ON e.exemplar_id = li.exemplar_id
JOIN book      b ON b.book_id = e.book_id
ORDER BY li.expected_return_date, s.full_name, b.title;
```

2. Empréstimos ativos por aluno (inclui quem tem zero)

```
SELECT
  s.student_id,
  s.full_name,
  COALESCE( COUNT(*) FILTER (WHERE li.return_date IS NULL), 0 ) AS emprestimos_ativos
FROM student s
LEFT JOIN loan      l  ON l.student_id = s.student_id
LEFT JOIN loan_item li ON li.loan_id    = l.loan_id
GROUP BY s.student_id, s.full_name
ORDER BY s.full_name;
```

3. Livros sem nenhum exemplar cadastrado

```
SELECT b.book_id, b.title
FROM book b
LEFT JOIN exemplar e ON e.book_id = b.book_id
WHERE e.exemplar_id IS NULL
ORDER BY b.title;
```

4. Livros e respectivos autores (não esconder livros sem autor)

```
SELECT
  b.book_id,
  b.title,
  a.author_id,
  a.name AS autor
FROM book b
LEFT JOIN book_author ba ON ba.book_id = b.book_id
LEFT JOIN author a       ON a.author_id = ba.author_id
ORDER BY b.title, a.name NULLS LAST;
```

5. Autores sem publicações + Livros sem autores

```
-- tipo: 'AUTOR_SEM_PUBLICACOES' ou 'LIVRO_SEM_AUTOR'
SELECT 'AUTOR_SEM_PUBLICACOES' AS tipo, a.author_id AS id, a.name AS nome_ou_titulo
FROM author a
LEFT JOIN book_author ba ON ba.author_id = a.author_id
WHERE ba.author_id IS NULL

UNION ALL

SELECT 'LIVRO_SEM_AUTOR' AS tipo, b.book_id AS id, b.title AS nome_ou_titulo
FROM book b
LEFT JOIN book_author ba ON ba.book_id = b.book_id
```

```
WHERE ba.book_id IS NULL

ORDER BY tipo, nome_ou_titulo;
```

6. Grade de combinações entre cursos e editoras

```
SELECT c.course_id, c.name AS curso, p.publisher_id, p.name AS editora
FROM course c
CROSS JOIN publisher p
ORDER BY c.name, p.name;
```

7. Pareamento de alunos do mesmo curso (sem repetições/espelhamentos)

```
WITH ranked AS (
  SELECT
    s.student_id,
    s.full_name,
    s.course_id,
    c.name AS curso,
    ROW_NUMBER() OVER (
      PARTITION BY s.course_id
      ORDER BY s.full_name, s.student_id
    ) AS rn
  FROM student s
  JOIN course c ON c.course_id = s.course_id
)
SELECT
  r1.curso,
  r1.student_id AS aluno1_id, r1.full_name AS aluno1,
  r2.student_id AS aluno2_id, r2.full_name AS aluno2
FROM ranked r1
LEFT JOIN ranked r2
       ON r2.course_id = r1.course_id
      AND r2.rn = r1.rn + 1
WHERE (r1.rn % 2) = 1
ORDER BY r1.curso, r1.full_name, r2.full_name NULLS LAST;
```

8. Aluno × Curso (quadro simples)

```
SELECT s.student_id, s.full_name AS aluno, c.name AS curso
FROM student s
JOIN course  c ON c.course_id = s.course_id
ORDER BY s.full_name;
```

9. Livros e seus autores (ordenado por título e autor)

```
SELECT b.title AS livro, a.name AS autor
FROM book b
LEFT JOIN book_author ba ON ba.book_id = b.book_id
LEFT JOIN author a       ON a.author_id = ba.author_id
ORDER BY b.title, a.name NULLS LAST;
```

10. Alunos que já pegaram 'Introduction to Algorithms (CLRS)'

```
SELECT DISTINCT s.student_id, s.full_name
FROM student s
JOIN loan      l  ON l.student_id = s.student_id
JOIN loan_item li ON li.loan_id    = l.loan_id
```

```
JOIN exemplar  e  ON e.exemplar_id = li.exemplar_id
JOIN book      b  ON b.book_id     = e.book_id
WHERE b.title LIKE '%Introduction to Algorithms%'
ORDER BY s.full_name;
```

11. Alunos com pelo menos um empréstimo ativo (lembretes)

```
SELECT DISTINCT s.student_id, s.full_name, s.email
FROM student s
JOIN loan      l  ON l.student_id = s.student_id
JOIN loan_item li ON li.loan_id    = l.loan_id
WHERE li.return_date IS NULL
ORDER BY s.full_name;
```

12. Livros sem autores associados (revisão de metadados)

```
SELECT b.book_id, b.title
FROM book b
LEFT JOIN book_author ba ON ba.book_id = b.book_id
WHERE ba.book_id IS NULL
ORDER BY b.title;
```

13. Editora com mais títulos cadastrados

```
SELECT p.publisher_id, p.name AS editora, COUNT(b.book_id) AS qtd_titulos
FROM publisher p
JOIN book b ON b.publisher_id = p.publisher_id
GROUP BY p.publisher_id, p.name
ORDER BY qtd_titulos DESC, p.name
LIMIT 1;
```

14. Top 3 alunos com mais itens emprestados no histórico

```
SELECT
  s.student_id,
  s.full_name,
  COUNT(li.*) AS total_itens
FROM student s
JOIN loan      l  ON l.student_id = s.student_id
JOIN loan_item li ON li.loan_id    = l.loan_id
GROUP BY s.student_id, s.full_name
ORDER BY total_itens DESC, s.full_name
LIMIT 3;
```

15. Alunos com mais de 1 empréstimo ativo (monitoramento)

```
SELECT
  s.student_id,
  s.full_name,
  COUNT(*) AS emprestimos_ativos
FROM student s
JOIN loan      l  ON l.student_id = s.student_id
JOIN loan_item li ON li.loan_id    = l.loan_id
WHERE li.return_date IS NULL
GROUP BY s.student_id, s.full_name
HAVING COUNT(*) > 1
ORDER BY emprestimos_ativos DESC, s.full_name;
```

16. Data mais recente de devolução por título

```
SELECT
  b.title,
  MAX(li.return_date) AS ultima_devolucao
FROM book b
JOIN exemplar  e  ON e.book_id     = b.book_id
JOIN loan_item li ON li.exemplar_id = e.exemplar_id
WHERE li.return_date IS NOT NULL
GROUP BY b.title
ORDER BY b.title;
```

17. Autores presentes em 2 ou mais livros

```
SELECT
  a.author_id,
  a.name AS autor,
  COUNT(DISTINCT ba.book_id) AS qtd_livros
FROM author a
JOIN book_author ba ON ba.author_id = a.author_id
GROUP BY a.author_id, a.name
HAVING COUNT(DISTINCT ba.book_id) >= 2
ORDER BY qtd_livros DESC, a.name;
```

18. Exemplares disponíveis hoje (sem empréstimo ativo)

```
SELECT
  e.exemplar_id,
  e.copy_code,
  b.title AS livro,
  e.location
FROM exemplar e
JOIN book b ON b.book_id = e.book_id
WHERE NOT EXISTS (
  SELECT 1
  FROM loan_item li
  WHERE li.exemplar_id = e.exemplar_id
    AND li.return_date IS NULL
)
ORDER BY b.title, e.copy_code;
```

19. Cursos com ao menos um aluno com empréstimo ativo

```
SELECT DISTINCT c.course_id, c.name AS curso
FROM course c
WHERE EXISTS (
  SELECT 1
  FROM student s
  JOIN loan      l  ON l.student_id = s.student_id
  JOIN loan_item li ON li.loan_id    = l.loan_id
  WHERE s.course_id = c.course_id
    AND li.return_date IS NULL
)
ORDER BY c.name;
```

20. Livros com a quantidade de autores (incluindo zero)

```
SELECT
  b.book_id,
  b.title,
  COUNT(ba.author_id) AS qtd_autores
```

```
FROM book b
LEFT JOIN book_author ba ON ba.book_id = b.book_id
GROUP BY b.book_id, b.title
ORDER BY b.title;
```