

SQL Joins - Aula 06

Gabriel de Paula Gaspar Pinto

Exercício 3

- a) A query não é satisfatória, já que não mostra os alunos e as matérias em que foram matriculados, mas sim a combinação de todas as linhas, múltiplas vezes. A query corrigida, fica:

```
1  SELECT
2  a.nome, c.codigo_disciplina
3  FROM
4  aluno AS a
5  INNER JOIN
6  aluno_cursa_disciplina AS c
7  ON a.matricula_aluno = c.matricula_aluno
8
```

É possível omitir a linha 7 e trocar de "INNER JOIN" para "NATURAL JOIN", usando o MySQL, que o resultado será o mesmo.

- b)
- ```
2 SELECT
3 a.nome, c.codigo_disciplina
4 FROM
5 aluno AS a
6 LEFT JOIN
7 aluno_cursa_disciplina AS c
8 ON a.matricula_aluno = c.matricula_aluno
9 UNION
10 SELECT
11 a.nome, c.codigo_disciplina
12 FROM
13 aluno AS a
14 RIGHT JOIN
15 aluno_cursa_disciplina AS c
16 ON a.matricula_aluno = c.matricula_aluno
```

- c)
- ```
2  SELECT
3  a.nome, c.codigo_disciplina
4  FROM
5  aluno AS a
6  LEFT JOIN
7  aluno_cursa_disciplina AS c
8  ON a.matricula_aluno = c.matricula_aluno
9  WHERE c.codigo_disciplina IS NULL
```

- d) Como as tabelas de alunos e de disciplinas não são conectadas diretamente, é necessário usar dois LEFT JOINS para mesclar as informações das duas tabelas, através da tabela de aluno_cursa_disciplina.

```
1  SELECT
2  d.nome, a.nome
3  FROM
4  disciplina AS d
5  LEFT JOIN
6  aluno_cursa_disciplina AS c
7  ON d.codigo_disciplina = c.codigo_disciplina
8  LEFT JOIN
9  aluno AS a
10 ON c.matricula_aluno = a.matricula_aluno
11
```

- e) Mostra o nome da disciplina quando nenhum aluno está matriculado. Se um aluno está matriculado, a disciplina está presente em C, caso contrário, a disciplina não está presente em C. Com isso, mostra todas as disciplinas presentes em D (tabela com TODAS as disciplinas) que não tenham seus correspondentes em C (tabela com alunos e disciplinas).

```
1      SELECT
2      d.nome
3      FROM
4      disciplina AS d
5      LEFT JOIN
6      aluno_cursa_disciplina AS c
7      ON c.codigo_disciplina = d.codigo_disciplina
8      WHERE c.codigo_disciplina IS NULL
9
```

- f) Para mostrar todas os alunos e suas respectivas disciplinas, é necessário usar o LEFT JOIN, da tabela de alunos para aluno_cursa_disciplina, que conterà todos os alunos, mesmo que não tenha nenhuma disciplina atrelado ao mesmo. Com o USING, a coluna de matricula_aluno é usada para unir as tabelas, já que esta coluna está presente nas duas tabelas.

```
1      SELECT a.nome, a.matricula_aluno, c.codigo_disciplina
2      FROM
3      aluno AS a
4      LEFT JOIN
5      aluno_cursa_disciplina AS c
6      USING (matricula_aluno)
7
```

- g) Para mostrar o nome de todas as disciplinas e o nome de seus respectivos alunos, usa-se o LEFT JOIN da tabela de disciplinas para aluno_cursa_disciplina, com o USING (codigo_disciplina), mostrando todas as disciplinas, sem exceção. Com o segundo LEFT JOIN, junta-se a tabela aluno_cursa_disciplina com a de alunos. Por fim, com o último USING, somente os alunos que estão matriculados serão mostrados, junto com todas as disciplinas.

```
1      SELECT d.nome, a.nome
2      FROM
3      disciplina AS d
4      LEFT JOIN
5      aluno_cursa_disciplina AS c
6      USING (codigo_disciplina)
7      LEFT JOIN
8      aluno AS a
9      USING (matricula_aluno)
10
```

- h) Fazendo um FULL JOIN, acontece a mesma coisa que aconteceu na questão anterior, com a exceção que todas as disciplinas (com alunos ou sem) e todos os alunos (com disciplinas ou não) são mostrados.

```
1      SELECT d.nome, a.nome
2      FROM
3      disciplina AS d
4      LEFT JOIN
5      aluno_cursa_disciplina AS c
6      USING (codigo_disciplina)
7      LEFT JOIN
8      aluno AS a
9      USING (matricula_aluno)
10
11     UNION
12
13     SELECT d.nome, a.nome
14     FROM
15     disciplina AS d
16     RIGHT JOIN
17     aluno_cursa_disciplina AS c
18     USING (codigo_disciplina)
19     RIGHT JOIN
20     aluno AS a
21     USING (matricula_aluno)
22
```

Exercício 4

a)

```
2 SELECT *
3 FROM
4 a
5 INNER JOIN
6 b
7 ON a.pk = b.pk
```

b)

```
2 SELECT *
3 FROM
4 a
5 LEFT JOIN
6 b
7 ON a.pk = b.pk
```

c)

```
2 SELECT *
3 FROM
4 a
5 RIGHT JOIN
6 b
7 ON a.pk = b.pk
```

d)

```
2 SELECT *
3 FROM
4 a
5 LEFT JOIN
6 b
7 ON a.pk = b.pk
8 WHERE b.pk IS NULL
```

e)

```
2 SELECT *
3 FROM
4 a
5 RIGHT JOIN
6 b
7 ON b.pk = a.pk
8 WHERE a.pk IS NULL
```

f)

```
2 SELECT *
3 FROM
4 a
5 LEFT JOIN
6 b
7 ON a.pk = b.pk
8
9 UNION
10
11 SELECT *
12 FROM
13 a
14 RIGHT JOIN
15 b
16 ON a.pk = b.pk
```

g)

```
2 SELECT *
3 FROM
4 a
5 LEFT JOIN
6 b
7 ON a.pk = b.pk
8 WHERE b.pk IS NULL
```

```
9      UNION
10
11     SELECT *
12     FROM
13     a
14     RIGHT JOIN
15     b
16     ON b.pk = a.pk
17     WHERE a.pk IS NULL
18
```

Considerações finais

A partir da questão D), eu coloquei explicações de como cada query funciona porque eu estava tendo dificuldades. Explicar como elas funcionam me ajudou a entender e a fixar o conteúdo. Este e outros trabalhos feitos usando \LaTeX estão disponíveis em meu [GitHub](#), com seus códigos-fonte.