

Autor: Tiago Goncalves da Silva - 2023644

Considerando o banco de dados University usado nas aulas:

1 – Analisando a seguinte query que busca uma lista dos títulos (titles) de todos os cursos ministrados no semestre Spring 2017, junto com o nome do professor:

```
select name, title
from instructor natural join teaches natural join section natural
join course
where semester = 'Spring' and year = 2017
```

O que tem de errado com a query?

---comentário:

O problema é que esta query não mostra uma disciplina ministrada por um professor fora de seu departamento, pois o natural join junta o dept_name de instructor e de course. Uma possível solução para isso seria usar a condição on do join para indicar os campos a serem comparados.

2 – Escreva as seguintes queries em SQL:

a – Mostre a lista de todos os professores, incluindo o ID e o número de cursos ministrados. Se o professor não ministrou nenhum curso, mostrar 0. Sua query deve usar um outer join e não usar subqueries.

---query:

```
SELECT i.id, count(course_id) as course_count
FROM instructor as i LEFT OUTER JOIN teaches as t
ON i.id = t.id
GROUP BY (i.id)
```

b – Mostre a lista de todos os cursos ofertados em Spring 2018, com o ID e nome de cada professor da oferta. Se o curso tem mais de um professor, a oferta deve aparecer múltiplas vezes mostrando os vários professores. Se a oferta não tiver nenhum professor, ainda assim deverá aparecer na listagem.

---query:

```
SELECT t.*, i.id, i.name
FROM teaches as t LEFT OUTER JOIN instructor as i
ON (i.id = t.id)
WHERE year = 2018 and semester = 'Spring'
```

c – Mostre a lista de todos os departamentos, com o número total de professores em cada departamento, sem usar subqueries.

---query:

```
SELECT d.*, count(i.id)
FROM department as d JOIN instructor as i
ON d.dept_name = i.dept_name
```

3 – Expressões SQL outer join podem ser resolvidas usando outros operadores na maioria das vezes. Isso pode ter implicações em desempenho e otimização, tanto para melhorar quanto para piorar. Sendo assim, altere as seguintes queries para remover o outer join.

a – select * from student natural left outer join takes

---query:

```
SELECT *
FROM student AS s JOIN takes AS t
ON s.id = t.id
UNION
SELECT ID, name, dept_name, tot_cred, null, null, null, null, null
FROM student as s
WHERE s.id NOT IN
(SELECT s.id
FROM student AS s JOIN takes AS t
ON s.id = t.id)
```

b – select * from student natural full outer join takes

---query:

```
SELECT *
FROM student AS s JOIN takes AS t
ON s.id = t.id
UNION
SELECT ID, name, dept_name, tot_cred, null, null, null, null, null
FROM student as s
WHERE s.id NOT IN
(SELECT s.id
FROM student AS s JOIN takes AS t
ON s.id = t.id)
UNION
SELECT null, null, null, null, ID, course_id, sec_id, semester, year, grade
FROM takes as t
WHERE t.id NOT IN
(SELECT t.id
FROM student AS s JOIN takes AS t
ON s.id = t.id)
```

4 – Mostre a definição de uma view que contenha o escore médio dos estudantes, levando em conta todas as disciplinas cursadas por estudante. Explique como considerou no cálculo os valores null nas notas.

---query:

```
CREATE VIEW student_grades AS
SELECT *, avg(year)
FROM student as s JOIN takes as t
ON s.id = t.id
GROUP BY s.id
```

---comentário:

optei por usar a média dos anos, pois as grades estão no formato de letras, não podendo assim usar `avg(grades)`