

Aluno: Tiago Gonçalves da Silva
RA: 2023644

Exercícios baseados no banco de dados University

1. criar a base de dados acadêmica como sugerida pelo autor e incluir registros de exemplo.
Feito

2. listar os professores ordenando o resultado de forma decrescente pelo campo salário
SELECT *
FROM instructor
ORDER BY salary desc

3. listar os professores com o orçamento do seu respectivo departamento
SELECT i.*, budget
FROM instructor as i, department as d
WHERE i.dept_name = d.dept_name

4. listar a soma dos salários de cada departamento
SELECT dept_name, sum(salary)
FROM instructor
GROUP BY dept_name

5. listar todos os nomes de estudantes que fazem pelo menos uma disciplina de computação, sem duplicados
SELECT DISTINCT name
FROM takes JOIN student
ON (student.ID = takes.ID)
WHERE course_id LIKE "%CS%"

6. para cada departamento, encontrar o maior salário entre os professores
SELECT max(salary), dept_name
FROM instructor
GROUP BY dept_name

7. listar os ids e nomes dos estudantes que não fizeram nenhuma disciplina antes da primavera de 2017
SELECT id, name
FROM
(SELECT s.ID, s.name, min(year)
FROM student AS s JOIN takes AS t
ON (t.ID = s.ID)
WHERE year > 2017 or (semester = "Fall" and year = 2017)
GROUP BY s.ID) as s

8. encontrar o menor salário, dos maiores por departamento da query anterior
SELECT min(max_salary)
FROM
(SELECT dept_name, max(salary) as max_salary
FROM instructor
GROUP BY dept_name) as T

9. mostrar uma lista de todos os professores, mostrando seu ID e o número de ofertas de disciplina que estão ministrando

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

10. listar todas as ofertas de disciplinas no semestre Spring 2018, com o ID e nome do professor ministrando a disciplina

```
SELECT t.*, i.name
FROM teaches as t JOIN instructor as i
ON (i.ID = t.ID)
WHERE semester = "Spring" and year = 2018
```

11. listar todos os departamentos, com o número total de instrutores em cada um

```
SELECT dept_name, count(ID) as teacher_count
FROM instructor
GROUP BY dept_name
ORDER BY dept_name
```

12. listar as disciplinas do departamento "Comp. Sci." que tem 3 créditos

```
SELECT *
FROM course
WHERE credits = 3 and dept_name = "Comp. Sci."
```

13. listar os IDs de todos os estudantes que tiveram aulas com o Prof. Einstein

```
SELECT s.ID
FROM teaches as t JOIN instructor as i JOIN student as s JOIN takes as ta
ON (t.ID = i.ID and t.course_id = ta.course_id and ta.ID = s.ID)
WHERE i.name = "Einstein"
```

14. encontrar o salário mais alto de um professor

```
SELECT max(salary)
FROM instructor
```

15. listar todos os professores ganhando o maior salario

```
SELECT i.*
FROM instructor as i JOIN
(SELECT max(salary) as max_salary
FROM instructor) as m
ON (i.salary = max_salary)
```

16. aumentar os salários dos professores de Comp. Sci. Em 10%

```
UPDATE instructor
SET salary = salary * 1.10
WHERE dept_name = "Comp. Sci."
```

---17. apagar todas as disciplinas que nunca foram ofertadas (que não constam na tabela de section)

```
DELETE FROM course as c
WHERE c.course_id NOT IN
```

```
(SELECT s.course_id  
FROM section as s)
```

18. incluir mais 5 estudantes na tabela apropriada
INSERT INTO student (ID, name, dept_name, tot_cred)
values ('12346', 'Tiago', 'Comp. Sci.', 0);
INSERT INTO student (ID, name, dept_name, tot_cred)
values ('12347', 'Vitor', 'Comp. Sci.', 0);
INSERT INTO student (ID, name, dept_name, tot_cred)
values ('12348', 'Marcos', 'Physics', 0);
INSERT INTO student (ID, name, dept_name, tot_cred)
values ('12349', 'Thiago', 'History', 0);
INSERT INTO student (ID, name, dept_name, tot_cred)
values ('12347', 'Julia', 'Music', 0);

19. incluir uma disciplina em Comp. Sci.
INSERT INTO course (course_id, title, dept_name, credits)
VALUES ('CS-F30', 'Intro. to Programming', 'Comp. Sci.', 6)

20. criar uma oferta de disciplina para a disciplina da questão anterior
INSERT INTO section (course_id, sec_id, semester, year, building, room_number, time_slot_id)
VALUES ('CS-F30', 1, 'Fall', '2022', 'Packard', '101', 'B')

21. matricular 10 alunos na oferta de disciplina da questão anterior
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12346','CS-F30',1,'Fall','2022','A+');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12347','CS-F30',1,'Fall','2022','B');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12348','CS-F30',1,'Fall','2022','B');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12349','CS-F30',1,'Fall','2022','C-');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12350','CS-F30',1,'Fall','2022','A');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('00128','CS-F30',1,'Fall','2022','C-');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('','CS-F30',1,'Fall','2022','');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('','CS-F30',1,'Fall','2022','');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('12345','CS-F30',1,'Fall','2022','B-');
INSERT INTO takes (ID, course_id, sec_id, semester, year, grade)
VALUES('76653','CS-F30',1,'Fall','2022','C+');

22. listar os departamentos que contém a string "sci", independente de caracteres maiúsculos ou minúsculos
SELECT *
FROM department
WHERE dept_name LIKE "%sci%"