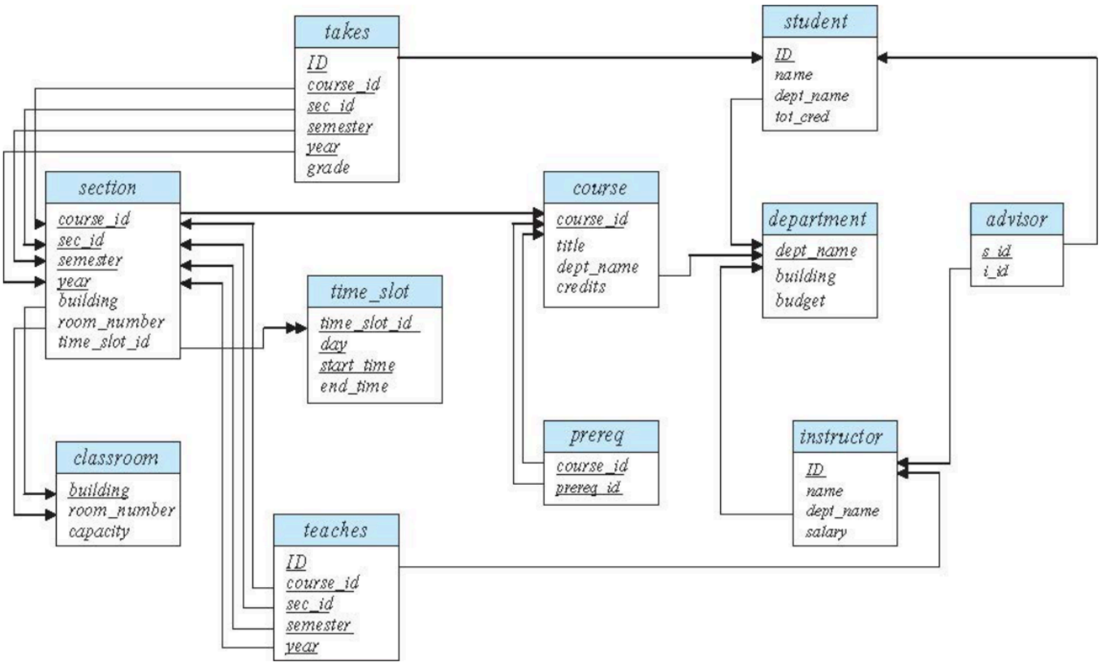
Atividade 02 Banco de dados - University
Amanda Jury Nakamura RA2582686



Respostas:

1)
```
create table NOME (
campo tipo(número de caracteres)
primary key (campo),
foreign key (campo) references campo);
```

2)
```
select ID, name, salary
from instructor
order by salary desc
– limit 1 (mostra só o primeiro da lista)
```

3)
```
select i.id, i.name, i.dept_name, d.budget
from instructor as i
        join department as d on i.dept_name = d.dept_name
-- order by budget desc;
-- order by dept_name;
```

4)
```
select dept_name, sum(salary)
from instructor
group by dept_name;
```

– se tem sum() e o outro campo n está contido na soma, ele tem que estar no **group by**

```
5) select distinct s.name -- c.course_id, c.dept_name
   from student as s
           join takes as t on s.id = t.id
           join course as c on t.course_id = c.course_id
      where c.dept_name = 'comp. sci.'
```

```
6) select d.dept_name, max(salary)
   from instructor as i
           join department as d on i.dept_name = d.dept_name
   group by dept_name;

   – teacher’s answer –
   select  i1.name, i1.dept_name, i1.salary
   from instructor as i1
   where i1.salary = (select max(i2.salary)
                      fro instructor i2
                      where i2.dept_name = i1.dept_name)
```

```
7)
   select distinct s.ID, s.name
   from student as s
           join takes as t on s.id = t.id
   where t.year < '2017';
```

```
8) select d.dept_name, max(salary)
   from instructor as i
           join department as d on i.dept_name = d.dept_name
   group by dept_name;
   order by salary desc;
```

```
9) select i.ID, i.name, count(t.ID) AS n_disciplinas_lecionadas
   from instructor as i
   LEFT join teaches as t on i.ID = t.ID
   GROUP BY i.ID, i.name;    – dá para agrupar por mais de um
```

-- A questão pede "todos os professores". Se houver algum professor que não esteja ministrando nenhuma disciplina (e, portanto, não tem entrada na tabela teaches), um INNER JOIN (que é o padrão do JOIN que você usou) não incluirá esse professor na lista. Para incluir todos os professores, mesmo aqueles com zero disciplinas, você precisa de um LEFT JOIN.

-- A função COUNT(t.ID) é aplicada a cada grupo para contar as entradas de teaches associadas àquele professor.

```
10) select i.ID, i.name, c.title, t.semester, t.year
    from teaches as t
            left join instructor as i on i.ID = t.ID
       left join course as c on  t.course_id = c.course_id
    where t.semester = 'spring' and t.year = 2018
```

```
     order by i.name

11)  select i.dept_name, count(i.name)
     from department as d
               join instructor as i on i.dept_name = d.dept_name
     group by d.dept_name   (aqui tem que ser uma coluna e NÂO uma tabela)
     – order by count(i.name);

12)  select c.title, credits
     from course as c
     where c.dept_name = 'comp. sci.' and credits = 3

13)
     select  s.id, s.name
     from student as s
               join takes as t on  s.id = t.id
               join teaches on t.course_id = teaches.course_id and
                                             t.sec_id = teaches.sec_id and
                                             t.semester = teaches.semester and
                                             t.year = teaches.year
               join instructor as i on i.id = teaches.id
     where i.name = 'Einstein'

     – teacher's answer –
     select ID, course_id, sec_id, year, semester
     from teaches join instructor using(ID)
               join takes using(course_id, sec_id, year, semester)
               join student as s on (s.id = t.id)
     where name = 'Einstein”

14)  select ID, name, salary
     from instructor
     where salary = (select max(salary) from instructor);
```

está tentando selecionar uma função agregada (`MAX(salary)`, que retorna um *único valor* para toda a tabela ou grupo) junto com uma coluna não agregada (`name`) **sem usar a cláusula `GROUP BY`**

```
15)  SELECT i.ID, i.name, i.salary, i.dept_name
     FROM instructor AS i
               JOIN (SELECT dept_name, MAX(salary) AS max_dept_salary
               FROM instructor
               GROUP BY dept_name)
     ON i.dept_name = max_sals.dept_name AND
     i.salary = max_sals.max_dept_salary;

16)  update instructor
     set salary = salary * 1.10
```

```
    where dept_name = 'comp. sci'

17) delete from course
    where course_id not in (select course_id from section);

    —-- ou —-

    delete from course
    where not exists (select 1 from section where section.course_id = course.course_id);

18) insert into student (id, name, dept_name, tot_cred) values
    ('20001', 'ana silva', 'comp. sci.', 0),
    ('20002', 'bruno costa', 'physics', 0),
    ('20003', 'carla martins', 'history', 0),
    ('20004', 'daniel rocha', 'comp. sci.', 0),
    ('20005', 'eduarda souza', 'math', 0);

19) insert into course (course_id, title, dept_name, credits) values
    ('cs-500', 'topicos avancados em bd', 'comp. sci.', 4);

20) insert into section (course_id, sec_id, semester, year, building, room_number,
    time_slot_id) values
    ('cs-500', '1', 'fall', 2025, 'watson', '100', 'a');

21) insert into takes (id, course_id, sec_id, semester, year, grade) values
    ('20001', 'cs-500', '1', 'fall', 2025, null),
    ('20002', 'cs-500', '1', 'fall', 2025, null),
    ('20003', 'cs-500', '1', 'fall', 2025, null),
    ('20004', 'cs-500', '1', 'fall', 2025, null),
    ('20005', 'cs-500', '1', 'fall', 2025, null),
    ('10001', 'cs-500', '1', 'fall', 2025, null),
    ('10006', 'cs-500', '1', 'fall', 2025, null),
    ('10010', 'cs-500', '1', 'fall', 2025, null),
    ('12345', 'cs-500', '1', 'fall', 2025, null),
    ('98765', 'cs-500', '1', 'fall', 2025, null);

    22) select dept_name
    from department
    where lower(dept_name) = like ‘%sci%i’
```