



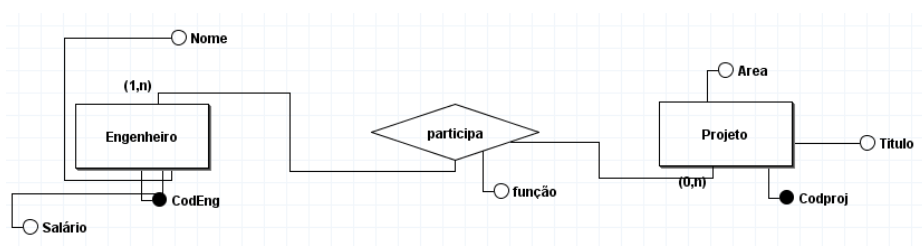
Disciplina: Banco de Dados II

Professora: Damires e Thiago

Aluno: \_\_\_\_\_ Matrícula: \_\_\_\_\_

### Tarefa 5 – Exercícios DDL, DML e Views

1. Observe o seguinte modelo em nível conceitual de uma aplicação sobre engenheiros e seus projetos. Como fica o modelo lógico relacional para este diagrama?



2. Crie um banco de dados no Postgres chamado “Projetos”. Em seguida, crie as tabelas conforme o esquema seguinte que deve estar compatível com o modelo conceitual e lógico anteriores. Observe que a chave primária da tabela Atuação é composta. Observe também que os tipos de codeng e codproj nas tabelas Engenheiro e Projeto são serial.

Table Atuação		Table Engenheiro		Table Projeto					
Properties	Definition	Inherits	Like	Columns	Properties	Definition	Inherits	Like	Columns
Column name	Definition	Column name	Definition	Column name	Column name	Definition			
codeng	integer NOT NULL	codeng	serial NOT NULL	codproj	codproj	serial NOT NULL			
codproj	integer NOT NULL	nome	character varying(30)	area	titulo	character varying(30)			
função	character varying(30)	salario	numeric(15,2)			character varying(30)			

3. Salve o script de criação de tabelas.
4. Caso não tenha ainda incluído as *constraints*, faça as alterações nas tabelas com as *constraints* de PK e FK.
5. Crie um check de validação em uma das tabelas. Explique o porquê desta regra.
6. Insira 5 engenheiros.
7. Insira 5 projetos.
8. Insira relacionamentos entre os projetos e engenheiros. Onde os relacionamentos foram inseridos?

\*\* Nas consultas seguintes, conteúdos podem ser adaptados a dados existentes em seu banco. Por exemplo, área = ‘Banco de Dados’.

9. Elabore consultas conforme se pede a seguir:

- Mostre os nomes dos engenheiros cujo salário seja menor que 15000.
- Apresente os nomes dos engenheiros que possuem a função “Analista” em projetos ou uma outra existente em seu banco.
- Mostre a quantidade de engenheiros por área de projeto.
- Verifique os nomes dos engenheiros que ganham **acima** da média salarial de todos os engenheiros.

10. Execute e analise o comando seguinte:

```
SELECT SUBSTR(e.nome, 1, 1) AS PrimeiraLetra, COUNT(*) AS Contagem
FROM engenheiro e
GROUP BY SUBSTR(e.nome, 1, 1);
```

11. Verifique o seguinte comando:

```
select nome
from engenheiro
where codeng in (select codeng
                  from atuacao
                  where codproj in (select codproj
                                    from projeto
                                    where area like 'Banco de Dados'))
```

O que o comando faz? Explique.

Reescreva-o, desta vez, usando **JOIN**.

12. Verifique o comando seguinte:

```
select codeng
from engenheiro
where salario > 1200
INTERSECT
select codeng
from atuação;
```

O que ele faz?

Refaça o comando usando uma *subquery*.

Depois, refaça-o usando JOIN.

13. Insira um novo engenheiro.

14. Verifique a seguinte consulta. Explique de que modo seu resultado pode ser interessante.

```
Select e.nome, e.codeng
from engenheiro e left join atuação a on e.codeng = a.codeng
where a.codeng is null
Order by e.nome;
```

Reescreva a consulta usando, desta vez, o operador EXCEPT. Compare os resultados.

15. Crie uma view permanente (salva no BD) mostrando os nomes dos engenheiros, sua função em cada projeto e o título do projeto. Consulte-a.

16. Analise e execute o comando seguinte:

```
WITH projs_por_area AS (  
    Select area, count(*) AS contagemArea  
    From projeto  
    Group by area )  
Select distinct p.titulo, ppa.contagemArea  
From projeto p JOIN projs_por_area ppa ON p.area = ppa.area  
Order by ppa.contagemArea DESC;
```

17. Analise, execute o comando seguinte e depois reescreva-o usando a cláusula WITH:

```
Select p.area, sum(E.salario) AS Area_Total  
From engenheiro e JOIN atuação a ON e.codeng = a.codeng  
    JOIN projeto p on a.codproj = p.codproj  
Group by p.area  
Having SUM(e.salario) >  
    (Select SUM(Area_Total)/COUNT(*)  
    From (Select p.area, SUM(e.salario) AS Area_Total  
        From engenheiro e JOIN atuação a ON e.codeng = a.codeng  
        JOIN projeto p on a.codproj = p.codproj  
        Group by p.area) as temp)  
Order by p.area;
```