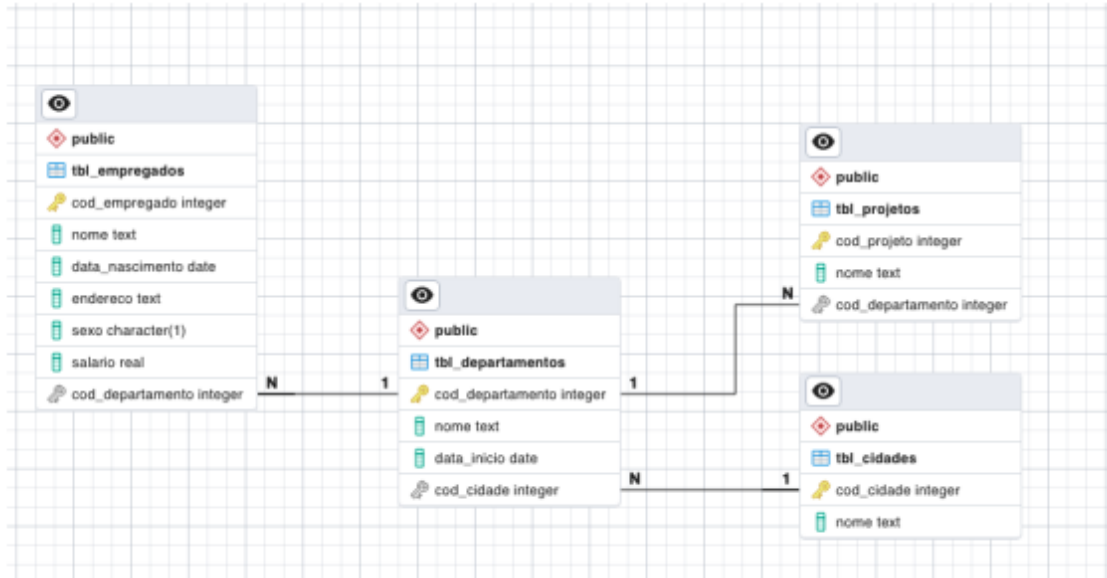


EXERCÍCIOS PRÁTICOS RELACIONADOS A AULA SOBRE VIEWS



Exercício 1

Objetivo: Criar uma **View v_depcidade** que lista os departamentos e as *cidades* onde estão localizados.

```
create view v_depcidade(dep_name, cidade) as
select d.nome, c.nome from tbl_departamentos d
inner join tbl_cidades c
on d.cod_cidade=c.cod_cidade;

select * from v_depcidade;
```

Explicação:

create view v_depcidade(dep_name, cidade) as

Cria uma View chamada **v_depcidade** com duas colunas nomeadas *dep_name* e *cidade*.

Essa View armazena uma consulta pré-definida.

select d.nome, c.nome

Seleciona o *nome* dos departamentos (d.nome) e das *cidades* (c.nome).

from tbl_departamentos d

A tabela *tbl_departamentos* é referenciada como *d*.

inner join tbl_cidades c

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25

Junta os dados da tabela *tbl_departamentos* com a tabela *tbl_cidades*.

on d.cod_cidade=c.cod_cidade

Define a condição de junção: a coluna *cod_cidade* deve ser igual em ambas as tabelas.

*select * from v_depcidade;*

Executa a View para retornar todos os registros.

O resultado exibirá 36 linhas, uma para cada departamento e sua cidade correspondente.

Aplicação em cenários reais

Utilidade:

Facilita a obtenção de informações sobre a localização dos departamentos, sendo útil para relatórios e análises logísticas ou administrativas.

Exemplo prático:

Em um sistema de gerenciamento de recursos humanos, o gestor quer verificar os departamentos disponíveis em cada cidade.

Consulta:

```
SELECT * FROM v_depcidade;
```

=====

Exercício 2

Objetivo: Criar a View **v_depcidadehouston** para listar os departamentos localizados em *Houston*.

```
create view v_depcidadehouston as
select * from v_depcidade where cidade='Houston';

select * from v_depcidadehouston;
```

Explicação:

create view v_depcidadehouston as

Cria uma nova View chamada *v_depcidadehouston*.

*select * from v_depcidade where cidade='Houston';*

Seleciona todas as colunas da View *v_depcidade*, filtrando apenas as linhas onde a cidade é *Houston*.

*select * from v_depcidadehouston;*

Executa a nova View para listar os departamentos localizados em *Houston*.

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25**Aplicação em cenários reais****Utilidade:**

Permite filtrar os departamentos por localidade específica, útil para análises ou planejamento regional.

Exemplo prático:

A empresa está planejando expandir as operações em *Houston* e deseja ver quais departamentos já estão instalados na cidade.

Consulta:

```
SELECT * FROM v_depcidadehouston;
```

=====

Exercício 3

Objetivo: Criar a View **v_opsalario** que lista a *soma* e a *média* dos salários dos empregados.

```
create view v_opsalario (soma, media)
as
select sum(salario), avg(salario) from tbl_empregados;

select * from v_opsalario;
```

Explicação:

create view v_opsalario (soma, media) as

Cria a View *v_opsalario* com duas colunas nomeadas *soma* e *media*.

select sum(salario), avg(salario) from tbl_empregados;

Calcula a soma total dos salários (*sum(salario)*) e a média dos salários (*avg(salario)*) da tabela *tbl_empregados*.

*select * from v_opsalario;*

Executa a View para exibir a *soma* e a *média* dos salários.

Aplicação em cenários reais**Utilidade:**

Ajuda a entender o impacto dos custos salariais, sendo útil para planejamento financeiro e elaboração de relatórios de recursos humanos.

Exemplo prático:

O departamento financeiro deseja saber o *total* gasto com salários e a *média* salarial da empresa para fins de orçamento.

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25

Consulta:

```
SELECT * FROM v_opsalario;
```

=====

Exercício 4

Objetivo: Criar a View **v_empregados_salarial** para listar *empregados* com salário *acima de 5500*.

```
CREATE VIEW v_empregados_salarial AS  
SELECT nome, salario  
FROM tbl_empregados  
WHERE salario > 5500;  
  
select * from v_empregados_salarial;
```

Explicação:

CREATE VIEW v_empregados_salarial AS

Cria a View *v_empregados_salarial*.

SELECT nome, salario

Seleciona as colunas *nome* e *salario* da tabela *tbl_empregados*.

WHERE salario > 5500;

Filtra os empregados que têm salário *maior que 5500*.

select * from v_empregados_salarial;

Executa a View para listar os empregados que atendem ao critério de salário.

Aplicação em cenários reais

Utilidade:

Permite identificar os empregados com remunerações mais altas, útil para análises de folha de pagamento e políticas salariais.

Exemplo prático:

O setor de RH está analisando os salários para criar uma política de bônus para os empregados mais bem pagos.

Consulta:

```
SELECT * FROM v_empregados_salarial;
```

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25

=====

Exercício 5

Objetivo: Criar a View **v_departemp** para listar *departamentos* e a *quantidade* de empregados em cada um.

```
create or replace view v_departemp(departamento, quantidade)
as
select d.nome, count(e.cod_empregado)
from tbl_empregados e
inner join tbl_departamentos d
on e.cod_departamento = d.cod_departamento
group by d.nome

select * from v_departemp order by quantidade DESC
```

Explicação:

create or replace view v_departemp(departamento, quantidade) as

Cria ou substitui a View *v_departemp*, com as colunas *departamento* e *quantidade*.

select d.nome, count(e.cod_empregado)

Seleciona o *nome* do departamento (*d.nome*) e *conta* o número de empregados (*count(e.cod_empregado)*) associados a ele.

from tbl_empregados e inner join tbl_departamentos d

Faz a junção entre as tabelas *tbl_empregados* e *tbl_departamentos* usando *cod_departamento*.

group by d.nome;

Agrupar os resultados por *departamento* para calcular a *quantidade* de empregados em cada grupo.

*select * from v_departemp order by quantidade DESC;*

Retorna os resultados ordenados de forma *decrecente* pela quantidade de empregados.

Aplicação em cenários reais**Utilidade:**

Ajuda a visualizar a distribuição dos empregados por departamento, útil para decisões de alocação de recursos e contratações.

Exemplo prático:

O gestor quer priorizar departamentos com maior número de funcionários para treinamentos corporativos.

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25

Consulta:

```
SELECT * FROM v_departemp ORDER BY quantidade DESC;
```

=====

Exercício 6

Objetivo: Criar a **View v_departrouble** para listar *departamentos* que têm mais *projetos* do que *empregados*.

```
create view v_departrouble (nome) as
select d.nome
from tbl_departamentos d
LEFT join tbl_empregados e
on e.cod_departamento = d.cod_departamento
left join tbl_projetos p
on d.cod_departamento = p.cod_departamento
group by d.nome
having count(p.*) > count(e.*)
```

Explicação:

create view v_departrouble (nome) as

Cria a View *v_departrouble*, com uma única coluna chamada *nome*.

select d.nome

Seleciona o *nome* dos departamentos (*d.nome*).

LEFT join tbl_empregados e

Junta a tabela *tbl_departamentos* com *tbl_empregados* para incluir todos os departamentos, mesmo os sem empregados.

left join tbl_projetos p

Junta a tabela *tbl_departamentos* com *tbl_projetos* para incluir todos os departamentos, mesmo os sem projetos.

group by d.nome

Agrupa os resultados por *nome* do departamento.

having count(p.) > count(e.*);*

Filtra os departamentos onde o número de *projetos* é maior que o número de *empregados*.

Banco de Dados Relacional – DSM – Prof.ª Lucineide – 01/04/25

Aplicação em cenários reais

Utilidade:

Identifica departamentos sobrecarregados com projetos em relação à sua equipe, útil para redistribuir recursos ou ajustar estratégias.

Exemplo prático:

A diretoria está avaliando a necessidade de contratar mais pessoal para departamentos com alta carga de projetos.

Consulta:

*SELECT * FROM v_departrouble;*

=====