MSC SANDIR
RODRIGUES CAMPOS

BANCO DE DADOS 2

AULA 5

3”

RELEMBRANDO - COMANDO INSERT COM SELECT

1

Popular tabelas com registros de outras tabelas;

2

SINTAXE: INSERT INTO nova_Tabela SELECT * FROM Tabela_original WHERE <condição>

3

Notação: tabela.coluna

4

Sintaxe: Criando com select

CREATE table dados_bkp AS SELECT * from dados_originais

INSERT COM SELECT

- INSERT INTO tabela_destino (coluna1, coluna2, ...)SELECT coluna1, coluna2, ...FROM tabela_origem[ONDE condição][ORDENAR POR coluna][LIMIT quantidade];

    - Parâmetros:- tabela_destino: a tabela onde os dados serão inseridos.
    - coluna1, coluna2, ...: as colunas da tabela destino que receberão os dados
    - .- tabela_origem: a tabela de onde os dados serão selecionados.
    - coluna1, coluna2, ...: as colunas da tabela origem que serão selecionadas.
    - condição: uma condição que filtra os dados da tabela origem.
    - coluna: a coluna pela qual os dados serão ordenados.
    - quantidade: o número máximo de linhas que serão inseridas.

EXEMPLOS INSERT COM SELECT

Nome	Email	Telefone	idade

1. Inserir todos os dados de uma tabela em outra:

   INSERT INTO clientes_novos (nome, email, telefone)

   SELECT nome, email, telefone

   FROM clientes_antigos;

2. Inserir apenas os dados que atendem a uma condição:

   INSERT INTO clientes_novos (nome, email, telefone)

   SELECT nome, email, telefone FROM clientes_antigos

   WHERE idade > 18;

Mínimo 5 tuplas

EXEMPLOS INSERT COM SELECT

Mínimo 5 tuplas

3. Inserir apenas os dados que estão ordenados por uma coluna:

```
INSERT INTO clientes_novos (nome, email, telefone)
SELECT nome, email, telefoneFROM clientes_antigos
    ORDER BY nome;
```

4. Inserir apenas os dados que estão limitados a uma quantidade:

```
INSERT INTO clientes_novos (nome, email, telefone)SELECT nome, email, telefoneFROM
clientes_antigosLIMIT 2;
```

CREATE COM SELECT

CREATE TABLE tabela_novaSELECT coluna1, coluna2, ...FROM tabela_existente[ONDE condição][ORDENAR POR coluna][LIMIT quantidade];

- - tabela_nova: o nome da nova tabela que será criada.
- - coluna1, coluna2, ...: as colunas que serão selecionadas da tabela existente.
- - tabela_existente: a tabela de onde os dados serão selecionados.
- - condição: uma condição que filtra os dados da tabela existente.
- - coluna: a coluna pela qual os dados serão ordenados.
- - quantidade: o número máximo de linhas que serão selecionadas.

CREATE COM SELECT

Mínimo 5 tuplas

1. Criar uma nova tabela com todos os dados de uma tabela existente:

   CREATE TABLE clientes_novíssimos

   SELECT *FROM clientes_antigos;

2. Criar uma nova tabela com apenas os dados que atendem a uma condição::

   CREATE TABLE clientes_ novíssimos_atual

   SELECT *FROM clientes_antigos

   WHERE idade > 18;

CREATE COM SELECT

Mínimo 5 tuplas

3. Criar uma nova tabela com apenas os dados que estão ordenados por uma coluna:

```
CREATE TABLE clientes_ novíssimos_atual_da_silva
SELECT *FROM clientes_antigos
ORDER BY nome;
```

4. Criar uma nova tabela com apenas os dados que estão limitados a uma quantidade:

```
CREATE TABLE clientes_ novíssimos_atual_da_silva_sauro
    SELECT *FROM clientes_antigos
    LIMIT 2;
```

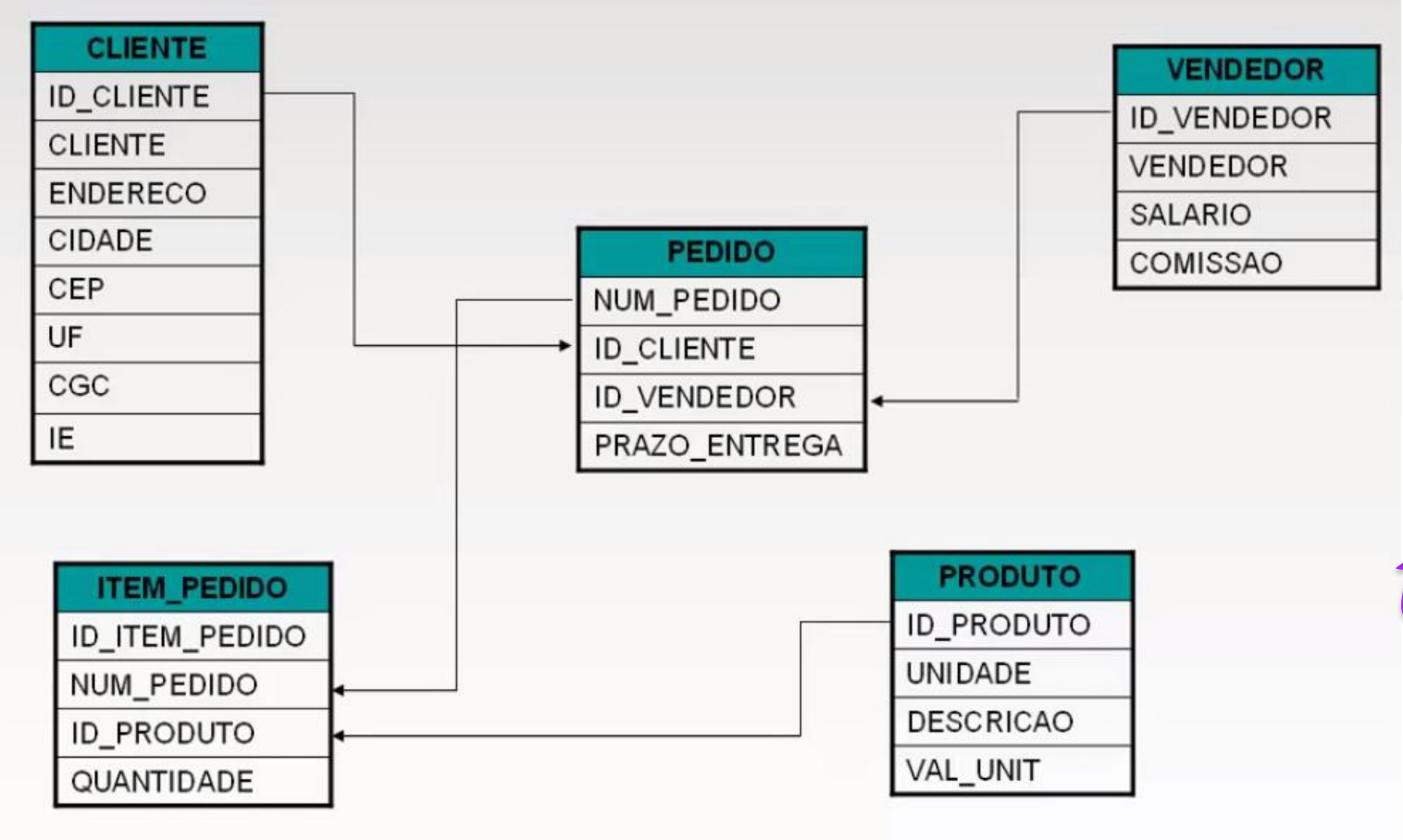
EXERCÍCIO INSET/CREATE COM SELECT

Mínimo 10 tuplas

Espelhe o banco de dados de vendas normalizado, popule tabelas temporárias com o nome exatamente igual mas a extensão tmp.

Exemplo: tb_produto_temp

EXERCÍCIO INSET/CREATE COM SELECT



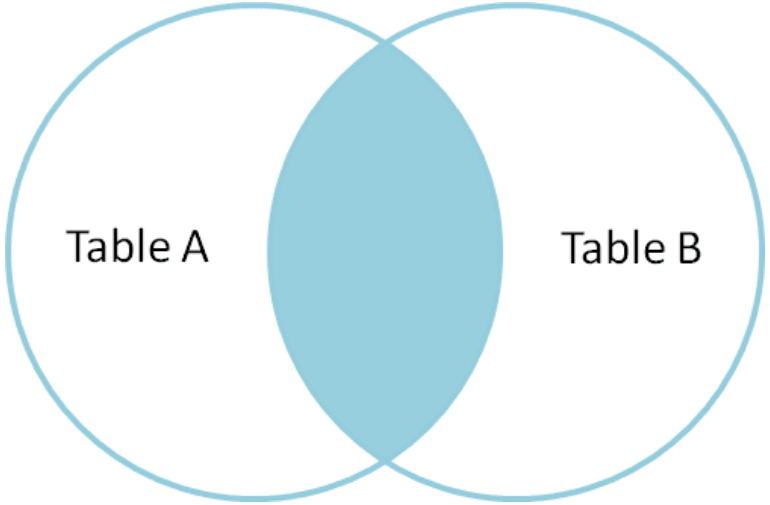
Mínimo 10 tuplas

TIPOS DE JOIN

- INNER
- LEFT
- RIGHT
- FULL
- CROSS

INNER JOIN

- O **INNER JOIN** é utilizado quando queremos retornar os registros que tenham correspondência nas duas tabelas presentes na junção.



INNER JOIN

Exemplo: em um banco de dados de uma empresa, existem duas tabelas que se relacionam entre si

COD_FUNCIONARIO	NOME	COD_CARGO
1	JOSE	3
2	DANIELLA	1
3	ANA	2
4	CARLOS	(null)

INNER JOIN

E a segunda possui os campos **código do cargo** e **descrição dos cargos**:

COD_CARGO	DESCRIÇÃO
1	VENDEDOR
2	CAIXA
3	GERENTE
4	ENTREGADOR

INNER JOIN NO SQL

	cod_cargo integer	dedcricao character varying(20)
1	1	vendedor
2	2	caixa
3	3	gerente
4	4	entregador

cargo

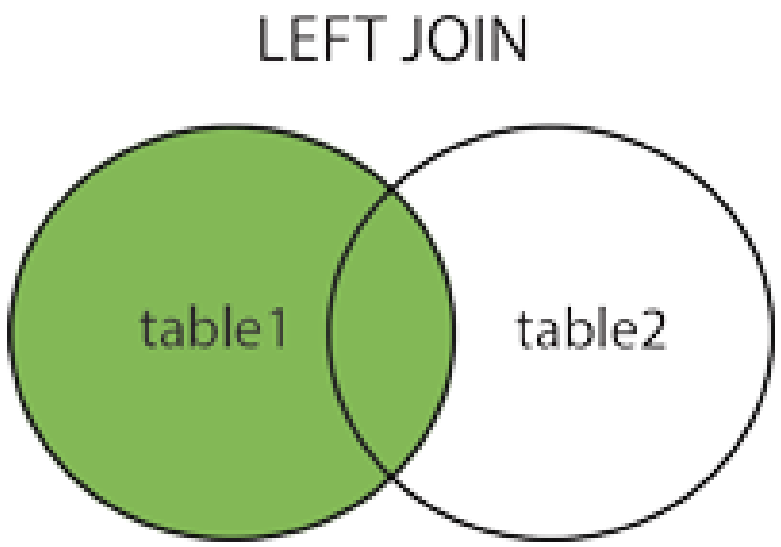
	cod_funcionario integer	nome character varying(20)	cod_cargo integer
1	1	jose	3
2	2	daniella	1
3	3	ana	2
4	4	carlos	

funcionario

- Exemplo: em um banco de dados de uma empresa, existem duas tabelas que se relacionam entre si

INNER JOIN RESULTADO

nome character varying(20)	cod_cargo integer	descricao character varying(20)
jose	3	gerente
daniella	1	vendedor
ana	2	caixa



LEFT JOIN

- O LEFT JOIN é utilizado quando queremos retornar apenas os registros da tabela da esquerda (tabela que está antes da cláusula LEFT JOIN) e os registros que tenham correspondência na tabela da direita.

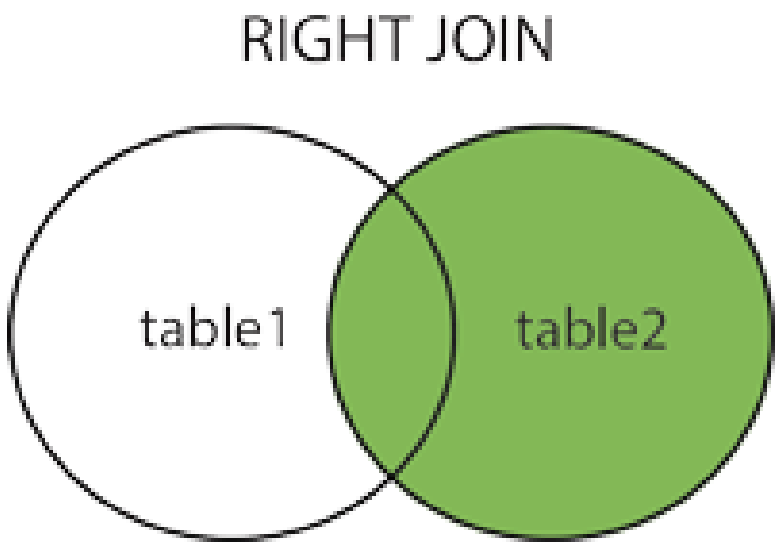
SINTAXE LEFT JOIN

SELECT <select_list> FROM Tabela A LEFT JOIN Tabela B ON A.Key = B.Key

SELECT f.nome, f.cod_cargo, c.descrição FROM funcionario f LEFT JOIN cargo c ON f.cod_cargo = c.cod_cargo;

LEFT JOIN RESULTADO

nome character varying(20)	cod_cargo integer	descricao character varying(20)
jose	3	gerente
daniella	1	vendedor
ana	2	caixa
carlos		



RIGHT JOIN

- O RIGHT JOIN é utilizado quando queremos retornar apenas os registros da tabela da direita(tabela que está após a cláusula RIGHT JOIN) e os registros que tenham correspondência na tabela da esquerda.

SINTAXE RIGTH JOIN

```
SELECT <select_list> FROM Tabela A RIGHT JOIN Tabela B ON A.Key = B.Key
```

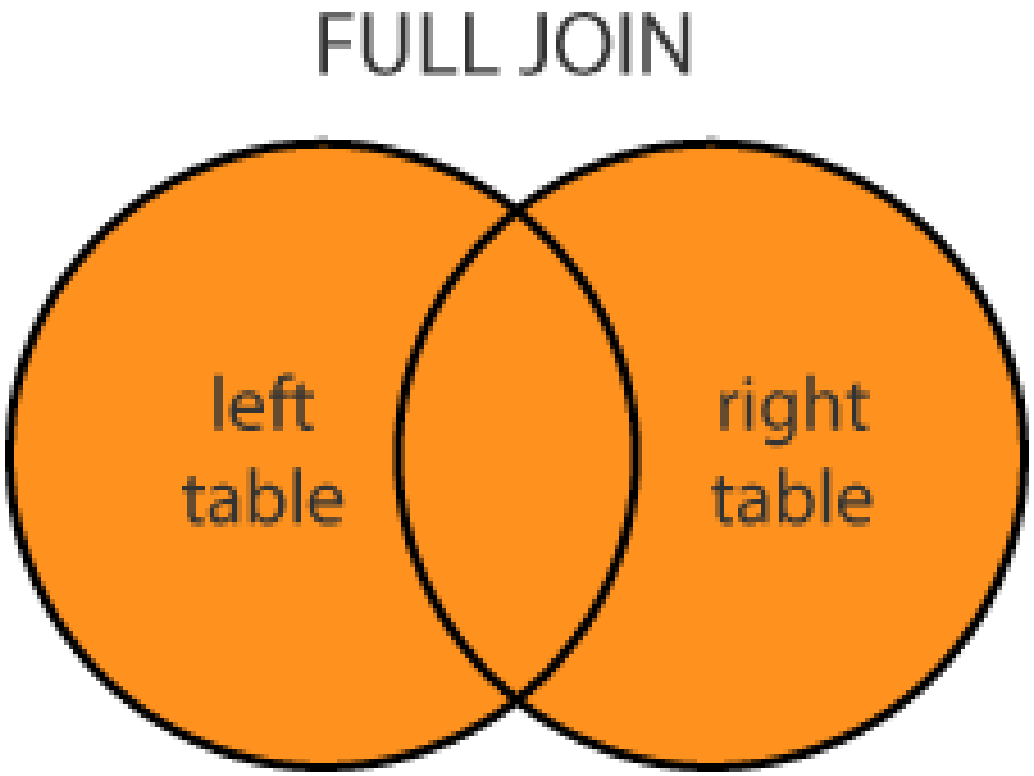
```
SELECT f.nome,f.cod_cargo,c.descricao FROM funcionario f RIGHT JOIN cargo c ON
f.cod_cargo = c.cod_cargo;
```

RIGTH JOIN RESULTADO

nome character varying(20)	cod_cargo integer	descricao character varying(20)
jose	3	gerente
daniella	1	vendedor
ana	2	caixa
		entregador

FULL JOIN

O FULL JOIN é utilizado quando queremos retornar registros que tenham correspondência em qualquer uma das tabelas presentes na junção.



SINTAXE FULL JOIN

```
SELECT *FROM tabela1

LEFT JOIN tabela2 ON tabela1.coluna = tabela2.coluna

UNION

SELECT *FROM tabela1

RIGHT JOIN tabela2 ON tabela1.coluna = tabela2.coluna;
```

FULL JOIN RESULTADO

nome character varying(20)	cod_cargo integer	descricao character varying(20)
jose	3	gerente
daniella	1	vendedor
ana	2	caixa
carlos		
		entregador

FIM DA AULA

EXERCÍCIOS

- **INNER JOIN básico**
  - Liste todos os pedidos junto com os nomes dos clientes.
- **INNER JOIN com múltiplas tabelas**
  - Exiba os detalhes dos itens de cada pedido, incluindo nome do cliente, nome do produto e quantidade.
- **LEFT JOIN (pedidos sem cliente?)**
  - Liste todos os pedidos, incluindo aqueles que não possuem um cliente associado (caso existam).
- **Contagem de pedidos por cliente (JOIN + GROUP BY)**
  - Mostre o nome do cliente e o número de pedidos que ele realizou.