

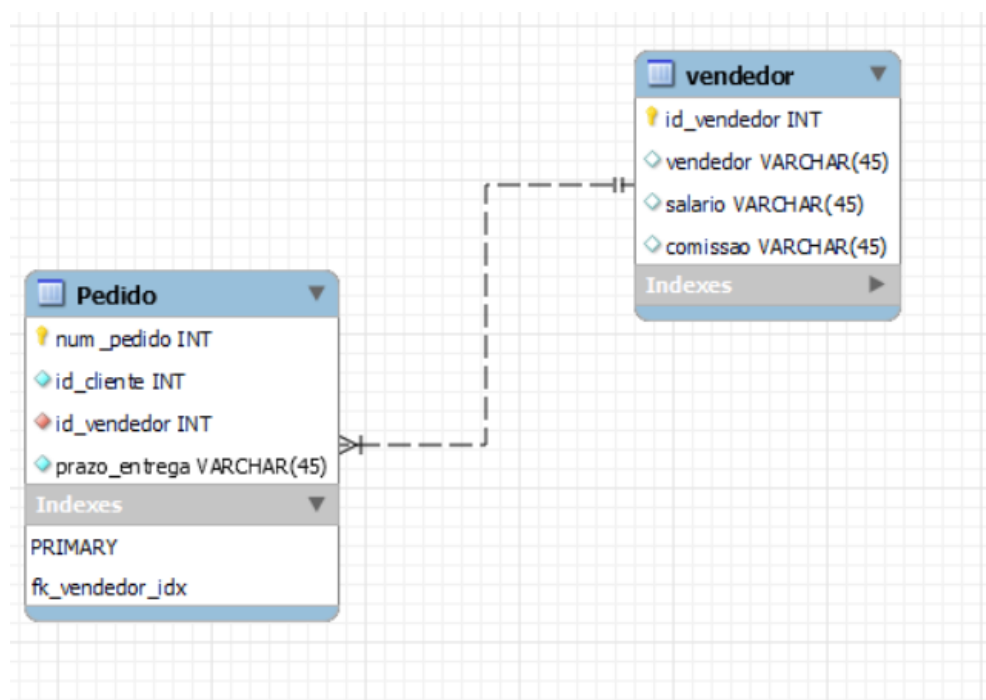
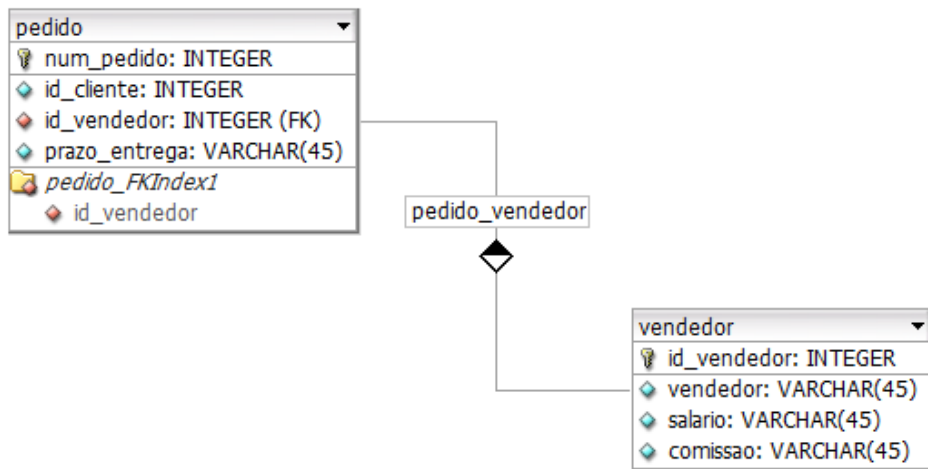
MSC SANDIR  
RODRIGUES CAMPOS

# BANCO DE DADOS 2

## AULA 3

6”



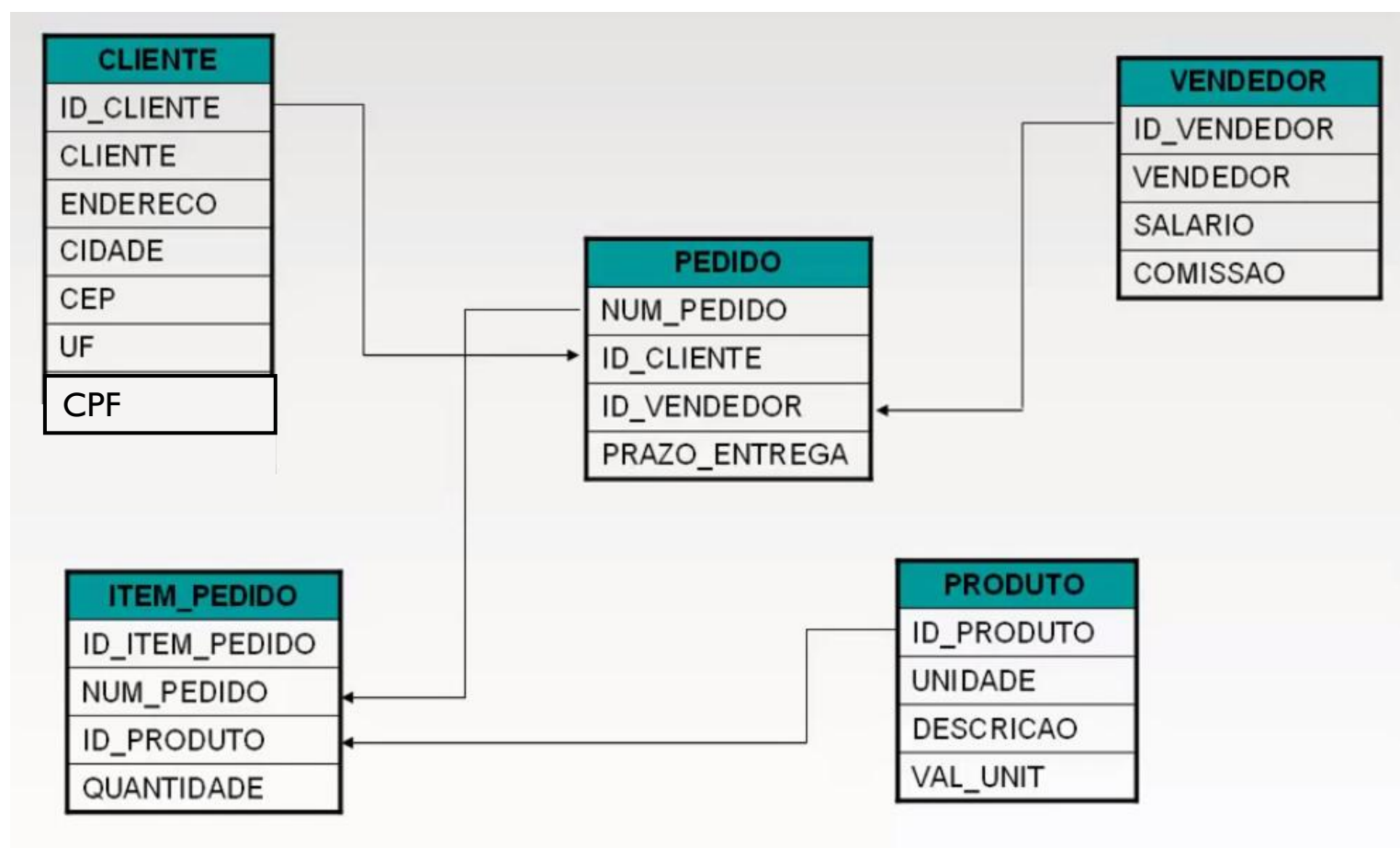


# RELACIONAMENTO E RELAÇÕES



# COM ESSE MODELO ATINGIMOS A 3ª FORMA NORMAL?

EXERCÍCIO 2 -  
CRIAR MAIS  
DUAS  
ENTIDADES  
PARA  
ADICIONAR A  
ESSE BANCO DE  
DADOS  
DEVIDAMENTE  
POPULADAS -  
CRIAR  
DIAGRAMA ER -  
MODELO DE  
DADOS E  
BANCO FÍSICO



# JUNÇÃO DE TABELAS

A junção de tabelas ocasiona uma tabela derivada de outras duas tabelas (reais ou derivadas), de acordo com as regras do tipo de junção.

No SQL as junções são classificadas como sendo qualificadas ou cruzadas.



# JUNÇÕES CRUZADAS

SINTAXE

```
SELECT* FROM Tabela1 CROSS  
JOIN Tabela2
```

**CROSS JOIN**

Gere uma lista com todas as combinações possíveis entre clientes e produtos



Cada linha de Tabela1 irá combinar-se com todas as linhas de Tabelas2. Para cada combinação de linhas de Tabela1 e Tabela2, a tabela derivada conterá uma linha com todas as colunas de Tabela1 seguidas por todas as colunas de Tabela2.



## JUNÇÃO DE TABELAS

# JUNÇÃO DE TABELAS

É óbvio que destas xxx linhas retornadas a maioria pode ser considerada inútil, portanto, devemos selecionar os nossos dados através de condições para nossa consulta. Essas condições são adicionadas através de cláusula WHERE.



# JUNÇÕES CRUZADAS

Exercício:

Criar uma consulta com junção cruzada onde os clientes só estejam relacionadas com seu respectivo produto.

Criar uma consulta com junção cruzada onde os clientes só estejam relacionadas com seu respectivo pedido.

Criar uma consulta com junção cruzada onde os vendedores só estejam relacionadas com seu respectivo pedidos.



## JUNÇÕES QUALIFICADAS

As junções qualificadas trazem um pouquinho mais de complexidade e são divididas em junções internas e externas. Na utilização de junção qualificada, se não for especificado como junção interna ou externa, por padrão o PostgreSQL considera como sendo interna.



# JUNÇÕES INTERNAS

A utilização da cláusula INNER é o que caracteriza o comando para uma junção interna, porém, ele não é obrigatório. Pode parecer à primeira vista que as junções internas se equiparam com as junções cruzadas vistas anteriormente, até por que as duas consultas a seguir são equivalentes:

- **SELECT \* FROM tabela CROSS JOIN tabela2**
- **SELECT \* FROM tabela INNER JOIN tabela2 ON TRUE**



# JUNÇÕES INTERNAS

- Mas nas junções internas é sempre obrigatória a especificação de condição de junção, ou seja, quais linhas de uma tabela têm alguma ligação com a linha de outra tabela. Para isso podemos utilizar uma das cláusulas `ON` ou `USING` ou utilizar a palavra `NATURAL` no nosso comando.
- A cláusula `ON` é o mais comumente utilizado por se assemelhar com a cláusula `WHERE`, ou seja, um par de linhas de Tabela1 e Tabela2 são correspondentes, se a expressão da cláusula `ON` produz um resultado verdade (`true`) para este par de linhas.



# JUNÇÕES INTERNAS



Exercício:



Criar uma consulta com junção interna onde os pedidos só estejam relacionadas com seu respectivo vendedor. E uma de pedidos com seus clientes



SINTAXE:



Crie um consulta que mostre o nome do produto comprado por cada cliente, usando junção interna qualificada

Criar de vendedor tbm

# JUNÇÕES INTERNAS

- A cláusula **USING** traz alguma semelhança com o ON, por também retornar um valor verdadeiro ou falso para aquele conjunto de linhas, no entanto, ele é uma forma mais rápida e abreviada de criação da consulta.
- Passando um nome de coluna, a execução desta consulta irá procurar nas tabelas a coluna especificada e comparar as duas.
- Por exemplo, `t1 INNER JOIN t2 USING (a, b, c)` equivale a `t1 INNER JOIN t2 ON (t1.a = t2.a AND t1.b = t2.b AND t1.c = t2.c)`. Portanto, a consulta anterior equivale à consulta abaixo:
- **SELECT \* FROM tabela1 INNER JOIN tabela2 USING (idtabela1)**



Para facilitar mais, existe a utilização de **NATURAL**, que nada mais é abreviação de **USING**. Com **NATURAL**, a consulta encontrará todas as colunas que tem nomes iguais nas duas tabelas e fará a comparação de igualdade.

O exemplo de **USING** anterior equivale ao seguinte:

```
SELECT * FROM tabela1 NATURAL INNER JOIN  
tabela2
```

# JUNÇÕES INTERNAS



# 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.

# TRABALHO DE CONSULTAS

1. Qual é o produto que os clientes de Brasília mais compram? (caso não haja nenhum de bsb – popular pelo menos 5 tuplas)
2. Qual vendedor tem mais clientes de fora de Brasília ?
3. Quantos clientes tem o principal vendedor?
4. Qual é o produto mais vendido pelo maior vendedor?
5. Qual é o produto vendido pelo pior vendedor (que tenha pelo menos uma venda)?



# APENDICE

# COMANDO INSERT COM SELECT

Serve para popular tabelas com registros de outras tabelas

SINTAXE:

```
INSERT INTO nova_Tabela SELECT * FROM Tabela_original WHERE <condição>
```

**Exercício:**

Use uma tabela para popular uma nova tabela que tenha pelo menos 50 tuplas

