

Linguagem Do Banco de Dados

Prof. Sergio Luiz

Conteúdo

1. Conceito DQL;
2. O que é DQL?
3. Praticando – Rodando Script;
4. Clausula SELECT;
5. Clausula WHERE;
6. Operação Rename(as);
7. Atividade;
8. Exercício.
9. Referencias

Data Query Language

1. Conceito - DQL

Data **Q**uery **L**anguage (DQL)
permite extrair dados do banco de dados.

O comando é:

➤ **SELECT** ➤ **FROM** ➤ **WHERE**

1. Conceito - DQL

A estrutura básica de uma consulta em SQL consiste em três cláusulas: **select**, **from** e **where**.

2. O que é ? - DQL

➤ SELECT

Ela é usada para relacionar os atributos desejados no resultado de uma consulta.

2. O que é ? - DQL

➤ FROM

Ela associa as relações que serão pesquisadas durante a evolução de uma expressão.

2. O que é ? - DQL

➤ WHERE

Ela consiste em um predicado envolvendo atributos da relação que aparece na cláusula **from**.

PRATICANDO

3. Rodando SCRIPT

➤ CRIE UM BANCO DE DADOS

Chame de:

AulasSql_Manha

```
/*CRIAR uma base de dados completa */  
CREATE DATABASE `AulasSql_Manha`
```

3. Rodando SCRIPT

- Crie uma tabela dentro do banco **AulasSql_Manha**

Chame de:

Colaborador_Manha

3. Rodando SCRIPT

Tabela → Colaborador

```
CREATE TABLE Colaborador (  
  Cod_Colaborador INT NOT NULL,  
  Primeiro_Nome VARCHAR (40) NOT NULL,  
  Ultimo_Nome VARCHAR (40) NOT NULL,  
  Ramal INT NOT NULL,  
  Data_Admissao DATE NOT NULL,  
  Nr_Depto INT NOT NULL,  
  Cod_Funcao VARCHAR (40) NOT NULL,  
  Grau_Funcao INT NOT NULL,  
  Local_Trabalho VARCHAR (40) NOT NULL,  
  Salario DOUBLE NOT NULL,  
  Nome_Completo VARCHAR (80) NOT NULL,  
  CONSTRAINT pk_Colaborador PRIMARY key (Cod_Colaborador)  
) ENGINE = InnoDB;
```

3. Rodando SCRIPT

➤ **Insira DADOS na Tabela
Colaborador**

**Próximo slide exemplo
de inserção de um dado
apenas!**

3. Rodando SCRIPT

- Linha de código para inserir DADOS na Tabela **Colaborador**

```
INSERT INTO `colaborador` (  
  `Cod_Colaborador`,  
  `Primeiro_Nome`,  
  `Ultimo_Nome`,  
  `Ramal`,  
  `Data_Admissao`,  
  `Nr_Depto`,  
  `Cod_Funcao`,  
  `Grau_Funcao`,  
  `Local_Trabalho`,  
  `Salario`,  
  `Nome_Completo`)
```

Código continua no próximo slide...

3. Rodando SCRIPT

➤ Linha de código para inserir DADOS na Tabela Colaborador

...continuação

```
VALUES (  
'28',  
'Ann',  
'Bennet',  
'5',  
'1991.02.01', /*FORMATO É: aaaa.mm.dd*/  
'120',  
'Admi',  
'5',  
'Inglaterra',  
'22935',  
'Bennet, Ann');
```

3. Rodando SCRIPT

- Envie e-mail para o professor e receba os próximos dados para ser inserido na tabela

Sergio.Isilveira@sesisenaipr.org.br


CLIQUE AQUI PARA VER O ARQUIVO

4. Clausula **SELECT**

- 1) Encontre todas as **tuplas** da relação **“colaborador”**, ordenando o resultado pelo atributo **“Primeiro_Nome”**.

```
select * from  
colaborador  
order by Primeiro_Nome
```

4. Clausula **SELECT**

O asterisco  pode ser usado para denotar “**todos os atributos**” da relação selecionada.

O resultado de uma consulta SQL é, naturalmente, uma relação.

RESPOSTA VISUAL

| Cod_Colaborador | Primeiro_Nome | Ultimo_Nome | Ramal | Data_Admissao | Nr_Depto | Cod_Funcao | Grau_Funcao | Local_Trabalho | Salario | Nome_Completo |
|-----------------|---------------|--------------|-------|---------------|----------|------------|-------------|----------------|---------|---------------------|
| 28 | Ann | Bennet | 5 | 1991-01-02 | 120 | Admi | 5 | Inglaterra | 22935 | Bennet, Ann |
| 45 | Ashok | Ramanathan | 209 | 1991-08-01 | 621 | Eng | 3 | Estados Unidos | 80689 | Ramanathan,Ashok |
| 114 | Bill | Parker | 247 | 1993-06-01 | 623 | Eng | 5 | Estados Unidos | 35000 | Parker,Bill |
| 4 | Bruce | Young | 233 | 1988-12-28 | 621 | Eng | 2 | Estados Unidos | 97500 | Young,Bruce |
| 52 | Carol | Nordstrom | 420 | 1991-10-02 | 180 | PRel | 4 | Estados Unidos | 42742 | Nordstrom,Carol |
| 20 | Chris | Papadopoulos | 887 | 1990-01-01 | 671 | Mngr | 3 | Estados Unidos | 89655 | Papadopoulos,Chris |
| 72 | Claudia | Sutherland | 0 | 1992-04-20 | 140 | SRep | 4 | Canada | 100914 | Sutherland,Claudia |
| 83 | Dana | Bishop | 290 | 1992-06-01 | 621 | Eng | 3 | Estados Unidos | 62500 | Bishop,Dana |
| 134 | Jacques | Glon | 0 | 1993-08-23 | 123 | SRep | 4 | França | 390500 | Glon,Jacques |
| 34 | Janet | Baldwin | 2 | 1991-03-21 | 110 | Sales | 3 | Estados Unidos | 53167 | Baldwin,Janet |
| 71 | Jennifer M. | Burbank | 289 | 1992-04-15 | 622 | Eng | 3 | Estados Unidos | 53167 | Burbank,Jennifer M. |
| 114 | John | Montgomery | 820 | 1994-03-30 | 672 | Eng | 5 | Estados Unidos | 35000 | Montgomery,John |

4. Clausula **SELECT**

2) Encontre todas as tuplas da relação **“colaborador”**, com dupla ordenação:

1) primeiro pelo atributo **“Nr_Depto”** e as tuplas com valor igual para o atributo **“Nr_Depto”** ficam ordenadas pelo atributo **“Primeiro_Nome”**.

4. Clausula **SELECT**

RESPOSTA

```
select * from  
colaborador  
order by Nr_Depto,  
Primeiro_Nome
```


RESPOSTA VISUAL

| Cod_Colaborador | Primeiro_Nome ▲ 2 | Ultimo_Nome | Ramal | Data_Admissao | Nr_Depto ▲ 1 | Cod_Funcao | Grau_Funcao | Local_Trabalho | Salario | Nome_Completo |
|-----------------|-------------------|--------------|-------|---------------|--------------|------------|-------------|----------------|---------|----------------------|
| 105 | Oliver H. | Bender | 225 | 1992-10-08 | 0 | CEO | 1 | Estados Unidos | 212850 | Bender, Oliver H. |
| 12 | Terri | Lee | 256 | 1990-05-01 | 0 | Admin | 4 | Estados Unidos | 53793 | Lee Terri |
| 85 | Mary S. | MacDonald | 447 | 1992-06-01 | 100 | VP | 2 | Estados Unidos | 111262 | MacDonald, Mary S. |
| 127 | Michael | Yanowski | 492 | 1993-08-09 | 100 | SRep | 4 | Estados Unidos | 44000 | Yanowski, Michael |
| 34 | Janet | Baldwin | 2 | 1991-03-21 | 110 | Sales | 3 | Estados Unidos | 53167 | Baldwin, Janet |
| 61 | Luke | Leung | 3 | 1992-02-18 | 110 | SRep | 4 | Estados Unidos | 68805 | Leung, Luke |
| 118 | Takashi | Yamamoto | 23 | 1993-07-01 | 115 | SRep | 4 | Japão | 7480000 | Yamamoto, Takashi |
| 110 | Yuki | Ichida | 22 | 1993-02-04 | 115 | Eng | 3 | Japão | 6000000 | Ichida Yuki |
| 28 | Ann | Bennet | 5 | 1991-01-02 | 120 | Admi | 5 | Inglaterra | 22935 | Bennet, Ann |
| 36 | Roger | Reeves | 6 | 1991-04-25 | 120 | Admin | 5 | Inglaterra | 22935 | Reeves, Roger |
| 37 | Willie | Stansbury | 7 | 1991-04-25 | 120 | Eng | 4 | Inglaterra | 39224 | Stansbury, Willie |
| 134 | Jacques | Glon | 0 | 1993-08-23 | 123 | SRep | 4 | França | 390500 | Glon, Jacques |
| 72 | Claudia | Sutherland | 0 | 1992-04-20 | 140 | SRep | 4 | Canada | 100914 | Sutherland, Claudia |
| 52 | Carol | Nordstrom | 420 | 1991-10-02 | 180 | PRel | 4 | Estados Unidos | 42742 | Nordstrom, Carol |
| 45 | Ashok | Ramanathan | 209 | 1991-08-01 | 621 | Eng | 3 | Estados Unidos | 80689 | Ramanathan, Ashok |
| 4 | Bruce | Young | 233 | 1988-12-28 | 621 | Eng | 2 | Estados Unidos | 97500 | Young, Bruce |
| 83 | Dana | Bishop | 290 | 1992-06-01 | 621 | Eng | 3 | Estados Unidos | 62500 | Bishop, Dana |
| 71 | Jennifer M. | Burbank | 289 | 1992-04-15 | 622 | Eng | 3 | Estados Unidos | 53167 | Burbank, Jennifer M. |
| 114 | Bill | Parker | 247 | 1993-06-01 | 623 | Eng | 5 | Estados Unidos | 35000 | Parker, Bill |
| 20 | Chris | Papadopoulos | 887 | 1990-01-01 | 671 | Mngr | 3 | Estados Unidos | 89655 | Papadopoulos, Chris |
| 114 | John | Montgomery | 820 | 1994-03-30 | 672 | Eng | 5 | Estados Unidos | 35000 | Montgomery, John |

4. Clausula **SELECT**

Encontre todas as tuplas da relação **“Colaborador”**, ordenando o resultado pelo atributo **“Primeiro_Nome”** e apresentando (ou projetando) somente os atributos: **“Primeiro_Nome”**, **“Ultimo_Nome”**, **“Nr_Depto”**.

4. Clausula **SELECT**

RESPOSTA

```
select Primeiro_Nome,  
Ultimo_nome, Nr_Depto  
from colaborador  
order by Primeiro_Nome
```


RESPOSTA VISUAL

**Nomes em
Ordem
Alfabética**

| Primeiro_Nome | 1 | Ultimo_nome | Nr_Depto |
|---------------|---|--------------|----------|
| Ann | | Bennet | 120 |
| Ashok | | Ramanathan | 621 |
| Bill | | Parker | 623 |
| Bruce | | Young | 621 |
| Carol | | Nordstrom | 180 |
| Chris | | Papadopoulos | 671 |
| Claudia | | Sutherland | 140 |
| Dana | | Bishop | 621 |
| Jacques | | Glon | 123 |
| Janet | | Baldwin | 110 |
| Jennifer M. | | Burbank | 622 |
| John | | Montgomery | 672 |
| Luke | | Leung | 110 |
| Mary S. | | MacDonald | 100 |
| Michael | | Yanowski | 100 |
| Oliver H. | | Bender | 0 |
| Roger | | Reeves | 120 |
| Takashi | | Yamamoto | 115 |
| Terri | | Lee | 0 |
| Willie | | Stansbury | 120 |
| Yuki | | Ichida | 115 |

4. Clausula **SELECT**

Encontre todas as tuplas da relação **“colaborador”**, ordenando o resultado pelo atributo **“Nr_Depto”** e apresentando (ou projetando) somente o atributo **“Nr_Depto”**.

Dê um tempo para os alunos descobrirem como fazer!

Como deve ser a Resposta!

| Nr_Depto | 1 |
|----------|---|
| 0 | |
| 0 | |
| 100 | |
| 100 | |
| 110 | |
| 110 | |
| 115 | |
| 115 | |
| 120 | |
| 120 | |
| 120 | |
| 123 | |
| 140 | |
| 180 | |
| 621 | |
| 621 | |
| 621 | |
| 622 | |
| 623 | |
| 671 | |
| 672 | |

4. Clausula **SELECT**

RESPOSTA

Primeira Solução!

```
select Nr_Depto  
from colaborador  
order by Nr_Depto
```

Primeira Solução!

| Nr_Depto | 1 |
|----------|---|
| 0 | |
| 0 | |
| 100 | |
| 100 | |
| 110 | |
| 110 | |
| 115 | |
| 115 | |
| 120 | |
| 120 | |
| 120 | |
| 123 | |
| 140 | |
| 180 | |
| 621 | |
| 621 | |
| 621 | |
| 622 | |
| 623 | |
| 671 | |
| 672 | |

Como
resolver
Esse
PROBLEMA???

Perceba que os Numeros dos Deptos estão repetidos

4. Clausula **SELECT**

SQL (como a maioria das linguagens comerciais de consulta) permite duplicidade nas relações.

Para eliminar as duplicidades deve-se inserir a palavra chave **distinct** depois da cláusula **select**.

Dê um tempo para os alunos descobrirem como fazer!

4. Clausula **SELECT**

RESPOSTA

Segunda Solução!

```
select distinct  
Nr_Depto  
from colaborador  
order by Nr_Depto
```

Segunda Solução!

**Retorno dos DEPTOS
sem REPETIÇÃO dos
números**

| Nr_Depto | 1 |
|----------|---|
| 0 | |
| 100 | |
| 110 | |
| 115 | |
| 120 | |
| 123 | |
| 140 | |
| 180 | |
| 621 | |
| 622 | |
| 623 | |
| 671 | |
| 672 | |

4. Clausula **SELECT**

Encontre todas as tuplas da relação **“colaborador”**, ordenando o resultado pelo atributo **“Nr_Depto”** e apresentando (ou projetando) somente o atributo **“Nr_Depto”**.

A SQL permite o uso da palavra-chave **all** para especificar **“explicitamente”** que as duplicidades não serão eliminadas.

4. Clausula **SELECT**

Dê um tempo para os alunos descobrirem como fazer!

4. Clausula **SELECT**

RESPOSTA

Solução!

```
select all Nr_Depto  
from colaborador  
order by Nr_Depto
```

Como deve ser a Resposta!



A screenshot of a list of department numbers (Nr_Depto) from a database. The list is enclosed in a green rectangular border. The numbers are: 0, 0, 100, 100, 110, 110, 115, 115, 120, 120, 120, 123, 140, 180, 621, 621, 621, 622, 623, 671, and 672. The number 100 is highlighted with a blue background.

| Nr_Depto |
|----------|
| 0 |
| 0 |
| 100 |
| 100 |
| 110 |
| 110 |
| 115 |
| 115 |
| 120 |
| 120 |
| 120 |
| 123 |
| 140 |
| 180 |
| 621 |
| 621 |
| 621 |
| 622 |
| 623 |
| 671 |
| 672 |

Retorna todos os depts, mesmo estando repetidos

4. Clausula **SELECT**

Encontre todas as tuplas da relação **“colaborador”**, ordenando o resultado pelo atributo **“Primeiro_Nome”** e apresentando (ou projetando) somente o atributo:

- a) **“Primeiro_Nome”**;
- b) **“Ultimo_Nome”**;
- c) **“Salario”**;
- d) **“Salario/12”**

Como deve ser a Resposta!

| Primeiro_Nome | Ultimo_Nome | Salario | Salario/12 |
|---------------|--------------|---------|--------------------|
| Ann | Bennet | 22935 | 1911.25 |
| Ashok | Ramanathan | 80689 | 6724.083333333333 |
| Bill | Parker | 35000 | 2916.6666666666665 |
| Bruce | Young | 97500 | 8125 |
| Carol | Nordstrom | 42742 | 3561.8333333333335 |
| Chris | Papadopoulos | 89655 | 7471.25 |
| Claudia | Sutherland | 100914 | 8409.5 |
| Dana | Bishop | 62500 | 5208.333333333333 |
| Jacques | Glon | 390500 | 32541.666666666668 |
| Janet | Baldwin | 53167 | 4430.583333333333 |
| Jennifer M. | Burbank | 53167 | 4430.583333333333 |
| John | Montgomery | 35000 | 2916.6666666666665 |
| Luke | Leung | 68805 | 5733.75 |
| Mary S. | MacDonald | 111262 | 9271.833333333334 |
| Michael | Yanowski | 44000 | 3666.6666666666665 |
| Oliver H. | Bender | 212850 | 17737.5 |
| Roger | Reeves | 22935 | 1911.25 |
| Takashi | Yamamoto | 7480000 | 623333.3333333334 |
| Terri | Lee | 53793 | 4482.75 |
| Willie | Stansbury | 39224 | 3268.6666666666665 |
| Yuki | Ichida | 6000000 | 500000 |

1º NOME

ULTIMO NOME

SALARIO

SALARIO/12

4. Clausula **SELECT**

A cláusula **select** pode conter expressões aritméticas envolvendo os operados **+**, **-**, ***** e **/**, e operandos constantes ou atributos das tuplas.

Dê um tempo para os alunos descobrirem como fazer!

4. Clausula **SELECT**

RESPOSTA

Solução!

select

Primeiro_Nome, Ultimo_No
me, Salário, Salario/12

from colaborador

order by Primeiro_Nome

Solução!

| Primeiro_Nome | Ultimo_Nome | Salario | Salario/12 |
|---------------|--------------|---------|--------------------|
| Ann | Bennet | 22935 | 1911.25 |
| Ashok | Ramanathan | 80689 | 6724.083333333333 |
| Bill | Parker | 35000 | 2916.6666666666665 |
| Bruce | Young | 97500 | 8125 |
| Carol | Nordstrom | 42742 | 3561.8333333333335 |
| Chris | Papadopoulos | 89655 | 7471.25 |
| Claudia | Sutherland | 100914 | 8409.5 |
| Dana | Bishop | 62500 | 5208.333333333333 |
| Jacques | Glon | 390500 | 32541.666666666668 |
| Janet | Baldwin | 53167 | 4430.583333333333 |
| Jennifer M. | Burbank | 53167 | 4430.583333333333 |
| John | Montgomery | 35000 | 2916.6666666666665 |
| Luke | Leung | 68805 | 5733.75 |
| Mary S. | MacDonald | 111262 | 9271.833333333334 |
| Michael | Yanowski | 44000 | 3666.6666666666665 |
| Oliver H. | Bender | 212850 | 17737.5 |
| Roger | Reeves | 22935 | 1911.25 |
| Takashi | Yamamoto | 7480000 | 623333.3333333334 |
| Terri | Lee | 53793 | 4482.75 |
| Willie | Stansbury | 39224 | 3268.6666666666665 |
| Yuki | Ichida | 6000000 | 500000 |

1º NOME

ULTIMO NOME

SALARIO

SALARIO/12

5. Clausula **WHERE**

Encontre todas as tuplas da relação **“colaborador”** para as quais o valor do atributo **“Nr_Depto”** seja **igual a 120**.

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

A SQL usa conectores lógicos **and**, **or** e **not**, em vez dos símbolos matemáticos \wedge , \vee e \neg , na cláusula **where**.

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

RESPOSTA

Solução!

```
Select *  
from colaborador  
Where Nr_Depto=120
```

Solução!

| Cod_Colaborador | Primeiro_Nome | Ultimo_Nome | Ramal | Data_Admissao | Nr_Depto | Cod_Funcao | Grau_Funcao | Local_Trabalho | Salario | Nome_Completo |
|-----------------|---------------|-------------|-------|---------------|----------|------------|-------------|----------------|---------|-------------------|
| 28 | Ann | Bennet | 5 | 1991-01-02 | 120 | Admi | 5 | Inglaterra | 22935 | Bennet, Ann |
| 36 | Roger | Reeves | 6 | 1991-04-25 | 120 | Admin | 5 | Inglaterra | 22935 | Reeves, Roger |
| 37 | Willie | Stansbury | 7 | 1991-04-25 | 120 | Eng | 4 | Inglaterra | 39224 | Stansbury, Willie |

Nr Depto = 120

5. Clausula **WHERE**

Os operandos dos conectivos lógicos podem ser expressões envolvendo **operadores de comparação** **<**, **<=**, **>**, **>=**, **=** e **<>** (diferente de).

5. Clausula **WHERE**

Encontre todas as tuplas da relação **“colaborador”** para as quais o valor do atributo **“salario”** esteja no intervalo fechado de **80000.00** até **90000.00**.

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

RESPOSTA

Solução 1

```
Select *  
from colaborador  
Where (salario >= 80000.00) and  
(salario <= 90000.00)
```


Solução 1

| Cod_Colaborador | Primeiro_Nome | Ultimo_Nome | Ramal | Data_Admissao | Nr_Depto | Cod_Funcao | Grau_Funcao | Local_Trabalho | Salario | Nome_Completo |
|-----------------|---------------|--------------|-------|---------------|----------|------------|-------------|----------------|----------|--------------------|
| 45 | Ashok | Ramanathan | 209 | 1991-08-01 | 621 | Eng | 3 | Estados Unidos | 80689.00 | Ramanathan,Ashok |
| 20 | Chris | Papadopoulos | 887 | 1990-01-01 | 671 | Mngr | 3 | Estados Unidos | 89655.00 | Papadopoulos,Chris |
| 11 | K.J | Weston | 34 | 1990-01-17 | 130 | SRep | 4 | Estados Unidos | 86292.00 | Weston,K.J |
| 24 | Pele | Fisher | 888 | 1990-09-12 | 671 | Eng | 3 | Estados Unidos | 81810.00 | Fisher,Pele |

SALARIO

5. Clausula WHERE

A SQL possui o operador de comparação between para simplificar a **cláusula where** que especifica que um atributo possa ter um valor maior ou igual a algum valor e menor ou igual a algum outro valor.

5. Clausula **WHERE**

Encontre todas as tuplas da relação **“colaborador”** para as quais o valor do atributo **“salario”** esteja no intervalo fechado de **80000.00** até **90000.00**.

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

RESPOSTA

Solução 2

```
Select *  
from colaborador  
Where salario between 80000.00  
and 90000.00
```

Solução 2

| Cod_Colaborador | Primeiro_Nome | Ultimo_Nome | Ramal | Data_Admissao | Nr_Depto | Cod_Funcao | Grau_Funcao | Local_Trabalho | Salario | Nome_Completo |
|-----------------|---------------|--------------|-------|---------------|----------|------------|-------------|----------------|----------|--------------------|
| 45 | Ashok | Ramanathan | 209 | 1991-08-01 | 621 | Eng | 3 | Estados Unidos | 80689.00 | Ramanathan,Ashok |
| 20 | Chris | Papadopoulos | 887 | 1990-01-01 | 671 | Mngr | 3 | Estados Unidos | 89655.00 | Papadopoulos,Chris |
| 11 | K.J | Weston | 34 | 1990-01-17 | 130 | SRep | 4 | Estados Unidos | 86292.00 | Weston,K.J |
| 24 | Pele | Fisher | 888 | 1990-09-12 | 671 | Eng | 3 | Estados Unidos | 81810.00 | Fisher,Pele |

SALARIO

5. Clausula **WHERE**

Encontre todas as tuplas da relação **“colaborador”** para as quais o valor do atributo **“Local_Trabalho”** seja igual a *Canada* ou igual a *Inglaterra* e apresentando(ou projetando) somente os atributos:

- a) **“Ultimo_Nome”**,
- b) **“Primeiro_Nome”** e
- c) **“Local_Trabalho”**

5. Clausula **WHERE**

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

A **SQL** possui o operador de teste de pertinência **in** que verifica se um dado valor é membro (ou pertence) a um conjunto de valores.

Na solicitação anterior tuplas que aparecerão no resultado da consulta são aquelas cujo valor do atributo **“Local_Trabalho”** pertença ao conjunto (**“Canada”, “Inglaterra”**).

5. Clausula **WHERE**

RESPOSTA

Solução!

Select

Ultimo_Nome, Primeiro_Nome, Local
_Trabalho

from colaborador

Where Local_Trabalho in
('Canada', 'Inglaterra')

Solução!

| Ultimo_Nome | Primeiro_Nome | Local_Trabalho |
|-------------|---------------|----------------|
| Bennet | Ann | Inglaterra |
| Sutherland | Claudia | Canada |
| Reeves | Roger | Inglaterra |
| Stansbury | Willie | Inglaterra |

**LOCAL
TRABALHO**

5. Clausula **WHERE**

Encontre todas as tuplas da relação **“colaborador”** para as quais o valor do atributo **“Data_Admissao”** seja:

- Um valor de data pertencente ao ano de 1991, ou seja, de 01/01/1991 até 31/12/1991;
- Ordene o resultado pelo atributo **“Data_Admissao”**;
- Apresente (ou projetando) somente os atributos: **“Primeiro_Nome”**, **“Salario”**, **“Data_Admissao”**

5. Clausula **WHERE**

Dê um tempo para os alunos descobrirem como fazer!

5. Clausula **WHERE**

RESPOSTA

Solução!

```
Select Primeiro_Nome, Salario,  
Data_Admissao  
from colaborador  
Where Data_Admissao between  
'1991/01/01' and '1991/12/31'  
Order by Data_Admissao
```

Solução!

| Primeiro_Nome | Salario | Data_Admissao |
|---------------|----------|---------------|
| Ann | 22935.00 | 1991-02-01 |
| Janet | 53167.00 | 1991-03-21 |
| Roger | 22935.00 | 1991-04-25 |
| Willie | 39224.00 | 1991-04-25 |
| Ashok | 80689.00 | 1991-08-01 |
| Carol | 42742.00 | 1991-10-02 |



Ordem por
Data

6. Operação Rename (as)

A **SQL** proporciona um mecanismo para rebatizar tanto relações (variáveis tuplas) quanto atributos, usando a cláusula as da seguinte forma:

nome_antigo as nome_novo

6. Operação Rename (as)

nome_antigo as nome_novo

```
select salario  
from colaborador  
where Nr_Depto = 120
```

salario
22935.00
22935.00
39224.00

```
select sum(salario) as tot_salario  
from colaborador  
where Nr_Depto = 120
```

tot_salario
85094.00

PRATICANDO

➤ CRIE UM BANCO DE DADOS

Chame de:

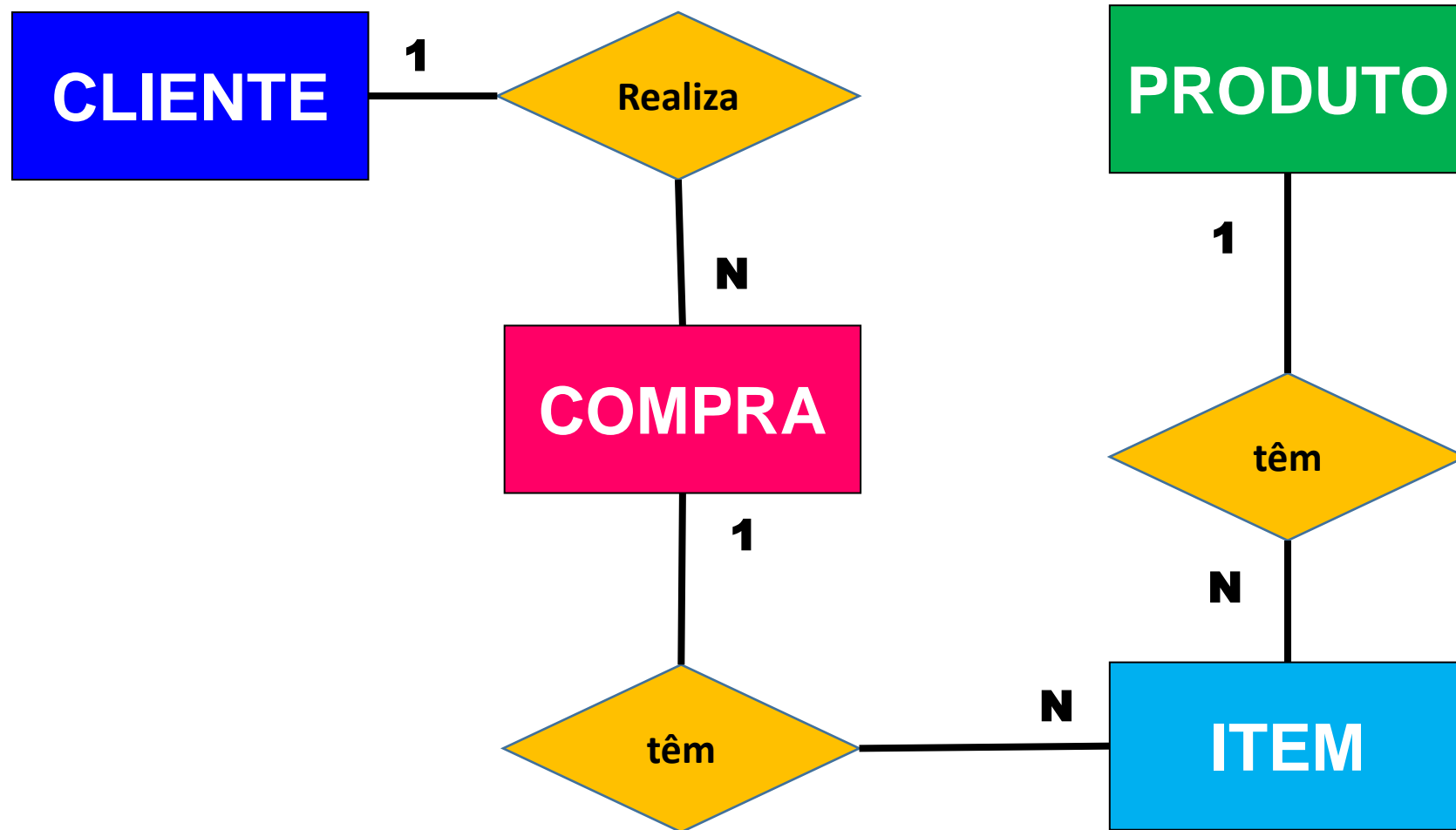
SeuNome_SobrenomeSql

```
/*CRIAR uma base de dados completa */  
CREATE DATABASE `Sergio_Silveira`
```

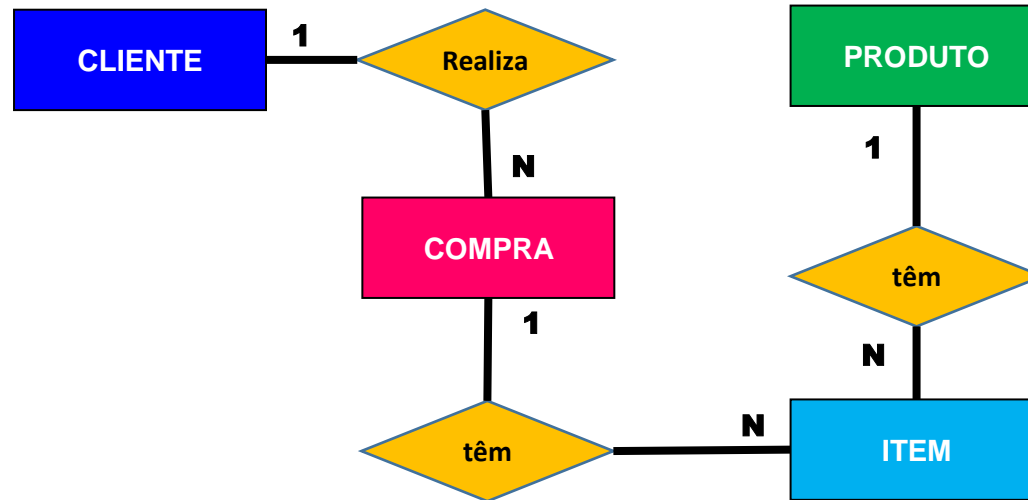
➤ **Crie as tabelas dentro do banco**

Conforme as imagens do
Próximo SLIDES

7. Atividade



7. Atividade



**Com base no D.E.R
acima crie as tabelas**

7. Atividade – Criando Tabela

| CLIENTE | |
|-----------|----------------------|
| CAMPO | TIPO |
| CPF | INTEGER PK – NOT NUL |
| NOME | VARCHAR(48) |
| RUA | VARCHAR (30) |
| NUMERO | INTEGER |
| BAIRRO | VARCHAR (30) |
| CIDADE | VARCHAR (20) |
| UF | CHAR (2) |
| SEXO | CHAR (1) |
| PROFISSAO | VARCHAR (50) |

7. Atividade – Script

```
CREATE TABLE CLIENTE (  
    cpf integer not null primary key,  
    nome varchar(48),  
    rua varchar(30),  
    numero integer,  
    bairro varchar(30),  
    cidade varchar(20),  
    estado char(2),  
    sexo char(1),  
    profissao varchar(50) );
```

7. Atividade – Criando Tabela

| PRODUTO | |
|-----------|----------------------|
| CAMPO | TIPO |
| CODPROD | INTEGER PK – NOT NUL |
| DESCRICAO | VARCHAR(25) |
| LOTE | VARCHAR (10) |
| VALIDADE | DATE |
| VALOR | DECIMAL (10,2) |

7. Atividade – Script

```
CREATE TABLE PRODUTO (  
    codprod integer not null primary key,  
    descricao varchar(25),  
    lote varchar(10),  
    validade date,  
    valor rdecimal (10,2)  
);
```

7. Atividade – Criando Tabela

| COMPRA | |
|----------------|----------------------|
| CAMPO | TIPO |
| CODCOMPRA | INTEGER PK – NOT NUL |
| CPF | INTEGER FK |
| DATA_COMPRA | DATE |
| TIPO_PAGAMENTO | CHAR (1) |

7. Atividade – Script

```
CREATE TABLE COMPRA (  
    codcompra integer not null primary key,  
    cpf integer,  
    data_compra date,  
    tipo_pagamento char(1),  
    foreign key (cpf) references cliente (cpf)  
);
```

7. Atividade – Criando Tabela

| ITEM | |
|-----------|----------------------|
| CAMPO | TIPO |
| CODITEM | INTEGER PK – NOT NUL |
| CODCOMPRA | INTEGER – FK |
| CODPROD | INTEGER - FK |
| QUATIDADE | INTEGER |

7. Atividade – Script

```
CREATE TABLE ITEM (  
    coditem integer not null primary key,  
    codcompra integer,  
    codprod integer,  
    quantidade integer,  
    foreign key (codcompra) references compra(codcompra),  
    foreign key (codprod) references produto (codprod)  
);
```

7. Atividade – Inserindo Registros

Inserir 12 registros na tabela CLIENTE

```
INSERT INTO cliente  
  (cpf, nome, rua, numero, bairro, cidade, estado, sexo, profissao)  
VALUES  
  ('11111', 'Pedro', 'Rua XV', '125', 'Alto XV', 'Curitiba', 'PR', 'M', 'militar')  
;
```

ou

```
INSERT INTO cliente  
VALUES  
  ('11111', 'Pedro', 'Rua XV', '125', 'Alto XV', 'Curitiba', 'PR', 'M', 'militar')  
;
```

7. Atividade – Inserindo Registros

**ENVIAR o SCRIPT de
INSERT
PARA OS ALUNOS**

**CLIQUE AQUI
PARA VIZUALIZAR O ARQUIVO**

1. Consultar todos os registros na tabela **CLIENTE**, **PRODUTO**, **COMPRA**

RESPOSTA

```
SELECT * FROM CLIENTE;
```

```
SELECT * FROM PRODUTO;
```

```
SELECT * FROM COMPRA;
```


Colocar a imagem da consulta aqui!

2. Consultar todos os registros na tabela **CLIENTE**, cujo número da residência seja maior que 10

RESPOSTA

```
SELECT * FROM CLIENTE  
WHERE NUMERO > 10;
```

Colocar a imagem da consulta aqui!

EXERCICIO

8. Exercício - ORIENTAÇÃO

Responda as consultas solicitadas no arquivo enviado pelo TEAMS;

Coloque o código SQL como resposta para cada consulta solicitada;

Não esqueça de preencher o cabeçalho;

Salve no formato PDF.

[Siga as orientações do arquivo. \(clique aqui e veja arquivo\)](#)

8. Exercício - ORIENTAÇÃO

Enviar o arquivo!
Clique aqui para abrir o arquivo

Apresente os comandos SQL para Consultas:

1 – Todos os produtos comprados com quantidade maior que 12.

8. Exercício

RESPOSTA

Apresente os comandos SQL para Consultas:

2 – Nome dos clientes que compraram mais de 23 peças de queijo.

Apresente os comandos SQL para Consultas:

3 – Cidade onde moram as pessoas que compram produtos com validade sempre menor que 3 meses.

Apresente os comandos SQL para Consultas:

4 - Todos os clientes cuja cidade é
“Guarapuava” e sexo masculino.

Apresente os comandos SQL para Consultas:

5 – Sexo das pessoas que moram na cidade que começa com a letra “G” e com a letra “C”.

Apresente os comandos SQL para Consultas:

6 – Quantidade total de produtos adquiridos com preço maior que R\$ 10,00.

Apresente os comandos SQL para Consultas:

- 7 - Nome das pessoas que compraram mais de 5 peças de queijo e mais de 3 litros de leite.

Apresente os comandos SQL para Consultas:

8 – Cidade onde moram os clientes, em ordem alfabética crescente.

Apresente os comandos SQL para Consultas:

9 – Profissão de todos os clientes que são professores, engenheiros, ou gestores.

Apresente os comandos SQL para Consultas:

10 – Nome da rua dos clientes que compram queijo com valor maior que R\$ 5,00 e menor que R\$ 25,00.

Apresente os comandos SQL para Consultas:

11 – Nome, Profissão e Sexo dos clientes que comprem mais de **10** litros de leite cujo valor esteja entre R\$ 1,00 e R\$ **4,75**.

Apresente os comandos SQL para Consultas:

12 - Nome, Profissão e Sexo dos clientes que compram mais de **10** litros de leite cujo valor esteja entre R\$ 1,00 e R\$ **4,75**, pelo nome decrescente.

Apresente os comandos SQL para Consultas:

- 13 – Profissão dos clientes que compram leite e queijo, ordenado pelo Nome em ordem crescente.

Apresente os comandos SQL para Consultas:

14 – Produtos comprados pelos clientes que moram em Curitiba e que comprem em quantidade maior que 5 unidades.

Apresente os comandos SQL para Consultas:

15 – Soma de todas as compras realizadas pelos clientes que moram em Curitiba.

Apresente os comandos SQL para Consultas:

16 – Lote dos produtos comprados pelos clientes cuja profissão seja professor.

Apresente os comandos SQL para Consultas:

17 – Validade dos lotes cuja venda foi realizada para a cidade de Guarapuava.

Apresente os comandos SQL para Consultas:

18 – Todos os lotes de todos os produtos.

Apresente os comandos SQL para Consultas:

19 – As quantidades de queijo compradas pelos clientes de Guarapuava.

Apresente os comandos SQL para Consultas:

20 – A data e a quantidade de leite compradas pelos clientes que moram em Curitiba ou em Guarapuava.

Apresente os comandos SQL para Consultas:

21 – Nome dos clientes que compraram queijo, bem como a data da compra e quantidade.

Apresente os comandos SQL para Consultas:

22 – Nome e cidade dos clientes, data da compra e tipo de pagamento, quantidade comprada e descrição dos produtos **em ordem decrescente por nome.**

RESPOSTA NO PROXIMO SLIDE

8. Exercício

RESPOSTA

8. Exercício

RESPOSTA

8. Exercício

RESPOSTA

Apresente os comandos SQL para Consultas:

- 23 – Compras efetuadas no **primeiro bimestre** do ano e **nome do cliente**.

8. Exercício

RESPOSTA

Apresente os comandos SQL para Consultas:

24 – Nome e lote de todos os produtos que foram comprados com quantidade maior **IGUAL a 4**.

Apresente os comandos SQL para Consultas:

25 – Lote, validade e descrição dos produtos que não foram comprados.

FIM

8. REFERENCIAS

Slide Projeto Conceitual de B.D - Crysthiane Carvalho
paola@spei.br

Sistema de Banco de Dados.

Abraham Silberschatz; Henry F. Korth; S. Sudarshan.
Capítulo 4: SQL - São Paulo: Makron Books, 3ª ed.,
1999.