

# PROGRAMADOR DE SISTEMAS

## UC 2 - Implementar Banco de Dados

### Aula 5

- Data Definition Language (DDL).
- Data Manipulation Language (DML).

Cesar Ricardo Velasque Trindade

Senac - Jaraguá do Sul  
cesar.trindade@prof.sc.senac.br

3 de abril de 2025



## 1 Introdução

- Data Definition Language (DDL) - Linguagem de Definição de Dados.
- Data Manipulation Language (DML) - Linguagem de Manipulação de Dados

## 2 Conclusão

Na aula de hoje continuaremos com o conteúdo sobre DDL e DML no SQL - **Structured Query Language**.



EMPRESA

Funcionários



Departamentos



Recursos Humanos



Financeiro



Almoxarifado



Montagem de Motores

- Relatório de funcionários e departamentos da empresa.

FUNCIONARIO	
codigo	nome
1001	Ana Lucia
1002	Breno Caldas
1003	Laercio Dias
1004	Carlos Ruiz
1005	Jeferson Deoclecio
1006	Guilherme Bras
1007	Fernanda Micaela
1008	Fabiane Mendes
1009	Leticia Alves
1010	Graziele Lima
1011	Carla Maria Fagundes
1012	Janaina Pires
1013	Leonardo Berto

DEPARTAMENTO	
codigo	nome
1	Recursos Humanos
2	Financeiro
3	Almoxarifado
4	Montagem de Motores

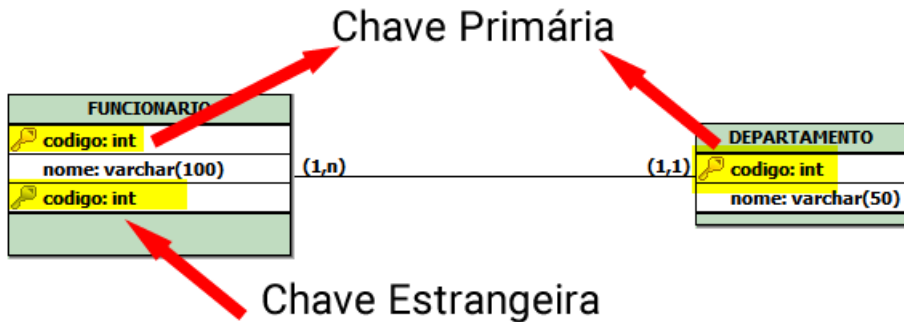
- Relatório de funcionários da empresa e locais de trabalho.

codigo	nome	departamento
1001	Ana Lucia	Financeiro
1002	Breno Caldas	Almoxarifado
1003	Laercio Dias	Recursos Humanos
1004	Carlos Ruiz	Montagem de Motores
1005	Jeferson Deoclecio	Financeiro
1006	Guilherme Bras	Recursos Humanos
1007	Fernanda Micaela	Montagem de Motores
1008	Fabiane Mendes	Montagem de Motores
1009	Leticia Alves	Almoxarifado
1010	Graziele Lima	Financeiro
1011	Carla Maria Fagundes	Financeiro
1012	Janaina Pires	Almoxarifado
1013	Leonardo Berto	Recursos Humanos

- **Chave Estrangeira.**

- ❶ Uma **chave estrangeira** é um atributo em uma tabela que faz referência à **chave primária** de outra tabela.
- ❷ A **chave primária** usada por uma **chave estrangeira** é também conhecida como **parent key**.
- ❸ **Integridade referencial** quer dizer que só é possível inserir valores na chave estrangeira que já exista na **parent table**.

- Relacionamento entre tabelas.





- Relacionamento entre tabelas.


```
11 • create table departamento
12   (
13     codigo int primary key,
14     nome varchar(50) not null
15   );
16
```



Criar primeiro a tabela em que sua chave primária será a chave estrangeira na outra tabela.

- Relacionamento entre tabelas.

```
18 • create table funcionario
19 (
20     codigo int primary key,
21     nome varchar(100) not null,
22     id_codigodepartamento int not null,
23     foreign key (id_codigodepartamento) references departamento(codigo)
24 );
```



Criar a tabela que terá a chave primária da outra tabela (chave estrangeira).

- Inserir os dados na tabela Departamento.

```
27 • insert into departamento
28   values (1, 'Recursos Humanos'),
29          (2, 'Financeiro'),
30          (3, 'Almoxarifado'),
31          (4, 'Montagem de motores');
32
```

- Inserir os dados na tabela Funcionário.

```
34 • insert into funcionario
35     values (1001, 'Ana Lucia', 2),
36            (1002, 'Breno Caldas', 3),
37            (1003, 'Laercio Dias', 1),
38            (1004, 'Carlos Ruiz', 4),
39            (1005, 'Jeferson Deoclecio', 2),
40            (1006, 'Guilherme Braz', 1),
41            (1007, 'Fernanda Micaela', 4),
42            (1008, 'Fabiane Mendes', 4),
43            (1009, 'Leticia Alves', 3),
44            (1010, 'Graziele Lima', 2),
45            (1011, 'Carla Maria Fagundes', 2),
46            (1012, 'Janaina Pires', 3),
47            (1013, 'Leonardo Berto', 1);
```

# SQL

## Declarações DML

### **Instrução SELECT**

- **SELECT \***

- O comando **SELECT** consiste de várias cláusulas que devem ser colocadas numa ordem prefixada.
- Em geral, fica mais legível colocando uma cláusula por linha.
- Considere o seguinte exemplo:

```
select * from funcionario;
```

- O que essa instrução em SQL retorna?

- **SELECT DISTINCT**

- O comando **SELECT DISTINCT** retorna apenas valores distintos (diferentes) no conjunto de resultados.
- Considere o seguinte exemplo:

```
select distinct id_codigodepartamento  
from funcionario;
```

- O que essa instrução em SQL retorna?

## OPERADORES DE COMPARAÇÃO

<b>=</b>	<b>igual</b>
<b>&lt;&gt;</b>	<b>diferente de</b>
<b>&gt;</b>	<b>maior que</b>
<b>&lt;</b>	<b>menor que</b>
<b>&gt;=</b>	<b>maior ou igual que</b>
<b>&lt;=</b>	<b>menor ou igual que</b>



## ● OPERADORES DE COMPARAÇÃO

- Os operadores de comparação são operadores que possuem o objetivo de analisar os valores de uma expressão e retornar um valor **booleano**, ou seja, verdadeiro (1) ou falso (0).
- Considere o seguinte exemplo:

```
select nome_livro  
from tbl_livro  
where data_pub < '1950-10-15';
```

- O que essa instrução em SQL retorna?

- CONECTORES LÓGICOS AND

- Usado na cláusula **WHERE** para compor uma expressão booleana.

```
select *  
from tbl_livro  
where id_autor = 5  
and data_pub > '1950-01-01';
```

- O que essa instrução em SQL retorna?

- CONECTOR LÓGICO OR

- Usado na cláusula **WHERE** para compor uma expressão booleana.

```
select *  
from tbl_livro  
where id_autor = 1  
or data_pub < '1950-01-01';
```

- O que essa instrução em SQL retorna?

- OPERADOR BETWEEN

- Utilizado para realizar uma consulta que seleciona registros entre um espaço.

```
select id_livro, nome_livro  
from tbl_livro  
where data_pub  
between '1920-01-01'  
and '1930-01-01';
```

- O que essa instrução em SQL retorna?

- OPERADOR IN

- Usado para verificar se um valor está contido (ou não) num conjunto de constantes.

```
select nome_livro  
from tbl_livro  
where id_editora in (2, 3, 4);
```

- O que essa instrução em SQL retorna?

- OPERADOR NOT IN

- Usado para verificar se um valor está contido (ou não) num conjunto de constantes.

```
select nome_livro  
from tbl_livro  
where id_editora not in (1, 5);
```

- O que essa instrução em SQL retorna?

## ● OPERADOR LIKE

- O operador **LIKE** é usado em uma cláusula **WHERE** para pesquisar um padrão especificado em uma coluna.
- O sinal de porcentagem (%) representa zero, um ou vários caracteres.

```
SELECT nome_livro, data_pub  
FROM tbl_livro  
WHERE nome_livro LIKE 'd%';
```

- O que essa instrução em SQL retorna?

- OPERADOR LIKE

```
SELECT nome_livro, data_pub  
FROM tbl_livro  
WHERE nome_livro LIKE '%a';
```

- O que essa instrução em SQL retorna?



- **CLÁUSULA ORDER BY**

- A cláusula **ORDER BY** define critérios de ordenação de linhas a partir de uma ou mais colunas do resultado do comando **SELECT**. É possível especificar múltiplas colunas para ordenação.

```
SELECT nome_autor, sobrenome_autor  
FROM tbl_autor  
ORDER BY nome_autor;
```

- O que essa instrução em SQL retorna?

## ● OPERAÇÃO DE JUNÇÃO

- Uma cláusula **JOIN** é usada para combinar linhas de duas ou mais tabelas, com base em uma coluna relacionada entre elas.
- Vejamos agora como fazer a importante operação de junção de duas tabelas. A seguinte consulta pode ser expressa com esta operação:
- **Selecione o código, nome do funcionário e o departamento em que trabalha (utilize as tabelas criadas na aula, para o banco de dados empresa).**



- OPERAÇÃO DE JUNÇÃO

```
SELECT funcionario.codigo, funcionario.nome, departamento.nome  
FROM funcionario  
INNER JOIN departamento  
ON funcionario.id_codigodepartamento=departamento.codigo;
```

## ● FUNÇÕES QUE PRODUZEM VALORES AGREGADOS

- SQL possui um conjunto de funções para obter estatísticas sobre os valores de uma dada coluna. Elas são chamadas de **funções agregado**.
- `count (nome_coluna)`: obtém o número de valores na coluna especificada.
- `min (nome_coluna)`: obtém o valor mínimo da coluna.
- `max (nome_coluna)`: obtém o valor máximo da coluna.
- `sum (nome_coluna)`: soma os valores da coluna.
- `avg (nome_coluna)`: obtém a média aritmética dos valores da coluna.
- **Observações:**
  - `count (*)`: conta o número de linhas.
  - `sum` e `avg`: só admitem colunas com valores numéricos.

- **COUNT**

```
SELECT COUNT (*)  
FROM tbl_livro;
```

Qual é a finalidade dessa consulta?

- **MAX**

```
SELECT MAX (premio)  
FROM torneos;
```

Qual é a finalidade dessa consulta?

- **AVG**

```
SELECT AVG (premio)  
FROM torneos;
```

**Qual é a finalidade dessa consulta?**

