



**INSTITUTO FEDERAL**  
Sul de Minas Gerais

Campus  
Poços de Caldas

# ***Banco de Dados - Linguagem SQL***

***Profa. Msc. Giselle Cristina Cardoso***

***2020***

## Sumário

Apresentação da disciplina .....	3
Introdução .....	4
Definição .....	4
Linguagem de Definição de Dados (DDL).....	4
Create Database .....	5
Use Database .....	5
Show Databases.....	5
Create Table.....	5
Alter Table.....	6
Drop Table .....	6
Show Tables .....	7
Describe Table .....	7
Exercícios Resolvidos .....	7
Exercícios Propostos .....	8
Linguagem de Manipulação de Dados (DML).....	9
Insert.....	9
Exercícios Resolvidos .....	9
Exercícios Propostos .....	10
Update .....	11
Exercícios Resolvidos .....	11
Exercícios Propostos .....	11
Delete.....	12
Exercícios Resolvidos .....	12
Exercícios Propostos .....	12
Linguagem de Consulta de Dados (DQL) .....	13
SELECT (utilizando uma tabela) .....	13
Exercícios Resolvidos .....	14
Exercícios Propostos .....	15
SELECT (utilizando mais de uma tabela).....	16
Exercícios Resolvidos .....	16
Exercícios Propostos .....	16
Referências Bibliográficas.....	17

## ***Apresentação da disciplina***

A disciplina Banco de Dados tem como objetivo principal ensinar você a modelar um Banco de Dados, criá-lo e manipulá-lo em um Sistema Gerenciador de Banco de Dados. Para que o objetivo seja alcançado a disciplina será dividida em quatro partes:

1. **Introdução a Banco de Dados:** será apresentada a definição de Dados, Informações, Banco de Dados, Modelo de Dados, Sistemas Gerenciador de Banco de Dados, linguagem padrão de Banco de Dados e os usuários de um Banco de Dados.
2. **Modelo Entidade Relacionamento:** será abordado como criar este modelo lógico do Banco de Dados e seus principais componentes.
3. **Modelo Relacional:** nesta parte irá apresentar como mapear o Modelo Entidade Relacionamento para o Modelo Relacional, apresentando as principais características deste novo modelo.
4. **Linguagem SQL:** para finalizar a disciplina será mostrado como criar, popular e consultar um Banco de Dados em um Sistema Gerenciador de Banco de Dados, utilizando a Linguagem SQL. Esta parte será o foco desta apostila.

Para um bom aproveitamento desta disciplina sugiro que todos os exemplos, exercícios resolvidos e propostos sejam implementados e executados em um Sistema Gerenciador de Banco de Dados.

O conteúdo desta apostila tem como referência os livros:

- *Sistemas de Banco de Dados: uma Abordagem Introdutória e Aplicada*, de autoria de Virgínia Cardoso e Giselle Cardoso, publicado pela Saraiva, em 2012.
- *Linguagem SQL: Fundamentos e Prática*, de autoria de Virgínia Cardoso e Giselle Cardoso, publicado pela Saraiva, em 2013.

## ***Introdução***

Um Banco de Dados deve possuir uma estrutura bem elaborada para armazenar os dados com segurança. O Sistema Gerenciador de Banco de Dados (SGBD) é utilizado para armazenar os dados do Banco de Dados de maneira segura e correta. A Linguagem SQL é a linguagem mais utilizada nos SGBDs.

## ***Definição***

A Linguagem SQL foi criada na década de 1970, nos laboratórios da IBM na Califórnia, por Codd. Ela é uma linguagem declarativa e na execução de qualquer comando o que importa é o resultado.

A Linguagem SQL é dividida nas seguintes partes:

- A DDL (linguagem de definição de dados) permite a criação do Banco de Dados, das tabelas, dos relacionamentos entre as tabelas, bem como alterar ou excluir os mesmos;
- A DML (linguagem de manipulação de dados) permite a inclusão, alteração e exclusão de dados em um Banco de Dados;
- A DCL (linguagem de controle dos dados) é responsável pelo controle de acesso dos usuários ao Banco de Dados;
- A DTL (linguagem de transação de dados) é responsável pelas transações no Banco de Dados;
- A DQL (linguagem de consulta de dados) permite a criação e execução de consultas no Banco de Dados.

Nessa disciplina o foco será nas três principais partes que compõem a Linguagem SQL, a DDL, a DML e a DQL.

## ***Linguagem de Definição de Dados (DDL)***

Para criar o Banco de Dados são utilizados os comandos da Linguagem de Definição de Dados (DDL) da Linguagem SQL. Os principais comandos dessa linguagem serão apresentados abaixo.

## Create Database

O comando **CREATE DATABASE** é responsável pela criação do Banco de Dados.

Sintaxe: **CREATE DATABASE** <nome do banco de dados>;

## Use Database

O comando **USE** indica qual Banco de Dados é selecionado para trabalhar.

Sintaxe: **USE** <nome do banco de dados>;

## Show Databases

Comando responsável por apresentar os Bancos de Dados existentes no SGBD.

Sintaxe: **SHOW DATABASES**;

## Create Table

O comando **CREATE TABLE** é responsável pela criação das tabelas, incluindo as chaves primárias e estrangeiras e os relacionamentos existentes entre as tabelas do Banco de Dados.

Sintaxe: **CREATE TABLE** <nome da tabela>(  
 <nome do atributo> tipo [NOT NULL],...,  
**PRIMARY KEY** (<nome do atributo1>,< nome do atributo2>,...),  
**FOREIGN KEY** (<nome do atributo>) **REFERENCES** <nome da tabela>(<nome do atributo>));

Onde:

- **PRIMARY KEY**: indica os atributos que formam a chave primária;
- **FOREIGN KEY**: indica os atributos que formam a chave estrangeira, o nome da tabela e do campo desse relacionamento.
- **Tipo**: determina o tipo do campo.

Tipo	Descrição
int	Números inteiros.
float	Números reais (7 dígitos).
double	Números reais (15 a 16 dígitos).
char (x)	Caracteres (comprimento fixo, determinado pelo valor de x).
varchar (x)	Caracteres (comprimento variável, onde o comprimento máximo é determinado pelo valor de x).

## Alter Table

O comando **ALTER TABLE** permite modificar a estrutura de uma tabela já definida. Podendo adicionar ou excluir uma coluna à tabela ou modificar uma coluna existente.

*Sintaxe:* **ALTER TABLE** <nome da tabela>

```
[ADD <nome do atributo> <tipo de dado>; |  
ADD PRIMARY KEY(<nome do atributo>; |  
ADD FOREIGN KEY(<nome do atributo>  
REFERENCES <nome da tabela>(<nome do atributo>; |  
CHANGE <nome do atributo><nome do atributo> <tipo de dado>; |  
CHANGE <nome do atributo><nome do atributo> <nome do atributo> <tipo de dado>; |  
CHANGE <nome do atributo><nome do atributo><nome do atributo> <tipo de dado>; |  
RENAME <nome da tabela>; |  
DROP <nome do atributo>;]
```

Onde:

- **ADD** <nome do atributo> <tipo de dado>: utilizado para criar uma nova coluna na tabela;
- **ADD PRIMARY KEY**(<nome do atributo>): utilizado para inserir uma chave primária na tabela;
- **ADD FOREIGN KEY**(<nome do atributo>) **REFERENCES** <nome da tabela>(<nome do atributo>): utilizado para inserir uma chave estrangeira na tabela;
- **CHANGE** <nome do atributo><nome do atributo> <tipo de dado>: utilizado para alterar o nome da coluna;
- **CHANGE** <nome do atributo><nome do atributo> <nome do atributo> <tipo de dado>: utilizado para alterar o tipo de dado da coluna;
- **CHANGE** <nome do atributo><nome do atributo> <nome do atributo> <tipo de dado>: utilizado para alterar o nome da coluna e o tipo de dado da coluna;
- **RENAME** <nome da tabela>: utilizado para renomear a tabela;
- **DROP** <nome do atributo>: utilizado para apagar uma determinada coluna da tabela.

## Drop Table

O comando **DROP TABLE** tem como função a remoção de uma tabela.

*Sintaxe:* **DROP TABLE** <nome da tabela>;



- Criar a tabela livros, sem a chave estrangeira CódigoEditora (a tabela Editora ainda não foi criada).

```
CREATE TABLE livros (codigo int,  
                        titulo varchar (50),  
                        edicao int,  
                        genero varchar (20),  
                        anoPublicacao int,  
                        PRIMARY KEY (codigo));
```

- Alterar a tabela usuários para incluir a chave estrangeira CódigoLivro.
  - Incluir o campo CódigoLivro do tipo int, pois na tabela livros este campo é do tipo int.

```
ALTER TABLE usuarios ADD codigoLivro int;
```

- Determinar que CódigoLivro é uma chave estrangeira.

```
ALTER TABLE usuarios ADD FOREIGN KEY (codigoLivro) REFERENCES  
livros(codigo);
```

- Criar a tabela retirar.

```
CREATE TABLE retirar (cpf int,  
                       codigoLivro int,  
                       PRIMARY KEY (cpf, codigoLivro),  
                       FOREIGN KEY (cpf) REFERENCES usuarios (cpf),  
                       FOREIGN KEY (codigoLivro) REFERENCES livros (codigo));
```

### ***Exercícios Propostos***

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

1. Criar a tabela Editora.
2. Alterar a tabela Livros, incluindo a chave estrangeira CódigoEditora.
3. Criar a tabela Autores e Funcionários.
4. Criar as tabelas RetirarFuncionário e Escreve.
5. Exclua a tabela Escreve.



## Linguagem de Manipulação de Dados (DML)

A Linguagem de Manipulação de Dados é responsável pela inclusão, alteração e exclusão dos dados nas tabelas do Banco de Dados. Ela possui três comandos, o **INSERT**, **UPDATE** e **DELETE**.

### Insert

O comando **INSERT** é responsável pela inclusão dos dados nas tabelas do Banco de Dados.

Sintaxe: **INSERT INTO** <nome databela> (campo1, campo2, ...)  
**VALUES** (valor1\_campo1, valor2\_campo2, ...);

### Exercícios Resolvidos

1. Dada à tabela Editora com dados, insira os dados na tabela criada.

Editora		
Código	Nome	Contato
1	Nova	Ana
2	Atual	-
3	Moderna	-

#### Solução:

Para inserir os dados na tabela, temos algumas opções:

- a) Nesse caso, o NULL é utilizado para deixar o campo em branco.  
**INSERT INTO** editora (codigo, nome, contato) **VALUES** (1, 'Nova', 'Ana');  
**INSERT INTO** editora (codigo, nome, contato) **VALUES** (2, 'Atual', NULL);  
**INSERT INTO** editora (codigo, nome, contato) **VALUES** (3, 'Moderna', NULL);
- b) **INSERT INTO** editora (codigo, nome, contato) **VALUES** (1, 'Nova', 'Ana'),  
(2, 'Atual', NULL),  
(3, 'Moderna', NULL);
- c) Como na criação da tabela os campos foram criados na seguinte sequencia: código, nome e contato e os dados serão inseridos na mesma sequencia, nesse caso pode-se excluir os nomes dos campos.  
**INSERT INTO** editora **VALUES** (1, 'Nova', 'Ana');  
**INSERT INTO** editora **VALUES** (2, 'Atual', NULL);  
**INSERT INTO** editora **VALUES** (3, 'Moderna', NULL);

- d) **INSERT INTO** editora **VALUES** (1, 'Nova', 'Ana'), (2, 'Atual', NULL), (3, 'Moderna', NULL);
- e) **INSERT INTO** editora (**codigo, contato, nome**) **VALUES** (1, 'Ana', 'Nova');  
**INSERT INTO** editora (**codigo, nome**) **VALUES** (2, 'Atual');  
**INSERT INTO** editora (**nome, codigo**) **VALUES** ('Moderna', 3);
- f) **INSERT INTO** editora (codigo, nome, contato) **VALUES** (1, 'Nova', 'Ana');  
**INSERT INTO** editora (codigo, nome) **VALUES** (2, 'Atual'), (3, 'Moderna');

2. Dada a tabela Livros com dados, insira os dados na tabela criada.

Livros					
<u>Código</u>	Título	Edição	Gênero	Ano de Publicação	CodigoEditora
1	A Terra	1	Romance	2020	1
2	A Lua	1	Terror	2020	1
3	O Mar	1	-	2020	3

**Solução:**

**INSERT INTO** livros **VALUES** (1, 'A Terra', 1, 'Romance', 2020, 1),  
 (2, 'A Lua', 1, 'Terror', 2020, 1),  
 (3, 'O Mar', 1, NULL, 2020, 3);

## Exercícios Propostos

1. Dada à tabela Autores, insira os dados na tabela criada.

Autores		
<u>Código</u>	Nome	Nacionalidade
10	Luiz	-
11	Antonio	-
12	Laura	-

2. Dada à tabela Funcionários, insira os dados na tabela criada.

Funcionários			
<u>CPF</u>	Nome	Função	Salário
345.678	Beatriz	TI	2.000,00
044.033	Sílvia	DBA	1.000,00
098.890	Maria	TI	1.500,00

3. Dada à tabela Usuários, insira os dados na tabela criada.

Usuários				
<u>CPF</u>	Nome	Endereço	Telefone	CódigoLivro
123.456	Ana	Rua A, 10	3712-1010	1
789.999	Maria	Rua WJ,30	3722-2020	1

876.985	José	Rua Dois, 33	3713-9090	2
---------	------	--------------	-----------	---

4. Dada à tabela Retirar, insira os dados na tabela criada.

Retirar	
CPF	Código Livro
123.456	1
789.999	1
789.999	3

5. Dada à tabela RetirarFuncionários, insira os dados na tabela criada.

RetirarFuncionário	
CPFFuncionário	CódigoLivro
044.033	1
345.678	1
345.678	3

6. Dada à tabela Escreve, insira os dados na tabela criada.

Escreve	
Código Livro	CódigoAutor
1	12
1	10
3	10

## Update

Comando utilizado para atualizar os dados nas tabelas do Banco de Dados.

Sintaxe: **UPDATE** <nome-da-tabela>

**SET** coluna = valor [, coluna = valor,...]

[**WHERE** condição];

## Exercícios Resolvidos

3. Altere na tabela Usuários o campo telefone para 035 (todos os usuários).

**Solução:** **UPDATE** usuarios **SET** telefone='035';

4. Altere uma tupla na tabela Usuários e modifique o nome para André.

**Solução:** **UPDATE** usuarios **SET** nome='Andre' **WHERE** cpf = 123456;

## Exercícios Propostos

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

7. Alterar na tabela Funcionários os funcionários que possuem salário maior que R\$1.200,00 para o cargo de DBA.
8. Na tabela Autores modifique todas as nacionalidades para brasileira.
9. Na tabela Livros altere todos os anos de publicação dos livros da editora 1 para 2019.
10. Altere uma tupla na tabela Autores.
11. Na tabela Usuários altere todas as tuplas para que o campo telefone fique em branco.

## **Delete**

Comando utilizado para apagar os dados inseridos nas tabelas do Banco de Dados.

Sintaxe: **DELETE FROM** <nome-da-tabela> [**WHERE** condição];

## **Exercícios Resolvidos**

5. Excluir todos os dados da tabela Escreve.

**Solução:** **DELETE FROM** escreve;

6. Excluir todos os dados da tabela Livros, onde o Código da Editora é 1.

**Solução:** **DELETE FROM** livros **WHERE** codigoEditora=1;

7. Excluir na tabela Autores, os autores de nacionalidade brasileira e espanhola.

**Solução:**

**DELETE FROM** autores **WHERE** nacionalidade = 'brasileira' or nacionalidade = 'espanhola';

8. Excluir na tabela Funcionários, os funcionários com salários inferiores a 1.600,00 e superiores a 800,00.

**Solução:** **DELETE FROM** funcionarios **WHERE** salario < 1600 and salario >800;

## **Exercícios Propostos**

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

12. Excluir todos os registros da tabela Autores.
13. Excluir na tabela Livros o livro com título “O Mar”.
14. Excluir um **conjunto** de tuplas da tabela Retirar.
15. Excluir na tabela Livros os livros de gênero romance e terror.
16. Excluir na tabela Funcionários os funcionários que possuem função TI e salário maior que R\$1.000,00.

## Linguagem de Consulta de Dados (DQL)

### SELECT (utilizando uma tabela)

Comando utilizado para realizar consultas no Banco de Dados.

Sintaxe: **SELECT** [DISTINCT] <campo1, campo2, ...>  
**FROM** <nome-da-tabela>  
 [WHERE condições]  
 [ORDER BY <campo\_ordenacao> [ASC/DESC]]  
 [LIMIT <valor>];

No comando **SELECT** utilizamos alguns operadores e comandos para auxiliar na consulta. A função de cada operador e comando serão descritos nas tabelas a seguir.

Operadores		
Tipo	Operador	Descrição
Comparação	=	Igual
	!=	Diferente
	>	Maior
	>=	Maior ou Igual
	<	Menor
	<=	Menor ou Igual
	LIKE	Encontrar um caractere ou um conjunto de caracteres nos valores armazenados em um determinado campo da tabela. Esse operador utiliza a % como complemento.
Lógicos	AND	E lógico.
	OR	OU lógico.
	NOT	NÃO lógico.

Funções	
Nome	Descrição
COUNT(*)	Retorna o número de registros que tem na tabela.
SUM (nome_do_campo)	Retorna a soma de todos os valores armazenados no campo especificado.
AVG (nome_do_campo)	Retorna a média de todos os valores armazenados no campo especificado.
MIN (nome_do_campo)	Retorna o menor valor de todos os valores armazenados no campo especificado.
MAX (nome_do_campo)	Retorna o maior valor de todos os valores armazenados no campo especificado.

## ***Exercícios Resolvidos***

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

1. Listar todos os dados da tabela Funcionários.

**Solução:** **SELECT** cpf, nome, funcao, salario **FROM** funcionarios;

Ou, quando desejamos trazer todos os campos, podemos utilizar o **asterisco (\*)**.

**SELECT \* FROM** funcionarios;

2. Selecionar os nomes e funções dos funcionários.

**Solução:** **SELECT** nome, funcao **FROM** funcionarios;

3. Liste os nomes e funções dos funcionários chamados Maria.

**Solução:** **SELECT** nome, funcao **FROM** funcionarios **WHERE** nome = 'Maria';

4. Selecione os nomes e funções dos funcionários que o nome começa com a letra B.

**Solução:** **SELECT** nome, funcao **FROM** funcionarios **WHERE** nome **LIKE** 'B%';

5. Selecione os nomes e telefones dos usuários que tenham o dígito 2 no telefone.

**Solução:** **SELECT** nome, telefone **FROM** usuarios **WHERE** telefone **LIKE** '%2%';

6. Selecione os nomes e funções dos funcionários que o nome termina com a letra A.

**Solução:** **SELECT** nome, funcao **FROM** funcionarios **WHERE** nome **LIKE** '%a';

7. Selecione os nomes e funções dos funcionários que tem salário entre R\$1.000,00 e R\$2.000,00.

**Solução:**

**SELECT** nome, funcao **FROM** funcionarios **WHERE** salario > 1000 and salario<2000;

8. Selecione os nomes e funções dos funcionários ordenados por nome.

**Solução:** **SELECT** nome, funcao **FROM** funcionarios **ORDER BY** nome;

9. Selecione os nomes dos funcionários ordenados de forma descendente (ordem alfabética).

**Solução:** **SELECT** nome **FROM** funcionarios **ORDER BY** nome **DESC**;

10. Selecione os nomes dos funcionários, mas somente 2 funcionários.

**Solução:** **SELECT** nome **FROM** funcionarios **LIMIT** 2;

11. Selecione os cargos dos funcionários, eliminando repetições.

**Solução:** `SELECT DISTINCT funcao FROM funcionarios;`

12. Exibir os nomes dos funcionários e o salário acrescido de 10%.

**Solução:** `SELECT nome, (salario + salario*10/100) FROM funcionarios;`

13. Informar o número funcionários existentes nesta tabela.

**Solução:** `SELECT COUNT(*) FROM funcionarios;`

14. Informar a média de salários dos funcionários.

**Solução:** `SELECT AVG(salario) FROM funcionarios;`

15. Informar o menor salário dos funcionários.

**Solução:** `SELECT MIN(salario) FROM funcionarios;`

16. Informar o maior salário dos funcionários.

**Solução:** `SELECT MAX(salario) FROM funcionarios;`

17. Mostrar a soma dos salários dos funcionários.

**Solução:** `SELECT SUM(salario) FROM funcionarios;`

### ***Exercícios Propostos***

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

1. Liste os nomes, função e salários dos funcionários que são TI.
2. Selecione os títulos e gêneros dos livros lançados em 2020.
3. Selecione os nomes dos funcionários que salario maior que R\$1.000,00.
4. Selecione os nomes dos funcionários que salario menor ou igual R\$1.500,00.
5. Selecione nomes dos funcionários que tem nome que começam com M ou S.
6. Selecione nomes dos funcionários que tem nome que começam com M ou S e tem salário menor que R\$1.500,00.
7. Selecione nomes e cargos dos funcionários que não são DBA.
8. Selecione os títulos dos livros ordenados por ano de publicação.

## ***SELECT (utilizando mais de uma tabela)***

Comando utilizado para realizar consultas no Banco de Dados utilizando duas ou mais tabelas.

Sintaxe: **SELECT** [**DISTINCT**] <tabela1.campo1, tabela1.campo2, tabela2.campo1, ...>  
**FROM** <tabela1, tabela2, ...>  
[**WHERE** condições]  
[**ORDER BY** <campo\_ordenacao> [**ASC/DESC**]]  
[**LIMIT** <valor>];

## ***Exercícios Resolvidos***

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

18. Selecionar o nome dos usuários que retiraram livros.

**Solução:** **SELECT** usuarios.nome **FROM** usuarios, retirar **WHERE** usuarios.cpf = retirar.cpf;

19. Apresentar o nome dos livros e seus autores.

**Solução:** **SELECT** livros.titulo, autores.nome **FROM** livros, autores, escreve  
**WHERE** livros.codigo = escreve.codigoLivro  
**AND** escreve.codigoAutor = autores.codigo;

20. Apresentar o nome dos livros e suas editoras.

**Solução:** **SELECT** livros.titulo, editora.nome **FROM** livros, editora  
**WHERE** livros.codigoEditora = editora.codigo;

## ***Exercícios Propostos***

Utilize o Banco de Dados “Biblioteca” dos *exercícios resolvidos* para fazer os exercícios a seguir.

9. Mostrar o nome do usuário e o título do livro reservado por ele.

10. Mostrar os nomes dos funcionários e os títulos dos livros retirados por ele.

11. Mostrar os títulos dos livros retirados pelo usuário e o nome da editora de cada um dos livros.

12. Apresentar o nome do livro, nome do autor e nome da editora.



## ***Referências Bibliográficas***

- CARDOSO, Virginia Mara; CARDOSO, Giselle Cristina. **Sistemas de Banco de Dados**: uma abordagem introdutória e aplicada. 1. ed. São Paulo: Saraiva, 2012. 144 p. ISBN 8502162829.
- CARDOSO, Virginia Mara; CARDOSO, Giselle Cristina. **Linguagem SQL**: Fundamentos e Prática. 1. ed. São Paulo: Saraiva, 2013. 196 p. ISBN 8502200453.