

Introdução à Banco de Dados

Instrutor: Tarik Ponciano

Links da Disciplina

1. Discord: <https://discord.gg/wt5CVZZWJs>
2. Drive:
<https://drive.google.com/drive/folders/1hOl0DaPeAor7gnhKBUIlZ5n8lLRDNvUY?usp=sharing>
3. Github:
<https://github.com/TarikPonciano/Programador-de-Sistema-SENAC>

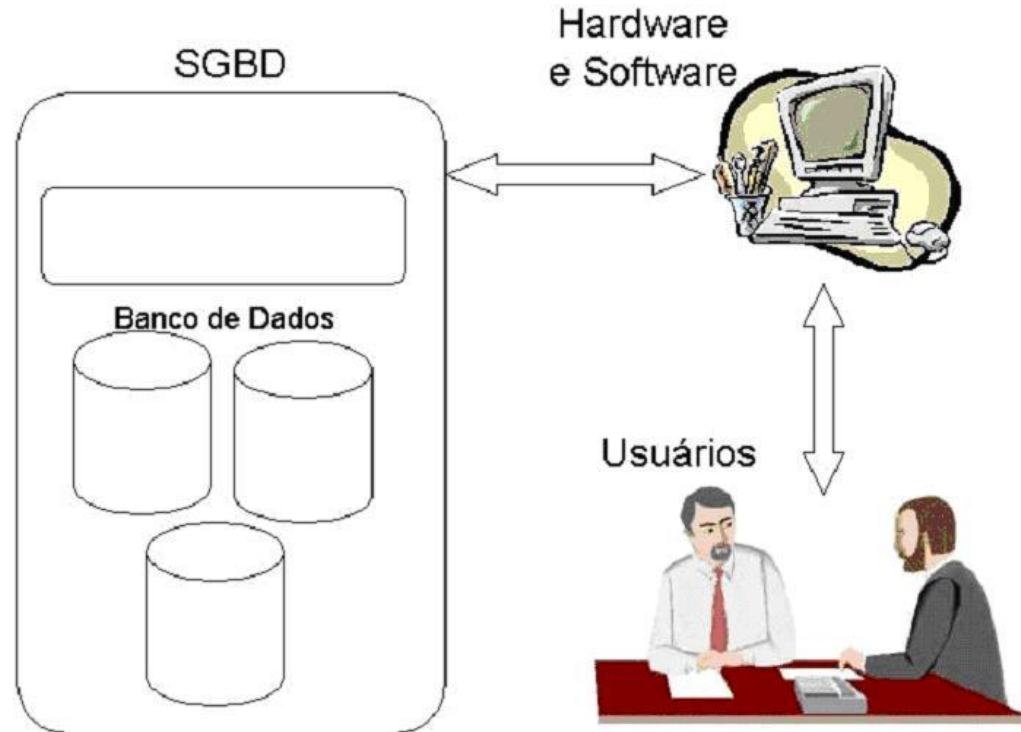
Banco de Dados – O que é?

Segundo Korth, um **banco de dados** “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

Podemos exemplificar situações clássicas como uma lista telefônica, um catálogo de CDs ou um sistema de controle de RH de uma empresa.

Já um sistema de gerenciamento de banco de dados (SGBD) é um software que possui recursos capazes de manipular as informações do banco de dados e interagir com o usuário. Exemplos de SGBDs são: Oracle, SQL Server, DB2, PostgreSQL, MySQL, o próprio Access ou Paradox, entre outros.

Sistema de Banco de Dados



Banco de Dados

Os objetivos de um **sistema de banco de dados** são o de isolar o usuário dos detalhes internos do banco de dados (promover a abstração de dados) e promover a independência dos dados em relação às aplicações, ou seja, tornar independente da aplicação, a estratégia de acesso e a forma de armazenamento.

Banco de Dados – Conceitos Gerais

- **Banco de Dados** - Representa o arquivo físico de dados, armazenado em dispositivos periféricos, onde estão armazenados os dados de diversos sistemas, para consulta e atualização pelo usuário.
- **Dado** - É o valor do campo quando é armazenado no Banco de Dados.
- **Tabelas Lógicas** - Representam as estruturas de armazenamento de dados (arquivos) dos sistemas.
- **S.G.D.B. (Sistema Gerenciador de Banco de Dados)** - É o software responsável pelo gerenciamento (armazenamento e recuperação) dos dados no Banco de Dados.
- **Conteúdo do campo** - É o valor do campo armazenado no Banco de Dados.
- **Modelo de Banco de Dados:** Modelo Relacional, Modelo Hierárquico e Modelo em Rede. Representa a estrutura física no qual o armazenamento dos dados foram projetados. O modelo identifica a estrutura interna de recuperação e armazenamento dos dados no qual o SGBD foi projetado.

Banco de Dados – Conceitos Gerais

- Um Bd é uma coleção de dados logicamente relacionados, com algum significado.
 - implica em que associações aleatórias de dados não possam ser chamadas de bases de dados.
- Um Bd é projetado, construído e preenchido com dados para um propósito específico.
 - Possui um grupo de usuários e algumas aplicações concebidas para atender a esses usuários .
- Um Bd é a representação de algum aspecto do mundo real.
 - nos leva a que, qualquer mudança na realidade, provocará mudanças nessa base de dados.

Banco de Dados – Conceitos Gerais

- **Visão Interna**

- É aquela vista pelo responsável pela manutenção e desenvolvimento do SGBD. Existe a preocupação com a forma de recuperação e manipulação dos dados dentro do Banco de Dados.

- **Visão Externa**

- É aquela vista pelo usuário que opera os sistemas aplicativos, através de interfaces desenvolvidas pelo analista (programas), buscando o atendimento de suas necessidades

Banco de Dados – Conceitos Gerais

■ Visão Conceitual

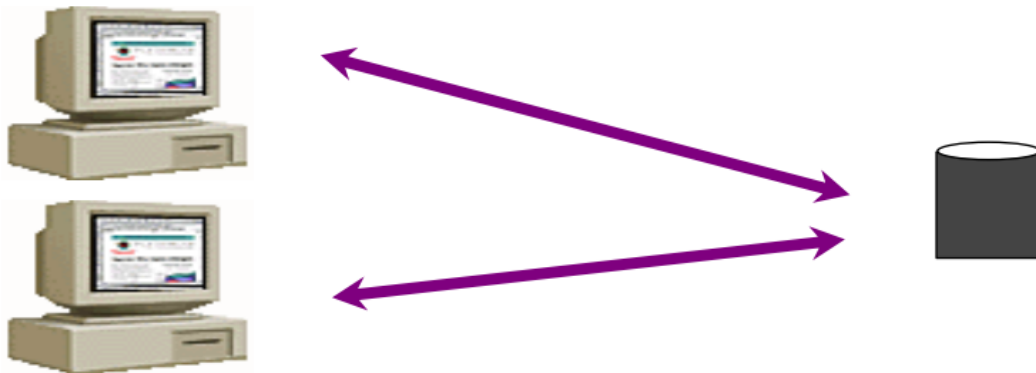
- É aquela vista pelo analista de desenvolvimento e pelo administrador das bases de dados. Existe a preocupação na definição de normas e procedimentos para manipulação dos dados, para garantir a sua segurança e confiabilidade, o desenvolvimento de sistemas e programas aplicativos e a definição no banco de dados de novos arquivos e campos. Na visão conceitual, existem 2 (duas) linguagens de operação que são:

Banco de Dados – Conceitos Gerais

- **Linguagem de definição dos dados (DDL)**
 - Linguagem que define as aplicações, arquivos e campos que irão compor o banco de dados (comandos de criação e atualização da estrutura dos campos dos arquivos).
- **Linguagem de manipulação dos dados (DML)** - Linguagem que define os comandos de manipulação e operação dos dados (comandos de consulta e atualização dos dados dos arquivos).

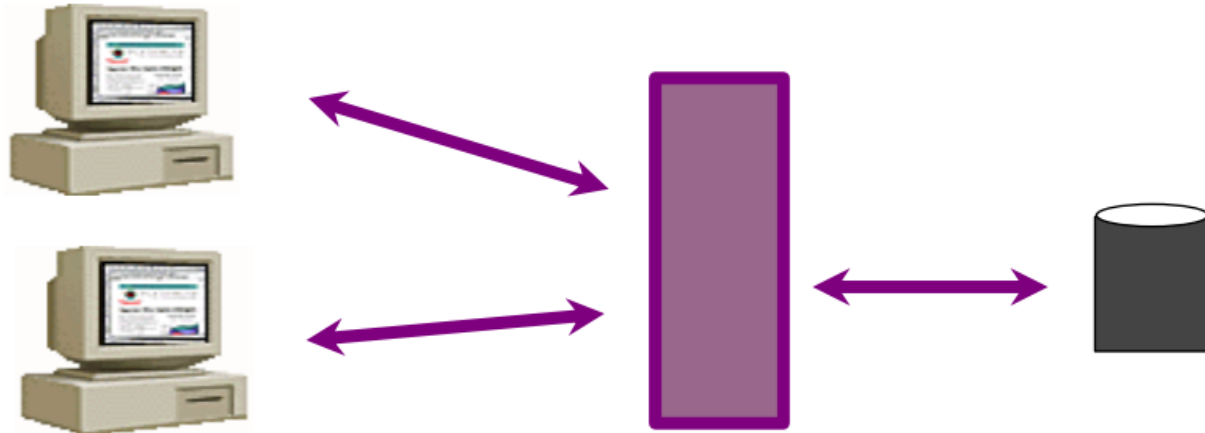
SGBD

- Se vários programas compartilham seus dados, todos devem conhecer e manipular as mesmas estruturas.
- Se algum programa precisar de alguma mudança na estrutura de dados, todos os programas terão que ser alterados, mesmo que a alteração ocorra em dados que ele não utiliza.



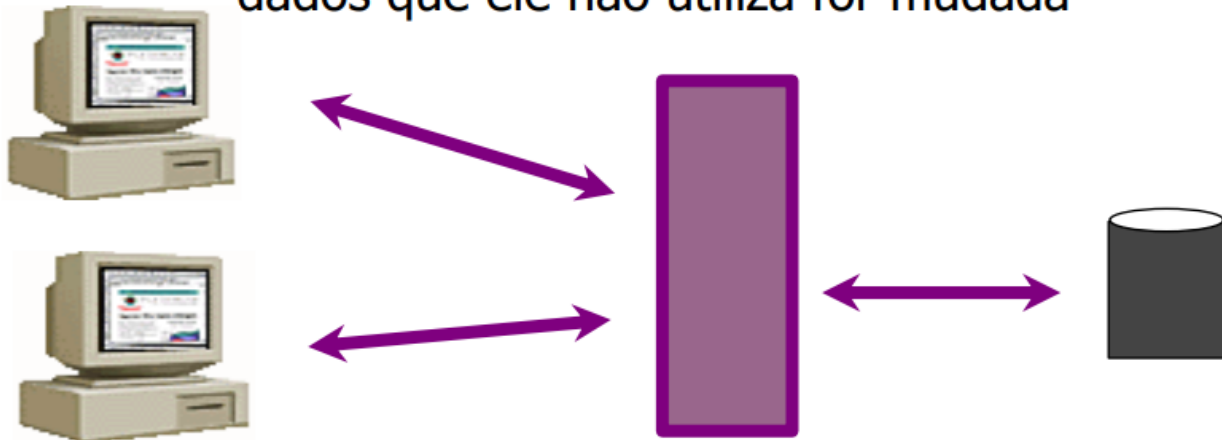
SGBD

- Solução: colocar um sistema entre os dados e os programas, que converta o formato em que os dados estão gravados para o formato específico que cada programa precisa dos dados.



SGBD

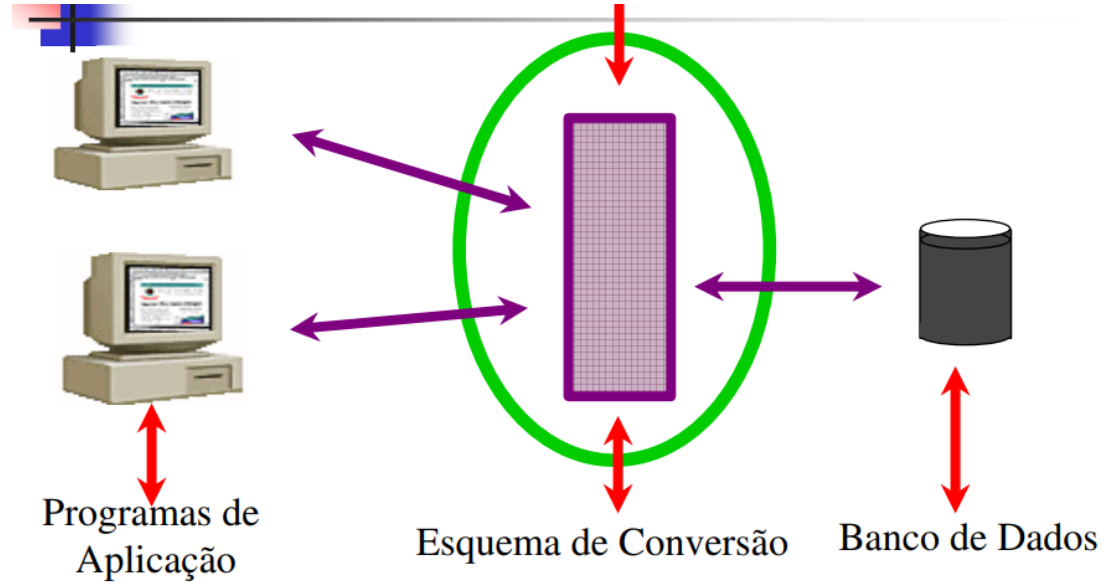
- Cada programa:
 - "Vê" apenas os dados que lhe interessam;
 - Não precisam entrar em detalhes de como seus dados estão fisicamente gravados;
 - Não precisa ser modificado se a estrutura de dados que ele não utiliza for mudada



SGBD

- Consistência de dados:
 - O SGBD avalia os dados recebidos através de regras de integridade, garantindo que sempre estejam corretos.
- Validade:
 - Os dados são válidos quando pertencem ao domínio de valores possíveis naquele caso.

SGBD



SGBD

Nome	Endereco	Telefone
José de Almeida	R. Tulipas 89	0162761112
Jose da Silva	Av. Clóvis Penteado 98	Residencial
Aparecido Navarro	R. Marechal Deodoro	0192445618

Dado inválido !

PostgreSQL

- É um SGBD que incorpora o modelo relacional para o banco de dados e suporta a linguagem SQL como linguagem de consulta padrão.
- Executa em diversas plataformas
 - Unix-like: FreeBSD; Linux; Mac OS X
 - Windows
- É um software livre com código-fonte aberto.

PostgreSQL

- Assim como vários SGBDs comerciais, o PostgreSQL pode ser usado em um ambiente **Cliente/Servidor**
- Existe um processo servidor que é executado em uma única máquina
- Os aplicativos que necessitam acessar os dados devem fazer isso via o processo servidor
- Os clientes não conseguem acessar o dado diretamente, mesmo quando estão rodando no mesmo computador que o servidor

PostgreSQL - Instalação



<https://www.youtube.com/watch?v=FoqXi0wpX4c>

PostgreSQL – Tipos de Dados

- **Primitivos:** Integer, Numeric, String, Boolean
- **Estruturada:** Date/Time, Array, Range, UUID
- **Documento:** JSON/JSONB, XML, Key-value (Hstore)
- **Geometria:** Ponto, Linha, Círculo, Polígono
- **Customizações:** Tipos compostos e personalizados

PostgreSQL – Tipos de Dados

Name	Aliases	Description
bigint	int8	signed eight-byte integer
bigserial	serial8	autoincrementing eight-byte integer
bit [(n)]		fixed-length bit string
bit varying [(n)]	varbit [(n)]	variable-length bit string
boolean	bool	logical Boolean (true/false)
box		rectangular box on a plane
bytea		binary data ("byte array")
character [(n)]	char [(n)]	fixed-length character string
character varying [(n)]	varchar [(n)]	variable-length character string
cidr		IPv4 or IPv6 network address
circle		circle on a plane
date		calendar date (year, month, day)
double precision	float8	double precision floating-point number (8 bytes)
inet		IPv4 or IPv6 host address
integer	int, int4	signed four-byte integer
interval [<i>fields</i>] [(p)]		time span

<https://www.postgresql.org/docs/current/datatype.html>

PostgreSQL – Integridade de Dados

- UNIQUE, NOT NULL
- Chaves Primárias
- Chaves Estrangeiras
- Restrições de Exclusão
- Bloqueios explícitos, bloqueios consultivos

Exercício

- Criar um banco de dados usando os recursos visuais do PostgreSQL. Tabelas:

- Alunos: NroMatricula, cpf, nome endereço, telefone, ano nascimento
- Disciplina: CodDisciplina, nome, código do curso a qual ela pertence
- Matricula: NroMatricula, CodDisciplina, Semestre, Ano, Nota, NroFaltas

Escolha os tipos adequados para cada um dos campos e as chaves primárias

Coloque dados nas tabelas

Exercício

- Teste as seguintes situações:
 - Inserir dados em um campo de uma das tabelas com o tipo diferente daquele definido na criação da tabela
 - Inserir dois registros duplicados em uma das tabelas
 - Inserir uma nova coluna em uma tabela já existente, sendo que a tabela já possui dados
 - Modificar o tipo de dados de uma coluna de uma das tabelas existentes sendo que a tabela já possui dados

Utilizando SQL

A linguagem SQL é o recurso mais conhecido por DBAs e programadores para a execução de comandos em bancos de dados relacionais. É por meio dela que criamos tabelas, colunas, índices, atribuímos permissões a usuários, bem como realizamos consultas a dados. Enfim, é utilizando a SQL que “conversamos” com o banco de dados.

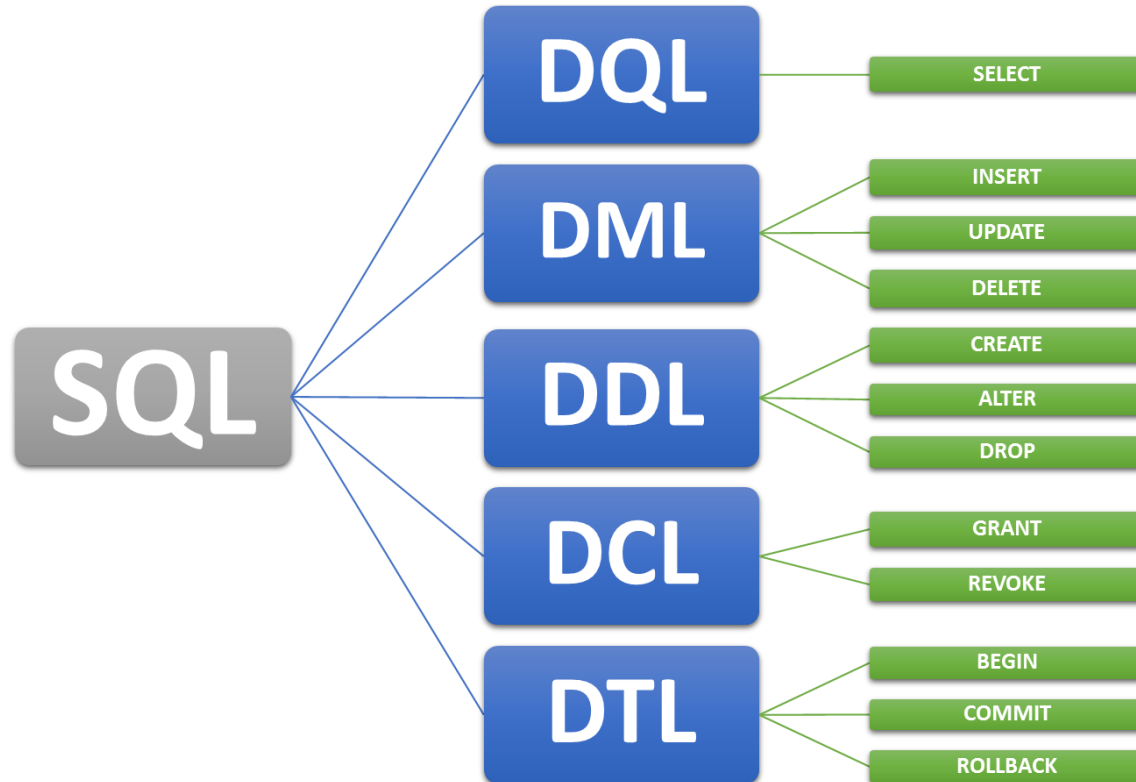
Ela facilita não apenas a execução de tarefas em SGBDs, mas também o diálogo entre profissionais de banco de dados e programadores.

Structured Query Language => Linguagem de Consultas Estruturada

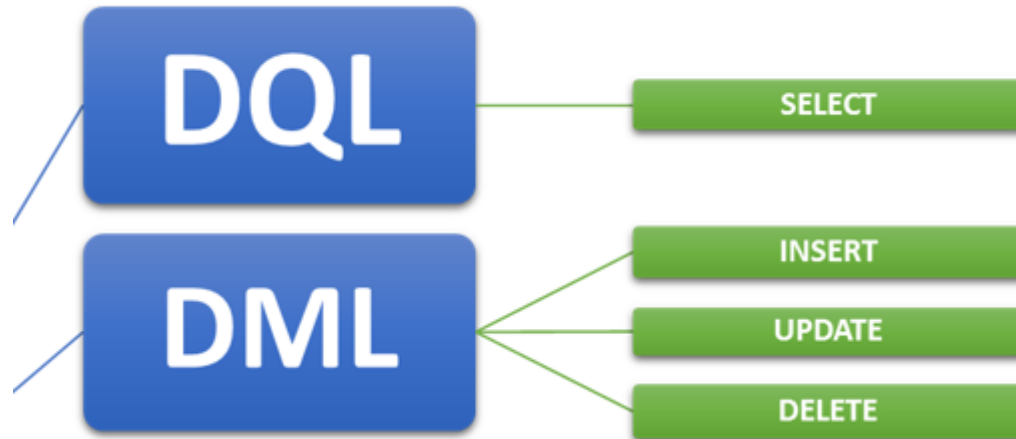
Organização da SQL

- **DQL - Linguagem de Consulta de Dados** - Define o comando utilizado para que possamos consultar (SELECT) os dados armazenados no banco;
- **DML - Linguagem de Manipulação de Dados** - Define os comandos utilizados para manipulação de dados no banco (INSERT, UPDATE e DELETE);
- **DDL - Linguagem de Definição de Dados** - Define os comandos utilizados para criação (CREATE) de tabelas, views, índices, atualização dessas estruturas (ALTER), assim como a remoção (DROP);
- **DCL - Linguagem de Controle de Dados** - Define os comandos utilizados para controlar o acesso aos dados do banco, adicionando (GRANT) e removendo (REVOKE) permissões de acesso;
- **DTL - Linguagem de Transação de Dados** - Define os comandos utilizados para gerenciar as transações executadas no banco de dados, como iniciar (BEGIN) uma transação, confirmá-la (COMMIT) ou desfazê-la (ROLLBACK).

Organização da SQL



Organização da SQL



Comando Select

O comando select é o “comando base” para todos os comandos de pesquisa de informações no banco de dados.

```
1 | select * from pcnf Said;
```

Listagem 1. Selecionando todas as colunas

```
1 | select numnota, dtSaida, vlTotal from pcnf Said;
```

Listagem 2. Selecionando colunas específicas

Comando Select

O comando select é o “comando base” para todos os comandos de pesquisa de informações no banco de dados.

```
1 | select numnota, dtsaida, vltotal, vl desconto, vltotal-vl desconto  
2 | from pcnfsaid;
```

Listagem 3. Como usar operadores aritméticos

Existem quatro maneiras diferentes de atribuir apelidos aos campos selecionados, são elas:

- select codprod as codigo from pcprodut;
- select codprod codigo from pcprodut;
- select codprod as “Codigo do Produto” from pcprodut;
- select codprod “Codigo do Produto” from pcprodut;

Comando Select - Concatenar

O comando **concatenar** consiste em unificar as informações de dois campos distintos da tabela utilizada, exibindo-as em apenas um campo. Também podem ser utilizadas strings que não sejam informações existentes nas tabelas.

```
1 | select especie || serie, numnota from pcnfsaid;
```

Listagem 4. Exemplo de concatenação

```
1 | select "Emitido em:" || "dtemissao:" || " com vencimento  
2 | em:" || dtvenc from pcprest;
```

Listagem 5. Outro exemplo de concatenação

ATENÇÃO. No PostgreSQL utilizamos a função CONCAT().

Exemplo: SELECT

```
CONCAT(especie, serie), numnota  
From pcnfsaid
```

Comando INSERT

O comando para inclusão no banco de dados é o `INSERT`, que possui a seguinte estrutura apresentada na **Listagem 1**.

```
1  INSERT INTO nome_tabela (lista-de-campos)
2  VALUES (lista_dados)
3
4  --OU
5
6  INSERT INTO nome_tabela VALUES (lista_dados)
```

Listagem 1. Sintaxe do comando insert.

- Nome_tabela: nome da tabela no qual será inserido os dados.
- Lista-de-campos: nome das colunas que receberão os valores.
- Lista-dados: valores que serão inseridos na tabela. Estes campos devem estar na mesma ordem descrita em lista-de-campos, todos separados por vírgula. Se for utilizado um **comando SELECT** o mesmo deve retornar a mesma quantidade de colunas com os mesmos tipos de dados especificados em lista-de-campos.

Comando INSERT

```
1 INSERT INTO EMPREGADOS(CODIGO, NOME, SALARIO, SECAO)
2 VALUES(1, "HELBERT CARVALHO", 1.500, 1)
3 INSERT INTO EMPREGADOS VALUES(1,"HELBERT CARVALHO",1500,1)
```

Listagem 2. Exemplos do comando insert.

Listagem 3: Insert com valores provenientes de um select.

```
1 INSERT INTO EMPREGADOS(CODIGO,NOME, SALARIO, SECAO)
2 SELECT CODIGO,NOME, SALARIO, SECAO
3 FROM EMPREGADOS_FILIAL
4 WHERE DEPARTAMENTO = 2
```

Comando INSERT

```
1 | INSERT INTO EMPREGADOS(CODIGO, NOME, SALARIO, SECAO)
2 | VALUES(1, "HELBERT CARVALHO", 1.500, 1)
3 | INSERT INTO EMPREGADOS VALUES(1,"HELBERT CARVALHO",1500,1)
```

Listagem 2. Exemplos do comando insert.

Comando UPDATE

```
1 UPDATE [ tabela ]
2 SET
3   [ coluna_1 ] = [ novo_valor_1 ],
4   [ coluna_2 ] = [ novo_valor_2 ]
5 WHERE
6   [ condicao-de-busca ]
7 ORDER [ ordem ]
8 LIMIT [ limite ];
```

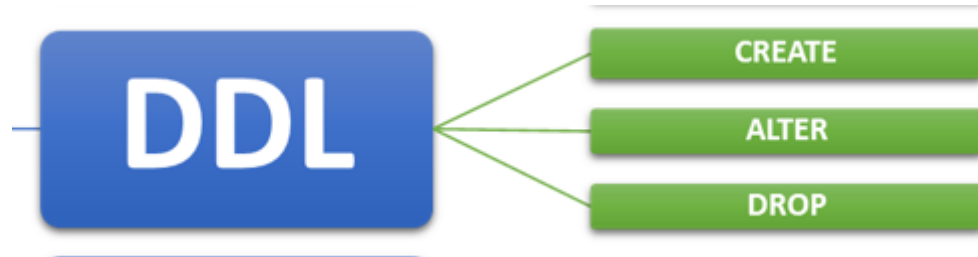
```
1 UPDATE
2   produtos
3 SET
4   descrição = 'Lápis preto (unid)'
5 WHERE
6   id = 2
```

Comando DELETE

```
1  DELETE FROM
2      produtos
3  WHERE
4      id = 3
```

```
1  DELETE FROM
2      [ tabela ]
3  WHERE
4      [ condicao_de_busca ];
```

Organização da SQL



Comando CREATE

```
1 CREATE TABLE Contato (  
2     Id int,  
3     Nome varchar(255),  
4     Telefone varchar(11)  
5 );
```

Acima, temos a criação de uma tabela no SQL. O comando **CREATE TABLE** cria uma tabela, que no nosso exemplo, terá o nome **Contato**. A estrutura interna da tabela é inserida dentro dos parênteses, onde vamos colocar as colunas com as suas propriedades. Nesse caso criamos três colunas, cada uma com um nome e com o seu tipo.

Comando DROP

```
1 | DROP TABLE Contato
```

Elimina a tabela completamente

```
TRUNCATE TABLE Categories;
```

Esvazia as linhas da tabela

Comando ALTER

```
1 ALTER TABLE Contato
2 Add email VARCHAR(255)
```

Adiciona uma coluna

```
ALTER TABLE Customers
DROP COLUMN ContactName;
```

Remove uma coluna

Exercício

Faça os exercícios anteriores, agora utilizando a ferramenta de Query do PostgreSQL e a linguagem SQL

Obrigado!!



<https://docs.ufpr.br/~ademir/p/BancoDados.pdf>

<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>

<https://www.devmedia.com.br/postgresql-tutorial/33025>

<https://www.devmedia.com.br/sql-basico/28877>

<https://www.w3schools.com/sql/>

<https://www.geeksforgeeks.org/postgresql-insert/>