



1

# Banco de Dados

Fabrício Tonetto Londero

Ricardo Frohlich da Silva

# Conceitos básicos

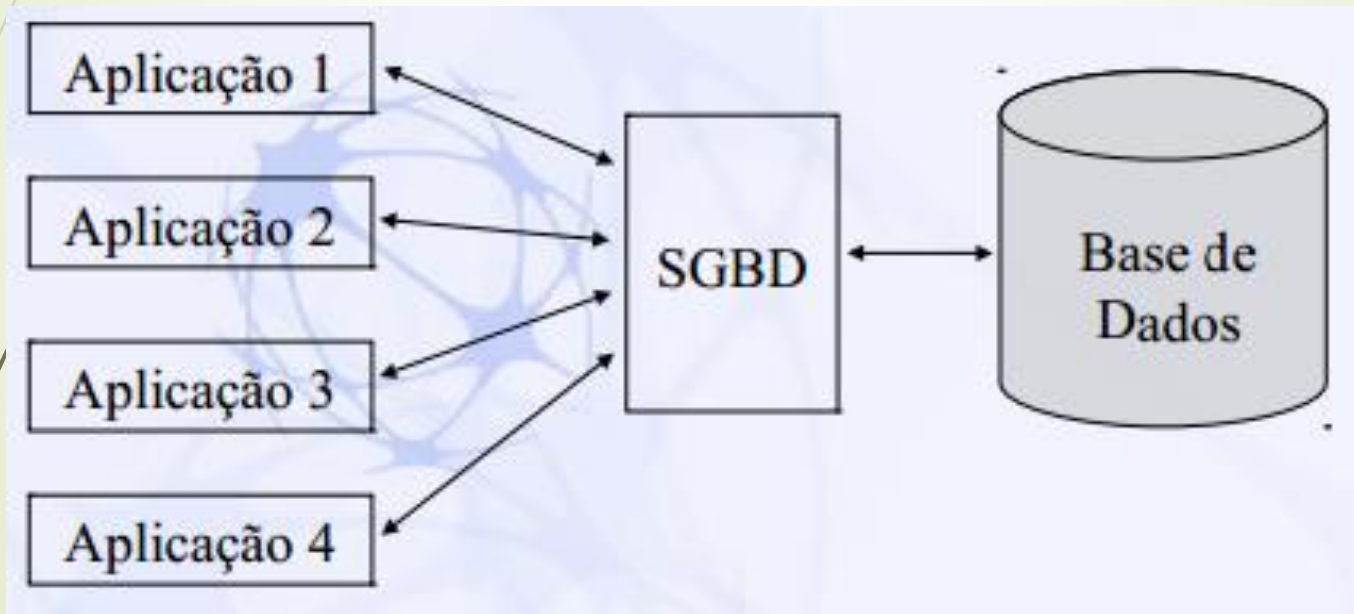
- O que é banco de dados? Seria a mesma coisa de manipulação de arquivos?
- Diferença entre dados e informação?
  - Dado: idade, telefone, nome
  - Informação: os termos acima ligados a algum contexto.
  - Ou seja, informação é o dado processado.
  - “Dados quando organizados, viram informação”
  - “Informação quando organizada, vira conhecimento”

# Conceitos Básicos

- Banco de dados:
  - é o nome dado a um conjunto de arquivos interligados que atendem a um conjunto de sistemas.
  - Tem por objetivo atender a uma comunidade de usuários.
  - São vitais para qualquer empresa, há duas décadas se tornaram a principal peça dos sistemas de informação

# O que é um SGDB?

- Sistema de gerenciamento de banco de dados



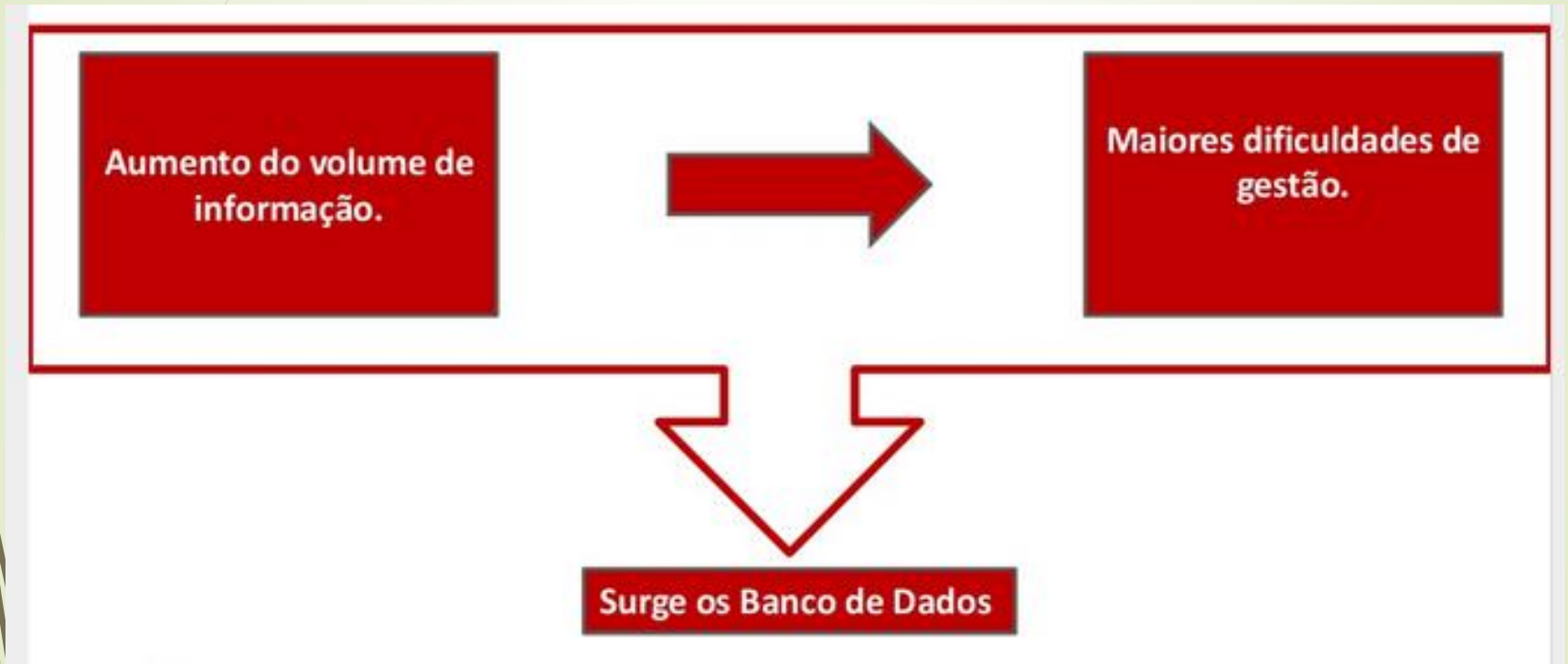
# O que pode-se fazer em um SGDB?

- As aplicações permitem:
  - Inserir
  - Atualizar
  - Recuperar dados
  - Organizar
  - Pesquisar dados de várias maneiras

# Onde se utiliza e o que é?

- A final, o que são banco de dados?
  - Coleção de dados relacionados
- Por que utilizar banco de dados?
  - Gerência de dados
- Onde se utilizam banco de dados?
  - Na agenda do seu celular
    - Pois permite que você edite, inclua ou exclua contatos.
  - Como podemos descrever esse banco de dados?

# Surgimento



# Exemplos de SGDB's

- SQLServer
- PostgreSQL
- MySQL
- Oracle
- FireBird
- MongoDB
- Sybase
- Cassandra
- INUMEROS OUTROS...



# O que é necessário para a criação?

- ▶ Como criar um banco de dados?
  - ▶ Precisamos analisar requisitos
  - ▶ O que será executado?
  - ▶ Que rotinas devem ser executadas para atender as necessidades do cliente?
  - ▶ Quais informações o cliente necessita para ter sucesso no seu negócio?
- ▶ Porém para obter essas informações é necessário definir quais dados serão armazenados no banco de dados.
- ▶ Para que depois possamos devolver ao cliente as informações que ele necessita através desses dados.

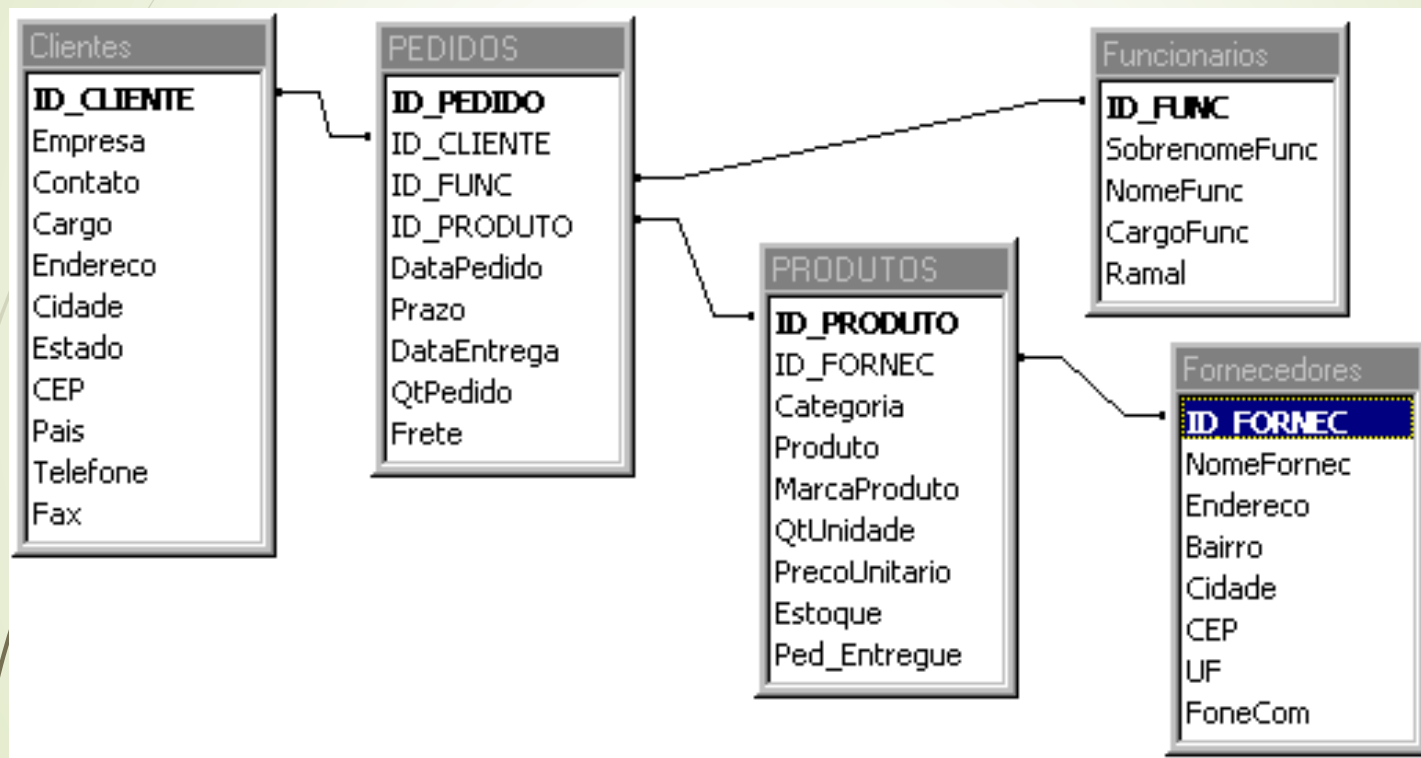
# Para quem?

- Antes de criar um banco de dados você deve responder as seguintes perguntas
  - Qual a finalidade deste banco de dados e quem irá usar?
  - Quais tabelas este banco de dados conterà?
  - De quais consultas e relatórios o usuário deste Banco de dados precisará?
  - Que formulários você precisa criar?

# Componentes básicos de um BD

- Tabelas
- Campos
- Registros
- Chaves (identificador de relacionamento)
- Relacionamentos
  
- Dentre outros

# Exemplo



# Modelo Conceitual, Lógico e Físico, Entidade-Relacionamento

- Modelo de banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados.
- Por exemplo, pode informar que o banco armazena informações sobre produtos e que, para cada produto, são armazenados seu código, preço e descrição.
- O modelo não informa QUAIS produtos estão armazenados, apenas que tipo de informações contém.

# Modelo Conceitual, Lógico e Físico, Entidade-Relacionamento

- Assista o vídeo:
- <https://www.youtube.com/watch?v=ZX7EuRWRdZg>

# Modelos

- Para construir um modelo de dados, usa-se uma linguagem de modelagem de dados.
- Existem linguagens textuais e linguagens gráficas.
- É possível descrever os modelos em diferentes níveis de abstração e com diferentes objetivos.
- Cada descrição recebe o nome de esquema de banco de dados.



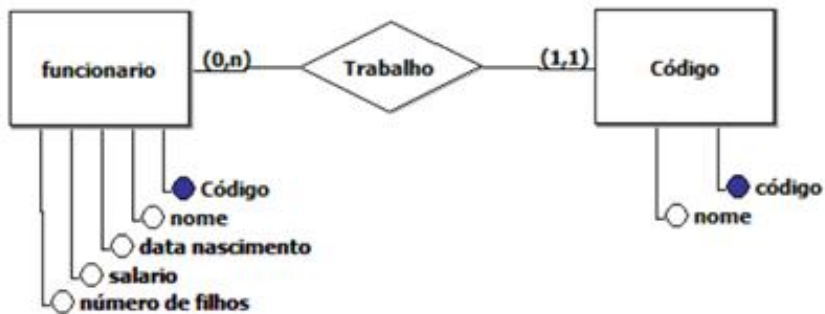


Figura 1- modelo Conceitual

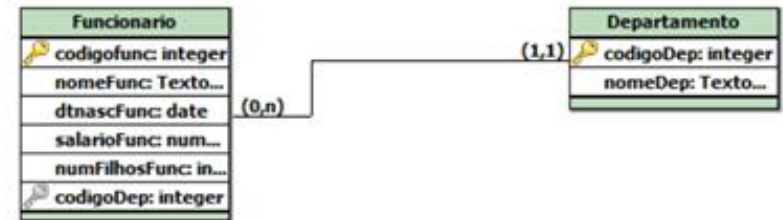


Figura 2- modelo Lógico

```
CREATE TABLE Departamento (
  codigoDep integer PRIMARY KEY,
  nomeDep varchar2(20)
)
```

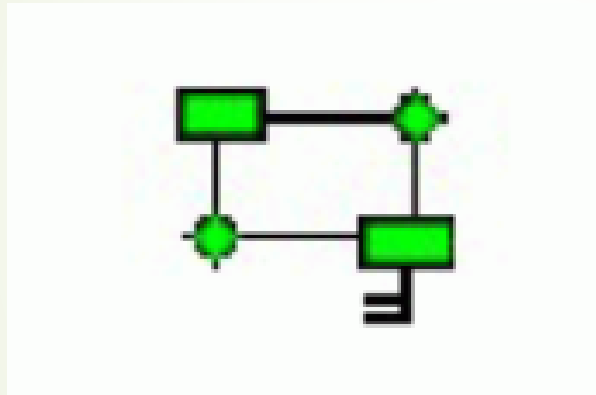
```
CREATE TABLE Funcionario (
  codigofunc integer PRIMARY KEY,
  nomeFunc varchar2(20),
  dtnascFunc date,
  salarioFunc number(15,2),
  numFilhosFunc integer,
  codigoDep integer,
  FOREIGN KEY(codigoDep) REFERENCES Departamento (codigoDep)
)
```

Figura 3 – modelo Físico



# Ferramenta útil

➤ BrModelo



# Linguagem de definição de dados

- DDL (linguagem de definição de dados) é um vocabulário usado para definir estruturas de dados no SGDB que iremos utilizar, o PostgreSQL. Utiliza-se estas instruções para criar, alterar ou remover estruturas de dados do seu BD.

# DDL's

- Instruções CREATE
- Instruções ALTER
- Instruções DROP

# Criando nosso BD

- Vamos a prática 😊 com o modelo físico
- Para isso, abrimos o pgAdmin e então, executamos o seguinte comando:
  - `CREATE DATABASE NOMEDOBANCO`
- Onde você deve escolher o NOMEDOBANCO

# Exemplos CREATE DATABASE

- Create Database ACELERA;
- Create DataBase ControleEstoque;
- Create DataBase Agenda;

# E para excluir?

- Tão simples quanto!
- Usa-se o comando DROP.
- Exemplo:
  - Drop DataBase NOMEDOBANCO
- Tenta ai!

# Tabelas

- Após possuir o banco de dados e estar conectado no mesmo, podemos começar a criação das tabelas.
- As tabelas são um conjunto de dados dispostos em colunas e linhas. A quantidade de colunas é definido pelo DBA ou projetista do banco, conforme a necessidade. A quantidade de linhas é a quantidade de registros cadastrados em uma tabela.
- \*DBA = Administrador do Banco de Dados

# Colunas

- As colunas são definidas como os *campos* da tabela, e são criadas conforme um nome atribuído a elas e um tipo, tais como variáveis em programação.



# Criando Tabelas

- Novamente fazemos uso do comando CREATE.
- Mas antes, precisamos saber quais colunas (campos) nossa tabela vai possuir!
  - Como assim?
    - O que vamos armazenar?
    - Do que se trata a tabela?

# Tabelas

- Exemplo
  - Queremos criar a tabela ALUNO
  - Quais campos esta tabela pode possuir?
    - Quais tipos serão esses campos?

# Tipos de Campos

- Principais tipos que iremos utilizar:
  - **Varchar(xx)**
    - Equivale a **string** em programação
    - XX seria o tamanho em caracteres
  - **Integer**
    - Números inteiros
    - **Int** da programação
  - **Text**
    - Mesmo que **varchar**, porem não precisa estipular tamanho
    - Não muito utilizado por ocupar muita memória

# Tipos de Campos

- Outros tipos:

- **Date**

- Utilizado para datas

- **Real** ou **float**

- Seria o **double** da programação

- Char/datetime/money/decimal/binary/image/timestamp/xml/...

## Voltamos a criação de Tabela

```
create table Contatos  
(  
    nome varchar(20),  
    email varchar(30),  
    datanasc date,  
    celular varchar(20)  
)
```

# Melhorando a tabela

- Precisamos ter em tabelas uma forma de termos um identificador único para registro de tabelas
- Mas como?

# Chave primária

- Uma chave primária é uma coluna ou grupo de colunas usado para identificar uma linha exclusivamente em uma tabela.
- Tecnicamente falando, uma restrição de chave primária é a combinação de uma not-null restrição e uma UNIQUE restrição.
- Deve existir apenas uma chave primária em uma tabela.
- O PostgreSQL produz um índice de árvore B exclusivo na coluna ou um grupo de colunas usado que define a chave primária sempre que uma chave primária é adicionada à tabela.
- Assista o vídeo:
- <https://www.youtube.com/watch?v=p9LUVzgnif8>

# Chave primária

```
CREATE TABLE tbl_autores (  
    ID_Autor integer CONSTRAINT pk_id_autor PRIMARY KEY,  
    Nome_Autor varchar(30) NOT NULL,  
    Sobrenome_Autor varchar(40) NOT NULL,  
    Data_Nasc date  
);
```



# Exercício 1

- Crie uma tabela para armazenar seus livros.

## Exercício 2

- Crie uma tabela para armazenar Veículos.

# Criei errado, e agora?

- Podemos simplesmente *dropa-la* e fazer de novo...
  - `DROP TABLE NOMETABELA;`
- Ou podemos altera-la!

# Alter Column

alter table contatos  
add **nomecoluna** tipodedado

**ou**

drop **column**  
alter **column** tipodedado

# Alter Column

- Adicionar coluna:
  - ALTER TABLE NOMETABELA  
ADD NOMECOLUNA TIPODADO
- Excluir coluna:
  - ALTER TABLE NOMETABELA  
DROP COLUMN NOMECOLUNA

# Chave estrangeira

```
CREATE TABLE tbl_livros (  
  ID_Livro SERIAL CONSTRAINT pk_id_livro PRIMARY KEY,  
  Nome_Livro varchar(50) NOT NULL,  
  Autor integer NOT NULL,  
  Editora integer NOT NULL,  
  Data_Pub date,  
  Genero integer NOT NULL,  
  Preco_Livro money,  
  FOREIGN KEY (Autor) REFERENCES tbl_autores (ID_Autor));
```

## Exercício 3

- Na tabela do exercício 1, acrescente o campo ISBN como inteiro.

## Exercício 4

- Na tabela do exercício 2, exclua o campo cor.



## Exercício 5

- Na tabela do exercício 3, altere o tipo do ISBN de int para varchar.



# DDL

- *Data Definition Language* ( DDL) são usadas para definir a estrutura de banco de dados ou esquema. Alguns exemplos:
  - CREATE- para criar objetos no banco de dados
  - ALTER - altera a estrutura da base de dados
  - DROP - exclusão de objetos




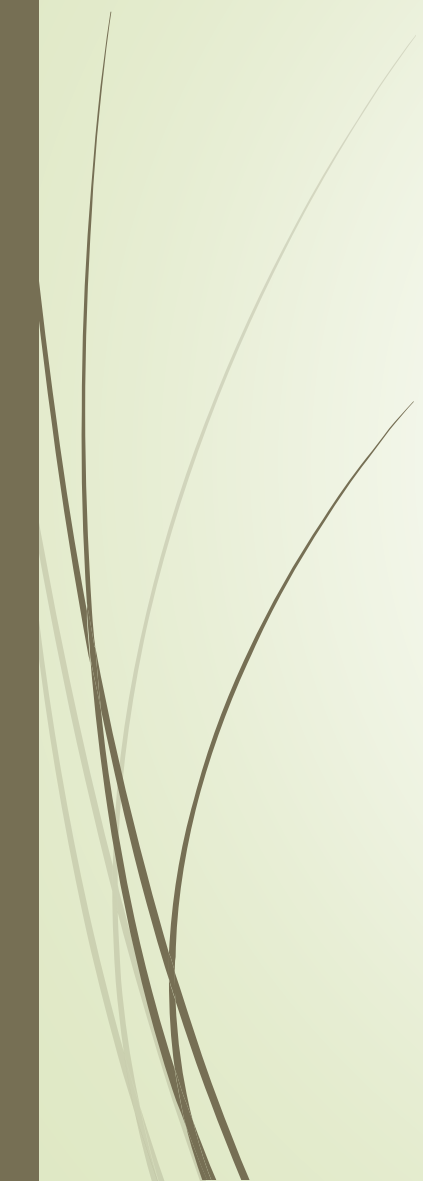
# DML

- *Data Manipulation Language* ( DML) são utilizados para o gerenciamento de dados dentro de objetos do banco. Alguns exemplos:
- SELECT- recuperar dados do banco de dados
- INSERT – inserir dados em uma tabela
- UPDATE – atualiza os dados existentes em uma tabela
- DELETE – exclui registros de uma tabela



# Inserção de Dados

- Estando com nossas tabelas criadas, podemos começar a incluir dados em nossas tabelas.
- Para inserção, utilizamos o comando INSERT

- 
- 
- insert into nometabela values (v1,v2,v3,v4, ...)
  - Insert into livros values ('Sherlock Holmes', 'Conan Doyle', 240)
  - insert into tbl\_autores values (1,'José','da Silva','2022-11-10');
  - *insert into tbl\_autores values (2,'Ricardo','da Silva','2000-11-3');*
  - NECESSARIO RESPEITAR OS TIPOS DE DADOS



# Exercício 1

- Insira os dados na tabela CONTATOS. Não tem a tabela? CRIE!

Pedro, 05597653487, [predro@gmail.com](mailto:predro@gmail.com)

Alan, 05184101245, [alan@gmail.com](mailto:alan@gmail.com)

Joana, 06799435687, [joana@gmail.com](mailto:joana@gmail.com)

Carlos, 05599105314, [carlos@gmail.com](mailto:carlos@gmail.com)

# Exercício 1

- E se um dos contatos não tem email ou outro dado?
- *insert into nometabela(c1,c2,c3,c4,...) values (v1,v2,v3,v4, ...)*
- *insert into tbl\_autores (id\_autor, nome\_autor,sobrenome\_autor)*
- *values (3,'Ana', 'Soares');*



# Exercício 2

- Insira os dados na tabela CONTATOS. Não tem a tabela? CRIE!


Pedro, 05597653487

Alan, 05184101245, [alan@gmail.com](mailto:alan@gmail.com)

Joana, [joana@gmail.com](mailto:joana@gmail.com)

Carlos, 05599105314, [carlos@gmail.com](mailto:carlos@gmail.com)





# Como verificar se a inserção esta correta?

- Para visualizarmos o que foi inseridos, devemos efetuar consultas.
- Para consultar, utilizamos o comando SELECT



# SELECT

`select * from MinhaTabela`

- \* = significa TUDO, ou no caso, TODOS os campos
- `select * from tbl_autores`