

Aula 3

Banco de Dados

Prof. Lucas Rafael Filipak

Conversa Inicial

Banco de dados

- Linguagem SQL
- Criando o banco de dados
- Criando tabelas
- Modificando tabelas

Linguagem SQL

Linguagem SQL

- 1974-1979 – IBM
- Banco de dados relacional
- Padrão de consulta
- Padronizada final da década de 1980

- O que pode ser feito, segundo Damas (2005):
 - Criar, alterar e remover elementos de um banco de dados
 - Inserir, alterar e apagar dados
 - Consultar os dados
 - Controlar o acesso do usuário

- Segundo Puga (2013), a SQL-92 é dividida
três categorias:
 1. DDL – *Data Definition Language*
 2. DML – *Data Manipulation Language*
 - ✓ DCL – *Data Control Language*
 3. TCL – *Transaction Control Language*
- DCL é uma subcategoria da DML

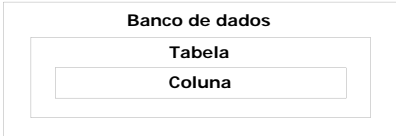
- **Padronização do SQL**
 - 1986 – ANSI
 - 1987 – ISSO
 - Os padrões são “dialetos” que os principais fabricantes de SGBDs implementam em seus bancos de dados

- Surgem quando o fabricante precisa implementar algum recurso no SGBD, mas esse recurso ainda não foi validado/criado no padrão ANSI
- Os padrões criados pelas empresas têm suas particularidades, mas seguem a base do SQL

Operações DDL

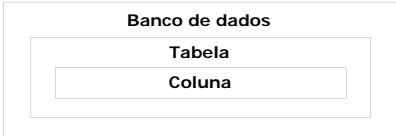
Data Definition Language (DDL)

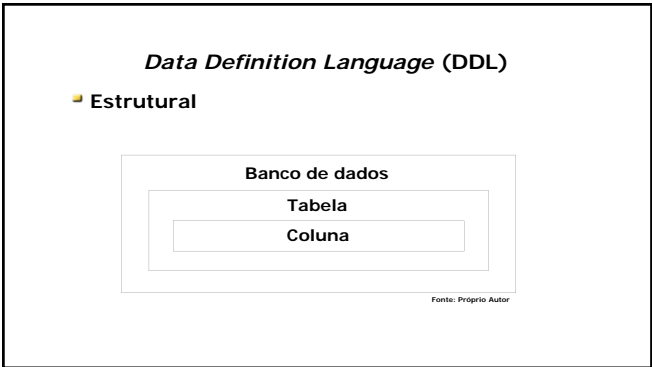
- Estrutural



```
graph TD; BD[Banco de dados] --> T[Tabela]; T --> C[Coluna]
```

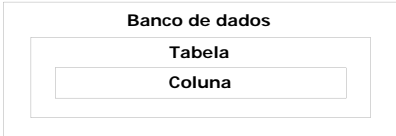
Fonte: Próprio Autor

- ## *Data Definition Language (DDL)*
- Estrutural
- 
- ```
graph TD; BD[Banco de dados] --> T[Tabela]; T --> C[Coluna]
```
- Fonte: Próprio Autor



## *Data Definition Language (DDL)*

- Estrutural



```
graph TD; BD[Banco de dados] --> T[Tabela]; T --> C[Coluna]
```

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

**Criando o banco de dados**

Nos nomes de bancos e tabelas, não são permitidos espaços. Use o underline.

`CREATE DATABASE sistema;`

- `CREATE DATABASE` é o comando
- O nome do banco de dados é `sistema`
- Os comandos devem terminar com ponto-e-vírgula

`> CREATE DATABASE sistema;`  
Query OK, 1 row affected (0.01 sec)

Query OK é o retorno que o SGBD traz quando a instrução é executada com sucesso

Fonte: Próprio Autor

## Visualizando bancos de dados do SGBD

```
> SHOW DATABASES;
+-----+
| Databases |
+-----+
| base_alunos |
| base_prefeitura |
| conjunto |
| sistema |
| sistema-bkp |
+-----+
5 rows in set (0.00 sec)
```

Fonte: Próprio Autor

## Escolhendo o banco de dados

USE sistema;

A partir da instrução executada, todos os comandos serão executados na base de dados selecionada (sistema).

```
> USE sistema;
Database changed
```

Database changed é a confirmação de que a base foi alterada.

Fonte: Próprio Autor

## Criando tabelas

- Tabelas no banco de dados
- Uma ou mais tabelas
- Nome único
- Dados são armazenados em linhas e colunas

## Criando tabelas

O parêntese inicia a lista de colunas a serem criadas

Comando para criar a tabela

Nomes da tabela sem espaços, com underline

```
CREATE TABLE cad_aluno
(
 cod_aluno varchar(5),
 nome varchar(30)
);
```

Nome da primeira coluna a ser criada

Nome da segunda coluna a ser criada

A vírgula separa as colunas que vão ser criadas

Tipo e tamanho máximo da coluna. No exemplo, a coluna será do tipo varchar (texto) com tamanho máximo de 30 caracteres

Fechamento da lista de colunas que vão ser criadas. O ponto-e-vírgula determina o fim do comando

Fonte: Próprio Autor

## Visualizando tabelas

```
> SHOW TABLES;
+-----+
| Tables_in_Sistema |
+-----+
| cad_aluno |
+-----+
1 row in set (0.00 sec)
```

Fonte: Próprio Autor

## Estrutura da tabela

> DESCRIBE cad\_aluno;

| Field     | Types       | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| cod_aluno | Varchar(5)  | YES  |     | NULL    |       |
| nome      | Varchar(30) | YES  |     | NULL    |       |

2 rows in set (0.01 sec)

Fonte: Próprio Autor

## Campos nulos

O comando NOT NULL informa que o Preenchimento da coluna é obrigatório

```
CREATE TABLE cad_aluno
(
 cod_aluno varchar(5) NOT NULL ,
 nome varchar(30) NOT NULL
);
```

Fonte: Próprio Autor

## Chave primária

A coluna cod\_material foi escolhida para ser a chave primária, ela não pode se repetir

```
CREATE TABLE materia
(
 cod_material int NOT NULL,
 data_inicio date NOT NULL,
 nome varchar(40) NOT NULL,
 PRIMARY KEY (cod_material)
);
```

Criando a chave primária. Na última instrução Entre parênteses o nome não vai vírgula da coluna escolhida

Fonte: Próprio Autor

## AUTO\_INCREMENT

Definindo a coluna cod\_turno como auto\_incrmente

```
CREATE TABLE turno
(
 Cod_turno int NOT NULL AUTO_INCREMENT,
 nome_turno varchar (20) NOT NULL
 PRIMARY KEY (cod_cor)
);
```

Fonte: Próprio Autor

## Modificando tabelas

## Modificando tabelas

- O Comando ALTER TABLE permite:
  - Renomear uma tabela
  - Alterar tipo e tamanho de uma coluna
  - Adicionar ou remover colunas na tabela

#### Renomeando tabelas

Antigo nome da tabela    Novo nome da tabela

```
ALTER TABLE cad_aluno RENAME alunos;
```

Fonte: Próprio Autor

#### Alterando o tamanho de uma coluna

Comando para modificar    Novo tamanho

```
ALTER TABLE alunos MODIFY nome varchar(70);
```

Fonte: Próprio Autor

#### Adicionando colunas

Adicionar uma nova coluna    Adicionar uma nova coluna

```
ALTER TABLE aluno ADD COLUMN email varchar(30);
```

Fonte: Próprio Autor

#### Excluindo colunas

Apagar uma coluna    Nome da coluna a ser excluída

```
ALTER TABLE aluno DROP COLUMN email;
```

Fonte: Próprio Autor

#### Modificando o AUTO\_INCREMENT

```
ALTER TABLE alunos AUTO_INCREMENT = 15;
```

Fonte: Próprio Autor

#### Excluindo tabelas

DROP TABLE é o comando    O nome da tabela que vai ser apagada

```
DROP TABLE cad_alunos;
```

Fonte: Próprio Autor

#### Excluindo DATABASES

DROP DATABASE é o comando    O nome da database que vai ser apagada

```
DROP DATABASE sistema;
```

Fonte: Próprio Autor

Finalizando

- SQL (DDL, DML, DTL, TCL)
- Tabelas
- CREATE DATABASE
- CREATE TABLE
- ALTER TABLE

## Referências

- DAMAS, L. SQL – Structured Query Language. 6. ed. atualizada e aumentada. Rio de Janeiro: FCA, 2005
- PUGA, S. Banco de dados: implementação em SQL, PL/SQL, Oracle 11g. São Paulo: Pearson Education do Brasil, 2013.