

# Banco de Dados I

Aula08

## SQL - DDL

---

**Prof. MSc. Adalto Selau Sparremerger**

 assparremerger@senacrs.com.br

  @adaltoss  
 

 /assparremerger



Fecomércio RS



Senac

# SQL – Structured Query Language

Linguagem de Consulta Estruturada

- ▷ É uma linguagem padrão para armazenar, manipular e recuperar dados em bancos de dados;
- ▷ **SEQUEL** - Structured **English QU**ery Language
- ▷ Projetada e implementada no Centro de Pesquisa da IBM na década de 1970 como interface de um banco de dados relacional experimental chamado System R.

```
CREATE DATABASE loja;
```

```
CREATE TABLE clientes (  
    id INT PRIMARY KEY NOT NULL AUTO_INCREMENT ,  
    nome VARCHAR(50) NOT NULL,  
    altura DOUBLE  
);
```

```
USE loja;
```

```
SELECT * FROM clientes;
```

```
SELECT nome, telefone FROM clientes;
```

```
UPDATE clientes SET  
    nome = 'Júlia',  
    altura = 1.7  
WHERE id = 1;
```

```
DELETE FROM clientes  
WHERE id = 1;
```

```
INSERT INTO clientes ( nome, altura ) VALUES ( 'Maria', 1.65 );
```



Fecomércio RS



Senac

# O que posso fazer com SQL

- ▷ Executar consultas contra um banco de dados
- ▷ Recuperar dados de um banco de dados
- ▷ Inserir registros em um banco de dados
- ▷ Atualizar registros em um banco de dados
- ▷ Excluir registros de um banco de dados
- ▷ Criar novos bancos de dados
- ▷ Criar novas tabelas numa base de dados
- ▷ Criar procedimentos armazenados em um banco de dados
- ▷ Criar visualizações em um banco de dados
- ▷ Definir permissões em tabelas, procedimentos e visualizações

# SQL é mais do que uma linguagem de consulta:

- Definição de dados
  - Permite a definição da estrutura do banco de dados
- Recuperação de dados
  - Possibilita a recuperação dos dados do banco
- Manipulação de dados
  - Permite alterações e exclusões de dados já existentes, bem como a adição de novos dados
- Controle de acesso
  - Fornece meios para determinar o controle de acesso de usuários aos dados do banco
- Compartilhamento de dados
  - Coordena o acesso concorrente de usuários, assegurando que um não interfira no trabalho do outro
- Integridade de dados
  - Permite definir restrições de integridade aos dados do banco

# SQL - DDL

▷ SQL como linguagem de definição de dados – DDL

- Definição de tabelas – CREATE
- Alteração de tabelas – ALTER
- Remoção de tabelas - DROP

# SQL - DDL

- ▷ Instruções para definição do esquema da base de dados:
- ▷ **CREATE TABLE**
  - Cria uma nova tabela na base de dados, especificando nome, atributos e restrições
- ▷ **ALTER TABLE**
  - Altera definições de uma tabela
- ▷ **DROP TABLE**
  - Remove uma tabela, quando suas definições não são mais necessárias

# Construindo um Banco de Dados

▷ **CREATE DATABASE** nome\_do\_banco;

▷ Para entrar neste banco para seguir construindo as tabelas:

- **USE** nome\_do\_banco;





# Destruindo um Banco de Dados

▷ **DROP DATABASE** nome\_do\_banco;

# Construção de tabela



## CREATE TABLE

- Colunas são especificadas primeiro, sob a forma:  
<nomeCol> <domínio> <restrição>
- Depois Chaves, integridade referencial e restrições de integridade

```
CREATE TABLE <nome_da_tabela>
(
    <nome_da_col1> <tipo_da_col1> NOT NULL ,
    <nome_da_col2> <tipo_da_col2> NOT NULL ,
    ...
    PRIMARY KEY <lista_de_nomes_de_col> ,
    FOREIGN KEY <nomes_de_col>
    REFERENCES <nome_tab_ref>
    (<nome_da_col_ref>)
);
```

# Restrições e Valores Default

- Restrições:

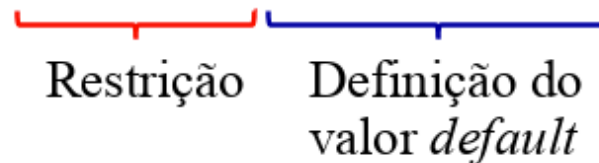
- **NOT NULL**

- Restrição aplicadas a colunas cujos valores não podem ser nulos

- Valores *Default*

- Adiciona-se a cláusula **DEFAULT** <valor> logo após a restrição:

```
CREATE TABLE Empregado
( ...
Esexo CHAR(1) NOT NULL DEFAULT "F",
...
);
```

  
Restrição      Definição do  
valor *default*

# Construção de Chave Estrangeira

```
CREATE TABLE Empregado  
(  
    ...,  
    CONSTRAINT TrabalhaEm  
    FOREIGN KEY (Cod_Dept) REFERENCES  
    Departamento (Dcod),  
    ...  
);
```

→ Opcional.  
Definido no SQL2.  
Útil para, por exemplo,  
excluir uma chave estrangeira

# Construção de Chave Alternativa

```
CREATE TABLE Empregado  
(  
    . . . ,  
    UNIQUE (CPF) );
```

# Remoção de Tabelas

- **DROP TABLE**

- Elimina completamente a tabela (vazia ou não)

```
DROP TABLE <nome_da_tabela>;
```

- Ex.:

```
DROP TABLE Empregado;
```

- Não há como recuperar a tabela removida

# Remoção de Tabelas

## ▫ CASCADE

- Se existirem outros objetos que dependam da tabela excluída serão excluídos também

## ▫ RESTRICT

- Exclui a tabela somente se não existirem objetos do banco de dados que dependam da tabela

**DROP TABLE** Empregado **CASCADE ;**



Fecomércio RS



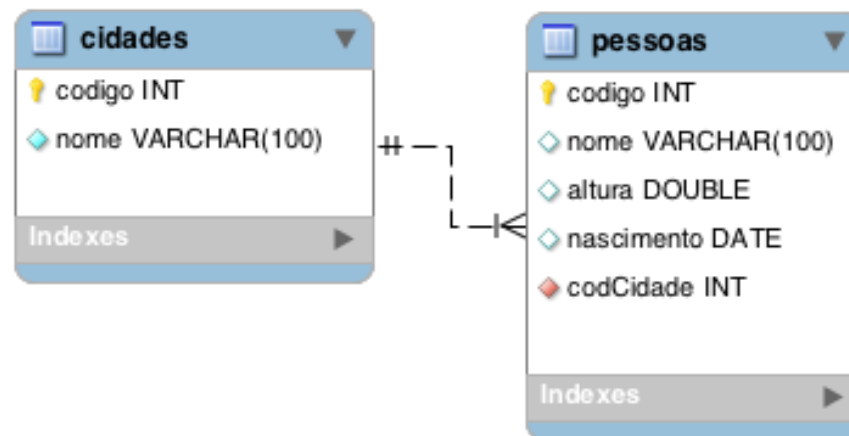
# Banco de Exemplo

Nome\_da\_Tabela (campo1

idades (codigo, nome)

essoas (codigo, nome, altura,  
nascimento, codCidade)

codCidade referencia cidades





```
CREATE TABLE cidades  
(  
    codigo INT NOT NULL  
    AUTO_INCREMENT ,  
    nome VARCHAR(100) ,  
    PRIMARY KEY ( codigo )  
);
```

```
CREATE TABLE pessoas
(
    codigo INT NOT NULL PRIMARY KEY AUTO_INCREMENT ,
    nome VARCHAR(100) NOT NULL ,
    altura DOUBLE ,
    nascimento DATE DEFAULT `1970-12-25` ,
    codCidade INT ,
    FOREIGN KEY ( codCidade ) REFERENCES cidades( codigo )
);
```

# Alterando a estrutura de uma tabela

```
ALTER TABLE pessoas
```

```
ADD COLUMN telefone VARCHAR(50) NOT NULL ;
```

```
ALTER TABLE pessoas
```

```
CHANGE telefone telefone VARCHAR(100) NOT NULL ;
```

```
ALTER TABLE pessoas
```

```
CHANGE telefone fone VARCHAR(100) NOT NULL ;
```

# Excluindo a estrutura de uma tabela

```
DROP TABLE pessoas;
```

- Crie as seguintes tabelas em seu bd, definindo as condições descritas abaixo.

Fornecedor (Fcod, Fnome, Status, Cidade)

Peça (Pcod, Pnome, Cor, Peso, Cidade)

Projeto (PRcod, Icod, PRnome, Cidade)

Icod referencia Instituição(Icod)

Fornecimento (Fcod, Pcod, PRcod, Quantidade)

Fcod referencia Fornecedor(Fcod)

Pcod referencia Peça (Pcod)

PRcod referencia Projeto (Prcod)

Instituição (Icod, nome)

- Nomes para restrições de chaves primária, estrangeira e alternativa
- Valor *default* para o atributo Status, de Fornecedor
- Todos os atributos de Peça não podem ser nulos
- Qual tabela deve ser criada por último? Porque?

## • BIBLIOGRAFIA

- HEUSER, Carlos Alberto. Projeto de Bancos de Dados: Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS. Bookman Editora, 2009.
- RAMAKRISHNAN, R.; GEHRKE, J. Sistema de Gerenciamento de Banco de dados. Terceira Edição. 2008. Mc Graw Hill.
- ORACLE. MySQL 5.7 Reference Manual. Disponível em: <https://dev.mysql.com/doc/refman/5.7/en/>
- SQL Tutorial. MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems. Disponível em: <https://www.w3schools.com/sql/>
- Documentação do SQL Server. Disponível em: <https://docs.microsoft.com/pt-br/sql/sql-server/sql-server-technical-documentation?view=sql-server-2017>