

# Ambiente de Dados

## Introdução ao SQL

### Aula 5

# SQL

- Surgimento da linguagem SEQUEL por pesquisadores da IBM.
- SEQUEL foi a primeira linguagem criada para consulta a banco de dados relacionais.
- Crescimento do bancos de dados sem uma linguagem padrão de consulta definida.
- Surgimento de uma linguagem de consulta padrão chamada **SQL** de um trabalho conjunto entre ISO e ANSI. (SQL-86).
- Aperfeiçoamentos da linguagem através do SQL-92, SQL99, SQL:2003 e SQL:2005.

## Classificação da Linguagem de Banco de Dados

- **Linguagem de Definição de Dados (DDL)** - Linguagem utilizada para **definir a estrutura de armazenamento dos dados**, também chamada de dicionário de dados ou metadados.
- **Linguagem de Controle de Dados (DCL)** – Linguagem utilizada para controle de sessão, permissão, além do controle do próprio sistema de banco de dados.
- **Linguagem de Manipulação de Dados (DML)** - Linguagem utilizada para **leitura e armazenamento** de dados.
- **Linguagem de Controle de Transação (TCL)** – Linguagem de controle de Transações

# Comandos DDL

- **CREATE**
- **ALTER**
- **DROP**
- **RENAME**

# Criando o Banco de Dados

CREATE DATABASE ***NOMEDATABASE***

# Excluindo o Banco de Dados

DROP DATABASE ***NOMEDATABASE***

# Criando Tabela *SQL Padrão*

```
CREATE TABLE NOME_TABELA (  
COL1 TIPO_COL1 [NOT NULL],  
COL2 TIPO_COL2 [NOT NULL],  
  
...,  
COLN TIPO_COLN [NOT NULL]  
)
```

# Criando Tabela

```
CREATE TABLE EDITORA (  
  CODIGO NUMBER NOT NULL,  
  NOME   VARCHAR2(80) NOT NULL  
);
```



# Criando Tabela

```
CREATE TABLE LIVRO (  
  CODIGO          NUMBER NOT NULL,  
  TITULO          VARCHAR2(80) NOT NULL,  
  PRECO           NUMBER,  
  LANCAMENTO     DATE,  
  COD_EDITORA    NUMBER(2)  
)
```

# Restrições de Tabela *SQL Padrão*

- Chave Primária
- Chave Estrangeira
- Restrição de Domínio

# Restrição de Integridade *SQL Padrão*

CONSTRAINT ***NOME\_RESTRIÇÃO*** TIPO\_RESTRIÇÃO

# Criando Tabela com Restrição de Chave Primária

```
CREATE TABLE EDITORA (  
CODIGO NUMBER CONSTRAINT PK_EDITORA PRIMARY KEY,  
NOME VARCHAR2(80) NOT NULL  
)
```

```
CREATE TABLE ASSUNTO (  
CODIGO NUMBER CONSTRAINT PK_ASSUNTO PRIMARY KEY,  
NOME VARCHAR2(100) NOT NULL  
)
```

## MYSQL

```
CREATE TABLE Funcionario (  
idFuncionario INT NOT NULL,  
Salario INT NULL,  
PRIMARY KEY (idFuncionario));
```

# Criando Tabela com Restrição de *Chave Estrangeira*

```
CREATE TABLE LIVRO (  
  CODIGO          NUMBER NOT NULL,  
  TITULO          VARCHAR2(80) NOT NULL,  
  PRECO           NUMBER,  
  LANCAMENTO     DATE,  
  COD_ASSUNTO    NUMBER  
                CONSTRAINT FK_LIVRO_ASSUNTO  
                REFERENCES ASSUNTO (CODIGO),  
  COD_EDITORA    NUMBER  
                CONSTRAINT FK_LIVRO_EDITORA  
                REFERENCES EDITORA (CODIGO )  
  )
```

# Criando Tabela com Restrição de *Domínio*

```
CREATE TABLE LIVRO (  
  CODIGO          NUMBER NOT NULL,  
  TITULO          VARCHAR2(80)  
    CONSTRAINT CHK_TITULO  
      CHECK (TITULO LIKE 'BANC%'),  
  PRECO NUMBER    CONSTRAINT CHK_PRECO  
    CHECK (PRECO > 1500),  
  LANCAMENTO     DATE,  
  COD_ASSUNTO    CONSTRAINT CHK_COD_ASSUNTO  
    NUMBER CHECK (COD_ASSUNTO IN (1,2,3))  
  )
```

# Criando Atributos Default em *SQL Padrão*

ATRIBUTO TIPO\_COLUNA **DEFAULT** VALOR\_PADRAO

# Criando Tabela com Atributos *Default*

```
CREATE TABLE LIVRO (  
  CODIGO          NUMBER NOT NULL,  
  TITULO          VARCHAR2(80) ,  
  PRECO           NUMBER DEFAULT 150,  
  LANCAMENTO      DATE       DEFAULT SYSDATE (Oracle) |  
                        Getdate() (SQL SERVER),  
  COD_ASSUNTO     NUMBER  
)
```



# Tipos de Ações para *Chave Estrangeira* em *SQL Padrão*

## 1. Impedindo a exclusão:

```
COD_ASSUNTO CHAR(1)
CONSTRAINT FK_LIVRO_ASSUNTO
REFERENCES ASSUNTO (CODIGO)
ON DELETE RESTRICT
```

## 2. Transformando o valor da tabela filha em nulo:

```
COD_ASSUNTO CHAR(1)
CONSTRAINT FK_LIVRO_ASSUNTO
REFERENCES ASSUNTO (CODIGO)
ON DELETE SET NULL
```

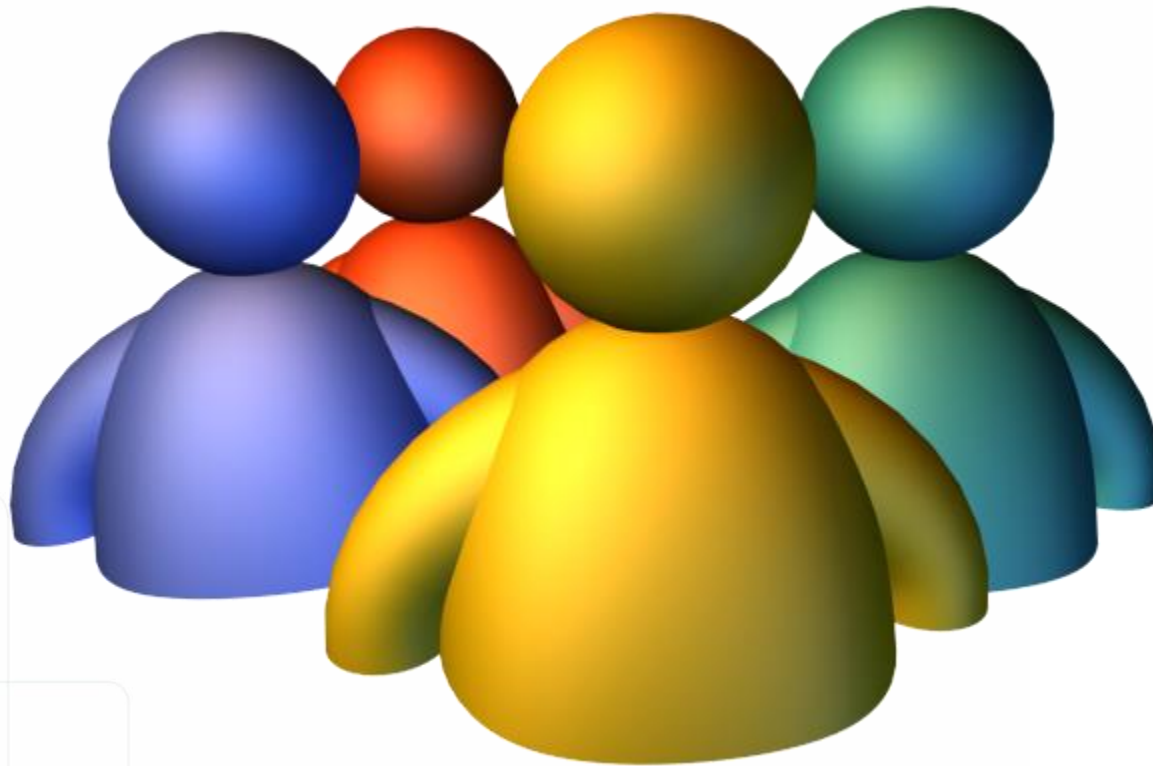
# Tipos de Ações para *Chave Estrangeira* em *SQL Padrão*

3. Excluindo em cascata: Quando um registro da tabela Pai for excluído, então todos os registros da tabela filha referenciada serão excluídos.

```
COD_ASSUNTO CHAR(1)  
CONSTRAINT FK_LIVRO_ASSUNTO  
REFERENCES ASSUNTO
```

**ON DELETE CASCADE**

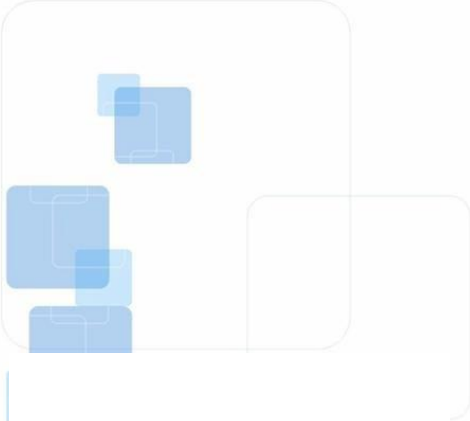
# Dúvidas?





# EXERCÍCIOS DE FIXAÇÃO

Realizar a implementação do Banco de Dados  
(Script DDL com a sintaxe de criação das tabelas)  
dos casos a seguir:



# Exercício 1

- Script DDL para um Sistema Acadêmico respeitando os seguintes requisitos de dados:
  1. Alunos possuem matrícula, nome, CPF, endereço (composto por logradouro, número, bairro, cidade e estado) e vários telefones (que podem ser de tipos diferentes). A matrícula de cada aluno é única.
  2. Alunos podem cursar mais de um curso na Universidade.
  3. Cursos possuem código e nome. O código de cada curso é único.
  4. Disciplinas possuem código, nome e carga horária. O código de cada disciplina é único.
  5. Uma disciplina pode ser alocada para vários cursos diferentes. Um curso terá várias disciplinas em sua grade.
  6. Alunos podem cursar disciplinas, desde que as disciplinas estejam disponíveis em seus cursos.
  7. Uma disciplina será ministrada por um único professor em todos os cursos onde esteja alocada.
  8. Professores possuem matrícula, nome, formação e salário. A matrícula de cada professor é única.

## Exercício 2

- Script DDL para o Sistema de RH de uma empresa, respeitando os seguintes requisitos de dados:
  1. Um **funcionário** terá matrícula, nome, endereço (logradouro e número), telefone fixo, telefone celular e salário. Cada matrícula é única.
  2. **Departamentos** são identificados por seus códigos. Departamentos possuem nome e ramal.
  3. Um funcionário poderá estar **lotado** em apenas um **departamento**. Um departamento poderá ter vários funcionários lotados nele. Quando um funcionário for lotado em um departamento, a data desta ocorrência deve ser guardada.
  4. Cada **projeto** possui um código único, nome, data de início e previsão de término.
  5. Um projeto poderá ser desenvolvido por vários funcionários. Um funcionário poderá se envolver em vários projetos.
  6. Um projeto será gerenciado por apenas um funcionário. Um funcionário poderá gerenciar vários projetos.

## Exercício 3

- Script DDL de um sistema bancário, segundo os seguintes requisitos:
  1. Uma **conta-corrente** terá um número e dígito, que a identificam. Uma conta terá um saldo.
  2. Uma **agência** é descrita por seu código único, seu nome, seu endereço e telefones.
  3. Uma conta-corrente está vinculada a apenas uma agência. Uma agência pode ter várias contas-correntes.
  4. Uma conta-corrente poderá ter vários correntistas. Um correntista pode abrir várias contas.
  5. São dados de **correntistas**: cpf, nome, endereço, telefones, data de nascimento, idade e renda mensal.
  6. Um correntista pode ter vários dependentes.
  7. Dependentes possuem nome.


## Exercício 4

- Script DDL de um sistema de controle de vôos, segundo os requisitos abaixo:
  1. Um **vôo** possui um número, data e hora de partida, data e hora de chegada, local de partida (aeroporto, cidade e país), local de chegada (aeroporto, cidade e país). Um vôo é identificado por seu número.
  2. Um **avião** possui modelo e quantidade de passageiros. O modelo identifica o avião.
  3. Um vôo possui um avião. Um avião pode ser utilizado em vários vôos.
  4. Um **passageiro** possui cpf, nome e endereço.
  5. Um passageiro pode voar em vários vôos diferentes. Um vôo tem vários passageiros diferentes.
  6. Pilotos e tripulantes são funcionários da empresa. Um funcionário possui matrícula única, nome e salário.
  7. Um vôo poderá ter vários tripulantes, que poderão participar de outros vôos.
  8. Um vôo terá um único capitão, que poderá pilotar outros vôos.



## Exercício 5

- Script DDL do sistema de uma loja, sabendo que:
  1. Um **produto** possui código (único), nome e quantidade de estoque.
  2. Os **clientes** possuirão CPF, nome, endereço e telefone.
  3. Um **cliente** poderá efetuar pedidos de compra. Cada **pedido** será identificado por um número e terá uma data.
  4. A cada pedido será composto por vários produtos e suas quantidades. Um produto poderá ser comprado várias vezes em outros pedidos.
  5. **Vendedores** terão matrícula, nome, endereço, telefone e salário. Cada matrícula é única.
  6. Cada pedido terá em seu registro o vendedor responsável. Um vendedor poderá efetuar vários pedidos.



Obrigado e até a Próxima Aula!

