



UNIVERSIDADE
FEDERAL DO PIAUÍ

Banco de Dados

Curso de Gestão de Dados

Centro de Educação Aberta e a Distância

Arlino Magalhães

arlino@ufpi.edu.br

Banco de Dados

Abordagem Relacional



TABELA: uma organização de palavras, números ou sinais, ou uma combinação de- les, como em colunas paralelas, para exibir um conjunto de fatos ou relações em uma forma definitiva, compacta e abrangente; uma sinopse ou esquema.

— Webster's Dictionary of the English Language

Agenda

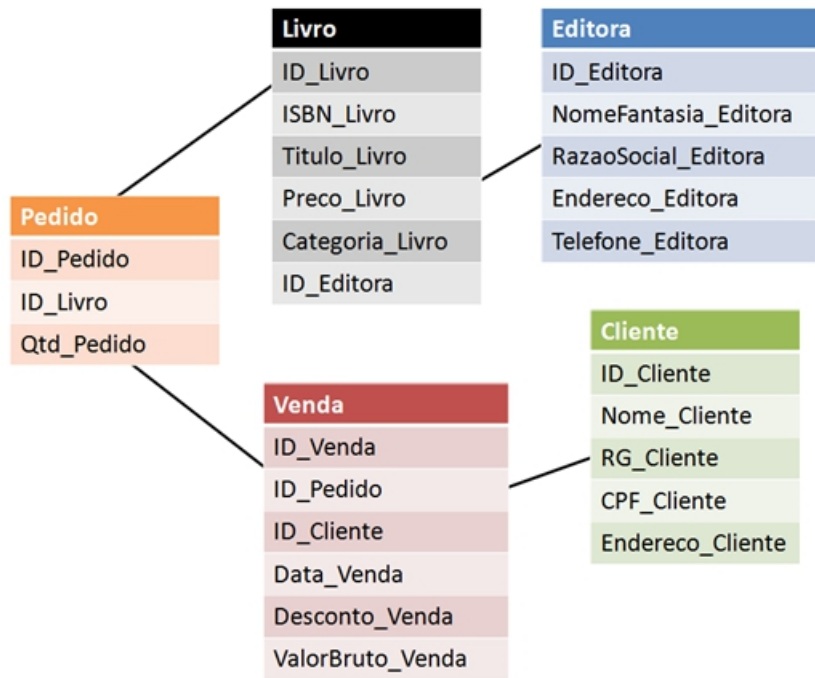
1. Conceitos básicos
2. Linguagens de bancos de dados
3. Construção de bancos de dados
4. Atualização de dados
5. Consultas a bancos d dados
6. Comandos adicionais
7. Exercícios

Conceitos básicos

1. Tabelas
2. Chave Primária
3. Chave Estrangeira
4. Domínios
5. Valores Vazios
6. Restrições de Integridade
7. Esquema de Banco de Dados

Bancos de dados relacionais

- Os bancos de dados relacionais utilizam a **abordagem relacional**.
- Um banco de dados relacional é composto de **tabelas** (ou relações).
- Uma tabela é composta por **linhas** que representam **registros** de valores armazenados.
- As **colunas** de uma tabela representam os atributos de cada registro.

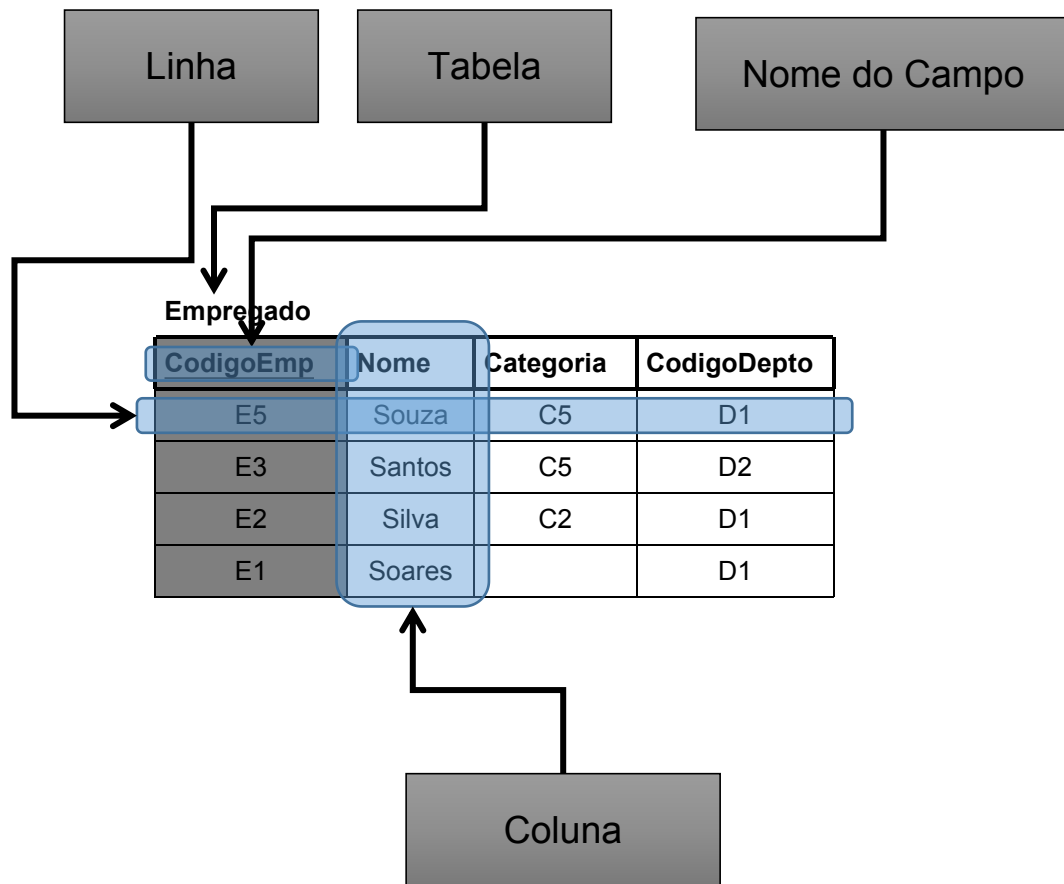


Tabela

É um conjunto não ordenado de **linhas** (tuplas). Cada linha é composta por uma série de **campos** (valor de atributo).

Cada campo é identificado pelo **nome de campo** (nome de atributo).

O conjunto de campos das linhas de uma tabela que possuem o mesmo nome formam uma **coluna**.



Chave primária

Uma chave primária (**primary key**) é uma coluna (ou uma combinação de colunas) cujos valores distinguem uma linha das demais dentro de uma tabela.

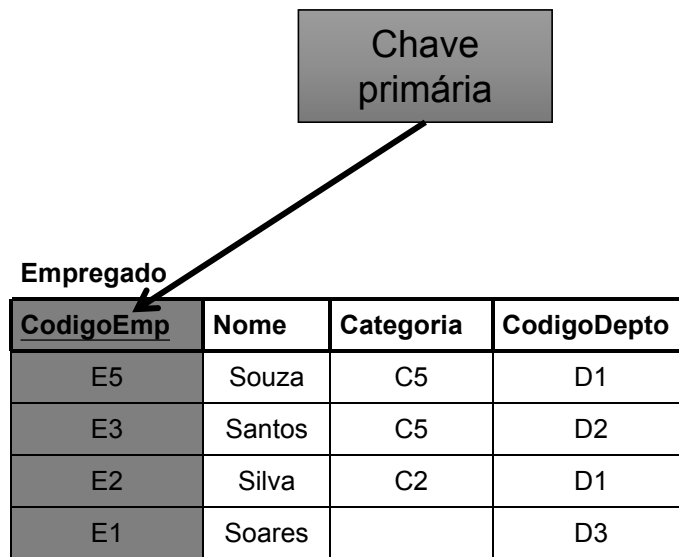


Diagram illustrating a primary key in a database table. A box labeled "Chave primária" points to the "CodigoEmp" column of the "Empregado" table, indicating it is the primary key.

| <u>CodigoEmp</u> | Nome | Categoria | CodigoDepto |
|------------------|--------|-----------|-------------|
| E5 | Souza | C5 | D1 |
| E3 | Santos | C5 | D2 |
| E2 | Silva | C2 | D1 |
| E1 | Soares | | D3 |

CódigoEmp é a chave primária de **Empregado**.

Empregado

| <u>CódigoEmp</u> | Nome | Categoria | CódigoDeppto |
|------------------|--------|-----------|--------------|
| E5 | Souza | C5 | D1 |
| E3 | Santos | C5 | D2 |
| E2 | Silva | C2 | D1 |
| E1 | Soares | | D3 |

CódigoEmp e **NoDepen** são a chave primária composta de **Dependente**.

Dependente

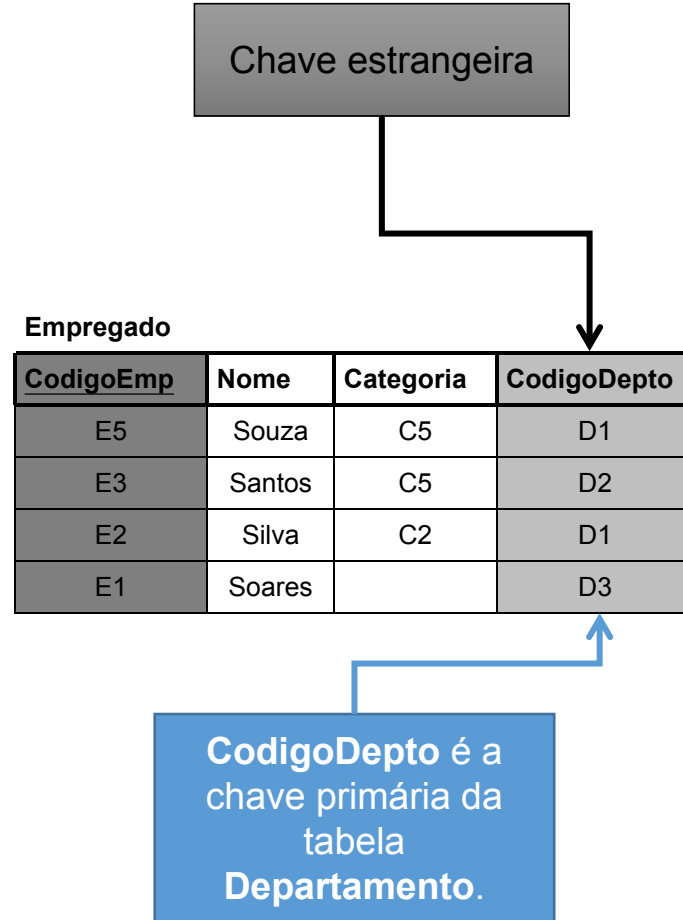
| <u>CódigoEmp</u> | <u>NumDep</u> | Nome | Tipo | DataNasc |
|------------------|---------------|-------|--------|------------|
| E1 | 1 | João | Filho | 12/12/1991 |
| E1 | 2 | Maria | Esposa | 01/01/1979 |
| E2 | 1 | Ana | Esposa | 05/11/1985 |
| E5 | 1 | Paula | Esposa | 04/07/1960 |
| E5 | 2 | José | Filho | 18/12/2004 |

Os valores (ou combinação de valores) da chave primária não podem se repetir na tabela.

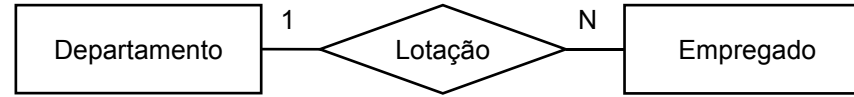
Chave estrangeira

É uma coluna (ou uma combinação de colunas) de uma tabela cujos valores são provenientes da chave primária de outra tabela.

Os relacionamentos entre tabelas são implementados através de chaves estrangeiras.



Chave estrangeira em relacionamento 1:N



A chave estrangeira deve possuir o mesmo valor da chave primária na tabela de origem.

Exemplo:
A chave estrangeira informa que os empregados E5 e E2 trabalham no departamento D1.

Departamento

| <u>CodigoDepto</u> | Nome |
|--------------------|------------|
| D1 | Compras |
| D2 | Engenharia |
| D3 | Vendas |

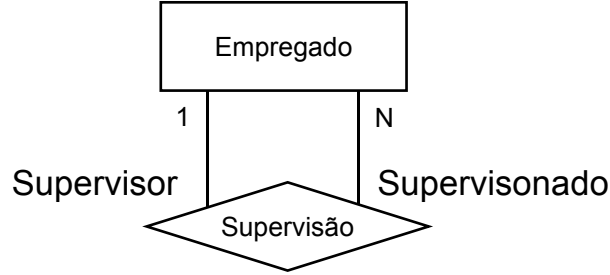
Chave estrangeira

Empregado

| <u>CodigoEmp</u> | Nome | Categoria | CodigoDepto |
|------------------|--------|-----------|-------------|
| E5 | Souza | C5 | D1 |
| E3 | Santos | C5 | D2 |
| E2 | Silva | C2 | D1 |
| E1 | Soares | | D3 |

O atributo **CodigoDepto** é chave estrangeira na tabela **Empregado** proveniente da chave primária **CodigoDepto** de **Departamento**.

Chave estrangeira em auto-relacionamento 1:N



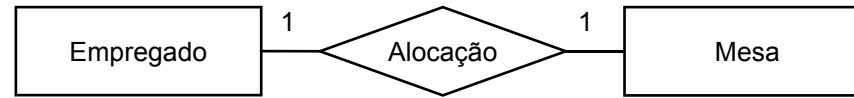
Uma chave estrangeira pode referenciar a chave primária da própria tabela.

Chave estrangeira

| Empregado | | | | |
|------------------|--------|-----------|-------------|-----------|
| <u>CodigoEmp</u> | Nome | Categoria | CodigoDepto | CodigoSup |
| E5 | Souza | C5 | D1 | |
| E3 | Santos | C5 | D2 | E5 |
| E2 | Silva | C2 | D1 | E5 |
| E1 | Soares | | D3 | E1 |

A chave estrangeira informa qual empregado é o supervisor de cada empregado.

Chave estrangeira em relacionamento 1:1



Empregado

| <u>CodigoEmp</u> | Nome | Categoria | CodigoMesa |
|------------------|--------|-----------|------------|
| E5 | Souza | C5 | 1 |
| E3 | Santos | C5 | 2 |
| E2 | Silva | C2 | 5 |
| E1 | Soares | | 4 |

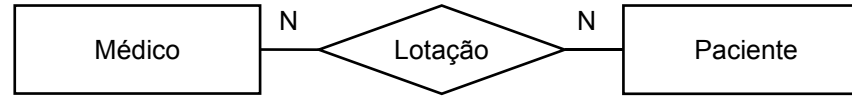
Chave estrangeira

Mesa ↓

| <u>Codigo</u> | Rótulo | Dimensão |
|---------------|--------|----------|
| 1 | M1 | 0,7x0,7 |
| 2 | M2 | 0,7x0,7 |
| 3 | M3 | 0,8x0,8 |
| 4 | M4 | 1,20x0,8 |
| 5 | M5 | 1,20x0,8 |

O atributo **CodigoMesa** é chave estrangeira na tabela **Empregado** proveniente da chave primária **Codigo** de **Mesa**.

Chave estrangeira em relacionamento N:N



Médico

| <u>Codigo</u> | Nome |
|---------------|-----------------|
| 1 | Hans Chucrutes |
| 2 | Stephen Strange |

Paciente

| <u>Codigo</u> | Nome |
|---------------|---------------|
| 1 | Woodpecker |
| 2 | Jane Foster |
| 3 | Faiza Hussain |

Chave estrangeira

Chave estrangeira

Lotacao

| <u>CodMed</u> | <u>CodPac</u> | <u>Data_hora</u> | Diagnóstico |
|---------------|---------------|---------------------|-------------|
| 1 | 1 | 04/09/2020 13:30:00 | |
| 1 | 1 | 11/11/2020 14:30:30 | |
| 2 | 2 | 11/11/2020 14:30:30 | |
| 2 | 3 | 30/07/2022 14:30:30 | |

O atributo **CodMed** e **CodPac** são chaves estrangeiras na tabela **Lotacao** provenientes das chaves primárias **Codigo** de **Médico** e **Codigo** de **Paciente**.

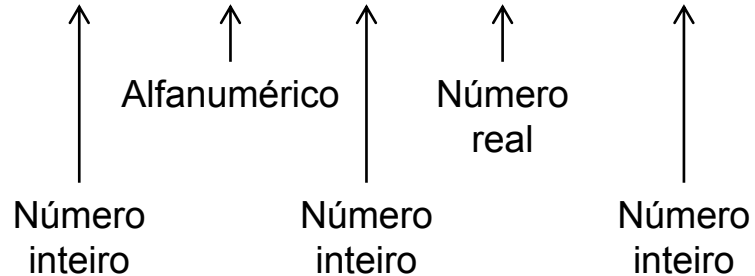
Os atributos **CodMed**, **CodPac** e **Data_hora** formam a chave primária em **Lotacao**.

Domínio

Conjunto de valores
que cada campo de
uma coluna pode
assumir.

Os valores mais comuns
são os numéricos e
alfanuméricos.

| <u>Codigo</u> | Nome | Idade | Salário | DataAdmissão |
|---------------|-------|-------|---------|--------------|
| 1 | José | 18 | 1000,00 | 04/07/2011 |
| 2 | Ana | 40 | 5000,00 | 18//09/2005 |
| 3 | Silva | 21 | 600,00 | 07/12/2015 |
| 3 | Maria | 32 | 1000,00 | 21/01/2010 |

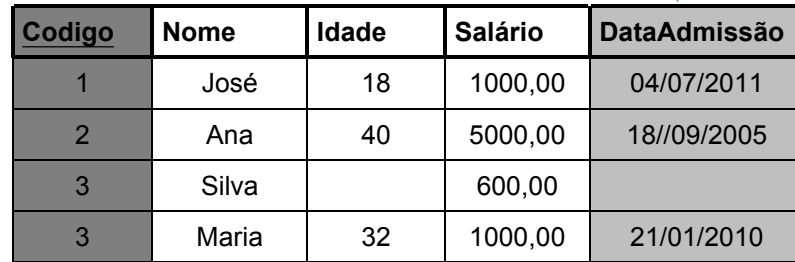


Valores Vazios

Colunas nas quais não são admitidos valores vazios.

As colunas nas quais podem aparecer campos vazios são chamadas de **colunas opcionais**.

Os campos **Idade** e **DataAdmissão** permite valores nulos.



| <u>Codigo</u> | Nome | Idade | Salário | DataAdmissão |
|---------------|-------|-------|---------|--------------|
| 1 | José | 18 | 1000,00 | 04/07/2011 |
| 2 | Ana | 40 | 5000,00 | 18//09/2005 |
| 3 | Silva | | 600,00 | |
| 3 | Maria | 32 | 1000,00 | 21/01/2010 |

Nos SGBD relacionais, a chave primária é sempre não vazia.

Os demais campos podem ser opcionais ou não.

Restrições de integridade

- Uma restrição de integridade é uma regra de consistência de dados que é garantida pelo próprio SGBD.
- Tipos de restrições:
 1. **Integridade de domínio:** os valores de um campo devem obedecer a definição do domínio especificado para a coluna).
 2. **Integridade de vazio:** especifica se os campos de uma coluna podem ou não ser vazios (se a coluna é obrigatória ou opcional).
 3. **Integridade de chave:** os valores da chave primária devem ser únicos e não vazios.
 4. **Integridade referencial:** os valores dos campos de uma chave estrangeira devem ser iguais a algum dos valores da chave primária da tabela referenciada.

Esquema de Banco de Dados

- Um esquema (especificação) de um banco de dados é a descrição da estrutura de como um banco de dados deve ser construído.
- O esquema de um banco de dados deve conter no mínimo as seguintes definições:
 - **tabelas** que formam o banco de dados;
 - **colunas** que as tabelas possuem e;
 - **restrições** de integridade.
- Na prática, na definição de esquemas relacionais são usadas diversas notações, que variam de um SGBD para o outro.

Notação resumida

Departamento

| <u>CodigoDepto</u> | Nome |
|--------------------|------------|
| D1 | Compras |
| D2 | Egnenharia |
| D3 | Vendas |

Empregado

| <u>CodigoEmp</u> | Nome | Categoria | CodigoDepto |
|------------------|--------|-----------|-------------|
| E5 | Souza | C5 | D1 |
| E3 | Santos | C5 | D2 |
| E2 | Silva | C2 | D1 |
| E1 | Soares | | D3 |



Notação resumida

Departamento (CodigoDepto, Nome)

Empregado (CodigoEmp, Nome, Categoria, CodigoDepto)

CodigoDepto referencia CodigoDepto em Departamento

Notação gráfica

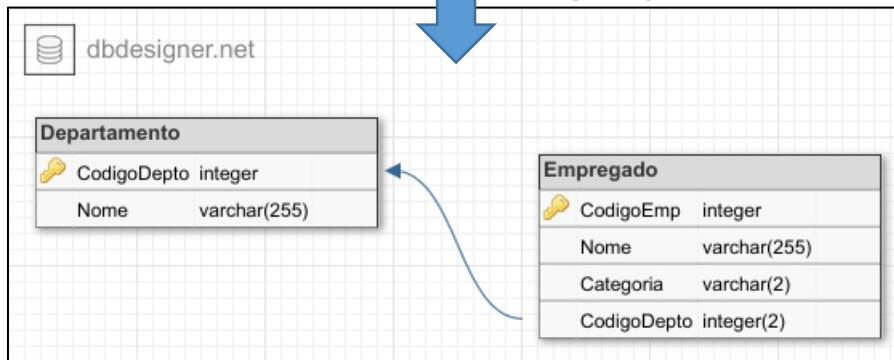
Departamento

| <u>CodigoDepto</u> | Nome |
|--------------------|-----------|
| D1 | Compras |
| D2 | Egneharia |
| D3 | Vendas |

Empregado

| <u>CodigoEmp</u> | Nome | Categoria | CodigoDepto |
|------------------|--------|-----------|-------------|
| E5 | Souza | C5 | D1 |
| E3 | Santos | C5 | D2 |
| E2 | Silva | C2 | D1 |
| E1 | Soares | | D3 |

Notação gráfica



Linguagens de bancos de dados

1. Álgebra relacional
2. SQL - Standard Query Language
3. Categorias de instruções Transact-SQL

Linguagens de bancos de dados

- Uma linguagem de banco de dados permite criar e manipular dados, como também controlar o acesso aos dados.
- Os sistemas de banco de dados comerciais utilizam linguagem de alto nível, como a **SQL - Structures Query Language** (Linguagem de Consulta Estruturada).
- Outros exemplos de linguagens de consulta são:
 - Álgebra Relacional e
 - Cálculo Relacional.

Álgebra relacional

- A álgebra relacional é uma linguagem de consultas procedural.
- Consiste em um conjunto de operações que tem como entrada uma ou duas relações e produz uma nova relação como resultado.
- A álgebra relacional é utilizada apenas no meio acadêmico. Os SGBDs comerciais utilizam a SQL.
- Principais operações:
 - σ , Π , \cup , \cap , \times .

SQL - Standard Query Language

- SQL é uma linguagem padrão de gerenciamento de dados utilizada pelos principais bancos de dados relacionais.
- Principais sistemas que utilizam SQL:
 - MySQL, Oracle, Firebird, Microsoft Access, PostgreSQL, HSQLDB.
- SQL é uma linguagem declarativa.
 - É necessário indicar apenas o objetivo pretendido.
- Principais comandos:
 - INSERT, SELECT, UPDATE, DELETE.
 - CREAT, ALTER, DROP.

Categorias de instruções (Transact-SQL)

DDL: Data Definition Language

Definição de dados

DML: Data Manipulation Language

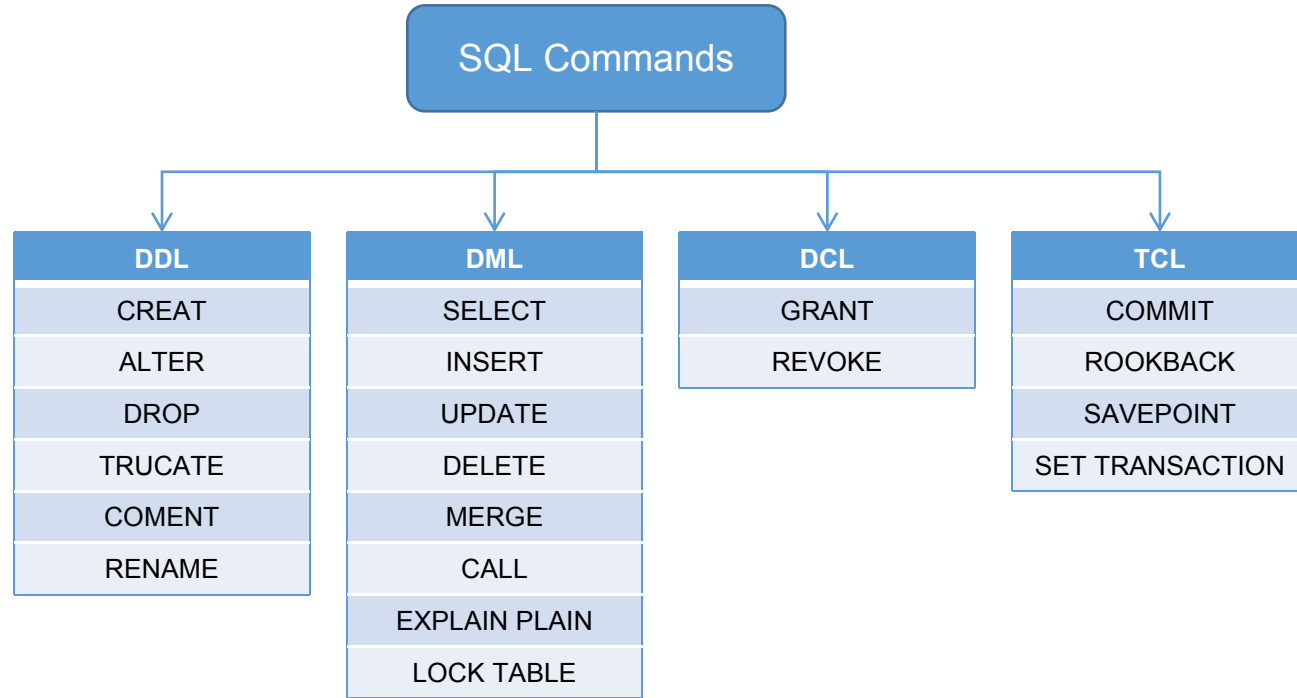
Manipulação de dados

DCL: Data Control Language

Controle de dados

TCL: Transactional Control Language

Controle de transações

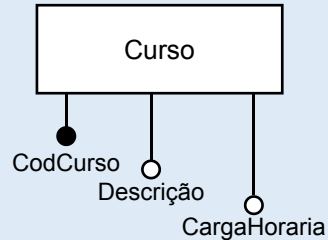


Construção de bancos de dados

1. Construção de tabelas usando SQL
2. Tipos de dados no PostgreSQL
3. Construção de tabelas no DB Designer
4. Construção de tabelas no PostgreSQL

Construção de tabelas usando SQL


Modelo Conceitual



Modelo Lógico

Curso (CodCurso, Descricao, CargaHoraria);

dbdesigner.net

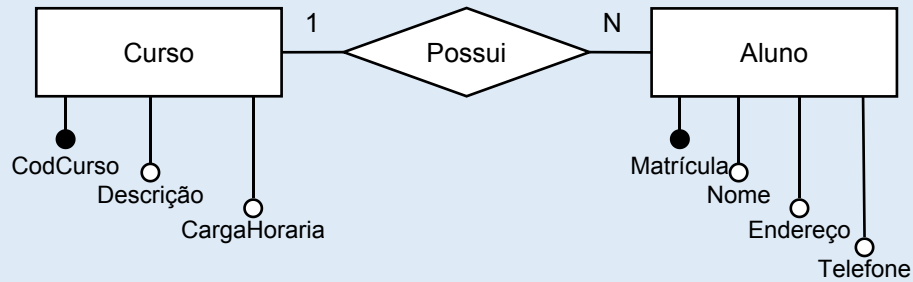
| Curso | |
|--|--------------|
|  CodCurso | integer |
| Descricao | varchar(255) |
| CargaHoraria | integer(255) |



```
CREATE TABLE Curso (  
    CodCurso INT NOT NULL,  
    Descricao VARCHAR(255) NOT NULL,  
    CargaHoraria INT NULL,  
    PRIMARY KEY (CodCurso )  
);
```

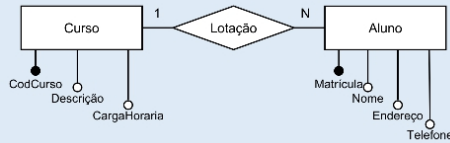
Construção de tabelas usando SQL

Modelo Conceitual



Construção de tabelas usando SQL

Modelo Conceitual



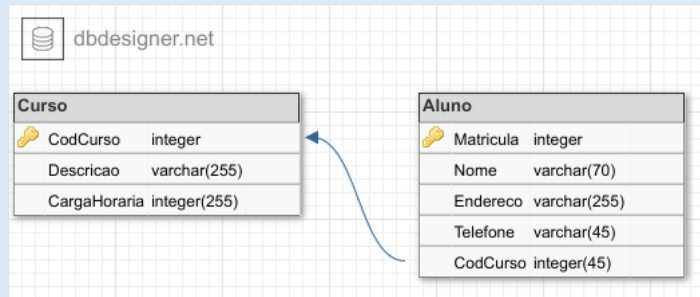
Modelo Lógico

```
Curso (CodCurso, Descricao, CargaHoraria);  
Alunos (Matricula, Nome, Endereco, Telefone, CodCurso)  
CodCurso referencia Cursos;
```

Construção de tabelas usando SQL

Modelo Lógico

Curso (CodCurso, Descricao, CargaHoraria);
Aluno (Matricula, Nome, Endereco, Telefone, CodCurso)
CodCurso referencia Cursos;



SQL

```
CREATE TABLE Curso (  
  CodCurso INT NOT NULL,  
  Descricao VARCHAR (255) NOT NULL,  
  CargaHoraria INT NULL,  
  PRIMARY KEY (CodCurso )  
);
```

```
CREATE TABLE Aluno (  
  Matrícula INT NOT NULL,  
  Nome VARCHAR (70) NOT NULL,  
  Endereco VARCHAR (255) NULL,  
  Telefone VARCHAR (45) NULL,  
  CodCurso INT NOT NULL,  
  PRIMARY KEY (Matrícula),  
  FOREIGN KEY (CodCurso)  
  REFERENCES Cursos (CodCurso)  
);
```

Tipos dados no PostgreSQL

- **Numérico:** smallint, integer, bigint, decimal, numeric, real, double precision, serial, bigserial.
- **Monetário:** money.
- **Data e hora:** timestamp, interval, time.
- **Booleano:** boolean.
- **Alfanumérico:** character varying(n), varchar(n), character(n), char(n), text.

Observação:

É necessário consultar a documentação de cada SGBD para saber quais tipos de dados ele possui.

DB Designer



DB DESIGNER

[FEATURES](#) [PLANS](#) [COMPANY](#) [SUPPORT](#) [LOGIN](#)

[START DESIGNING FOR FREE](#)

DB DESIGNER

Online Database Schema
Design and Modeling

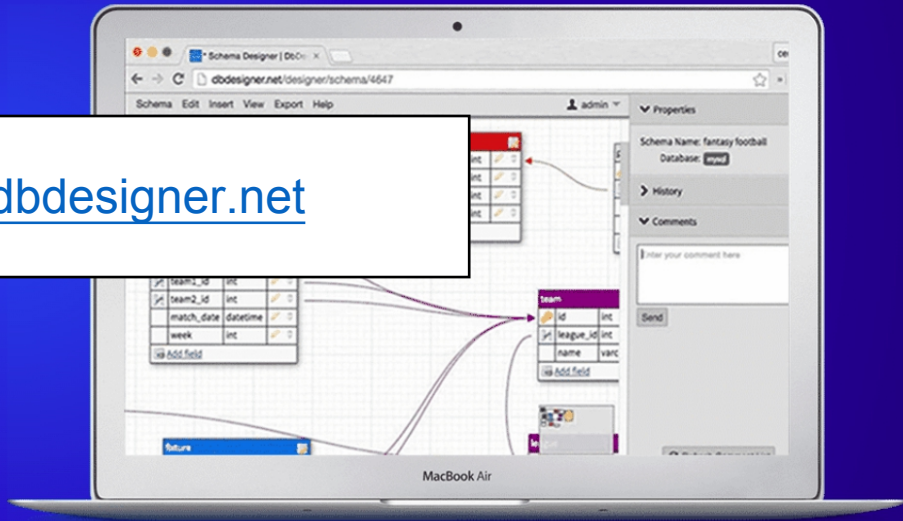
  

[REGISTER NOW](#)

[LOGIN AS GUEST](#)

START DESIGNING ONLINE FOR FREE

<https://www.dbdesigner.net>



PostgreSQL



[Home](#) [About](#) [Download](#) [Documentation](#) [Community](#) [Developers](#) [Support](#) [Donate](#) [Your account](#)



30th June 2022: [PostgreSQL 15 Beta 2 Released!](#)

PostgreSQL: The World's Most Advanced Open Source Relational Database

<https://www.postgresql.org>



New to PostgreSQL?

PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

There is a wealth of information to be found describing how to [install](#)



Latest Releases

2022-06-30 - PostgreSQL 15 Beta 2 Released!

The PostgreSQL Global Development Group announces that the [second beta release of PostgreSQL 15](#) second beta release of PostgreSQL 15 is now [available for download](#). This release contains

pgAdmin no PostgreSQL

The banner features a dark, atmospheric background with three elephants in a savanna setting. The pgAdmin logo is in the top left, and a search icon is in the top right. The main heading 'pgAdmin' is large and white, with 'PostgreSQL Tools' below it. A paragraph describes pgAdmin as the most popular Open Source administration and development platform for PostgreSQL. Another paragraph states it can be used on Linux, Unix, macOS, and Windows to manage PostgreSQL and EDB Advanced Server 10 and above. At the bottom, a screenshot of the pgAdmin 4 web interface is shown, displaying a tree view of servers and databases, and a dashboard with server sessions and transaction statistics.

pgAdmin

Home ▾ Development ▾ Documentation ▾ Download Support ▾

pgAdmin

PostgreSQL Tools

pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

pgAdmin may be used on Linux, Unix, macOS and Windows to manage PostgreSQL and EDB Advanced Server 10 and above.

pgAdmin 4

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser

- Servers (1)
 - PostgreSQL 13
 - Databases (6)

Dashboard Properties SQL Statistics Dependencies Dependents

Server sessions

| Total | Active | Idle |
|-------|--------|------|
| 12 | | |

Transactions per second

| Transactions | Commits | Rollbacks |
|--------------|---------|-----------|
| 3 | | |

Quick Links

Latest Videos



EDB Big Animal Dep...



pgAdmin: Query Tool



Amazon RDS Deplo...

Atualização de dados

1. Inserção de tuplas
2. Alteração de tuplas
3. Remoção de tuplas
4. Atualização de dados no PostgreSQL

Inserção de tuplas

INSERT INTO Aluno (Matricula, Nome, Endereco, Telefone, CodCurso)
VALUES (123, 'Flávio José', 'Rua Flores', 9999-0000, 1)



Aluno

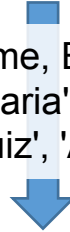
| <u>Matricula</u> | Nome | Endereco | Telefone | CodCurso |
|------------------|-------------|------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |

Inserção de tuplas

Aluno

| Matricula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |

INSERT INTO Aluno (Matricula, Nome, Endereco, Telefone, CodCurso)
VALUES (205, 'Aline Maria', 'Rua Pacheco', '3232-3232', 1),
(330, 'José Luiz', 'Av. Ipueiras', '3244-4444', 2)



Aluno


| Matricula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |

Inserção de tuplas

Aluno

| <u>Matricula</u> | Nome | Endereco | Telefone | CodCurso |
|------------------|-------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |

INSERT INTO Aluno **VALUES** (440, 'Flávia Silva', 'Rua P.', 'xxxx-xxxx', 2)



Os campos em INTO podem ser omitidas se todos os valores dos campos da tabela forem utilizados em VALUES seguindo a mesma ordem que foram criados na tabela.

Inserção de tuplas

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |

INSERT INTO Aluno VALUES (440, 'Flávia Silva', 'Rua P.', 'xxxx-xxxx', 2)



Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua P. | xxxx-xxxx | 2 |

Alteração de tuplas

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua P. | xxxx-xxxx | 2 |

UPDATE Aluno **SET** (Endereco = 'Rua Pacheco', Telefone = '5555-5555')
WHERE Matrícula = 440



Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------------|------------------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Exclusão de tuplas

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |



DELETE FROM Aluno WHERE Matrícula = 440

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |



Cuidado!



UPDATE Aluno **SET** (Nome= 'Fulano de tal')

DELETE FROM Aluno

Todas as linhas da
tabela serão atualizadas.

When you delete a
table instead of
row in Production DB



Consultas a bancos de dados

1. Operador de seleção
2. Operadores de comparação
3. Operador Like
4. Operadores lógicos
5. Projeção
6. Produto cartesiano

Operador de seleção

Álgebra Relacional

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as
linhas da tabela.

σ (Aluno)

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Operador de seleção

SQL

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas da tabela.

SELECT * FROM Aluno

O asterísco (*) indica que todos os campos serão selecionados.

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Operadores de comparação

- As operações de seleção podem obedecer a critérios de seleção através do uso de predicados.
- Para a formulação do predicado pode-se utilizar qualquer campo da tabela juntamente com qualquer símbolo de comparação válido.

| Principais símbolos | |
|---------------------|----|
| igual | = |
| diferente | <> |
| maior | > |
| menor | < |
| maior ou igual | >= |
| menor ou igual | <= |

Exemplos:

- `Codigo = 18`
- `Nome = 'Joaquim Magalhaes'`
- `Salario >= 2000`
- `Nascimento = '18/10/2014'`
- `Sexo <> 'M'`

Operadores de comparação

Álgebra Relacional

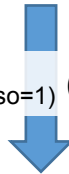
Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas cujo atributo CodCurso é igual a 1.



$\sigma_{(\text{CodCurso}=1)}$ (Aluno)



| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|-------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 |

Operadores de comparação

SQL

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas cujo atributo CodCurso é igual a 1.

→ **SELECT * FROM Aluno WHERE CodCurso = 1**

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|-------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 |

Operadores de comparação

SQL

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas cujo atributo Endereco é igual a 'Rua Pacheco'.

SELECT * FROM Aluno
WHERE Endereco = 'Rua Pacheco'

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|-------------|-------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 |

Operadores de comparação

SQL

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas
cujos valores do atributo
Nome comecem com a
letra 'F'.

SELECT * FROM Aluno
WHERE Nome = 'F%'

O símbolo % indica a
seleção de qualquer
combinação alfanumérica.

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|-------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Operadores de lógicos

- Operadores lógicos pode ser utilizados em predicados nas operações de seleção.

| Principais símbolos | |
|---------------------|-----|
| AND | e |
| OR | ou |
| NOT | não |

Exemplos:

- (Salario \geq 2000) **AND** (Salario \leq 5000)
- (CodCurso = 1) **OR** (CodCurso = 2)
- NOT** (Sexo \neq 'M')

Operadores de lógicos

SQL

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todas as linhas
cujos valores do atributo
Nome comecem com a
letra 'F' e também cujo
atributo CodCurso seja
igual a 2.

SELECT * FROM Alunos
WHERE Nome = 'F%' AND
CodCurso = 2

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|-------------|-----------|----------|
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Projeção

Álgebra Relacional

Aluno

| <u>Matricula</u> | Nome | Endereco | Telefone | CodCurso |
|------------------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona apenas os campos Nome e Telefone de todas as linhas da tabela.

$\Pi_{(\text{Nome}, \text{Telefone})}(\text{Aluno})$

| Nome | Telefone |
|--------------|-----------|
| Flávio José | 9999-0000 |
| Aline Maria | 3232-3232 |
| José Luiz | 3244-4444 |
| Flávia Silva | 5555-5555 |

Projeção SQL

Aluno

| Matricula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona apenas os campos Nome e Telefone de todas as linhas da tabela.

SELECT Nome, Telefone **FROM** Aluno

| Nome | Telefone |
|--------------|-----------|
| Flávio José | 9999-0000 |
| Aline Maria | 3232-3232 |
| José Luiz | 3244-4444 |
| Flávia Silva | 5555-5555 |

Produto Cartesiano

Curso

| <u>CodCurso</u> | Descricao | CargaHoraria |
|-----------------|-------------|--------------|
| 1 | Gestão | 120 |
| 2 | Veterinária | 220 |
| 3 | Agronomia | 190 |

Aluno

| <u>Matricula</u> | Nome | Endereco | Telefone | CodCurso |
|------------------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

O produto cartesiano envolve duas tabelas relacionadas.

CodCurso da tabela Aluno é chave estrangeira proveniente da tabela Curso.

Produto Cartesiano

Álgebra Relacional

Curso

| CodCurso | Descricao | CargaHoraria |
|----------|-------------|--------------|
| 1 | Gestão | 120 |
| 2 | Veterinária | 220 |
| 3 | Agronomia | 190 |

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todos os alunos e seus respectivos cursos.

$\sigma_{(\text{Aluno.CodCurso}=\text{Curso.CodCurso})} (\text{Aluno X Curso})$

O predicado '**Aluno.CodCurso = Curso.CodCurso**' relaciona a chave estrangeira com sua chave primária.

| Matrícula | Nome | Endereco | Telefone | CodCurso | CodCurso | Descricao | CargaHoraria |
|-----------|--------------|--------------|-----------|----------|----------|-------------|--------------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 | 1 | Gestão | 120 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 | 1 | Gestão | 120 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 | 2 | Veterinária | 220 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 | 2 | Veterinária | 220 |

Produto Cartesiano

Álgebra Relacional

Curso

| CodCurso | Descricao | CargaHoraria |
|----------|-------------|--------------|
| 1 | Gestão | 120 |
| 2 | Veterinária | 220 |
| 3 | Agronomia | 190 |

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

O produto cartesiano pode ser representado através da operação de **junção**.

Aluno ⋈ Curso

Na projeção não é necessário relacionar a chave estrangeira com a chave primaria.

| Matrícula | Nome | Endereco | Telefone | CodCurso | CodCurso | Descricao | CargaHoraria |
|-----------|--------------|--------------|-----------|----------|----------|-------------|--------------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 | 1 | Gestão | 120 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 | 1 | Gestão | 120 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 | 2 | Veterinária | 220 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 | 2 | Veterinária | 220 |

Produto Cartesiano

Álgebra Relacional

Curso

| CodCurso | Descricao | CargaHoraria |
|----------|-------------|--------------|
| 1 | Gestão | 120 |
| 2 | Veterinária | 220 |
| 3 | Agronomia | 190 |

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

Seleciona todos os alunos e seus respectivos cursos.

SELECT * FROM Aluno, Curso
WHERE Aluno.CodCurso = Curso.CodCurso

| Matrícula | Nome | Endereco | Telefone | CodCurso | CodCurso | Descricao | CargaHoraria |
|-----------|--------------|--------------|-----------|----------|----------|-------------|--------------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 | 1 | Gestão | 120 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 | 1 | Gestão | 120 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 | 2 | Veterinária | 220 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 | 2 | Veterinária | 220 |

Produto Cartesiano

Álgebra Relacional

Curso

| CodCurso | Descricao | CargaHoraria |
|----------|-------------|--------------|
| 1 | Gestão | 120 |
| 2 | Veterinária | 220 |
| 3 | Agronomia | 190 |

Aluno

| Matrícula | Nome | Endereco | Telefone | CodCurso |
|-----------|--------------|--------------|-----------|----------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 |
| 205 | Aline Maria | Rua Pacheco | 3232-3232 | 1 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 |

SELECT * FROM Aluno A, Curso C
WHERE A.CodCurso = C.CodCurso

Um *alias* pode ser utilizado para substituir o nome da tabela no predicado.

| Matrícula | Nome | Endereco | Telefone | CodCurso | CodCurso | Descricao | CargaHoraria |
|-----------|--------------|--------------|-----------|----------|----------|-------------|--------------|
| 123 | Flávio José | Rua Flores | 9999-0000 | 1 | 1 | Gestão | 120 |
| 205 | Aline Maria | Rua do Beco | 3232-3232 | 1 | 1 | Gestão | 120 |
| 330 | José Luiz | Av. Ipueiras | 3244-4444 | 2 | 2 | Veterinária | 220 |
| 440 | Flávia Silva | Rua Pacheco | 5555-5555 | 2 | 2 | Veterinária | 220 |

Comandos adicionais

1. Funções de agregação
2. Ordenação

Funções de agregação

- As funções de agregação executam um cálculo em um conjunto de valores e retornam um único valor.

| Função | Ação |
|--------|---|
| COUNT | Retorna a quantidade de linhas afetadas pelo comando. |
| SUM | Retorna o somatório do valor das colunas especificadas. |
| AVG | Retorna a média aritmética dos valores das colunas. |
| MIN | Retorna o menor valor da coluna de um grupo de linhas. |
| MAX | Retorna o maior valor da coluna de um grupo de linhas. |

Quantidade (count)

Empregado

| <u>Matrícula</u> | Nome | Salário | CodDepto |
|------------------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

Retorna a quantidade
de linhas da seleção.

SELECT COUNT (*) FROM Empregado

4

Quantidade (count)

Empregado

| <u>Matrícula</u> | Nome | Salário | CodDepto |
|------------------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT COUNT (*) AS quantidade FROM Empregado

Um rótulo pode ser definido para identificar o valor retornado.

quantidade

4

Somatório (sum)

Empregado

| Matrícula | Nome | Salário | CodDepto |
|-----------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT SUM (Salario) AS pagamentos FROM Empregado

Retorna o somatório
dos salários.

pagamentos

11.100,00

Média (avg)

Empregado

| <u>Matrícula</u> | Nome | Salário | CodDepto |
|------------------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT AVG (Salario) AS media FROM Empregado

Retorna o valor da
média dos salários.

media

2.775,00

Mínimo (min)

Empregado

| <u>Matrícula</u> | Nome | Salário | CodDepto |
|------------------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT MIN (Salario) AS menor FROM Empregado

Retorna o menor dos
salários.

menor

600,00

Máximo (min)

Empregado

| <u>Matrícula</u> | Nome | Salário | CodDepto |
|------------------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT MAX (Salario) AS maior FROM Empregado

Retorna o maior dos
salários.

maior

7.000,00

Ordenação

Empregado

| Matrícula | Nome | Salario | CodDepto |
|-----------|--------------|----------|----------|
| 123 | Flávio José | 2.000,00 | 1 |
| 205 | Aline Maria | 1.500,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |
| 440 | Flávia Silva | 600,00 | 2 |

SELECT (*) FROM Empregado ORDER BY Nome

Retorna todos os empregados ordenados por nome.

| Matrícula | Nome | Salario | CodDepto |
|-----------|--------------|----------|----------|
| 205 | Aline Maria | 1.500,00 | 1 |
| 440 | Flávia Silva | 600,00 | 2 |
| 123 | Flávio José | 2.000,00 | 1 |
| 330 | José Luiz | 7.000,00 | 2 |


Mais comandos SQL ...

- Agregação
 - Group by
 - Having
- Junções
 - Inner join
 - Left join
 - Right join
 - Full join
- Conjuntos
 - Union
 - Intersct
 - Except

Folha de dicas

https://github.com/swapnakpanda/Infographics/blob/main/Cheat%20Sheet/Database/SQL_CheatSheet.png

Abordagem



@swapnakpanda

SQL CHEATSHEET

✓ Genuine
✓ Authentic
✓ Quality

Categories

DDL : Data Definition Language
DQL : Data Query Language
DML : Data Manipulation Language
DCL : Data Control Language
TCL : Transaction Control Language

Commands

DDL
CREATE | DROP | ALTER | TRUNCATE
RENAME | COMMENT

DQL
SELECT

DML
INSERT | UPDATE | DELETE | LOCK
CALL | EXPLAIN PLAN

DCL
GRANT | REVOKE

TCL
COMMIT | ROLLBACK
SAVEPOINT | SET TRANSACTION

Operators

Arithmetic
+ - * / %

Bitwise
& | ^

Comparison
= < > <= >= !< !> <> !=

Compound
+= -= *= /= %=&= |= ^=

Logical
AND | OR | NOT | ANY
SOME | ALL | BETWEEN
IN | EXISTS | LIKE
IS NULL | UNIQUE

Database Objects

TABLE | VIEW | SYNONYM
SEQUENCE | INDEX | TRIGGER

Constraints

NOT NULL | UNIQUE
PRIMARY KEY | FOREIGN KEY
CHECK | DEFAULT

Aggregation Functions

AVG | COUNT
MAX | MIN | SUM

Aggregation Keywords

GROUP BY | HAVING

Important Keywords

WHERE | DISTINCT | LIMIT
ORDER BY | DESC | ASC
AS | FROM | SET | VALUES
CASE | DEFAULT

Joins

INNER JOIN

LEFT [OUTER] JOIN

FULL [OUTER] JOIN

RIGHT [OUTER] JOIN

Set Operations

UNION
UNION ALL

INTERSECT

EXCEPT
MINUS

DDL Examples

Create a Table
CREATE TABLE Students(
rollno int PRIMARY KEY,
fname varchar(255) NOT NULL,
lname varchar(255)
);

Adding a new column to the Table
ALTER TABLE Students
ADD email varchar(255);

Modifying the data type of existing column
ALTER TABLE Students
ALTER COLUMN lname varchar(512);

Removing an existing column from the Table
ALTER TABLE Students
DROP COLUMN email;

Truncate (remove all data) a Table
TRUNCATE TABLE Students;

Drop a Table
DROP TABLE Students;

DQL Examples

Fetch all data from a Table
SELECT * FROM Students;

Filter data from a Table
SELECT * FROM Students
WHERE rollno > 1234;

Fetch Maximum Age
SELECT max(age)
FROM Students;

Fetch Minimum Age
SELECT min(age)
FROM Students;

Fetch Sum of Age
SELECT sum(age)
FROM Students;

Fetch Average Age
SELECT avg(age)
FROM Students;

Fetch Average Age for each gender
SELECT avg(age)
FROM Students
GROUP BY gender;

Fetch count of records
SELECT count(*)
FROM Students;

Sort (order) fetched records
SELECT fname, lname
FROM Students
WHERE rollno > 1234
AND age < 15
ORDER BY gender;

Sort in descending order
SELECT fname, lname
FROM Students
WHERE rollno > 1234
AND age < 15
ORDER BY gender DESC;

Fetch from 2 Tables
SELECT fname, clsteacher
FROM Students
INNER JOIN Section
ON Students.section = Section.id;

DML Examples

Insert data (rows) into a Table
INSERT INTO Students(rollno, fname, lname)
VALUES (1234, 'Christiano', 'Ronaldo');

Update data (value of column) of a Table
UPDATE Students SET lname = 'Messi'
WHERE rollno = 1234;

Delete data (rows) from a Table
DELETE FROM Students WHERE rollno = 1234;

Aggregate and, Filter
SELECT section, count(*) AS studentcount
FROM Students
GROUP BY section
HAVING count(*) > 20;

Full Outer Join
SELECT fname, clsteacher
FROM Students
FULL JOIN Section
ON Students.section = Section.id;

Take prior permission before using it for commercial purposes. Attribution is required for all non-commercial uses.

Exercícios de SQL

Exercícios

- Dado o esquema de empregados e departamentos de uma empresa abaixo, fazer as SQLs que se pede:

```
dept (num, nome, orcamento);  
emp (matricula, nome, funcao, salario, depNum)  
      depNum referencia num em dept;
```


Exercícios

1. Fazer o script de criação das tabelas.
2. Povoar as tabelas (inserir dados) com as informações abaixo.

dept

| <u>Código</u> | Nome | Orçamento mensal |
|---------------|---------------|------------------|
| 1 | Administração | 5000 |
| 2 | Financeiro | 40000 |
| 3 | Informática | 10000 |
| 4 | Vendas | 1000 |

emp

| <u>Matrícula</u> | Nome | Função | Salário | Código do Departamento |
|------------------|-----------------|------------|---------|------------------------|
| 1 | João Henrique | Vendedor | 300 | 4 |
| 2 | Maria José | Vendedor | 1000 | 4 |
| 3 | Clara Silva | Estoquista | 300 | 4 |
| 4 | Pedro Silva | Digitador | 1500 | 3 |
| 5 | José Wagner | Diretor | 6000 | 1 |
| 6 | Carlos Henrique | Contador | 2000 | 2 |
| 7 | José Dias | Zelador | 200 | 1 |

Exercícios

3. Selecione de todas os campos da tabela de departamentos.
4. Selecione de todas os campos da tabela de empregados.
5. Selecione de todas os campos da tabela de empregados que possuem salário inferior a 1.000,00.
6. Selecionar os empregados que ganham salários entre 1000,00 e 2000,00.
7. Listar todos os salário pagos na empresa. Não mostrar valores iguais.
8. Selecione todos os departamentos cujo orçamento mensal seja maior que 10.000,00. Apresente o nome de tal departamento e seu orçamento anual.
9. Apresente os nomes e funções da cada funcionário contidos na tabela empresa, porém classificados pelo nome em ordem alfabética e depois em ordem alfabética invertida.

Exercícios

10. Selecionar os empregados que NÃO ganham salários entre 1000,00 e 2000,00.
11. Selecionar os empregados que tenham o primeiro nome igual a José.
12. Selecionar os empregados que possuem José em seus nomes.
13. Selecionar os empregados que NÃO possuam José em seus nomes.
14. Selecionar os vendedores que ganham salários entre 1000,00 e 2000,00.
15. Apresente a média, o maior, o menor e somatório dos salários pagos aos empregados. Cada valor pode ser feito utilizando uma seleção diferente.
16. Apresente a quantidade de empregados que ganham mais de 2.000,00.

Exercícios

17. Selecionar os nomes dos empregados e seus respectivos departamentos.
18. Selecionar os nomes dos empregados e seus respectivos departamento dos departamentos com orçamento maior 10.000,00.
19. Selecionar os nomes e salários dos empregados que trabalham no Financeiro.
20. Selecionar os nomes dos departamentos que possuem empregados com salário inferior a 400,00.

Respostas

```
1. CREATE TABLE dept (  
    num INT NOT NULL,  
    nome VARCHAR (50) NOT NULL,  
    orcamento INT NULL,  
    PRIMARY KEY (num)  
);
```

```
CREATE TABLE emp (  
    matricula INT NOT NULL,  
    nome VARCHAR (70) NOT NULL,  
    funcao VARCHAR (255) NULL,  
    salario FLOAT NULL,  
    CodCurso INT NOT NULL,  
    PRIMARY KEY (matricula),  
    FOREIGN KEY (depNum)  
    REFERENCES dept (num)  
);
```

Respostas

```
2. INSERT INTO dept VALUES(1, 'Administração', 5000);  
   INSERT INTO dept VALUES(2, 'Financeiro', 40000);  
   INSERT INTO dept VALUES(3, 'Informática', 10000);  
   INSERT INTO dept VALUES(4, 'Vendas', 1000);  
   INSERT INTO emp VALUES(1, 'João Henrique', 'Vendedor', 300, 4);  
   INSERT INTO emp VALUES(2, 'Maria José', 'Estoquista', 1000, 4);  
   INSERT INTO emp VALUES(3, 'Clara Silva', 'Vendedor', 300, 4);  
   INSERT INTO emp VALUES(4, 'Pedro Silva', 'Digitador', 1500, 3);  
   INSERT INTO emp VALUES(5, 'José Wagner', 'Diretor', 6000, 1);  
   INSERT INTO emp VALUES(6, 'Carlos Henrique', 'Contador', 2000, 2);  
   INSERT INTO emp VALUES(7, 'José Dias', 'Zelador', 200, 1);
```

Respostas

3. `SELECT * FROM dept`
4. `SELECT * FROM emp`
5. `SELECT * FROM emp WHERE salario < 1000`
6. `SELECT * FROM emp WHERE salario > 1000 and salario < 2000`
`SELECT * FROM emp WHERE salario BETWEEN 1000 AND 2000`
7. `SELECT DISTINCT(salario) FROM emp`
8. `SELECT nome, orcamento*12 FROM dept WHERE orcamento > 10000`
9. `SELECT nome, funcao FROM emp ORDER BY nome`
`SELECT nome, funcao FROM emp ORDER BY nome DESC`
10. `SELECT * FROM emp WHERE salario <= 1000 and salario => 2000`
`SELECT * FROM emp WHERE NOT (salario > 1000 and salario < 2000)`
11. `SELECT * FROM emp WHERE nome LIKE 'José%'`

Respostas

12. `SELECT * FROM emp WHERE nome LIKE '%José%'`
13. `SELECT * FROM emp WHERE NOT nome LIKE '%José%'`
14. `SELECT * FROM emp WHERE (salario BETWEEN 1000 AND 2000) AND (funcao = 'Vendedor')`
15. `SELECT AVG (salario) FROM emp;`
`SELECT MAX (salario) FROM emp;`
`SELECT MIN(salario) FROM emp;`
`SELECT SUM (salario) FROM emp;`
16. `SELECT COUNT(*) FROM emp WHERE salario > 2000`
17. `SELECT e.nome, d.nome FROM emp e, dept d WHERE e.depNum = d.num`
18. `SELECT e.nome, d.nome FROM emp e, dept d WHERE e.depNum = d.num AND d.orcamento > 10000`

Respostas

19. `SELECT e.nome, d.nome FROM emp e, dept d WHERE e.depNum = d.num AND d.orcamento > 10000`
20. `SELECT e.nome, e.salario FROM emp e, dept d WHERE e.depNum = d.num AND d.nome = 'Financeiro'`
`SELECT nome, salario FROM emp WHERE depNum = (SELECT num FROM departamento WHERE nome = 'Financeiro')`
20. `SELECT d.nome FROM emp e, dept d WHERE e.depNum = d.num AND e.salario < 400,00`
`SELECT nome FROM dept WHERE num IN (SELECT depNum FROM emp)`

Thanks!

Any questions?

You can find me at:
arlino@ufpi.edu.br
@arlino.magalhaes