



# Banco de Dados - SQL

Eduardo Arruda

# Eduardo Arruda

- SELECT(projeção de dados)
  - Junções
    - Critérios
    - Funções

## DML - Consultando Dados em Tabelas

- Estrutura Básica

**SELECT** → PROJEÇÃO

**FROM** → TABELA OU PRODUTO CARTESIANO DELAS

**WHERE** → SELEÇÃO

$$\Pi_{Coluna1[,Coluna2[,...]]} (\sigma_{Condição}(Tabela1 [X Tabela2 [X ... ]]))$$

**SELECT** *Coluna1[,Coluna2[, ... ]]*  
**FROM** *Tabela1,[Tabela2[, ... ]]*  
**WHERE** *Condição*

# DML - Consultando Dados em Tabelas

- Estrutura Genérica

```
SELECT [DISTINCT | ALL] { * | [Tabela.]Coluna1 [AS Alias1]  
    [ [Tabela.]Coluna2 [AS Alias2] [, ...]]}  
FROM Tabela1 [, Tabela2 [, ... ] ]  
[WHERE {Condição Simples | Condição de Sub-consulta} ]  
[ORDER BY Coluna1 [ASC | DESC] [, Coluna2 [ASC | DESC] [, ... ]]]  
[GROUP BY Coluna1 [, Coluna2 [, ... ]]] [HAVING Condição ] ]  
[ {UNION | INTERSECT | EXCEPT} SELECT ... ]
```

# SQL – CONSULTANDO DADOS

- Existem várias formas de obtenção de dados em mais de uma tabela, a utilização do JOIN com diversos parâmetros diferentes para atender todas as necessidades de aglutinação de dados, assim como, usar os operadores de comparação entre os campos chaves das tabelas. Também será possível obter dados utilizando resultado de uma consulta como parâmetro de comparação.

# SQL – CONSULTANDO DADOS

- **Ligação entre tabelas (Join)**
- Diz-se que tabelas estão relacionadas se tiverem campos comuns (ch.primária e ch. estrangeira).
- O efeito do JOIN é a criação de uma tabela temporária em que cada par de linhas, que satisfaça a condição de ligação, são ligados para formar uma única linha.
- A ligação é sempre estabelecida à frente da cláusula WHERE usando o operador relacional da igualdade (=).

# SQL – CONSULTANDO DADOS

- **FROM** estabelece o produto cartesiano entre as tabelas listadas
- **WHERE** filtra as linhas úteis segundo a condição especificada
- Pode-se misturar as cláusulas de comparação, vistas anteriormente, juntamente com AND, OR e NOT para formar equações de ligações mais complexas
- **Para se ligar várias tabelas, usa-se o operador lógico AND.**
- É preciso ter muito cuidado com os JOINS, pois exigem alto custo de execução (implicam diretamente na performance).

# SQL – CONSULTANDO DADOS

## Ligação entre tabelas (Join) – exemplos

**/\* Projetar todos os livros (título) e seus autores (nome) nomeando a tabela para facilitar a digitação\*/:**

```
SELECT AU.Nome, LI.Titulo  
FROM AUTOR AU, LIVRO LI  
WHERE AU.CodAutor = LI.CodAutor;
```



# SQL – CONSULTANDO DADOS

## Ligação entre tabelas (Join) – exemplos

**/\* Projetar todos os autores (nome), seus livros (título) e editoras (razão), onde a razão da editora seja Campos ou Makron \*/:**

```
SELECT AUTOR.Nome, LIVRO.Titulo, EDITORA.Razao  
FROM AUTOR, LIVRO, EDITORA  
WHERE AUTOR.CodAutor = LIVRO.CodAutor AND  
LIVRO.CodEditora = EDITORA.CodEditora AND  
EDITORa.Razao IN ('Campos', 'Makron');
```

# SQL – CONSULTANDO DADOS

## Ligação entre tabelas (Join) – exemplos

**/\* Projetar Autores (nome), a Editora (Razão) e a Quantidade de Livros Publicados por editora, nomeando a tabela para facilitar a digitação\*/:**

```
SELECT AU.Nome, ED.Razão, Count(LI.Publicação) as QuantPub  
FROM AUTOR AU, LIVRO LI, ED.Editora  
WHERE AU.CodAutor = LI.CodAutora and  
LI.CodEditora = ED.CodEditora  
Group by AU.Nome, ED.Razão;
```

# SQL – CONSULTANDO DADOS

- Todos os comando vistos até agora, poderão ser utilizados juntos, dentro de uma consulta(query), para obtenção de dados, incluindo as junções, dependendo do objetivo do relatório.

# Obrigado!

---

Eduardo Arruda