

# Linguagem SQL – Query simples – restrições – álgebra

Profa Dra Jeroniza Nunes  
Marchaukoski

## Linguagem SQL DML - Álgebra

- **DML (Data Manipulation Language)- Linguagem de Manipulação de Dados**
  - Consultas a dados
  - Inclusões, Modificações e Exclusões de dados
- **Álgebra relacional: coleção de operações utilizadas para manipular relações.**

# Linguagem SQL DML - Álgebra

- **SQL DML de consulta – Estrutura básica**
- **Sintaxe:**
  - **Select <campo1>, <campo2>**  
corresponde à operação de projeção da álgebra relacional.
  - **From <tabela>, <tabela>** corresponde à operação de produto cartesiano da álgebra relacional.
  - **Where <condição>** corresponde à seleção do predicado da álgebra relacional.

# Linguagem SQL DML - Álgebra

- **Álgebra – Operadores Básicos**

- **SELECT** é usada para selecionar um subconjunto de tuplas (linhas) de uma relação (tabela) as quais devem satisfazer um predicado (condição) de seleção.
- **Sintaxe ( $\sigma$  sigma):**
  - $\sigma$  <predicado de seleção> (<nome da relação>)
- **Exemplo:**
  - $\sigma$  **salario > 1000** (empregado)
- **Em SQL:**
  - **Select \* from empregado** **where salario > 1000**

# Linguagem SQL DML - Álgebra

- **Álgebra – Operadores Básicos**

- PROJECT seleciona certos atributos (colunas) da relação (tabelas) e descarta outros.
- Sintaxe ( $\pi$  pi):
  - $\pi$  <lista de atributos> (<nome da relação>)
- Exemplo:
  - $\pi$  nome, salario (empregado)
- Em SQL:
  - **Select nome, salario** from empregado

# Linguagem SQL DML - Álgebra

- **Consulta projeção: Mostre o nome e telefone de todos os alunos da tabela Aluno**
  - **Select nome, tel from Aluno**
  - **$\pi$  nome, tel**<sup>(aluno)</sup>

# Linguagem SQL DML - Álgebra

- **Consulta seleção e projeção: Mostre o nome e telefone dos alunos do sexo 'M' tabela Aluno**
  - **Select nome, tel from Aluno where sexo= 'M'**
  - **$\pi$  nome, tel ( $\sigma$  sexo = 'M'(aluno) )**

# Linguagem SQL DML - Álgebra

- **Consulta todos os campos sem seleção**

– **Select \* from Aluno**

— (aluno)



# Linguagem SQL DML - Operadores

- **Consulta simples com restrições para números**

- **=** : igual a
- **>** : maior que
- **>=** : maior que ou igual a
- **<** : menor que
- **<=** : menor que ou igual a
- **Between <início> and <fim>** : intervalo de valores

# Linguagem SQL DML - Operadores

- **Consulta simples com restrições para números**

- **Exemplo com intervalo:**

- **Select \* from empregado where salario  
between 1000 and 5000;**

## Linguagem SQL DML - Operadores IS NULL

- Is null: é um valor nulo
  - Exemplo 1:
    - SELECT NOME, CONTATO
    - FROM EMPRESA
    - WHERE CONTATO IS NULL;
  - Exemplo 2 (negando):
    - SELECT NOME, CONTATO
    - FROM EMPRESA
    - WHERE CONTATO IS NOT NULL;

# Linguagem SQL DML - Operadores String

- **Consulta com String**

- **Select \* from empregado where nome like 'AN%'**
  - Mostra todos os empregados cujo nome comece com "AN"
- **Select \* from empregado where nome like '%O'**
  - Mostra todos os empregados cujo nome termine com "O"
- **Select \* from empregado where nome like '%MA%'**
  - Mostra todos os empregados cujo nome tenha em qualquer parte a string MA (MARIA, ROMARIO)
- **select \* from aluno where nome like ' \_ \_ \_ '**
  - Mostra todos os alunos cujo nome tenha três letras (underline)
- **select \* from aluno where nome like ' \_ \_ A '**
  - Mostra todos os alunos cujo nome tenha três letras sendo que a última deve ser "A"

## Linguagem SQL DML - Operadores Lista

- **IN: Com lista (constantes, SQL encadeado)**
  - **Select \* from empregado where salario in (500, 700);**
  - **Select \* from empregado where nome in ('Joana', 'Ana');**

## Linguagem SQL DML - Operadores Negativos

- **<>** : diferente
- **not nome\_coluna =** : diferente da coluna
- **not nome\_coluna >** : não maior que
- **not between** : não entre dois valores informados
- **not in** : não existente numa dada lista de valores
- **not like** : diferente do padrão de caracteres informado
- **is not null** : não é um valor nulo

## Linguagem SQL DML - Operadores Negativos

- **Exemplo:**
- **Selecione os Empregados cujos salários sejam menores que 1000 ou maiores que 3500.**
  - SELECT NOME, SALARIO
  - FROM EMPREGADO
  - WHERE SALARIO NOT BETWEEN 1000 AND 3500;

# Linguagem SQL DML - Operadores (AND, OR)

- **Exemplos:**
- **Apresente todos os funcionários com salários entre 200 e 700 e que sejam Vendedores.**
  - SELECT NOME, SALARIO, CARGO
  - FROM EMPREGADO
  - WHERE SALARIO BETWEEN 700 AND 2000
  - AND CARGO = 'VENDEDOR';
- **Apresente todos os funcionários com salários entre 200 e 700 ou que sejam Vendedores.**
  - SELECT NOME, SALARIO, CARGO
  - FROM EMP
  - WHERE SALARIO BETWEEN 700 AND 2000
  - OR CARGO = 'VENDEDOR';



## Linguagem SQL DML - Operadores (AND, OR)

- **Exemplos (continuação)**
- **Apresente todos os funcionários com salários entre 200 e 700 e que sejam Vendedores ou Balconistas.**
  - SELECT EMPNOME, EMPSALA, EMPSERV
  - FROM EMP
  - WHERE EMPSALA BETWEEN 700 AND 2000
  - AND ( EMPSERV = 'BALCONISTA' OR EMPSERV = 'VENDEDOR' );

# Linguagem SQL DML - Rename

- **Campo:**
  - nome\_antigo as nome\_novo ... Campo
  - nome\_campo as apelido
- **Tabela:**
  - nome\_antigoTB nome\_novoTB...  
Tabela
  - nome\_antigoTabela apelido
- **Exemplo:**
  - Select **a**.nome, **a.tel** as **telefone** from **aluno a**;

## Linguagem SQL DML - Cálculo

- **Exemplo: mostre o salário atual e o salário com aumento de 10%**
  - **SELECT** salario as atualSalario, **salario**  
**\* 1.1** as novoSalario
  - **FROM** empregado;

# Linguagem SQL DML - Ordenação

- **O padrão é ordem crescente (asc).**
  - **Exemplo 1: Mostre os funcionários em ordem ascendente (A a Z) e seus cargos**
    - **SELECT NOME, CARGO**
    - **FROM EMPREGADO ORDER BY NOME;**
  - **Exemplo 2: Mostre os funcionários em ordem descendente (Z a A) e seus cargos**
    - **SELECT NOME, CARGO**
    - **FROM EMP ORDER BY NOME DESC;**

# Linguagem SQL DML - Álgebra

- **Produto cartesiano**

```
SELECT CURSO. Nome, ALUNO. Nome,  
       ALUNO.dtNasc  
FROM CURSO, ALUNO  
WHERE CURSO.cod = ALUNO.codCurso
```

PK                      FK

# Linguagem SQL DML - Álgebra

- **Produto cartesiano**

- $\pi_{\text{aluno.nome, curso.nome}}$   
 $(\sigma(\text{aluno.curso}=\text{curso.cod\_curso})$  **(alunoXcurso)**

- **Usando apelido no SQL**

- **Select a.nome as aluno, c.nome as curso**  
**from aluno a, curso c**  
**where a.curso = c.cod\_curso;**