



# SQL

## Linguagem de Manipulação de Dados

Prof. Humberto Razente

Bloco B - sala 1B144

# Semana passada...

## ◆ Linguagem de Definição dos Dados (DDL)

- CREATE

- ALTER

- DROP

# SQL DML - Linguagem de Manipulação de Dados

## ◆ SELECT ... FROM ...

- lista linhas de uma ou mais tabelas

## ◆ INSERT INTO ...

- insere dados em uma tabela

## ◆ DELETE FROM ... WHERE ...

- remove dados de tabelas já existentes

## ◆ UPDATE ... SET ... WHERE ...

- altera dados específicos de uma tabela

# INSERT

```
INSERT INTO nome_tabela  
VALUES ( V1, V2, ..., VN );
```

```
INSERT INTO nome_tabela (A1, A2, ..., An)  
VALUES ( V1, V2, ..., VN );
```

# Outra construção: INSERT

```
INSERT INTO nome_tabela  
  SELECT ...  
  FROM ...  
  WHERE ... ;
```

- ◆ Tuplas resultantes da cláusula SELECT serão inseridas na tabela nome\_tabela

# DELETE

```
DELETE FROM nome_tabela  
WHERE predicado ;
```

## ◆ Cláusula WHERE

- é opcional:
  - ◆ todas as tuplas da tabela são eliminadas quando restrição não é especificada
  - ◆ a tabela continua a existir

# DELETE ...

- ◆ Remove tuplas inteiras
- ◆ Opera apenas em uma relação
- ◆ Tuplas de mais de uma relação a serem removidas:
  - um comando DELETE para cada relação
- A remoção de uma tupla de uma relação deve ser propagada para tuplas em outras relações devido às restrições de integridade referencial

# Exemplos

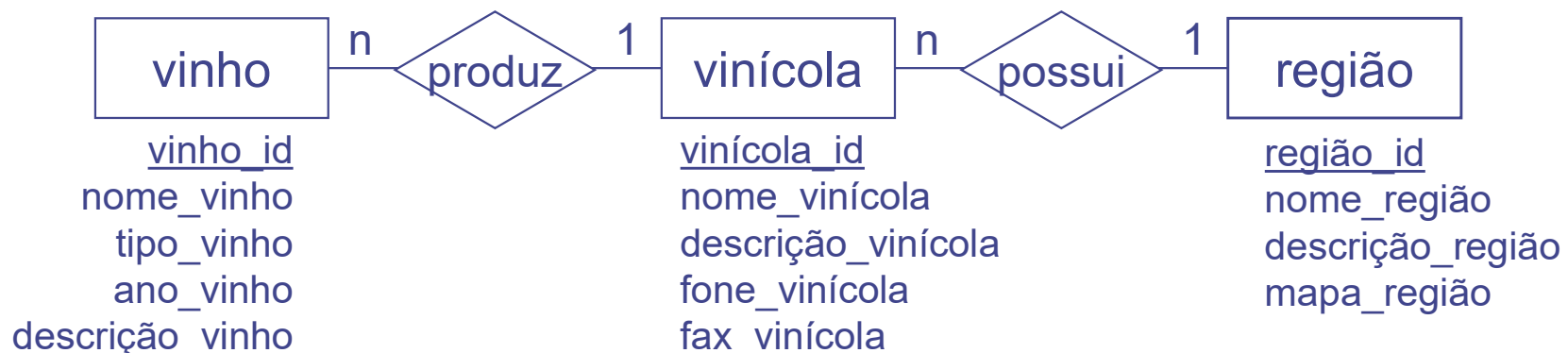
## ◆ DELETE FROM vinícola

WHERE vinícola\_id = 10;

- remove a tupla referente a vinícola\_id = 10
  - ◆ tabela vinícola
  - ◆ tabela vinho (i.e., se a opção CASCADE foi especificada na cláusula ON DELETE do campo vinícola\_id desta tabela)

## ◆ DELETE FROM região;

- remove todas as linhas da tabela região





# UPDATE

```
UPDATE nome_tabela  
SET coluna = <valor>  
WHERE predicado ;
```

## ◆ Cláusula WHERE

- é opcional

## ◆ Exemplos de <valor>

- NULL
- 'string'
- UPPER 'string'

# UPDATE ...

- ◆ Opera apenas em uma relação
- A atualização da chave primária deve ser propagada para tuplas em outras relações devido às restrições de integridade referencial

# Exemplos

- ◆ Alterar os anos de produção de vinhos de 2007 para 2003.

```
UPDATE vinho
```

```
    SET ano_vinho = 2003
```

```
    WHERE ano_vinho = 2007;
```

- ◆ Suponha o atributo adicional **preço** na tabela vinho. Aumentar os preços dos vinhos em 10%.

```
UPDATE vinho SET preço = preço * 1.10;
```

# Exemplos


◆ UPDATE vinícola

SET vinícola\_id = 10

WHERE vinícola\_id = 2;

- altera o valor de vinícola\_id = 2 para vinícola\_id = 10
  - ◆ tabela vinícola
  - ◆ tabela vinho (i.e., se a opção CASCADE foi especificada na cláusula ON UPDATE do campo vinícola\_id desta tabela)

# SELECT



```
SELECT <lista de atributos e funções>  
FROM <lista de tabelas>  
[ WHERE predicado ]  
[ GROUP BY <atributos de agrupamento> ]  
[ HAVING <condição para agrupamento> ]  
[ ORDER BY <lista de atributos> ]
```

# SELECT

## ◆ Cláusula SELECT

- lista os atributos e/ou as funções a serem exibidos no resultado da consulta
- **corresponde à operação de projeção da álgebra relacional**

## ◆ Cláusula FROM

- especifica as relações a serem examinadas na avaliação da consulta
- **corresponde à operação de produto cartesiano da álgebra relacional**

# SELECT

## ◆ Cláusula WHERE

- especifica as condições para a seleção das tuplas no resultado da consulta
  - ◆ as condições devem ser definidas sobre os atributos das relações que aparecem na cláusula FROM
- inclui condições de junção
- corresponde ao predicado de **seleção** da álgebra relacional

# SELECT

## ◆ Resultado de uma consulta

- ordem de apresentação dos atributos
  - ◆ ordem dos atributos na cláusula SELECT
- ordem de apresentação dos dados
  - ◆ ordem ascendente ou descendente de acordo com a cláusula ORDER BY
  - ◆ sem ordenação
- duas ou mais tuplas podem possuir valores idênticos de atributos
  - ◆ eliminação de tuplas duplicadas
    - SELECT DISTINCT



# Cláusula WHERE

SELECT

FROM

WHERE <atributo> <operador>  
          <valor | atributo | lista de valores>

## ◆ Operadores

- conjunção de condições: AND
- disjunção de condições: OR
- negação de condições: NOT

# Cláusula WHERE

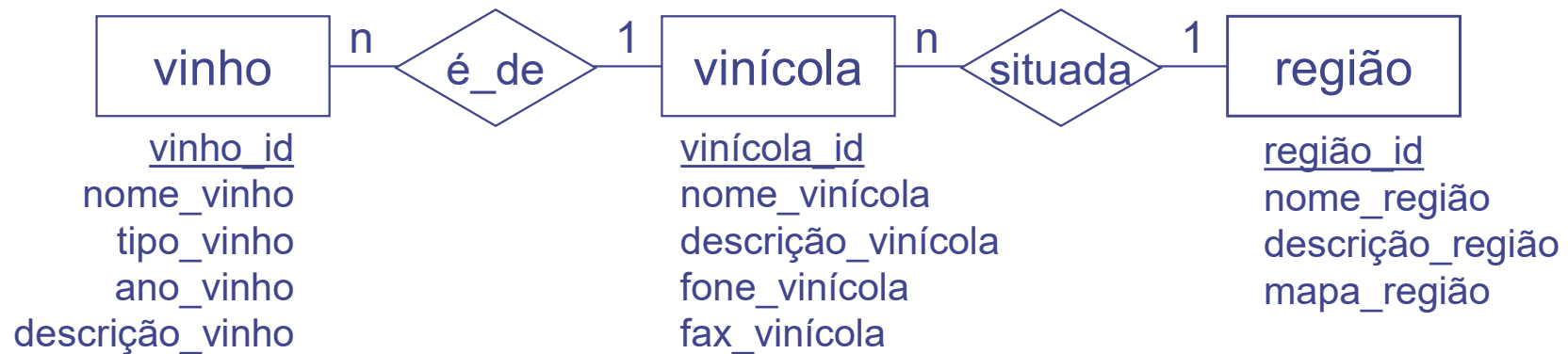
## ◆ Alguns operadores de comparação

igual a	=	diferente de	<> !=
maior que	>	maior ou igual a	>=
menor que	<	menor ou igual a	<=
teste de nulo	IS NULL <i>ou</i> IS NOT NULL	igual a algum de vários valores	<i>expressão</i> IN ( <i>lista_valores</i> )
entre <i>dois</i> valores	BETWEEN <i>valor1</i> AND <i>valor2</i>	de cadeias de caracteres	LIKE <i>ou</i> NOT LIKE

# Operador LIKE

- caracter coringa
  - ◆ % (por cento): substitui qualquer *string*
  - ◆ \_ (underscore): substitui qualquer *caractere*
- característica
  - ◆ operador sensíveis ao caso
    - letras maiúsculas são consideradas diferentes de letras minúsculas
- exemplos:
  - ◆ select \* from aluno where nome like "JOAO%"
  - ◆ select \* from aluno where nome like "%JOAO%"
  - ◆ select \* from aluno where nome like "MARI\_"

# Relações Base



- ◆ **região** (região\_id, nome\_região, mapa\_região, descrição\_região)
- ◆ **vinícola** (vinícola\_id, nome\_vinícola, descrição\_vinícola, fone\_vinícola, fax\_vinícola, **região\_id**)
- ◆ **vinho** (vinho\_id, nome\_vinho, tipo\_vinho, ano\_vinho, descrição\_vinho, **vinícola\_id**)

# Cláusula WHERE

## ◆ Exemplos

- `select * from região`  
`WHERE nome_região LIKE 'Mar%'`
  - ◆ qualquer string que se inicie com 'Mar'
- `select * from região`  
`WHERE nome_região LIKE 'Mar_'`
  - ◆ qualquer string de 4 caracteres que se inicie com 'Mar'

# Exercício

Paciente (CPF, Nome, DataNascimento, Endereco)

Consulta (CPF, CodigoConvenio, CRM, UF, Data, Horario, Valor)

Médico (CRM, UF, Nome, CPF, Endereco, telefone)

- 1) Criar esquema em SQL do modelo relacional acima.
- 2) Escrever comandos SQL para inserir os seguintes dados no esquema:
  - 1) Utilizando a sintaxe do insert que contém os nomes dos atributos
    - 1) Paciente < 456, 'Jose Maria', 'Av. Floriano Peixoto, 123' >
  - 2) Utilizando a sintaxe do insert que não contém os nomes dos atributos
    - 1) Paciente < 123, 'Joao Silva', '7/8/90', 'Av. Joao Naves, 2121' >
    - 2) Paciente < 234, 'Maria Pereira', '8/9/91', 'Av. Rondon Pacheco, 1000' >
    - 3) Medico < 1000, 'MG', 'Antonio Carlos Sousa', 789, 'Av. Getulio Vargas, 10', '3210-1234' >
    - 4) Medico < 2000, 'SP', 'Ana Julia Matos', 890, 'Av. Cipriano, 1', '3210-3456' >
    - 5) Consulta < 123, 1, 1000, 'MG', '19/01/2015', '14:00', 100.00 >
- 3) Considere que existe uma tabela de nome ConsultasAgendadas de um sistema antigo, com mesmo esquema da tabela Consulta, que contém outras consultas agendadas. Escrever insert para inserir esses dados da tabela ConsultasAgendadas na tabela consulta

# Exercício

Paciente (CPF, Nome, DataNascimento, Endereco)

Consulta (CPF, CodigoConvenio, CRM, UF, Data, Horario, Valor)

Médico (CRM, UF, Nome, CPF, Endereco, telefone)

Responda as seguintes consultas em SQL:

- 1) Buscar o nome e o CPF dos pacientes que nasceram na década de 90.
- 2) Buscar os médicos com nomes que iniciam com "J"
- 3) Buscar os médicos com nomes que iniciam com as letras de "A" a "F" ordenado por nome
- 4) Buscar os médicos com campo CPF com valor nulo
- 5) Buscar as consultas que custaram entre 200 e 300.

# Bibliografia

- ◆ Elmasri, Ramez; Navathe, Shamkant B. **Sistemas de banco de dados.** 6ª ed. São Paulo: Addison Wesley, 2011
  - capítulo 4: SQL Básica