



## Introdução ao Modelo Relacional e a SQL-DML

TIAGO G MORAES

### Roteiro

- ❑ Modelo relacional
- ❑ SQL
- ❑ DML

### Modelo Relacional

- ❑ Concebido na década de 70 por Codd
  - Baseado na teoria dos conjuntos
- ❑ Utiliza tabelas relacionadas para agrupar dados de maneira a:
  - Evitar redundâncias no armazenamento
  - Possibilitar um desempenho nas operações básicas nos dados

col1	col2	col3

BANCO DE DADOS

3

### Modelo Relacional

- ❑ O nome “Relacional” deriva de “Relação” conceito da teoria dos conjuntos que expressa uma tabela.
- ❑ Uma relação ou tabela é composta por:
  - Linhas => tuplas
  - Colunas => atributos ou campos

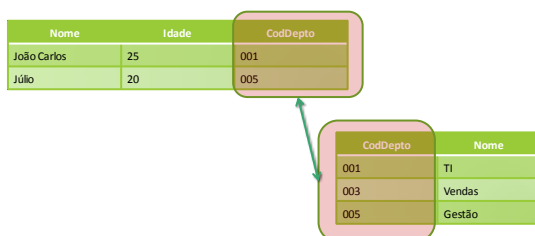
Nome	Idade	Sexo
João Carlos	25	Masculino
Júlio	20	Feminino

BANCO DE DADOS

4

### Modelo Relacional

- ❑ As informações entre duas tabelas se correlacionam por uma coluna

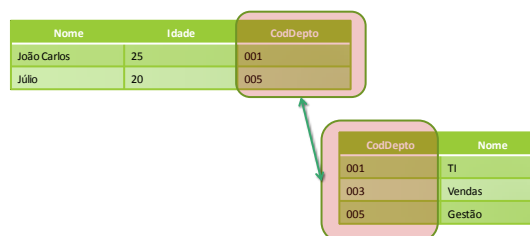


BANCO DE DADOS

5

### Modelo Relacional

- ❑ As informações entre duas tabelas se correlacionam por uma coluna



BANCO DE DADOS

6

## Modelo Relacional



- ❑ Bancos de dados Relacionais possuem uma linguagem para acesso aos dados
  - Definida e padronizada para os SGBD's relacionais
- ❑ SQL – Structured Query Language
  - Linguagem para acesso e definição de dados

BANCO DE DADOS

7

## SQL



- ❑ A SQL possui comandos que podem ser agrupados em “sublinguagens”:
  - DDL: Data Definition Language
    - Cria as estruturas necessárias
  - DML: Data Manipulation Language
    - Manipulação dados: inserir, alterar, deletar e consultar dados
  - DCL: Data Control Language
    - Definição de usuários e controle de permissões

BANCO DE DADOS

8

## DML



- ❑ Inserindo dados
  - Comando INSERT

```
INSERT INTO <nome_tabela> [(<col1>, <col2>, ...)]
VALUES (<val1>, <val2>, ...)
```

[] = opcionalidade  
 <> = preencher com algum valor  
 {} = escolha um dos itens

BANCO DE DADOS

9

## DML



- ❑ Inserindo dados
  - Comando INSERT

```
INSERT INTO <nome_tabela> [(<col1>, <col2>, ...)]
VALUES (<val1>, <val2>, ...)
```

- Exemplo

```
INSERT INTO empregado (nome, codigo)
VALUES ('Carlos', 005)
```

BANCO DE DADOS

10

## DML



- ❑ Excluindo dados
  - Comando DELETE

```
DELETE FROM <tabela> [WHERE <condição>]
```

- Condição: como em um if, pode ter:
  - Operadores Relacionais: <, >, <=, >=, =, ...
  - Operadores Lógicos: and, or, ...

BANCO DE DADOS

11

## DML



- ❑ Excluindo dados
  - Comando DELETE

```
DELETE FROM <tabela> [WHERE <condição>]
```

- Exemplo

```
DELETE FROM empregado WHERE codempregado=9
```

BANCO DE DADOS

12

## DML



### Alterando dados da tabela

#### Comando UPDATE

```
UPDATE <tabela> SET <col1>=<val1>, <col2>=<val2> ...
[WHERE <condição>]
```

#### Exemplo

```
UPDATE empregado SET nome='Jéssica', idade=20
WHERE cpf='012.321.654-96' AND id=25
```

BANCO DE DADOS

13

## DML



### Alterando dados da tabela

#### Comando UPDATE

```
UPDATE <tabela> SET <col1>=<val1>, <col2>=<val2> ...
[WHERE <condição>]
```

#### Exemplo

```
UPDATE empregado SET nome='Jéssica', idade=20
WHERE cpf='012.321.654-96' AND id=25
```

Para o tipo texto (String) em SQL usa-se 'aspas simples'

BANCO DE DADOS

14

## DML



### Consulta a dados da tabela

#### Comando SELECT

```
SELECT <col1>,<col2>,... FROM <tabela>
[WHERE <condição>]
```

#### Exemplo

```
SELECT nome, idade FROM empregado
WHERE salario >= 5000
```

BANCO DE DADOS

15

## DML



### Consulta a dados da tabela

#### Comando SELECT

```
SELECT <col1>,<col2>,... FROM <tabela>
[WHERE <condição>]
```

#### Exemplo

```
SELECT nome, idade FROM empregado
WHERE salario >= 5000
```

Para acesso a todas colunas usa-se o \*  
SELECT \* FROM ...

BANCO DE DADOS

16