

Universidade Presbiteriana Mackenzie

Banco de Dados – Aula 11

Linguagem SQL

INSERT, UPDATE e DELETE



Profa. Elisângela Botelho Gracias

Faculdade de Computação e Informática

Roteiro da Apresentação

- Introdução
- INSERT INTO
- UPDATE
- DELETE



Roteiro da Apresentação

- Introdução
- INSERT INTO
- UPDATE
- DELETE



Introdução

- Os comandos da linguagem SQL podem ser divididos em três classes
 - Linguagem de Definição de Dados (DDL)
 - Linguagem de Manipulação de Dados (DML)
 - Linguagem de Controle de Dados (DCL)

Introdução

- Linguagem de Manipulação de Dados (DML)
 - inclui comandos para inserir, remover, atualizar e consultar os dados armazenados nas tabelas

Introdução

- A Linguagem de Manipulação de Dados (DML) inclui os seguintes comandos em SQL
 - INSERT INTO: insere dados em uma tabela
 - UPDATE: atualiza dados de uma tabela
 - DELETE: elimina registros de uma tabela
 - SELECT: consulta dados de uma tabela

Roteiro da Apresentação

- Introdução
- **INSERT INTO**
- UPDATE
- DELETE



INSERT INTO

- INSERT INTO
 - possibilita a inclusão de dados em uma tabela

INSERT INTO

- Sintaxe do comando INSERT INTO

INSERT

INTO nome_tabela (nome_atributo1, ..., nome_atributoN)

VALUES

(valor_atributo1, ..., valor_atributoN);



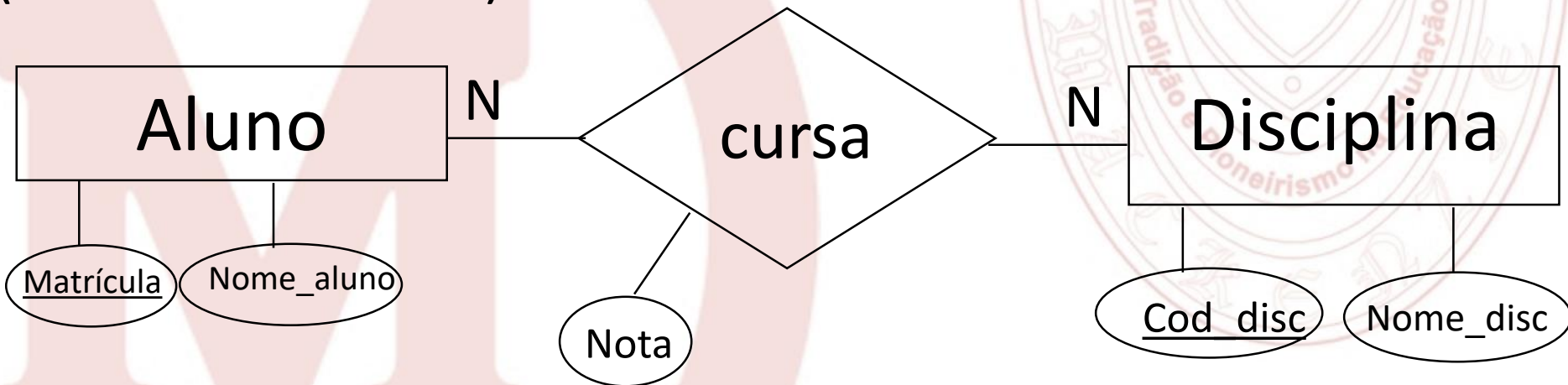
INSERT INTO

- Onde:
 - nome_tabela: indica em qual tabela os dados serão inseridos
 - nome_atributo: indica o nome do atributo que será adicionado valores
 - valor_atributo: valores que serão inseridos nos atributos especificados acima (os valores tem que seguir a ordem em que os atributos foram especificados)

INSERT INTO

Para inserir dados, vamos considerar o Banco de Dados da aula anterior:

Considere parte de um Modelo Entidade-Relacionamento (modelo conceitual) de um banco de dados:



INSERT INTO

O Modelo Entidade-Relacionamento anterior gera o seguinte **Modelo Relacional (modelo lógico)**, ou seja, as tabelas que serão criadas no banco de dados:

Aluno = {Matricula, Nome_aluno}

Disciplina = {Cod_disc, Nome_disc}

Aluno_Disciplina = {Matricula, Cod_disc, Nota}

- *Matricula é **chave estrangeira** que referencia Matricula da tabela Aluno*

- *Cod_disc é **chave estrangeira** que referencia Cod_disc da tabela Disciplina*

INSERT INTO

- Criação das tabelas do Banco de Dados mostrado anteriormente:

```
CREATE TABLE Aluno  
(Matricula INTEGER CHECK(Matricula >= 1),  
Nome_aluno VARCHAR(50) NOT NULL,  
PRIMARY KEY (Matricula)  
);  
CREATE TABLE Disciplina  
(Cod_disc INTEGER CHECK(Cod_disc >= 1),  
Nome_disc VARCHAR(25) NOT NULL UNIQUE,  
PRIMARY KEY (Cod_disc)  
);
```


INSERT INTO

- Criação das tabelas do Banco de Dados mostrado anteriormente:

```
CREATE TABLE Aluno_Disciplina  
(Matricula INTEGER,  
Cod_disc INTEGER,  
Nota INTEGER CHECK(Nota BETWEEN 0 AND 10),  
PRIMARY KEY (Matricula, Cod_disc),  
FOREIGN KEY (Matricula) REFERENCES Aluno(Matricula),  
FOREIGN KEY (Cod_disc) REFERENCES Disciplina(Cod_disc)  
);
```

INSERT INTO

- Para **inserir dados em uma tabela** é necessário conhecer o **script de criação da tabela** para ver as restrições existentes, como PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK e **NÃO VIOLÁ-LAS**.

INSERT INTO

- Exemplo: Insira um novo aluno com matrícula 1 e nome 'Joao'

INSERT

INTO Aluno (Matricula, Nome_aluno)

VALUES

(1,

'Joao');

INSERT INTO

- **Observações importantes**

- Valores numéricos são inseridos como se escrevem, enquanto as **cadeias de caracteres** tem sempre que ser delimitadas por **aspas simples**
- Como foram inseridos dados em todos os atributos da tabela, pode-se omitir os atributos, desde que seja obedecida a ordem dos atributos da tabela, como se segue:

```
INSERT
```

```
INTO Aluno
```

```
VALUES (1, 'Joao');
```

INSERT INTO

- Vamos inserir mais dois alunos, lembrando que a Matricula é chave primária

```
INSERT
```

```
INTO Aluno (Matricula, Nome_aluno)
```

```
VALUES (2, 'Maria');
```

```
INSERT
```

```
INTO Aluno (Matricula, Nome_aluno)
```

```
VALUES (3, 'Sergio');
```


INSERT INTO

Aluno

Matricula	Nome_aluno
1	Joao
2	Maria
3	Sergio

INSERT INTO

- **Exemplo:** Serão inseridas duas disciplinas. Observe que como o atributo Nome_disc foi definido como UNIQUE, não é possível ter **duas disciplinas com mesmo nome**

```
INSERT
```

```
INTO Disciplina (Cod_disc, Nome_disc)
```

```
VALUES (100, 'Banco de Dados I');
```

```
INSERT
```

```
INTO Disciplina (Cod_disc, Nome_disc)
```

```
VALUES (101, 'Banco de Dados II');
```

INSERT INTO

Disciplina

Cod_disc	Nome_disc
100	Banco de Dados I
101	Banco de Dados II

Obs: não seria possível inserir disciplina sem valor para Nome_disc, pois foi definido que este atributo é NOT NULL. Caso contrário, poderia ter uma disciplina com valor NULO para Nome_disc

INSERT INTO

- **Exemplo:** Serão inseridas linhas na tabela Aluno_Disciplina. Observe que:
 - só é possível inserir alunos que já existam na tabela Aluno e disciplinas que já existam na tabela Disciplina, pois na tabela Aluno_Disciplina tem 2 chaves estrangeiras
 - no atributo Nota só poderá ser inserido um valor entre 0 e 10

INSERT

INTO Aluno_Disciplina (Matricula, Cod_disc, Nota)

VALUES (1, 100, 9);

INSERT

INTO Aluno_Disciplina (Matricula, Cod_disc, Nota)

VALUES (1, 101, 8);

INSERT INTO

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	9
1	101	8

Aluno

Matricula	Nome_aluno
1	Joao
2	Maria
3	Sergio

Disciplina

Cod_disc	Nome_disc
100	Banco de Dados I
101	Banco de Dados II

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	9
1	101	8

INSERT INTO

- O INSERT poderá falhar em inúmeras situações, dentre as quais se destacam:
 - ao tentar inserir mais de uma vez a **mesma chave primária**
 - ao tentar inserir mais de uma vez o mesmo valor em um atributo **UNIQUE**
 - ao tentar inserir o valor **NULL** em um atributo **NOT NULL**
 - se o tipo do dado enviado na cláusula VALUES não estiver de acordo com o tipo de dado definido para aquele atributo
 - se algum dos atributos obrigatórios for ignorado
 - se o número de atributos for diferente do número de valores
 - se existir algum tipo de restrição no atributo que os dados não obedeçam

Roteiro da Apresentação

- Introdução
- INSERT INTO
- UPDATE
- DELETE



UPDATE

- UPDATE

– possibilita a atualização de dados em uma tabela

UPDATE

- Sintaxe do comando UPDATE

UPDATE nome_tabela

SET nome_atributo1 = valor1, ... nome_atributoN = valorN

[WHERE (condições)];

Obs: tudo que está entre [] é opcional, mas se for utilizar tire-o

UPDATE

- Onde:
 - nome_tabela: indica qual tabela será atualizada
 - nome_atributo: indica o nome do atributo a ser atualizado
 - WHERE: condições para atualização dos dados

UPDATE

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	9
1	101	8

UPDATE

- Exemplo: Aumentar em 1 ponto a nota dos alunos que tiraram uma nota maior ou igual a 7 ou menor ou igual a 9.

```
UPDATE Aluno_Disciplina
```

```
SET Nota = Nota + 1
```

```
WHERE (Nota >= 7) AND (Nota <= 9);
```

UPDATE

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	10
1	101	9

UPDATE

- Exemplo: O nome da disciplina 'Banco de Dados II' mudará para 'Topicos em Banco'

UPDATE Disciplina

SET Nome_disc = 'Topicos em Banco'

WHERE (Nome_disc = 'Banco de Dados II');

INSERT INTO

Disciplina

Cod_disc	Nome_disc
100	Banco de Dados I
101	Topicos em Banco

Roteiro da Apresentação

- Introdução
- INSERT INTO
- UPDATE
- DELETE



DELETE

- DELETE

– possibilita a remoção de dados em uma tabela



DELETE

- **DELETE x DROP TABLE**

- DELETE (comando de manipulação de dados):
possibilita a remoção de dados em uma tabela
- DROP TABLE (comando de definição de dados):
remove uma tabela existente no banco de dados
(toda sua estrutura) juntamente com todos os seus
dados

DELETE

- Sintaxe do comando DELETE

DELETE

FROM nome_tabela

[WHERE (condições)];

Obs: tudo que está entre [] é opcional, mas se for utilizar tire-o



DELETE

- Onde:
 - nome_tabela: indica a tabela que os dados serão removidos
 - WHERE: condições para remoção dos dados

UPDATE

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	10
1	101	9

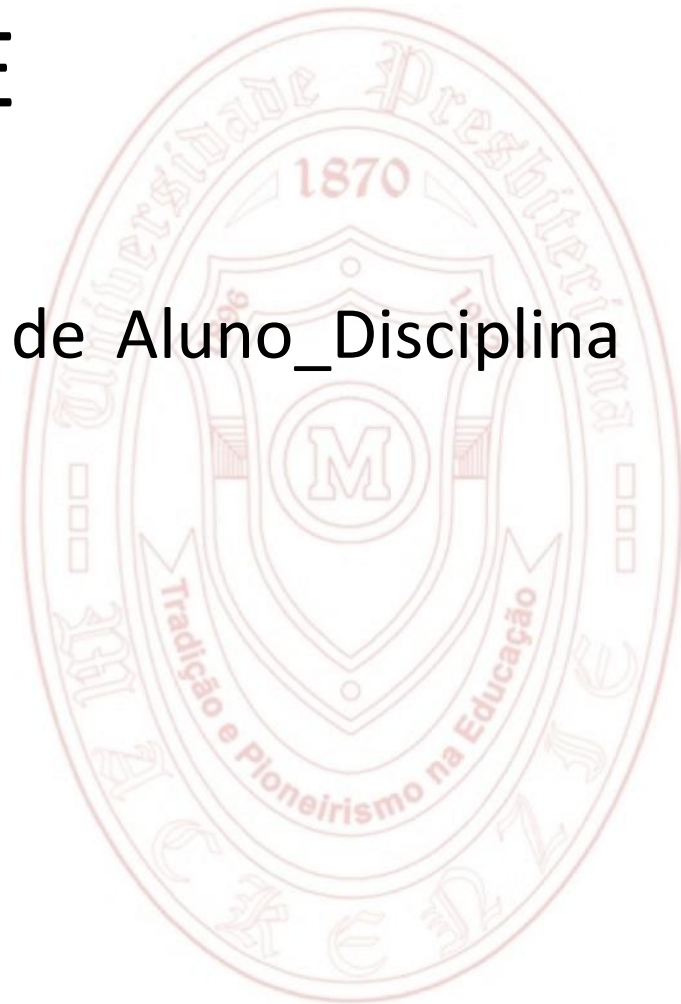
DELETE

- Exemplo: Remover todas as linhas de Aluno_Disciplina que tenham o Cod_disc = 101

DELETE

FROM Aluno_Disciplina

WHERE (Cod_disc = 101);



UPDATE

Aluno_Disciplina

Matricula	Cod_disc	Nota
1	100	10
1	101	9

DELETE

- Na tabela Disciplina, seria possível eliminar somente a disciplina de código 101. Isso porque a **tabela Aluno_Disciplina não referencia mais a disciplina de código 101**

```
DELETE
```

```
FROM Disciplina
```

```
WHERE (Cod_disc = 101);
```


DELETE

Se não colocar nenhuma condição para a remoção dos dados em uma tabela, todos os dados desta tabela serão removidos, desde que **NÃO VIOLE as restrições definidas!!!**

A tabela ainda existe, mas agora sem nenhum dado!!!

Obrigado

Elisângela Botelho Gracias
elisangela.botelho@mackenzie.br