



INSTITUTO FEDERAL

Goiano

Campus Morrinhos

BANCO DE DADOS

DATA MANIPULATION LANGUAGE (DML)

PROFESSOR MARCEL MELO

MARCEL.MELO@IFGOIANO.EDU.BR

Linguagem de definição de dados (DDL, do Inglês *Data Definition Language*) é uma linguagem de computador usada para a definição de estruturas de dados.

Permite especificar o esquema do Banco de Dados, através de um conjunto de definições de dados.

Comandos

- **CREATE** – utilizada para construir um novo banco de dados, tabela, índice ou consulta armazenada.
- **ALTER** - Alter - Modifica um objeto existente do banco de dados.
- **DROP** – remove um banco de dados, tabela, índice ou visão existente.

Linguagem de Manipulação de Dados (ou DML, *Data Manipulation Language*) é uma família de linguagens de computador utilizadas para a recuperação, inclusão, remoção e modificação de informações em bancos de dados.

Comandos:

- **INSERT** - Instrução utilizada para incluir uma nova tupla na relação.
- **UPDATE** - Esta operação modifica os valores de um ou mais atributos de uma tupla de uma certa relação.
- **DELETE** - Esta operação promove a remoção de uma ou mais tuplas

INSERT

Usado para adicionar uma nova tupla (linha) a uma relação (tabela).

A sentença **INSERT** é composta basicamente de duas cláusulas: **INSERT INTO** e **VALUES**.

- **INSERT INTO** – Define o nome da relação a ser inserido a informação.
- **VALUES** – Lista de valores que serão inseridos na tupla.

Os valores devem ser especificados na mesma ordem em que os atributos correspondentes foram especificados no comando CREATE TABLE.

A operação INSERT, pode violar as quatro restrições de integridade:

- **Domínio** - O valor informado não está de acordo com o tipo de dados definido para uma determinada coluna;
- **Chave** - O valor informado para a chave está de com os conceitos de chave primária;
- **Integridade de entidade** - Não é informado todos os valores obrigatórios e/ou valor de chave primária como NULL
- **Integridade referencial** - O valor informado para chave estrangeira não está de acordo com sua referência.

Quando qualquer uma dessas violações ocorre, a inserção da nova tupla será rejeitada.

```
CREATE TABLE Usuario(  
    id int NOT NULL AUTO_INCREMENT,  
    nome varchar(50) NOT NULL,  
    email varchar(100) NOT NULL,  
    telefone varchar(15) NULL,  
    senha varchar(10) NOT NULL,  
    PRIMARY KEY (id)  
)
```

INSERT - COMANDO

```
INSERT INTO <nome da tabela> (<lista de atributos>)  
VALUES (<lista de valores>);
```

```
INSERT INTO Usuário (nome, email, telefone, senha)  
VALUES ('Marcel', 'marcel.melo@ifgoiano.edu.br',  
        '99999-9999', '123456');
```

```
INSERT INTO Usuário (nome, email, senha) VALUES  
('Marcel', 'marcel.melo@ifgoiano.edu.br', '123456');
```


Caso seja informado dados de todas as colunas definidas no CREATE TABLE, utilizando a mesma ordem da criação, a lista de atributos pode ser omitida. **MAS somente se definir dados para TODAS as colunas!**

```
INSERT INTO Usuário VALUES (1, 'Marcel',  
'marcel.melo@ifgoiano.edu.br', '99999-9999',  
'123456') ;
```

Neste caso, mesmo que o valor ID seja definido como **auto_increment**, o comando será ignorado e deve ser informado o valor a ser utilizado na inserção. Nesse caso a responsabilidade de garantia que o ID não existe na tabela passa a ser do usuário.

UPDATE

O comando UPDATE é usado para modificar valores de atributos de uma ou mais tuplas selecionadas.

As tuplas que sofrerão as modificações são especificadas por meio de condições associadas a determinados atributos da relação. Para isso é utilizado a cláusula WHERE.

Violação de integridade

Como acontece na operação INSERT, a operação **UPDATE** também pode violar as quatro restrições de integridade: domínio, chave, integridade de entidade e de integridade referencial.

A sentença **UPDATE** é composta basicamente de três cláusulas: **UPDATE**, **SET** e **WHERE**.

- A cláusula **UPDATE** é responsável pela definição do nome da tabela alvo.
- A cláusula **SET** define qual atributo sofrerá alteração bem como seu novo valor. Pode ser uma lista de atributos.
- A cláusula **WHERE** define a condição para que a alteração ocorra.

```
UPDATE <nome da tabela> SET coluna1 = <novo valor>,  
coluna2 = <novo valor>, ... WHERE <condição 1>,  
<condição 2>, ...
```

COMANDO UPDATE - EXEMPLO

```
UPDATE Usuario SET telefone = '99393-8888' WHERE id =  
1
```

```
UPDATE Usuario SET email = 'marcel.msmelo@gmail.com'  
WHERE nome = 'Marcel'
```

COMANDO DELETE

O comando **DELETE** remove tuplas (linhas) de uma relação de acordo as condições associadas a determinados atributos da relação. Para isso é utilizado a cláusula **WHERE**.

As tuplas são explicitamente excluídas de apenas uma tabela por vez. Porém, a exclusão pode propagar para as tuplas em outras relações.

Depende da condição associada ao comando DELETE, zero, uma ou várias tuplas podem ser excluídas por um único comando.

Restrição de Integridade

A operação **DELETE**, ela pode violar a restrição de integridade referencial. Quando isso ocorre, ou a operação de remoção é rejeitada ou propagada por meio da remoção das tuplas que referenciam a tupla inicialmente removida.



COMANDO DELETE

```
DELETE FROM <nome da tabela> WHERE <condição 1>,  
<condição 2>
```

```
DELETE FROM Usuario WHERE id = 1;
```

```
DELETE FROM Usuario WHERE id < 10;
```

DESABILITAR SAFE_UPDATE DO MYSQL

SET SQL_SAFE_UPDATES = 0;

AND Raio!

EXEMPLO

