

MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS

Pedro Henrique Chagas Freitas

sagah⁺



Linguagem de manipulação de dados: DML (*Data Manipulation Language*)

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar a linguagem DML.
- Exemplificar a linguagem DML.
- Implementar a linguagem DML.

Introdução

A DML (*Data Manipulation Language*) é uma linguagem de manipulação utilizada na composição do SQL (*Structured Query Language*), que é uma linguagem estruturada e difundida para bancos de dados relacionais e inspirada em álgebra relacional.

Neste capítulo, você vai estudar a modelagem e o desenvolvimento de banco de dados a partir da perspectiva da linguagem DML (*Data Manipulation Language*), observando as estruturas que compõem a DML, bem como exemplos de sua utilização e da sua implementação na inserção, seleção, atualização e exclusão de dados. Logo, a DML será apresentada de forma holística, com uma abordagem de todas as características que vinculam essa linguagem, bem como sua implementação na modelagem e no desenvolvimento de bancos de dados.

Conceituando a linguagem DML

A linguagem DML (*Data Manipulation Language*) nasce para respaldar o contexto de manipulação de dados em bancos de dados, principalmente em sistemas de informação que fazem consulta em bancos de dados relacionais por meio de SGBDs (Sistemas de Gerenciamento de Bancos de

Dados). Quando os dados inseridos nos bancos de dados começaram a ser consultados por diversos tipos de sistemas de informação em concorrência de recursos, o dado logo se tornou um recurso manipulável e uma linguagem em comum precisava ser utilizada, a fim de que se adotasse um padrão para a manipulação dos dados — o que, por fim, fez com que a DML fosse utilizada para essa manipulação.

O objetivo da manipulação dos dados pode responder a vários contextos diferentes, mas o fato é que cada contexto vai tratar o dado como um recurso variável; assim, a consistência dos dados e, logicamente, dos bancos de dados poderia ficar comprometida caso os dados fossem manipulados de forma despadronizada e irregular.

Por sua vez, outras linguagens de manipulação foram surgindo como derivações das principais funções de manipulação de dados, isto é:

- Inserir o dado (INSERT);
- Selecionar o dado (SELECT);
- Atualizar o dado (UPDATE);
- Deletar o dado (DELETE).

É importante observar que a derivação desse conhecimento não muda as ações de manipulação dos dados. Assim, por meio da linguagem SQL, o DML se difundiu, sendo, hoje, o principal instrumento de relacionamento entre os sistemas de informação, que utilizam os dados a fim de gerar informações, e os sistemas gerenciadores de bancos de dados (SGBDs), que efetuam a manipulação dos dados nos bancos de dados, nos quais, de fato, os dados são armazenados.

A linguagem DML pode utilizar recursos de forma procedural, com a qual existe a especificação de como os dados serão obtidos no banco de dados, ou não procedural (declarativa), com a qual não existe necessidade de especificação do caminho de acesso ao dado. A linguagem DML também é utilizada no contexto de sistemas de informação como uma linguagem verbal, em que um sistema de informação externo ao relacionamento entre SGBD e banco de dados faz uma averbação da instrução que necessita, isto é, utiliza um verbo para demonstrar a ação desejada. Os verbos DML, por assim dizer, seriam: SELECT, INSERT, UPDATE e DELETE (Figura 1).

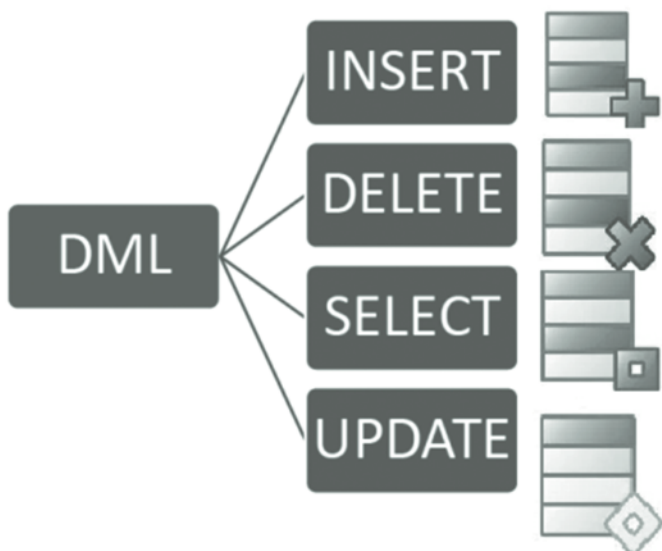


Figura 1. A linguagem DML.

Na modelagem e no desenvolvimento de bancos de dados, é necessário compreender o dado nas suas diferentes perspectivas. Pensando nisso Elmasri e Navathe (2010) definem o banco de dados como uma coleção de dados relacionados. A partir dessa perspectiva, podemos concluir que, por meio da manipulação dos dados, novos dados podem surgir ou deixar de existir a partir do relacionamento entre as instruções DML do SGBDs. Assim, é muito importante saber que as instruções DML manipulam diretamente o dado; dessa forma, quem manipula esses dados deve ter plena ciência da natureza e do relacionamento dos dados.



Fique atento

O que é um *schema* de banco de dados? *Schema* é um conjunto lógico agrupado de estruturas em banco de dados, representado por uma coleção de objetos comuns, como tabelas, índices, campos, etc.

Os *schemas* podem ser segmentados ou não, dependendo de como o banco se encontra normalizado, tendo em vista que o *schema* vai representar uma composição de objetos no banco de dados.

Assim, a partir da coleção de dados nos *schemas* de um banco de dados, novos dados podem ser inseridos (INSERT) nos campos relacionais de uma tabela. Essa inserção ocorre por meio do comando “INSERT INTO tabela (...) values (...)”, onde, no primeiro parênteses, temos as colunas e, no segundo parênteses, temos os valores.

```
INSERT INTO CAMPEONATO (Seleção, Atacante, Meio_Campo,
Lateral, Zagueiro) VALUES ('Brasil', 'Ronaldo', 'Kaká',
'Roberto Carlos', 'Lúcio').
```

Observe, então, que temos colunas que vão compor a tabela do ‘CAMPEONATO’ no primeiro campo (essas colunas serão a posição de cada jogador). Depois, teremos os valores que vão compor a tabela, que são os nomes dos jogadores que ocuparão as respectivas posições.

Quadro 1. Composição da tabela com os nomes dos jogadores

Seleção	Atacante	Meio-campo	Lateral	Zagueiro
Brasil	Ronaldo	Kaká	Roberto Carlos	Lúcio

Se continuarmos usando a instrução INSERT, iremos popular o banco de dados — neste caso, iríamos popular a tabela CAMPEONATO.

Dentro desse contexto, após a tabela estar populada, poderíamos utilizar a instrução DML ‘UPDATE’ para realizar a atualização dos dados inseridos na tabela. **Por lógica, então, devemos, primeiramente, inserir dados para que estes possam ser atualizados.** A instrução ‘UPDATE’ é responsável pelo estado atual do dado, isto é, qual dado temos em determinado espaço de tempo pré-definido. A instrução DML é importante porque ela garante a modificação consistente dos dados por meio de uma linha temporal; isto significa que, em determinado instante do tempo, é possível ter a rastreabilidade do dado e sua respectiva atualização. Assim, se em determinado espaço de tempo for necessário mudar o dado, ou seja, atualizá-lo, a instrução que deverá ser utilizada é UPDATE.

```
INSERT INTO CAMPEONATO (Seleção, Atacante, Meio_campo, Lateral,
Zagueiro) VALUES ('Brasil', 'Neymar', 'Kaká', 'Roberto Carlos',
'Lúcio');
UPDATE CAMPEONATO SET Atacante = 'Neymar' WHERE Seleção = 'Brasil';
```


Quadro 2. Atualização da tabela com a instrução UPDATE

Seleção	Atacante	Meio-campo	Lateral	Zagueiro
Brasil	Neymar	Kaká	Roberto Carlos	Lúcio

Caso tenhamos a inserção de várias seleções como, por exemplo, em um banco de dados de convocações da Copa do Mundo, faríamos diversos INSERTs para incluir as seleções que participariam da disputa:

```
INSERT INTO CAMPEONATO(Seleção, Atacante, Meio_campo,  
Lateral, Zagueiro) VALUES ('Brasil', 'Neymar', 'Kaká',  
'Roberto Carlos', 'Lúcio');  
  
INSERT INTO CAMPEONATO(Seleção, Atacante, Meio_campo,  
Lateral, Zagueiro) VALUES ('Argentina', 'Messi', 'Di Maria',  
'Gabriel Mercado', 'Mascherano');  
  
INSERT INTO CAMPEONATO(Seleção, Atacante, Meio_campo,  
Lateral, Zagueiro) VALUES ('Espanha', 'Juan', 'Xavi',  
'Iniesta', 'Lavi');  
  
INSERT INTO CAMPEONATO(Seleção, Atacante, Meio_campo,  
Lateral, Zagueiro) VALUES ('Portugal', 'Cristiano Ronaldo',  
'Émerson', 'João', 'Davi');
```



Fique atento

Existem situações nas quais o comando UPDATE não poderá ser usado, de modo que o dado precisa ser reinserido (INSERT) e, posteriormente, o dado desatualizado deverá ser deletado (DELETE).

Depois de inserir (INSERT) e atualizar (UPDATE) os dados, poderemos selecioná-los, isto é, realizar a consulta a esses dados, que, de forma geral, é a principal função dos sistemas de informação. Ou seja, um sistema de informação, ao realizar o SELECT, consulta o dado na base de dados.

Quadro 3. Base de dados CAMPEONATO montada com os nomes dos jogadores

Seleção	Atacante	Meio-campo	Lateral	Zagueiro
Brasil	Neymar	Kaká	Roberto Carlos	Lúcio
Argentina	Messi	Di Maria	Gabriel Mercado	Mascherano
Espanha	Juan	Xavi	Iniesta	Lavi
Portugal	Cristiano Ronaldo	Émerson	João	Pepe

```
SELECT Lateral, Zagueiro FROM CAMPEONATO where Seleção = 'Argentina'
```

A tabela apresentará, conforme o comando anterior, os seguintes dados:

Quadro 4. Tabela CAMPEONATO com a instrução SELECT de alguns dados

Seleção	Lateral	Zagueiro
Argentina	Gabriel Mercado	Mascherano

A seleção consiste, então, na consulta aos dados armazenados em uma determinada tabela, fazendo referência a um dado parâmetro, que pode ser a chave primária, por exemplo.

Podemos ainda unir dois ou mais comandos SQL para manipular o banco de dados. Podemos fazer uso dos comandos INSERT e SELECT em conjunto, por exemplo, para copiar todos os dados de uma tabela para outra, assim:

```
INSERT INTO tabela_Y (nome_colunal, nome_coluna_2) SELECT  
nome_colunal, nome_coluna_2 FROM tabela_X WHERE (condição);
```

O comando INSERT é portanto usado para inserir os dados na tabela_Y e o comando SELECT para selecionar os dados que serão copiados da tabela_X.

Por fim, teremos o comando DELETE, que tem o papel de excluir um dado, por exemplo:

```
DELETE FROM CAMPEONATO WHERE Seleção = 'Portugal';
```

Neste caso, excluimos Portugal da nossa tabela CAMPEONATO:

Quadro 5. Tabela CAMPEONATO com a instrução DELETE aplicada no campo \

Seleção	Atacante	Meio-campo	Lateral	Zagueiro
Brasil	Neymar	Kaká	Roberto Carlos	Lúcio
Argentina	Messi	Di Maria	Gabriel Mercado	Mascherano
Espanha	Juan	Xavi	Iniesta	Lavi

Agora, se precisássemos excluir todos os dados da tabela CAMPEONATO usá-riamos:

```
DELETE from CAMPEONATO;
```



Fique atento

Algumas pessoas confundem a instrução DDL: DROP com a instrução DML: DELETE. A instrução DDL: DROP é responsável por eliminar “Tabelas”, enquanto a instrução DML: DELETE é utilizada para deletar “dados”. O que muda, então, é a granularidade da exclusão.

Exemplificando a linguagem DDL

Temos, então, que a linguagem DML compõe o SQL e é utilizada na modelagem e no desenvolvimento de banco de dados para a manipulação de dados presentes em colunas, tabelas, linhas e índices.



Exemplo

```
INSERT INTO CASA (Janela, Porta, Parede) VALUES (1.3, 2.0, 3.0);  
UPDATE CASA SET Parede = 4.0 WHERE Janela = 1.3;  
SELECT Porta FROM CASA WHERE Parede = 4.0;  
DELETE FROM CASA WHERE Janela = 1.3;
```



Saiba mais

Instruções DML podem vir acompanhadas de mais de uma transação e de variações a depender dos tipos de tecnologia que estão baseando o SQL. Por exemplo:

T-SQL (Microsoft)

PL/SQL (Oracle)

Logo, temos, no caso do T-SQL, uma variação do SQL de propriedade da Microsoft, o que, naturalmente, acarretará em variações no conjunto de instruções DML. Em suma, as variações não são difíceis de assimilar, a essência, isto é, INSERT, UPDATE, SELECT e DELETE, permanece.

Implementação da DDL

Temos, agora, a tabela HOSPITAL. Vamos popular essa tabela com alguns dados:

```
INSERT INTO HOSPITAL (CodMEDICO, nomeMedico,  
especialidadeMedico) VALUES (568547, 'Henrique',  
'Cirurgião');
```

Temos, então:

Quadro 6. Tabela HOSPITAL: preenchimento dos campos

HOSPITAL		
codMEDICO	nomeMedico	especialidadeMedico
5658547	Henrique	Cirurgião

Foram contratados novos médicos:

```
INSERT INTO HOSPITAL(codMEDICO, nomeMedico,  
especialidadeMedico) VALUES (986752, 'Marcos', 'Clínico');  
  
INSERT INTO HOSPITAL(codMEDICO, nomeMedico,  
especialidadeMedico) VALUES (843221, 'Gabriela',  
'Ginecologista');  
  
INSERT INTO HOSPITAL(codMEDICO, nomeMedico,  
especialidadeMedico) VALUES (843957, 'Felipe',  
'Ortopedista');
```

Temos, então:

Quadro 7. Inserção de novos nomes nos campos da base de dados

HOSPITAL		
codMEDICO	nomeMedico	especialidadeMedico
5658547	Henrique	Cirurgião
986752	Marcos	Clínico
842321	Gabriela	Ginecologista
843957	Felipe	Ortopedista

Vamos supor que Marcos terminou sua residência em oftalmologia, necessitando, assim, que sua especialidade seja atualizada.

```
UPDATE HOSPITAL SET especialidadeMedico = 'Oftalmologia' WHERE
codMEDICO = 986752;
```

Temos, então:

Quadro 8. Atualização de campo com a instrução UPDATE

HOSPITAL		
codMEDICO	nomeMedico	especialidadeMedico
5658547	Henrique	Cirurgião
986752	Marcos	Oftalmologista
842321	Gabriela	Ginecologista
843957	Felipe	Ortopedista

Porém, Felipe recebeu um convite para fazer uma especialização fora do país em traumatologia óssea, de modo que precisará ausentar-se de suas atividades no hospital. Vamos excluir Felipe com a seguinte instrução:

```
DELETE FROM HOSPITAL WHERE codMEDICO = 843957;
```

Logo, temos:

Quadro 9. Tabela atualizada com a exclusão de um dos médicos.

HOSPITAL		
codMEDICO	nomeMedico	especialidadeMedico
5658547	Henrique	Cirurgião
986752	Marcos	Oftalmologista
842321	Gabriela	Ginecologista

Agora que já inserimos dados (INSERT), atualizamos (UPDATE) e apagamos (DELETE), vamos supor que desejássemos realizar alguma consulta simples à base de dados. Utilizaríamos, então, a instrução SELECT.

“Quero o nome do médico e sua especialidade pelo código médico que este possui”:

Quadro 10. Resultado da busca usando a instrução SELECT

HOSPITAL	
nomeMedico	especialidadeMedico
Henrique	Cirurgião

```
SELECT nomeMedico, especialidadeMedico FROM HOSPITAL WHERE codMEDICO  
= 568547;
```

Temos, então:

Lembre-se: o DML tem sido uma das principais linguagens na modelagem e no desenvolvimento de bancos de dados. Alguns DBAs, inclusive, atribuem a própria evolução dos SGBDs à utilização do DML em diferentes cenários, inclusive na intercomunicação e no tráfego de dados entre sistemas de informação e bancos de dados.

As instruções DML vão, então, ajudar-nos a manipular os dados por meio do SGBDs com as instruções: INSERT, UPDATE, SELECT e DELETE.



Referência

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. São Paulo: Pearson, 2010.

Leituras recomendadas

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. *Sistema de banco de dados*. 6. ed. Rio de Janeiro: Campus, 2012.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2008. (Série Livros Didáticos Informática UFRGS, v. 4).

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de bancos de dados*. 3. ed. São Paulo: McGraw-Hill, 2008.

SETZER, V. W. *Banco de dados: conceitos, modelos, gerenciadores, projeto lógico, projeto físico*. 3. ed. São Paulo: Blücher, 2002.



Fique atento

Os *links* para *sites* da *web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Conteúdo:

sagah⁺