Sistemas de Banco de Dados

Fundamentos em Bancos de Dados Relacionais

Wladmir Cardoso Brandão
www.wladmirbrandao.com

SQL: MANIPULAÇÃO DE DADOS

Structured Query Language

- Linguagem de consulta estruturada
- Linguagem declarativa de alto nível
- Usuário especifica o que deseja, deixando decisões sobre como executar a consulta para o SGBD
- Contém instruções para definição e manipulação de dados
- Padrão em SGBDs relacionais comerciais
- Mantém equivalência com o modelo relacional
  - Relação → Tabela
  - Tupla → Linha
  - Atributo → Coluna

SQL: Manipulação de Dados

Existem diferentes instruções (comandos) para manipulação de dados

- **INSERT** → inserir linhas em tabelas
- **DELETE** → remover linhas de tabelas
- **UPDATE** → atualizar valores de colunas em linhas de tabelas
- **SELECT** → recuperar dados em tabelas

Instrução INSERT

Acrescenta uma linha em uma tabela

- Necessário especificar o nome da tabela e uma lista de valores para a linha
- Valores devem ser listados na mesma ordem em que as colunas correspondentes foram definidas na tabela

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento

```
INSERT INTO PROFESSOR VALUES
('12345678900', 'Ricardo Marini', 'M', 3000.00, 1);
```

Instrução INSERT

É possível especificar nomes de colunas correspondentes a valores fornecidos

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento

```
INSERT INTO PROFESSOR (CPF, Sexo, Nome, Departamento)
VALUES ('12345678900', 'M', 'Ricardo Marini', 1);
```

Coluna não especificada tem seu valor definido como DEFAULT ou NULL, sendo que valores e colunas devem ser listadas na mesma ordem

Instrução INSERT

Se alguma restrição for violada a operação é rejeitada

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
INSERT INTO PROFESSOR (CPF, Nome, Sexo, Departamento)
VALUES ('68345618900', 'Amanda Ramirez', 'F', 4);
```

Caso não exista tupla na tabela DEPARTAMENTO com chave primária *Numero = 4* para manter integridade referencial com a coluna *Departamento* da tabela PROFESSOR a operação será rejeitada

Instrução INSERT

Se alguma restrição for violada a operação é rejeitada

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
INSERT INTO PROFESSOR (Nome, Sexo, Departamento)
VALUES ('Amanda Ramirez', 'F', 1);
```

Valor da chave primária *CPF* não foi fornecido, o que viola a restrição de chave, logo a operação será rejeitada

Instrução INSERT

É possível inserir múltiplas linhas na tabela usando a instrução INSERT combinada com a instrução SELECT

```
INSERT INTO PROFESSOR (CPF, Nome, Sexo, Departamento)
SELECT CPF, Nome, Sexo, 1
FROM ALUNO;
```

Nesse caso, todas as linhas da tabela ALUNO serão inseridas na tabela PROFESSOR, sendo que para todas as linhas inseridas a coluna *Departamento* terá valor *1*

Instrução DELETE

Remove linhas de uma tabela

- Linhas são excluídas de apenas uma tabela
  - Exceção → exclusão pode se propagar para linhas em outras tabelas, de acordo com restrições de integridade referencial
- Condição (cláusula WHERE) inexistente especifica que todas as linhas na tabela serão excluídas
  - Tabela permanece no BD como uma tabela vazia

Instrução DELETE

Exemplo:

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
DELETE FROM PROFESSOR
WHERE Salario < 1000,00;
```

Instrução não removerá nenhuma linha da tabela

Instrução DELETE

Exemplo:

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
DELETE FROM PROFESSOR
WHERE Sexo = 'M';
```

Instrução removerá duas linha da tabela

Instrução DELETE

Exemplo:

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
DELETE FROM PROFESSOR;
```

Instrução removerá todas as linha da tabela

Instrução UPDATE

Modifica valores em colunas de uma ou mais linhas

- Cada instrução afeta apenas uma tabela
  - Exceção → atualização de uma chave primária pode ser propagada para os valores de chave estrangeira das linhas em outras tabelas de acordo com restrições de integridade referencial
- Cláusula SET especifica colunas a serem modificadas e seus novos valores

Instrução UPDATE

Exemplos:

```
UPDATE  PROFESSOR
SET     Salario = 2500,00, Departamento = 2
WHERE   CPF = '12345678900';
```

Altera o salário e o número do departamento do professor de determinado CPF

```
UPDATE  PROFESSOR
SET     Salario = Salario * 1.1;
```

Aumenta em 10% o salário de todos os professores

Instrução SELECT

Recupera linhas em múltiplas tabelas

- MAPEAMENTO → forma básica da instrução SELECT, também chamado de BLOCO SELECT-FROM-WHERE

  SELECT <lista de colunas>
  FROM <lista de tabelas>
  WHERE <condição>;

- **<lista de colunas>** → lista de nomes de colunas que valores devem ser recuperados pela consulta
- **<lista de tabelas>** → lista dos nomes de tabelas necessárias para processar a consulta
- **<condição>** → expressão condicional que identifica linhas que devem ser recuperadas pela consulta

Instrução SELECT

Exemplo → recuperar o nome e o salário de todos os professores do sexo masculino do departamento de número 1

PROFESSOR

CPF	Nome	Sexo	Salario	Departamento
12345678900	Roberto Machado	M	1200.00	1
12345678901	Manuela Costa	F	2700.00	3
21345678900	Carlos A. Martins	M	3200.00	1
32145678900	Ana Maria Freitas	F	7500.00	2

```
SELECT  Nome, Salario
  FROM  PROFESSOR
 WHERE  Departamento = 1
   AND  Sexo = 'M';
```

Instrução SELECT

Exemplo → recuperar o CPF e o nome dos professores do sexo masculino que também são alunos

```
SELECT  A.CPF, A.Nome
  FROM  PROFESSOR A, ALUNO B
 WHERE  A.CPF = B.CPF
   AND  A.Sexo = 'M';
```

Resultado:

CPF	Nome
12345678900	Roberto Machado
21345678900	Carlos A. Martins

Instrução SELECT

Exemplo → recuperar o nome do departamento e do professor para todos os professores que são alunos e que trabalham no departamento de nome *Pesquisa*

```
SELECT  A.Nome AS Departamento, B.Nome AS Professor
  FROM  DEPARTAMENTO A, PROFESSOR B, ALUNO C
 WHERE  A.Numero = B.Departamento
   AND  B.CPF = C.CPF
   AND  A.Nome = 'Pesquisa';
```

Resultado:

Departamento	Professor
Pesquisa	Roberto Machado
Pesquisa	Carlos A. Martins

Instrução SELECT

Junções podem ser especificadas tanto na cláusula **WHERE** quanto na cláusula **FROM** com o uso do operador **JOIN**

```
SELECT  A.CPF, A.Nome
  FROM  PROFESSOR A, ALUNO B
 WHERE  A.CPF = B.CPF
   AND  A.Sexo = 'M';

SELECT  A.CPF, A.Nome
  FROM  PROFESSOR A JOIN ALUNO B ON A.CPF = B.CPF
 WHERE  A.Sexo = 'M';
```

Variações do operador de junção podem ser especificados, como **INNER JOIN**, **LEFT OUTER JOIN** e **FULL JOIN**

Ambiguidade

Mesmo nome pode ser usado em mais de uma coluna, desde que as colunas pertençam a tabelas diferentes e estejam devidamente prefixadas para evitar ambiguidade

```
SELECT  PROFESSOR.Nome, ALUNO.Nome
  FROM  PROFESSOR, ALUNO
 WHERE  PROFESSOR.CPF = ALUNO.CPF
   AND  PROFESSOR.Sexo = 'M';

SELECT  A.Nome, B.Nome
  FROM  PROFESSOR A, ALUNO B
 WHERE  A.CPF = B.CPF
   AND  A.Sexo = 'M';
```

Ausência de Cláusula WHERE

Inexistência de condições para seleção e junção de linhas traz impactos diferentes no resultado das consultas

- Tabela Única → todas as linhas da única tabela especificada na cláusula FROM são retornadas

```
SELECT  CPF
  FROM  PROFESSOR;
```

- Múltiplas Tabelas → todas as combinações possíveis entre linhas das tabelas especificadas na cláusula FROM são retornadas, equivalendo à operação **Produto Cartesiano** da álgebra relacional

```
SELECT  A.CPF
  FROM  PROFESSOR A, DEPARTAMENTO B;
```

Duplicatas

Uma tabela constitui um multiconjunto e linhas duplicadas podem aparecer no resultado de uma consulta

- DISTINCT → elimina linhas duplicadas no resultado

```
SELECT  Departamento
  FROM  PROFESSOR
 WHERE  Salario < 5000.00;
```

Departamento
1
3
1

```
SELECT  DISTINCT Departamento
  FROM  PROFESSOR
 WHERE  Salario < 5000.00;
```

Departamento
1
3

Operadores Especiais

Asterisco (*) → recupera todas as colunas das linhas selecionadas sem a necessidade de listar seus nomes explicitamente

```
SELECT  *
  FROM  PROFESSOR
 WHERE  Departamento = 1
```

Nesse caso, recupera todas as colunas de professores que trabalham no departamento de número 1

Operadores Especiais

LIKE → comparação sobre subcadeias de caracteres

- Subcadeias são especificadas usando dois caracteres especiais
  - % substitui zero ou mais caracteres
  - _ substitui um único caracter

```
SELECT  CPF, Nome
  FROM  PROFESSOR
 WHERE  Endereco LIKE '%Belo Horizonte%';
```

Recupera o CPF e nome de todos os professores em que seu endereço contenha a subcadeia de caracteres *Belo Horizonte*

Operadores Especiais

BETWEEN → comparação com intervalos

- Valores para colunas comparadas devem estar entre um intervalo de valores

```
SELECT  *
  FROM  PROFESSOR
 WHERE  Salario BETWEEN 2000,00 AND 5000,00;
```

Recupera todas as colunas de professores com salários entre 2 e 5 mil

Ordenação de Resultados

ORDER BY → ordena linhas do resultado de uma consulta

- ASC → operador **padrão** para ordenação crescente
- DESC → operador para ordenação decrescente

```
SELECT      A.Nome, B.Nome
  FROM      DEPARTAMENTO A, PROFESSOR B
 WHERE      A.Numero = B.Departamento
ORDER BY    B.Nome, A.nome DESC;
```

Recupera o nome do departamento e do professor para todos os professores que trabalham em um departamento, ordenando o resultado de maneira crescente pelo nome do professor e decrescente pelo nome do departamento

Valores NULL

IS NULL (IS NOT NULL) → verifica se valor de coluna é NULL

- NULL tem semântica imprecisa
  - Valor desconhecido?
  - Valor indisponível?
  - Valor não aplicável?

```
SELECT      CPF, Nome
  FROM      PROFESSOR
 WHERE      Departamento IS NULL;
```

Recupera o CPF e o nome dos professores que não trabalham em algum departamento

Consulta Aninhada

Bloco SELECT completo na cláusula WHERE de outra consulta, denominada **consulta externa**

- IN (NOT IN) → verifica se um conjunto de valores pertence a um multiconjunto de valores

```
SELECT  A.Nome, A.Salario
  FROM  PROFESSOR A
 WHERE  (A.CPF, A.Nome) IN
        (SELECT B.CPF, B.Nome
           FROM ALUNO B
          WHERE A.Sexo = B.Sexo);
```

Recupera nome e salário dos professores que possuem mesmo CPF e nome de algum aluno, desde que tenham o mesmo sexo

Consulta Aninhada

Bloco SELECT completo na cláusula WHERE de outra consulta, denominada **consulta externa**

- EXISTS (NOT EXISTS) → verifica se o resultado da consulta interna é conjunto vazio

```
SELECT  A.CPF, A.Nome
  FROM  PROFESSOR A
 WHERE  NOT EXISTS
        (SELECT  *
           FROM  DEPARTAMENTO B
          WHERE  A.Departamento = B.Numero
            AND  B.Nome = 'Pesquisa');
```

Retorna CPF e nome dos professores que não trabalham no departamento de nome *Pesquisa*

Agregação

GROUP BY → agrupa múltiplas linhas em uma utilizando **função de agregação**

- **COUNT** → conta o número de linhas agrupadas
- **SUM** → soma o valor na coluna de linhas agrupadas
- **MAX** → retorna o valor máximo na coluna de linhas agrupadas
- **MIN** → retorna o valor mínimo na coluna de linhas agrupadas
- **AVG** → retorna a média dos valores na coluna de linhas agrupadas

Funções de agregação não têm efeito em linhas com colunas participantes da função com valor NULL

Agregação

Exemplos:

```
SELECT  COUNT(*), SUM(Salario)
  FROM  PROFESSOR
 WHERE  Sexo = 'F';
```

Retorna o número de professores do sexo feminino e o salário pago a elas

```
SELECT      Departamento, COUNT(*), SUM(Salario),
            MAX(Salario), MIN(Salario), AVG(Salario)
  FROM      PROFESSOR
GROUP BY    Departamento;
```

Para cada departamento retorna seu número, a quantidade de professores, a soma de salários, o salário máximo e mínimo e a média salarial

Agregação

HAVING → remove linhas do resultado agregado de acordo com condição imposta sobre as funções de agregação

```
SELECT      Departamento, COUNT(*), SUM(Salario)
  FROM      PROFESSOR
GROUP BY    Departamento
HAVING      AVG(Salario) > 8000.00;
```

Para cada departamento retorna seu número, a quantidade de professores e a soma de salários, desde que a média salarial do departamento seja maior que 8000,00

Resumo de Instruções

```
INSERT INTO <tabela>
[(<lista de atributos>)]
VALUES (<lista de valores>);

UPDATE <tabela>
SET <lista de atribuicoes>
[WHERE <condicao>];

DELETE FROM <tabela>
[WHERE <condicao>];

SELECT
FROM
[WHERE <condicao>]
[GROUP BY <atributos de agrupamento>]
[HAVING <condicao de grupo>]
[ORDER BY <lista de atributos>];
```

Referências Bibliográficas

[1] Elmasri, Ramez; Navathe, Sham. *Fundamentals of Database Systems*. 7ed. Pearson, 2016.

[2] Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. *Database System Concepts*. 6ed. McGraw-Hill, 2011.

[3] Date, Christopher J. *An Introduction to Database Systems*. 8ed. Pearson, 2004.