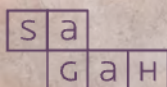


CONSULTAS EM BANCO DE DADOS

Marcel Santos Silva



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Junção interna em SQL

Objetivos de aprendizagem

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

- Definir a operação de junção interna em álgebra relacional.
- Comparar as cláusulas `NATURAL JOIN`, `JOIN . . . USING` e `JOIN . . . ON` em SQL.
- Reconhecer a implementação da operação de junção interna em SQL.

Introdução

Um modelo lógico descreve a estrutura de um banco de dados de acordo com o nível de abstração do usuário do sistema gerenciador de banco de dados (SGBD) (HEUSER, 2008). Nesse contexto de banco de dados relacionais, surgem as consultas estruturadas, que possibilitam um melhor aproveitamento das informações armazenadas no processo de tomada de decisão. Uma das características que contribuem para essas consultas estruturadas é a relação entre as diversas tabelas de dados, implementada por meio da junção de operações internas.

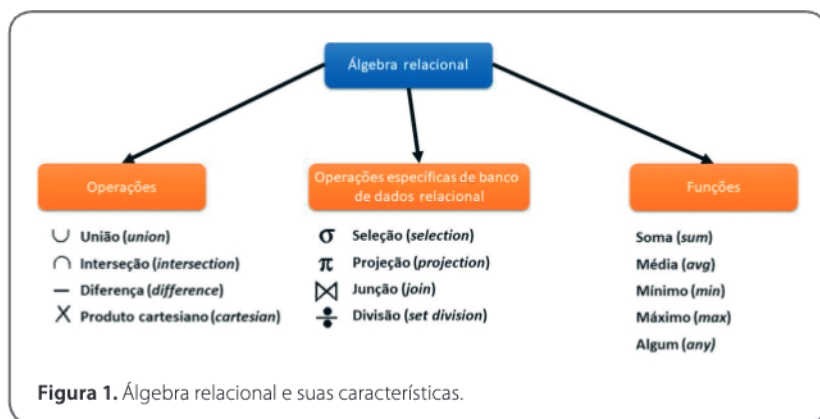
Neste capítulo, você conhecerá os conceitos das operações de junção interna e sua relação com a álgebra relacional, identificará, por meio de exemplos práticos, o uso e as diferenças de cada uma das cláusulas de condição que compõem essas operações, e, por fim, analisará instruções em código Structured Query Language (SQL) com o padrão utilizado pelo SGBD PostgreSQL.

1 Junção interna e álgebra relacional

A álgebra relacional é uma linguagem de consulta formal associada ao modelo relacional que utiliza um conjunto de operadores (Figura 1). Uma das suas características fundamentais reside naquela que considera que todo operador na álgebra relacional aceita uma ou duas instâncias de relação com argumentos, resultando apenas em uma instância de relação. De acordo com

Ramakrishnan e Gehrke (2008), essa propriedade facilita a composição de operadores para estruturar uma consulta completa, ou seja, uma expressão de álgebra relacional é recursivamente definida com uma relação, um operador de álgebra binário aplicado a duas expressões ou um operador unário aplicado a uma única expressão.

Vale ressaltar que cada consulta relacional descreve um procedimento detalhado para obter a resposta esperada, respeitando-se a ordem de aplicação na consulta dos operadores. A essência procedural da álgebra considera que uma expressão algébrica é como uma receita ou um plano, utilizada pelos sistemas relacionais, na avaliação de uma consulta, para representar planos de avaliação das consultas. A Figura 1 apresenta as características da álgebra relacional organizadas em três categorias — operações de conjuntos, operações específicas de banco de dados relacional e funções aritméticas.



Junções

No contexto da álgebra relacional, a “junção” é considerada uma das operações mais utilizadas, bem como a maneira mais empregada para combinar informações de duas ou mais relações. Mesmo que sejam definidas como um produto cartesiano seguido de seleções e projeções, as junções são mais frequentes na prática do que os produtos cartesianos normais (RAMAKRISHNAN; GEHRKE, 2008).

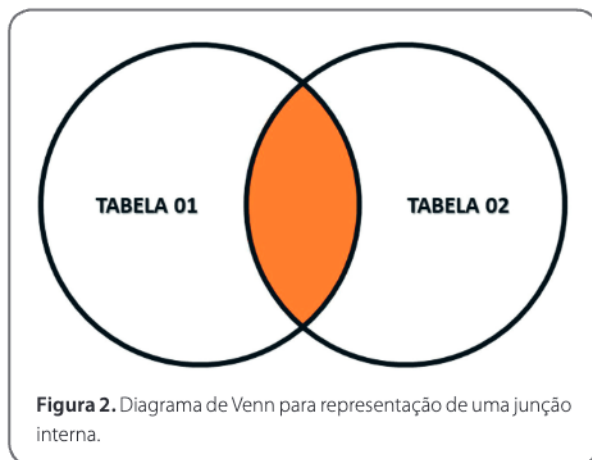
Ainda, o resultado de determinado produto cartesiano é, em geral, bem maior do que o de uma junção, além de ser imprescindível reconhecer junções e implementá-las sem materializar o produto cartesiano subjacente. Portanto, há diversas variantes da operação “junção” a que devemos nos atentar.

De acordo com Heuser (2008), o processo de integração de modelos se inicia pela junção de tabelas que têm uma mesma chave primária, ou seja, quando se usa a “mesma chave primária”, define-se que os domínios e os conteúdos dos atributos que compõem a chave primária são idênticos. Quando isso ocorre, as distintas tabelas devem ser unidas em uma única tabela no modelo global.

A versão mais básica da operação junção aceita uma **condição de junção** c e um par de instâncias de relação como argumentos, resultando em uma instância de relação, como mostrado na Figura 2. A condição de junção é idêntica à **condição de seleção** quanto ao seu formato, uma operação definida pela expressão:

$$R \bowtie_c M = \sigma_c (R \times M)$$

Assim, \bowtie é definida como um produto cartesiano seguido de uma seleção. Note que a condição c pode referenciar os atributos tanto de R quanto de M . A referência a determinado atributo de uma relação, por exemplo, R , pode se dar pela posição ($R.i$) ou pelo seu nome ($R.nome$).



2 Cláusulas de junção em SQL

Quando abordamos junções internas, é imprescindível especificar a condição de junção, ou seja, quais as linhas de determinada tabela têm relação com a linha de outra tabela. Portanto, é necessário usar as cláusulas `ON` ou `USING`, além do termo `NATURAL` no comando.

Com frequência, a cláusula `ON` é utilizada por semelhança à cláusula `WHERE`, isto é, um par de tuplas (linhas) da `Tabela_01` e `Tabela_02` será equivalente se a expressão da cláusula `ON` produzir um resultado verdadeiro (`TRUE`).

Já a cláusula `USING` é similar a `ON` por também retornar um valor booleano (verdadeiro ou falso) para um conjunto de linhas específico, embora se trate de uma das maneiras mais rápidas e abreviadas para implementar a consulta. Utilizando o nome de coluna, a execução da consulta fará a busca nas tabelas da coluna especificada e, depois de encontrá-la, realizará a comparação. Por exemplo:

```
Tabela_01 INNER JOIN Tabela_02 USING (x, y, z)
```

é o mesmo que:

```
Tabela_01 INNER JOIN Tabela_02 ON  
  (Tabela_01.x = Tabela_02.x AND  
   Tabela_01.y = Tabela_02.y AND  
   Tabela_01.z = Tabela_02.z)
```

Com o objetivo de facilitar o entendimento, utiliza-se o termo `NATURAL`, que pode ser considerado a abreviação de `USING`, já que, por meio dele, a consulta buscará todas as colunas com nomes iguais nas duas tabelas, realizando a comparação de igualdade. Contudo, é preciso se atentar ao uso de `NATURAL`, pois ele realizará a verificação de todas as colunas com nomes iguais, o que pode apresentar resultados inesperados quando da existência de duas colunas com o mesmo nome e nenhuma relação.

Equijunção

De acordo com Ramakrishnan e Gehrke (2008), outra maneira comum da operação de junção $R \bowtie M$ refere-se a uma situação em a condição de junção consiste em igualdades (conectadas pelo símbolo \wedge) no formato $R.nome1 = M.nome2$, ou seja, igualdades entre dois atributos em R e M . Nesse caso, nota-se que há redundância em manter os atributos no resultado. Para junções que têm apenas essas igualdades, a operação junção pode ser refinada fazendo-se uma projeção adicional na qual se exclui $M.nome2$, uma operação conhecida como **equijunção**.

O esquema do resultado de uma equijunção contém os campos de R seguidos pelos campos de M que não aparecem nas condições de junção. Se esse conjunto de campos na relação do resultado incluir dois campos que herdam os mesmos nomes de R e M , não são nomeados na relação do resultado.

Junção natural

Outra forma de realizar a operação junção $R \bowtie M$ consiste em uma equijunção na qual as igualdades são especificadas para todos os campos com os mesmos nomes em R e M . Nesse caso, podemos simplesmente omitir a condição de junção. Vale destacar que o padrão deve ser de que a condição de junção seja somente uma coleção de igualdades em todos os campos comuns, situação conhecida como junção natural e que tem a como propriedade garantir que o resultado não tenha dois campos com os mesmos nomes (HEUSER, 2008; RAMAKRISHNAN; GEHRKE, 2008).

Na Figura 3, é apresentada a abrangência de cada uma das junções discutidas até o momento.

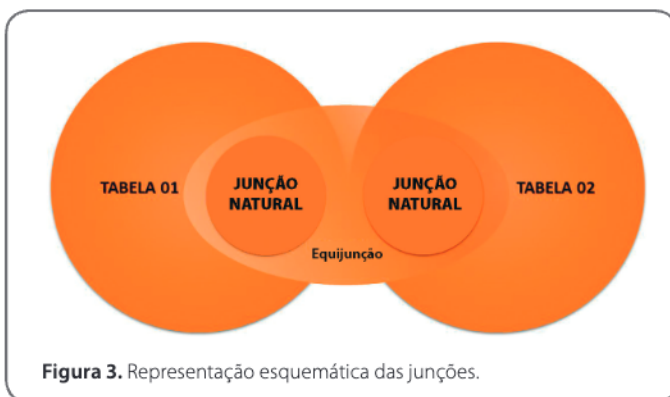


Figura 3. Representação esquemática das junções.

É importante ressaltar que existem outras terminologias utilizadas além de tabela, linha e coluna. Basicamente, utilizam-se três terminologias equivalentes, em razão dos diferentes grupos de usuários de bancos de dados: a terminologia orientada a tabelas é mais usada por usuários finais, a orientada a conjuntos, por pesquisadores acadêmicos, e a orientada a registros, por profissionais de sistemas de informação (MANNINO, 2008).

O operador de junção natural não é primitivo, já que pode ser derivado de outros operadores (MANNINO, 2008). O operador de junção natural compõe-se por três etapas, listadas a seguir.

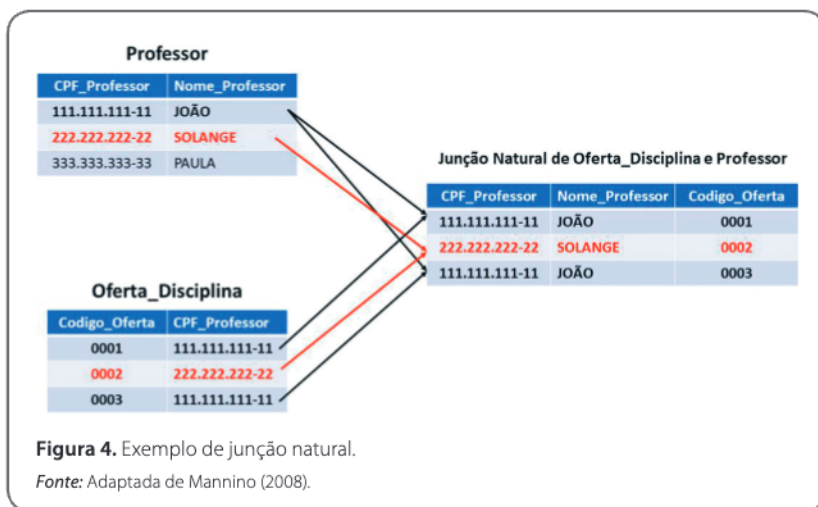
1. Uma operação de produto para combinar as linhas.
2. Uma operação de restrição para remover as linhas que não atendem à condição de junção.
3. Uma operação de projeção para remover uma das colunas de junção.



Fique atento

Os operadores de **restrição** e de **projeção**, que criam subconjuntos de uma tabela, são muito utilizados porque, na maioria das vezes, os usuários desejam ver um subconjunto, e não uma tabela inteira.

Para fixar o conceito de junção natural, a Figura 4 apresenta a junção das amostras das tabelas *Professor* e *Oferta_Disciplina*, em que a condição de junção estabelece que as colunas *CPF_Professor* sejam iguais. As setas indicam como as linhas das tabelas de entrada são combinadas para formar as tuplas da tabela resultante, ou seja, a primeira linha da tabela *Professor* é combinada com as linhas 1 e 3 da tabela *Oferta_Disciplina*, apresentadas na tabela resultante.



Para compreendermos melhor o conceito de banco de dados relacional, torna-se necessário entender as ligações, isto é, os relacionamentos existentes entre as tabelas. As linhas de uma tabela, também conhecidas como tuplas, geralmente estão relacionadas a linhas de outras tabelas. E a correspondência entre os valores (idênticos) indica os relacionamentos entre as tabelas (MANNINO, 2008).



Saiba mais

As junções (*joins*) são utilizadas em substituição às subconsultas (*subquery*), pois são mais simples de entender, ler e executar, além de possibilitarem uma instrução muito mais limpa.

Existem muitos bancos de dados relacionais disponíveis no mercado que executam instruções SQL, como Oracle, MySQL, SQL Server e PostgreSQL, sendo o último utilizado como ferramenta dos exemplos que mostraremos a seguir. O PostgreSQL é um poderoso sistema gerenciador de banco de dados objeto-relacional de código aberto, cuja popularidade aumentou nos últimos anos pela busca de um sistema com melhores garantias de confiabilidade, melhores recursos de consulta, mais operação previsível ou apenas algo fácil de aprender, entender e usar (CARVALHO, 2017).

3 Implementação de junções em banco de dados

No PostgreSQL, uma tabela unida é criada a partir de duas outras tabelas com base nas regras do tipo específico de operação de junção. Dos vários tipos de operação de junção no PostgreSQL, a junção NATURAL é um tipo especial de operação de junção INNER, retornando registros das tabelas do PostgreSQL e agrupando-os com base em correspondências encontradas nos valores dos atributos das diversas tabelas.

Uma junção natural cria uma junção implícita com fundamento nos mesmos nomes de coluna nas tabelas que se relacionam nesse processo. Sua sintaxe do PostgreSQL corresponde a:

```
SELECT *  
FROM Tabela_01  
NATURAL INNER JOIN Tabela_02;
```

Uma junção natural pode ser interna (INNER JOIN), esquerda (LEFT JOIN) ou direita (RIGHT JOIN), porém trabalharemos aqui apenas a interna. Caso não seja especificada explicitamente uma junção, o PostgreSQL considerará a junção interna por padrão.

Vale destacar que, se utilizarmos o asterisco (*) na lista de seleção, o resultado conterá as seguintes colunas:

- todas as colunas comuns — as colunas nas duas tabelas que têm o mesmo nome;
- todas as colunas da primeira e da segunda tabelas que não são comuns;

Para facilitar o entendimento dos exemplos, serão criadas as tabelas `categorias` e `produtos`, para o qual utilizaremos a instrução `CREATE TABLE`.

```
CREATE TABLE categorias (  
  categoria_id serial PRIMARY KEY,  
  categoria_nome VARCHAR (255) NOT NULL);  
CREATE TABLE produtos (  
  produto_id serial PRIMARY KEY,  
  produto_nome VARCHAR (255) NOT NULL,  
  categoria_id INT NOT NULL,  
  FOREIGN KEY (categoria_id) REFERENCES categorias (categoria_id));
```

Cada categoria tem “nenhum” ou “muitos” produtos, enquanto “um” produto pertence a “apenas” uma categoria. O atributo `categoria_id` na tabela de produtos é a chave estrangeira que faz referência à chave primária da tabela de categorias. E a `categoria_id` é o atributo comum usado para realizar a junção natural.

O próximo passo consiste em inserir alguns dados nas tabelas `categorias` e `produtos`, o que será feito pelo comando `INSERT`.

```
INSERT INTO categorias (categoria_nome)  
VALUES  
  ('Celular'),  
  ('Notebook'),  
  ('Tablet');  
INSERT INTO produtos (produto_nome, categoria_id)  
VALUES  
  ('iPhone', 1),  
  ('Dell Inspiron', 2),  
  ('iPad', 3),  
  ('Acer Aspire', 2),  
  ('Samsung A30', 1),  
  ('Galaxy Tab', 3);
```

A instrução a seguir usa a cláusula `NATURAL JOIN` para associar a tabela de produtos à tabela `categorias`, como mostrado no Quadro 1.

```
SELECT *  
FROM produtos  
NATURAL JOIN categorias;
```

Quadro 1. Resultado da seleção realizada nas tabelas `produtos` e `categorias`

CATEGORIA_ ID	PRODUTO_ID	PRODUTO_ NOME	CATEGORIA_ NOME
1	1	iPhone	Celular
2	2	Dell Inspiron	Notebook
3	3	iPad	Tablet
2	4	Acer Aspire	Notebook
1	5	Samsung A30	Celular
3	6	Galaxy Tab	Tablet

A instrução que gerou o Quadro 1 é equivalente à declaração a seguir, que utiliza a cláusula `INNER JOIN`.

```
SELECT *  
FROM produtos  
INNER JOIN categorias USING (categoria_id);
```

A vantagem do `NATURAL JOIN` reside no fato de que não é preciso especificar a cláusula de `JOIN`, pois utiliza uma cláusula `JOIN` implícita com base no atributo comum. No entanto, devemos evitar usar o `NATURAL JOIN`, pois pode causar um resultado inesperado.

Por exemplo, vamos dar uma olhada nas tabelas `cidade` e `países`; ambas têm a mesma coluna `Pais_ID`, para que possamos usar o `NATURAL JOIN` para ingressar nessas tabelas da seguinte maneira:

```
SELECT *  
FROM cidade  
NATURAL JOIN pais;
```

O resultado do `NATURAL JOIN` executado é apresentado no Quadro 2.

Quadro 2. Resultado da seleção realizada nas tabelas `cidade` e `pais`

PAIS_ID	ULTIMA_ATUALIZACAO	CIDADE_ID	PAIS

A consulta retorna um conjunto de resultados vazio, visto que ambas as tabelas também têm um atributo comum chamado `ULTIMA_ATUALIZACAO`, que não pode ser usado para a associação. No entanto, a cláusula `NATURAL JOIN` emprega somente o atributo `ULTIMA_ATUALIZACAO`.



Saiba mais

Uma junção `NATURAL` agrupa registros com base nas semelhanças com os valores da coluna encontrados em outras tabelas. Uma junção `NATURAL` pode ser uma junção esquerda (`LEFT JOIN`), interna (`INNER JOIN`) ou direita (`RIGHT JOIN`), mas o tipo de junção deve ser especificado, pois o PostgreSQL usará a operação de junção interna (`INNER JOIN`) por padrão.

Um novo exemplo é apresentado a seguir, ilustrando a criação das tabelas empregados e departamentos, conforme a seguinte instrução:

```
CREATE TABLE empregados (  
    empreg_id serial PRIMARY KEY,  
    empreg_nome VARCHAR(20) NOT NULL);  
CREATE TABLE departamentos (  
    depart_cod serial PRIMARY KEY,  
    depart_nome VARCHAR (20) NOT NULL,  
    empreg_id INT NOT NULL,  
    FOREIGN KEY (empreg_id)  
    REFERENCES empregados (empreg_id));
```

Observe que, na tabela departamentos, o atributo empreg_id é uma chave estrangeira e faz referência aos empregados como a chave primária da tabela do PostgreSQL. Como as duas tabelas apresentam a coluna empreg_id, ambas podem ser usadas para demonstrar a operação de junção NATURAL do PostgreSQL.

1. Com as tabelas criadas, agora podem ser inseridas novas linhas (tuplas) nas tabelas PostgreSQL empregados e departamentos. A instrução `INSERT INTO` é usada para inserir registros em uma tabela, cadastrando os empregados no banco de dados.

```
INSERT INTO empregados (empreg_nome)  
VALUES  
    ('João'),  
    ('Pedro'),  
    ('Maria'),  
    ('Otávio'),  
    ('Amanda');
```

2. O próximo passo refere-se à inserção dos departamentos no banco de dados.

```
INSERT INTO departamentos (depart_nome, empreg_id)
VALUES
('Folha Pagamento', 1),
('Financeiro', 1),
('Faturamento', 2),
('Vendas', 2),
('Marketing', 3),
('Recursos Humanos', 4),
('Contabilidade', 5),
('Fiscal', 5);
```

3. Para demonstrar a união das tabelas PostgreSQL empregados e departamentos usando a junção NATURAL, executa-se a instrução a seguir.

```
SELECT *
FROM departamentos
NATURAL JOIN empregados;
```

4. O resultado da instrução SELECT com uma junção NATURAL deve apresentar os registros do Quadro 3.

Quadro 3. Resultado da seleção realizada nas tabelas departamentos e empregados

empreg_id	depart_cod	depart_nome	empreg_nome
1	1	Folha Pagamento	João
1	2	Financeiro	João
2	3	Faturamento	Pedro
2	4	Vendas	Pedro
3	5	Marketing	Maria
4	6	Recursos Humanos	Otávio
5	7	Contabilidade	Amanda
5	8	Fiscal	Amanda

O objetivo do modelo relacional se baseia na premissa de que as informações em um banco de dados podem ser consideradas relações matemáticas e representadas, de maneira uniforme, por meio do uso de tabelas (com linhas, que são as ocorrências de uma entidade, e colunas, os atributos de uma entidade do modelo conceitual). Por fim, nota-se a importância da utilização das junções, sendo a natural a primeira entre outras, no processo de construção de consultas que buscam uma análise mais aprofundada, permitindo aos administradores de banco de dados relacionais recuperar as informações de maneira consistente e eficaz.



Referências

CARVALHO, V. *PostgreSQL: banco de dados para aplicações web modernas*. [S. l.]: Caso do Código, 2017.

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

MANNINO, M. V. *Projeto, desenvolvimento de aplicações e administração de banco de dados*. 3. ed. Porto Alegre: AMGH, 2008.

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: AMGH, 2008.

Leitura recomendada

CARLO, D. *et al.* Um estudo exploratório das ferramentas de código aberto para a replicação de dados no PostgreSQL. In: ESCOLA REGIONAL DE BANCO DE DADOS (ERBD), 15., 2019, Chapecó. *Anais [...]*. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 11–20. Disponível em: <https://doi.org/10.5753/erbd.2019.8474>. Acesso em: 16 jun. 2020



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.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

