CONSULTAS EM BANCO DE DADOS

Marcel Santos Silva

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Junção externa em SQL

Objetivos de aprendizagem

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

- Descrever a operação de junção externa em álgebra relacional.
- Diferenciar junção externa à esquerda, à direita e completa.
- Aplicar a operação de junção externa em SQL.

Introdução

Com o surgimento dos computadores, as informações necessitavam de um local para armazenamento, o que deu origem aos bancos de dados, que consistem em guardar grandes volumes de informações digitais, e, por consequência, a um novo seguimento, conhecido como sistemas de gerenciamento de banco de dados (SGBD), que têm como principal vantagem o estado coerente dos dados armazenados no banco. De acordo com Alves (2014), atualmente os bancos de dados representam um componente essencial na vida da sociedade, uma vez que possibilitam encontrar as mais variadas atividades de algum modo relacionadas a eles. Assim, surge a junção, a maneira mais utilizada para combinar as informações de duas ou mais relações, no contexto de banco de dados, a tabela.

Neste capítulo, você conhecerá os conceitos das operações de junção externa em SQL 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 (p. ex., junção direita, esquerda e completa), e, por fim, será capaz de analisar instruções em código SQL (Structured Query Language) com a codificação-padrão do sistema gerenciador de banco de dados, o PostgreSQL.

1 Junção externa em álgebra relacional

A álgebra relacional se define como uma linguagem de consulta formal associada ao modelo relacional, estruturada em operações específicas de banco de dados relacional, em operações de conjuntos e em funções aritméticas. Algumas variações da operação junção, suportadas por SQL, baseiam-se em valores nulos, conhecidas como junções externas (*outer join*). Tais associações retornam todas as tuplas a partir de, ao menos, uma das tabelas descritas na cláusula `FROM`, contanto que as linhas atendam às condições de busca explicitadas na cláusula `WHERE` (RAMAKRISHNAN; GEHRKE, 2008).

A junção externa se caracteriza por um conjunto de operações que possibilitem manter parte ou a totalidade das linhas de relações a combinar, mesmo que não satisfaçam à condição de junção (DATE, 2005).

De acordo com Ramakrishnan e Gehrke (2008), existem três tipos de operações de junção externa, como apresentado a seguir.

Junção externa à esquerda (`LEFT JOIN`) ⟕

Mantém todas as linhas da relação à esquerda ***R*** e, para as linhas que não satisfazem à condição de junção, preenche os atributos da relação à direita ***S*** com valores `NULL`, sendo representada pela expressão:

$$R \leftouterjoin_{COND} S$$

Com o uso de `LEFT`, todas as tuplas da tabela da esquerda que não apresentam correspondentes na tabela da direita são acrescidas ao resultado da consulta. Por meio do diagrama de Venn, a Figura 1 mostra o retorno dessa condição, conforme a álgebra relacional, em que a parte preenchida corresponde aos registros retornados.

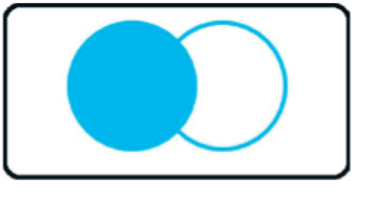

Figura 1. `LEFT JOIN`.

Junção externa à direita (RIGHT JOIN) ⟖

Mantém todas as linhas da relação à direita ***S*** e, para as linhas que não satisfazem à condição de junção, preenche os atributos da relação à esquerda ***R*** com valores NULL, sendo representada pela expressão:

$$R \rtimes_{COND} S$$

O RIGHT é o oposto da junção à esquerda, ou seja, a tabela resultante tem, incondicionalmente, uma linha para cada tupla da tabela à direita. De certo modo, a diferença entre RIGHT e LEFT reside na escolha da tabela em que os elementos que não apresentam correspondentes serão escolhidos para serem acrescidos ao resultado da consulta. A Figura 2 apresenta, pelo diagrama de Venn, o retorno dessa condição, conforme a álgebra relacional, em que a parte preenchida corresponde aos registros retornados.

Figura 2. RIGHT JOIN.

Junção externa completa (FULL JOIN) ⟗

Mantém todas as linhas de ambas as relações e, para as linhas que não satisfazem à condição de junção, preenche os atributos da relação combinada com valores NULL, sendo representada pela expressão:

$$R \rtimes_{COND} S$$

A palavra `FULL` implica na execução de `LEFT` e `RIGHT` em uma única instrução. Pelo diagrama de Venn, a Figura 3 exibe o retorno dessa condição, conforme a álgebra relacional, em que a parte preenchida corresponde aos registros retornados, ou seja, todos os registros existentes nas duas relações.

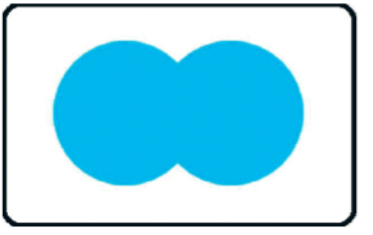

Figura 3. `FULL JOIN`.

Pode-se desabilitar os valores nulos especificando diretamente `NOT NULL` como parte da definição do atributo. Ainda, como os campos de uma chave primária não podem assumir valores nulos, existe uma restrição `NOT NULL` implícita para todo atributo definido como chave primária (`PRIMARY KEY`) (RAMAKRISHNAN; GEHRKE, 2008)

Portanto, quando se deseja selecionar linhas na tabela à esquerda que não têm linhas correspondentes na tabela à direita, `LEFT JOIN`, usa-se a restrição `IS NULL`. Com isso, o resultado apresentado considerará somente as linhas da tabela à esquerda sem correspondência com a tabela à direita, conforme a Figura 4.

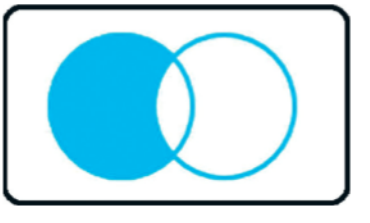

Figura 4. `LEFT JOIN` com linhas sem correspondência à direita.

Outra maneira refere-se a quando se deseja selecionar linhas da tabela à direita que não têm linhas correspondentes na tabela à esquerda, `RIGHT JOIN`, em que se usa a restrição `IS NULL`. Com isso, o resultado apresentado considerará somente as linhas da tabela à direita sem correspondência com a tabela à esquerda, como mostra a Figura 5.

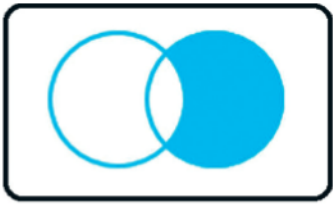

Figura 5. `RIGHT JOIN` com linhas sem correspondência à esquerda.

Por fim, quando precisamos selecionar linhas das duas tabelas que não são correspondentes, `FULL JOIN`, usamos a restrição `IS NULL` para ambos os códigos. Com isso, o resultado apresentado considerará somente as linhas sem correspondência em ambas as tabelas, de acordo com a Figura 6.

Figura 6. `FULL JOIN` com linhas sem correspondência às duas tabelas.

2 Tipos de junções externas em SQL

Para representar uma junção externa, utiliza-se a cláusula `OUTER JOIN`, emprego, contudo, opcional, uma vez que são as cláusulas `LEFT JOIN`, `RIGHT JOIN` e `FULL JOIN` que caracterizam as junções externas.

Para contextualizar os conceitos discutidos, apresentamos as tabelas já criadas com as informações necessárias para realização das consultas SQL.

- Tabela `cidade`:

codigo_cidade	nome_cidade	codigo_subregiao
1	Curitiba	1
2	Sao Paulo	2
3	Guarulhos	2
4	Buenos Aires	4
5	La Plata	4
6	Cordoba	5
7	Los Angeles	6
8	San Francisco	6
9	Orlando	7
10	Miami	7
11	Siena	8
12	Florenca	8
13	Milao	9
14	Yokohama	

- Tabela `subregião`:

codigo_subregiao	nome_subregiao	codigo_pais
1	Paraná	1
2	Sao Paulo	1
3	Rio Grande do Sul	1
4	Buenos Aires	2
5	Cordoba	2
6	Califórnia	3
7	Flórida	3
8	Toscana	4
9	Lombardia	4
10	Aquitania	5
11	Borgonha	5
12	Calábria	5
13	Massachussetts	3
14	Chiapas	

- Tabela `pais`:

codigo_pais	nome_pais
1	Brasil
2	Argentina
3	Estados Unidos
4	Itália
5	França
6	Noruega

Com as tabelas dispondo de registros, podemos realizar as operações de junções externas: a primeira é a junção à esquerda, na qual todas as tuplas da tabela sub-região sem correspondentes na tabela de cidades são acrescidas ao resultado da consulta, confirmado pela instrução a seguir:

```
SELECT * FROM subregiao LEFT OUTER JOIN cidade
USING (codigo _ subregiao);
```

O resultado da instrução utilizando o `LEFT JOIN` é apresentado a seguir:

codigo_ subregiao	nome_ subregiao	codigo_ pais	codigo_ cidade	nome_ cidade
1	Paraná	1	1	Curitiba
2	Sao Paulo	1	2	Sao Paulo
2	Sao Paulo	1	3	Guarulhos
5	Cordoba	2	6	Cordoba
6	Califórnia	3	7	Los Angeles
6	Califórnia	3	8	San Francisco
7	Flórida	3	9	Orlando
7	Flórida	3	10	Miami
8	Toscana	4	11	Siena
8	Toscana	4	12	Florença
9	Lombardia	4	13	Milao
11	Borgonha	5	NULL	NULL
12	Calábria	5	NULL	NULL
10	Aquitania	5	NULL	NULL
13	Massachussetts	3	NULL	NULL
3	Rio Grande do Sul	1	NULL	NULL
14	Chiapas	NULL	NULL	NULL

A junção à direita apresentará todas as tuplas da tabela cidade que não apresentam correspondentes na tabela sub-região como resultado da consulta, afirmação confirmada pela instrução a seguir:

```
SELECT * FROM subregiao RIGHT OUTER JOIN cidade
USING (codigo _ subregiao);
```

O resultado da instrução utilizando o `RIGHT JOIN` é o seguinte:

codigo_ subregiao	nome_ subregiao	codigo_ pais	codigo_ cidade	nome_ cidade
1	Paraná	1	1	Curitiba
2	Sao Paulo	1	2	Sao Paulo
2	Sao Paulo	1	3	Guarulhos
4	NULL	NULL	4	Buenos Aires
4	NULL	NULL	5	La Plata
5	Cordoba	2	6	Cordoba
6	Califórnia	3	7	Los Angeles
6	Califórnia	3	8	San Francisco
7	Flórida	3	9	Orlando
7	Flórida	3	10	Miami
8	Toscana	4	11	Siena
8	Toscana	4	12	Florença
9	Lombardia	4	13	Milao
NULL	NULL	NULL	14	Yokohama

Por fim, a instrução utilizando o `FULL JOIN` apresentará todas as tuplas das duas tabelas, independentemente de terem ou não relação, necessidade ilustrada pela instrução a seguir:

```
SELECT * FROM subregiao FULL OUTER JOIN cidade
USING (codigo _ subregiao);
```

O resultado da instrução utilizando o `FULL JOIN` é:

codigo_subregiao	nome_subregiao	codigo_pais	codigo_cidade	nome_cidade
1	Paraná	1	1	Curitiba
2	Sao Paulo	1	2	Sao Paulo
2	Sao Paulo	1	3	Guarulhos
4	NULL	NULL	4	Buenos Aires
4	NULL	NULL	5	La Plata
5	Cordoba	2	6	Cordoba
6	Califórnia	3	7	Los Angeles
6	Califórnia	3	8	San Francisco
7	Flórida	3	9	Orlando
7	Flórida	3	10	Miami
8	Toscana	4	11	Siena
8	Toscana	4	12	Florença
9	Lombardia	4	13	Milao
NULL	NULL	NULL	14	Yokohama
11	Borgonha	5	NULL	NULL
12	Calábria	5	NULL	NULL
10	Aquitania	5	NULL	NULL
13	Massachus-setts	3	NULL	NULL
3	Rio Grande do Sul	1	NULL	NULL
14	Chiapas	NULL	NULL	NULL

Saiba mais

Alguns SGBD não dispõem de suporte diretamente ao operador de junção externa completa. Portanto, nesses sistemas, a junção externa completa é formulada tomando a união (`UNION`) de duas junções externas de um único lado, ou seja, duas `LEFT JOIN` ou `RIGHT JOIN`.

3 Aplicação de junções externas em bancos de dados

Com o objetivo de facilitar o entendimento dos exemplos, serão criadas as tabelas `cesta1` e `cesta2`. Para isso, usaremos a instrução `CREATE TABLE`.

```
CREATE TABLE cesta1 (
cesta1 _ id INT PRIMARY KEY,
cesta1 _ fruta VARCHAR (100) NOT NULL);
CREATE TABLE cesta2 (
cesta2 _ id INT PRIMARY KEY,
cesta2 _ fruta VARCHAR (100) NOT NULL);
```

O próximo passo consiste em inserir alguns dados nas tabelas `cesta1` e `cesta2`, o que será realizado pelo comando `INSERT`.

```
INSERT INTO cesta1 (cesta1 _ id, cesta1 _ fruta)
VALUES
(1, 'Laranja'),
(2, 'Banana'),
(3, 'Melancia'),
(4, 'Maçã');
INSERT INTO cesta1 (cesta2 _ id, cesta2 _ fruta)
VALUES
(1, 'Pera'),
(2, 'Maçã'),
(3, 'Laranja'),
(4, 'Morango');
```

As tabelas apresentam algumas frutas em comum, como maçã e laranja — chamaremos a `cesta1` de tabela esquerda e a `cesta2` de tabela direita.

A instrução a seguir une a tabela esquerda à tabela direita usando a junção esquerda (ou junção externa esquerda):

```
SELECT
    a.cesta1_id,
    a.cesta1_fruta,
    b.cesta2_id,
    b.cesta2_fruta
FROM
    cesta1 a
LEFT JOIN cesta2 b ON a.cesta1_fruta = b.cesta2_fruta;
```

O resultado da instrução é apresentado no Quadro 1, a partir da utilização da cláusula `LEFT JOIN`. Note que a junção esquerda retorna um conjunto completo de linhas da tabela à esquerda com as linhas correspondentes, se disponíveis na tabela à direita. Se não houver correspondência, o lado direito terá valores nulos.

Quadro 1. Resultado do uso da junção esquerda

cesta1_id	cesta1_fruta	cesta2_id	cesta2_fruta
1	Laranja	3	Laranja
2	Banana	(null)	(null)
3	Melancia	(null)	(null)
4	Maçã	2	Maçã

Para obter somente os registros da tabela à esquerda que não têm correspondência na tabela da direita, usamos a cláusula `WHERE`, seguida de `IS NULL`:

```
SELECT
    a.cesta1 _ id,
    a.cesta1 _ fruta,
    b.cesta2 _ id,
    b.cesta2 _ fruta
FROM
    cesta1 a
LEFT JOIN cesta2 b ON a.cesta1 _ fruta = b.cesta2 _ fruta
WHERE b.cesta2 _ id IS NULL;
```

O resultado dessa instrução é apresentado no Quadro 2.

Quadro 2. Resultado do uso da junção esquerda para relações nulas

cesta1_id	cesta1_ fruta	cesta2_id	cesta2_ fruta
2	Banana	(null)	(null)
3	Melancia	(null)	(null)

A junção direita ou a junção externa direita corresponde a uma versão invertida da junção esquerda, produzindo um conjunto de resultados que contém todas as linhas da tabela direita, com as linhas correspondentes da tabela à esquerda. Se não houver correspondência, o lado esquerdo conterá valores nulos.

A instrução a seguir executa a junção direita entre as tabelas da esquerda e da direita:

```
SELECT
    a.cesta1 _ id,
    a.cesta1 _ fruta,
    b.cesta2 _ id,
    b.cesta2 _ fruta
FROM
    cesta1 a
RIGHT JOIN cesta2 b ON a.cesta1 _ fruta = b.cesta2 _ fruta;
```

O resultado da instrução é apresentado no Quadro 3, que utilizou a cláusula `RIGHT JOIN`.

Quadro 3. Resultado do uso da junção direita

cesta1_id	cesta1_ fruta	cesta2_id	cesta2_ fruta
(null)	(null)	1	Pera
4	Maçã	2	Maçã
1	Laranja	3	Laranja
(null)	(null)	4	Morango

Do mesmo modo, para obter somente os registros da tabela à direita, que não têm correspondência na tabela à esquerda, usamos a cláusula `WHERE`, seguida de `IS NULL`, conforme a seguir:

```
SELECT
    a.cesta1_id,
    a.cesta1_fruta,
    b.cesta2_id,
    b.cesta2_fruta
FROM
    cesta1 a
RIGHT JOIN cesta2 b ON a.cesta1_fruta = b.cesta2_fruta
WHERE a.cesta1_id IS NULL;
```

O resultado da instrução, com a inserção da cláusula `WHERE`, é apresentado no Quadro 4.

Quadro 4. Resultado do uso da junção direita para relações nulas

cesta1_id	cesta1_ fruta	cesta2_id	cesta2_ fruta
(null)	(null)	1	Pera
(null)	(null)	4	Morango

A junção externa completa ou junção completa produz um conjunto de resultados que contém todas as linhas das tabelas à esquerda e à direita, com as linhas correspondentes de ambos os lados, quando disponíveis. Se não houver correspondência, o lado ausente conterá valores nulos.

A declaração a seguir ilustra a junção externa completa:

```
SELECT
    a.cesta1 _ id,
    a.cesta1 _ fruta,
    b.cesta2 _ id,
    b.cesta2 _ fruta
FROM
    cesta1 a
FULL JOIN cesta2 b ON a.cesta1 _ fruta = b.cesta2 _ fruta;
```

Observe que a palavra-chave `OUTER` é opcional, pois não foi empregada na instrução anterior. O conjunto de resultados é apresentado no Quadro 5.

Quadro 5. Resultado do uso da junção completa

cesta1_id	cesta1_ fruta	cesta2_id	cesta2_ fruta
1	Laranja	3	Laranja
2	Banana	(null)	(null)
3	Melancia	(null)	(null)
4	Maçã	2	Maçã
(null)	(null)	1	Pera
(null)	(null)	4	Morango

Para retornar um conjunto de linhas exclusivas das tabelas à esquerda e à direita, é necessário, primeiro, utilizar a junção completa e, depois, excluir as linhas não desejadas dos dois lados usando uma cláusula `WHERE`, seguida de `IS NULL`:

```
SELECT
    a.cesta1 _ id,
    a.cesta1 _ fruta,
    b.cesta2 _ id,
    b.cesta2 _ fruta
FROM
    cesta1 a
FULL JOIN cesta2 b ON a.cesta1 _ fruta = b.cesta2 _ fruta
WHERE a.cesta1 _ id IS NULL OR b.cesta2 _ id IS NULL;
```

O conjunto de resultados é apresentado no Quadro 6.

Quadro 6. Resultado do uso da junção completa para relações nulas

cesta1_id	cesta1_fruta	cesta2_id	cesta2_fruta
2	Banana	(null)	(null)
3	Melancia	(null)	(null)
(null)	(null)	1	Pera
(null)	(null)	4	Morango

Como você deve ter notado, o objetivo das junções externas consiste em promover a busca de informações a partir da relação entre tabelas em bancos de dados relacionais, a fim de possibilitar a realização de operações nas quais se pode manter parte dos registros ou a sua totalidade, em relações de combinação, mesmo que tais registros não satisfaçam à condição explicitada na instrução SQL criada.

Referências

ALVES, W. P. *Banco de dados*. São Paulo: Érica, 2014. 160 p.

DATE, C. J. *SQL e teoria relacional*: como escrever códigos SQL precisos. São Paulo: Novatec. 2015. 536 p.

RAMAKRISHNAN, R.; GEHRKE, J. *Sistemas de gerenciamento de banco de dados*. 3. ed. Porto Alegre: AMGH; Bookman, 2008. 905 p.

Leituras recomendadas

CARVALHO, V. *PostgreSQL*: banco de dados para aplicações web modernas. São Paulo: Casa do Código, 2017. 220 p.

HEUSER, C. A. *Projeto de banco de dados*. 6. ed. Porto Alegre: Bookman, 2008. 282 p. (Série Livros Didáticos Informática UFRGS, 4).

MILANI, A. *PostgreSQL*: guia do programador. São Paulo: Novatec, 2008. 392 p.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS