



Operadores de conjuntos

Prof. Ramon Souza

O SQL permite trabalhar no resultado de duas consultas através dos operadores **UNION** (e **UNION ALL**), **INTERSECT** e **EXCEPT**.

O **UNION** combina os resultados de duas ou mais consultas, retornando todas as linhas pertencentes a todas as consultas envolvidas na execução.

Para utilizar o **UNION**, o número e a ordem das colunas precisam ser idênticos em todas as consultas e os tipos de dados precisam ser compatíveis.

O **UNION** **combina os resultados de duas ou mais consultas**, retornando todas as linhas pertencentes a todas as consultas envolvidas na execução.

Para utilizar o **UNION**, o número e a ordem das colunas precisam ser idênticos em todas as consultas e os tipos de dados precisam ser compatíveis.

Existem dois tipos de operador **UNION**, sendo eles **UNION** e **UNION ALL**.

O operador **UNION**, por padrão, executa o **equivalente a um SELECT DISTINCT**. Em outras palavras, ele combina o resultado de execução das duas consultas e então executa um SELECT DISTINCT a fim de **eliminar as linhas duplicadas**. Este processo é executado mesmo que não haja registros duplicados.

```
SELECT colunas FROM tabela1  
  
UNION  
  
SELECT colunas FROM tabela2;
```

O operador **UNION ALL** tem a mesma funcionalidade do **UNION**, porém, **não executa o SELECT DISTINCT no resultado** e apresenta todas as linhas, inclusive as linhas duplicadas.

```
SELECT colunas FROM tabela1  
  
UNION ALL  
  
SELECT colunas FROM tabela2;
```

ATENÇÃO!!!

As colunas de ambas as consultas não precisam ser exatamente as mesmas. Elas só precisam aparecer na mesma quantidade e ordem. E cada coluna deve ter o mesmo tipo de dado.

Assim, na primeira consulta podemos ter uma coluna de cadeia de caracteres chamada “nome_cliente” e na segunda consulta podemos ter uma coluna de cadeia de caracteres chamada “nome_fornecedor”. Ainda assim, será possível realizar a união, pois elas possuem o mesmo tipo de dados e estão na mesma ordem no conjunto de colunas de cada subconsulta.

| IDCliente | Nome_Cliente | Nome_Conhecido | Endereco | Cidade | CEP | Pais |
|-----------|------------------------------------|----------------|-------------------------------|-------------|-------|---------|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |

| IDFornecedor | Nome_Fornecedor | Nome_Contato | Endereco | Cidade | CEP | Pais |
|--------------|----------------------------|------------------|----------------|-------------|---------|------|
| 1 | Exotic Liquid | Charlotte Cooper | 49 Gilbert St. | London | EC1 4SD | UK |
| 2 | New Orleans Cajun Delights | Shelley Burke | P.O. Box 78934 | New Orleans | 70117 | USA |
| 3 | Grandma Kelly's Homestead | Regina Murphy | 707 Oxford Rd. | Ann Arbor | 48104 | USA |


```
SELECT Cidade FROM Clientes UNION SELECT Cidade  
FROM Fornecedores ORDER BY Cidade;
```

| Cidade |
|-------------|
| Ann Arbor |
| Berlin |
| London |
| México D.F. |
| New Orleans |

```
SELECT Cidade FROM Clientes UNION ALL SELECT  
Cidade FROM Fornecedores ORDER BY Cidade;
```

| Cidade |
|-------------|
| Ann Arbor |
| Berlin |
| London |
| México D.F. |
| México D.F. |
| New Orleans |

O operador **INTERSECT** **permite a intersecção entre consultas**, retornando as linhas que existem tanto na primeira quanto na segunda consulta. INTERSECT possui um SELECT DISTINCT implícito, logo **não retorna linhas repetidas**.

```
SELECT colunas FROM tabela1  
  
INTERSECT  
  
SELECT colunas FROM tabela2;
```

| IDCliente | Nome_Cliente | Nome_Contato | Endereco | Cidade | CEP | Pais |
|-----------|------------------------------------|----------------|-------------------------------|-------------|-------|---------|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |

| IDCliente | Nome_Cliente | Nome_Contato | Endereco | Cidade | CEP | Pais |
|-----------|---------------------|--------------------|-----------------|--------|----------|---------|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | Brazil |
| 3 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Brazil |

SELECT Cidade FROM ClientesA INTERSECT SELECT Cidade FROM ClientesB ORDER BY Cidade;

| Cidade |
|--------|
| Berlin |

O operador **EXCEPT** retorna os registros que aparecem na primeira consulta e não aparecem na segunda. EXCEPT possui um SELECT DISTINCT implícito, logo **não retorna linhas repetidas**.

```
SELECT colunas FROM tabela1  
  
EXCEPT  
  
SELECT colunas FROM tabela2;
```

| IDCliente | Nome_Cliente | Nome_Contato | Endereco | Cidade | CEP | Pais |
|-----------|------------------------------------|----------------|-------------------------------|-------------|-------|---------|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |

| IDCliente | Nome_Cliente | Nome_Contato | Endereco | Cidade | CEP | Pais |
|-----------|---------------------|--------------------|-----------------|--------|----------|---------|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | Brazil |
| 3 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Brazil |

SELECT Cidade **FROM** ClientesA **EXCEPT** **SELECT** Cidade **FROM** ClientesB **ORDER BY** Cidade;

Cidade

México D.F.

| OPERADOR | RETORNO |
|-----------|---|
| UNION | Todas as linhas pertencentes as consultas envolvidas, sem as repetições. |
| UNION ALL | Todas as linhas pertencentes as consultas envolvidas, incluindo as repetições. |
| INTERSECT | Linhas que estão tanto na primeira quanto na segunda consulta. Intersecção, sem repetições. |
| EXCEPT | Linhas que estão na primeira, mas não estão na segunda, sem repetições. |

(CESPE - 2016 - TCE-PA - Auxiliar Técnico de Controle Externo - Área Informática) No que concerne a banco de dados, julgue o item a seguir.

O resultado da consulta a seguir, que utiliza o operador UNION, não elimina os registros duplicados entre as tabelas.

```
SELECT depto FROM emp
```

```
UNION
```

```
SELECT depto FROM depto;
```

(FGV - 2018 - Prefeitura de Niterói - RJ - Analista de Políticas Públicas e Gestão Governamental - Gestão de Tecnologia) A questão deve ser respondida a partir das tabelas de banco de dados t1 e t2, a seguir.

Analise o comando SQL exibido abaixo.

```
select * from T1 where C > 5
```

```
UNION
```

```
select * from T1 where C <= 5
```

A execução desse comando no MS SQL Server produz um resultado que contém, além da linha de títulos, n linhas.

Assinale o valor de n.

- a) 3 b) 4 c) 5 d) 6 e) 8

| T1 | | | T2 | | |
|----|---|------|----|----|------|
| A | B | C | D | E | F |
| 1 | 2 | 4 | 1 | 2 | NULL |
| 2 | 3 | 5 | 2 | 5 | 5 |
| 4 | 2 | 4 | 4 | 2 | 1 |
| 6 | 2 | NULL | 7 | 12 | 1 |

(FGV - 2018 - Banestes - Analista em Tecnologia da Informação - Desenvolvimento de Sistemas) Considere um banco de dados com duas tabelas, R e S, cujas instâncias são exibidas a seguir.

Na execução do comando SQL

```
select * from R left join S on a=b
```

UNION

```
select * from R right join S on b=a
```

o número de células contendo o valor nulo no resultado é:

- a) 0 b) 3 c) 6 d) 9 e) 12

| R | |
|---|---|
| a | b |
| 1 | 2 |
| 2 | 3 |
| 4 | 5 |

| S | |
|---|---|
| c | d |
| 3 | 2 |
| 4 | 2 |
| 6 | 1 |

O left join gerará o seguinte resultado:

| a | b | c | d |
|---|---|------|------|
| 1 | 2 | null | null |
| 2 | 3 | null | null |
| 4 | 5 | null | null |

O right join gerará o seguinte resultado:

| a | b | c | d |
|------|------|---|---|
| null | null | 3 | 2 |
| null | null | 4 | 2 |
| null | null | 6 | 1 |

O UNION irá juntar os resultados, logo teremos:

| a | b | c | d |
|------|------|------|------|
| 1 | 2 | null | null |
| 2 | 3 | null | null |
| 4 | 5 | null | null |
| null | null | 3 | 2 |
| null | null | 4 | 2 |
| null | null | 6 | 1 |

(FGV - 2017 - MPE-BA - Analista Técnico - Tecnologia) Atenção

Tabelas R e S referentes a um banco de dados relacional.

Considerando as tabelas R e S apresentadas anteriormente, o comando SQL

`select a FROM R UNION ALL select d FROM S`

produz um resultado que contém, além dos títulos:

- a) 1 linha;
- b) 3 linhas;
- c) 4 linhas;
- d) 5 linhas;
- e) 6 linhas.

| R | |
|---|---|
| a | b |
| 1 | 2 |
| 2 | 3 |
| 4 | 5 |

| S | |
|---|---|
| c | d |
| 3 | 2 |
| 4 | 2 |
| 6 | 1 |



Junções (Joins)

Prof. Ramon Souza