

[< VOLTAR](#)

Operações com conjuntos

Criar relatórios trabalhando as seguintes operações de conjuntos: união, intersecção e subtração.

NESTE TÓPICO



Marcar
tópico



O operador UNION (união) combina os resultados de duas ou mais instruções em um único resultado, retornando todas as linhas pertencentes a todas as instruções envolvidas no processamento. Para utilizar o UNION, o número e a ordem das colunas precisam ser idênticos em todas as instruções e, além disso, os tipos de dados precisam ser compatíveis.

Existem dois tipos de operador UNION, a saber: UNION e UNION ALL.

1. `SELECT nome_cliente, telefone from cliente`
2. `UNION`
3. `SELECT nome_fornec, telefone from fornecedor;`

Operador UNION

O operador UNION, por *default*, executa o equivalente a um `SELECT DISTINCT` no resultado final. Ele combina o resultado de execução das duas instruções e, então, executa um `SELECT DISTINCT`, a fim de eliminar as linhas duplicadas. Este processo é executado mesmo que não haja registros duplicados. Por exemplo:

Resultado:

José da Silva

64643333

Maria Aparecida Mafra	44990087
Marlene Maria Maia	88776655
Ana Maria de Jesus	31312121

Operador UNION ALL

O operador UNION ALL tem a mesma funcionalidade do UNION, porém não executa o SELECT DISTINCT no resultado final e, além disso, apresenta todas as linhas, inclusive as linhas duplicadas. Por exemplo:

1. SELECT nome_cliente, telefone from cliente
2. UNION ALL
3. SELECT nome_fornec, telefone from fornecedor;

Resultado:

José da Silva	64643333
Maria Aparecida Mafra	44990087
Marlene Maria Maia	88776655
Ana Maria de Jesus	31312121
José da Silva	64643333
Maria Aparecida Mafra	44990087
Marlene Maria Maia	88776655
Ana Maria de Jesus	31312121

Recomendações

1) Se não existe a possibilidade de haver registros duplicados em suas tabelas ou se não houver problemas para a aplicação que o *record set* final apresente duplicações, utilize o operador UNION ALL. A vantagem é que este operador não executa a função SELECT DISTINCT, utiliza menos recursos do SQL e, como consequência, melhora o desempenho da aplicação.

2) Não utilize o operador UNION em conjunto com a função SELECT DISTINCT, pois o resultado final será exatamente o mesmo, porém, o SQL estará executando a mesma operação duas vezes, causando queda de desempenho.

3) Uma instrução com uma ou mais cláusula OR pode ser reescrita utilizando o operador UNION ALL:

1. SELECT employeeID, firstname, lastname FROM names WHERE dept = "prod"
2. UNION ALL
3. SELECT employeeID, firstname, lastname FROM names WHERE city = "Orlando"
4. UNION ALL
5. SELECT employeeID, firstname, lastname FROM names WHERE division = "food"

O operador INTERSECT computa o conjunto de interseção das linhas retornadas pelos comandos SELECT envolvidos. Uma linha estará na interseção dos dois conjuntos de resultados se estiver presente nos dois conjuntos de resultados.

O resultado do operador INTERSECT não contém nenhuma linha duplicada, a menos que a opção ALL seja especificada. Usando ALL, uma linha contendo m duplicatas na tabela à esquerda e n duplicatas na tabela à direita, aparecerá *min* (m, n) vezes no conjunto de resultados.

Havendo vários operadores INTERSECT no mesmo comando SELECT, estes são avaliados da esquerda para a direita, a menos que os parênteses indiquem outra ordem. Por exemplo:

1. SELECT cd_municipio, nm_municipio FROM município
2. WHERE cd_municipio IN (SELECT cd_municipio FROM fornecedor)
3. INTERSECT
4. SELECT cd_municipio, nm_municipio FROM município
5. WHERE cd_municipio IN (SELECT nm_municipio FROM cliente);

1	Balneário Camboriú
4	Criciúma
5	Curitiba

Resultado:

O operador MINUS, traz todos os registros da primeira consulta **MENOS (MINUS)** os registros da segunda consulta (ele também ignora os repetidos).

Por exemplo:

1. SELECT cd_municipio, nm_municipio FROM município
2. WHERE cd_municipio IN (SELECT cd_municipio FROM fornecedor)
3. MINUS
4. SELECT cd_municipio, nm_municipio FROM município
5. WHERE cd_municipio IN (SELECT nm_municipio FROM cliente);

Resultado:

2	Blumenau
6	Ponta Grossa

Referências

BEIGHLEY, Lynn. *Use a Cabeça SQL*. Rio de Janeiro: Alta Books, 2008.

FANDERUFF, Damaris. *Dominando o Oracle 9i: Modelagem e Desenvolvimento*, São Paulo: Makron, 2003.

GRAVES, Mark. *Projeto de banco de dados com XML*. São Paulo: Pearson, 2003.

MORELLI, Eduardo Terra. *Oracle 9i Fundamental: SQL, PL/SQL e Administração*, São Paulo, Editora Érica, 2002.

PRICE, Jason. *Oracle Database 11g SQL*. (tradução: João Eduardo Nóbrega Tortello). Porto Alegre: Bookman, 2009.

SILVA, Robson. *Oracle Database 10g Express Edition*. São Paulo: Editora Érica, 2007.



Avalie este tópico



ANTERIOR

Criação de relatórios utilizando mais de uma
tabela



Índice

Biblioteca
(<https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

Ajuda?

PRÓXIMO
(<https://ava.uninove.br/criando-views-views>)

Criando VIEWS - Views

© Todos os direitos reservados

