Union

Union

Assim como ocorre com JOIN, uma operação UNION permite **combinar dados provenientes de duas ou mais tabelas** (ou da mesma tabela, com condições diferentes). Porém, em vez de combinar as colunas dessas tabelas, UNION combina as linhas de dois ou mais conjuntos de resultados. Enquanto *INNER JOIN é uma operação de intersecção entre conjuntos*, ***UNION é uma operação de soma de conjuntos***.

A operação UNION **combina ("une") duas ou mais** declarações SELECT. O **resultado de cada SELECT deve possuir o mesmo número de colunas e o tipo de dado de cada coluna correspondente deve ser compatível**.

Para ordenar o resultado de uma operação UNION, pode-se utilizar uma declaração ORDER BY após o último SELECT codificado. Neste ORDER BY somente é possível usar os nomes de colunas que foram declarados no primeiro SELECT da declaração UNION completa. Os nomes de colunas no resultado final são tirados deste primeiro SELECT.

Sintaxe básica de uma subconsulta:

```
SELECT declaração 1
UNION [ALL]
SELECT declaração 2
UNION [ALL]
SELECT declaração 3 …
[ORDER BY];
```

Union

Diferença entre Union e Union ALL

O operador **UNION** elimina resultados duplicados. Em resumo ele produz o mesmo resultado que um SELECT DISTINCT no resultado final da operação.

O operador **UNION ALL** é usado para incluir todos os resultados intermediários (SELECTs) no resultado final, independente de serem repetidos ou não.

Recomendações

1. Se não existe a possibilidade de haver registros duplicados nas tabelas ou se não houver problemas para a aplicação que resultado final apresente duplicações, utilize o operador UNION ALL. A vantagem é que este operador não executa a função SELECT DISTINCT, utiliza menos recursos do BD e como consequência, melhora a performance 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 BD estará executando a mesma operação duas vezes, causando queda de desempenho.
3. O apelido (alias) das colunas, quando necessário, deve ser incluído somente no primeiro SELECT.
4. A inclusão de WHERE pode ser feita em qualquer um dos comandos SELECT.
5. É possível escrever qualquer SELECT com JOIN ou SUBQUERY.

Union

Modelo de dados para os exemplos

TB_LIVROS		
IdLivro	Int	PK (auto increment)
NomeLivro	Varchar(35)	
PrecoLivro	Decimal(6,2)	
IdAssunto	Int	FK (TB_ASSUNTOS)

TB_ASSUNTOS		
IdAssunto	Int	PK (auto increment)
DescAssunto	Varchar(25)	

Union

***EXEMPLO 1*:**

Mostrar os nomes de livros, preços e assuntos dos livros.
Caso o assunto seja “Auto ajuda”, mostrar o preço acrescido de 15% em seu valor.
Caso o livro custe mais de R$ 30, descontar 10% em seu valor.
Ordenar por preço, do mais caro para o mais barato.

```
SELECT
        L.NomeLivro          AS ‘Título do livro’,
        L.PrecoLivro         AS 'Preço Normal',
        L.PrecoLivro * 0.90  AS  'Preço Ajustado',
        A.DescAssunto        AS ‘Assunto do livro’
FROM tb_Livros as L INNER JOIN tb_Assuntos as A ON L.IdAssunto = A.IdAssunto
WHERE L.PrecoLivro > 30.00
UNION ALL
SELECT
        L.NomeLivro,
        L.PrecoLivro,
        L.PrecoLivro * 1.15,
        A.DescAssunto
FROM tb_Livros as L INNER JOIN tb_Assuntos as A ON L.IdAssunto = A.IdAssunto
WHERE A.DescAssunto = ‘Auto ajuda'
ORDER  BY  3   DESC;
```

Select 1

Cláusula UNION ALL

Select 2

Número da coluna utilizada para ordenação

Union

***EXEMPLO 2*:**

Mostrar nomes de livros e preços dos livros.
Caso o preço do livro seja igual ou superior a R$ 30 mostrar a mensagem “Livro Caro” em uma coluna à direita no resultado da consulta.
Caso contrário, mostrar a mensagem “Preço Razoável”
Ordenar por preço, do mais barato para o mais caro.

```
SELECT
        L.NomeLivro         AS ‘Título do livro’,
        L.PrecoLivro        AS 'Preço Normal',
        ‘Livro caro’        AS ‘Resultado’
FROM tb_Livros as L
WHERE L.PrecoLivro >= 30.00
UNION ALL
SELECT
        L.NomeLivro         AS ‘Título do livro’,
        L.PrecoLivro        AS 'Preço Normal',
        ‘Preço razoável’    AS ‘Resultado’
FROM tb_Livros as L
WHERE L.PrecoLivro < 30.00
ORDER  BY  2  ASC;
```

Select 1

Cláusula UNION ALL (Não elimina resultados duplicados)

Select 2

Número da coluna utilizada para ordenação

Union

***EXEMPLO 3*:**

Mostrar nomes e preços dos livros que pertençam ao assunto “Romance”.
Além disso mostrar as informações dos livros com valor superior R$ 28.
Ordenar pelo nome do livro.

```
SELECT
        L.NomeLivro         AS ‘Título do livro’,
        L.PrecoLivro        AS 'Preço Normal',
        A.DescAssunto       AS ‘Assunto do livro’
FROM tb_Livros as L INNER JOIN tb_Assuntos as A ON L.IdAssunto = A.IdAssunto
WHERE A.DescAssunto = ‘Romance'

UNION

SELECT
        L.NomeLivro     AS ‘Nome do livro’,
        L.PrecoLivro,
        A.DescAssunto
FROM tb_Livros as L INNER JOIN tb_Assuntos as A ON L.IdAssunto = A.IdAssunto
WHERE L.PrecoLivro >= 28,00
ORDER  BY  1  DESC;
```

Select 1

Cláusula UNION (elimina resultados duplicados)

Select 2

Número da coluna utilizada para ordenação

Union

***EXEMPLO 4*:**

Mostrar nomes e preços dos livros que pertençam ao assunto “Romance”. Somar ao resultado anterior as informações dos livros que pertençam ao assunto “Literatura”. A ordenação do resultado final deverá ser pelo nome do livro.

```
SELECT
        L.NomeLivro,   L.PrecoLivro, A.DescAssunto
FROM tb_Livros as L INNER JOIN tb_Assuntos as A ON L.IdAssunto = A.IdAssunto
WHERE A.DescAssunto = ‘Romance'

UNION ALL

SELECT
        L.NomeLivro          AS ‘Título do livro’,
        L.PrecoLivro         AS 'Preço Normal',
        ( SELECT A.DescAssunto FROM tb_Assuntos as A
          WHERE A.IdAssunto = L.IdAssunto ) AS 'Assunto de interesse'
FROM tb_Livros as L
WHERE L.IdAssunto =
        ( SELECT A.IdAssunto FROM tb_Assuntos as A
          WHERE A.DescAssunto = 'Literatura' )
ORDER  BY  1  ASC;
```

Select 1

Cláusula UNION ALL (Não elimina resultados duplicados)

Alias devem estar no primeiro SELECT

Select 2 com subuery

Número da coluna utilizada para ordenação

Dúvidas ?