

INFORMA, FORMA, TRANSFORMA.

Banco de Dados



Fabício Curvello Gomes



INFORMA, FORMA, TRANSFORMA.




Agrupamentos e União de Tabelas



  INFORMA, FORMA, TRANSFORMA.

Introdução

Agrupamento de Dados – É o processo de combinação de colunas com valores repetidos em uma ordem lógica. O agrupamento é feito pela coluna que será usada como pivô para obter informações da tabela, como as estatísticas por meio das funções de agregação. Por exemplo, numa tabela escolar, saber rapidamente quantos alunos são do sexo masculino e quantos são do sexo feminino.

União de Dados – Significa combinar mais de uma consulta em uma única consulta. É possível combinar o resultado de duas ou mais consultas sem a apresentação de linhas de registros que sejam duplicadas. Por exemplo, imagine obter uma lista dos funcionários ativos e dos demitidos da empresa em uma mesma consulta.


3



  INFORMA, FORMA, TRANSFORMA.

Agrupamento de Dados

O agrupamento de dados na linguagem de consulta SQL é obtido com a cláusula **GROUP BY** combinada com o comando **SELECT** após o uso da cláusula **WHERE** e antes da cláusula **ORDER BY**. A estrutura sintática da instrução completa passa a ser:

```
SELECT [tipo] <campos> FROM <tabela> WHERE [condição]
      GROUP BY <campos> [ASC | DESC]
      ORDER BY <campos> [ASC | DESC]
;
```

4



 INFORMA, FORMA, TRANSFORMA.

Agrupamento de Dados (Cont.)


Na tabela **funcionario** existem as colunas DEPTO, FUNCAO, SALARIO, ADMISSAO e FILHOS, os quais têm vários registros cadastrados com denominações **repetidas** e devido a essas características é possível estabelecer várias formas de agrupamento de dados. Eis alguns exemplos:



- 1) Apresentar o número de departamentos da empresa e a quantidade de registros cadastrados em cada departamento:

```
SELECT DEPTO, COUNT(*) FROM funcionario GROUP BY DEPTO;
```

- 2) Apresentar a quantidade de funções disponíveis na empresa e de funcionários em cada função:

```
SELECT FUNCAO, COUNT(*) FROM funcionario GROUP BY FUNCAO;
```


5



 INFORMA, FORMA, TRANSFORMA.

Agrupamento de Dados (Cont.)


Para a cláusula **GROUP BY** gerar eficientemente uma determinada questão, ela deve estar associada ao uso de uma das funções de agregação: **AVG()**, **MAX()**, **MIN()**, **SUM()** e **COUNT()**.



Exemplo de agrupamento para saber o total de salários pagos por departamento:

```
SELECT DEPTO, SUM(SALARIO) FROM funcionario GROUP BY DEPTO;
```

Agora o total de salários pagos apenas por 2 departamentos:

```
SELECT DEPTO, SUM(SALARIO) FROM funcionario
WHERE DEPTO IN ('2', '4')
GROUP BY DEPTO
```



6



  INFORMA, FORMA, TRANSFORMA.

Agrupamento de Dados (Cont.)

Exemplo que apresenta a quantidade de departamentos existentes na empresa e os registros que existem em cada departamento, com resultado em ordem decrescente:

```
SELECT DEPTO, COUNT(*) FROM funcionario  
GROUP BY DEPTO ORDER BY DEPTO DESC;
```

 7

  INFORMA, FORMA, TRANSFORMA.


Agrupamento de Dados (Cont.)



A cláusula **GROUP BY** possui como auxiliar a cláusula **HAVING**, a qual estabelece as condições de saída.

O exemplo seguinte apresenta numa consulta o total de salários pagos por departamento acima de R\$8.000,00.

```
SELECT DEPTO, SUM(SALARIO) FROM funcionario  
GROUP BY DEPTO  
HAVING SUM(SALARIO)>8000  
;
```

A cláusula **GROUP BY** deve ser usada basicamente com uma função de agregação. Pode-se utilizar para auxiliar o processo de agrupamento as funções **AVG**, **COUNT**, **MIN**, **MAX** e **SUM**.

 8



INFORMA, FORMA, TRANSFORMA.



União de Dados

O recurso de união de dados baseia-se na união de mais de uma consulta em uma única consulta por meio de múltiplos comandos **SELECT**. Esse recurso é utilizado com os operadores **UNION** ou **UNION ALL** que possuem a seguinte estrutura sintática:

```
SELECT ... UNION [ALL | DISTINCT]
SELECT ... [UNION [ALL | DISTINCT]
SELECT ... ];
```

Em que **ALL** representa todos os registros encontrados na consulta, independentemente de esses registros serem ou não duplicados nas tabelas envolvidas, **DISTINCT** representa a restrição de exibição de registros, impossibilitando a apresentação de registros duplicados. Se **ALL** for omitido, o comando **UNION** assume como padrão o valor **DISTINCT** e apresenta apenas um único registro caso haja nas tabelas envolvidas alguma duplicidade.

9



INFORMA, FORMA, TRANSFORMA.



União de Dados (Cont.)

Para executar este recurso, será necessário proceder a algumas mudanças na tabela **funcionario**. Considere o fato de que a empresa decidiu demitir os oito funcionários mais antigos.

Para isso, você irá copiar os registros dos 8 funcionários mais antigos da tabela **funcionario** para a tabela morta:

- 1) Primeiro ver todos os registros da tabela **funcionario**. Observe os funcionários mais antigos.
SELECT * FROM funcionario;
- 2) Agora copiar cada um dos 8 funcionários antigos para a tabela morta:
INSERT INTO morto SELECT CODFUN, NOME, DEPTO, FUNCAO, SALARIO, ADMISSAO, FILHOS FROM funcionario WHERE CODFUN = X;

Substituir **X** pelo código do funcionário a ser copiado.
Repetir este procedimento para todos os 8 funcionários.



INFORMA, FORMA, TRANSFORMA.

União de Dados (Cont.)


3) Agora que os 8 registros foram copiados para a tabela morto, eles podem ser removidos da tabela funcionario. Execute a seguinte instrução para cada um dos 8 registros:

```
DELETE FROM funcionario WHERE CODFUN = X;
```

Substituir **X** pelo código do funcionário a ser excluído.
Os 8 funcionários excluídos precisam ser os mesmos que foram copiados anteriormente para a tabela morto.

4) Agora confirme o seu trabalho verificando as duas tabelas:

```
SELECT * FROM funcionario;  
SELECT * FROM morto;
```


11

União de Dados (Cont.)


5) Agora sim, podemos demonstrar o operador **UNION**.



Para que este operador funcione corretamente, é necessário que cada comando **SELECT** envolvido na consulta possua o mesmo número de colunas e que estas colunas estejam na mesma ordem, que possuam o mesmo número de expressões de colunas e que tenham os mesmos tipos de dados, não existindo obrigatoriedade com relação ao tamanho dos campos entre as tabelas unidas.

Ex:

```
SELECT * FROM funcionario UNION  
SELECT * FROM morto;
```

*OBS: Caso algum registro esteja repetido entre as duas tabelas, ele irá aparecer somente uma vez. Faça um teste inserindo um registro na tabela morto, idêntico a algum já existente na tabela funcionario. Depois de ver o resultado com o comando acima, repita este comando substituindo o operador **UNION** por **UNION ALL** e observe o resultado. Qual é a diferença?*


12


  INFORMA, FORMA, TRANSFORMA.

União de Dados (Cont.)

Uma consulta baseada no operador **UNION** também pode ser ordenada com a cláusula **ORDER BY**, mas para que o resultado seja o esperado, é necessário usar parênteses, como descrito abaixo:

```
(SELECT * FROM funcionario ORDER BY NOME)  
UNION  
(SELECT * FROM morto ORDER BY NOME)  
ORDER BY NOME;
```

Faça o teste e observe o resultado apresentado.


 13

  INFORMA, FORMA, TRANSFORMA.


Dúvidas?



14



Bibliografia



MySQL 5.1 Interativo (3ª Edição)
José Augusto N.G. Manzano
Ed. Érica

15