02

Agrupando dados usando o group by

Agrupando dados usando o group by

No curso anterior, nós vimos como formatar as saídas dos nossos dados e as funções que aplicávamos em um única linha. Agora, nós iremos falar mais sobre **grupos**. Estou disponibilizando aqui (https://cursos.alura.com.br/course/certificacao-oracle-4/task/16104) um dump com os dados que iremos trabalhar no decorrer do curso.

Se olharmos no meu desktop, veremos o arquivo que iremos trabalhar: `sql-oracle-4.sql` . Vou entrar no *SQL* |Plus*, utilizando o usuário `alura` .

```
 Directory of C:\Users\Alura\Desktop

03/05/2016  16:58    <DIR>          .
03/05/2016  16:58    <DIR>          ..
10/04/2016  08:16    <DIR>          oracle
22/04/2016  10:27             1.529 sql-oracle-3.sql
03/05/2016  16:58             1.524 sql-oracle-4.sql
20/04/2016  09:41         5.262.916 teste1.trec
               3 File(s)      5.265.969 bytes
               3 Dir(s) 356.626.780.160 bytes free

C:\Users\Alura\Desktop>sqlplus alura

SQL \*Plus: Release 12.1.0.2.0 Production on Ter Mai 3 17:35:03 2016

Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

Vou me logar com a minha senha e ele se conectará com sucesso.

Para executar o arquivo, nós usaremos o comando `start` .

```
SQL> start sql-oracle-4.
ORA-00942: a tabela ou view nao existe

Tabela criada.

1 linha criada.

1 linha criada.

1 linha criada.

1 linha criada.

1 linha criada.

1 linha criada.

1 linha criada.
```

```
1 linha criada.

\...
```

Ele informou que foi criada uma tabela com várias linhas.

Vamos visualizar a tabela com o comando `select`:

```
SQL> select * from funcionarios;

        ID NOME                      SALARIO PCT_COMISSAO   SETOR_ID
---------- ------------------------ ------- -------------- --------
         1 Felipe                      3000             ,1        1
         2 Davi                        2200             ,1        2
         3 Gustavo                     2200                       3
         4 Dorval                      1000             ,1        2
         5 Tereza                      2600             ,1        1
         6 Kunika                      6200                       1
         7 Joselir                     5300                       2
         8 Joseane                     2000             ,1        2
         9 Isabel                      1500             ,1        1
        10 Alexandre                   2400             ,1        3
        11 Jenifer                     2100             ,1        3

        ID NOME                      SALARIO PCT_COMISSAO   SETOR_ID
---------- ------------------------ ------- -------------- --------
        12 Hilda                      2200                ,1        2
        13 Joselita                   3450                ,1        3
        14 Edson                      4200                ,1        1

14 linhas.
```

Com o `select` vimos todos os funcionários. Agora, queremos gerar um relatório informando quando funcionário temos em cada setor. Podemos fazer este calculo manualmente, mas seria uma maneira muito numerosa.

Temos uma função que conta a quantidade de registros: `count()`. Faremos um teste com os dados da coluna `ID`.

```
SQL> select count(id) from funcionarios;

 COUNT(ID)
----------
        14
```

O Oracle retornou que temos 14 funcionários. Até aqui nenhuma novidade. A grande sacada é que a função `COUNT` possui uma pequena diferença em relação às que vimos no curso anterior: ela é uma função de **agrupamento** e não se aplica a um único registro. Por exemplo, quando usávamos a função `replace()` em uma *string*, a substituição seria feita em cada linha em que utilizássemos o `select`. No entanto, quando usamos a `count()`, ela considerou a quantidade de elementos do meu grupo. Como não definimos como separar os grupos, ela considerou o conjunto de dados da tabela.

Agora, iremos definir uma regra para separamos estes grupos. Qual será a regra? Como quero fazer uma contagem por cada setor.

Para isto, usaremos a função `group by()` - uma cláusula que usaremos bastante no curso. Também selecionaremos a coluna que iremos trabalhar com `setor_id`.

```
SQL> select setor_id from funcionarios group by setor_id;

  SETOR_ID
__________
         1
         2
         3
```

Ele retornou três registros, porque separou três grupos.

Agora, para contarmos a quantidade de registros em cada grupo, usaremos a função de agrupamento `count()`. Isto significa que descobriremos a quantidade de registros no grupo 1,2 e 3. Então, além do `setor_id`, vou selecionar `count(id)`.

```
SQL> select setor_id,count(id) from funcionarios group by setor_id;

  SETOR_ID COUNT(ID)
---------- ---------
         1         5
         2         5
         3         4
```

Observe que quando definimos o `group_by`, a regra passou a ser definida para cada agrupamento e não mais para todos os dados.

No entanto, não gostei do nome da coluna `setor_id` e irei alterá-lo para `setor`.

```
SQL> select setor_id as setor,count(id) from funcionarios group by `setor_id`;

     SETOR COUNT(ID)
---------- ---------
         1         5
         2         5
         3         4
```

Também podemos modificar o nome da coluna `count(id)` para `qtdFuncionarios`. Lembrando que adicionar o `as` é opcional.

```
SQL> select setor_id as setor,count(id) as qtdFuncionarios from funcionarios group by setor_id;

     SETOR QTDFUNCIONARIOS
---------- ---------------
         1               5
         2               5
         3               4
```

Formatamos os nomes das colunas das minhas colunas. Como modificamos o nome da coluna `setor_id`, será que podemos usar `group by()` seguida por `setor`?

```
SQL> select setor_id as setor,count(id) as qtdFuncionarios from funcionarios group by setor;
select setor_id as setor_id as setor,count(id) as qtdFuncionarios from funcionarios group by setor

ERRO na linha 1:
ORA-00904: "SETOR": identificador invalido
```

Ele retornou que `setor` é um identificador inválido, porque não podemos usar um *alias* com o `GROUP BY`. Logo, teremos que continuar chamando a coluna de `setor_id` no fim do `select`.

Também podemos nos perguntar se com o `group by()`, nós podemos passar a posição da coluna. Por exemplo, podemos querer ordenar a partir da coluna `1`. Porém, o Oracle irá informar que não é uma expressão válida.

```
SQL> select setor_id as setor, count(id) as qtdfuncionarios from funcionarios group by 1;
select setor_id as setor, count(id) as qtdFuncionarios from funcionarios group by 1

ERRO na linha:
ORA-00979: nao e uma expressao GROUP BY
```

Podemos tentar também identificar o nome dos meus funcionários.

```
SQL> select setor_id as setor, count(id) as qtdfuncionarios,nome from funcionarios group by 1;
select setor_id as setor, count(id) as qtdFuncionarios from funcionarios group by 1

ERRO na linha:
ORA-00979: nao e uma expressao GROUP BY
```

Por que nome não é uma expressão do `group by()`? Agora, não temos mais registros "soltos", eles estão baseados pelo número do setor. Como temos apenas três registros - quantidade de setores existentes - como iremos pesquisar um nome de funcionário dentro do grupo? Não é possível.

Mas o que podemos selecionar quando temos a cláusula `group by()`? Quando adicionamos o `GROUP BY` ao `select`, incluímos a regra de que só poderemos selecionar funções de agrupamento, como a `COUNT`, além das colunas integrantes do grupo. Vimos isto no seguinte exemplo:

```
SQL> select setor_id as setor,count(id) as qtdFuncionarios from funcionarios group by setor_id;
```

O `setor_id` funciona como um identificador de cada grupo, então, conseguimos selecioná-lo. Então, não conseguiremos selecionar o nome do funcionário porque ele não faz parte do identificador do grupo.

Nós já conseguimos quantificar quantos funcionários temos por setor. Em seguida, iremos selecionar os salários. Para isto, além de agruparmos por setor, criaremos um subgrupo: `salario`.

Quando usamos o `group by()`, podemos passar mais de uma coluna. Quando definimos uma segunda coluna, criamos um subgrupo. Fique atento que a **ordem** será importante. Estamos agrupando por setor e dentro de cada um, queremos agrupar pelo salário.

```
SQL> select setor_id,count(id) as qtdFuncionarios from funcionarios group by setor_id,salario;

     SETOR QTDFUNCIONARIOS
---------- ---------------
         2               1
         2               1
         1               1
         1               1
         2               1
         3               1
         3               1
         1               1
         2               2
         3               1
         1               1
     SETOR QTDFUNCIONARIOS
---------- ---------------
         1               1
         3               1

13 linhas selecionadas.
```

Vamos selecionar `salario` também, para conseguirmos visualizá-lo.

```
SQL> select setor_id as setor,salario,count(id) as qtdFuncionarios from funcionarios group by salari

     SETOR    SALARIO QTDFUNCIONARIOS
---------- ---------- ---------------
         1       3000               1
         2       5300               1
         2       2200               2
         2       2000               1
         1       1500               1
         1       4200               1
         2       1800               1

         1       2600               1
         1       6200               1
         3       2100               1
         3       2200               1
     SETOR    SALARIO QTDFUNCIONARIOS
---------- ---------- ---------------
         3       2400               1
         3       3450               1

13 linhas selecionadas.
```

A diferença no resultado da *query* será mais sobre a ordem dos dados.

O único valor que se repete no `select` é `2200`, dois registros no grupo `2` e um no grupo `3`. Todas as funções de agrupamento foram aplicadas no subgrupo. Observe que a ordem das colunas que informamos no `group by()` influência no retorno. Se alterarmos a ordem e informarmos `salario` e depois `setor_id`, o resultado será diferente. Determinamos que primeiro queremos agrupar por salário e em seguida, por setor:

```
SQL> select setor_id,count(id) as qtdFuncionarios from funcionarios group by setor_id,salario;

     SETOR    SALARIO QTDFUNCIONARIOS
---------- ---------- ---------------
         2       2000               1
         2       1800               1
         1       4200               1
         1       2600               1
         2       5300               1
         3       2100               1
         3       2400               1
         1       3000               1
         2       2200               2
         3       2200               1
         1       6200               1
     SETOR    SALARIO QTDFUNCIONARIOS
---------- ---------- ---------------
         1       1500               1
         3       3450               1

13 linhas selecionadas.
```

Quando executamos o `select` o banco de dados irá retornar o que for mais fácil. Mas se queremos definir uma ordem, podemos utilizar o `order by()` no fim da *query* e especificar que ele organize a partir do `setor_id` .

```
SQL> select setor_id as setor,salario,count(id) as qtdFuncionarios from funcionarios group by salari

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         1      1500               1
         1      2600               1
         1      3000               1
         1      4200               1
         1      6200               1
         2      1800               1
         2      2000               1
         2      2200               2
         2      5300               1
         3      2100               1
         3      2200               1

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         3      2400               1
         2      3450               1

13 linhas selecionadas.
```

E se quisermos ordenar pela quantidade de funcionários? Nós iremos passar para o `order by` que queremos ordenar pelo `count(id)` . Mais adiante veremos a diferença entre usarmos `count(id)` ou `count(*)` . Agora, testaremos a nossa *query*:

```
SQL> select setor_id as setor,salario,count(id) as qtdfuncionarios group by salario,setor_id order
```

```
     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         1      3000               1
         2      5300               1
         2      1800               1
         2      2000               1
         1      1500               1
         1      4200               1
         3      3450               1
         1      2600               1
         1      6200               1
         3      2100               1
         3      2200               1

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         3      2400               1
         2      2200               2

13 linhas selecionadas.
```

Observe que o Oracle ordenou primeiramente todos os registros com `1` funcionário e por último, o que tem `2`.

Com o `order by()` também temos a opção de usar o *alias* `qtdfuncionarios`.

```
SQL> select setor_id as setor,salario,count(id) as qtdfuncionarios group by salario,setor_id order b

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         1      3000               1
         2      5300               1
         2      1800               1
         2      2000               1
         1      1500               1
         1      4200               1
         3      3450               1
         1      2600               1
         1      6200               1
         3      2100               1
         3      2200               1

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         3      2400               1
         2      2200               2

13 linhas selecionadas.
```

O `order by()` também informa que podemos ordenar a partir da coluna. Vamos verificar com a coluna `3`:

```
SQL> select setor_id as setor,salario,count(id) as qtdfuncionarios group by salario,setor_id order b

     SETOR   SALARIO QTDFUNCIONARIOS
```

```
---------- --------- ---------------
         1      3000               1
         2      5300               1
         2      1800               1
         2      2000               1
         1      1500               1
         1      4200               1
         3      3450               1
         1      2600               1

         1      6200               1
         3      2100               1
         3      2200               1

     SETOR   SALARIO QTDFUNCIONARIOS
---------- --------- ---------------
         3      2400               1
         2      2200               2

13 linhas selecionadas.
```

Nesta primeira aula conhecemos um pouco mais sobre o `GROUP BY` e como funciona uma função de agrupamento. Por enquanto só vimos o `COUNT`, mas conheceremos outras a seguir.

Até a próxima aula!