

Subconsultas em SQL

Parte 2

- Cláusulas da Instrução Select
- Subconsulta na cláusula WHERE
 - O exemplo abaixo procura todos os empregados que executam a mesma função que o empregado SILVA (subconsulta simples)

```
select nome,funcao  
from empregados  
where funcao = (select funcao  
                from empregados  
                where nome = 'SILVA');
```

- Subconsulta na cláusula WHERE
 - A consulta abaixo procura os empregados que ganham um salário superior ao salário médio do respectivo departamento (subconsulta correlacionada):

```
select e1.codEmpregado, e1.nome, e1.salario, e1.codDepartamento
from empregados e1
where e1.salario > (select avg(e2.salario)
                    from empregados e2
                    where e2.codDepartamento = e1. codDepartamento);
```

- Subconsulta na cláusula Having
 - Uma subconsulta pode ser colocada na cláusula HAVING.
 - O exemplo abaixo determina os departamentos cujo salário médio é superior ao do departamento 30:

```
select codDepartamento, avg(salario)
from empregados
group by codDepartamento
having avg(salario > (select avg(salario)
                      from empregados
                      where codDepartamento = 30));
```

- Subconsulta na cláusula Having
 - Uma subconsulta pode ser colocada na cláusula HAVING.
 - O exemplo abaixo determina os departamentos cujo salário médio é superior ao do departamento 30:

```
select codDepartamento, avg(salario)
from empregados
group by codDepartamento
having avg(salario > (select avg(salario)
                      from empregados
                      where codDepartamento = 30));
```

- A consulta interna determina o salário médio do departamento 30;
- A consulta externa determina os salários médios de todos os departamentos mas só mostra os que têm valor superior ao devolvido pela subconsulta;

- Subconsulta na cláusula Having
- O exemplo abaixo determina a função com o salário médio mais elevado:

```
select funcao, avg(salario)
from empregados
group by funcao
having avg(salario) = (select max(avg(salario))
                      from empregados
                      group by funcao);
```

- A consulta interna começa por determinar o salário médio de cada função, indo depois determinar o maior salário de entre as médias;
- A consulta externa determina o nome da função e respectivo salário médio, só mostrando aqueles que igualam o valor devolvido pela subconsulta;

- Subconsulta na cláusula From

- Uma subconsulta pode ser colocada na cláusula FROM, o que é o mesmo que referir uma view (visão) ou uma tabela virtual.
- O exemplo abaixo determina os 5 empregados que têm o salário mais elevado:

```
select *  
from (select nome, salario  
      from empregados  
      order by salario desc)  
where rownum <=5;
```

- A subconsulta lista os salários por ordem decrescente, criando uma tabela virtual;
- A consulta externa lista a tabela virtual gerada anteriormente considerando apenas as 5 primeiras linhas

- Subconsulta na cláusula From

- A consulta abaixo lista os salários máximos da tabela EMPREGADOS e TERCEIRIZADOS usando apenas um query e mostrando os resultados numa linha única:

```
select
  SalarioEmp.sal1,
  SalarioTerc.sal2
from
  (select max(salario) SalarioEmp from empregados) sal1 ,
  (select max(remuneracao) SalarioTerc from terceirizados) sal2;
```


- Subconsulta na cláusula From

```
select
  SalarioEmp.sal1,
  SalarioTerc.sal2
from
  (select max(salario) SalarioEmp from empregados) sal1 ,
  (select max(remuneracao) SalarioTerc from terceirizados) sal2;
```

- Cada consulta interna devolve um valor que é considerado uma tabela virtual. Cada tabela virtual recebe um nome (SalarioEmp e SalarioTerc);
- O valor devolvido em cada consulta interna é colocado numa coluna com o nome sal1 e sal2;
- A consulta externa produz o produto cartesiano entre as tabelas virtuais. Cada uma tem uma linha, pelo que o resultado tem cardinalidade 1;

- Subconsulta na cláusula Select

- As subconsultas podem ser colocadas na cláusula SELECT.
- O exemplo abaixo pretende listar os salários máximos da tabela EMPREGADOS e TERCEIRIZADOS usando apenas um query e mostrando os resultados numa linha única:

```
select  
  (select max(salario) from empregados),  
  (select max(remuneracao) from terceirizados)  
from  
  dual;
```

- DUAL é uma tabela criada automaticamente pelo Oracle e contém apenas uma linha com o valor solicitado – tabela para executar algumas operações:

```
select sysdate from dual
```

- Subconsulta na cláusula Select

- O exemplo abaixo considera apenas os empregados do departamento 20 que executam a função 'CONTADOR'. Pretende mostrar o salário de cada um e compará-lo com a média da função e a média do departamento.

```
select
  nome,  salario,  funcao,
  (select round(avg(salario)) from empregados where job='CONTADOR') "Media_Salarial_Funcao",
  codDepartamento,
  (select round(avg(salario)) from empregados where codDepartamento=20) "Media_Salarial_Depto"
from empregados
where job='CONTADOR'
and codDepartamento=20;
```

- Subconsulta na cláusula Select

```
select
  nome, salario, funcao,
  (select round(avg(salario)) from empregados where job='CONTADOR') "Media_Salarial_Funcao",
  codDepartamento,
  (select round(avg(salario)) from empregados where codDepartamento=20) "Media_Salarial_Depto"
from empregados
where job='CONTADOR'
and codDepartamento=20;
```

- A primeira subconsulta determina a média de salários para a função enquanto a segunda determina a média do departamento;
- A consulta exterior usa os valores anteriores no meio dos valores de cada linha da tabela que satisfaz a condição de filtro;
- Estas subconsultas são simples (não correlacionadas);

- Subconsulta na cláusula Select

- Vamos generalizar o exemplo anterior para todos os empregados da empresa, o que nos obriga a considerar todas as funções e todos os departamentos. Vamos recorrer a subconsultas correlacionadas:

```
select
  e1.nome,
  e1.salario,
  e1.funcao,
  (select round(avg(e2.salario)) from empregados e2 where e2.funcao=e1.funcao)
  "Media_Salarial_Funcao",
  e1.codDepartamento,
  (select round(avg(e3.salario)) from empregados e3 where
e3.codDepartamento=e1.codDepartamento) "Media_Salarial_Departamento"
from empregados e1;
```

- Encadeamento Múltiplo

- Uma subconsulta pode receber outra, pelo que as subconsultas podem ser encadeadas umas dentro das outras até um limite de 255
- O exemplo abaixo determina o nome, função e data de admissão dos empregados cujo salário é superior ao salário mais elevado do departamento 'VENDAS'

[illegible]

- Encadeamento Múltiplo

```
select nome, funcao, dataContratacao, salario
from empregados
where salario > (select avg(salario)
                 from empregados
                 where codDepartamento = (select codDepartamento
                                           from departamento
                                           where nomeDepartamento = 'VENDAS'));
```

- A subconsulta que será feita em primeiro lugar é a que está mais abaixo e à direita e vai determinar o número do departamento que tem o nome 'VENDAS';
- Em seguida vamos determinar a média de salários desse departamento;
- Por último vamos determinar quais os empregados que ganham mais que essa média;

Funções em SQL

- LOWER(string): Converte 'string' para letras minúsculas:

```
select LOWER(nome), LOWER('Computação') from departamento;
```

- UPPER(string): Converte 'string' para letras maiúsculas.

```
select nome, UPPER(sobrenome) from empregados;
```

- INITCAP(string): Coloca letra maiúscula na primeira letra de cada palavra e minúsculas nas restantes

```
select INITCAP(nomeDepartamento) from departamento;
```

- **ROUND(valor, [n]):** Arredonda o valor. Se n for omitido arredonda para as unidades. Se n for positivo arredonda na posição n à direita da virgula. Se n for negativo arredonda na posição n à esquerda da virgula.

```
select ROUND(45.923,1), ROUND(salario/32,2)
      from empregados
      where codDepartamento = 10;
```

- **TRUNC(valor, [n]):** Trunca o valor. Se n for omitido trunca nas unidades. Se n for positivo trunca na posição n à direita da virgula. Se n for negativo trunca na posição n à esquerda da virgula, colocando os algarismos à direita de n e até à virgula com zero.

```
select TRUNC(45.923,1), TRUNC(45.323,-1),TRUNC(salario/32,2)
      from empregados
      where codDepartamento = 10;
```

- **SQRT(valor)**: Devolve a raiz quadrada de valor.
- **POWER(valor, n)**: Eleva à potência de n o valor. Pode ser negativo.
- **FLOOR(valor)**: Determina o inteiro menor ou igual que o valor introduzido como parâmetro.
- **CEIL(valor)**: Determina o menor inteiro **maior ou igual** que o valor introduzido como argumento.
- **SIGN(valor)**: Devolve -1 se for negativo, zero se for nulo, 1 se for positivo.

- **TO_CHAR(número, máscara de número):** A função TO_CHAR(número) converte do formato numérico para cadeia de caracteres, seguindo a máscara escolhida pelo usuário.
- A mascara '\$999,999.99' representa uma quantia monetária que possui parte decimal e separador de milhares

```
select to_char(salario,'$999,999.99')  
from empregados;
```

Formato	Significado	Exemplo	Resultado
9	Posição numérica - o número de 9s determina a largura de visualização	999999	1234
0	Zeros antes do número	099999	001234
\$	Símbolo de moeda americana	\$999999	\$1234
L	Símbolo de moeda que varia com a definição de local	L999999	?1234
.	Usar o ponto como separador das casas decimais	999999.99	1234.00
,	Usar a virgula como separador dos milhares (também conhecido por símbolo de grupo)	999,999	1,234
D	Usar um separador decimal que varia com a definição de local	999999.99	1234.00 ou 1234,00
G	Separador dos milhares que varia com a definição de local	999,999	1,234 ou 1.234
MI	Coloca sinal menos à direita nos valores negativos	999999MI	1234-
PR	Coloca números negativos entre parêntesis	999999PR	(1234)
EEEE	Notação científica. O número de 9s depois do ponto define a precisão	9.9999EEEE	1.2340E+03
V	Multiplicar por 10 ⁿ . n=número de 9s depois de V	9999V99	123400
B	Converte os nulos em espaços em branco	B9999.99	1234.00
FM	Remover os espaços em branco à esquerda e direita	L999999	?1234

- **TO_DATE(string, máscara de data):** A função TO_DATE() converte do formato cadeia de caracteres para data, seguindo a máscara escolhida pelo utilizador. Os valores possíveis para a mascara estão descritos a seguir.

```
select
```

```
    to_date('2006-07-17 12:15:45','yyyy-mm-dd hh24:mi:ss')+1  
    from dual;
```

- **NVL(v1,v2):** Se o valor de v1 for nulo, a função devolve v2.

```
select salario, comissao, salario*12+NVL(comissao,0)  
    from empregado;
```

FORMATO	SIGNIFICADO
SCC ou CC	Século; 'S' faz preceder de '-' as datas 'BC' (Antes de Cristo)
YYYY ou SYYYY	Século; 'S' faz preceder de '-' as datas 'BC' (Antes de Cristo)
YYY ou YY ou Y	Ultimos 3 digitos do ano ou 2 ultimos ou último digito do ano
Y,YYY	Ano com separador dos milhares nessa posição
SYEAR ou YEAR	Ano por extenso (em inglês); 'S' faz preceder de '-' as datas 'BC' (Antes de Cristo)
BC ou AD	Indicador de BC/AD (Antes de Cristo / Depois de Cristo)
B.C. ou A.D.	Indicador de BC/AD com ponto
Q	Trimestre do ano
MM	Mês em número de 1..12
MONTH	Nome do mês preenchido com espaços até 9 caracteres (Setembro)
MON	Nome do mês com três letras (JAN, FEB, MAY,...)
DDD ou DD ou D	dia do ano, mês ou semana
DAY	Nome do dia da semana, por extenso, preenchido com espaços até 9 caracteres
DY	Nome do dia da semana com dois caracteres
J	Dia do calendário Juliano; o número de dias desde 31 de Dezembro de 4713 A.C.
AM ou PM	Indicador de After Moon (AM) ou Post Moon (PM)
A.M.ou P.M.	Indicador de After Moon ou Post Moon com pontos
HH ou HH12	Hora do dia em formato 1-12
HH24	Hora do dia em formato 0-23
MI	Minutos
SS	Segundos

- **NVL(v1,v2):** Se o valor de v1 for nulo, a função devolve v2.

```
select salario, comissao, salario*12+NVL(comissao,0)  
      from empregado;
```

- **ROWNUM:** Devolve o número da linha dentro do resultado do query. O número é atribuído de acordo com a ordem em que as linhas são retiradas da tabela. Esta ordem é definida antes do ORDER BY, pois este é feito depois do "resultSet" estar criado.

- **ROWNUM:**

- Devolve o número da linha dentro do resultado do query.
- O número é atribuído de acordo com a ordem em que as linhas são retiradas da tabela.
- Esta ordem é definida antes do ORDER BY, pois este é feito depois do "resultSet" estar criado.

`select rownum,funcao,nome from empregados;`

- O valor da coluna ROWNUM é atribuído à medida que as linhas são extraídas da tabela, e isto é feito antes da ordenação. O rownum é independente da sequência de ordenação;

- **ROWNUM:**
- A função ROWNUM conjugada com uma sub consulta pode ser usada para resolver o seguinte problema:

Quais são os 3 empregados com o maior salário?

```
select *  
from (select nome, salario from empregados order by salario desc)  
where rownum <= 3;
```

- **DEFINE:**
- Serve para definir uma variável de substituição e atribuir-lhe um valor.
- A variável pode ser utilizada numa instrução SELECT tanto no modo interativo como numa lista de comandos.
- O conteúdo da variável pode ser apagado com o comando UNDEFINE

```
DEFINE remuneracaoAnual='salario*12'  
SELECT nome,funcao,&remuneracaoAnual  
FROM empregados  
ORDER BY &remuneracaoAnual;
```

Exercícios

1. Determine qual o departamento com o MAIOR encargo anual de remunerações (12 vezes o salário mais a comissão).
2. Em que ano entraram mais pessoas para a empresa? Visualize o ano e número de empregados
3. Modifique a questão 2 para apresentar também o valor do salário médio para o departamento
4. Escreva uma consulta para visualizar um '*' ao lado da linha do empregado mais recente. Visualize NOME, DATACONTRATAÇÃO e a coluna (maxdate) com o '*'
5. Quem são os 3 empregados mais bem pagos da companhia ?