

AULA DE LABORATÓRIO NR. 4 – SUBCONSULTAS

Objetivo: Ensinar a utilização de subconsultas na cláusula WHERE (consultas aninhadas).

Pré-requisito importante: Saber utilizar ALIAS (apelidos).

TEORIA:

A linguagem SQL permite que se coloque uma consulta em outra consulta. Dessa forma, podemos ter várias consultas aninhadas. Um exemplo de consulta aninhada foi visto na aula de laboratório NR 3. Veja uma consulta sem aninhamento no **Exemplo 1**:

Exemplo 1:

Consulta **sem** aninhamento:

```
select d.deptno, d.dname, sum(sal)
from emp e, dept d
where e.deptno=d.deptno
group by d.deptno, d.dname
having max(sal) < 5000
```

A consulta do **Exemplo 1** retorna a soma total dos salários de todos os departamentos em que o maior salário não chega a 5000,00.

A mesma consulta pode ser feita **usando aninhamento**:

Exemplo 2:

```
select d.deptno, d.dname, sum(sal)
from emp e, dept d
where d.deptno=e.deptno
and
5000 > (select max(sal) from emp i
       where i.deptno = d.deptno
       group by deptno)
group by d.deptno, d.dname
```

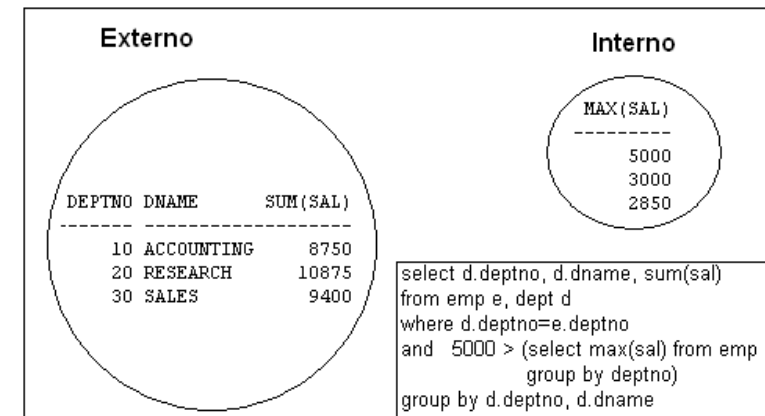
Apesar da solução **usando aninhamento** ser mais complexa, novas possibilidades surgem. Trocamos *simplicidade* por *poder*. No exemplo anterior, a subconsulta está sombreada. Verifique que ela fica na cláusula WHERE da consulta principal (chamada de consulta externa). Suponha agora que devemos retornar todos os nomes de funcionário, seu salário e nome do departamento em que trabalha. Além disso, o funcionário deverá ganhar mais que a média salarial da empresa. Nesse caso, não é possível uma solução usando **having**. Portanto, devemos aprender a usar consultas aninhadas para resolver esse tipo de questão.

Voltando ao **Exemplo 2**, vamos dividir a consulta em suas partes: a parte **externa** e a parte **interna** (sombreada). Dessa forma o entendimento de como o aninhamento

funciona será mais fácil. Além dessa divisão, vamos retirar a linha **where i.deptno = d.deptno** da consulta interna. No futuro iremos acrescentá-la novamente. Feita a divisão, ficaremos com as seguintes consultas:

EXTERNA	INTERNA
<pre>select d.deptno, d.dname, sum(sal) from emp e, dept d where d.deptno=e.deptno group by d.deptno, d.dname</pre>	<pre>Select max(sal) from emp Group by deptno</pre>

Os conjuntos resultantes das consultas são mostrados nos diagramas abaixo:



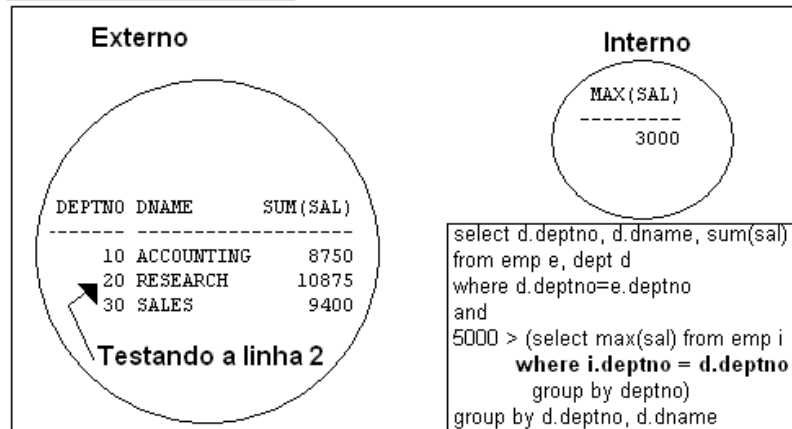
Temos agora dois conjuntos e agora vamos entender como o SQL funciona: Para cada linha do conjunto EXTERNO, ele faz o teste da cláusula WHERE (onde está a subconsulta) com o conjunto INTERNO. No **Exemplo 2**, o teste é verificar se 5000 é maior que o valor retornado para cada linha do conjunto INTERNO. Se o teste for verdadeiro, a linha testada do conjunto EXTERNO é retornada, caso contrário, ela não é retornada.

Vamos imaginar que ele já testou a linha 1 e agora está testando a linha 2 (20 RESEARCH 10875). O teste será realizado três vezes (uma teste para cada linha do conjunto INTERNO). Sendo assim os testes realizados serão:

```
5000 > 5000 ? (falso)
5000 > 3000 ? (verdade)
5000 > 2850 ? (verdade)
```

Como nos testes acima, temos dois testes verdadeiros, o resultado final irá mostrar 20 RESEARCH 10875 **DUAS VEZES!**. Isso aconteceu porque retiramos da consulta interna a linha **where i.deptno = d.deptno**. Essa linha faz com que o conjunto

INTERNO retorne apenas uma linha, ao invés de três. A linha retornada será exatamente o maior salário do departamento que se está testando no momento. Como estamos pegando como exemplo a linha 20 RESEARCH 10875, no momento que o SQL estiver testando essa linha, teremos os conjuntos abaixo, se a linha **where i.deptno = d.deptno.** for acrescentada:



Sendo assim, teremos apenas um teste que será verdadeiro e a linha 2 será retornada apenas uma vez, ao invés de duas. Dando assim, o resultado correto.

Perguntas:

- + Qual a linha do conjunto interno quando o SQL estiver testando a linha 3 do conjunto externo?
- + A cláusula **where i.deptno = d.deptno** afeta o resultado para a linha 1? Por quê?

Observe a importância do ALIAS. Só conseguimos acrescentar a cláusula **where i.deptno = d.deptno** utilizando uma ALIAS para DEPT na consulta EXTERNA e um ALIAS para EMP na consulta INTERNA.

No nosso exemplo, utilizamos um teste em que o operador era maior que (>). Nas consultas aninhadas, podemos ter ainda os seguintes operadores, além dos comuns:

OPER	EXEMPLO DE USO	EXPLICAÇÃO
IN	5000 IN (select Max(sal) from emp)	5000 é um valor QUE ESTÁ no conjunto interno?
NOT IN	5000 NOT IN (select Max(sal) from emp)	5000 é um valor QUE NÃO ESTÁ no conjunto interno?
EXISTS	5000 EXISTS (select Max(sal) from emp)	5000 é um valor QUE EXISTE no conjunto interno?
NOT EXISTS	5000 NOT EXISTS (select Max(sal) from emp)	5000 é um valor QUE NÃO EXISTE no conjunto interno?
ALL	5000 > ALL (select Max(sal) from emp)	5000 é MAIOR QUE TODOS os valores do conjunto interno?
ANY	5000 > ANY (select Max(sal) from emp)	5000 é MAIOR QUE QUALQUER um dos valores do conjunto interno?

EXERCÍCIOS PRÁTICOS

1. Retorne o NOME, SALARIO E NUMERO DO DEPARTAMENTO dos funcionários que ganham mais que a média salarial da empresa.
2. Retorne o NOME dos departamentos que possuem mais de 4 funcionários. (use consultas aninhadas);
3. Retorne o NOME, SALARIO E NOME DO DEPARTAMENTO dos funcionários que ganham mais que QUALQUER um dos funcionários do departamento 20.
4. Retorne o NOME, SALARIO E NOME DO DEPARTAMENTO dos funcionários que ganham mais que a média de QUALQUER um dos departamentos da empresa.
5. Retorne o NOME, SALARIO E NOME DO DEPARTAMENTO dos funcionários que não trabalham no departamento 10 (use consultas aninhadas).
6. Selecione o NOME DO FUNCIONARIO e NOME DO DEPARTAMENTO dos funcionários que trabalham em um departamento que possui mais de 4 funcionários lotados.
7. Selecione os funcionários que gerenciam mais de dois funcionários;
8. Selecione os funcionários que não gerenciam ninguém;
9. **Desafio 1:** Selecione os funcionários que ganham mais que a média de seus departamentos e que não gerenciam ninguém.
10. **Desafio 2:** Selecione os funcionários gerentes que ganham mais que a média salarial da empresa.