

Implementação de banco de dados

Aula 08: Subconsulta e Operadores de Conjunto

Apresentação

Muitas vezes, em uma consulta, necessitamos filtrar as linhas por valores que não existem no banco de dados. Por exemplo: selecionar os empregados cujo salário seja maior que o salário médio de todos os empregados.

Como fazer isso? Como obter a média dos valores segurados e utilizá-la para filtrar as linhas de uma tabela?

Para isso existem as subconsultas, que serão objetos de estudos nesta nossa aula, junto com os operadores de conjunto.

Objetivos

- Elaborar comandos utilizando subconsultas e operadores de conjunto.

Banco de dados de exemplo

Nesta aula, continuaremos utilizando o banco de dados da empresa para os exemplos.

Modelo lógico:

As tabelas têm os seguintes dados:

[Região](#)



[Departamento](#)



[Empregado](#)



[Cliente](#)



Tendo este banco em mente, é extremante recomendável que você execute os comandos de exemplo no PostGreSQL.

Comentário

Foi escolhido como base o PostgreSQL por ser um SGBD mais leve e fácil de instalar, porém você pode usar o SQLServer ou o Oracle. Quando houver diferença entre os SGBDs, você receberá um alerta.

Subconsultas

O que são subconsultas?

São consultas secundárias dentro de outras consultas, utilizadas para gerar valores intermediários necessários para executar a consulta principal. As subconsultas são muito utilizadas em comandos de select.

Veja o seguinte exemplo:

Comentário

Você deseja recuperar todos os dados dos empregados que trabalham no departamento de vendas. Para isso, necessitaríamos acessar os dados das tabelas Empregado e Departamento.

Neste caso, o comando seria:

```
SELECT *  
  
FROM EMPREGADO  
  
WHERE ID_DEPTO = ( SELECT ID FROM DEPARTAMENTO  
  
WHERE UPPER (NOME)= 'VENDAS')
```

Observe que existe um comando select (consulta inicial) e um outro comando select dentro deste primeiro (interno). Sua execução gera o resultado desta figura:

Modo de execução

Conforme vimos no exemplo anterior, o segundo select é executado primeiramente, gerando a informação que será utilizada pelo select principal. Acompanhe:

O valor retornado (20) é então utilizado para a consulta principal e esta é executada como se tivesse sido escrita como:

Outro exemplo

Uma subconsulta pode também ser utilizada para retornar linhas que atendam a condições que não existem diretamente nos dados de uma tabela.

Por exemplo: desejamos retornar todos os dados dos empregados que ganham mais que o salário médio da empresa. Não existe em nenhuma tabela o valor do salário médio, mas ele pode ser calculado em uma consulta utilizando-se a função AVG:

De posse da média, você poderia obter os empregados que atendem a condição.

O problema desta solução é que você teria emitido dois comandos:

- O primeiro para obter a média.

```
SELECT AVG(SALARIO)
```

```
FROM EMPREGADO
```

- O segundo utilizando o valor gerado pelo primeiro para obter os dados dos empregados.

```
SELECT *
```

```
FROM EMPREGADO
```

```
WHERE SALARIO > 13750
```

Isso poderia ser feito escrevendo uma subconsulta.

Mais um exemplo

Você deseja retornar o identificador do departamento e o valor salarial médio do departamento cujo salário médio seja maior ou igual ao salário médio da empresa. Veja o seguinte comando, que retorna o ID do departamento e o seu salário médio:

Como você já viu, o salário médio da empresa é 13750; portanto, os departamentos 30 e 10 atendem à condição. Neste caso, a subconsulta deve estar na cláusula having, já que se deseja filtrar os grupos, e não as linhas da tabela.

Analise agora o comando a seguir. O que mudou em relação ao anterior?

Comentário

Note a subconsulta na cláusula select. Ela é executada uma vez para cada linha que retorna e, no caso, serviu para listar ao lado dos dados do departamento a média de todos os veículos.

Saiba mais

Utilizando o nosso banco de dados de exemplo, vamos fazer alguns [exercícios](#).

Retornando mais de uma coluna

Uma subconsulta pode retornar mais de uma coluna. O maior uso dessa situação é criar uma tabela em memória para ser utilizada, por exemplo, em uma junção.

Suponha que você deseja retornar o nome do departamento e o valor de sua folha salarial (soma dos salários de seus empregados) para departamentos cuja média salarial seja maior ou igual à média da empresa.

Obviamente a consulta poderia ser construída realizando primeiro a junção das tabelas Empregado e Departamento e gerando os grupos depois, conforme este comando:

Para exemplificar o uso do retorno da tabela em memória, o comando pode ser construído executando primeiro uma subconsulta e, a seguir, utilizando seu retorno como uma tabela com a qual se fará a junção com a tabela Departamento.

Primeiro, como seria a subconsulta?

Esta subconsulta deve ser escrita na cláusula from entre parênteses, atribuir-lhe um apelido e fazer a junção com a tabela Modelo:

Observe no comando o seguinte:

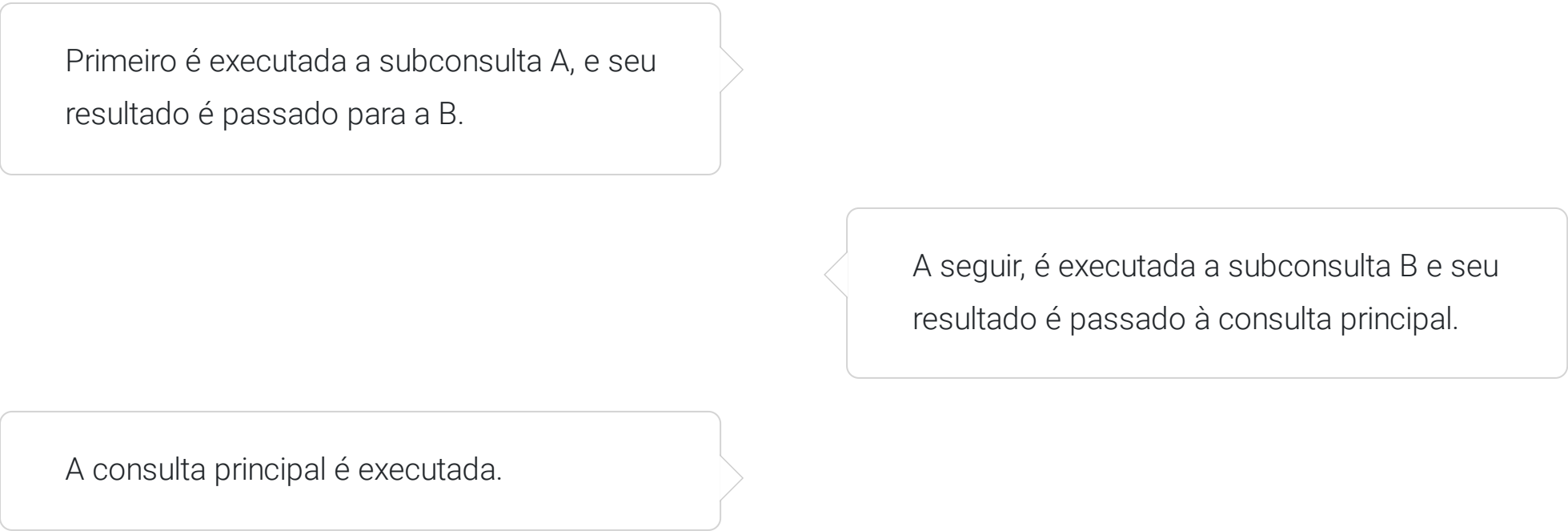
_____ 1 _____

A subconsulta da cláusula from foi apelidada de T, e pudemos referenciar suas colunas ID_DEPTO, Folha e Média na consulta exterior.

_____ 2 _____

Temos uma subconsulta aninhada dentro de outra na cláusula from.

Neste caso, a subconsulta A é interna à B, que por sua vez é interna à PRINCIPAL. A execução ocorre, então, na seguinte ordem:



Subconsultas simples

São aquelas que retornam apenas uma linha, como acontece, por exemplo, ao se utilizar uma função de grupo sem a cláusula group by, como no caso da subconsulta A que vimos há pouco.

Neste tipo de subconsulta, podem ser utilizados os seguintes operadores na comparação:

Operador	Descrição
=	Igual a
< >	Diferente
>	Maior que
<	Menor que
> =	Maior ou igual
< =	Menor ou igual

Subconsultas multilinhas

São aquelas que retornam mais de uma linha, como a subconsulta B do último exemplo. Quando a subconsulta retorna mais de uma linha, temos que utilizar na comparação in ou not in, pois somente estes operadores conseguem lidar com um conjunto de valores.

Para maior clareza, veja os seguintes exemplos:

1. A tabela Departamentos tem o seguinte conteúdo:

Temos dois departamentos da região 1 e um da região 2. Se todos os IDs, ult_nome e id_depto dos empregados do departamento da região 2 utilizarem subconsulta, o comando será:

Se você quisesse listar os empregados dos departamentos da região 1, o que ocorreria?

Ocorreria um erro; a subconsulta no caso retorna dois valores, já que existem dois departamentos da região 1.

Deve ser utilizado, então, o operador **in**.

2. Considere, agora, que você deseja listar todos os dados dos empregados que atendem a algum cliente.

Logicamente, via uma junção interior isso poderia ser resolvido, mas também poderia ser feito via subconsulta. Para isto, devemos listar os valores dos vendedores da tabela Clientes e, a seguir, ver quais empregados estão nesta lista.

A subconsulta seria:

Observe que retorna um vendedor nulo. O comando seria então:

Retornaram dois empregados. Como temos seis empregados, isso significa que quatro deles não atendem a clientes. Para encontrá-los, podemos fazer uma junção exterior ou via subconsulta. Substituir o **in** por **not in**, ou seja, o empregado cujo código não está na lista.

O que aconteceu? Deveriam voltar quatro empregados, mas não voltou nenhum.

Como já destacamos, existe um nulo no retorno da subconsulta. Quando se usa in, não ocorre nenhum problema, mas quando é utilizado not in, a comparação com nulo resulta nulo.

Atenção

Lembre-se de que a cláusula where faz com que retornem apenas as linhas cujo resultado do teste seja verdadeiro. Desta forma, **nunca** permita o retorno de linha nula em uma subconsulta.

O comando da subconsulta deveria, então, ser:

Repare que a linha nula sumiu... O comando a ser executado seria então:

Esse comando retorna os dados dos empregados que não atendem a nenhum cliente.

Subconsultas correlatas

Nos exemplos feitos até agora, a subconsulta era executada uma vez, e retornavam os valores para serem utilizados pela consulta principal. Essas são chamadas de subconsultas aninhadas.

Existe outro tipo de subconsulta denominada correlata, que para poder ser executada, necessita de valores da consulta principal. Por isso, deve ser executada uma vez para cada linha da consulta externa.

Exemplo

Suponha que você deseje saber todos dados dos veículos cujo valor segurado seja menor que o valor segurado médio de seu modelo. Inicialmente, pergunte-se como seria gerado o salário médio de um departamento.

Para o Departamento 30, o comando seria:

Então, para retornarem os dados dos empregados do Departamento 30 cujo salário é menor que a média do departamento, basta comandar:

Para o Departamento 30, está ok, mas e para os outros departamentos? Afinal, cada um tem uma média diferente.

O necessário para resolver a situação é que:

1
A subconsulta acesse na consulta principal qual o departamento do empregado cuja linha está sendo processada naquele momento.

2
Calcule a média do valor para aquele departamento.

3
Retorne a média para a consulta principal.

A consulta principal compare a média com o salário do empregado da linha que está sendo processada.

Retorne os dados do empregado se o valor segurado for menor que a média.

Para obter a situação descrita no comando, a subconsulta deve referenciar dados da consulta principal. Para isto, você deve dar um apelido para a tabela da consulta principal e outro para a da subconsulta, e nesta, “chamar” os dados da tabela principal, conforme mostra esta imagem:

Observe no comando que a tabela externa foi apelidada de E a interna de I; no where foi utilizado o valor da externa para filtrar as linhas da subconsulta.

Princípios de subconsulta

Em resumo, as subconsultas obedecem aos seguintes princípios:

1

São escritas entre parênteses.

2

Podem existir várias subconsulta para uma única consulta.

3

Podem existir subconsultas dentro de subconsultas (subconsultas aninhadas).

4

Podem retornar uma ou várias colunas.

5

Podem retornar uma ou várias linhas.

6

Normalmente são utilizadas nas cláusulas where ou having, podendo, mais raramente, serem utilizadas nas cláusulas from ou select.

7

Eventualmente podem referenciar colunas da consulta principal (subconsulta correlata).

8

Não devem retornar linhas nulas.

Utilizando operadores de conjunto

Algumas vezes, é necessário fazer a chamada **junção vertical**, que nada mais é que retornar em uma mesma coluna dados de mais de uma tabela. Isto pode ser feito utilizando os operadores de conjunto union, intersect e except.

Toda tabela pode ser considerada um conjunto de valores, e os operadores de conjunto nos permitem realizar as operações de união (union), interseção (intersect) ou diferença (except).

UNION

INTERSECT

EXCEPT

Da mesma forma que fizemos quando utilizamos uma subconsulta na cláusula from, gerando uma tabela, os conjuntos a serem operados são gerados por subconsultas.

Para que o comando funcione, é necessário que as tabelas geradas sejam compatíveis. Mas o que significam tabelas compatíveis?

Duas ou mais tabelas são compatíveis para fazer operações de conjunto se tiverem o mesmo esquema, ou seja, o mesmo número de colunas, com o mesmo tipo, na mesma ordem.

Observe a figura:

As tabelas Empregados e Produtos são compatíveis, já que têm três colunas: a primeira e a segunda numéricas e a terceira alfanumérica. Já a tabela Animais não é compatível com as outras duas, já que, embora também tenha três colunas, sendo duas numéricas e um alfanumérica, elas não se encontram na mesma ordem.

Por definição, conforme você deve ter aprendido na matemática, as operações de conjunto eliminam valores duplicados. Isto quer dizer que, se duas tabelas tiverem linhas iguais, apenas uma linha irá retornar no resultado.

Observe a figura:

Ao comparar a primeira linha da tabela Empregado com cada uma das linhas da tabela Acionista, observa-se que não existe nenhuma linha igual na tabela de Acionista, pois apesar de existir um acionista com o mesmo ID (primeira linha, ID 101), o nome é diferente. Portanto, estas linhas não são iguais.

O mesmo ocorre em relação à terceira linha de Empregado, pois a segunda linha de acionista tem o mesmo valor para nome (Marco Aurélio), porém com valor de ID diferente.

Já a segunda linha da tabela Empregado é exatamente igual à última linha de Acionista, já que elas têm o mesmo nome e ID. Caso as duas fizessem parte do conjunto Resposta de uma operação de conjunto, apenas uma delas apareceria no resultado.

O Operador Union

Realiza a operação de união, ou seja, retorna todas as linhas do primeiro conjunto acrescidas de todas as linhas do segundo, eliminando as duplicadas.

Se quisermos, por exemplo, o ID dos empregados que trabalham nos departamentos 20 ou 30 ou que tenham o cargo de vendedor, como ficaria o comando utilizando operadores de conjunto?

Observe os dados da tabela:

Note que os empregados de ID 2 a 6 são dos departamentos 20 ou 30 (destacados em azul) e os de ID 5 e 6 são vendedores (destacados em vermelho). Queremos os que estão em um conjunto ou no outro, portanto os empregados de ID 2, 3, 4, 5 e 6. Note ainda que os empregados de ID 5 e 6 aparecem nos dois conjuntos. O comando que resolve o problema é:

Note que:

1. Retornaram os IDs 2,3 e 4, apesar de não serem vendedores.
2. Os empregados de IDs 5 e 6 retornaram apenas uma vez, apesar de estarem nos dois conjuntos.

Existe uma variação no SQL do comando de união que retorna os duplicados; é o **union all**.

Agora os IDs 5 e 6 aparecem duas vezes.

Se você desejasse os IDs dos empregados que trabalham nos departamentos 20 ou 30 e que tenham o cargo de vendedor, ou seja, que aparecem nos dois conjuntos, deveria utilizar **intersect**.

Essas duas operações apresentadas (união e interseção) são comutativas, ou seja, tanto faz a primeira subconsulta ser pelo modelo e a segunda pela cor ou o inverso; o resultado será igual. Teste e comprove.

O mesmo não acontece na diferença (except). Neste caso, se eu inverter a ordem, o resultado mudará.

Para se retornar os IDs dos empregados que trabalham nos departamentos 20 ou 30 e que **não tenham** o cargo de vendedor, ou seja, que estão no primeiro conjunto e não estão no segundo, seria necessário comandar:

Coisas do Oracle

Todos os comandos que vimos até agora, sejam de subconsulta ou operadores de conjunto, funcionam de forma igual no PostgreSQL, SQLServer e Oracle, com exceção do except. No Oracle, o operador de diferença é o **minus**.

Finalmente, uma última observação: nos operadores de conjunto, o order by somente pode aparecer ao final do comando, e não no meio.

Saiba mais

Resumo das regras para utilização dos operadores de conjuntos

- Os comandos select participantes têm que ter o mesmo número de colunas, ou seja, devem ser compatíveis;
- As colunas correspondentes têm que ser do mesmo tipo de dado;
- Linhas duplicadas são automaticamente descartadas; a exceção é o operador union all;
- A cláusula order by só pode ser utilizada ao final do comando;
- Os operadores de conjuntos podem ser utilizados em subconsultas.

Atividade

Para as atividades abaixo, utilize o banco de dados da escola.

Modelo de dados da escola

Modelo Conceitual

Modelo lógico

Questão 1

Dê os comandos de subconsulta solicitados:

1. Quantos registros existem na tabela Cursou com media maior que a média dos alunos reprovados por nota?

- 2. Quantos alunos diferentes foram aprovados com média superior à média dos alunos aprovados?
- 3. Quantos alunos cursaram apenas uma disciplina do curso de Sistemas de Informações?
- 4. Qual a sigla do curso que teve menos disciplinas cursadas?
- 5. Qual o nome do curso que tem a média da nota de aprovação dos alunos mais alta?
- 6. Qual o código da disciplina que foi cursada por mais alunos?
- 7. Qual a maior matrícula de um aluno que cursou nove disciplinas do curso de código 101?
- 8. Qual a maior quantidade de disciplinas cursadas por um aluno no curso de Redes de Computadores?
- 9. Quantos alunos diferentes foram reprovados por nota em uma disciplina com média superior à média de reprovação por nota da disciplina?

Questão 2

Dê os comandos solicitados utilizando operadores de conjunto.

- 1. Qual o nome do curso que não tem disciplina com carga horária de 40 horas?
- 2. Quantos alunos cursaram apenas disciplinas do curso de Banco de Dados?
- 3. Quantos alunos cursaram disciplinas do curso de Redes e não cursaram disciplinas do curso de Banco de Dados?
- 4. Quantos alunos cursaram disciplinas de todos os cursos?

Questão 3

Transforme os comandos de junção de forma que utilizem subconsultas.

- 1. Retorne a matrícula e o nome do aluno e a quantidade de vezes que cursou disciplinas.
- 2. Retorne quantos alunos diferentes foram reprovados por falta em disciplinas de carga horária de 40 horas.
- 3. Retorne de quantos cursos diferentes o aluno de CPF 62551881242 cursou disciplinas.
- 4. Retorne todos os dados do aluno que não cursou nenhuma disciplina.

Notas

Título modal ¹

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Título modal ¹

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos. Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos.

Referências

DATE, C. J. **Introdução a sistemas de banco de dados**. 7. ed. Rio de Janeiro: Campus, 2000.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7. ed. São Paulo: Pearson Addison Wesley, 2015.

Próxima aula

- Linguagem SQL - Transação

Explore mais

Veja:

- [T-SQL e a utilização de subconsultas correlacionadas](#)
- [Como fazer subconsultas: um passo a passo](#)
- [PostgreSQL Prático/DML/Subconsultas](#)
- [Expressões de subconsulta](#)
- [Entendendo Except, Intersect e Union do SQL](#)
- [Combinação de consultas](#)
- [SQL: utilizando o Operador Union e Union All](#)