

Subconsultas

- um método adicional para tratar consultas usando múltiplas tabelas
- pode ser usado
 - WHERE
 - HAVING
 - INSERT, UPDATE, DELETE
 - qualquer número de aninhamentos é permitido
- predicados
 - IN
 - <operador-comparador>
 - <operador-comparador> ANY
 - <operador-comparador> ALL
 - EXISTS
 - todos predicados podem ser negados com NOT

Subconsultas

- Alguns Usos
 - atualização de tuplas cujo critério de seleção deva ser expresso usando mais de uma tabela
 - suprima os atores que participaram de 'E o vento levou'
 - para implementar a interseção e diferença (não são padrão SQL92)
 - os atores que participaram de 'E o vento levou' e jamais atuaram com Silvester Stalone
 - os atores que participaram de 'E o vento levou' e que atuaram pelo menos uma vez com Silvester Stalone
 - para expressar uma condição em função de um grupo de tuplas
 - a atriz que ganha cachês acima da média de cachês
 - o filme com a maior folha de pagamento

Subconsultas

- funcionamento
- consultas externa e interna
 - subconsulta não-relacionada
 - processamento da consulta interna não precisa de dados vindos da consulta externa
 - predicados IN, operador de comparação
 - subconsulta relacionada
 - processamento da consulta interna precisa de dados vindos da consulta externa
 - todos os tipos de predicado
- consultas internas retornam
 - um único valor
 - externa: operador de comparação, IN
 - lista de valores
 - externa: IN, operador de comparação + modificador (ANY, ALL)
 - conjunto (possivelmente vazio) de tuplas
 - externa: EXISTS

Predicado IN

- testa se um atributo possui um valor pertencente a um conjunto de valores
 - where pais IN ('BRASIL', 'CHILE', 'ARGENTINA')
 - where pais = 'BRASIL' or pais = 'ARGENTINA' or pais = 'CHILE'
- valor pode ser o resultado de uma subconsulta

```
where pais in (select pais
               from pais
               where continente = 'America' )
```
- normalmente, subconsulta deve projetar um único atributo
- alguns SGBDs aceitam múltiplos atributos

```
where (cidade, pais) in (select cidade, pais
                        from pais
                        where continente = 'America' )
```

Subconsultas

- uso em operações de atualização onde as condições de alteração, remoção ou inserção são expressas em termos de atributos existentes em outras tabelas
- Exemplos
- suprimir os pilotos da Varig

```
delete from piloto
where companhia = 'Varig'
```
- suprimir os pilotos que voam de MD11

```
delete from piloto
where codp in (select distinct codp
               from escala
               where aviao = 'MD11')
```

Subconsultas

- Exemplos (cont.)
- aumentar em 10% a gratificação dos funcionários da Varig que voam ao aeroporto de código Gru

```
update piloto
set gratificacoes = gratificacoes*1.1
where companhia = 'Varig' and
      codp in (select distinct codp
               from escala, voo
               where escala.codv = voo.codv and
                  dest = 'gru')
```
- observações:
 - deve-se procurar expressar na consulta externa todas as restrições possíveis

Subconsultas

- Comparação com valores calculados ou encontrados em outras tuplas
- O maior salário pago pela Varig

```
select max(salario)
from piloto
where companhia = 'Varig'
```
- O nome do piloto que ganham pelo menos o maior salário pago pela Varig

```
select nomep
from piloto
where salario >= (select max(salario)
                  from piloto
                  where companhia = 'Varig')
```

Subconsultas

- Qualquer número de aninhamento de consultas
- Consulta Externa não conhece os atributos/tabelas de sua(s) subconsulta(s), apenas o seu resultado
 - não pode fazer referência a atributos/tabelas da(s) subconsulta(s)
- Consulta Interna conhece atributos/tabelas de sua(s) consulta(s) externa(s) (direta ou indireta)
- esquema de nomeação na consulta interna
 - tenta interpretar nome de tabela/atributo localmente
 - caso não case com atributo/tabela da consulta, procura recursivamente na consulta externa

Subconsultas

- O nome do piloto que ganham pelo menos o maior salário pago pela Varig

```
select nomep
from piloto
where salario >= (select max(salario)
                  from piloto
                  where companhia = 'Varig')
```

equivale a

```
select pext.nomep
from piloto pext
where pext.salario =
      (select max(pint.salario)
       from piloto pint
       where pint.companhia = 'Varig')
```

Subconsultas

- O nome do piloto que ganha o maior salário pago por sua companhia

```
select nomep
from piloto pext
where salario=
      (select max(salario)
       from piloto
       where companhia = pext.companhia)
```

equivale a

```
select nomep
from piloto pext
where pext.salario =
      (select max(pint.salario)
       from piloto pint
       where pint.companhia = pext.companhia)
```

Subconsultas

- Operações de Interseção e Diferença
- Exemplo: Interseção
- O código dos pilotos que voam de MD11 e de Boeing

```
select codp
from escala
where aviao = 'MD11' and aviao = 'Boeing'
```

 - não retorna nada: um mesmo atributo jamais tem dois valores simultaneamente!!
 - Para encontrar a resposta, é necessário procurar em várias tuplas!

Subconsultas

- Operações de Interseção e Diferença
- Exemplo: Interseção
- O código dos pilotos que voam de MD11 e de Boeing

```
select codp
from escala
where aviao = 'MD11' and
      codp IN (select distinct codp
               from escala
               where aviao = 'Boeing')
```

Subconsultas

- Operações de Interseção e Diferença
- Exemplo: Diferença
- O nome dos pilotos que não voam de MD11

```
select nomep
from escala, piloto
where escala.codp = piloto.codp and aviao <> 'MD11'
```

 - retorna as tuplas nas quais a escala não foi de MD11
 - não garante que em outra escala daquele piloto o avião usado não foi MD11
 - Para encontrar a resposta, é necessário procurar em várias tuplas!

Subconsultas

- Operações de Interseção e Diferença
- Exemplo: Diferença
- O nome dos pilotos que não voam de MD11

```
select nomep
from piloto
where codp NOT IN (select distinct codp
from escala
where aviao = 'MD11')
```

Operador de Comparação Modificadores ANY, ALL

- operador de comparação: garantir que consulta retorne UM único valor
- modificadores: valores múltiplos
 - ANY: algum valor
 - ALL: todos os valores

> ALL(1, 2, 3)	>3
< ALL(1, 2, 3)	<1
= ALL(1, 2, 3)	igual a todos os valores (ao mesmo tempo)
> ANY(1, 2, 3)	>1
< ANY(1, 2, 3)	<3
= ANY(1, 2, 3)	= 1 or =2 or =3 (IN)

Predicado EXISTS

- impõe sempre o processamento **relacionado** de subconsultas
- avalia verdadeiro se a subconsulta retorna uma tabela com no mínimo uma tupla
- NOT EXISTS avalia verdadeiro se resultado da consulta é uma relação vazia (zero tuplas)
- usos aconselhados
 - consultas cujo resultado é dado pela interseção ou diferença entre relações (dar preferência a IN quando possível)
 - consultas do tipo TODOS ou NENHUM

EXISTS: exemplo

- O nome dos pilotos que voam de MD11 e de 737

```
select nomep
from piloto p, escala e
where p.codp = e.codp and aviao = 'MD11' and
p.codp IN (select distinct codp
from escala
where aviao = '737')
```
- observações
 - interseção entre os pilotos que voam de 737 e os que voam de MD11
 - pode (e deve) ser resolvido com o uso de IN

EXISTS: exemplo

- O nome dos pilotos que voam de MD11 e de 737

```
select nomep
from piloto p, escala e
where p.codp = e.codp and aviao = 'MD11' and
EXISTS (select *
from escala
where aviao = '737' and
codp = p.codp)
```
- observações
 - interseção entre os pilotos que voam de 737 e os que voam de MD11
 - alternativa ao uso de IN (deve ser evitado !!!)
 - tanto faz o resultado projetado na subconsulta: o importante é que exista um resultado com **pelo menos 1 tupla**
 - existem condições que relacionam consulta EXTERNA e INTERNA (**codp = p.codp**)

EXISTS: exemplo

- O nome dos pilotos que voam de MD11 e não voam de 737

```
select nomep
from piloto p, escala e
where p.codp = e.codp and aviao = 'MD11' and
p.codp NOT IN ( select distinct codp
from escala
where aviao = '737')
```

- observações
 - diferença entre os pilotos que voam de MD11 e os que voam de MD11
 - pode (e deve) ser resolvido com o uso de NOT IN

EXISTS: exemplo

- O nome dos pilotos que voam de MD11 e não voam de 737

```
select nomep
from piloto p, escala e
where p.codp = e.codp and aviao = 'MD11' and
NOT EXISTS (select *
from escala
where aviao = '737' and
codp = p.codp)
```

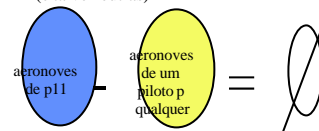
- observações
 - diferença entre os pilotos que voam de MD11 e os que voam de MD11
 - alternativa ao uso de NOT IN (deve ser evitado !!!)
 - tanto faz o resultado projetado na subconsulta: o importante é que exista um resultado com **pelo menos 1 tupla**
 - existem condições que relacionam consulta EXTERNA e INTERNA (codp = p.codp)

Uso de NOT EXISTS

- Para comparar grupos de tuplas com grupos de tuplas
- Comparar
- o nome dos pilotos que (IN, NOT IN)
 - voam com pelo menos uma aeronave usada por piloto p11
 - voam com pelo menos uma aeronave que piloto p11 não pilota
- o nome dos pilotos que (NOT EXISTS)
 - voam com todas as aeronaves que o piloto p11 voa (e talvez outras)
 - não voam com nenhuma das aeronaves usadas pelo piloto de código p11
 - voam com exatamente as mesmas aeronaves com as quais o piloto p11 voa

Exists (sempre, todos)

- o nome dos pilotos que voam com todas as aeronaves que o piloto p11 voa (e talvez outras)



- para selecionar um dado piloto p, não pode existir nenhuma aeronave que p11 tenha pilotado que p não tenha pilotado

Exists (sempre, todos)

- o nome dos pilotos que voam com todas as aeronaves que o piloto p11 voa (e talvez outras)

- alternativa preferencial

```
select nomep
from piloto p
where codp <> 'p11' and
not exists (select *
from escala e
where codp = 'p11' and
aviao not in
(select distinct aviao
from escala
where codp = p.codp))
```

Exists (sempre, todos)

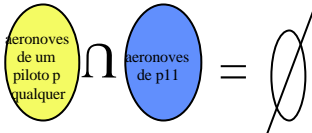
- o nome dos pilotos que voam com todas as aeronaves que o piloto p11 voa (e talvez outras)

- outra alternativa mais custosa (a ser evitada)

```
select nomep
from piloto p
where codp <> 'p11' and
not exists (select *
from escala e
where codp = 'p11' and
not exists (select *
from escala
where aviao = e.aviao
and codp = p.codp))
```

Exists (nunca, nenhum)

- o nome dos pilotos que não voam com nenhuma das aeronaves usadas pelo piloto de código p11



- para selecionar um dado piloto p, não pode existir nenhuma aeronave que p11 tenha pilotado e que p também tenha pilotado
- tanto faz a ordem, pois é interseção

Exists (nunca, nenhum)

- o nome dos pilotos que não voam com nenhuma das aeronaves usadas pelo piloto de código p11

- alternativa preferencial

```
select nomep
from piloto p
where not exists (select *
                  from escala e
                  where codp = 'p11' and
                  aviao in (select distinct aviao
                           from escala
                           where codp = p.codp))
```

- observações:
 - tanto faz a ordem da subconsulta 1 ou 2, pois é interseção

Exists (nunca, nenhum)

- o nome dos pilotos que não voam com nenhuma das aeronaves usadas pelo piloto de código p11

- outra alternativa mais custosa (a ser evitada)

```
select nomep
from piloto p
where codp <> 'p11' and
not exists (select *
            from escala e
            where codp = 'p11' and
            exists (select *
                   from escala
                   where aviao = e.aviao
                   and
                   codp = p.codp))
```

- Observações:
 - tanto faz a ordem da subconsulta 1 ou 2, pois é interseção