

# Subconsultas ou Consultas Aninhadas

- Forma alternativa de especificar consultas envolvendo relacionamentos entre tabelas
- **Otimização**
  - filtragens prévias de dados na subconsulta
    - apenas tuplas/atributos de interesse são combinados com dados da(s) tabela(s) da consulta externa
- Cláusulas de subconsulta
  - *nome\_atributo* [NOT] IN (*consulta\_SQL*)
  - *nome\_atributo* [< | <= | > | >= | < > | !=] ANY (*consulta\_SQL*)
  - *nome\_atributo* [< | <= | > | >= | < > | !=] ALL (*consulta\_SQL*)

# Subconsultas com IN

- Testam a relação de pertinência ou não-pertinência elemento-conjunto

```
select lista_atributos
from tabela1 [...]
where atributo_ou_expressão [NOT] IN
      (consulta_SQL)
```

- Mapeamento para a álgebra relacional

```
select a1, ..., an
from t1
where c IN
      (select x from t2
       where d > 5)
```

⇒  $\pi_{a_1, \dots, a_n} (t_1 \theta X (\pi_x (\sigma_{d > 5} (t_2))))$

$\theta = t_1.c = t_2.x$

# Exemplos

## Álgebra

$\pi_{\text{nome}} ($   
 $(\text{Médicos} \theta X$   
 $\theta = \text{Médicos.codm} = \text{Consultas.codm}$   
 $(\pi_{\text{codm}} (\sigma_{\text{data} = '2010/10/13'} (\text{Consultas}))) )$   
 $)$

$(\pi_{\text{CPF}} (\text{Funcionários})) \text{ — }$   
 $(\pi_{\text{CPF}} (\text{Pacientes}))$

$(\pi_{\text{CPF}} (\text{Médicos})) \cap$   
 $(\pi_{\text{CPF}} (\text{Pacientes}))$

## SQL

```
Select nome  
From Médicos  
Where codm in  
(select codm  
from Consultas  
where data = '2010/10/13')
```

```
Select CPF  
From Funcionários  
Where CPF not in  
(select CPF  
from Pacientes)
```

```
Select CPF  
From Médicos  
Where CPF in  
(select CPF  
from Pacientes)
```

# Diferença/Intersecção de Tabelas

- Exige tabelas compatíveis

Álgebra	SQL
$relação1 \text{ — } relação2$	$consultaSQL1 \text{ except } consultaSQL2$
$relação1 \cap relação2$	$consultaSQL1 \text{ intersect } consultaSQL2$

- Exemplos

$(\pi_{RG}(\text{Funcionários})) \text{ — } (\pi_{RG}(\text{Pacientes}))$	Select RG From Funcionários <b>except</b> Select RG From Pacientes
$(\pi_{RG}(\text{Médicos})) \cap (\pi_{RG}(\text{Pacientes}))$	Select RG From Médicos <b>intersect</b> Select RG From Pacientes

- Observação

– Nem todos os SGBDs implementam estas operações

# Subconsultas com ANY

- Permitem outras comparações do tipo elemento-conjunto
  - testa se um valor é  $>$ ,  $<$ ,  $=$ , ... que *algum* valor em um conjunto

```
select lista_atributos
from tabela1 [, ...]
where atributo_ou_expressão [=|<|<=|>|>=|<>|!=] ANY
(consulta_SQL)
```

- Mapeamento para a álgebra relacional

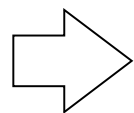
```
select a1, ..., an
```

```
from t1
```

```
where c > ANY
```

```
(select x from t2
```

```
where d > 5)
```



$\pi_{a_1, \dots, a_n} (t_1 \theta X (\pi_x (\sigma_{d > 5} (t_2))))$

$\theta = t_1.c > t_2.x$

# Exemplos

Álgebra	SQL
$\pi_{\text{nome}} \left( \begin{array}{l} \text{Médicos} \theta X \\ \theta = \text{Médicos.codm} = \text{Consultas.codm} \\ \left( \pi_{\text{codm}} \left( \sigma_{\text{data} = '2010/10/13'} \right. \right. \\ \left. \left. \left( \text{Consultas} \right) \right) \right) \end{array} \right)$	<pre>Select nome From Médicos Where codm = any (ou in) (select codm  from Consultas  where data = '2010/10/13')</pre>
$\pi_{\text{Funcionários.idade}} \left( \begin{array}{l} \left( \pi_{\text{idade}} \left( \text{Funcionários} \right) \right) \theta X \\ \theta = \text{Funcionários.idade} < f2.idade \\ \left( \pi_{\text{idade}} \left( \rho_{f2} \left( \text{Funcionários} \right) \right) \right) \end{array} \right)$	<pre>Select nome From Funcionários Where idade &lt; any ( Select      idade      from Funcionários)</pre>

# Subconsultas com ALL

- Realiza uma comparação de igualdade ou desigualdade de um elemento com todos os elementos de um conjunto

```
select lista_atributos  
from tabela1 [, ...]  
where atributo_ou_expressão [=|<|<=|>|>=|<>|!=]  
      ALL(consulta_SQL)
```

- Não tem mapeamento para a álgebra relacional
  - não é equivalente a divisão
    - na divisão existe apenas comparação de igualdade
    - dividendo deve ter mais atributos que o divisor
    - não filtra automaticamente atributos do dividendo

# Exemplos

```
Select nome
From Funcionários
Where salário > all
      (Select salário
       From Funcionários
       Where departamento = 'contábil')
```

```
Select nome
From Funcionários
Where CPF < > all (or not in)
      (Select CPF
       From Pacientes)
```



# Comparações Elemento-Elemento

- Casos em que a subconsulta retorna apenas **um elemento como resultado**
  - cardinalidade da subconsulta = 1
  - não é utilizada nenhuma cláusula de subconsulta neste caso

```
select lista_atributos
```

```
from tabela1 [, ...]
```

```
where atributo_ou_expressão [=|<|<=|>|>=|<>|!  
=] (consulta_SQL com um único elemento)
```

# Exemplos

```
Select nome
From Funcionários
Where salário >
      (Select salário
       From Funcionários
       Where CPF = 22000200002)
```

```
select nome, CPF
from Médicos
where CPF < > 10000100001
and especialidade =
      (select especialidade
       from Médicos
       where CPF = 10000100001)
```

# Subconsultas com EXISTS

- **Quantificador existencial** do cálculo relacional
  - testa se um predicado é V ou F na subconsulta
  - para cada tupla da consulta externa a ser analisada, a subconsulta é executada

```
select lista_atributos
from tabela1 [, ...]
where [NOT] EXISTS (consulta_SQL)
```

- Mapeamento para o cálculo relacional

```
select a1, ..., an
```

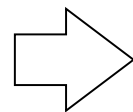
```
from T1
```

```
where EXISTS
```

```
(select * from T2
```

```
where d > 5
```

```
and T2.x = T1.c)
```


$$\{t_1.a_1, \dots, t_1.a_n \mid t_1 \in T_1 \wedge \exists t_2 \in T_2 (t_2.d > 5 \wedge t_2.x = t_1.c)\}$$

# Exemplos

Cálculo	SQL
$\{m.\text{nome} \mid m \in \text{Médicos} \wedge \exists c \in \text{Consultas} (c.\text{data} = '10/10/13' \wedge c.\text{codm} = m.\text{codm})\}$	<pre>Select m.nome From Médicos m Where exists   (Select *    From Consultas    Where data = '2010/10/13'     and codm = m.codm)</pre>
$\{f.\text{nome} \mid f \in \text{Funcionários} \wedge f.\text{depto} = \text{'pessoal'} \wedge \neg \exists p \in \text{Pacientes} (p.\text{CPF} = f.\text{CPF})\}$	<pre>Select f.nome From Funcionários f Where f.depto = 'pessoal' and not exists   (Select *    From Pacientes    Where CPF = f.CPF)</pre>

# Relembrando...

- Propriedade de Equivalência dos Quantificadores Universal e Existencial

$$\forall t \in R (P(t)) \equiv \neg \exists t \in R (\neg P(t))$$

- Buscar o nome dos médicos que têm consulta marcada com todos os pacientes

$\{m.\text{nome} \mid m \in \text{Médicos} \wedge \forall p \in \text{Pacientes} (\exists c \in \text{Consultas} (p.\text{codp} = c.\text{codp} \wedge c.\text{codm} = m.\text{codm}))\}$

$\downarrow \equiv$

$\{m.\text{nome} \mid m \in \text{Médicos} \wedge \neg \exists p \in \text{Pacientes} (\neg \exists c \in \text{Consultas} (p.\text{codp} = c.\text{codp} \wedge c.\text{codm} = m.\text{codm}))\}$

# Exemplo

Cálculo	SQL
$\{p.codp, p.nome \mid p \in \text{Pacientes} \\ \wedge \neg \exists m \in \text{Médicos} \\ (\neg \exists c \in \text{Consultas} \\ (c.codm = m.codm \wedge \\ p.codp = c.codp))\}$	<pre>Select p.codp, p.nome From Pacientes p Where not exists   (Select *    From Médicos m    Where not exists      (Select *       From Consultas c       Where c.codm = m.codm             and c.codp = p.codp))</pre>

- A consulta acima equivale ao uso do operador de **divisão** da álgebra relacional:

$$\pi_{codp, nome} (\text{Pacientes}) \bowtie (\pi_{codp, codm} (\text{Consultas}) \div \pi_{codm} (\text{Médicos}))$$

# Exercícios

Resolva o que se pede utilizando subconsultas IN:

- 1) nome e CPF dos médicos que também são pacientes do hospital
- 2) código e nome dos pacientes com consulta marcada para horários após às 14 horas
- 3) nome e idade dos médicos que têm consulta com a paciente Ana
- 4) número e andar dos ambulatórios onde nenhum médico dá atendimento
- 5) nome, CPF e idade dos pacientes que têm consultas marcadas sempre para dias anteriores ao dia 16

Resolva o que se pede utilizando subconsultas ANY e/ou ALL:

- 1) números e andares de todos os ambulatórios, exceto o de menor capacidade
- 2) nome e idade dos médicos que têm consulta com a paciente Ana
- 3) nome e a idade do médico mais jovem (sem usar função MIN!)
- 4) nome e CPF dos pacientes com consultas marcadas para horários anteriores a todos os horários de consultas marcadas para o dia 12 de Outubro de 2010
- 5) nome e CPF dos médicos que não atendem em ambulatórios com capacidade superior à capacidade dos ambulatórios do segundo andar

Resolva o que se pede utilizando subconsultas EXISTS:

- 1) nome e CPF dos médicos que também são pacientes do hospital
- 2) nome e idade dos médicos que têm consulta com a paciente Ana
- 3) número do ambulatório com a maior capacidade (sem usar função MAX!)
- 4) nome e CPF dos médicos que têm consultas marcadas com todos os pacientes
- 5) nome e CPF dos médicos ortopedistas que têm consultas marcadas com todos os pacientes de Florianópolis