

Sub-consultas Embebidas

A linguagem SQL permite a inclusão de sub-consultas na cláusula FROM de uma consulta.

Por exemplo: achar a média dos balanços dos balcões cuja média dos balanços é maior do que 12000€.

```
SELECT balcaoNome, mediaBalanco
FROM (SELECT balcaoNome, avg(balanco)
        FROM Conta
        GROUP BY balcaoNome) AS mediaBalcao (balcaoNome, mediaBalanco)
WHERE mediaBalanco > 12000
```

Deixa de ser necessário a cláusula HAVING dado que se calcula a sub-consulta (vista temporária) e os atributos da mesma podem ser usados na consulta principal.

Relações Derivadas

A cláusula **WITH** permite a definição de uma sub-consulta (vista temporária) cujo alcance é somente o da consulta aonde a mesma ocorre.

Exemplo: achar todas as contas cujo balanço seja igual ao valor máximo achado.

```
WITH BalancoMax(valor)
  AS SELECT max(balanco)
     FROM Conta
SELECT numero
  FROM Conta, BalancoMax
 WHERE Conta.balanco = BalancoMax.valor
```

A cláusula **WITH** é o correspondente à instrução de atribuição na álgebra relacional. No entanto é de notar que a mesma não pode ser usada sem estar associada a uma consulta.

Relações Derivadas

Outro exemplo: Achar todos os balcões aonde o valor total dos depósitos é maior do que a média dos valores dos depósitos em todos os balcões.

```
WITH TotalBalcao(balcaoNome, valor)
  AS SELECT balcaoNome, sum(balanco)
  FROM Conta
  GROUP BY balcaoNome
WITH MediaTotalBalcao(valor)
  AS SELECT avg(valor)
  FROM TotalBalcao
SELECT balcaoNome
  FROM TotalBalcao, MediaTotalBalcao
 WHERE TotalBalcao.valor > MediaTotalBalcao.valor
```

Notar que após a definição da relação derivada a mesma pode ser usada de imediato.

Vistas

- Em certas circunstâncias, não é desejável que todos os utilizadores possam aceder a todo o modelo lógico (i.e. a todas as relações armazenadas na base de dados)
- Considere o caso de um empregado que necessita de saber o número de empréstimo de um cliente, mas que não precisa de saber o montante desse empréstimo. Este empregado deveria ver apenas a relação

```
(SELECT nome, numero  
  FROM ClienteEmp, Empréstimo  
  WHERE ClienteEmp.numero = Empréstimo.numero)
```

- Uma **vista** providencia um mecanismo de sonegação de informação.
- Qualquer relação que não pertença ao modelo conceptual mas que se torne visível ao utilizador como uma «relação virtual» é designada por **vista**.

Definição de Vistas

- Uma vista é definida por intermédio da instrução **CREATE VIEW**
CREATE VIEW NomeVista **AS** <consulta>

Em que <consulta> é uma consulta SQL qualquer. O nome da vista é NomeVista.

- Após a definição de uma vista, o seu nome pode ser utilizado para se referir à relação virtual gerada pela vista.
- Uma definição de uma vista não é o mesmo que criar uma nova relação a partir da avaliação da sua expressão. Em vez disso, a definição da vista permite guardar a expressão que depois é substituída nas consultas que utilizam a vista.

Exemplos de Vistas

- Considere-se a vista (com o nome `todosOsClientes`) contendo os nomes das agências e seus clientes.

```
CREATE VIEW TodosOsClientes
AS (SELECT balcaoNome, nome
      FROM ClienteEmp, Conta
      WHERE ClienteEmp.numero = Conta.numero)
UNION
(SELECT balcaoNome, nome
  FROM ClienteEmp, Emprestimo
  WHERE ClienteEmp.numero = Emprestimo.numero)
```

Exemplos de Vistas

- Considere-se a vista (com o nome todosOsClientes) contento os nomes das agências e seus clientes.

```
CREATE VIEW TodosOsClientes
AS (SELECT balcaoNome, nome
      FROM ClienteEmp, Conta
      WHERE ClienteEmp.numero = Conta.numero)
UNION
(SELECT balcaoNome, nome
  FROM ClienteEmp, Emprestimo
  WHERE ClienteEmp.numero = Emprestimo.numero)
```

- Pode-se assim encontrar todos os clientes da agência de Penacova através de:

```
SELECT nome
FROM TodosOsClientes
WHERE balcaoNome = 'Penacova'
```

Vistas definidas a partir de outras vistas

- Uma vista pode ser utilizada na expressão de definição de outra vista.
- Uma vista v_1 **depende directamente** de uma vista v_2 se v_2 é utilizada na expressão que define v_1 .
- Uma vista v_1 **depende** de uma vista v_2 se v_1 depende directamente de v_2 ou se existe um caminho de dependências entre v_1 e v_2 .

Expansão de vistas

- Forma de atribuir significado a vistas definidas em termos de outras vistas.
- Seja a vista v_1 definida em termos de uma expressão e_1 que pode ela própria conter vistas.
- Para expandir as vistas numa expressão repete-se sucessivamente o seguinte passo:

Repetir

Encontrar uma vista v_i em e_1

Substituir a vista v_i pela expressão que a define

Até que não ocorram mais vistas em e_1

Mudanças Através de Vistas

É possível usar uma **Vista** para actualizar uma tabela, utilizando as instruções UPDATE, DELETE, ou INSERT.

Para que seja possível actualizar uma tabela com uma **Vista** é necessário que haja uma relação injectiva (um-para-um) entre as linhas da **Vista** e as linhas da tabela que ela referencia.

Além das restrições eventualmente impostas pelas permissões de acesso às tabelas subjacentes existem outras restrições que podem impedir a actualização de valores através de **Vistas**.

Mudanças Através de Vistas (cont.)

Para poder actualizar as tabelas subjacentes a **Vista** não pode conter:

- funções de agregação;
- DISTINCT; GROUP BY; HAVING; UNION;
- uma sub-consulta na lista de selecção;
- alguns tipos de junções;
- uma **Vista** não actualizável na cláusula FROM;
- múltiplas referências a uma dada coluna na tabela base.

Quanto à inserção de valores verifica-se ainda que:

- não pode haver duplicação nos nomes das colunas;
- a **Vista** tem de conter todas as colunas de preenchimento obrigatório e em que não esteja definido um valor por omissão;
- as colunas da **Vista** não pode ser uma coluna derivada de outras colunas.