

Bases de Dados 2005/2006

Aula 5

Sumário

- 1. (T.P.C.) Indique diferenças entre uma tabela e uma relação.
 - 0. A base de dados Projecto
 - 1. SQL – Join (variantes)
 - a. Cross Join
 - b. Equi-Join
 - c. Natural Join
 - d. Inner Join
 - e. Outer Join
 - i. Left Join
 - ii. Right Join
 - iii. Full Join
 - f. Self Join
 - 2. Operações sobre conjuntos
 - a. Union
 - b. Intersect
 - c. Except
 - 3. Inserção, actualização e remoção de dados
 - a. Insert
 - b. Update
 - c. Delete
-

-1.(T.P.C.) Uma tabela representa sempre uma relação?

Questão para reflexão em casa.

0. A base de dados Projecto

Use o programa *SQL Query Analyzer* e estabeleça uma ligação ao servidor de base de dados *ferro.win.di.ubi.pt*. Não se esqueça de seleccionar a base de dados apropriada (Projecto!).

1. SQL – Join (variantes)

Objectivo: estudo dos diferentes tipos de ligação entre tabelas

A ligação entre tabelas é frequentemente designada por *join*.

a. Cross join

O *produto cartesiano* de duas tabelas associa a cada linha da primeira tabela o conjunto das linhas da segunda tabela.

```
Select *  
From Empregado, Departamento
```

b. Equi-join

```
Select *  
From Empregado E, Departamento D  
Where E.DepNum = D.DepNum
```

A ligação entre as tabelas (via cláusula *Where*) é realizada através da igualdade entre duas colunas (*equi-join*).

Na equi-join todas as colunas (de todas as tabelas) são apresentadas e a ligação entre as tabelas faz-se através de uma igualdade (nota: aparecem colunas com conteúdo igual.). Só os registos que satisfazem a condição de junção aparecem no resultado.

c. Natural join

```
Select E.*, D.Nome, Local
From Empregado E, Departamento D
Where E.DepNum = D.DepNum
```

Na *junção natural* não são apresentadas colunas (com conteúdo) repetidas.

d. Inner join

```
Select Departamento.*
From Empregado, Departamento
Where Empregado.DepNum = Departamento.DepNum
```

Numa *inner join* só são apresentadas as linhas em que exista ligação entre as tabelas.

Em SQL Server:

```
Select Departamento.*
From Empregado INNER JOIN Departamento ON Empregado.DepNum = Departamento.DepNum
```

e. Outer join (em SQL Server)

A *outer join* permite obter a totalidade das linhas de uma tabela, mesmo quando não existe o correspondente valor na outra tabela. A *junção externa* pode ser realizada à esquerda ou à direita.

i. Left join

Quando a junção externa é realizada à esquerda, aparecem todas as linhas da tabela na esquerda e apenas as linhas correspondentes da tabela na direita.

```
Select EmpNum, Salário, Departamento.Nome
From Empregado LEFT JOIN Departamento ON Empregado.DepNum = Departamento.DepNum
```

ii. Right join

Quando a junção externa é realizada à direita, aparecem todas as linhas da tabela na direita e apenas as linhas correspondentes da tabela na esquerda.

```
Select EmpNum, Salário, Departamento.Nome
From Empregado RIGHT JOIN Departamento ON Empregado.DepNum = Departamento.DepNum
```

iii. Full join

Combina as duas junções anteriores...

```
Select EmpNum, Salário, Departamento.Nome
From Empregado FULL JOIN Departamento ON Empregado.DepNum = Departamento.DepNum
```

f. Self join

Variante da *Inner join* em que se comparam duas colunas da mesma tabela.

```
Select Distinct A1.EmpNum
From Atribuicao A1, Atribuicao A2
Where A1.EmpNum = A2.EmpNum
and A1.ProjNum <> A2.ProjNum
```

2. Operações sobre conjuntos

a. Union

A *união* permite juntar o conteúdo de vários comandos Select.

As tabelas que participam na união têm que ser *compatíveis em união*:

1. Ambas têm de ter o mesmo grau (mesmo número de colunas)
2. Colunas correspondentes em cada tabela têm de ter o mesmo domínio (o mesmo tipo de dados).

```
Select Nome  
From Empregado  
UNION  
Select Nome  
From Departamento
```

- O nome das colunas que aparece no resultado é o nome das colunas seleccionadas com o primeiro Select.
- As restrições presentes na cláusula Where são aplicadas ao respectivo Select.
- Só poderá aparecer uma única cláusula Order By (e no último Select), sendo a ordenação aplicada a todo o resultado.
- Por defeito uma união remove as linhas duplicadas (para duplicados usar UNION ALL).

```
Select EmpNum, Nome, Categoria  
From Empregado  
Where Salario <= 200000  
UNION  
Select DepNum, Nome, NULL  
From Departamento  
Where nome Like '%c%'  
Order By 2
```

b. Intersect

Não está disponível no SQL Server 2000.

```
Select EmpNum, Nome, Categoria  
From Empregado  
Where Salario <= 200000  
INTERSECT  
Select EmpNum, Nome, Categoria  
From Empregado  
Where Nome LIKE '%da%'
```

c. Except

Não está disponível no SQL Server 2000.

```
Select EmpNum, Nome, Categoria  
From Empregado  
Where Salario <= 200000  
EXCEPT  
Select EmpNum, Nome, Categoria  
From Empregado  
Where Salario Between 140000 and 180000
```

3. Inserção, actualização e remoção de dados

a. Insert

```
Insert Into table_name [(column_list)]
Values (data_value_list)
```

- *column_list* é opcional.
- Se omitida, é assumida a ordem indicada no CREATE TABLE.
- As colunas omitidas devem ser declaradas como NULL quando a tabela foi criada, a menos que DEFAULT tenha sido especificado aquando da criação da coluna.
- *data_value_list* deve corresponder ao *column_list* do seguinte modo:
 - O número de elementos em cada lista deve ser igual.
 - Deve haver uma correspondência directa na posição dos elementos de cada lista.
 - O tipo de dados de cada item na *data_value_list* deve ser compatível com o tipo de dados na correspondente coluna.

```
Insert Into Departamento
Values (13, 'Informática', 'Covilhã')
```

```
Insert Into Departamento (DepNum, Nome, Local)
Values (14, 'Matemática', 'Covilhã')
```

Nota: consulte a DDL para a criação da tabela Departamento e teste as *constraints*.

INSERT...SELECT

Uma segunda forma de INSERT permite que várias linhas sejam copiadas de uma ou mais tabelas para outra.

```
Insert Into table_name [(column_list)]
Select ...
```

b. Update

O comando UPDATE permite alterar os valores existentes existentes nos campos de uma (única) tabela.

```
Update table_name
Set column_name1 = data_value1
[, column_name2 = data_value2...]
[Where search_condition]
```

- *table_name* pode ser o nome de uma tabela de base ou de uma *view* actualizável.
- A cláusula SET especifica uma ou mais colunas a serem actualizadas.
- A cláusula WHERE é opcional:
 - Se omitida, as colunas indicadas são actualizadas para todas as linhas da tabela.
 - Se especificada, somente as linhas que satisfazem a *search_condition* são actualizadas.
- Os *data_value(s)* devem ser compatíveis com os tipos de dados das colunas correspondentes.

```
Update Departamento
Set Local = 'Fundão'
Where DepNum = 13
```

Nota: consulte a DDL para a criação da tabela Departamento e teste as *constraints*.

c. Delete

O comando DELETE permite apagar linhas de uma (única) tabela.

```
Delete From table_name  
[Where search_condition]
```

- *table_name* pode ser o nome de uma tabela de base ou de uma *view* atualizável.
- A cláusula WHERE é opcional:
 - Se omitida, todas as linhas da tabela são apagadas da tabela (a *tabela não é apagada*).
 - Se especificada, somente as linhas que satisfazem a *search_condition* são apagadas.

```
Delete Departamento  
Where DepNum = 13
```