

TLBD II

Continuação (05)

Esta operação junta linhas de uma tabela com linhas da própria tabela. Para isso é necessário utilizar sinônimos para os nomes das tabelas. Estes criam uma **cópia virtual** da tabela, originando assim uma relação entre duas tabelas, que já foi exemplificada na aula anterior.

Sintaxe antiga	ON
<pre>SELECT e.pnome "Nome empregado" , e.salario "Salário empregado", m.pnome "Nome Gerente", m.salario "Salário Gerente" FROM empregados e, empregados m WHERE e.gerente_id=m.emp_id AND e.salario < m.salario;</pre>	<pre>SELECT e.pnome "Nome empregado" , e.salario "Salário empregado", m.pnome "Nome Gerente", m.salario "Salário Gerente" FROM empregados e INNER JOIN empregados m ON (e.gerente_id=m.emp_id) WHERE e.salario < m.salario;</pre>

Junção de uma tabela com ela própria

```
CREATE TABLE gradesal (  
grade CHAR(2) PRIMARY KEY,  
minsal NUMERIC(8,2),  
maxsal NUMERIC(8,2)) ENGINE=INNODB;
```

grade	minsal	maxsal
1	1500.00	2500.00
2	1000.00	1499.00
3	500.00	999.00

Crie esta tabela GRADESAL com os seguintes dados.

Sempre que o relacionamento entre duas tabelas é obtido por um operador que não seja = estamos numa NÃO-EQUI-JUNÇÃO. Para determinar o nível salarial de um empregado (GRADE) temos que encontrar o intervalo de valores em que o salário do empregado se enquadra, estando as categorias armazenadas na tabela GRADESAL. O relacionamento entre as tabelas EMPREGADOS e GRADESAL é uma não-equi-junção, pois nenhuma coluna de EMPREGADOS corresponde diretamente a uma coluna de GRADESAL sendo a relação obtida utilizando o operador BETWEEN.

Não equi-junção entre tabelas

- Também poderíamos ter utilizado os operadores $\geq$ e $\leq$

Sintaxe antiga	ON
SELECT E.pnome, E.salario, S.grade FROM empregados E, gradesal S WHERE E.salario BETWEEN S.minsal AND S.maxsal	SELECT E.pnome, E.salario, S.grade FROM empregados E INNER JOIN gradesal S ON (E.salario BETWEEN S.minsal AND S.maxsal)

Não equi-junção entre tabelas

O SQL disponibiliza três operadores que permitem a partir de dois conjuntos produzir um terceiro. Estes operadores atuam sobre as **colunas** das tabelas de origem, produzindo uma **Junção Vertical**:

- UNIÃO
- INTERSEÇÃO (não disponível em MySQL)
- DIFERENÇA (não disponível em MySQL)

Junção Vertical

São úteis na junção de diferentes consultas que se refiram a tabelas diferentes mas que tenham colunas com o mesmo tipo de dados. A sua utilização segue as seguintes regras:

- As duas instruções SELECT têm que selecionar o **mesmo número de colunas**;
- As colunas correspondentes têm que ser do **mesmo tipo de dados**;
- As linhas duplicadas são eliminadas pelo operador UNION. Se usarmos UNION ALL os duplicados não são eliminados, o que poupa uma operação de SORT;
- Os **nomes de coluna** da primeira consulta aparecem no resultado;
- A cláusula ORDER BY aparece no fim da instrução. Se as colunas correspondentes dos diferentes SELECT's tiverem nomes diferentes, a referência à(s) coluna(s) deverá ser feita por **número de posição**;
- As instruções SELECT são executadas de **cima para baixo**;
- Se forem utilizados vários operadores de conjunto é possível usar **parêntesis** para definir prioridades de execução;

Junção Vertical

O comando abaixo apresenta os nomes e salários dos empregados das tabelas EMP e EMPREGADOS. São duas consultas unidas numa única.

```
SELECT enome, sal  
FROM EMP  
UNION  
SELECT pnome, salario  
FROM empregados;
```

UNIÃO (UNION)

O comando UNION elimina automaticamente as linhas repetidas. Se pretendermos que estas apareçam teremos que utilizar UNION ALL:

UNION	UNION ALL
<pre>SELECT funcao FROM EMP UNION SELECT emprego_id FROM empregados;</pre>	<pre>SELECT funcao FROM EMP UNION ALL SELECT emprego_id FROM empregados;</pre>

UNIÃO (UNION)

O comando abaixo permite listar os salários que existem em EMP e em EMPREGADOS:

INTERSECT	INNER JOIN (MySQL)
SELECT sal FROM EMP INTERSECT SELECT salario FROM empregados;	SELECT EMP.sal FROM EMP INNER JOIN empregados USING (sal)

INTERSEÇÃO (INTERSECT)

O comando abaixo permite listar os salários que estão em EMP e não estão em empregados:

MINUS	LEFT JOIN (MySQL)
<pre>SELECT sal FROM EMP MINUS SELECT salario FROM empregados;</pre>	<pre>SELECT DISTINCT EMP.sal FROM EMP LEFT JOIN empregados USING (sal) WHERE empregados.sal IS NULL</pre>

DIFERENÇA (MINUS)

Exercícios03