Universidade Federal de Minas Gerais
Colégio Técnico - COLTEC
Prof. Márcio Fantini
Estudo Orientado 4 - Operações em Conjunto

1 Introdução

- Nesse estudo orientado vamos rever as operações em conjunto do SQL e Mysql.
- Usaremos o banco de dados *conjuntos*.
- `http://galileu.coltec.ufmg.br/fantini/hp/CursoBD/BancosDadosSQLTestes/conjuntos.sql`
- Ele possui as seguintes tabelas:

```
mysql> use conjuntos
Database changed
mysql> show tables;
+---------------------+
| Tables_in_conjuntos |
+---------------------+
| t1                  |
| t2                  |
| tab1                |
| tab2                |
+---------------------+
4 rows in set (0,01 sec)
```

1.1 Exemplo Inicial

Pergunta: como mostrar as colunas das tabelas t1 e t2 juntas? (como mostrado abaixo):

```
+------+--------+------+--------+
| id1  | valor1 | id2  | valor2 |
+------+--------+------+--------+
|    1 |      1 |    1 |      0 |
|    2 |      1 |    2 |      1 |
|    3 |      2 |    3 |      2 |
|    4 |      3 |    4 |      3 |
|    5 |      3 |    5 |      4 |
|    6 |      4 |    6 |      5 |
+------+--------+------+--------+
6 rows in set (0,00 sec)
```

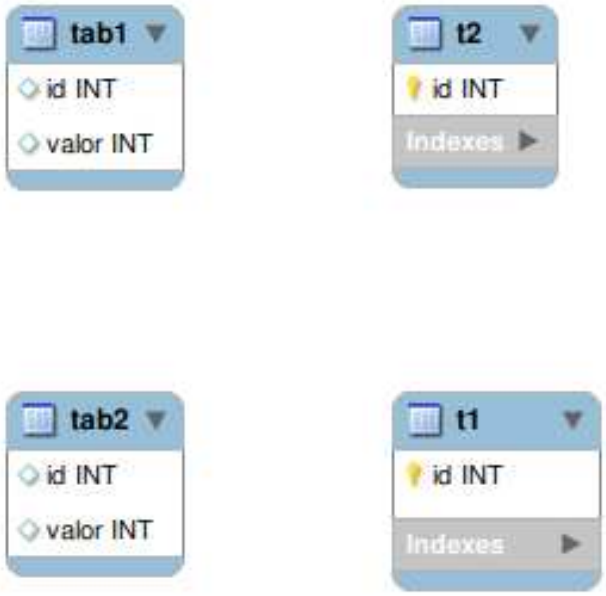

Figura 1: Modelo do banco de dados ”conjuntos”exportado pelo MysqlWorkbench.

Se fizermos apenas um select das colunas, não obteremos esse resultado. Veja:

```
mysql> select t1.id id1,t1.valor valor1,t2.id id2,t2.valor valor2 from tab1 t1,tab2 t2;
+------+--------+------+--------+
| id1  | valor1 | id2  | valor2 |
+------+--------+------+--------+
|    6 |      4 |    1 |      0 |
|    5 |      3 |    1 |      0 |
|    4 |      3 |    1 |      0 |
|    3 |      2 |    1 |      0 |
|    2 |      1 |    1 |      0 |
|    1 |      1 |    1 |      0 |
|    6 |      4 |    2 |      1 |
|    5 |      3 |    2 |      1 |
|    4 |      3 |    2 |      1 |
|    3 |      2 |    2 |      1 |
|    2 |      1 |    2 |      1 |
|    1 |      1 |    2 |      1 |
|    6 |      4 |    3 |      2 |
|    5 |      3 |    3 |      2 |
|    4 |      3 |    3 |      2 |
|    3 |      2 |    3 |      2 |
|    2 |      1 |    3 |      2 |
|    1 |      1 |    3 |      2 |
|    6 |      4 |    4 |      3 |
```

```
|    5 |      3 |    4 |      3 |
|    4 |      3 |    4 |      3 |
|    3 |      2 |    4 |      3 |
|    2 |      1 |    4 |      3 |
|    1 |      1 |    4 |      3 |
|    6 |      4 |    5 |      4 |
|    5 |      3 |    5 |      4 |
|    4 |      3 |    5 |      4 |
|    3 |      2 |    5 |      4 |
|    2 |      1 |    5 |      4 |
|    1 |      1 |    5 |      4 |
|    6 |      4 |    6 |      5 |
|    5 |      3 |    6 |      5 |
|    4 |      3 |    6 |      5 |
|    3 |      2 |    6 |      5 |
|    2 |      1 |    6 |      5 |
|    1 |      1 |    6 |      5 |
+------+--------+------+--------+
36 rows in set (0,00 sec)
```

Nesse caso, fizemos um produto cartesiano entre as duas tabelas. Não é isso que queremos. Queremos mostrar as duas colunas de cada tabela. Para isso usamos uma junção. Uma opção é fazer como mostrado abaixo:

```
mysql> select t1.id id1,t1.valor valor1,t2.id id2,t2.valor valor2 from tab1 t1
inner join tab2 t2 using (id);

+------+--------+------+--------+
| id1  | valor1 | id2  | valor2 |
+------+--------+------+--------+
|    1 |      1 |    1 |      0 |
|    2 |      1 |    2 |      1 |
|    3 |      2 |    3 |      2 |
|    4 |      3 |    4 |      3 |
|    5 |      3 |    5 |      4 |
|    6 |      4 |    6 |      5 |
+------+--------+------+--------+
6 rows in set (0,00 sec)
```

2 União

Nesse exemplo acima realizamos junções entre tabelas, par Quando realizamos junções entre tabelas, formamos uma relação resultante com as colunas que contêm as colunas das tabelas originais.
Podemos ter situações que requerem resultados de consultas que não são relacionadas mas que o resultado de ambas sejam importantes.
Em situações como essas usamos as Uniões.
O predicado UNION é utilzado posicionado entre dois comandos de consulta. Podemos, por exemplo mostrar os resultados das colunas id e valor das duas tabelas:

```
mysql> select * from tab1
    -> union
    -> select * from tab2;
+------+-------+
| id   | valor |
+------+-------+
|    1 |     1 |
|    2 |     1 |
|    3 |     2 |
|    4 |     3 |
|    5 |     3 |
|    6 |     4 |
|    1 |     0 |
|    5 |     4 |
|    6 |     5 |
+------+-------+
9 rows in set (0,00 sec)
```

Nesse caso exemplificado acima, a consulta teve como resultado os valores (id,valor) de ambas as tabelas, com a condição de não serem valores repetidos.
Uma outra consulta possível é quando usamos o UNION ALL.
Veja o exemplo abaixo:

```
mysql> select * from tab1 UNION ALL select * from tab2;
+------+-------+
| id   | valor |
+------+-------+
|    1 |     1 |
|    2 |     1 |
|    3 |     2 |
|    4 |     3 |
|    5 |     3 |
|    6 |     4 |
|    1 |     0 |
```

```
|    2 |     1 |
|    3 |     2 |
|    4 |     3 |
|    5 |     4 |
|    6 |     5 |
+------+-------+
12 rows in set (0,00 sec)
```

Ou seja:

- UNION: união de valores, apenas os diferentes
- UNION ALL: união de todos os valores.

Condição para termos a UNIÃO: as colunas devem ter o mesmo tipo de dado e os comandos devem retornar o mesmo número de colunas.

3 Interseção

O operador INTERSECT do SQL é um operador de conjunto que retorna apenas linhas distintas de duas ou mais consultas.
O operador INTERSECT compara os conjuntos de resultados de duas consultas e retorna as linhas distintas que são geradas por ambas as consultas.
Para usar o operador INTERSECT para duas consultas, siga estas regras:

- A ordem e o número de colunas na lista de seleção das consultas devem ser iguais.
- Os tipos de dados das colunas correspondentes devem ser compatíveis.

O diagrama a seguir ilustra o operador INTERSECT.

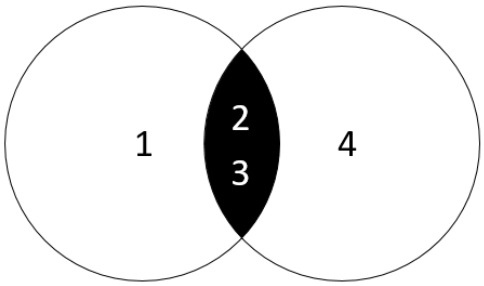

Figura 2: Exemplo básico de uma interseção entre dois conjuntos, representado pelo diagrama de Venn

O myslq não possui o comando INTERSECT. Para fazermos uma interseção temos que usar uma junção. Veja o exemplo a seguir, no qual fazemos a interseção entre o conjunto (tabela) t1 e o conjunto (tabela) t2).

```
mysql> select * from t1;
+----+
| id |
+----+
|  1 |
|  2 |
|  3 |
+----+
3 rows in set (0,03 sec)

mysql> select * from t2;
+----+
| id |
+----+
|  2 |
|  3 |
|  4 |
+----+
3 rows in set (0,02 sec)


mysql> select distinct id from t1
    -> inner join t2
    -> using (id);
+----+
| id |
+----+
|  2 |
|  3 |
+----+
2 rows in set (0,00 sec)
```

É importante usarmos a palavra chave DISTINCT.
Veja as operações de JOIN quando não usamos distinct e depois usando distinct, para as tabelas conj1 e conj2 dadas no exemplo abaixo:

```
mysql> select * from conj1;
+--------+
| numero |
+--------+
|      1 |
```

```
|      2 |
|      2 |
|      3 |
|      3 |
+--------+
5 rows in set (0,00 sec)


mysql> select * from conj2;
+--------+
| numero |
+--------+
|      1 |
|      2 |
|      3 |
|      4 |
+--------+
4 rows in set (0,01 sec)



mysql> select numero from conj1
    -> join conj2
    -> using(numero);
+--------+
| numero |
+--------+
|      1 |
|      2 |
|      2 |
|      3 |
|      3 |
+--------+
5 rows in set (0,01 sec)

mysql> select distinct numero from conj1 join conj2 using(numero);
+--------+
| numero |
+--------+
|      1 |
|      2 |
|      3 |
+--------+
```

4 Diferença

As cláusulas EXCEPT e MINUS da SQL são duas maneiras de se obter a diferença entre dois conjuntos (ambas fazem a mesma coisa).
A diferença é que EXCEPT está disponível no banco de dados PostgreSQL, enquanto MINUS está disponível em Oracle e SQL Server.
No mysql usamos NOT IN junto com uma subquery.

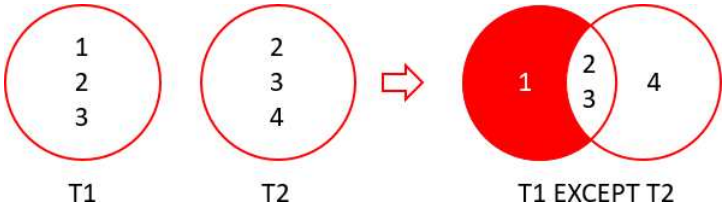

Figura 3: Exemplo básico de diferença entre dois conjuntos, representado pelo diagrama de Venn

```
mysql> select numero from conj1 where numero not in (select numero from conj2);
Empty set (0,00 sec)

mysql> select numero from conj2 where numero not in (select numero from conj1);
+--------+
| numero |
+--------+
|      4 |
+--------+
1 row in set (0,00 sec)
```

Repare que as colunas envolvidas nas operações de conjunto precisam ter o mesmo tipo, mas não o mesmo nome. Veja por exemplo o caso da diferença abaixo. A subconsulta feita na tabela conj1 retorna uma coluna, que é avaliada pela consulta externa.

```
mysql> ALTER TABLE `conj2` CHANGE `numero` `num` INT NULL DEFAULT NULL;

mysql> select * from conj2;
+------+
| num  |
+------+
|    1 |
|    2 |
|    3 |
|    4 |
+------+
4 rows in set (0,00 sec)
```

```
mysql> select num from conj2 where num not in (select numero from conj1);
+------+
| num  |
+------+
|    4 |
+------+
1 row in set (0,01 sec)
```

5 Exercício

Crie o banco de dados conjuntos com todas as tabelas apresentadas acima. Insira os valores e execute as consultas apresentadas.

Compilado 12 de novembro de 2021 - 11:33:18