

02

## Múltiplos valores na condição e o IN

### Selecionando valores distintos, mas sem repetições

Recebi um e-mail do pessoal do financeiro, e eles querem saber todas as formas de pagamento que temos cadastradas em nosso banco de dados, para saber se são as mesmas formas permitidas por eles.

Se formos olhar a estrutura da tabela `matricula` :

```
desc matricula;
```

Temos uma coluna `tipo`, que representa o tipo de pagamento. Vamos selecionar essa coluna para ver os tipos de pagamento:

```
select tipo from matricula;
```

Vemos que alguns valores se repetem, certo? Por que isso acontece? Porque tem vários alunos que usaram a mesma forma de pagamento. Mas eu não quero pegar as repetições, quero pegar todas as formas, mas só uma vez cada. Quero pegar as formas de pagamento **distintas**, mas sem as repetidas.

Para isso podemos usar o `DISTINCT` :

```
select distinct tipo from matricula;
```

Agora vem só uma ocorrência de cada tipo, legal né?

### Filtrando por um conjunto determinado de valores

Com as formas de pagamento em mãos, o financeiro também quer saber quantas matrículas temos, mais especificamente para pessoas jurídica ou física, por curso. Ou seja, queremos as matrículas que tenham como forma de pagamento os tipos `PAGA_PF` ou `PAGA_PJ`.

Então vamos começar, primeiramente que, se eu quero a quantidade de matrículas, tenho que fazer a contagem delas:

```
select count(m.id) as quantidade from matricula m;
```

Agora, vamos selecionar o nome do curso e associar a matrícula com o curso:

```
select c.nome, count(m.id) as quantidade from matricula m
join curso c on m.curso_id = c.id;
```

Se estamos fazendo a contagem, temos que agrupar esse resultado, vamos agrupar pelo nome do curso:

```
select c.nome, count(m.id) as quantidade from matricula m
join curso c on m.curso_id = c.id
group by c.nome;
```

E por fim, vamos filtrar:

```
select c.nome, count(m.id) as quantidade from matricula m
join curso c on m.curso_id = c.id
where m.tipo = 'PAGA_PJ' or m.tipo = 'PAGA_PF'
group by c.nome;
```

Para melhorar, vamos adicionar a coluna do tipo da matrícula no resultado:

```
select c.nome, m.tipo, count(m.id) as quantidade from matricula m
join curso c on m.curso_id = c.id
where m.tipo = 'PAGA_PJ' or m.tipo = 'PAGA_PF'
group by c.nome, m.tipo;
```

Agora o pessoal do financeiro quer fazer o mesmo relatório mas com mais um tipo diferente. Segundo essa lógica eu teria que colocar **mais um OR**, certo? Meio trabalhoso né?

Repare que temos um grupo de valores, no nosso caso são 2, `PAGA_PJ` e `PAGA_PF`, mas poderíamos ter mais, e quando queremos passar um grupo de valores em uma condição, podemos usar o `IN`:

```
select c.nome, m.tipo, count(m.id) as quantidade from matricula m
join curso c on m.curso_id = c.id
where m.tipo in ('PAGA_PJ', 'PAGA_PF')
group by c.nome, m.tipo;
```

Caso necessitássemos filtrar por mais alguma outra forma de pagamento, bastaria colocar esse valor dentro do conjunto, ao invés de ter que escrever outro `OR`.

## Sabendo quais cursos determinados alunos fizeram

Um instrutor comentou que três alunos foram exemplares, em uma turma que ele deu aula mês passado. Queremos saber todos os cursos que esses alunos fizeram.

Primeiro, vamos procurar os alunos:

```
select * from aluno;
```

```
SQL> select * from aluno;

      ID NOME                           EMAIL
-----+
      1 Joao da Silva                  joao@dasilva.com
      2 Frederico Jose                fred@jose.com
      3 Alberto Santos                alberto@santos.com
      4 Renata Alonso                 renata@alonso.com
      5 Paulo da Silva                paulo@dasilva.com
      6 Carlos Cunha                  carlos@cunha.com
      7 Paulo Jose                   paulo@jose.com
      8 Manoel Santos                 manoel@santos.com
      9 Renata Ferreira                renata@ferreira.com
     10 Paula Soares                 paula@soares.com
     11 Jose da Silva                 jose@dasilva.com
     12 Danilo Cunha                  danilo@cunha.com
     13 Zilmira Jose                  zilmira@jose.com
     14 Cristaldo Santos              cristaldo@santos.com
     15 Osmir Ferreira                 osmir@ferreira.com
     16 Claudio Soares                claudio@soares.com

16 rows selected.
```

O instrutor lembrava do nome dos alunos, são o João, o Alberto e a Renata. Ids 1, 3 e 4.

Então, vamos selecionar os cursos:

```
select c.nome from curso c;
```

Agora, vamos até a tabela de matrículas:

```
select c.nome from curso c
join matricula m on m.curso_id = c.id;
```

Agora, adicionamos o `WHERE` com o `IN`, para selecionar só os cursos dos alunos com os ids que queremos:

```
select c.nome from curso c
join matricula m on m.curso_id = c.id
where m.aluno_id in (1, 3, 4);
```

```
SQL> select c.nome from curso c
  2      join matricula m on m.curso_id = c.id
  3  where m.aluno_id in (1, 3, 4);

      NOME
-----
SQL e banco de dados
Scrum e metodos ageis
C# e orientacao a objetos
C# e orientacao a objetos
C# e orientacao a objetos
Desenvolvimento mobile com Android

6 rows selected.
```

Ficou meio estranho esse resultado. Vamos selecionar o nome dos alunos também, sem esquecer de fazer o `JOIN`:

```
select a.nome, c.nome from curso c
join matricula m on m.curso_id = c.id
join aluno a on m.aluno_id = a.id
where m.aluno_id in (1, 3, 4);
```

Para melhorar esse resultado, vamos ordenar pelo nome dos alunos:

```
select a.nome, c.nome from curso c
join matricula m on m.curso_id = c.id
join aluno a on m.aluno_id = a.id
where m.aluno_id in (1, 3, 4)
order by a.nome;
```

```
SQL> select a.nome, c.nome from curso c
  2      join matricula m on m.curso_id = c.id
  3      join aluno a on m.aluno_id = a.id
  4  where m.aluno_id in (1, 3, 4)
  5  order by a.nome;

NOME                      NOME
-----
Alberto Santos              C# e orientacao a objetos
Alberto Santos              Scrum e metodos ageis
Joao da Silva               C# e orientacao a objetos
Joao da Silva               SQL e banco de dados
Renata Alonso              Desenvolvimento mobile com Android
Renata Alonso              C# e orientacao a objetos

6 rows selected.
```

Bem melhor, né?

## Divulgando um novo curso para determinados ex-alunos

Agora que estamos lançando um curso de **Oracle Database**, queremos divulgá-lo para ex-alunos. Mas não pra todos ex-alunos, somente os que fizeram o curso de **SQL e banco de dados** ou **PHP e MySql**.

Vamos dar uma olhada nesses cursos então:

```
select * from curso;
```

```
SQL> select * from curso;

ID NOME
-----
1 SQL e banco de dados
2 Desenvolvimento web com VRaptor
3 Scrum e metodos ageis
4 C# e orientacao a objetos
5 Java e orientacao a objetos
6 Desenvolvimento mobile com iOS
7 Desenvolvimento mobile com Android
8 Ruby on Rails
9 PHP e MySql

9 rows selected.
```

Então eu quero os cursos com ids 1 e 9.

Então, vamos selecionar as matrículas para esses cursos:

```
select c.nome from curso c
join matricula m on m.curso_id = c.id;
```

Para buscar os alunos, vamos até a tabela de alunos:

```
select c.nome from curso c
join matricula m on m.curso_id = c.id
join aluno a on m.aluno_id = a.id;
```

Vamos selecionar o nome do aluno:

```
select a.nome, c.nome from curso c
join matricula m on m.curso_id = c.id
join aluno a on m.aluno_id = a.id;
```

Aqui temos todos os alunos matriculados, certo? Mas queremos somente os alunos dos cursos 1 e 9!

Então:

```
select a.nome, c.nome from curso c
join matricula m on m.curso_id = c.id
join aluno a on m.aluno_id = a.id
where c.id in (1, 9);
```