

02

## Entendendo o LEFT JOIN

Agora os instrutores querem um relatório para saber quem são os alunos mais participativos em sala de aula. Uma forma de saber quem participa mais, é ver quais alunos estão respondendo mais exercícios. Consequentemente, encontraremos também quem são os que não participam muito, daí já conversamos com eles e vemos o porquê disso.

Então vamos começar selecionando os alunos:

```
select a.nome from aluno a;
```

Depois vamos até a tabela de respostas, selecionando-as através do JOIN :

```
select a.nome from aluno a
join resposta r on r.aluno_id = a.id;
```

Como queremos a quantidade de respostas, vamos adicionar o COUNT e um alias para ele:

```
select a.nome, count(r.id) as quantidade from aluno a
join resposta r on r.aluno_id = a.id;
```

Mas não podemos esquecer de agrupar o resultado:

```
select a.nome, count(r.id) as quantidade from aluno a
join resposta r on r.aluno_id = a.id
group by a.nome;
```

```
SQL> select a.nome, count(r.id) as quantidade from aluno a
  2      join resposta r on r.aluno_id = a.id
  3  group by a.nome;

NOME          QUANTIDADE
-----          -----
Joao da Silva          7
Frederico Jose          4
Renata Alonso          7
Alberto Santos          9
```

Será que estamos vendo todos os alunos aqui? Vamos confirmar vendo quantos alunos temos na tabela de alunos, fazendo uma contagem:

```
select count(*) from aluno;
```

Na contagem temos 16 alunos, mas só 4 foram listados na query que fizemos anteriormente! Porque? Se olharmos o resultado de nossa query, parece que ela só retorna os alunos que tem resposta.

Será que é isso mesmo? Vamos conferir.

Vamos selecionar os alunos:

```
select * from aluno;
```

Vamos pegar como exemplo o Paulo da Silva e ver se ele não tem resposta mesmo. Vamos selecionar tudo da tabela de respostas:

```
select * from resposta;
```

Agora vamos selecionar as respostas do aluno com id 5 (id do aluno Paulo da Silva):

```
select * from resposta where aluno_id = 5;
```

Nenhuma resposta, então estávamos certo! Nossa query só traz os alunos que têm resposta, e há alunos sem respostas.

## Selecionando também os alunos sem respostas

Do lado esquerdo, temos dados da tabela de alunos (no nosso caso, o seu nome), e do lado direito, os dados da tabela de respostas (a quantidade delas). Quando fizemos o `JOIN`, ele trouxe os dados das duas tabelas - através do id do aluno.

Mas na tabela de aluno, temos outros alunos certo? Queremos trazer esses registros também, todos os registros da tabela da esquerda, independentemente se esses alunos tiverem resposta ou não.

Vamos refazer nossa query usando um `JOIN` diferente, o `LEFT JOIN`. O `LEFT JOIN` sempre traz os registros da tabela da esquerda, mesmo se não houver dados correspondentes a eles na tabela da direita.

```
select a.nome, count(r.id) as quantidade from aluno a
  left join resposta r on r.aluno_id = a.id
  group by a.nome;
```

Agora ele trouxe todos os alunos!

## Mas e as respostas sem aluno?

Acabamos de ver como buscar um aluno sem resposta. Mas e se quiséssemos fazer o contrário: buscar uma resposta sem aluno? Nesse caso queremos pegar todas as respostas, todos os exercícios, mesmo os que não tem aluno, ou seja, que não foram respondidos.

Primeiro, iríamos na tabela de aluno:

```
select a.nome from aluno a;
```

E faríamos um `JOIN` com a tabela de respostas, mas nesse caso queremos trazer **todos** os registros da tabela da **direita**. Então, assim como temos o `LEFT JOIN` para trazer todos os dados da tabela da esquerda, temos o `RIGHT JOIN` para trazer os dados da tabela da direita, que é o que queremos:

```
select a.nome from aluno a
  right join resposta r on r.aluno_id = a.id;
```

Agora, vamos adicionar a resposta no `SELECT` :

```
select a.nome, r.resposta_dada from aluno a
right join resposta r on r.aluno_id = a.id;
```

Executando essa query, vemos que todos os exercícios têm alunos. Isso porque não temos em nosso banco um exercício sem aluno.

Para verificar a funcionalidade do `RIGHT JOIN`, vamos fazer um teste: qual a coluna que associa uma resposta a um aluno? `ALUNO_ID`, da tabela de respostas.

Então, vamos inserir um registro na tabela de resposta associado a um aluno que não existe.

Será que existe um aluno com id 50000? Acho que não né?

Mas vamos conferir:

```
select * from aluno where id = 50000;
```

Ok, vamos inserir uma resposta com um `aluno_id` igual a 50000 então:

```
insert into resposta (id, exercicio_id, aluno_id, resposta_dada)
values (28, 1, 50000, 'c# e vb');
```

Vamos executar para inserir o registro. Lembrando que poderíamos prevenir esse `INSERT` inválido de acontecer se a coluna `ALUNO_ID` fosse uma **foreign key**.

Então, vamos selecionar os dados novamente e ver o que acontece:

```
select a.nome, r.resposta_dada from aluno a
right join resposta r on r.aluno_id = a.id;
```

Agora conseguimos ver que há uma resposta sem aluno!

Mas não podemos nos esquecer de excluir esse registro que incluímos:

```
delete from resposta where aluno_id = 50000;
```

Então, usamos o `LEFT JOIN` para pegar todos os registros, inclusive todos da tabela da esquerda. Para pegar todos os registros, incluindo todos da tabela da direita usamos o `RIGHT JOIN`. Para pegar só os do meio, ou seja, só os registros que existem nas duas tabelas, usamos o `JOIN` “normal”, o `JOIN` nem da esquerda nem da direita, o de dentro.

Chamamos esse `JOIN` de `INNER JOIN`.

