

## Aluno sem matrícula e o EXISTS

### Populando o nosso banco de dados

Vimos no curso anterior que SQL é uma ferramenta bastante útil para lidar com dados. Por meio dela, conseguimos juntar, filtrar, e agregar dados de diferentes tabelas, de maneira fácil e expressiva.

O objetivo deste curso, agora que SQL já é uma linguagem conhecida, é mostrar algumas consultas um pouco mais complexas. Elas são importantes quando queremos um relatório mais específico ou quando precisamos ver o dado sob um ponto de vista diferente.

Neste curso, utilizaremos um conjunto de tabelas e dados de um sistema de ensino online (parecido com este aqui). Temos tabelas que representam os alunos, as matrículas, os cursos, as seções, os exercícios, as respostas, as avaliações, e etc.

Como no curso anterior, usaremos o SQL\*Plus para nos comunicar com o Oracle Database, através do terminal, basta digitar `sqlplus`, e logo após digitar seu usuário e sua senha.

Assim que logar, o Oracle estará pronto para receber instruções. Basta digitar e elas serão executadas pelo Oracle.

Precisamos agora importar os dados e tabelas que já estão pré-prontas. Para isso, baixe o arquivo `.sql` que está disponibilizado no primeiro exercício deste curso, de acordo com a linguagem padrão do seu banco de dados, e envie-o para o Oracle. Exemplo:

```
SQL> @DIRETORIO_DO_ARQUIVO_SQL/sql1br.sql
```

Você pode abrir esse arquivo num editor de textos para vê-lo. Ele simplesmente contém instruções SQL que serão executadas pelo banco de dados.

Agora veja todas as tabelas existentes, e familiarize-se um pouco com o modelo. Para listar todas as tabelas do seu usuário, você pode usar o seguinte comando:

```
select table_name from user_tables;
```

E para detalhar, para ver a estrutura de uma tabela, você lembra? Use `desc <nome da tabela>`.

### Em que cursos os alunos estão matriculados?

Vamos detalhar as tabelas `aluno` e `curso`:

```
SQL> desc aluno;
Name                               Null?    Type
-----
ID                                 NOT NULL NUMBER
NOME                              NOT NULL VARCHAR2(30)
EMAIL                             NOT NULL VARCHAR2(30)

SQL> desc curso;
Name                               Null?    Type
-----
ID                                 NOT NULL NUMBER
NOME                              NOT NULL VARCHAR2(40)
```

Um aluno pode fazer dois cursos ao mesmo tempo, um de Delphi e um de Java, por exemplo. Se ele não for bem no de Java, ele pode fazer de novo. Esse é o cenário que temos aqui: um aluno pode fazer vários cursos ao mesmo tempo ou várias vezes o mesmo curso. Ou seja, muitos alunos podem estar matriculados em muitos cursos. Isso é o que chamamos de uma relação **many-to-many**. E para essa relação ser consumada, uma outra tabela precisa ser criada:

```
SQL> desc matricula;
Name                               Null?    Type
-----
ID                                 NOT NULL NUMBER
ALUNO_ID                          NOT NULL NUMBER
CURSO_ID                          NOT NULL NUMBER
DATA                              NOT NULL DATE
TIPO                              NOT NULL VARCHAR2(20)
```

Veja que a tabela de matrículas é uma tabela de ligação, que faz a correspondência de ambos os lados. Por isso ela contém duas chaves estrangeiras (uma para aluno e uma para curso) e outras informações como data e tipo de matrícula.

Tendo isso em vista, eu quero selecionar o aluno e quais cursos ele está matriculado. Vamos por passos, primeiro selecionando o nome do aluno:

```
select nome from aluno;
```

Agora vamos até a tabela de matrículas para selecionar as matrículas dele:

```
select nome from aluno
join matricula on matricula.aluno_id = aluno.id;
```

Por último, temos que ir na tabela de cursos, para ver qual curso o aluno está matriculado:

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

Repare que, como estamos selecionando colunas de tabelas diferentes, precisamos dizer ao Oracle a qual tabela a coluna pertence, pois pode haver uma ou mais colunas com o mesmo nome em tabelas diferentes, caso de `nome`, que existe em `aluno` e `curso`. Mas para não precisarmos repetir sempre os nomes das tabelas, podemos dar apelidos a elas, simplificando as queries.

Podemos dizer ao Oracle, "quero selecionar a.nome" ( `select a.nome` ), mas o Oracle não sabe que tabela `a` é essa, por isso terminamos a query dizendo a ele que `a`, na verdade, é um apelido da tabela `aluno` ( `select a.nome from aluno a` ).

Agora que já vimos como funcionam os apelidos das tabelas, podemos simplificar a última query que fizemos, dando apelidos para as tabelas, e é assim que iremos fazer daqui pra frente:

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

Olhando o resultado, vemos que Paulo José, Renata Ferreira e Paula Soares estão matriculados em um curso cada um, que João da Silva, Alberto Santos, Renata Alonso e Manoel Santos estão matriculados em dois cursos, além do Frederico José, que está matriculado em 3 cursos. Mas lembra do primeiro `select` que fizemos? Seleccionamos os nomes dos alunos. Se formos contá-los ( `select count(*) from aluno` ) vemos que os resultados não estão iguais! Temos 8 alunos matriculados em um ou mais cursos, mas 16 alunos estão cadastrados, será que temos alunos sem matrícula?

## Exibindo alunos com e sem matrícula

Podemos fazer uma query para verificar isso, primeiro vamos selecionar todos os alunos:

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

E depois buscar só os alunos que tem matrícula, onde a matrícula **existe**, para isso vamos usar o `EXISTS` :

```
select a.nome from aluno a where exists(...);
```

Mas o que vamos passar para o `EXISTS` ? Queremos verificar se o aluno tem matrícula, certo?

```
select a.nome from aluno a where exists(
  select m.id from matricula m where m.aluno_id = a.id
);
```

Pronto, agora temos os alunos que estão matriculados em algum curso, mas não é isso que queremos, certo? Queremos os alunos que **não** estão matriculados em nenhum curso. Para ter esse resultado, basta fazermos a mesma query anterior, mas utilizando o operador de negação **NOT**:

```
select a.nome from aluno a where not exists(
  select m.id from matricula m where m.aluno_id = a.id
);
```

Rodando a query, vemos que vários alunos não estão matriculados em nenhum curso, exatamente 8 alunos, por isso o resultado da contagem dos alunos cadastrados no banco de dados não coincidia com o resultado dos alunos matriculados em algum curso.

## Existem exercícios não respondidos?

Com todos os exercícios no banco de dados, gostaríamos de saber os que ainda não foram respondidos.

Como faço para selecionar todos os exercícios?

```
select * from exercicio e;
```

E os que tiverem respostas?

```
select * from exercicio e where exists(  
    select r.id from resposta r where r.exercicio_id = e.id  
);
```

E os que não tiverem respostas? Basta adicionar o NOT :

```
select * from exercicio e where not exists(  
    select r.id from resposta r where r.exercicio_id = e.id  
);
```

Com isso temos todos os exercícios que ainda não foram respondidos!

## Quais cursos estão sem matrícula?

O pessoal do comercial me enviou um e-mail falando que quer intensificar a venda, o marketing dos cursos que não estão vendendo muito. Vamos saber então quais são os cursos sem matrícula.

Começamos pegando o nome do curso:

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

Em seguida, vamos até a tabela de matrícula e fazemos o EXISTS :

```
select c.nome from curso c where exists (  
    select m.id from matricula m where m.curso_id = c.id  
);
```

Mais uma vez, aqui temos os cursos que têm matrícula, quero os que não têm, então:

```
select c.nome from curso c where not exists (  
    select m.id from matricula m where m.curso_id = c.id  
);
```

Com essa query temos todos os cursos que não têm matrícula! Se não der nenhuma retorno, sinal que todos os cursos possuem alguma matrícula.