

1) Acessar o database **tipos**:

USE TIPOS;

2) Criar a tabela **alunos**.

```
CREATE TABLE ALUNOS(  
  COD          INT AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  NOME         VARCHAR(30) NOT NULL,  
  NR_CURSO    INT,  
  IDADE        INT NOT NULL,  
  SEXO         CHAR NOT NULL CHECK (SEXO IN ('F', 'M'))  
);
```

3) Criar a tabela **graduacao**.

```
CREATE TABLE GRADUACAO(  
  COD          INT AUTO_INCREMENT PRIMARY KEY NOT NULL,  
  CURSO        VARCHAR(30) NOT NULL,  
  VALOR        DECIMAL(10,2) NOT NULL  
);
```

4) Observe as tabelas abaixo e crie a query para inserir os registros:

| Alunos | | | | |
|--------|---------|---------|-----|-------|
| cod | nome | nr_curs | Sex | Idade |
| 1 | Amanda | 4 | F | 20 |
| 2 | Bianca | 3 | F | 21 |
| 3 | Carlos | 3 | M | 32 |
| 4 | Emerso | 2 | M | 30 |
| 5 | Fabio | 2 | M | 28 |
| 6 | Ivan | | M | 23 |
| 7 | Juliana | | F | 26 |

| Graduacao | | |
|-----------|---------|--------|
| cod | curso | Valor |
| 1 | Redes | 500,00 |
| 2 | Turismo | 480,00 |
| 3 | Letras | 370,00 |
| 4 | Adm | 530,00 |

```
INSERT INTO graduacao (curso, valor) VALUES ('Redes', 500);
INSERT INTO graduacao (curso, valor) VALUES ('Turismo', 480);
INSERT INTO graduacao (curso, valor) VALUES ('Letras', 370);
INSERT INTO graduacao (curso, valor) VALUES ('ADM', 530);
```

```
INSERT INTO alunos (nome, nr_curso, idade, sexo) VALUES ('Amanda', 4, 20, 'F');
INSERT INTO alunos (nome, nr_curso, idade, sexo) VALUES ('Bianca', 3, 21, 'F');
INSERT INTO alunos (nome, nr_curso, idade, sexo) VALUES ('Carlos', 3, 32, 'M');
INSERT INTO alunos (nome, nr_curso, idade, sexo) VALUES ('Emerson', 2, 30, 'M');
INSERT INTO alunos (nome, nr_curso, idade, sexo) VALUES ('Fabio', 2, 28, 'M');
INSERT INTO alunos (nome, curso, idade, sexo) VALUES ('Ivan', 23, 'M');
INSERT INTO alunos (nome, curso, idade, sexo) VALUES ('Juliana', 26, 'F');
```

5) Observe as tabelas abaixo e confira os registros nas tabelas criadas.

| Alunos | | | | |
|--------|---------|---------|-----|-------|
| cod | nome | nr_curs | Sex | Idade |
| 1 | Amanda | 4 | F | 20 |
| 2 | Bianca | 3 | F | 21 |
| 3 | Carlos | 3 | M | 32 |
| 4 | Emerso | 2 | M | 30 |
| 5 | Fabio | 2 | M | 28 |
| 6 | Ivan | | M | 23 |
| 7 | Juliana | | F | 26 |

| Graduacao | | |
|-----------|---------|--------|
| cod | curso | Valor |
| 1 | Redes | 500,00 |
| 2 | Turismo | 480,00 |
| 3 | Letras | 370,00 |
| 4 | Adm | 530,00 |

```
select * from graduacao;
```

```
select * from alunos;
```

6) Dada a query abaixo, responda qual será o RESULTADO após execução.

a)
Select alunos.nome, graduacao.curso
From alunos
INNER JOIN graduacao
On alunos.nr_curso = graduacao.cod;

Apresenta os nomes dos alunos e o curso de graduação referente a cada aluno, definido através da relação de igualdade entre nr_curso da tabela alunos com cod da tabela curso, alunos que não o tenham o código de curso ou alunos que tenham códigos de curso não apresentados na tabela graduação... não serão mostrados!

b)
Select alunos.nome, graduacao.curso
From alunos
RIGHT JOIN graduacao
On alunos.nr_curso = graduacao.cod;

Apresenta todos os cursos da tabela graduacao e os nomes da tabela aluno que tenha algum curso presente na tabela graduação associado ao seu cadastro.
No caso de conter algum curso sem ligação com pelo menos um aluno, o campo nome ficará em branco(nulo). Ex: (nulo) Redes

c)
Select alunos.nome, graduacao.curso
From **alunos**
LEFT JOIN graduacao
On alunos.nr_curso = graduacao.cod;

Apresenta todos os nomes da tabela alunos e os cursos da tabela graduacao que tenha pelo menos um aluno associado ao seu cadastro.
No caso de conter algum aluno sem ligação com pelo menos um curso, o campo curso ficará em branco(nulo). EX: Ivan (nulo)
Juliana (nulo)

d)
Select alunos.nome, graduacao.curso
From alunos
RIGHT JOIN graduacao
On alunos.nr_curso = graduacao.cod
UNION
Select alunos.nome, graduacao.curso
From **alunos**
LEFT JOIN graduacao
On alunos.nr_curso = graduacao.cod;

Apresenta a união do left join com right join, ou seja, mostra todos os nomes da tabela aluno em uma coluna e na outra coluna mostra os cursos associados a cada nome, por meio da condição: alunos.nr_curso = graduacao.cod, havendo algum curso sem aluno associado, o campo onde seria o nome fica com valor nulo, e o nome que não tiver curso associado, no campo curso fica com valor nulo também.
EX: Ivan (nulo) // (s/ curso associado)
Juiana (nulo) // (s/ curso associado)
(nulo) Redes // (s/ aluno associado)

e)
Select alunos.nome, graduacao.curso
From alunos
CROSS JOIN graduação;

Apresenta todos os cursos da tabela graduação ligados, a cada nome da tabela aluno.

Ex:

| | |
|--------|---------|
| Amanda | Redes |
| Amanda | Turismo |
| Amanda | Letras |
| Amanda | ADM |

obs: o mesmo se repete apara os demais nomes da tabela aluno.