

Exercícios de Fixação – Part I

Para a resolver os exercícios de fixação, será usado o banco de dados final_space_db, execute o script abaixo no seu MySQL Workbench:

```
DROP SCHEMA IF EXISTS final_space_db ;

CREATE SCHEMA IF NOT EXISTS final_space_db ;
USE final_space_db ;

CREATE TABLE IF NOT EXISTS final_space_db.Location (
    Location_id INT NOT NULL AUTO_INCREMENT,
    Location_Name VARCHAR(45) NOT NULL,
    Location_Type VARCHAR(45) NOT NULL,
    PRIMARY KEY (Location_id)
);

CREATE TABLE IF NOT EXISTS final_space_db.Specie (
    Specie_id INT NOT NULL AUTO_INCREMENT,
    Specie_Name VARCHAR(45) NOT NULL,
    PRIMARY KEY (Specie_id)
);

CREATE TABLE IF NOT EXISTS final_space_db.Character (
    Character_id INT NOT NULL AUTO_INCREMENT,
    Name VARCHAR(45) NOT NULL,
    Origin INT NULL,
    Specie INT NULL,
    PRIMARY KEY (Character_id),
    FOREIGN KEY (Origin)
        REFERENCES final_space_db.Location (Location_id),
    FOREIGN KEY (Specie)
        REFERENCES final_space_db.Specie (Specie_id)
);

INSERT INTO final_space_db.Location(
    Location_id,
    Location_Name,
    Location_Type
)
VALUES
    (1, 'Earth', 'Planet'),
    (2, 'Final Space', 'Dimension'),
    (3, 'Tera Con Prime', 'Destroyed planet'),
    (4, 'Inner Space', 'Dimension'),
    (5, 'Yarno', 'Planet'),
    (6, 'Dark Zone', 'Black hole'),
    (7, 'Deathcropolis', 'Gladiatorial Colosseum'),
    (8, 'Lazarus Trap', 'Trap'),
    (9, 'The Order of the Twelve', 'Temple'),
    (10, 'Serepentis', 'Planet'),
    (11, 'Ventrexia', 'Planet'),
    (12, 'Invictus Prison', 'Dimension'),
```

```

(13, 'Tryvuulian', 'Planet'),
(14, 'Hatched', 'Planet');

```

```

INSERT INTO final_space_db.Specie(Specie_id, Specie_Name)

```

```

VALUES

```

```

(1, 'Human'),
(2, 'Ventrexian'),
(3, 'Robot'),
(4, 'Serepentian'),
(5, 'Tryvuulian'),
(6, 'Artificial Intelligence'),
(7, 'Titan'),
(8, 'Cosmic Entity'),
(9, 'Energy being'),
(10, 'Alien'),
(11, 'Hooblot'),
(12, 'Cyborg'),
(13, 'Fire Snake'),
(14, 'Tiger Tiger');

```

```

INSERT INTO final_space_db.Character(

```

```

    Character_id,

```

```

    Name,

```

```

    Origin,

```

```

    Specie

```

```

)

```

```

VALUES

```

```

(1, 'Gary Goodspeed', 1, 1),
(2, 'Mooncake', NULL, NULL),
(3, 'Quinn Ergon', 1, 1),
(4, 'Little Cato', 11, 2),
(5, 'Avocato', 11, 2),
(6, 'KVN', 1, 3),
(7, 'Ash Graven', 10, 4),
(8, 'Fox', 13, 5),
(9, 'H.U.E', NULL, 6),
(10, 'Tribore Menendez', 14, NULL),
(11, 'Clarence', NULL, NULL),
(12, 'Biskit', NULL, 14),
(13, 'Quatronostro Menendez', 14, NULL),
(14, 'Lord Commander', NULL, NULL),
(15, 'Sheryl Goodspeed', 1, 1),
(16, 'Bolo', 2, 7),
(17, 'Invictus', 2, 7),
(18, 'John Goodspeed', 1, 1),
(19, 'A.V.A', NULL, 6),
(20, 'Evra', 2, 9),
(21, 'Queen of Ventrexia', 11, 2),
(22, 'King of Ventrexia', 11, 2),
(23, 'Mega-KVN', 1, 3),
(24, 'Melanie Dewinter', 3, 11),
(25, 'Harp Graven', 10, 4),
(26, 'Shannon Thunder', 1, 1),
(27, 'David Dewinter', 3, 11),
(28, 'Meowlapeño', 3, 2),
(29, 'Fraskenhaur', NULL, 10),
(30, 'Oreskis"', 2, 7 );

```

PART I

- 1 - Escreva uma query que retorne uma coluna com o nome dos personagens e uma coluna com o nome de sua espécie. ✓
- 2 - Escreva uma query que retorne uma coluna com os nomes dos personagens e uma coluna com o nome do seu local de origem. ✓
- 3 - Agora, retorne o nome do personagem, sua espécie e o lugar de origem em uma query. ✓
- 4 - Escreva uma query que retornará uma coluna com o nome da espécie, nomeie como Espécie, e uma coluna com a quantidade de personagens daquela espécie, nomeie como Quantidade. Por fim, organize a lista de forma ascendente pela quantidade. ✓
- 5 - Escreva uma query que retorne uma coluna com o nome do Local, nomeie como Local de origem, e uma coluna com quantidade de personagens que tenham origem desse local, nomeie como Quantidade. Por fim, organize a lista de forma decrescente pela quantidade. ✓
- 6 - Escreva uma query que retorne os nomes dos personagens que tem como local de origem 'Earth'. Por fim, organize os nomes em ordem alfabética. ✓
- 7 - Escreva uma query que retorne nome e local de origem de todos personagens que são de espécie 'Ventrexian'. Por fim, ordene os nome de forma alfabética-invertida. ✓

PART II

- 1 - Escreva uma query que exiba os nomes de todos os personagens e sua espécie. Observação: O nome de todos os personagens deve aparecer, inclusive os que não tem uma espécie definida. ✓
- 2 - Escreva uma query que exiba os nomes de todos os personagens e seu local de origem. Observação: O nome de todos os personagens deve aparecer, inclusive os que não tem um local de origem definido. ✓
- 3 - Escreva uma query que mostre nome do personagem e nome de todas as espécies, mesmo as que não possuem personagem com aquela espécie. ✓
- 4 - Escreva uma query que retorne nome do personagem e nome de todos locais de origem, inclusive os personagens sem origem desse local. ✓

5 - Escreva uma query que exiba nomes de todos os personagens e sua espécie. Se o personagem não tiver espécie definida, o campo da espécie deverá mostrar 'Unknown'. Por fim, nomeie a coluna espécie de 'Specie' e a coluna nomes dos personagens de 'Character'. ✓

6 - Escreva uma query que retorne nome do personagem e nome de todos locais de origem. Nomeie coluna nome do personagem de 'Character' e coluna do local de 'Local'. Caso não houver personagem com origem do local, o campo que apresentaria no nome do personagem deve mostrar 'Unknown'. ✓

PART III

1 - Escreva uma query que exiba nomes dos personagens em uma coluna e nomes dos personagens com mesmo local de origem na coluna ao lado; ✓

2 - Escreva uma query que exiba nomes dos personagens em uma coluna e nomes dos personagens com mesmo local de origem na coluna ao lado. Agora, sem que nome do personagem apareça duas vezes na mesma linha. ✓

3 - Escreva uma query que exiba nomes dos personagens em uma coluna e nomes dos personagens da mesma espécie na coluna ao lado. ✓

4 - Escreva uma query que exiba nomes dos personagens em uma coluna e nomes dos personagens da mesma espécie na coluna ao lado. Agora, sem que nome do personagem apareça duas vezes na mesma linha. ✓

Exercícios – Part II

Para resolver os exercícios, será o usado o banco de dados academia:

```
DROP SCHEMA IF EXISTS academia ;
```

```
CREATE SCHEMA IF NOT EXISTS academia ;
```

```
USE academia ;
```

```
CREATE TABLE IF NOT EXISTS academia.Endereco (
```

```
    Endereco_ID INT NOT NULL,
```

```
    Logradouro VARCHAR(100) NOT NULL,
```

```
Numero INT NOT NULL,  
  
Bairro VARCHAR(100) NOT NULL,  
  
CEP INT NOT NULL,  
  
PRIMARY KEY (Endereco_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Telefone (  
  
    Telefone_ID INT NOT NULL,  
  
    Telefone_Contato VARCHAR(45) NULL,  
  
    Telefone_Emergencia VARCHAR(45) NULL,  
  
    PRIMARY KEY (Telefone_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Plano (  
  
    Plano_ID INT NOT NULL,  
  
    Nome_plano VARCHAR(45) NOT NULL,  
  
    Valor DOUBLE NOT NULL,  
  
    PRIMARY KEY (Plano_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Horario (  
  
    Horario_ID INT NOT NULL,  
  
    Período VARCHAR(45) NOT NULL,  
  
    PRIMARY KEY (Horario_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Modalidade (  
  
    Modalidade_ID INT NOT NULL,  
  
    Nome_modalidade VARCHAR(45) NOT NULL,  
  
    PRIMARY KEY (Modalidade_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Pessoa_Associada (  
  
    Pessoa_Associada_ID INT NOT NULL AUTO_INCREMENT,  
  
    Nome VARCHAR(45) NOT NULL,  
  
    Sobrenome VARCHAR(45) NOT NULL,  
  
    Endereco_ID INT NOT NULL,  
  
    Telefone_ID INT NULL,  
  
    Plano_ID INT NOT NULL,  
  
    PRIMARY KEY (Pessoa_Associada_ID),  
  
    FOREIGN KEY (Endereco_ID)  
  
        REFERENCES academia.Endereco (Endereco_ID),  
  
    FOREIGN KEY (Telefone_ID)  
  
        REFERENCES academia.Telefone (Telefone_ID),  
  
    FOREIGN KEY (Plano_ID)  
  
        REFERENCES academia.Plano (Plano_ID)  
  
);
```

```
CREATE TABLE IF NOT EXISTS academia.Pessoa_Treinadora (  
  
    Pessoa_Treinadora_ID INT NOT NULL,  
  
    Nome VARCHAR(45) NOT NULL,  
  
    Sobrenome VARCHAR(45) NOT NULL,
```

```

Endereco_ID INT NOT NULL,

Telefone_ID INT NULL,

PRIMARY KEY (Pessoa_Treinadora_ID),

FOREIGN KEY (Endereco_ID)

REFERENCES academia.Endereco (Endereco_ID),

FOREIGN KEY (Telefone_ID)

REFERENCES academia.Telefone (Telefone_ID)

);

CREATE TABLE IF NOT EXISTS academia.Treino (

Pessoa_Associada_ID INT NOT NULL,

Modalidade_ID INT NOT NULL,

Horario_ID INT,

PRIMARY KEY (Pessoa_Associada_ID , Modalidade_ID),

FOREIGN KEY (Pessoa_Associada_ID)

REFERENCES academia.Pessoa_Associada (Pessoa_Associada_ID),

FOREIGN KEY (Modalidade_ID)

REFERENCES academia.Modalidade (Modalidade_ID),

FOREIGN KEY (Horario_ID)

REFERENCES academia.Horario (Horario_ID)

);

```

```

CREATE TABLE IF NOT EXISTS academia.Aula (

Pessoa_Treinadora_ID INT NOT NULL,

Modalidade_ID INT NOT NULL,

Horario_ID INT NOT NULL,

```

```

PRIMARY KEY (Pessoa_Treinadora_ID , Modalidade_ID , Horario_ID),

FOREIGN KEY (Pessoa_Treinadora_ID)

    REFERENCES academia.Pessoa_Treinadora (Pessoa_Treinadora_ID),

FOREIGN KEY (Horario_ID)

    REFERENCES academia.Horario (Horario_ID),

FOREIGN KEY (Modalidade_ID)

    REFERENCES academia.Modalidade (Modalidade_ID)

);

```

```

INSERT INTO academia.Plano ( Plano_ID, Nome_plano, Valor)

VALUES

    (1, 'Mensal', 90.00),

    (2, 'Trimestral', 210.00),

    (3, 'Semestral', 380.00),

    (4, 'Anual', 720.00);

```

```

INSERT INTO academia.Horario (Horario_ID, Período)

VALUES

    (1, 'Manhã'),

    (2, 'Tarde'),

    (3, 'Noite');

```

```

INSERT INTO academia.Modalidade (Modalidade_ID, Nome_modalidade)

VALUES

    (1, 'Musculação'),

    (2, 'Natação'),

```



```
(3, 'Dança'),  
(4, 'Boxe'),  
(5, 'Aeróbica');
```

```
INSERT INTO academia.Endereco (
```

```
Endereco_ID,
```

```
Logradouro,
```

```
Numero,
```

```
Bairro,
```

```
CEP
```

```
)
```

```
VALUES
```

```
(1, 'Rua Visconde de Parnaíba', 355, 'Brás', 03045000),  
(2, 'Travessa Capachos II', 659, 'Jardim Célia', 08191331),  
(3, 'Rua Domenico Venturi', 365, 'Jardim Patente', 04243040),  
(4, 'Rua Varadouro', 100, 'Jardim Soares', 03919040),  
(5, 'Rua Cardoso de Almeida', 175, 'Sumaré' , 01251001),  
(6, 'Rua Fábio Monte Negro', 928, 'Cidade Nova América', 04897500),  
(7, 'Praça Eduardo Berlendis', 237, 'Jardim Lourdes', 08452445),  
(8, 'Rua Doutor Antônio Ruggiero Júnior', 782, 'Jardim Monte Alegre',  
05165120),  
(9, 'Rua Antônio Romero Fernandes', 919, 'Vila Alpina', 03208050),  
(10, 'Rua Estado de Rondônia', 180, 'Jardim Cidade Pirituba',  
02945150),  
(11, 'Rua Júlio Brandão', 617, 'Jardim Porteira Grande', 03917040),  
(12, 'Rua Ipacaráí', 581, 'Perdizes', 05011020),
```

(13, 'Rua Professor Arnaldo João Semeraro', 941, 'Jardim Santa Emília', 04184900),

(14, 'Rua Carlos Motta Marchi', 994, 'Parque Bristol', 04193210),

(15, 'Rua Doutor Mello Nogueira', 639, 'Vila Baruel', 02510040),

(16, 'Rua José Maria Lisboa 480', 673, 'Jardim Paulista', 01423908),

(17, 'Rua Jaguapeba', 672, 'Conjunto Habitacional Fazenda do Carmo', 08421521),

(18, 'Rua José Henrique Tomaz de Lima', 958, 'Vila Verde', 08230850),

(19, 'Travessa Paulo Leitão', 717, 'Capão Redondo', 05866003),

(20, 'Rua Equador', 198, 'Jardim América', 01436080),

(21, 'Avenida Brigadeiro Luís Antônio', 452, 'Jardim Paulista', 01402002),

(22, 'Rua Doutor Raul Manso Sayão Filho', 688, 'Cidade São Mateus', 03965010),

(23, 'Rua Edgar Codazzi', 309, 'Cidade Antônio Estevão de Carvalho', 08225280),

(24, 'Rua Piracanjuba', 805, 'Vila Carbone', 02750040),

(25, 'Rua Canori', 910, 'Jardim Olinda', 05765170),

(26, 'Passagem Sebastião Festa', 870, 'Conjunto Promorar Estrada da Parada', 02873710),

(27, 'Avenida General Charles de Gaulle', 380, 'Parque São Domingos', 05124901),

(28, 'Rua Pepita de Fogo', 317, 'Jardim São Carlos (Zona Leste)', 08062740),

(29, 'Rua Paulo da Silva', 793, 'Jardim do Lago', 05397110),

(30, 'Praça Daniel Berciano Villasol', 291, 'Perdizes', 05009050),

(31, 'Rua Doutor Sílvio Dante Bertacchi', 154, 'Vila Sônia', 05625001);

INSERT INTO academia.Telefone (

Telefone_ID,

```

Telefone_Contato,

Telefone_Emergencia

)

VALUES

(1, '(11)997016542', '(11)36178405'),

(2, '(11)999124939', NULL),

(3, '(11)987492668', '(11)36215274'),

(4, '(11)996133401', '(11)38382770'),

(5, '(11)981875877', '(11)26640540'),

(6, '(11)991828746', '(11)26074255'),

(7, '(11)996702111', '(11)35659534'),

(8, '(11)988589105', NULL),

(9, '(11)993762497', '(11)28598239'),

(10, '(11)997629440', '(11)26449065'),

(11, '(11)982903669', '(11)29823025'),

(12, '(11)981651295', '(11)27846478'),

(13, '(11)29544808', NULL),

(14, '(11)981126496', '(11)26736144'),

(15, '(11)982591935', '(11)25415453'),

(16, '(11)993509407', '(11)25945608'),

(17, '(11)994293997', NULL),

(18, '(11)998975027', '(11)26611809'),

(19, '(11)993792683', '(11)26127452'),

(20, '(11)981055416', '(11)36162485'),

(21, '(11)996293237', '(11)35919556'),

(22, '(11)991352745', '(11)29929050'),

```

```

(23, '(11)993498573', '(11)38092873'),
(24, '(11)983345058', '(11)26853653'),
(25, '(11)985477714', '(11)39195446'),
(26, '(11)993912951', '(11)39126508'),
(27, '(11)996175387', '(11)39789749'),
(28, '(11)985190150', '(11)39287080'),
(29, '(11)989988795', '(11)38647391'),
(30, '(11)987760331', '(11)35953526'),
(31, '(11)986352698', '(11)27106247'),
(32, '(11)984650287', '(11)36072875'),
(33, '(11)985429620', '(11)28698585'),
(34, '(11)996005453', '(11)27739729'),
(35, '(11)982872235', '(11)28210058');

```

```

INSERT INTO academia.Pessoa_Associada (
    Pessoa_Associada_ID,
    Nome,
    Sobrenome,
    Endereco_ID,
    Telefone_ID,
    Plano_ID
)
VALUES
    (1, 'Ayla', 'Cunha', 9, 11, 1),
    (2, 'Catarina', 'Almeida', 10, 12, 1),

```

(3, 'Heitor', 'Rocha', 11, 13, 3),
(4, 'Noah', 'Soares', 12, 14, 3),
(5, 'Anderson', 'Silva', 13, 15, 3),
(6, 'Luiza', 'Ramos', 14, 16, 2),
(7, 'Bárbara', 'Soares', 12, 17, 1),
(8, 'Nair', 'Ramos', 15, 18, 2),
(9, 'André', 'Silva', 16, 19, 2),
(10, 'Yuri', 'Pereira', 17, 20, 1),
(11, 'José', 'Antunes', 18, NULL, 2),
(12, 'Betina', 'Texeira', 19, 22, 3),
(13, 'Antônia', 'Souza', 20, 23, 3),
(14, 'Antônio', 'Nunes', 21, 24, 4),
(15, 'Vitória', 'Santos', 22, 25, 1),
(16, 'Emily', 'Oliveira', 23, 26, 1),
(17, 'Reginaldo', 'Rocha', 24, 27, 2),
(18, 'Carla', 'Perez', 25, 28, 1),
(20, 'Joana', 'Gonçalves', 26, 29, 1),
(21, 'Paula', 'Costa', 27, 30, 3),
(22, 'Sonia', 'Nunes', 28, 31, 1),
(23, 'Francisco', 'Rodrigues', 29, 32, 1),
(24, 'Alessandra', 'Rodrigues', 29, 33, 1),
(25, 'Cauê', 'Costa', 30, 34, 2),
(26, 'Marcelo', 'Souza', 15, 21, 3),
(27, 'Angela', 'Pereira', 17, NULL, 1),
(29, 'Saulo', 'Gonçalves', 26, NULL, 3),
(30, 'Maria', 'Silva', 13, NULL, 1),

```
(31, 'Livia', 'Almeida', 10, NULL, 1),  
(32, 'Oswaldo', 'Garibaldi', 29, NULL, 3);
```

```
INSERT INTO academia.Pessoa_Treinadora (  
    Pessoa_Treinadora_ID,  
    Nome,  
    Sobrenome,  
    Endereco_ID,  
    Telefone_ID  
)
```

```
VALUES
```

```
(1, 'Jorge', 'Galvão', 1, 1),  
(2, 'Sophia', 'Galvão', 1, 2),  
(3, 'Camila', 'Cunha', 2, 3),  
(4, 'Mário', 'Peixoto', 3, 4),  
(5, 'Gustavo', 'Bernades', 4, 5),  
(6, 'Aline', 'Oliveira', 5, 6),  
(7, 'Beatriz', 'Oliveira', 6, 7),  
(8, 'Maya', 'Galvão', 1, 8),  
(9, 'Marcos', 'Fernades', 7, 9),  
(10, 'Vinicios', 'Morais', 8, 10),  
(11, 'Laura', 'Fernandes', 31, 35);
```

```
INSERT INTO academia.Aula (  

```

```
Pessoa_Treinadora_ID,  
  
Modalidade_ID,  
  
Horario_ID  
)
```

```
VALUES
```

```
(1, 1, 1),  
  
(6, 1, 2),  
  
(1, 1, 3),  
  
(2, 2, 1),  
  
(4, 2, 2),  
  
(2, 2, 3),  
  
(3, 3, 1),  
  
(10, 3, 2),  
  
(3, 3, 3),  
  
(5, 4, 1),  
  
(7, 4, 2),  
  
(5, 4, 3),  
  
(6, 5, 1),  
  
(8, 5, 2),  
  
(6, 5, 3);
```

```
INSERT INTO academia.Treino (  
  
Pessoa_Associada_ID,  
  
Modalidade_ID,  
  
Horario_ID  
)
```

VALUES

(1, 3, 1),
(2, 2, 3),
(3, 1, NULL),
(4, 1, 2),
(5, 4, NULL),
(6, 4, 1),
(7, 2, 3),
(8, 5, 1),
(9, 4, 2),
(10, 1, 2),
(11, 4, 3),
(12, 1, NULL),
(13, 5, 3),
(14, 1, 1),
(15, 2, 3),
(16, 3, 1),
(17, 4, 2),
(18, 3, NULL),
(20, 1, 3),
(21, 4, 1),
(22, 2, 2),
(23, 4, NULL),
(24, 2, 3),
(25, 2, NULL),
(26, 3, 3),


```
(27, 5, NULL),  
(29, 5, 3),  
(30, 5, 1),  
(31, 2, 3),  
(32, 1, 1);
```

EXERCÍCIOS

1 - Crie uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo das pessoas associadas a academia, uma coluna com o logradouro e outra com o número. Utilize as tabelas Pessoa_Associada e Endereco. ✓

2 - Escreva uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo das pessoas associadas a academia, uma coluna com o nome do plano dessa pessoa. Utilize as tabelas Pessoa_Associada e Plano. ✓

3 - Escreva uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo das pessoas associadas a academia, uma coluna com telefone e outra com o cep dessa pessoa. Utilize as tabelas Pessoa_Associada, Telefone e Endereco. ✓

4 - Escreva uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo das pessoas associadas a academia, uma coluna com telefone de contato e outra com o cep dessa pessoa. Utilize as tabelas Pessoa_Associada, Telefone e Endereco. REPETIDO COM A 3

5 - Crie uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo de todas as pessoas associadas a academia, e uma coluna com o telefone de emergência dessa pessoa, caso a pessoa não tiver telefone de emergência, exiba 'PEDENTE', nomeie essa coluna como 'Telefone de Emergência'. Utilize as tabelas Pessoa_Associada e Telefone. ✓

6 - Crie uma query que retorne duas colunas com os nomes das pessoas associadas que possuem o mesmo endereço. Porém, não deixe que o mesmo nome se repita na mesma linha. ✓

7 - Crie uma query que retorne duas colunas com os nomes das pessoas treinadoras que possuem o mesmo endereço. Porém, não deixe que o mesmo nome se repita na mesma linha. ✓

8 - Crie uma query que retorne uma coluna com o nome dos planos e outra que com a quantidade de pessoas associadas que possuem esse plano, nomeie como 'Pessoas Inclusas'. ✓

9 - Crie uma query que retorne uma coluna nomeada como 'Nome Completo', com o nome completo das pessoas associadas a academia, e uma coluna com o período dessa pessoa. Utilize as tabelas Pessoa_Associada, Treino e Horario. ✓

10 - Crie uma query que retorne uma coluna nomeada como 'Nome Completo' com o nome completo de todas as pessoas associadas a academia e uma coluna com o nome 'Horario', mostrando o período dessa pessoa. Caso a pessoa não tiver período estipulado, exiba 'LIVRE'. Utilize as tabelas Pessoa_Associada, Aula e Horario. ✓

Bonus

Para resolver os exercícios bonus use o banco de dados academia.

1 - Mostre o nome da pessoa treinadora e a modalidade em que atua, de modo que a mesma informação não se repita na tabela. Use as tabelas Pessoa_Treinadora, Aula, Modalidade. ✓

2 - Retorne o nome da pessoa treinadora e o período em que atua. Utilize as tabelas Pessoa_Treinadora, Aula, Horario. ✓

3 - Retorne o nome de todas pessoas treinadoras e uma coluna com o nome 'Horario', mostrando o período em que a pessoa atua. Caso não tenha um horário estipulado, mostre 'EVENTUAL'. Por fim, ordene em forma alfabética-invertida pelo nome da pessoa. Utilize as tabelas Pessoa_Treinadora, Aula, Horario.



4 - Retorne o nome da modalidade e a quantidade de pessoas que a praticam, nomeie essa coluna como 'Pessoas' e ordena por essa coluna em ordem crescente. Utilize as Tabelas Modalidade e Treino.



5 - Retorne o nome da pessoa treinadora e a quantidade de pessoas associadas que participam de suas aulas, nomeie essa coluna como 'Pessoas'. Utilize as tabelas Pessoa_Treinadora, Treino, Aula.



6 - Escreva uma query que retorne o nome da pessoa associada e o nome da pessoa treinadora responsável pelo seu treino de acordo com a modalidade e o horário.