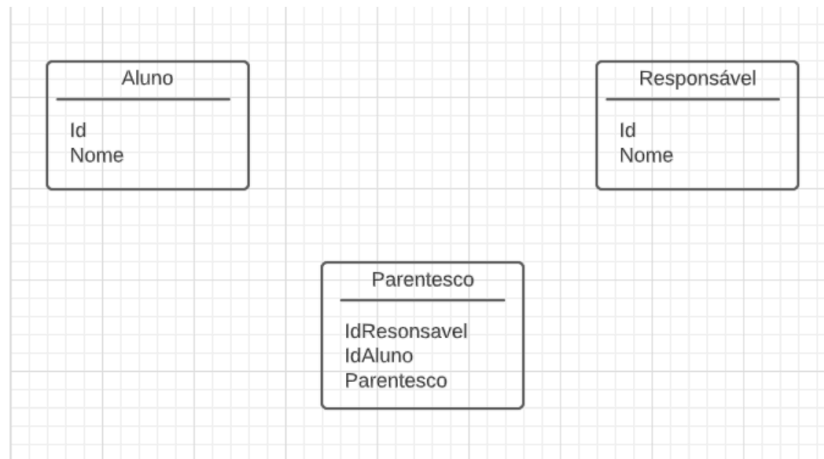


SQL



Perguntas:

- 1) Escreva comandos SQL para inserir os dados abaixo conforme o diagrama apresentado.
 - a) Pablo é Pai de Lucas
 - b) Brenda é Mãe de Lucas
- 2) Escreva uma consulta SQL para retornar dados únicos conforme tabela abaixo. Caso o aluno tenha mais de dois responsáveis, traga apenas os dois primeiros responsáveis encontrados na tabela.

Aluno	Responsável	Parentesco	Responsável	Parentesco

Bônus:

- 1) Escreva uma consulta SQL para trazer todos os dados. Seja criativo.

BANCO DE DADOS POSTGRE

```
CREATE TABLE Aluno (Id BIGSERIAL NOT NULL PRIMARY KEY, Nome VARCHAR(255));
CREATE TABLE Responsavel (Id BIGSERIAL NOT NULL PRIMARY KEY, Nome
VARCHAR(255));
CREATE TABLE Parentesco (
    IdResponsavel BIGINT NOT NULL,
    IdAluno BIGINT NOT NULL,
    Parentesco VARCHAR(100) NOT NULL,
    FOREIGN KEY (IdResponsavel) REFERENCES Responsavel(Id),
    FOREIGN KEY (IdAluno) REFERENCES Aluno
);
```

- Pergunta 1

```

INSERT INTO Aluno (Nome) VALUES ('Lucas');
INSERT INTO Responsavel (Nome) VALUES ('Pablo');
INSERT INTO Responsavel (Nome) VALUES ('Brenda');

```

```

INSERT INTO Parentesco (IdResponsavel, IdAluno, Parentesco) VALUES
(1, 1, 'Pai'),
(2, 2, 'Mãe');

```

- **Pergunta 2**

	Aluno character varying (255) 🔒	Responsável character varying 🔒	Parentesco character varying 🔒	Responsável character varying 🔒	Parentesco character varying 🔒
1	Lucas	Pablo	Pai	Brenda	Mãe
2	Eduardo	Brenda	Vó	Sem Responsável	Sem Responsável

```

SELECT
aluno.nome AS "Aluno",
COALESCE(
(SELECT Responsavel.nome FROM Responsavel
INNER JOIN parentesco ON parentesco.idResponsavel = Responsavel.id
WHERE parentesco.IdAluno = aluno.id LIMIT 1
), 'Sem Responsável'
)AS "Responsável",
COALESCE(
(SELECT parentesco.parentesco FROM parentesco
INNER JOIN Responsavel ON Responsavel.id = Parentesco.IdResponsavel
WHERE parentesco.idAluno = aluno.id LIMIT 1
), 'Sem Responsável'
)AS "Parentesco",
COALESCE(
(SELECT Responsavel.nome FROM Responsavel
INNER JOIN parentesco ON parentesco.idResponsavel = Responsavel.id
WHERE parentesco.IdAluno = aluno.id LIMIT 1 OFFSET 1
), 'Sem Responsável'
)AS "Responsável",
COALESCE(
(SELECT parentesco.parentesco FROM parentesco
INNER JOIN Responsavel ON Responsavel.id = parentesco.IdResponsavel
WHERE parentesco.idAluno = aluno.id LIMIT 1 OFFSET 1
), 'Sem Responsável'
)AS "Parentesco"

FROM aluno;

```

- **Bônus**

```

SELECT
  A.Id AS "Id Aluno",
  A.Nome AS "Nome Aluno",
  R.Id AS "Id Responsável",
  R.Nome AS "Nome Responsável",
  P.Parentesco As "Parentesco"
FROM
  Parentesco P
INNER JOIN
  Aluno A ON P.IdAluno = A.Id
INNER JOIN
  Responsavel R ON P.IdResponsavel = R.Id;

```

BANCO DE DADOS ORACLE

Também havia criado no banco de dados Oracle Express Edition, porém devido que não havia a função LIMIT e OFFSET ou NUMROW escolhi trocar para o banco de dados postgresql, assim subi dois containers docker um para o Banco de dados e outro para o Pgadmin. Assim tive que criar sequences.

```

INSERT INTO Aluno (Id, Nome) VALUES (SequenceAluno.NEXTVAL, 'Lucas');

```

```

INSERT INTO Responsavel (Id, Nome) VALUES (SequenceRespon.NEXTVAL, 'Pablo');
INSERT INTO Responsavel (Id, Nome) VALUES (SequenceRespon.NEXTVAL, 'Brenda');

```

```

INSERT INTO Parentesco (IdResponsavel, IdAluno, Parentesco) VALUES (1, 1, 'Pai');
INSERT INTO Parentesco (IdResponsavel, IdAluno, Parentesco) VALUES (2, 1, 'Mãe');

```