

Relatório do 1º Trabalho de Base de Dados:**1. Relações da Base de Dados:**

- membro(Nome, IdMemb, País, DataNasc):
Chave(s) Candidata(s) - {IdMemb}.
Chave primária - {IdMemb}.
Chave(s) Estrangeira(s) - Não existem chaves estrangeiras nesta relação.
- doce(Nome, Descrição, Género):
Chave(s) Candidata(s) - {Nome}.
Chave primária - {Nome}.
Chave(s) Estrangeira(s) - Não existem chaves estrangeiras nesta relação.
- amigo(Membro1, Membro2):
Chave(s) Candidata(s) - {Membro1, Membro2}; {Membro2, Membro1}.
Chave primária - {Membro1, Membro2}; {Membro2, Membro1}.
Chave(s) Estrangeira(s) - {Membro1} e {Membro2} referentes ao atributo {IdMemb} da relação membro.
- criou(Membro, Doce):
Chave(s) Candidata(s) - {Membro, Doce}.
Chave primária - {Membro, Doce}.
Chave(s) Estrangeira(s) - {Membro} referente ao atributo {IdMemb} da relação membro e {Doce} referente ao atributo {Nome} da relação doce.
- fez(Membro, Doce, Tempo, Aspeto, Sabor):
Chave(s) Candidata(s) - {Membro, Doce}.
Chave primária - {Membro, Doce}.
Chave(s) Estrangeira(s) - {Membro} referente ao atributo {IdMemb} da relação membro e {Doce} referente ao atributo {Nome} da relação doce.
- ingrediente(Nome, Custo):
Chave(s) Candidata(s) - {Nome}.
Chave primária - {Nome}.
Chave(s) Estrangeira(s) - Não existem chaves estrangeiras nesta relação.
- temIngrediente(Doce, Ingrediente, Quantidade):
Chave(s) Candidata(s) - {Doce, Ingrediente}.
Chave primária - {Doce, Ingrediente}.
Chave(s) Estrangeira(s) - {Doce} referente ao atributo {Nome} da relação doce e {Ingrediente} referente ao atributo {Nome} da relação ingrediente.

2. Constituição da Base de Dados:

- Criação da tabela da relação membro(Nome, IdMemb, País, DataNasc):
CREATE TABLE membro
(Nome VARCHAR(50) NOT NULL,
IdMemb INT UNIQUE CONSTRAINT IdMembKey PRIMARY KEY,
País VARCHAR(50) NOT NULL,
DataNasc DATE NOT NULL);
- Criação da tabela da relação doce(Nome, Descrição, Género):
CREATE TABLE doce
(Nome VARCHAR(50) UNIQUE CONSTRAINT NomeDoceKey PRIMARY KEY,
Descricao TEXT NOT NULL,
Genero VARCHAR(50) NOT NULL);

- Criação da tabela da relação amigo(Membro1, Membro2):

```
CREATE TABLE amigo
(Membro1 INT NOT NULL,
Membro2 INT NOT NULL,
CONSTRAINT amigoKey PRIMARY KEY (Membro1, Membro2),
CONSTRAINT Membro1Key FOREIGN KEY (Membro1) REFERENCES membro(IdMemb),
CONSTRAINT Membro2Key FOREIGN KEY (Membro2) REFERENCES membro(IdMemb));
```
- Criação da tabela da relação criou(Membro, Doce):

```
CREATE TABLE criou
(Membro INT NOT NULL,
Doce VARCHAR(50) NOT NULL,
CONSTRAINT criouKey PRIMARY KEY (Membro, Doce),
CONSTRAINT criouMembroKey FOREIGN KEY (Membro) REFERENCES membro(IdMemb),
CONSTRAINT criouDoceKey FOREIGN KEY (Doce) REFERENCES doce(Nome));
```
- Criação da tabela da relação fez(Membro, Doce, Tempo, Aspeto, Sabor):

```
CREATE TABLE fez
(Membro INT NOT NULL,
Doce VARCHAR(50) NOT NULL,
Tempo INT NOT NULL CHECK (Tempo BETWEEN 1 AND 5),
Aspeto INT NOT NULL CHECK (Aspeto BETWEEN 1 AND 5),
Sabor INT NOT NULL CHECK (Sabor BETWEEN 1 AND 5),
CONSTRAINT fezKey PRIMARY KEY (Membro, Doce),
CONSTRAINT fezMembroKey FOREIGN KEY (Membro) REFERENCES membro(IdMemb),
CONSTRAINT fezDoceKey FOREIGN KEY (Doce) REFERENCES doce(Nome));
```
- Criação da tabela da relação ingrediente(Nome, Custo):

```
CREATE TABLE ingrediente
(Nome VARCHAR(50) CONSTRAINT ingredienteKey PRIMARY KEY,
Custo DECIMAL CHECK (Custo > 0));
```
- Criação da tabela da relação temIngrediente(Doce, Ingrediente, Quantidade):

```
CREATE TABLE temIngrediente
(Doce VARCHAR(50) NOT NULL,
Ingrediente VARCHAR(50) NOT NULL,
Quantidade INT NOT NULL CHECK (Quantidade > 0),
CONSTRAINT temIngredienteKey PRIMARY KEY (Doce, Ingrediente),
CONSTRAINT temIngredienteDoceKey FOREIGN KEY (Doce) REFERENCES doce(Nome),
CONSTRAINT temIngredienteIngKey FOREIGN KEY (Ingrediente) REFERENCES ingrediente(Nome));
```

3. Inserção dos Valores Iniciais da Base de Dados:

a) Inserção dos 10 primeiros membros:

```
INSERT INTO membro VALUES ('Miguel', 1, 'Portugal', '2004-11-07');
INSERT INTO membro VALUES ('Leonor', 2, 'Portugal', '2006-03-01');
INSERT INTO membro VALUES ('Catarina', 3, 'Portugal', '1998-12-12');
INSERT INTO membro VALUES ('Louis', 4, 'França', '2000-07-06');
INSERT INTO membro VALUES ('Melissa', 5, 'Itália', '2005-10-18');
INSERT INTO membro VALUES ('Maria', 6, 'Portugal', '1977-07-21');
INSERT INTO membro VALUES ('Lamine', 7, 'Espanha', '2003-05-03');
INSERT INTO membro VALUES ('Thomas', 8, 'Alemanha', '1990-01-13');
INSERT INTO membro VALUES ('Jasmine', 9, 'Inglaterra', '2002-02-08');
INSERT INTO membro VALUES ('João', 10, 'Portugal', '2004-11-17');
```

b) Inserção dos 13 ingredientes seleccionados:

```
INSERT INTO ingrediente VALUES ('Açúcar', 0.01);           --PREÇO POR GRAMA
INSERT INTO ingrediente VALUES ('Farinha', 0.01);         --PREÇO POR GRAMA
```

INSERT INTO ingrediente VALUES ('Chocolate', 0.01);	--PREÇO POR GRAMA
INSERT INTO ingrediente VALUES ('Ovos', 0.23);	--PREÇO POR UNIDADE
INSERT INTO ingrediente VALUES ('Natas', 0.01);	--PREÇO POR GRAMA/MILILITRO
INSERT INTO ingrediente VALUES ('Leite', 0.01);	--PREÇO POR GRAMA/MILILITRO
INSERT INTO ingrediente VALUES ('Água', 0.01);	--PREÇO POR GRAMA/MILILITRO
INSERT INTO ingrediente VALUES ('Maçã', 0.36);	--PREÇO POR UNIDADE
INSERT INTO ingrediente VALUES ('Manteiga', 0.09);	--PREÇO POR GRAMA
INSERT INTO ingrediente VALUES ('Alfarroba', 0.01);	--PREÇO POR GRAMA
INSERT INTO ingrediente VALUES ('Baunilha', 6.99);	--PREÇO POR COLHERES DE CAFÉ
INSERT INTO ingrediente VALUES ('Canela', 0.01);	--PREÇO POR COLHERES DE CAFÉ
INSERT INTO ingrediente VALUES ('Pimenta', 0.02);	--PREÇO POR COLHERES DE CAFÉ

c) Inserção dos 10 doces e dados de criação:

```

INSERT INTO doce VALUES ('Bolo de Chocolate', 'Bolo rico, feito com chocolate derretido.', 'Tradicional');
INSERT INTO doce VALUES ('Bolo de Maçã', 'Bolo macio, com pedaços de maçã e canela.', 'Tradicional');
INSERT INTO doce VALUES ('Pudim de Leite', 'Pudim cremoso, sabor suave de leite', 'Tradicional');
INSERT INTO doce VALUES ('Tarte de Natas', 'Tarte crocante, recheio cremoso de natas.', 'Tradicional');
INSERT INTO doce VALUES ('Pudim Flan de Chocolate', 'Pudim de chocolate, textura suave e cremosa',
'Tradicional');
INSERT INTO doce VALUES ('Biscoitos de Canela', 'Biscoitos crocantes, sabor intenso de canela.',
'Regional');
INSERT INTO doce VALUES ('Bolo de Alfarroba', 'Bolo úmido, sabor adocicado de alfarroba.',
'Tradicional');
INSERT INTO doce VALUES ('Torta de Maçã', 'Torta recheada, maçã com cobertura crocante.',
'Tradicional');
INSERT INTO doce VALUES ('Pudim de Pimenta', 'Pudim levemente picante, toque de pimenta.',
'Experimental');
INSERT INTO doce VALUES ('Pavê de Maçã', 'Camadas de maçã e creme saboroso.', 'Regional');

```

```

INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Açúcar', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Farinha', 150);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Chocolate', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Ovos', 3);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Manteiga', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Natas', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Chocolate', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Açúcar', 150);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Farinha', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Ovos', 2);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Manteiga', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Maçã', 2);
INSERT INTO temIngrediente VALUES ('Bolo de Maçã', 'Canela', 1);
INSERT INTO temIngrediente VALUES ('Pudim de Leite', 'Açúcar', 200);
INSERT INTO temIngrediente VALUES ('Pudim de Leite', 'Leite', 500);
INSERT INTO temIngrediente VALUES ('Pudim de Leite', 'Ovos', 3);
INSERT INTO temIngrediente VALUES ('Pudim de Leite', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Tarte de Natas', 'Açúcar', 150);
INSERT INTO temIngrediente VALUES ('Tarte de Natas', 'Natas', 500);
INSERT INTO temIngrediente VALUES ('Tarte de Natas', 'Farinha', 200);
INSERT INTO temIngrediente VALUES ('Tarte de Natas', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Pudim Flan de Chocolate', 'Açúcar', 150);
INSERT INTO temIngrediente VALUES ('Pudim Flan de Chocolate', 'Natas', 300);
INSERT INTO temIngrediente VALUES ('Pudim Flan de Chocolate', 'Leite', 200);
INSERT INTO temIngrediente VALUES ('Pudim Flan de Chocolate', 'Chocolate', 100);

```

```

INSERT INTO temIngrediente VALUES ('Pudim Flan de Chocolate', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Açúcar', 100);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Farinha', 250);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Manteiga', 125);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Canela', 1);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Biscoitos de Canela', 'Ovos', 1);
INSERT INTO temIngrediente VALUES ('Bolo de Alfarroba', 'Açúcar', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Alfarroba', 'Farinha', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Alfarroba', 'Ovos', 3);
INSERT INTO temIngrediente VALUES ('Bolo de Alfarroba', 'Manteiga', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Alfarroba', 'Alfarroba', 100);
INSERT INTO temIngrediente VALUES ('Torta de Maçã', 'Açúcar', 100);
INSERT INTO temIngrediente VALUES ('Torta de Maçã', 'Farinha', 200);
INSERT INTO temIngrediente VALUES ('Torta de Maçã', 'Maçã', 3);
INSERT INTO temIngrediente VALUES ('Torta de Maçã', 'Canela', 1);
INSERT INTO temIngrediente VALUES ('Torta de Maçã', 'Manteiga', 100);
INSERT INTO temIngrediente VALUES ('Pavê de Maçã', 'Açúcar', 150);
INSERT INTO temIngrediente VALUES ('Pavê de Maçã', 'Natas', 300);
INSERT INTO temIngrediente VALUES ('Pavê de Maçã', 'Maçã', 2);
INSERT INTO temIngrediente VALUES ('Pavê de Maçã', 'Canela', 1);
INSERT INTO temIngrediente VALUES ('Pudim de Pimenta', 'Açúcar', 150);
INSERT INTO temIngrediente VALUES ('Pudim de Pimenta', 'Ovos', 3);
INSERT INTO temIngrediente VALUES ('Pudim de Pimenta', 'Leite', 500);
INSERT INTO temIngrediente VALUES ('Pudim de Pimenta', 'Pimenta', 1);

```

```

INSERT INTO criou VALUES (1, 'Bolo de Chocolate');
INSERT INTO criou VALUES (1, 'Bolo de Maçã');
INSERT INTO criou VALUES (2, 'Pudim de Leite');
INSERT INTO criou VALUES (2, 'Tarte de Natas');
INSERT INTO criou VALUES (3, 'Pudim Flan de Chocolate');
INSERT INTO criou VALUES (3, 'Biscoitos de Canela');
INSERT INTO criou VALUES (4, 'Bolo de Alfarroba');
INSERT INTO criou VALUES (4, 'Torta de Maçã');
INSERT INTO criou VALUES (5, 'Pudim de Pimenta');
INSERT INTO criou VALUES (5, 'Pavê de Maçã');

```

```

INSERT INTO fez VALUES (1, 'Bolo de Chocolate', 1, 5, 4);
INSERT INTO fez VALUES (6, 'Bolo de Chocolate', 1, 4, 4);
INSERT INTO fez VALUES (7, 'Bolo de Chocolate', 1, 3, 4);
INSERT INTO fez VALUES (5, 'Bolo de Maçã', 3, 4, 3);
INSERT INTO fez VALUES (6, 'Bolo de Maçã', 4, 4, 4);
INSERT INTO fez VALUES (6, 'Pudim de Leite', 5, 5, 5);
INSERT INTO fez VALUES (7, 'Pudim de Leite', 2, 2, 2);
INSERT INTO fez VALUES (7, 'Tarte de Natas', 1, 1, 1);
INSERT INTO fez VALUES (8, 'Tarte de Natas', 2, 3, 4);
INSERT INTO fez VALUES (8, 'Pudim Flan de Chocolate', 3, 2, 5);
INSERT INTO fez VALUES (9, 'Pudim Flan de Chocolate', 3, 2, 3);
INSERT INTO fez VALUES (9, 'Biscoitos de Canela', 4, 5, 5);
INSERT INTO fez VALUES (10, 'Biscoitos de Canela', 4, 5, 4);
INSERT INTO fez VALUES (1, 'Bolo de Alfarroba', 2, 4, 2);
INSERT INTO fez VALUES (2, 'Bolo de Alfarroba', 4, 3, 4);
INSERT INTO fez VALUES (2, 'Torta de Maçã', 4, 1, 3);
INSERT INTO fez VALUES (3, 'Torta de Maçã', 3, 4, 4);

```

```

INSERT INTO fez VALUES (3, 'Pudim de Pimenta', 4, 3, 1);
INSERT INTO fez VALUES (4, 'Pudim de Pimenta', 4, 3, 4);
INSERT INTO fez VALUES (4, 'Pavê de Maçã', 4, 5, 4);
INSERT INTO fez VALUES (5, 'Pavê de Maçã', 3, 4, 5);

```

d) Inserção das amizades:

```

INSERT INTO amigo VALUES (1, 2);
INSERT INTO amigo VALUES (2, 1);
INSERT INTO amigo VALUES (1, 3);
INSERT INTO amigo VALUES (3, 1);
INSERT INTO amigo VALUES (1, 4);
INSERT INTO amigo VALUES (4, 1);
INSERT INTO amigo VALUES (1, 5);
INSERT INTO amigo VALUES (5, 1);
INSERT INTO amigo VALUES (1, 6);
INSERT INTO amigo VALUES (6, 1);
INSERT INTO amigo VALUES (1, 7);
INSERT INTO amigo VALUES (7, 1);
INSERT INTO amigo VALUES (1, 8);
INSERT INTO amigo VALUES (8, 1);
INSERT INTO amigo VALUES (1, 9);
INSERT INTO amigo VALUES (9, 1);
INSERT INTO amigo VALUES (1, 10);
INSERT INTO amigo VALUES (10, 1);

INSERT INTO amigo VALUES (2, 3);
INSERT INTO amigo VALUES (3, 2);
INSERT INTO amigo VALUES (2, 4);
INSERT INTO amigo VALUES (4, 2);
INSERT INTO amigo VALUES (4, 5);
INSERT INTO amigo VALUES (5, 4);
INSERT INTO amigo VALUES (5, 6);
INSERT INTO amigo VALUES (6, 5);
INSERT INTO amigo VALUES (6, 7);
INSERT INTO amigo VALUES (7, 6);

```

e) Inserção dos dados complementares necessários para o exercício seguinte:

- Exercício 4 alínea a):


```

INSERT INTO doce VALUES ('Bolo de Canela e Chocolate', 'Combinação do sabor intenso do chocolate com o toque aromático da canela.', 'Regional');
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Açúcar', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Farinha', 200);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Chocolate', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Ovos', 3);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Manteiga', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Natas', 100);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Canela', 1);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Baunilha', 1);
INSERT INTO temIngrediente VALUES ('Bolo de Canela e Chocolate', 'Leite', 50);
INSERT INTO criou VALUES (7, 'Bolo de Canela e Chocolate');

```
- Exercício 4 alínea b):


```

INSERT INTO membro VALUES ('Joaquim José', 11, 'Portugal', '1980-06-01');
INSERT INTO amigo VALUES (1, 11);
INSERT INTO amigo VALUES (11, 1);

```

INSERT INTO amigo VALUES (2, 11);

INSERT INTO amigo VALUES (11, 2);

- Exercício 4 alínea g):

INSERT INTO doce VALUES ('Arroz Doce', 'Arroz doce cremoso com canela.', 'Tradicional');

INSERT INTO ingrediente VALUES ('Arroz', 0.01);

INSERT INTO ingrediente VALUES ('Limão', 0.99);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Arroz', 200);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Leite', 1000);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Açúcar', 200);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Ovos', 2);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Canela', 1);

INSERT INTO temIngrediente VALUES ('Arroz Doce', 'Limão', 1);

- Exercício 4 alínea j):

INSERT INTO criou VALUES (1, 'Arroz Doce');

- Exercício 4 alínea k):

INSERT INTO fez VALUES (1, 'Bolo de Canela e Chocolate', 5, 5, 5);

INSERT INTO fez VALUES (10, 'Bolo de Canela e Chocolate', 5, 5, 5);

4. Resolução das questões em Álgebra Relacional e SQL:

a) Álgebra Relacional:

$$\pi_{\text{membro.Nome}} (\sigma_{\text{membro.Pais}='Espanha' \wedge \text{doce.Genero}='Regional' \wedge \text{membro.IdMemb}=\text{criou.Membro} \wedge$$
$$\wedge \text{criou.Doce}=\text{doce.Nome} \wedge \text{doce.Nome}=\text{temIngrediente.Doce} \wedge \text{temIngrediente.Ingrediente}='Chocolate' (\text{membro} \times \text{criou} \times$$
$$\times \text{doce} \times \text{temIngrediente}))$$
$$\cap$$
$$\pi_{\text{membro.Nome}} (\sigma_{\text{membro.Pais}='Espanha' \wedge \text{doce.Genero}='Regional' \wedge \text{membro.IdMemb}=\text{criou.Membro} \wedge$$
$$\wedge \text{criou.Doce}=\text{doce.Nome} \wedge \text{doce.Nome}=\text{temIngrediente.Doce} \wedge \text{temIngrediente.Ingrediente}='Canela' (\text{membro} \times \text{criou} \times$$
$$\times \text{doce} \times \text{temIngrediente}))$$

SQL:

```
SELECT DISTINCT membro.Nome
FROM membro, criou, doce, temIngrediente
WHERE membro.Pais='Espanha'
      AND doce.Genero='Regional'
      AND membro.IdMemb=criou.Membro
      AND criou.Doce=doce.Nome
      AND doce.Nome=temIngrediente.Doce
      AND temIngrediente.Ingrediente='Chocolate'
INTERSECT
SELECT DISTINCT membro.Nome
FROM membro, criou, doce, temIngrediente
WHERE membro.Pais='Espanha'
      AND doce.Genero='Regional'
      AND membro.IdMemb=criou.Membro
      AND criou.Doce=doce.Nome
      AND doce.Nome=temIngrediente.Doce
      AND temIngrediente.Ingrediente='Canela';
```

b) Álgebra Relacional:

$$\pi_{doce.Nome}(\sigma_{membro.Nome='Joaquim José' \wedge doce.Genero='Tradicional' \wedge fez.Aspetto \geq 5 \wedge (membro.IdMemb=amigo.Membro1 \vee membro.IdMemb=amigo.Membro2) \wedge (amigo.Membro1=criou.Membro \vee membro.IdMemb=amigo.Membro2) \wedge criou.Doce=fez.Doce \wedge doce.Nome=criou.Doce}(membro \times amigo \times criou \times fez \times doce))$$

SQL:

```
SELECT DISTINCT doce.Nome
FROM membro, amigo, criou, fez, doce
WHERE membro.Nome='Joaquim José'
      AND doce.Genero='Tradicional'
      AND fez.Aspetto >= 5
      AND (membro.IdMemb=amigo.Membro1 OR membro.IdMemb=amigo.Membro2)
      AND (amigo.Membro1=criou.Membro OR amigo.Membro2=criou.Membro)
      AND criou.Doce=fez.Doce
      AND doce.Nome=criou.Doce;
```

c) Álgebra Relacional:

SQL:

```
SELECT DISTINCT membro.Nome
FROM membro
WHERE NOT EXISTS
  (SELECT 1
   FROM fez
   WHERE fez.Membro = membro.IdMemb AND fez.Doce NOT IN
     (SELECT criou.Doce
      FROM criou
      WHERE criou.Membro IN
        (SELECT amigo.Membro2
         FROM amigo
         WHERE amigo.Membro1 = membro.IdMemb)));
```

d) Álgebra Relacional:

$$\pi_{fez.Doce}(\sigma_{temIngrediente.Ingrediente='Natas'}(fez \times temIngrediente)) - \pi_{fez.Doce}(\sigma_{temIngrediente.Ingrediente='Natas'}(fez \times temIngrediente))$$

SQL:

```
SELECT DISTINCT fez.Doce
FROM fez, temIngrediente
WHERE temIngrediente.Ingrediente='Natas'
      AND temIngrediente.Doce=fez.Doce
EXCEPT
SELECT DISTINCT fez.Doce
FROM fez, temIngrediente
WHERE fez.Sabor=5
      AND temIngrediente.Doce=fez.Doce;
```

e) Álgebra Relacional:

$$\pi_{\text{membro.Nome}} (\sigma_{\text{temIngrediente.Ingrediente}='Canela' \wedge \text{temIngrediente.Doce}=\text{criou.Doce} \wedge \text{criou.Membro}=\text{membro.IdMemb}} (\text{membro X temIngrediente X criou}))$$
$$\cap$$
$$\pi_{\text{membro.Nome}} (\sigma_{\text{temIngrediente.Ingrediente}='Canela' \wedge \text{temIngrediente.Doce}=\text{criou.Doce} \wedge \text{criou.Membro}=\text{membro.IdMemb}} (\text{membro X temIngrediente X criou}))$$

SQL:

```
SELECT DISTINCT membro.Nome
FROM membro, temIngrediente, criou
WHERE temIngrediente.Ingrediente='Canela'
      AND temIngrediente.Doce=criou.Doce
      AND criou.Membro=membro.IdMemb
INTERSECT
SELECT DISTINCT membro.Nome
FROM membro, temIngrediente, criou
WHERE temIngrediente.Ingrediente='Leite'
      AND temIngrediente.Doce=criou.Doce
      AND criou.Membro=membro.IdMemb;
```

f) Álgebra Relacional:

$$\pi_{\text{doce.Nome}} (\sigma_{\text{temIngrediente.Ingrediente}='Maçã' \wedge \text{temIngrediente.Doce}=\text{doce.Nome}} (\text{temIngrediente X doce}))$$
$$\cup$$
$$\pi_{\text{doce.Nome}} (\sigma_{\text{temIngrediente.Ingrediente}='Baunilha' \wedge \text{temIngrediente.Doce}=\text{doce.Nome}} (\text{temIngrediente X doce}))$$

SQL:

```
SELECT DISTINCT doce.Nome
FROM temIngrediente, doce
WHERE temIngrediente.Ingrediente='Maçã'
      AND temIngrediente.Doce=Doce.Nome
UNION
SELECT DISTINCT doce.Nome
FROM temIngrediente, doce
WHERE temIngrediente.Ingrediente='Baunilha'
      AND temIngrediente.Doce=Doce.Nome;
```

g) Álgebra Relacional:

$$\gamma_{\text{SUM(ingrediente.Custo * temIngrediente.Quantidade)}} (\sigma_{\text{temIngrediente.Doce}='Arroz Doce' \wedge \text{temIngrediente.Ingrediente}=\text{ingrediente.Nome}} (\text{temIngrediente X ingrediente}))$$

SQL:

```
SELECT SUM(ingrediente.Custo * temIngrediente.Quantidade)
FROM temIngrediente, ingrediente
WHERE temIngrediente.Doce = 'Arroz Doce'
      AND temIngrediente.Ingrediente=ingrediente.Nome;
```

h) Álgebra Relacional:

$$\gamma_{\text{doce.Nome, SUM(ingrediente.Custo * temIngrediente.Quantidade)}} (\sigma_{\text{temIngrediente.Doce}=\text{doce.Nome} \wedge \text{temIngrediente.Ingrediente}=\text{ingrediente.Nome}} (\text{temIngrediente X ingrediente X doce}))$$

SQL:

```
SELECT SUM(ingrediente.Custo * temIngrediente.Quantidade)
FROM temIngrediente, ingrediente, doce
WHERE temIngrediente.Doce = doce.Nome
      AND temIngrediente.Ingrediente=ingrediente.Nome
GROUP BY doce.Nome;
```


i) Álgebra Relacional:

$$\gamma_{\text{membro.Nome, COUNT(criou.Doce)}} (\sigma_{\text{criou.Membro=Membro.IdMemb}} (\text{membro X criou}))$$

SQL:

```
SELECT membro.Nome, COUNT(criou.Doce)
FROM membro, criou
WHERE criou.Membro=Membro.IdMemb
GROUP BY membro.Nome;
```

j) Álgebra Relacional:

$$\gamma_{\text{membro.Nome, COUNT(criou.Doce)}} (\sigma_{\text{criou.Membro=Membro.IdMemb}} (\text{membro X criou}))$$

SQL:

```
SELECT membro.Nome, COUNT(criou.Doce) AS Doces_Criados
FROM membro, criou
WHERE membro.IdMemb = criou.Membro
GROUP BY membro.Nome
ORDER BY Doces_Criados DESC LIMIT 1;
```

k) Álgebra Relacional:

$$\gamma_{\text{doce.Nome, SUM(ingrediente.Custo * temIngrediente.Quantidade)}} (\sigma_{\text{temIngrediente.Doce=doce.Nome} \wedge \text{temIngrediente.Ingrediente=ingrediente.Nome}} (\text{temIngrediente X ingrediente X doce}))$$

SQL:

```
SELECT doce.Nome, SUM(temIngrediente.Quantidade * ingrediente.Custo)
FROM doce, temIngrediente, ingrediente
WHERE temIngrediente.Ingrediente = ingrediente.Nome
    AND temIngrediente.Doce = doce.Nome
GROUP BY doce.Nome
ORDER BY SUM(temIngrediente.Quantidade * ingrediente.Custo) DESC LIMIT 1;
```

l)

m) Álgebra Relacional:

$$\pi_{\text{membro.Nome}} (\sigma_{\text{membro.IdMemb=fez.Membro}} (\text{membro X fez})) \div \\ \div \pi_{\text{membro.Nome}} (\sigma_{\text{membro.IdMemb=fez.Membro} \wedge \text{temIngrediente.Doce=fez.Doce} \wedge \text{temIngrediente.Ingrediente='Baunilha'}} (\text{membro X fez X temIngrediente}))$$

SQL:

```
SELECT membro.Nome
FROM membro
WHERE NOT EXISTS (
    SELECT Doce
    FROM temIngrediente
    WHERE Ingrediente = 'Baunilha'
    EXCEPT
    SELECT fez.Doce
    FROM fez
    WHERE fez.Membro = membro.idMemb);
```

n) Álgebra Relacional:

$$\pi_{\text{Doce}}(\sigma_{\text{Tempo}=1}(\text{fez})) - \pi_{\text{Doce}}(\sigma_{\text{Tempo} \neq 1}(\text{fez}))$$

SQL:

```
SELECT DISTINCT Doce
FROM fez
WHERE Tempo=1
EXCEPT
SELECT DISTINCT Doce
FROM fez
WHERE Tempo<>1;
```

o)