**Uma imagem com eletrónica

Descrição gerada automaticamente**

****

ANO LETIVO 2020/2021

RELATÓRIO DO 2º TRABALHO (1ª Parte)

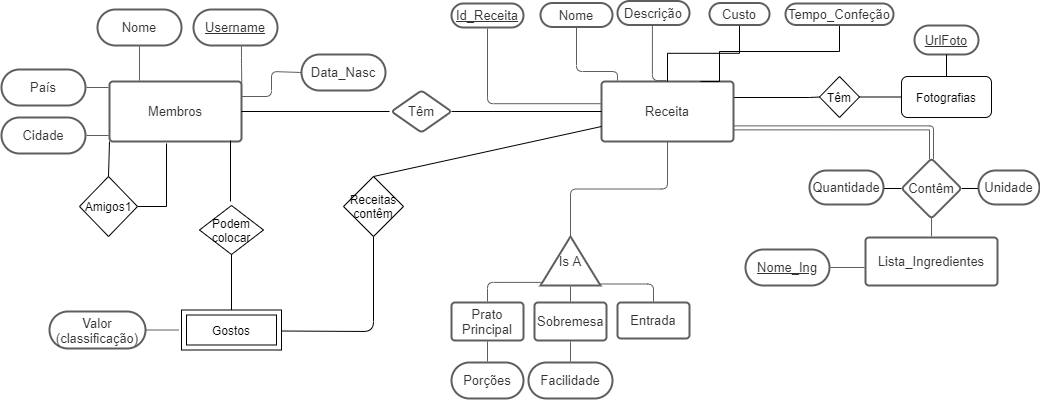
BASE DE DADOS

**Alunos:** Diogo Castanho Nº42496

Pedro Grilo Nº43012

**Disciplina:** Base de dados

**Diagrama E-R**

****

**Relações**

1. Membros(username, nome, data\_nasc, país, cidade)
2. Receita(id\_receita, Nome, Descrição, Custo, Tempo\_Confeção)
3. Lista\_Ingredientes(Nome\_Ing, Id\_Receita)
4. Fotografias(UrlFoto, Id\_Receita)
5. Gostos(Username, IdReceita, Valor)
6. Prato Principal(Id\_Receita, Porções)
7. Sobremesa(Id\_Receita, Facilidade)
8. Receitas\_Contêm(Id\_Receita, Username)
9. Amigos1(Username, Username2)
10. Têm(Membros têm)(Username, Id\_Receita)
11. Receita-> Têm (Id\_Receita, UrlFoto)
12. Contêm(Id\_Receita, Quantidade,Unidade, Nome\_Ing)
13. Podem\_Colocar(Username, Valor)

**Chaves**

**membro(Nome, username , PaisM, CidadeM, DataNasc)**

CHAVES PRIMARIAS: {Username}

CHAVES ESTRANGEIRA: {}

**receita(NomeR, IdReceita, Descrição, Tipo, N\_Porcoes, Facilidade, TempoDeConfecao, Custo, Nivel)**

CHAVES PRIMARIAS: {IdReceita}

CHAVES ESTRANGEIRAS: {}

**fotografias(IdReceita, UrlFoto)**

CHAVES PRIMARIAS: {IdReceita, UrlFoto}

CHAVES ESTRANGEIRAS: {IdReceita} da relação receita

**lista\_ingredientes(IdReceita, NomeIngrediente)**

CHAVES PRIMARIAS: {IdReceita, NomeIngrediente}

CHAVES ESTRANGEIRAS: {IdReceita} da relação receita

**Faz\_receita(username, IdReceita) (têm)**

CHAVES PRIMARIAS: {username, IdReceita}

CHAVES ESTRANGEIRAS: {IdReceita} da relação receita e {username} da relação membro

**Amigos1(username1, username2)**

CHAVES PRIMARIAS: {username1, username2}

CHAVES ESTRANGEIRAS: {Username1} da relação membros

**(Continuação - Relações)**

**gostos(username, IdReceita, Valor)**

CHAVES PRIMARIAS:{username, IdReceita}

CHAVES ESTRANGEIRAS: {username} da relação membro e {IdReceita} da relação receita

**Podem\_colocar(username, Valor)**

CHAVES PRIMARIAS:{username}

CHAVES ESTRANGEIRAS: {username} da relação membro

**Prato principal(Id\_Receita, Porções)**

CHAVES PRIMARIAS:{Id\_Receita}

CHAVES ESTRANGEIRAS: {Id\_Receita} da relação receita

**sobremesa(Id\_Receita, Facilidade)**

CHAVES PRIMARIAS:{Id\_Receita}

CHAVES ESTRANGEIRAS: {Id\_Receita} da relação receita

**entrada(Id\_Receita)**

CHAVES PRIMARIAS:{Id\_Receita}

CHAVES ESTRANGEIRAS: {Id\_Receita} da relação receita

**Têm(receita -> fotografias) (Id\_Receita, UrlFoto)**

CHAVES PRIMARIAS:{Id\_Receita, UrlFoto}

CHAVES ESTRANGEIRAS: {Id\_Receita} da relação receita e {Url\_Foto} da relação fotografias

**(Continuação - Relações)**

**Contêm (Id\_Receita, Quantidade,Unidade, Nome\_Ing)**

CHAVES PRIMARIAS:{Id\_Receita, Nome\_Ing}

CHAVES ESTRANGEIRAS: {Id\_Receita} da relação receita e {Nome\_Ing} da relação lista\_ingredientes

**Criação de tabelas SQL**

Create table membro (

Nome varchar(30),

username varchar(15) primary key,

PaisM varchar(20),

CidadeM varchar(20),

DataNasc char(14)

);

Create table receita (

NomeR varchar(50),

IdReceita char(3) primary key,

Descrição varchar(200),

Tipo varchar(15),

N\_Porcoes char(1),

Facilidade varchar(10),

TempoDeConfecao varchar(10),

Custo char(3),

Nível varchar(15)

);

**(Continuação - Criação de tabelas SQL)**

Create table fotografias (

IdReceita char(3),

UrlFoto varchar(30),

primary key(IdReceita, UrlFoto),

foreign key (IdReceita) references receita on delete restrict

);

Create table lista\_ingredientes(

IdReceita char(3),

NomeIngrediente varchar(40),

Primary Key(IdReceita,NomeIngrediente),

foreign key (IdReceita) references receita on delete restrict

);

Create table faz\_receita(

username varchar(15),

IdReceita char(3),

PRIMARY KEY(username, IdReceita),

foreign key (username) references membro on delete restrict,

foreign key (IdReceita) references receita on delete restrict

);

Create table amigos1(

username1 varchar(15),

username2 varchar(15),

PRIMARY KEY(username1, username2)

);

**(Continuação - Criação de tabelas SQL)**

Create table gostos (

username varchar(15),

IdReceita char(3),

Valor integer,

PRIMARY KEY(username, IdReceita),

foreign key (username) references membro on delete restrict,

foreign key (IdReceita) references receita on delete restrict

);

**Expressões SQL e Álgebra Relacional**

**a)** Quais as Receitas com Pato?

**SQL:**

***select*** IdReceita, NomeR

***from*** receita ***natural inner join*** lista\_ingredientes

***where*** NomeIngrediente ***like*** 'Pato'

**ÁLGEBRA RELACIONAL:**

ΠIdReceita,NomeR  (σ NomeIngrediente=’Pato’(receita ⟗ lista\_ingredientes))

**(Continuação – Expressões SQL e Álgebra Relacional)**

**b)** Quais as Receitas que não têm Pato?

**SQL:**

***select*** IdReceita, NomeR

***from*** receita

***except***

***select*** IdReceita, NomeR

***from*** receita ***natural inner join*** lista\_ingredientes

***where*** NomeIngrediente like 'Pato'

**ÁLGEBRA RELACIONAL:**

πIdReceita,NomeR  (receita) ÷ πIdReceita,NomeR (σ NomeIngrediente=’Pato’(receita ⟗ lista\_ingredientes))

**c)** Quais os membros que têm receitas com Pato?

**SQL:**

***select*** username

***from*** faz\_receita ***natural inner join*** membro ***natural inner join*** lista\_ingredientes

***where*** NomeIngrediente ***like*** 'Pato'

**ÁLGEBRA RELACIONAL:**

ΠUsername (σNomeIngrediente = ‘Pato’ (faz\_receita ⟗ membro ⟗ lista\_ingredientes))

**(Continuação – Expressões SQL e Álgebra Relacional)**

**d)** Quais os amigos dos membros que têm receitas com Pato?

**SQL:**

***select distinct*** username

***from*** membro

***where*** username ***in (select*** username1

***from*** amigos1

***where*** username2 ***in (select*** username

***from*** faz\_receita ***natural inner join*** membro ***natural inner join*** lista\_ingredientes

***where*** NomeIngrediente ***like*** 'Pato')***);***

**ÁLGEBRA RELACIONAL:**

ΠUsername (σusername (ΠUsername1(σusername2(ΠUsername(NomeIngrediente ***=*** 'Pato'(faz\_receita⟗membro⟗lista\_ingredientes))) (amigos1)))(membro))

**e)** Quais os membros que dão mais de 1 estrela a Receitas com Pato?

**SQL:**

***select distinct*** username

***from*** gostos ***natural inner join*** lista\_ingredientes

***where*** valor > 1 ***and*** NomeIngrediente ***like*** 'Pato'

**(Continuação – Expressões SQL e Álgebra Relacional)**

**ÁLGEBRA RELACIONAL:**

ΠUsername (σvalor > ‘1’ ∧ NomeIngrediente=’Pato’ (gostos ⟗ lista\_ingredientes))

**f)** Quais os membros que têm receitas com Ovos e Amêndoa?

**SQL:**

***select*** username, IdReceita

***from*** lista\_ingredientes ***natural inner join*** faz\_receita

***where*** NomeIngrediente ***like*** 'Ovos'

***intersect***

***select*** username, IdReceita

***from*** lista\_ingredientes ***natural inner join*** faz\_receita

***where*** NomeIngrediente ***like*** 'Amêndoa'

**ÁLGEBRA RELACIONAL:**

ΠUsername,IdReceita (σNomeIngrediente=’Ovos’ (faz\_receita ⟗ lista\_ingredientes)) ⋂ ΠUsername,IdReceita (σNomeIngrediente=’Amêndoa’ (faz\_receita ⟗ lista\_ingredientes))

**(Continuação – Expressões SQL e Álgebra Relacional)**

**g)** Quais os membros que têm receitas com Pato ou Perú?

**SQL:**

***select*** username, IdReceita, NomeIngrediente

***from*** lista\_ingredientes ***natural inner join*** faz\_receita

***where*** NomeIngrediente ***like*** 'Pato'

***union***

***select***  username, IdReceita, NomeIngrediente

***from*** lista\_ingredientes ***natural inner join*** faz\_receita

***where*** NomeIngrediente ***like*** 'Perú'

**ÁLGEBRA RELACIONAL:**

ΠUsername,IdReceita,NomeIngrediente(σNomeIngrediente=’Pato’(faz\_receita⟗lista\_ingredientes)) ⋃ ΠUsername,IdReceita,NomeIngrediente(σNomeIngrediente=’Perú’(faz\_receita⟗ lista\_ingredientes))

**h)** Quais são as sobremesas que tem mais fotografias?

**SQL:**

**WITH** x **as (select** IdReceita**, count(**fotografias.UrlFoto**) as** n\_fotos

**from** receita **natural inner join** fotografias

**where** Tipo **like** 'Sobremesa' **group by** IdReceita**)**

**select** IdReceita, NomeR **,** n\_fotos

**from** x **natural inner join** receita

**where** n\_fotos**= (SELECT max(**n\_fotos**) FROM** x**)**

**(Continuação – Expressões SQL e Álgebra Relacional)**

**ÁLGEBRA RELACIONAL:**

***X* < - - (**IdReceita, **Gcount** (fotografias.UrlFoto) ***as*** N\_Fotos

σTipo=’Sobremesa’ (faz\_receita⟗ lista\_ingredientes) -> Nome**), ->(group by)**

ΠIdReceita,NomeR,N\_Fotos (n\_fotos = (Gmax (n\_fotos)(x)(x ⟗ receita))

**i)** Quais são os membros que dão 3 estrelas a todas as entradas com cogumelos?

**SQL:**

***select*** username

***from*** membro ***as*** T

***where not exists(select*** IdReceita

***from*** receita ***natural inner join*** lista\_ingredientes

***where*** tipo ***like*** 'Entrada' ***and*** nomeingrediente ***like*** 'Cogumelos'

***except***

***select*** IdReceita

***from*** gostos ***natural inner join*** membro ***natural inner join*** receita

***where*** T.username ***=*** gostos.username ***and*** valor = '3')

**ÁLGEBRA RELACIONAL:**

ΠUsername (not exists(ΠIdReceita(σTipo=’Entrada’ ∧ nomeIngrediente=’Cogumelos’ (receita ⟗ lista\_ingredientes)) ÷ ΠIdReceita(σT.username=gostos.username ∧ valor = ‘3’ (gostos ⟗ membro ⟗ receita)))(membro -> T))

**(Continuação – Expressões SQL e Álgebra Relacional)**

**j)** Qual é o membro que tem mais gostos com 3 estrelas dos seus amigos nas suas receitas?

**k)** Para cada Sobremesa rápida que usa chocolate indique o número de gostos com 3 estrelas e o custo.

**SQL:**

***select*** IdReceita, Custo, ***count(***valor***) as*** N\_gostos

***from*** receita ***natural inner join*** gostos ***natural inner join*** Lista\_Ingredientes

***where*** nomeingrediente ***like*** 'Chocolate’ ***and*** TempoDeConfecao ***like***‘Rápido’

***group by*** IdReceita, Custo

**ÁLGEBRA RELACIONAL:**

IdReceita,Custo,**Gcount**(valor) ***as*** N\_Gostos (σNomeIngrediente=‘Chocolate’ ∧ TempoDeConfecao=’Rápido’ (receita ⟗ gostos ⟗ lista\_ingredientes)) -> IdReceita,Custo

**(Continuação – Expressões SQL e Álgebra Relacional)**

**l)** Quantas sobremesas têm o nome mousse de chocolate e têm mais de cinco membros que deram 2 ou mais estrelas?

**SQL:**

***WITH*** x ***as (select*** IdReceita**, *count(***distinct NomeR***) as*** N\_Mousse

***from*** receita

***where*** NomeR ***like*** 'Mousse de chocolate’ ***group by*** IdReceita**)**

***y as (select*** IdReceita, ***count(distinct*** username**) *as*** N\_Membros

***from*** membro ***natural inner join*** gostos

***where*** Valor > '1' ***group by I***dReceita***)***

***select*** IdReceita, NomeR, N\_Mousse, N\_Membros

***from*** x ***natural inner join*** receita ***natural inner join*** y

**ÁLGEBRA RELACIONAL:**

***X* < - - (**IdReceita, **Gcount** (distinct NomeR) ***as*** N\_Mousse

σNomeR=’Mousse de Chocolate’ (receita)**)** -> IdReceita

***Y* < - - (**IdReceita, **Gcount** (distinct username) ***as*** N\_Membros

σValor > ‘1’ (membro ⟗ gostos)**)** -> IdReceita

ΠIdReceita,NomeR,N\_Mousse,N\_Membros (x ⟗ y ⟗ receita)

**m)** Sabendo que uma receita pode ser recomendada a um membro se todos os seus amigos a avaliaram com duas ou mais estrelas. Indique quais são as receitas recomendadas ao membro Manuel Silva.

A informação foi acrescentada à base de dados, mas o grupo não conseguiu chegar a uma resolução correta do exercício.