



RELATÓRIO DO 2º TRABALHO (1ª Parte)

BASE DE DADOS

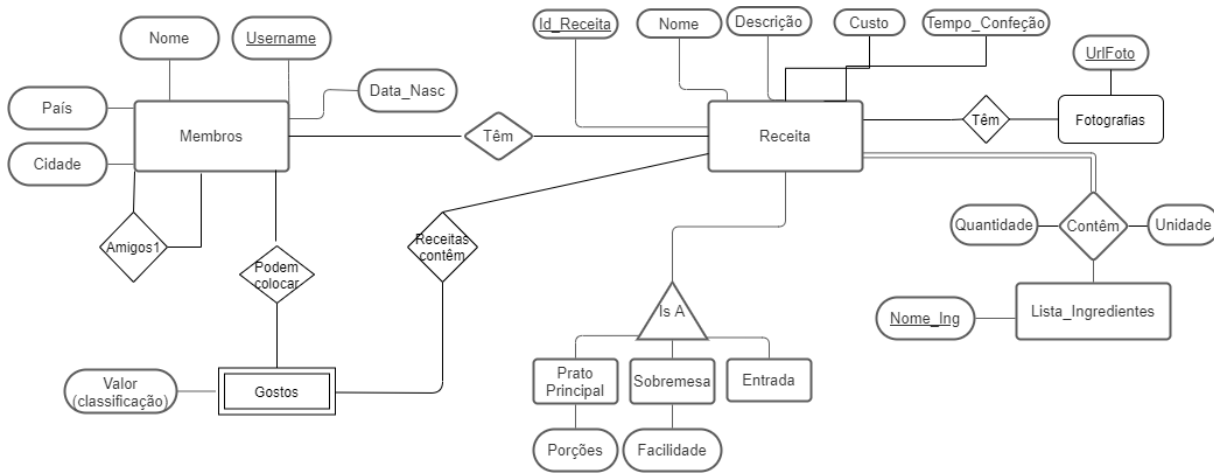


Alunos: Diogo Castanho Nº42496

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(UriFoto, 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, UriFoto)
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:

$$\Pi_{\text{IdReceita, NomeR}} (\sigma_{\text{NomeIngrediente}='Pato'}(\text{receita} \bowtie \text{lista_ingredientes}))$$

(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:

$\pi_{IdReceita, NomeR} (receita) \div \pi_{IdReceita, NomeR} (\sigma_{NomeIngrediente='Pato'}(receita \bowtie lista_ingredientes))$

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:

$\pi_{Username} (\sigma_{NomeIngrediente = 'Pato'} (faz_receita \bowtie membro \bowtie lista_ingredientes))$

(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:

$$\Pi_{\text{Username}} (\sigma_{\text{username}} (\Pi_{\text{Username1}} (\sigma_{\text{username2}} (\Pi_{\text{Username}} (\text{NomeIngrediente} = \text{'Pato'} (\text{faz_receita} \bowtie \text{membro} \bowtie \text{lista_ingredientes})))) (\text{amigos1}))) (\text{membro})))$$

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:

$\Pi_{\text{Username}} (\sigma_{\text{valor} > '1' \wedge \text{NomeIngrediente} = \text{'Pato'}} (\text{gostos} \bowtie \text{lista_ingredientes}))$

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:

$\Pi_{\text{Username}, \text{IdReceita}} (\sigma_{\text{NomeIngrediente} = \text{'Ovos'}} (\text{faz_receita} \bowtie \text{lista_ingredientes})) \cap$
 $\Pi_{\text{Username}, \text{IdReceita}} (\sigma_{\text{NomeIngrediente} = \text{'Amêndoa'}} (\text{faz_receita} \bowtie \text{lista_ingredientes}))$

(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:

$$\Pi_{\text{Username, IdReceita, NomeIngrediente}}(\sigma_{\text{NomeIngrediente}='Pato'}(\text{faz_receita} \bowtie \text{lista_ingrediente})) \cup \Pi_{\text{Username, IdReceita, NomeIngrediente}}(\sigma_{\text{NomeIngrediente}='Perú'}(\text{faz_receita} \bowtie \text{lista_ingredientes}))$$

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:

$X \leftarrow - (IdReceita, Gcount (fotografias.UrlFoto) \text{ as } N_Fotos$

$\sigma_{Tipo='Sobremesa'} (faz_receita \bowtie lista_ingredientes) \rightarrow Nome), \rightarrow (group\ by)$

$\Pi_{IdReceita, NomeR, N_Fotos} (n_fotos = (Gmax (n_fotos)(x) (x \bowtie receita)))$

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:

$\Pi_{Username} (not\ exists(\Pi_{IdReceita} (\sigma_{Tipo='Entrada' \wedge nomeIngrediente='Cogumelos'} (receita \bowtie lista_ingredientes))) \div \Pi_{IdReceita} (\sigma_{T.username=gostos.username \wedge valor='3'} (gostos \bowtie membro \bowtie receita))) (membro \rightarrow T))$

(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:

$\text{IdReceita, Custo, } \mathbf{Gcount}(\text{valor}) \quad \mathbf{as} \quad \text{N_Gostos} \quad (\sigma_{\text{NomeIngrediente}='Chocolate' \wedge \text{TempoDeConfecao}='Rápido'} (\text{receita} \bowtie \text{gostos} \bowtie \text{lista_ingredientes})) \rightarrow \text{IdReceita, Custo}$

(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 IdReceita)
```

```
select IdReceita, NomeR, N_Mousse, N_Membros  
from x natural inner join receita natural inner join y
```

ÁLGEBRA RELACIONAL:

$X \leftarrow \pi_{(IdReceita, Gcount(distinct\ NomeR))} \sigma_{NomeR = 'Mousse\ de\ Chocolate'}(receita)$

$\sigma_{NomeR = 'Mousse\ de\ Chocolate'}(receita) \rightarrow IdReceita$

$Y \leftarrow \pi_{(IdReceita, Gcount(distinct\ username))} \sigma_{Valor > '1'}(membro \bowtie gostos)$

$\sigma_{Valor > '1'}(membro \bowtie gostos) \rightarrow IdReceita$

$\pi_{IdReceita, NomeR, N_Mousse, N_Membros} (x \bowtie y \bowtie receita)$

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.