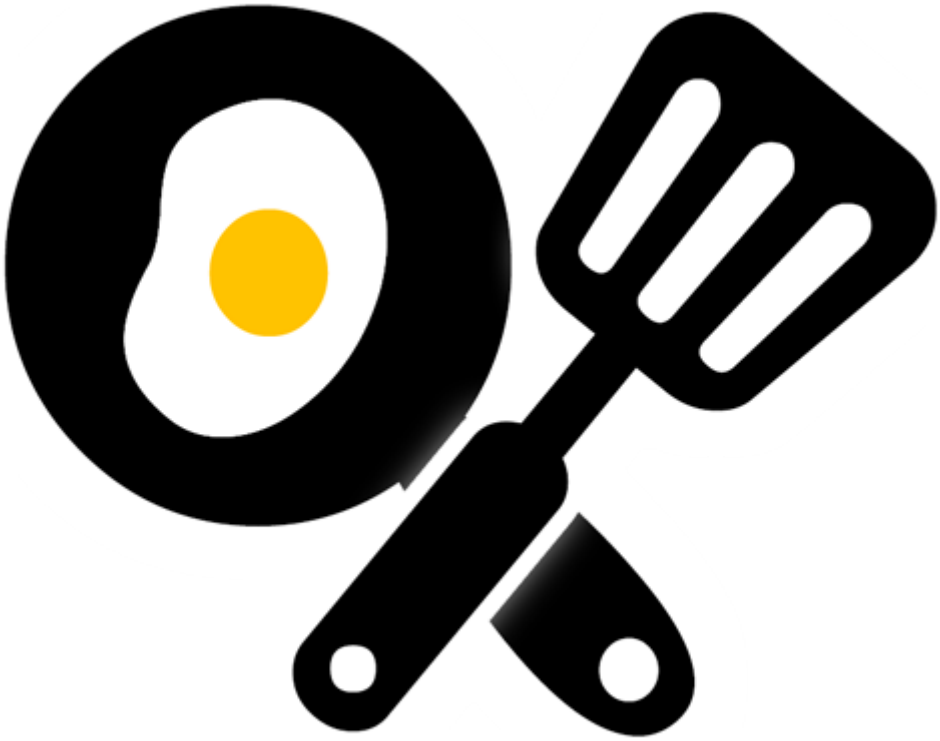
Universidade de Évora
Engenharia Informática

Base de Dados

Fãs de Culinária



Trabalho realizado por:

Filipe Alfaiate l43315

José Santos l43017

Professor:

Irene Rodrigues

Introdução e Objetivos

Pretende-se desenvolver uma base de dados para gerir a informação de uma rede social de *Fãs de Culinária* para partilha e recomendação de receitas.

Para gerir a informação na rede é necessário representar os dados sobre: os membros, receitas, amigos e 'gostos'.

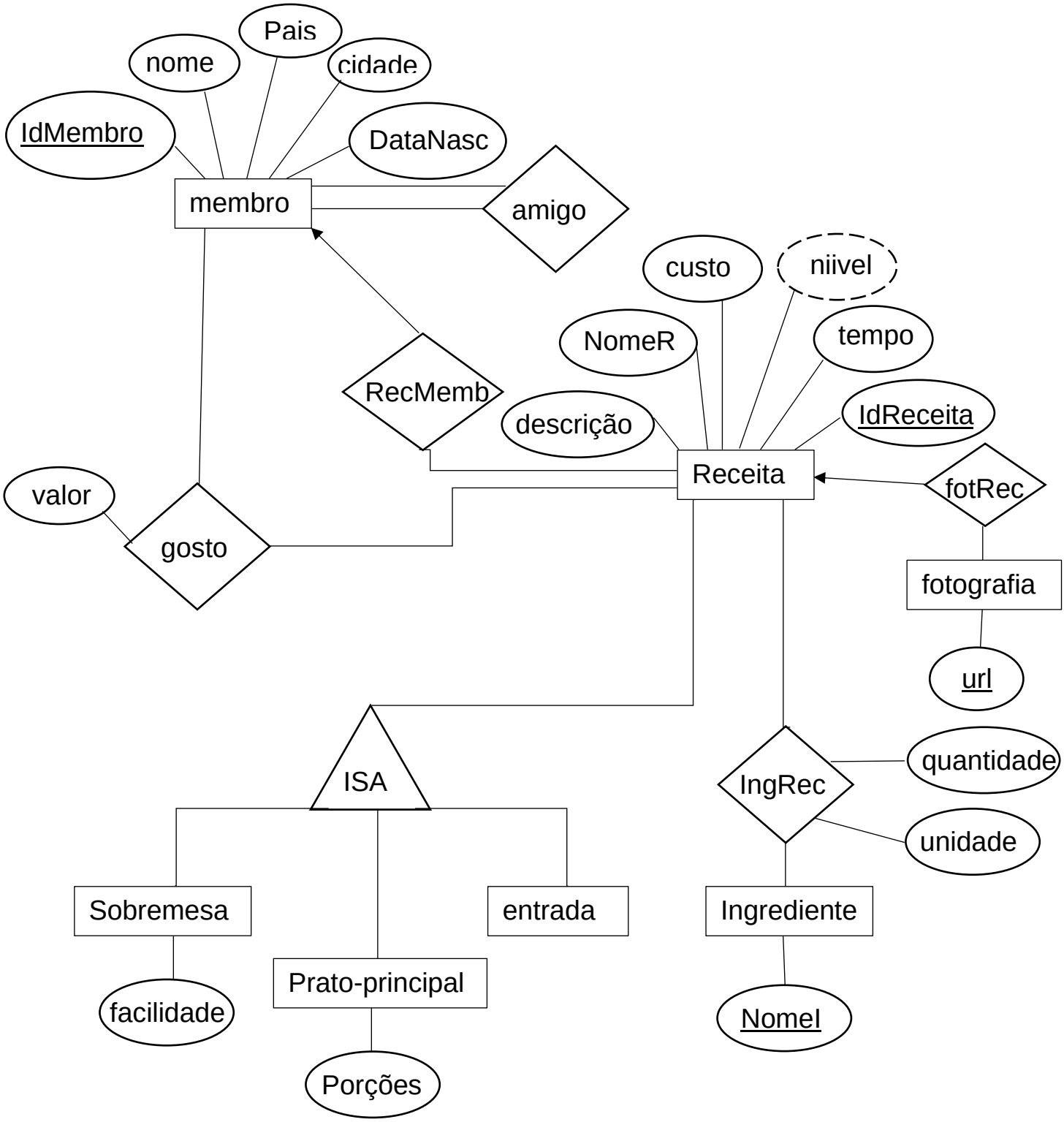
Sobre os membros pretende-se registar: o nome, o IdMemb que é um identificador único para cada membro, o país onde nasceu, a cidade onde vive e a data de nascimento. Os membros têm amigos, se o membro A é amigo do membro B então B também é amigo de A. Os membro têm receitas e podem colocar gostos nas receitas de outros membros.

Uma receita tem um Nome; um identificador único que identifica cada receita; uma lista de ingredientes com o nome do ingrediente (farinha, arroz, pato, ovos, etc.) a quantidade e a unidade (grama, chávena, colher de sopa, etc.); um texto com uma descrição; uma ou mais fotografias (represente uma fotografia com uma sequência de caracteres, um url qualquer que não precisa de ser de uma fotografia).

As receitas podem ser de sobremesas, entradas, ou pratos principais. Os pratos principais têm sempre o número de porções indicado (ex.: 2, 4 ou 6 pessoas). As sobremesas e as entradas não indicam o número de porções . As sobremesas têm sempre a facilidade (fácil, médio, difícil), os outros tipos de receitas não indicam a facilidade. Todas as receitas têm o tempo de confeção e o custo ($, $$, $$$) e um nível que corresponde ao valor médio dos 'gostos'. Um 'gosto' pode ter um valor entre 1 e 3 estrelas.

Universidade de Évora
Engenharia Informática

Diagrama Entidade Relação (ER)



Universidade de Évora

Engenharia Informática

Criar tabelas da Base Dados

membro(Idmemb, nome, pais, cidade, DataNasc)

```
drop table if exists membro cascade;
create table membro(
        Idmemb varchar (20) primary key,
        Nome varchar (20),
        Pais varchar (2),
        cidade varchar (15),
        DataNasc char(8)
);
```

amigo(IdMemb, IdMemb)

```
drop table if exists amigo cascade;
create table amigo(
        Idmemb1 varchar(20),
        Idmemb2 varchar(20),
        primary key (IdMemb1, IdMemb2),
        foreign key (IdMemb1) references membro on delete restrict,
        foreign key (IdMemb2) references membro on delete restrict
);
```

receita(IdReceita, Descrição, Nome, Custo, Nivel, Tempo)

```
drop table if exists receita cascade;
create table receita(
        IdReceita varchar(3) primary key,
        Descrição varchar (100),
        NomeR varchar (50),
        Custo varchar (3),
        Tempo integer
);
```

gosto(IdMemb, IdReceita, valor)

```
drop table if exists gosto cascade;
create table gosto(
        IdMemb varchar (20),
        IdReceita varchar (3),
        valor smallint,
        primary key (IdMemb, IdReceita),
        foreign key (IdReceita) references receita on delete restrict,
        foreign key (IdMemb) references membro on delete restrict
);
```

recMemb(IdReceita, IdMemb)

```
drop table if exists recMemb cascade;
create table recMemb(
        IdReceita varchar (3) primary key,
        IdMemb varchar (20),
        foreign key (IdReceita) references receita on delete restrict,
        foreign key (IdMemb) references membro on delete restrict
);
```

sobremesa(IdReceita, facilidade)

```
drop table if exists sobremesa cascade;
create table sobremesa(
        IdReceita varchar (3) primary key,
        facilidade varchar (7),
        foreign key (IdReceita) references receita on delete restrict
);
```

prato-principal(IdReceita, porções)

```
drop table if exists prato_principal cascade;
create table prato_principal(
        IdReceita varchar (3) primary key,
        Porcoes smallint,
        foreign key (IdReceita) references receita on delete restrict
);
```

entrada(IdReceita)

```
drop table if exists entrada cascade;
create table entrada(
        IdReceita varchar (3) primary key,
        foreign key (IdReceita) references receita on delete restrict
);
```

fotografia(Url)

```
drop table if exists fotografia cascade;
create table fotografia(
        Url varchar (200) primary key
);
```

fotRec(Url, IdReceita)

```
drop table if exists fotRec cascade;
create table fotRec(
        Url varchar (200) primary key,
        IdReceita varchar (3),
        foreign key (Url) references fotografia on delete restrict
);
```

ingrediente(NomeI)

```
drop table if exists ingrediente cascade;
create table ingrediente(
        NomeI varchar (20) primary key
);
```

ingRec(IdReceita, NomeI, Quantidade, Unidade)

```
drop table if exists ingRec cascade;
create table ingRec(
        IdReceita varchar (3),
        NomeI varchar (20),
        Quantidade smallint,
        Unidade varchar (20),
        primary key (IdReceita, NomeI),
        foreign key (IdReceita) references receita on delete restrict,
        foreign key (NomeI) references ingrediente on delete restrict
);
```

Universidade de Évora

Engenharia Informática

Álgebra Relacional e SQL

a) Quais as Receitas com Pato?

Álgebra Relacional:

a) $\Pi_{NomeR}\left(\sigma_{NomeI = 'pato'}(receita \bowtie ingRec \bowtie ingrediente)\right)$

SQL:

```
select NomeR
from receita natural inner join ingRec natural inner join ingrediente
where NomeI like 'pato';
```

b) Quais as Receitas que não têm Pato?

Álgebra Relacional:

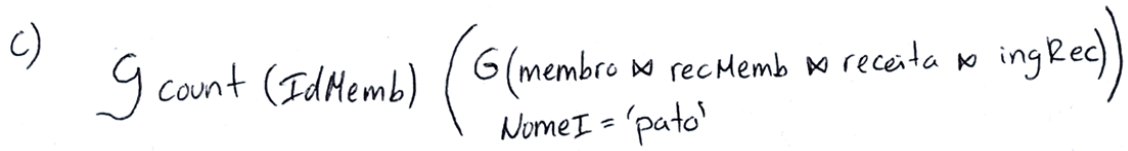
b) $\Pi_{NomeR}(receita) - \Pi_{NomeR}\left(\sigma_{NomeI = 'pato'}(receita \bowtie ingRec \bowtie ingrediente)\right)$

SQL:

```
select IdReceita, NomeR
from receita
except
select IdReceita, NomeR
from receita natural inner join ingRec natural inner join ingrediente
where NomeI like 'pato'
Order by IdReceita;
```

c) Quais os membros que têm receitas com Pato?

Álgebra Relacional:

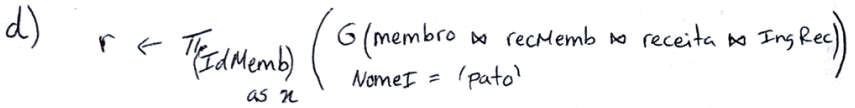


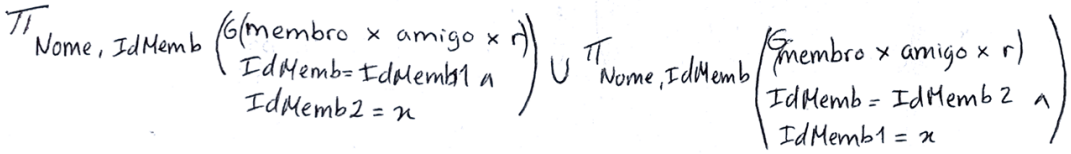
SQL:

```
select count(IdMemb)
from membro natural inner join recMemb natural inner join receita natural inner join ingRec
where NomeI like 'pato';
```

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

Álgebra Relacional:





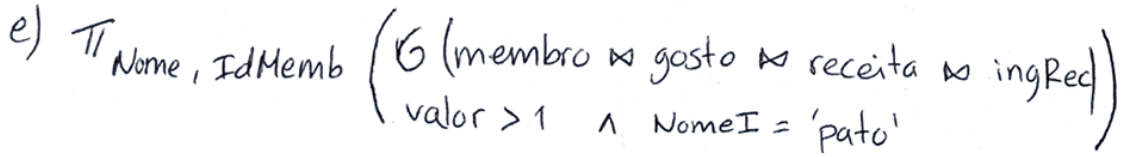
SQL:

```
with autoresPato as (select IdMemb as x
from membro natural inner join recMemb natural inner join receita natural inner join ingRec
where NomeI like 'pato')

select Nome,IdMemb
from membro,amigo,autoresPato
where Idmemb=IdMemb1 and IdMemb2=x
Union
select Nome,Idmemb
from membro,amigo,autoresPato
where IdMemb=IdMemb2 and IdMemb1=x
Order by IdMemb;
```

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

Álgebra Relacional:

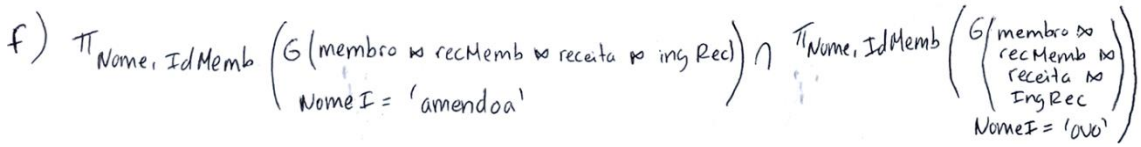


SQL:

```
select distinct Nome,IdMemb
from membro natural inner join gosto natural inner join receita natural inner join ingRec
where valor > 1 and NomeI like 'pato';
```

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

Álgebra Relacional:

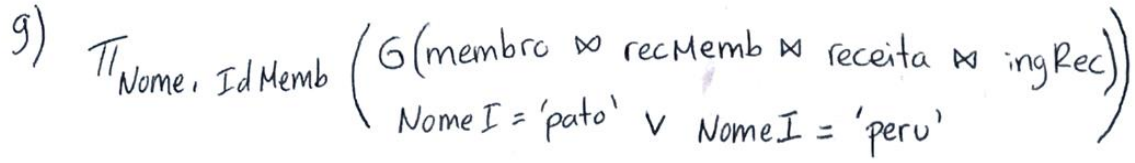


SQL:

```
select distinct Nome, IdMemb
from membro natural inner join recMemb natural inner join receita natural inner join ingRec
where NomeI like 'ovo'
intersect
select distinct Nome, IdMemb
from membro natural inner join recMemb natural inner join receita natural inner join ingRec
where NomeI like 'amendoa';
```

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

Álgebra Relacional:

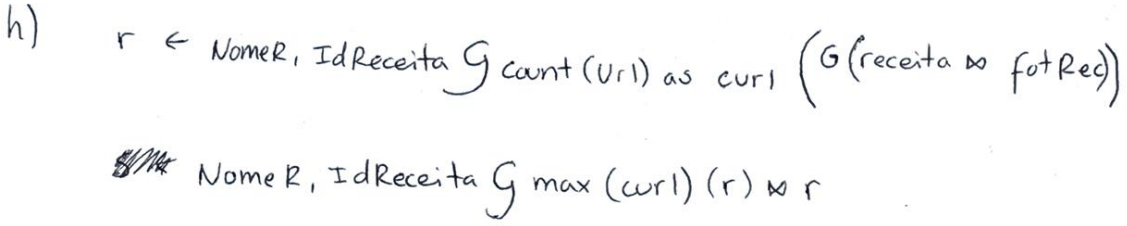


SQL:

```
select distinct Nome, IdMemb
from membro natural inner join recMemb natural inner join receita natural inner join ingRec
where NomeI like 'pato' or NomeI like 'peru';
```

h) Quais são as sobremesas que tem mais fotografias?

Álgebra Relacional:



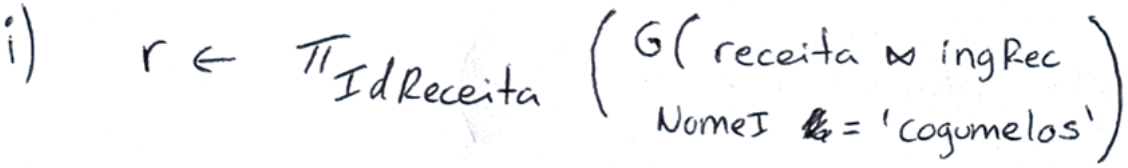
SQL:

```
with a as (select NomeR, IdReceita, count(Url) as x
from receita natural inner join fotRec group by (NomeR,IdReceita))

select NomeR, IdReceita
from a
where x = (select max(x) from a);
```

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

Álgebra Relacional:



SQL:

```
with aux as (select IdReceita
from receita natural inner join ingRec natural inner join entrada
where NomeI like 'cogumelos' group by IdReceita),

aux2 as (select Nome, count(Nome) as cNome
from membro natural inner join gosto natural inner join aux
where valor = 3 and IdReceita = aux.IdReceita group by Nome),

nEntradas as (select count(aux.IdReceita) as cEntradas from aux)

select Nome
from aux2,nEntradas
where cEntradas = cNome;
```

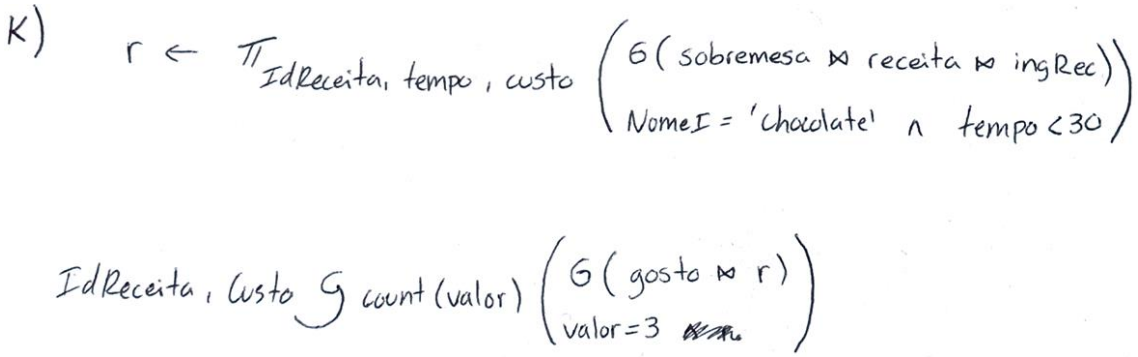
j) Qual é o membro que tem mais gosto com 3 estrelas dos seus amigos nas suas receitas?

Álgebra Relacional:

SQL:

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

Álgebra Relacional:



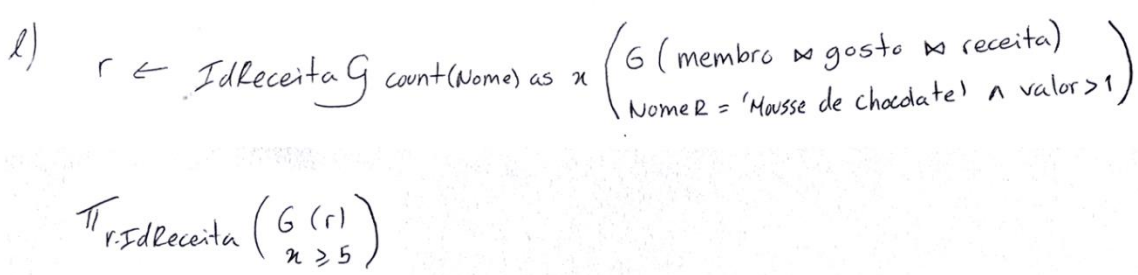
SQL:

```
with sobRapChoco as ( select NomeR, IdReceita, tempo, custo
from sobremesa natural inner join receita natural inner join
ingRec
where NomeI like 'chocolate' and tempo<30)

select NomeR, IdReceita, count(valor), custo
from gosto natural inner join sobRapChoco
where valor = 3 group by NomeR,IdReceita,custo;
```

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

Álgebra Relacional:



SQL:

```
with r as (select IdReceita, count(Nome) as x
from membro natural inner join gosto natural inner join receita
where NomeR like 'Mousse de chocolate' and valor >1 group by
IdReceita)

select count(r.IdReceita)
from r
where x>=5;
```

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.

Álgebra Relacional:

m)

$$r \leftarrow \Pi_{IdMemb}\left(\sigma_{nome='Manuel\ Silva'}(membro)\right)$$

$$s \leftarrow \Pi_{membro.IdMemb}\left(\sigma_{membro.IdMemb = amigo.IdMemb1 \wedge amigo.IdMemb2 = r.IdMemb}(membro \times amigo \times r)\right) \cup$$

$$\Pi_{membro.IdMemb}\left(\sigma_{membro.IdMemb = amigo.IdMem2 \wedge amig.IdMemb1 = r.IdMemb}(membro \times amigo \times r)\right)$$

$$u \leftarrow IdReceita\ \mathcal{G}\ count(IdReceita)\ as\ cidR\left(\sigma_{valor \geq 2}(s \bowtie gosto)\right)$$

$$y \leftarrow \mathcal{G}\ count(IdMemb)\ as\ cidm\ (s)$$

$$\Pi_{NomeR,\ receita.IdReceita}\left(\sigma_{u=y \wedge u.IdReceita = receita.IdReceita}(u \times y \times receita)\right)$$

SQL:

```
with IdManuel as (select IdMemb from membro where nome like
'Manuel Silva'),

amigosManuel as (select membro.IdMemb
                 from membro,amigo,IdManuel
                 where membro.IdMemb =
amigo.IdMemb1 and amigo.IdMemb2 = IdManuel.IdMemb
                 Union
                 select membro.IdMemb
                 from membro,amigo,IdManuel
                 where membro.IdMemb =
amigo.IdMemb2 and amigo.IdMemb1 = IdManuel.IdMemb),

nReceitas as (select idReceita, count(IdReceita) as x
from amigosManuel natural inner join gosto
where valor >= 2 group by IdReceita),

nAmigos as (select count(idMemb) as y from amigosManuel)

select NomeR, receita.IdReceita
from nAmigos,nReceitas, receita
where x=y and nReceitas.IdReceita = receita.IdReceita
```