**Uma imagem com eletrónica

Descrição gerada automaticamente**

****

ANO LETIVO 2020/2021

**RELATÓRIO DO 1º TRABALHO - BASE DE DADOS**

**Alunos:** Diogo Castanho Nº42496

Pedro Grilo Nº43012

**Disciplina:** Base de dados

**Relações**

**membro(Nome, IdMemb, Pais, Cidade, DataNasc)**

CHAVES CANDIDATAS: {IdMemb}

CHAVES PRIMARIAS: {IdMemb}

CHAVES ESTRANGUEIRA: {}

**amigo(IdMemb, IdMemb)**

CHAVES CANDIDATAS: {IdMemb, IdMemb}

CHAVES PRIMARIAS: {IdMemb, IdMemb}

CHAVES ESTRANGEIRAS: {IdMemb} da relação membro

**livro(ISBN,Titulo)**

CHAVES CANDIDATAS: {ISBN}

CHAVES PRIMARIAS: {ISBN}

CHAVES ESTRANGEIRAS: {}

**(Continuação - Relações)**

**gosta(IdMemb,ISBN)**

CHAVES CANDIDATAS: {IdMemb,ISBN}

CHAVES PRIMARIAS: {IdMemb,ISBN}

CHAVES ESTRANGEIRAS: {IdMemb} da relação membro

{ISBN} da relação livro

**genero(ISBN,Genero)**

CHAVES CANDIDATAS: {ISBN,Genero}

CHAVES PRIMARIAS: {ISBN,Genero}

CHAVES ESTRANGEIRAS: {ISBN} da relação livro

**autor(Coda,Nome,Pais)**

CHAVES CANDIDATAS:{Coda}

CHAVES PRIMARIAS:{Coda}

CHAVES ESTRANGEIRAS:{}

**(Continuação – Relações)**

**autoria(ISBN,Coda)**

CHAVES CANDIDATAS: {ISBN,Coda}

CHAVES PRIMARIAS: {ISBN,Coda}

CHAVES ESTRANGEIRAS: {ISBN} da relação livro

{Coda} da relação autor

**Expressões SQL e Álgebra Relacional**

**a)** Qual é o nome dos autores de obras do género drama?

**SQL:**

***select*** NomeA

***from*** autor, autoria, genero

***where*** Genero = 'Drama' and genero.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

**ÁLGEBRA RELACIONAL:**

ΠNomeA (σ Genero='Drama'(autor ⟗ autoria ⟗ genero))

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

**b)** Qual o nome dos membros que gostam de livros da Agatha Christie?

**SQL:**

***select*** Nome

***from*** membro, gosta, autor, autoria

***where*** NomeA = 'Agatha Christie' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

**ÁLGEBRA RELACIONAL:**

πNome (σnomeA='Agatha Christie' (membro ⟗ gosta ⟗ autor ⟗ autoria))

**c)** Qual o nome dos membros que gostam de um livro de um autor que nasceu no seu país?

**SQL:**

***select*** Distinct Nome

***from*** membro, autor, gosta, autoria

***where*** membro.PaisM = autor.PaisA and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

**ÁLGEBRA RELACIONAL:**

ΠNome (σmembro.PaisM = autor.PaisA (membro ⟗ gosta ⟗ autor ⟗ autoria))

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

**d)** Quais os membros que não gostam de algum livro da Agatha Christie?

**SQL:**

***select*** Nome

***from*** membro, autor, autoria, gosta

***except***

***select*** Nome

***from*** membro, autor, autoria, gosta

***where*** NomeA = 'Agatha Christie' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

**ÁLGEBRA RELACIONAL:**

ΠNome(membro ⟗ gosta ⟗ autor ⟗ autoria) - ΠNome(σnomeA = 'Agatha Christie' (membro ⟗ gosta ⟗ autor ⟗ autoria))

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

**e)** Quais os membros que não são amigos do membro com o idMemb oleitor?

**SQL:**

***select*** *IdMemb,Nome*

***from*** membro

***except***

***(select*** IdMemb2,Nome

***from*** membro natural inner join amigo1

***where*** IdMemb='oleitor'

***union***

***select*** IdMemb,Nome

***from*** membro natural inner join amigo1

***where*** IdMemb2='oleitor'***)***

**ÁLGEBRA RELACIONAL:**

πIdMemb,Nome (membro) – ((πIdMemb2,Nome(σIdMemb = 'oleitor'(*membro ⟗ amigo1*)) ) + (πIdMemb,Nome (σIdMemb2= 'oleitor'(*membro ⟗ amigo1*)))

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

**f)** Qual o nome dos amigos do oleitor que são mais jovens que ele?

**SQL:**

***select*** membro2.Nome

***from*** membro natural inner join amigo1 natural inner join membro2

***where*** IdMemb='oleitor' and membro.DataNasc < membro2.DataNasc

**ÁLGEBRA RELACIONAL:**

πNome (σIdMemb = 'oleitor' and membro.DataNasc < membro2.DataNasc(membro ⟗ amigo1))

**g)** Qual o nome dos membros que gostam de livros da Agatha Christie e do Francisco José Viegas?

**SQL:**

***select*** Nome

***from*** membro, gosta, autor, autoria

***where*** NomeA = 'Agatha Christie' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

***intersect***

***select*** Nome

***from*** membro, gosta, autor, autoria

***where*** NomeA = 'Francisco José Viegas' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

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

**ÁLGEBRA RELACIONAL:**

πNome (σNomeA = ‘Agatha Christie' (membro ⟗ gosta ⟗ autor ⟗ autoria)) \* πNome(σNomeA = ‘Francisco José Viegas ' (membro ⟗ gosta ⟗ autor ⟗ autoria)

**h)** Qual o nome dos membros que gostam de livros da Agatha Christie ou do Francisco José Viegas?

**SQL:**

***select*** Nome

***from*** membro, gosta, autor, autoria

***where*** NomeA = 'Agatha Christie' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

***union***

***select*** Nome

***from*** membro, gosta, autor, autoria

***where*** NomeA = 'Francisco José Viegas' and membro.IdMemb = gosta.IdMemb and gosta.ISBN = autoria.ISBN and autoria.CodA = autor.CodA

**ÁLGEBRA RELACIONAL:**

πNome (σNomeA = ‘Agatha Christie' (membro ⟗ gosta ⟗ autor ⟗ autoria)) + πNome(σNomeA = ‘Francisco José Viegas ' (membro ⟗ gosta ⟗ autor ⟗ autoria)

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

**i)** Quantos amigos tem o membro oleitor?

**SQL:**

**select count (\*)**

***from (select*** nome

***from*** amigo natural inner join membro

***where*** IdMemb like 'oleitor'**) as** NumberOfFriends

**ÁLGEBRA RELACIONAL:**

nome **G count (\*) as** NumberOfFriends (σ IdMemb = 'oleitor’ (amigo ⟗ membro))

**j)** Qual é o membro que tem mais amigos?

**SQL:**

***With*** x ***as* (*select*** membro.Nome***, count(***amigo1.IdMemb***) as*** n\_amigos

***from*** membro, amigo1

***where*** membro.IdMemb = amigo1.IdMemb ***group by*** Nome***)***

***select*** nome,n\_amigos

***from*** x

***where*** n\_amigos **= *(select max(***n\_amigos***) from*** x***)***

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

**ÁLGEBRA RELACIONAL:**

X <-- (πmembro.Nome(**Gcount** (amigo1.IdMemb) as n\_amigos

σmembro.IdMemb = amigo1.IdMemb

Nome, n\_amigos(σ**Gmax**(n\_amigos)(X))

**l)** Para cada livro indique o número de géneros.

**SQL:**

***select*** ISBN**, *count(distinct*** genero***)***

***from*** genero

***group by*** ISBN

**ÁLGEBRA RELACIONAL:**

ISBN **G count( distinct** genero**)** (genero)

**m)** Para cada livro indique o número de géneros e o número de gostos.

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

**SQL:**

***WITH*** X **as (*select*** ISBN***, count(distinct*** genero***) as*** NGeneros

***from*** genero

***group by*** ISBN**),**

Y ***as (select*** ISBN**, *count(distinct*** IdMemb***) as*** NGostos

***from*** membro ***natural inner join*** gosta

***group by*** ISBN**)**

***select*** ISBN, NGeneros, NGostos ***from*** x **natural join** y

**ÁLGEBRA RELACIONAL:**

**X <-- (ISBN, Gcount(distinct genero) as n\_Generos (genero)**

**Y <-- (ISBN, Gcount(distinct IdMemb) as nGostos (membro ⟗ gosta)**

**πISBN, NGeneros, Ngostos(X ⟗ Y)**

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

**n)** Para cada autor indique o número de livros, o número de géneros e o número de gostos

**SQL:**

***WITH*** X ***as (select*** nomeA***, count(distinct*** ISBN***) as*** NLivros

***from*** autor ***natural inner join*** livro ***natural inner join*** autoria

***group by*** nomeA**),**

Y ***as (select*** nomeA, ***count(distinct*** genero***) as*** NGeneros

***from*** autor ***natural inner join*** genero ***natural inner join*** autoria

***group by*** nomeA**),**

Z ***as (select*** nomeA***, count(*I**dMemb**) *as*** NGostos

***from*** autor ***natural inner join*** gosta ***natural inner join* autoria**

***group by*** *nomeA***)**

***select*** nomeA, NLivros, NGeneros, NGostos

***from*** x ***natural join*** y ***natural join*** z

**ÁLGEBRA RELACIONAL:**

**X <-- (NomeA, Gcount(distinct ISBN) as NLivros (autor ⟗ livro ⟗ autoria)**

**Y <-- (NomeA, Gcount(distinct genero) as NGeneros (autor ⟗ genero ⟗ autoria)**

**Z <-- (NomeA, Gcount(distinct IdMemb) as NGostos (autor ⟗ gosta ⟗ autoria)**

**πnomeA, NLivros, NGeneros, Ngostos(X ⟗ Y ⟗ Z)**

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

**o)** Para cada membro, nome, indique o número de amigos e o número de livros de que gosta.

**SQL:**

***WITH*** X ***as (select*** nome, ***count(distinct*** IdMemb2***) as*** NAmigos

***from*** amigo1 ***natural inner join*** membro

***group by*** nome**)**

Y ***as (select*** nome***, count(distinct*** ISBN***) as*** NGostos

***from*** membro ***natural inner join*** gosta

***group by*** nome***)***

***select*** nome, NAmigos, NGostos

***from*** x ***natural join*** y

**ÁLGEBRA RELACIONAL:**

**X <-- (Nome, Gcount(distinct IdMemb2) as NAmigos (amigo1 ⟗ membro)**

**Y <-- (Nome, Gcount(distinct ISBN) as NGostos (membro ⟗ gosta)**

**πNome, NAmigos, NGostos(X ⟗ Y)**

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

**p)** Qual o nome dos membros que são amigos de todos os membros?

**SQL:**

***select*** t.Nome

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

***where not exists(select*** amigo1.IdMemb

***from*** amigo1

***except***

***select*** amigo1.IdMemb

***from*** membro ***natural inner join*** amigo1

***where*** T. IdMemb=amigo1.IdMemb**);**

**ÁLGEBRA RELACIONAL:**

**πT.Nome(σ((πamigo1.IdMemb (amigo1))-(πamigo1.IdMemb(σT.IdMemb=amigo1.IdMemb(membro⟗amigo1)) (membro as T)**