

## Questão 4 da Prova com Exemplos e SQL

tipoim(codt, descr)

propriet(codp, cpf, nome, tel)

imovel(codi, descr, ender, codt(tipoim), qtqua, qtwc, demp, codp(propriet))

loca(codl, cpf, nome, e-mail)

imoloca(codi(imovel), codl (loca), dtloc, dtvenc, vlloc, dtsai)

### tipoim

codt	descr	codp	cpf	nome	tel
01	Apto	0001	111222333	John MacRoe	2223312
02	Casa	0002	444555666	Ivan Lendl	3331122
03	Terreno	0003	222333444	Peter Sampras	4442233
04	Prédio	0004	123123123	Kuerten	2224545
05	Galpão				

### propriet

### imovel

codi	descr	ender	codt	qtqua	qtwc	demp	codp
001	Ampla e arejado	R. Limpa, 10	01	3	2	S	0004
002	Bem situado	R. Próxima, 5	03	0	0		0001
003	Ótima vista	R. Alta, 1023	01	2	1		0004
004	Alvenaria	R. Das Casas, 1	02	4	3	S	0002
005	Ampla	R. Canapé, 23	05	0	1		0002
006	Residencial	R. Flores, 111	04	0	0		0003

### loca

codl	cpf	nome	e-mail
001	321123	Rafael Nadal	<a href="mailto:nadal@uffs">nadal@uffs</a>
002	444111	Roger Federer	<a href="mailto:federer@uffs">federer@uffs</a>
003	555666	Novak Djokovic	<a href="mailto:djoko@uffs">djoko@uffs</a>
004	900001	Andy Murray	<a href="mailto:murray@uffs">murray@uffs</a>

### imoloca

codi	codl	dtloc	dtvenc	vlloc	dtsai
002	002	01/05/2010	06/07/2012	1200	08/07/2012
001	003	09/08/2011	06/07/2013	2200	null
004	001	10/11/2012	10/07/2013	1450	null
002	004	10/09/2012	10/07/2013	1300	null
005	002	10/09/2013	10/06/2013	2450	null

$\pi_{p.nome, i.descr, l.nome} (propriet \bowtie_{i.codi=im.codi} imovel \bowtie (imoloca \bowtie loca))$

codl	cpf	nome	e-mail	codi	codl	dtloc	dtvenc	vlloc	dtsai
001	321123	Rafael Nadal	<a href="mailto:nadal@uffs">nadal@uffs</a>	004	001	10/11/2012	10/07/2013	1450	
002	444111	Roger Federer	<a href="mailto:federer@uffs">federer@uffs</a>	002	002	01/05/2010	06/07/2012	1200	08/07/2012
002	444111	Roger Federer	<a href="mailto:federer@uffs">federer@uffs</a>	005	002	10/09/2013	10/06/2013	2450	
003	555666	Novak Djokovic	<a href="mailto:djoko@uffs">djoko@uffs</a>	001	003	09/08/2011	06/07/2013	2200	
004	900001	Andy Murray	<a href="mailto:murray@uffs">murray@uffs</a>	002	004	10/09/2012	10/07/2013	1300	

codp	cpf	nome	tel	codi	descr	ender	codt	qtqua	qtwc	demp
0001	111222333	John MacRoe	2223312	002	Bem situado	R. Próxima, 5	03	0	0	
0002	444555666	Ivan Lendl	3331122	004	Alvenaria	R. Das Casas, 1	02	4	3	S
0002	444555666	Ivan Lendl	3331122	005	Ampla	R. Canapé, 23	05	0	1	
0003	222333444	Peter Sampras	4442233	006	Residencial	R. Flores, 111	04	0	0	
0004	123123123	Kuerten	2224545	001	Ampla e areja.	R. Limpa, 10	01	3	2	S
0004	123123123	Kuerten	2224545	003	Ótima vista	R. Alta, 1023	01	2	1	

a)

$\pi_{p.nome, i.descr, l.nome} (propriet \bowtie_{i.codi=im.codi} imovel \bowtie (imoloca \bowtie loca))$

**select** p.nome, i.descr, l.nome

**from** propriet p **natural join** imovel i **left join** (imoloca im **natural join** loca l) **on** i.codi=im.codi;

Na solução abaixo, como as tabelas foram invertidas, foi necessário utilizar o RIGHT JOIN

```
select p.nome, i.descr, l.nome
from imoloca im natural join loca l
right join (propriet p natural join imovel i) on im.codl=i.codl;
```

b)

$$\pi_{nome, codl}(loca \bowtie imoloca) \div \pi_{codl}(imovel)$$

$$\{t.nome | t \in loca \wedge \forall i \in imovel (\exists il \in imoloca (t.codl = il.codl \wedge i.codl = il.codl))\}$$

$$\{t.nome | t \in loca \wedge \nexists i \in imovel (\nexists il \in imoloca (t.codl = il.codl \wedge i.codl = il.codl))\}$$

```
select nome
from loca l
where not exists (select codl from imovel i
                  where not exists (select codl
                                     from imoloca il
                                     where i.codl=il.codl
                                     and il.codl=i.codl
                                     )
                  )
```

c)

$$\pi_{descr}(\sigma_{vlloc=(G_{\max(vlloc)}(imoloca))}(imovel \bowtie imoloca))$$

```
select i.descr
from imovel i natural join imoloca il
where vlloc=(select max(vlloc) from imoloca);
```

A descrição do imóvel que foi mais locado:

```
select descr
from imovel natural join imoloca
group by descr
having count(*)= (select max(qt)
                  from (select codl, count(*) as qt from imoloca group by codl) qtloca );
```

descr

-----  
bem situado

descr	count(*)	
-----		
amplo	1	} Resultado do select mais interno (renomeado para qtloca)
bem situado	2	
alvenaria	1	
amplo e arejado	1	

d)

$$t_1 \leftarrow \rho_{(codp, qt)} codp G_{count(codp)}(propriet \bowtie imovel)$$

$$t_2 \leftarrow \rho_{(mx)} G_{\max(qt)}(t_1)$$

$$\pi_{nome}(propriet \bowtie (t_1 \bowtie_{qt=mx} t_2))$$

```

select  nome
from    propriet
group by nome
having  count(*) = ( select max(qt) //sub-query
                    from (select codp, count(*) as qt
                        from propriet natural join imovel
                        group by codp) t1
                    )

```

t1		t2		t1 junção t2	
codp	qt	mx		codp	qt
001	4	10		011	10
002	3				
010	1				
011	10				

e)

$$\pi_{nome}(loca \bowtie \sigma_{today() > dtvenc \wedge dtsai \text{ is null}}(imoloca))$$

$$\{t.nome \mid t \in loca \wedge \exists i \in imoloca (t.codl = i.codl \wedge today() > i.dtvenc)\}$$

```

select  nome
from    loca natural join imoloca
where   today() > dtvenc
and     dtsai is null;

```

## Alguns exercícios resolvidos da lista de álgebra

integer : inteiro (4b)  
real : ponto flutuante (6b)  
varchar (n) : sequência de n caracteres  
date : data (6b)

```
create table cliente (  
    cod integer not null,  
    nome varchar(40) not null,  
    dtnasc date not null,  
    clcliente integer not null,  
    constraint fk_clienteclasse foreign key (clcliente) references classe(cod),  
    constraint pk_cliente primary key (cod)  
)
```

```
create table propriet (  
    codp integer not null constraint pk_propriet primary key,  
    cpf varchar(11) not null,  
    nome varchar(30) not null,  
    tel varchar(20) not null);
```

```
create table tipoim (  
    codt integer not null constraint pk_tipoim primary key,  
    descr varchar(30) not null);
```

```
insert into tipoim values (1,'Apartamento');  
insert into tipoim (descr,codt) values ('Apartamento',1);
```

```
create table imovel (  
    codi integer not null,  
    descr varchar(30) not null,  
    ender varchar(50) not null,  
    codt integer not null,  
    qtqua integer not null,  
    qtwc integer not null,  
    demp char,  
    codp integer not null,  
    constraint pk_imovel primary key (codi),  
    constraint fk_imovel_tipoim foreign key (codt) references tipoim(codt),  
    constraint fk_imovel_propriet foreign key (codp) references propriet(codp)  
);
```

```
create table loca (  
    codl integer not null constraint pk_loca primary key,  
    cpf varchar(11) not null constraint uk_loca unique key,  
    nome varchar(30) not null,  
    email varchar(30) not null);
```

```
create table imoloca (  
    codi integer not null,  
    codl integer not null,  
    dtloc date not null,  
    dtvenc date not null,  
    villoc real not null,  
    dtsai date,  
    constraint pk_imoloca primary key (codi,codl),  
    constraint fk_imoloca_imovel foreign key (codi) references imovel(codi),  
    constraint fk_imoloca_loca foreign key (codl) references loca(codl)  
);
```

### DML

```
insert into <tabela> [(<atributos>)] values (<valores>);
```

```
update <tabela> set <atributo>=<novo valor> [where <predicado>];  
update funcionario set salario=salario *1.1 where coddep=5;  
delete from <tabela> [where <predicado>];  
delete from funcionario where coddep=5;
```

select [distinct] <atributos ou \*> ← corresponde à projeção ( $\pi$ )  
 from <tabela(s)> [<joins>]  
 [where <predicados>] ← seleção ( $\sigma$ )  
 [having <predicados com funções de grupo>]  
 [group by <atributos de agrupamento>] ← utilizado apenas se existir uma função de grupo  
 [order by <atributos>] ← ordena o resultado pelos <atributos>

and or not

select \* from (select descr, ender from imovel where qtqua > 2) t;

select i.descr, t.descr  
 from imovel i **join** tipoimo t **on** i.codt=t.codt;

select i.descr, t.descr  
 from imovel i **natural join** tipoimo t;

## LISTA

- 1)      **select** nome, dtnasc      **select** \*  
          **from** velejador;      **from** (select nome, dtnasc **from** velejador) t;
- 2)      **select** nome, descr  
          **from** velejador **join** classe **on** classe=cod;
- 3)      **select** nome, tam  
          **from** barco  
          **where** cor='Blue';
- 4)      **select** v.nome, v.dtnasc, v.classe, b.nome, b.cor, b.tam, b.pot  
          **from** barco b **join** reserva r **on** b.cod=r.codb  
               **join** velejador v **on** r.codv=v.codv  
          **where** r.dhoras > 20;
- 5)      **select** \*  
          **from** barco  
          **where** cor='Blue'  
          **and** pot > 30;
- 6)      **select** b.nome, v.nome, r.data, r.dhoras  
          **from** barco b **join** reserva r **on** b.codb=r.codb  
               **join** velejador v **on** r.codv=v.codv  
               **join** classe c **on** v.classe=c.cod  
          **where** c.descr='Iniciante'  
          **and** b.cor='White';
- 18)      select nome  
          from velejador natural join reserva

**Select ← consulta**

**join**

**2 maneiras**

1. **from <tabela1> [natural] join <tabela2> on <condição de join>**
2. **where <condição de junção>**

**select \* from medico m join espec e on **m.code=e.cod**  
where city='XAP';**

**select \* from medico m, espec e  
where **m.code=e.cod**  
and city='XAP';**

**operadores de conjunto**

**união: union**

**intersecção: intersect**

**diferença: except**

**select codcli as cod, nomecli as nome from A  
union all  
select codfo as cod, nomefo as nome from B;**

**select codcli as cod, nomecli as nome from A  
intersect all  
select codfo as cod, nomefo as nome from B;**

**select codcli as cod, nomecli as nome from A  
except all  
select codfo as cod, nomefo as nome from B;**

**select **nome** from paciente where cidade='Chapecó'  
union  
select **nome** from medico where cidade='Chapecó';**

**select nome from paciente  
intersect  
select nome from medico;**

1. select nome from medico  
except

2. select m.nome from medico m natural join consulta c;

1.

Gregory House

Chris Taub

Carlos Chagas

==>

Chris Taub

2.

Gregory House

Gregory House

Carlos Chagas

select m.nome from medico m left join consulta c on m.crm=c.crm  
where c.cpf is null;

in → similar ao pertence

not in

exists → operador existencial

not exists

cidade in ('Chapecó', 'Caibi', 'Seara');

cidade = 'Chapecó' or cidade = 'Caibi' or cidade = 'Seara'

select nome from medico

where crm not in (123, 342, 123); //(select crm from consulta);

Gregory House → 123 in (123, 342, 123) => V ~V => F

\*\*\*\* Chris Taub → 222 in (123, 342, 123) => F ~F => V \*\*\*\*

Carlos Chagas → 342 in (123, 342, 123) => V ~V => F

select \* from consulta where dtcons in ('21/08/2012', '22/08/2012');

select \* from consulta

where dtcons = '21/08/2012'

or dtcons = '22/08/2012';

Q1:

select nome from medico

where exists (1); → todos os nomes dos médicos existentes

**Q2:**

**select nome from medico**

**where not exists (select crm from consulta);**

**a)Gregory House**

**b)Gregory House**

**c) Não sei**

**Chris Taub**

**Carlos Chagas**

**Carlos Chagas**

**18**

**4**

**1**

**(123,342,123)**

**(null)**

**Q3:**

**select nome**

**from medico m**

**where NOT exists (select crm from consulta where crm=**m.crm**);**

**Q4:**

**Select p.nome**

**from paciente p**

**where not exist (select m.crm**

**from medico m**

**where code=5 // imaginando que 5 seja ginecologista**

**and not exists (select c.crm**

**from consulta c**

**where c.cpf=p.cpf**

**and c.crm=m.crm)**

**)**

**like**

**between**

**like é um igual para string mais flexível.**

**Select \* from paciente where nome like '%Silva%';**

**\_ nome like 'J\_\_\_\_';**

**between => condição em faixa de valores**

**where idade >=18 and idade <=20;**

**where idade between 18 and 20;**