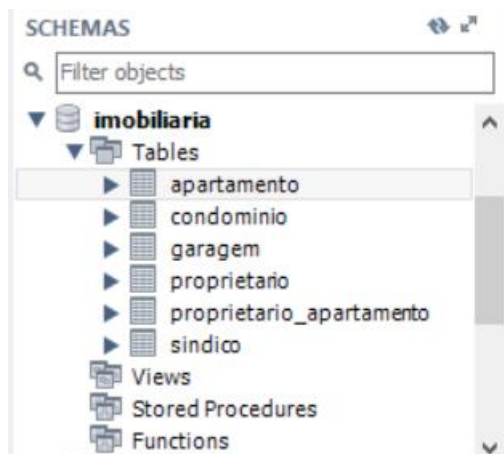


Tecnologia da Informação II	Modulo II
Fichário	Agenda 5 – Linguagem de consulta de dados - DQL
Nome: Mônica Zungalo Quintal	

Estes exercícios devem ser entregues de forma on-line como atividade da agenda.



Utilize as estruturas do Banco de Dados da Imobiliária João de Barro para desenvolver as instruções de acordo com o que se pede em cada um dos exercícios a seguir:

Imagens 1 a 6 – Interface Gráfica Workbench – Estruturas da Base de Dados da Imobiliária São João

Sindico

Field	Type	Null	Key	Default	Extra
matricula	int(3)	NO	PRI	NULL	auto increment
nome	varchar(80)	YES		NULL	
endereco	varchar(80)	YES		NULL	
telefone	varchar(15)	YES		NULL	

Imagem 1

Proprietario

Field	Type	Null	Key	Default	Extra
ra	varchar(15)	NO	PRI	NULL	
nome	varchar(80)	YES		NULL	
telefone	varchar(15)	YES		NULL	
email	varchar(50)	YES		NULL	

Imagem 2

Condominio

Result Grid

Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
	codiao	int(5)	NO	PRI	NULL	auto increment
	nome	varchar(50)	YES		NULL	
	endereco	varchar(80)	YES		NULL	
	matricula_sind	int(3)	YES	MUL	NULL	


Imagem 3

Apartamento

Result Grid


Filter Rows:

Export:


Wrap Cell Content:


	Field	Type	Null	Key	Default	Extra
	numero	varchar(5)	NO	PRI	NULL	
	tipo	varchar(20)	YES		NULL	
	codigo cond	int(5)	YES	MUL	NULL	

Imagem 4

Garagem

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
	numero	int(3)	NO	PRI	NULL	auto increment
	tipo	varchar(20)	YES		NULL	
	numero ap	varchar(5)	YES	MUL	NULL	

Imagem 5

Proprietario_Apartamento

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
	prop ap id	int(3)	NO	PRI	NULL	auto increment
	numero ap	varchar(5)	YES	MUL	NULL	
	ra prop	varchar(15)	YES	MUL	NULL	

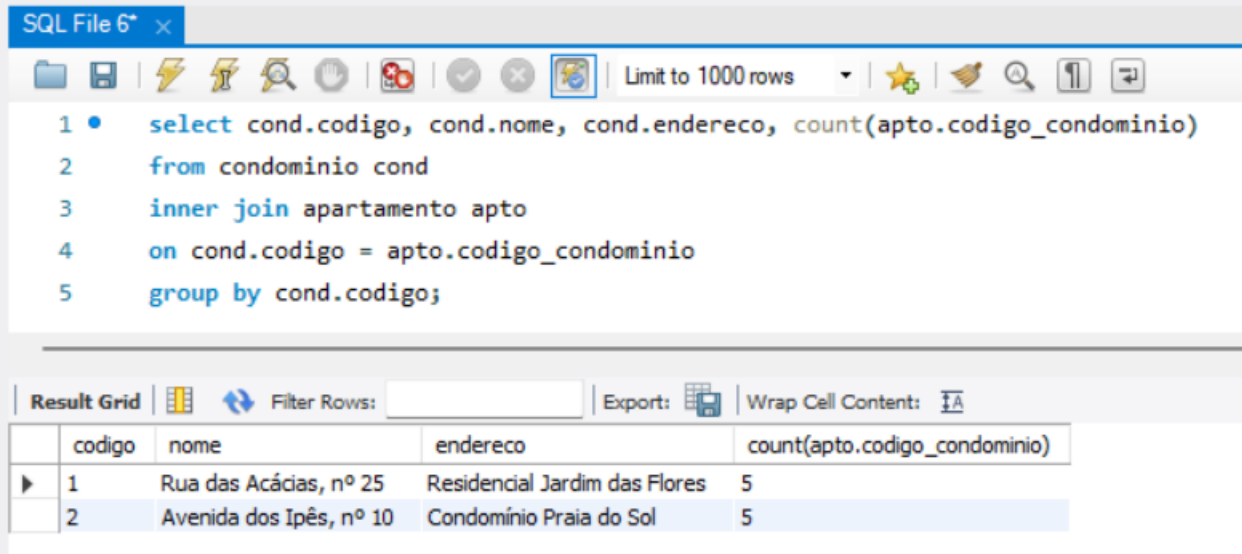
Imagem 6

Obs: para obter a descrição da Tabela utilize o comando `describe <tabela>`.

Insira em cada exercício as instruções (os comandos SQL) utilizadas para fazer o que é solicitado em cada exercício.

1) Relação de condomínios apresentando além do nome, o endereço e o total de apartamentos.

```
select cond.codigo, cond.nome, cond.endereco, count(apto.codigo_condominio)
from condominio cond
inner join apartamento apto
on cond.codigo = apto.codigo_condominio
group by cond.codigo;
```



The screenshot shows a SQL IDE window titled "SQL File 6* x". The query editor contains the following SQL code:

```
1 • select cond.codigo, cond.nome, cond.endereco, count(apto.codigo_condominio)
2   from condominio cond
3  inner join apartamento apto
4   on cond.codigo = apto.codigo_condominio
5  group by cond.codigo;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The grid has four columns: "codigo", "nome", "endereco", and "count(apto.codigo_condominio)". There are two rows of data:

	codigo	nome	endereco	count(apto.codigo_condominio)
▶	1	Rua das Acácias, nº 25	Residencial Jardim das Flores	5
	2	Avenida dos Ipês, nº 10	Condomínio Praia do Sol	5

2) Altere a estrutura da Tabela apartamento adicionando o campo valor com a seguinte definição.

Nome	Tipo	Padrão
valor	double(10,2)	0

```
alter table apartamento add valor double(10,2);
```

```
describe apartamento;
```

The screenshot shows a SQL client window titled "SQL File 6* x". The toolbar includes icons for file operations, execution, and viewing. The command list shows two steps: 1. `alter table apartamento add valor double(10,2);` and 2. `describe apartamento;`. Below the commands, the "Result Grid" displays the table structure for 'apartamento'.

Field	Type	Null	Key	Default	Extra
numero	varchar(5)	NO	PRI	NULL	
tipo	varchar(20)	YES		NULL	
codigo_condominio	int	YES	MUL	NULL	
valor	double(10,2)	YES		NULL	

preços.

Condomínio 1	
Apartamento	Valor
A101	100000.00
A201	115000.00
A301	125000.00
A401	135000.00
A501	150000.00

Condomínio 2	
Apartamento	Valor
B101	200000.00
B201	215000.00
B301	225000.00
B401	235000.00
B501	250000.00

alterados, ou seja, que os apartamentos tenham sido valorizados.

	numero	tipo	codigo_cond	valor
	A101	Padrão	1	100000.00
	A201	Padrão	1	115000.00
	A301	Padrão	1	125000.00
	A401	Padrão	1	135000.00
	A501	Cobertura	1	150000.00
	B101	Padrão	2	200000.00
	B201	Padrão	2	215000.00
	B301	Padrão	2	225000.00
	B401	Padrão	2	235000.00
	B501	Cobertura	2	250000.00

Antes de realizar as alterações indicadas:

SQL File 6* x

Limit to 1000 rows

```
1 select * from apartamento;
```

Result Grid

	numero	tipo	codigo_condominio	valor
▶	101	Padrão	1	NULL
	102	Padrão	1	NULL
	201	Padrão	1	NULL
	202	Padrão	1	NULL
	301	Cobertura	1	NULL
	100	Cobertura	2	NULL
	11	Padrão	2	NULL
	12	Padrão	2	NULL
	21	Padrão	2	NULL
	22	Padrão	2	NULL
*	NULL	NULL	NULL	NULL

Alterando os dados conforme proposto no enunciado:

```
update apartamento set numero = 'A101', valor = 100000.00 where numero = '101';
update apartamento set numero = 'A201', valor = 115000.00 where numero = '102';
update apartamento set numero = 'A301', valor = 125000.00 where numero = '201';
update apartamento set numero = 'A401', valor = 135000.00 where numero = '202';
update apartamento set numero = 'A501', valor = 150000.00 where numero = '301';
update apartamento set numero = 'B101', valor = 200000.00 where numero = '11';
update apartamento set numero = 'B201', valor = 215000.00 where numero = '12';
update apartamento set numero = 'B301', valor = 225000.00 where numero = '21';
update apartamento set numero = 'B401', valor = 235000.00 where numero = '22';
update apartamento set numero = 'B501', valor = 250000.00 where numero = '100';
select * from apartamento;
```

Estrutura final da tabela:

The screenshot shows a SQL IDE window titled "SQL File 6". The top toolbar includes icons for file operations, execution, and search. The SQL editor contains 11 statements: 10 UPDATE statements for the 'apartamento' table and 1 SELECT statement. The statements update the 'numero' and 'valor' fields for various apartment units. The result grid below shows the data after execution.

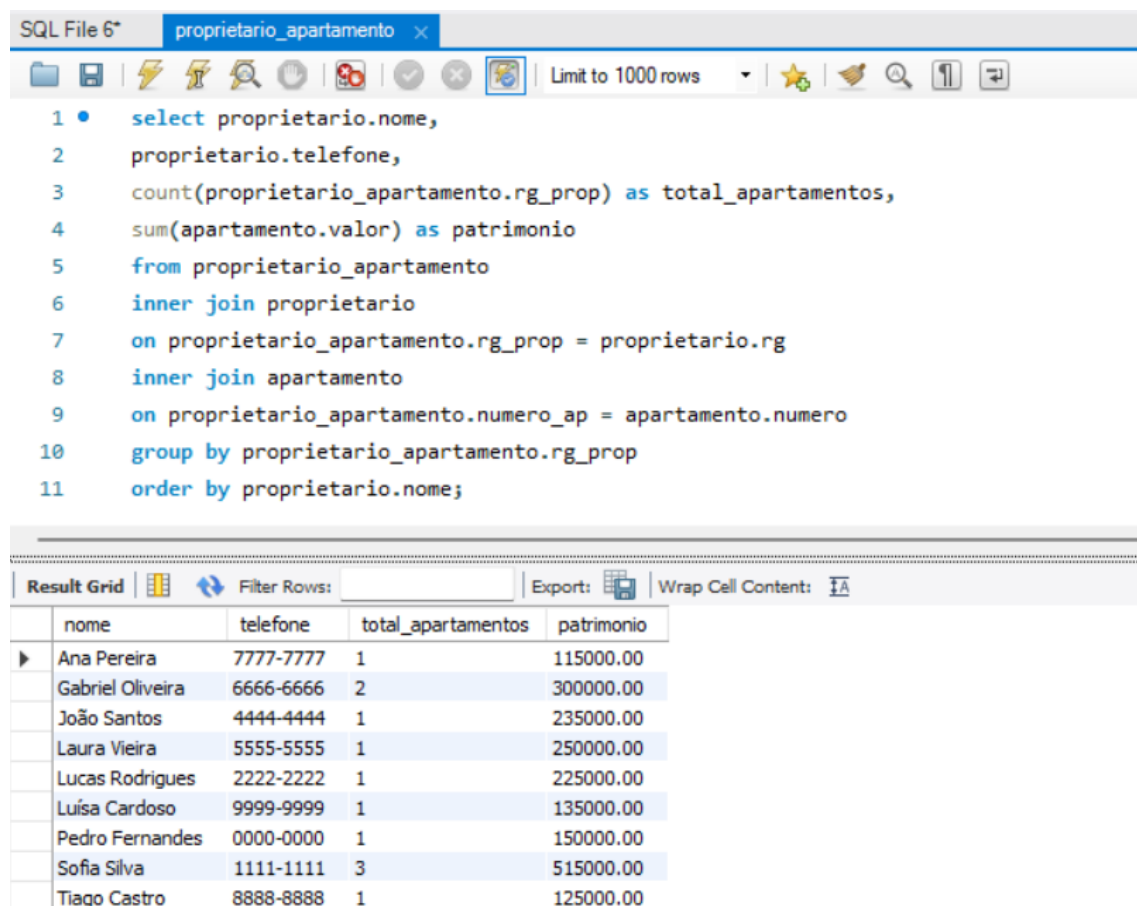
```
1 update apartamento set numero = 'A101', valor = 100000.00 where numero = '101';
2 • update apartamento set numero = 'A201', valor = 115000.00 where numero = '102';
3 • update apartamento set numero = 'A301', valor = 125000.00 where numero = '201';
4 • update apartamento set numero = 'A401', valor = 135000.00 where numero = '202';
5 • update apartamento set numero = 'A501', valor = 150000.00 where numero = '301';
6 • update apartamento set numero = 'B101', valor = 200000.00 where numero = '11';
7 • update apartamento set numero = 'B201', valor = 215000.00 where numero = '12';
8 • update apartamento set numero = 'B301', valor = 225000.00 where numero = '21';
9 • update apartamento set numero = 'B401', valor = 235000.00 where numero = '22';
10 • update apartamento set numero = 'B501', valor = 250000.00 where numero = '100';
11 • select * from apartamento;
```

numero	tipo	codigo_condominio	valor
A101	Padrão	1	100000.00
A201	Padrão	1	115000.00
A301	Padrão	1	125000.00
A401	Padrão	1	135000.00
A501	Cobertura	1	150000.00
B101	Padrão	2	200000.00
B201	Padrão	2	215000.00
B301	Padrão	2	225000.00
B401	Padrão	2	235000.00
B501	Cobertura	2	250000.00
NULL	NULL	NULL	NULL

Vale ressaltar que o campo “numero” trata-se de chave estrangeira de outras duas tabelas (garagem e proprietário_apartamento). Portanto, para realizar as alterações, alterei as chaves estrangeiras para ON UPDATE CASCADE (para que, ao modificar um registro em uma tabela, um registro relacionado em uma coluna de chave estrangeira em outra tabela tenha o valor automaticamente atualizado).

4) Relacione os proprietários apresentando o nome, o telefone, o total de apartamentos que ele possui e o valor total do seu patrimônio considerando somente os apartamentos dos dois condomínios, por ordem alfabética.

```
select proprietario.nome,  
proprietario.telefone,  
count(proprietario_apartamento.rg_prop) as total_apartamentos,  
sum(apartamento.valor) as patrimonio  
from proprietario_apartamento  
inner join proprietario  
on proprietario_apartamento.rg_prop = proprietario.rg  
inner join apartamento  
on proprietario_apartamento.numero_ap = apartamento.numero  
group by proprietario_apartamento.rg_prop  
order by proprietario.nome;
```



The screenshot shows a SQL IDE window titled "SQL File 6*" with a tab for "proprietario_apartamento". The query is entered in the editor and is identical to the one in the previous block. Below the editor, the "Result Grid" is displayed, showing the results of the query. The results are sorted alphabetically by owner name. The table has five columns: nome, telefone, total_apartamentos, and patrimonio. The data is as follows:

nome	telefone	total_apartamentos	patrimonio
Ana Pereira	7777-7777	1	115000.00
Gabriel Oliveira	6666-6666	2	300000.00
João Santos	4444-4444	1	235000.00
Laura Vieira	5555-5555	1	250000.00
Lucas Rodrigues	2222-2222	1	225000.00
Lúisa Cardoso	9999-9999	1	135000.00
Pedro Fernandes	0000-0000	1	150000.00
Sofia Silva	1111-1111	3	515000.00
Tiago Castro	8888-8888	1	125000.00