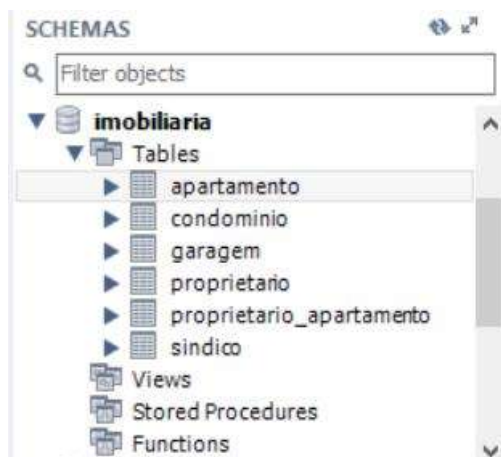


Tecnologia da Informação II	Modulo II
Fichário	Agenda 4 – Linguagem de manipulação de dados - DML
Nome: Mônica Zungalo Quintal	

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 01 a 06 – Interface Gráfica Workbench – Estruturas da Base de Dados da Imobiliária São João

Síndico

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

Imagem 01

Proprietário

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	

Condomínio

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 03

Apartamento

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

Imagem 04

Garagem

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 05

Proprietario_Apartamento

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 06

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) Insira os dados de 2 (dois) **Síndicos**.
- 2) Insira os dados de 10 (dez) **Proprietários**.
- 3) Insira os dados de 2 (dois) **Condomínios** definido um **Síndico** para cada um.
- 4) Insira os dados de 5 (cinco) **Apartamentos** para cada **Condomínio** com uma cobertura em cada um.
- 5) Insira os dados de **Garagem** sabendo-se que um dos **Condomínios** possui duas vagas por **Apartamento**, enquanto o outro só uma e que somente quem mora no **Apartamento** do tipo cobertura, possui vaga coberta.
- 6) Insira os registros necessários para distribuir os **Apartamentos** entre os **Proprietários**, sabendo-se que:
 - a) nenhum **Proprietário** pode ficar sem **Apartamento**,
 - b) um **Proprietário** pode possuir mais de um **Apartamento** e
 - c) um **Apartamento** por ter mais de um **Proprietário**.
- 7) Utilize o comando de alteração para simular a venda de pelo menos 2 (dois) **Apartamentos**, fazendo com que pelo um **Proprietário** fique sem apartamento.
- 8) Utilize o comando de exclusão para excluir o registro do **Proprietário** que ficou sem **Apartamento**.

Criando o Banco de Dados “imobiliaria” e suas**respectivas tabelas, conforme enunciado:**

```
create database imobiliaria;
```

```
create table sindico (  
    matricula integer(3) not null auto_increment,  
    nome varchar(80),  
    endereco varchar(80),  
    telefone varchar(15),  
    constraint pk_matricula primary key (matricula)  
);
```

```
create table proprietario (  
    rg varchar(15) not null,  
    email varchar(50),  
    telefone varchar(15),  
    nome varchar(80),  
    constraint pk_rg primary key (rg)  
);
```

```
create table condominio (  
    codigo int(5) not null auto_increment,  
    endereco varchar(80),  
    nome varchar(50),  
    matricula_sindico int(3),  
    constraint pk_codigo primary key (codigo),  
    constraint fk_matricula_sindico foreign key (matricula_sindico) references sindico(matricula)  
);
```

```
create table apartamento (  
    numero varchar(5) not null,  
    tipo varchar(20),  
    codigo_condominio int(5),  
    constraint pk_numero primary key (numero),  
    constraint fk_codigo_condominio foreign key (codigo_condominio) references condominio (codigo)  
);
```

```
create table garagem (  
    numero int(3) not null auto_increment,  
    tipo varchar(20),  
    numero_apto varchar(5) not null,  
    constraint pk_numero primary key (numero),  
    constraint fk_numero_apto foreign key (numero_apto) references apartamento(numero)  
);
```

```
create table proprietario_apartamento (  
    prop_ap_id int(3) not null auto_increment,  
    numero_ap varchar(5),  
    rg_prop varchar(15),
```

```
constraint pk_prop_ap_id primary key (prop_ap_id),  
constraint fk_numero_ap foreign key (numero_ap) references  
apartamento(numero),  
constraint fk_rg_prop foreign key (rg_prop) references proprietario(rg)  
);
```

Exercícios:

A. Insira os dados de 2 (dois) **Síndicos**.

```
insert into sindico (nome, endereco, telefone) values ('José da Silva', 'Rua A, 110 - ABC', '1234-5678');
```

```
insert into sindico (nome, endereco, telefone) values ('Maria de Souza', 'Rua B, 220 - DEF', '5678-1234');
```

```
select * from sindico;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 insert into sindico (nome, endereco, telefone) values ('José da Silva', 'Rua A, 110 - ABC', '1234-5678');  
2 insert into sindico (nome, endereco, telefone) values ('Maria de Souza', 'Rua B, 220 - DEF', '5678-1234');  
3  
4 select * from sindico;
```

The Result Grid shows the output of the SELECT query:

matricula	nome	endereco	telefone
1	José da Silva	Rua A, 110 - ABC	1234-5678
2	Maria de Souza	Rua B, 220 - DEF	5678-1234

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	13:25:08	describe sindico	4 row(s) returned	0.000 sec / 0.000 sec
2	13:29:25	insert into sindico (nome, endereco, telefone) values ('José da Silva', 'Rua A, 110 - ABC', '1234-56...	1 row(s) affected	0.016 sec
3	13:29:25	insert into sindico (nome, endereco, telefone) values ('Maria de Souza', 'Rua B, 220 - DEF', '5678...	1 row(s) affected	0.000 sec
4	13:29:25	select * from sindico LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

B. Insira os dados de 10 (dez) **Proprietários**.

```
insert into proprietario (rg, email, telefone, nome) values ('123456789', 'sofia_silva@gmail.com', '1111-1111', 'Sofia Silva');
insert into proprietario (rg, email, telefone, nome) values ('234567890', 'lrodrigues@hotmail.com', '2222-2222', 'Lucas Rodrigues');
insert into proprietario (rg, email, telefone, nome) values ('345678901', 'costa.maria@uol.com', '3333-3333', 'Maria Costa');
insert into proprietario (rg, email, telefone, nome) values ('456789012', 'jsantos@gmail.com', '4444-4444', 'João Santos');
insert into proprietario (rg, email, telefone, nome) values ('567890123', 'laurav@ig.com', '5555-5555', 'Laura Vieira');
insert into proprietario (rg, email, telefone, nome) values ('678901234', 'gabrieloliveira@gmail.com', '6666-6666', 'Gabriel Oliveira');
insert into proprietario (rg, email, telefone, nome) values ('789012345', 'ana.pereira@yahoo.com', '7777-7777', 'Ana Pereira');
insert into proprietario (rg, email, telefone, nome) values ('890123456', 'tcastro@gmail.com', '8888-8888', 'Tiago Castro');
insert into proprietario (rg, email, telefone, nome) values ('901234567', 'lucardoso@outlook.com', '9999-9999', 'Luísa Cardoso');
insert into proprietario (rg, email, telefone, nome) values ('012345678', 'pedrof_@gmail.com', '0000-0000', 'Pedro Fernandes');

select * from proprietario;
```

The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tree with a filter on 'imobiliaria'. The main area contains an 'SQL File' editor with 12 lines of SQL code, including 10 INSERT statements and a final SELECT statement. Below the editor is the 'Result Grid' showing the output of the queries. The first query (line 1) returns one row for 'Pedro Fernandes'. The second query (line 2) returns one row for 'Sofia Silva'. The third query (line 3) returns one row for 'Lucas Rodrigues'. The fourth query (line 4) returns one row for 'Maria Costa'. The fifth query (line 5) returns one row for 'João Santos'. The sixth query (line 6) returns one row for 'Laura Vieira'. The seventh query (line 7) returns one row for 'Gabriel Oliveira'. The eighth query (line 8) returns one row for 'Ana Pereira'. The ninth query (line 9) returns one row for 'Tiago Castro'. The tenth query (line 10) returns one row for 'Luísa Cardoso'. The eleventh query (line 11) returns one row for 'Pedro Fernandes'. The twelfth query (line 12) returns all rows from the 'proprietario' table. The 'Result Grid' shows the following data:

rg	email	telefone	nome
012345678	pedrof_@gmail.com	0000-0000	Pedro Fernandes
123456789	sofia_silva@gmail.com	1111-1111	Sofia Silva
234567890	lrodrigues@hotmail.com	2222-2222	Lucas Rodrigues
345678901	costa.maria@uol.com	3333-3333	Maria Costa
456789012	jsantos@gmail.com	4444-4444	João Santos
567890123	laurav@ig.com	5555-5555	Laura Vieira
678901234	gabrieloliveira@gmail.com	6666-6666	Gabriel Oliveira
789012345	ana.pereira@yahoo.com	7777-7777	Ana Pereira
890123456	tcastro@gmail.com	8888-8888	Tiago Castro
901234567	lucardoso@outlook.com	9999-9999	Luísa Cardoso

The bottom of the interface shows the 'Administration' tab with 'Schemas' selected. The 'Table: sindico' is highlighted, showing its columns: matricula (int AI PK), nome (varchar(80)), endereco (varchar(80)), and telefone (varchar(15)). The 'Output' pane at the bottom shows the execution of the queries, with messages indicating the number of rows affected and the duration of each query.

C. Insira os dados de 2 (dois) **Condomínios** definido um **Síndico** para cada um.

insert into condominio (endereco, nome, matricula_sindico) values ('Residencial Jardim das Flores', 'Rua das Acácias, nº 25', 1);

insert into condominio (endereco, nome, matricula_sindico) values ('Condomínio Praia do Sol', 'Avenida dos Ipês, nº 10', 2);

select * from condominio;

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 insert into condominio (endereco, nome, matricula_sindico) values ('Residencial Jardim das Flores', 'Rua das Acácias, nº 25', 1);
2 insert into condominio (endereco, nome, matricula_sindico) values ('Condomínio Praia do Sol', 'Avenida dos Ipês, nº 10', 2);
3
4 select * from condominio;
```

The Results window displays the data from the 'condominio' table:

codigo	endereco	nome	matricula_sindico
1	Residencial Jardim das Flores	Rua das Acácias, nº 25	1
2	Condomínio Praia do Sol	Avenida dos Ipês, nº 10	2

The left sidebar shows the database structure, including the 'condominio' table. The bottom status bar shows the execution of the queries, with the first query affecting 1 row and the second query returning 2 rows.

D. Insira os dados de 5 (cinco) **Apartamentos** para cada **Condomínio** com uma cobertura em cada um.

```
insert into apartamento (numero, tipo, codigo_condominio) values ('101', 'Padrão', '1');
insert into apartamento (numero, tipo, codigo_condominio) values ('102', 'Padrão', '1');
insert into apartamento (numero, tipo, codigo_condominio) values ('201', 'Padrão', '1');
insert into apartamento (numero, tipo, codigo_condominio) values ('202', 'Padrão', '1');
insert into apartamento (numero, tipo, codigo_condominio) values ('301', 'Cobertura', '1');
insert into apartamento (numero, tipo, codigo_condominio) values ('11', 'Padrão', '2');
insert into apartamento (numero, tipo, codigo_condominio) values ('12', 'Padrão', '2');
insert into apartamento (numero, tipo, codigo_condominio) values ('21', 'Padrão', '2');
insert into apartamento (numero, tipo, codigo_condominio) values ('22', 'Padrão', '2');
insert into apartamento (numero, tipo, codigo_condominio) values ('100', 'Cobertura', '2');
```

```
select * from apartamento;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 insert into apartamento (numero, tipo, codigo_condominio) values ('101', 'Padrão', '1');
2 insert into apartamento (numero, tipo, codigo_condominio) values ('102', 'Padrão', '1');
3 insert into apartamento (numero, tipo, codigo_condominio) values ('201', 'Padrão', '1');
4 insert into apartamento (numero, tipo, codigo_condominio) values ('202', 'Padrão', '1');
5 insert into apartamento (numero, tipo, codigo_condominio) values ('301', 'Cobertura', '1');
6 insert into apartamento (numero, tipo, codigo_condominio) values ('11', 'Padrão', '2');
7 insert into apartamento (numero, tipo, codigo_condominio) values ('12', 'Padrão', '2');
8 insert into apartamento (numero, tipo, codigo_condominio) values ('21', 'Padrão', '2');
9 insert into apartamento (numero, tipo, codigo_condominio) values ('22', 'Padrão', '2');
10 insert into apartamento (numero, tipo, codigo_condominio) values ('100', 'Cobertura', '2');
11
12 select * from apartamento;
```

The Result Grid shows the following data:

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

The Output tab shows the following messages:

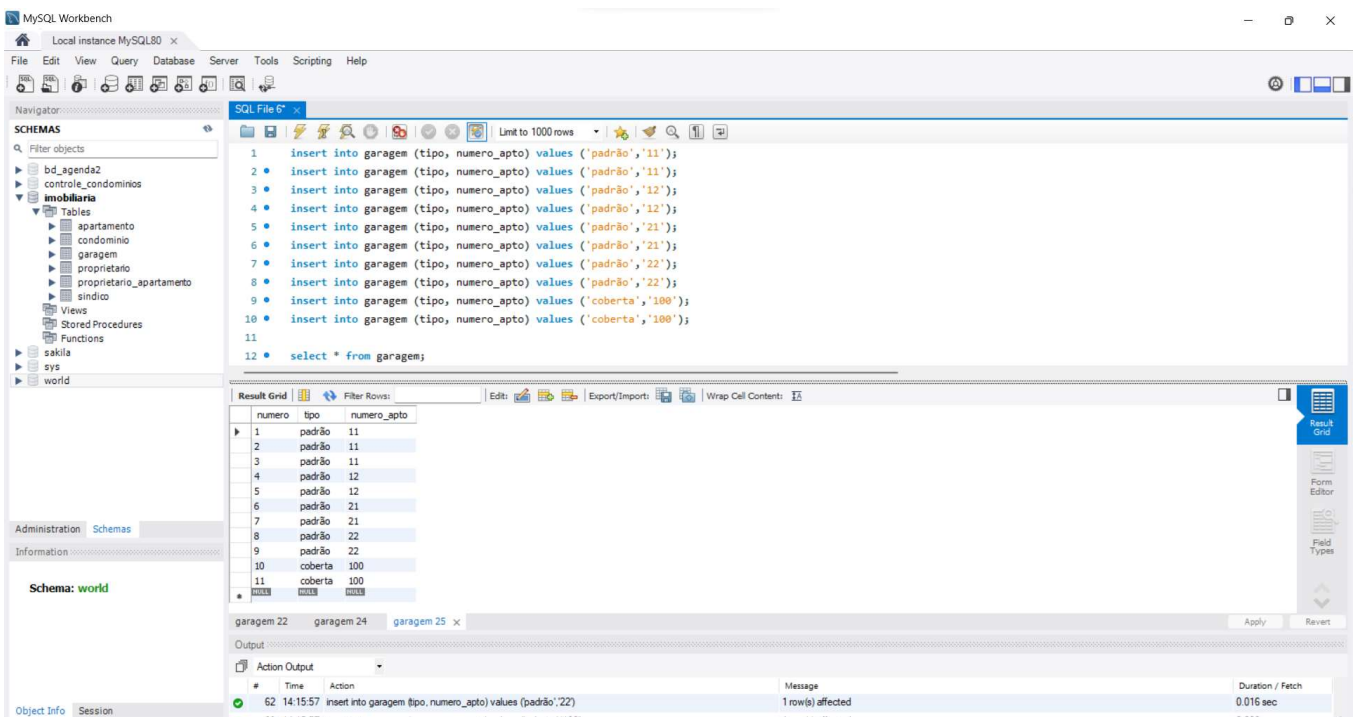
#	Time	Action	Message	Duration / Fetch
39	14:04:17	insert into apartamento (numero, tipo, codigo_condominio) values ('100', 'Cobertura', '2')	1 row(s) affected	0.000 sec
40	14:04:17	select * from apartamento LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

E. Insira os dados de **Garagem** sabendo-se que um dos **Condomínios** possui duas vagas por **Apartamento**, enquanto o outro só uma e que somente quem mora no **Apartamento** do tipo cobertura, possui vaga coberta.

i. Condomínio que possui 2 vagas por apto:

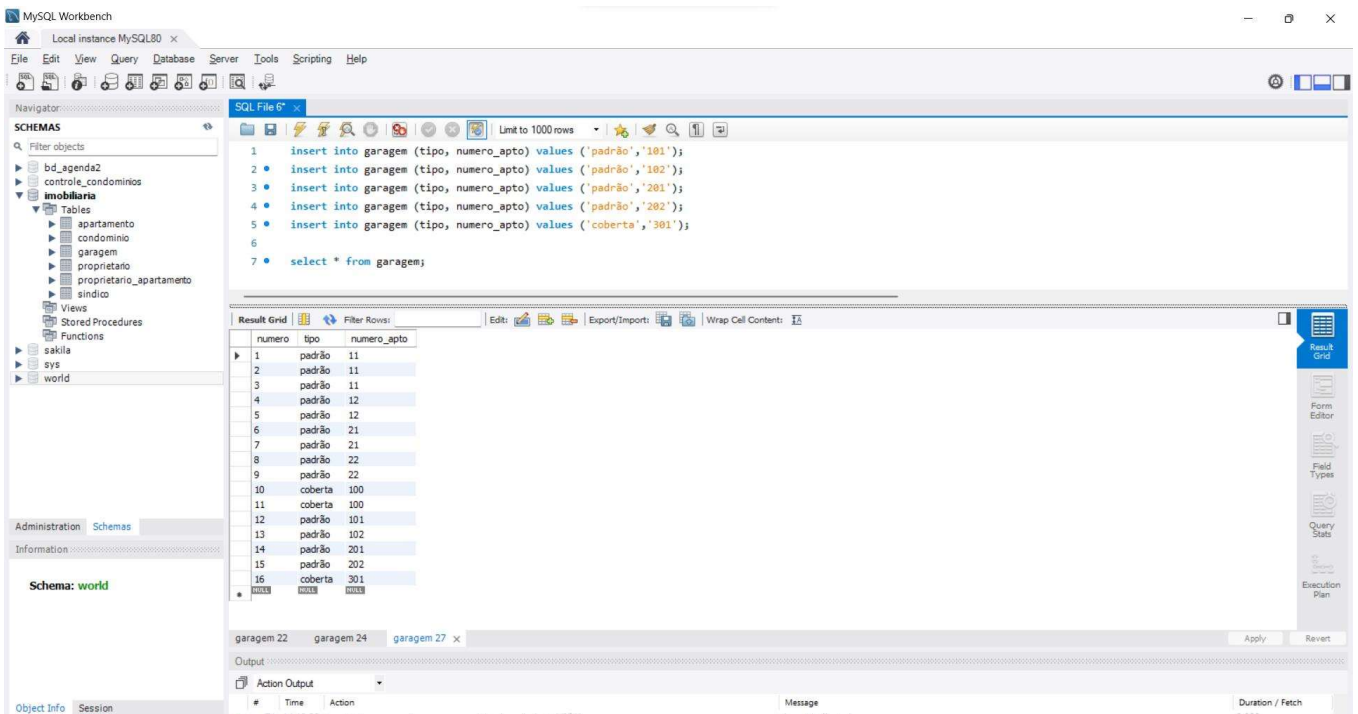
```
insert into garagem (tipo, numero_apto) values ('padrão','11');
insert into garagem (tipo, numero_apto) values ('padrão','11');
insert into garagem (tipo, numero_apto) values ('padrão','12');
insert into garagem (tipo, numero_apto) values ('padrão','12');
insert into garagem (tipo, numero_apto) values ('padrão','21');
insert into garagem (tipo, numero_apto) values ('padrão','21');
insert into garagem (tipo, numero_apto) values ('padrão','22');
insert into garagem (tipo, numero_apto) values ('padrão','22');
insert into garagem (tipo, numero_apto) values ('coberta','100');
insert into garagem (tipo, numero_apto) values ('coberta','100');
```

```
select * from garagem;
```



ii. Condomínio que possui apenas uma vaga de garagem por apartamento:

```
insert into garagem (tipo, numero_apto) values ('padrão','101');  
insert into garagem (tipo, numero_apto) values ('padrão','102');  
insert into garagem (tipo, numero_apto) values ('padrão','201');  
insert into garagem (tipo, numero_apto) values ('padrão','202');  
insert into garagem (tipo, numero_apto) values ('coberta','301');  
  
select * from garagem;
```



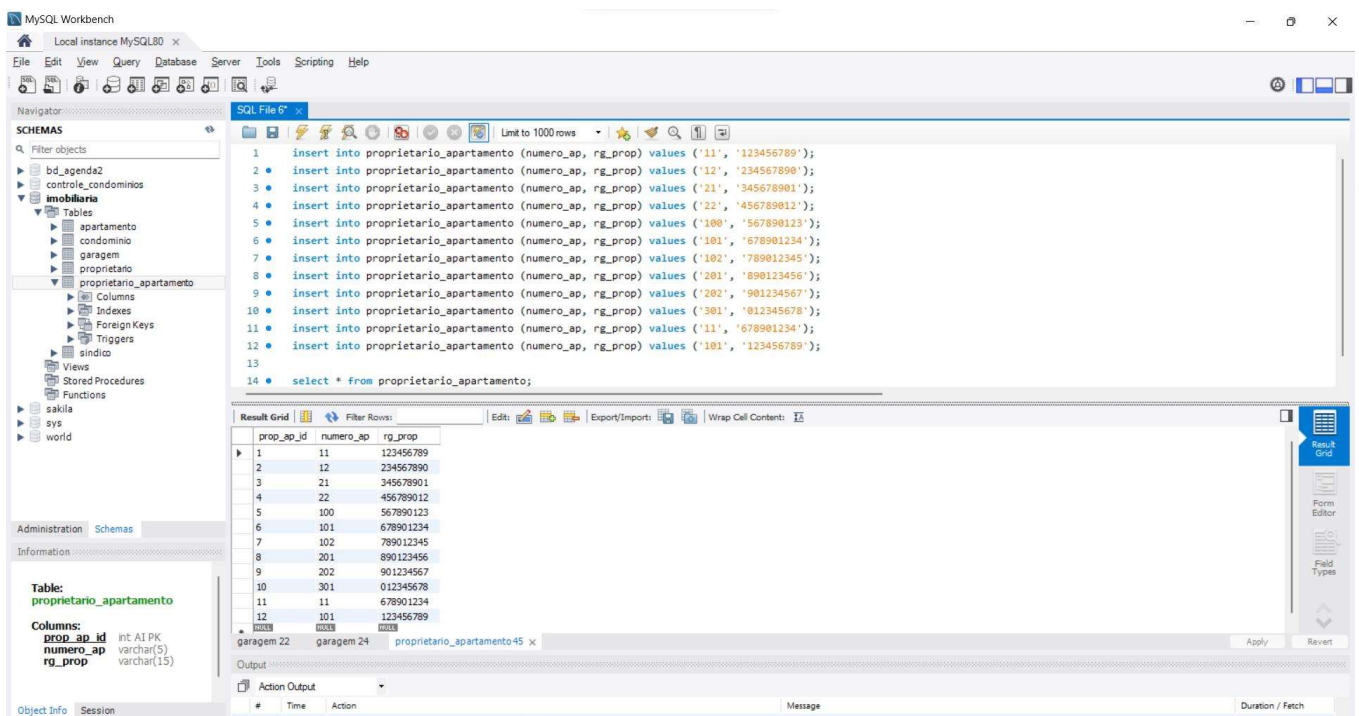
F. Insira os registros necessários para distribuir os

Apartamentos entre os **Proprietários**, sabendo-se que:

- nenhum **Proprietário** pode ficar sem **Apartamento**,
- um **Proprietário** pode possuir mais de um **Apartamento** e
- um **Apartamento** pode ter mais de um **Proprietário**.

```
insert into proprietario_apartamento (numero_ap, rg_prop) values ('11', '123456789');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('12', '234567890');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('21', '345678901');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('22', '456789012');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('100', '567890123');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('101', '678901234');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('102', '789012345');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('201', '890123456');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('202', '901234567');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('301', '012345678');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('11', '678901234');
insert into proprietario_apartamento (numero_ap, rg_prop) values ('101', '123456789');

select * from proprietario_apartamento;
```



G. Utilize o comando de alteração para simular a

venda de pelo menos 2 (dois) **Apartamentos**, fazendo com que pelo menos um **Proprietário** fique sem apartamento.

```
update proprietario_apartamento set rg_prop = '123456789' where prop_ap_id = 2;
```

```
update proprietario_apartamento set rg_prop = '234567890' where prop_ap_id = 3;
```

```
select * from proprietario_apartamento;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 update proprietario_apartamento set rg_prop = '123456789' where prop_ap_id = 2;
2 update proprietario_apartamento set rg_prop = '234567890' where prop_ap_id = 3;
3
4 select * from proprietario_apartamento;
```

The Result Grid displays the following data:

prop_ap_id	numero_ap	rg_prop
1	11	123456789
2	12	123456789
3	21	234567890
4	22	456789012
5	100	567890123
6	101	678901234
7	102	789012345
8	201	890123456
9	202	901234567
10	301	012345678
11	11	678901234
12	101	123456789

The Action Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
108	14:46:57	select * from proprietario_apartamento LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
109	14:47:42	select * from proprietario_apartamento LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
110	14:48:32	update proprietario_apartamento set rg_prop = '123456789' where prop_ap_id = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
111	14:48:32	update proprietario_apartamento set rg_prop = '234567890' where prop_ap_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
112	14:48:32	select * from proprietario_apartamento LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec

H. Utilize o comando de exclusão para excluir o registro do **Proprietário** que ficou sem **Apartamento**.

```
delete from proprietario where rg = '345678901';
```

```
select * from proprietario;
```

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 delete from proprietario where rg = '345678901';
2
3 select * from proprietario;
```

The Result Grid displays the following data:

rg	email	telefone	nome
012345678	pedrof@gmail.com	0000-0000	Pedro Fernandes
123456789	sofia_silva@gmail.com	1111-1111	Sofia Silva
234567890	lrodrigues@hotmail.com	2222-2222	Lucas Rodrigues
456789012	jsantos@gmail.com	4444-4444	João Santos
567890123	laurav@ig.com	5555-5555	Laura Vieira
678901234	gabrieloliveira@gmail.com	6666-6666	Gabriel Oliveira
789012345	ana.pereira@yahoo.com	7777-7777	Ana Pereira
890123456	lcastro@gmail.com	8888-8888	Tiago Castro
901234567	lucardoso@outlook.com	9999-9999	Luisa Cardoso

The Action Output pane shows the following log:

#	Time	Action	Message	Duration / Fetch
111	14:48:32	update proprietario_apartamento set rg_prop = '234567890' where prop_ap_id = 3	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
112	14:48:32	select * from proprietario_apartamento LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
113	14:50:13	select * from proprietario LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
114	14:51:24	delete from proprietario where rg = '345678901'	1 row(s) affected	0.016 sec
115	14:51:24	select * from proprietario LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec