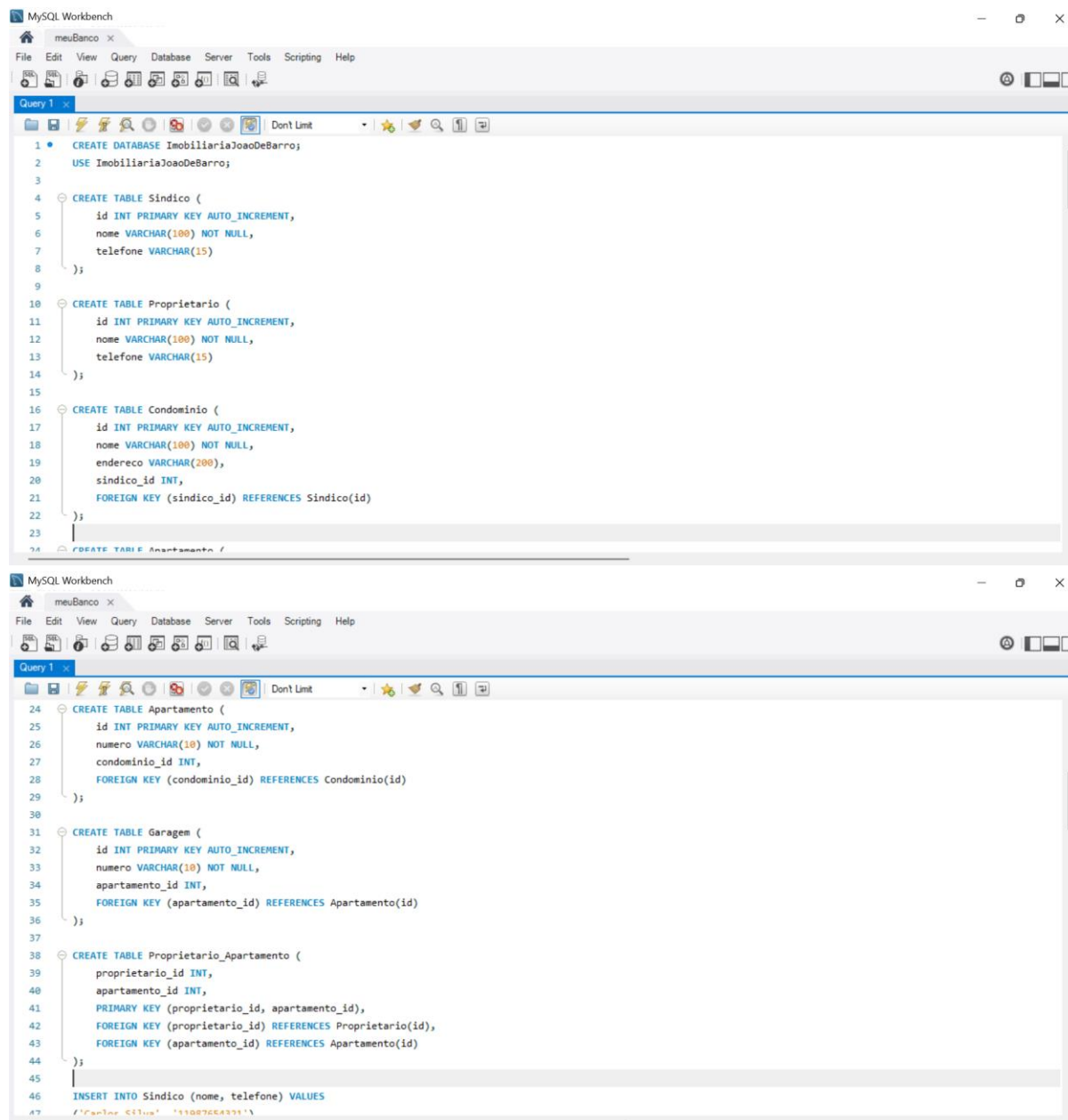


Desenvolvimento do Banco de Dados Imobiliária João de Barro

1. Criação do Banco de Dados e Tabelas

O primeiro passo foi criar o banco de dados **ImobiliariaJoaoDeBarro** e, em seguida, as tabelas necessárias para armazenar as informações da imobiliária. As tabelas criadas foram:



The image displays two screenshots of the MySQL Workbench interface. The top screenshot shows the creation of the database and three tables: Sincico, Proprietario, and Condominio. The bottom screenshot shows the creation of the remaining three tables: Apartamento, Garagem, and Proprietario_Apartamento, along with an initial data insertion into the Sincico table.

```
1 CREATE DATABASE ImobiliariaJoaoDeBarro;
2 USE ImobiliariaJoaoDeBarro;
3
4 CREATE TABLE Sincico (
5     id INT PRIMARY KEY AUTO_INCREMENT,
6     nome VARCHAR(100) NOT NULL,
7     telefone VARCHAR(15)
8 );
9
10 CREATE TABLE Proprietario (
11     id INT PRIMARY KEY AUTO_INCREMENT,
12     nome VARCHAR(100) NOT NULL,
13     telefone VARCHAR(15)
14 );
15
16 CREATE TABLE Condominio (
17     id INT PRIMARY KEY AUTO_INCREMENT,
18     nome VARCHAR(100) NOT NULL,
19     endereco VARCHAR(200),
20     sincico_id INT,
21     FOREIGN KEY (sincico_id) REFERENCES Sincico(id)
22 );
23
24 CREATE TABLE Apartamento (
25     id INT PRIMARY KEY AUTO_INCREMENT,
26     numero VARCHAR(10) NOT NULL,
27     condominio_id INT,
28     FOREIGN KEY (condominio_id) REFERENCES Condominio(id)
29 );
30
31 CREATE TABLE Garagem (
32     id INT PRIMARY KEY AUTO_INCREMENT,
33     numero VARCHAR(10) NOT NULL,
34     apartamento_id INT,
35     FOREIGN KEY (apartamento_id) REFERENCES Apartamento(id)
36 );
37
38 CREATE TABLE Proprietario_Apartamento (
39     proprietario_id INT,
40     apartamento_id INT,
41     PRIMARY KEY (proprietario_id, apartamento_id),
42     FOREIGN KEY (proprietario_id) REFERENCES Proprietario(id),
43     FOREIGN KEY (apartamento_id) REFERENCES Apartamento(id)
44 );
45
46 INSERT INTO Sincico (nome, telefone) VALUES
47 ('João de Barro', '11987654321');
```

ESSAS TABELAS FORAM CRIADAS COM CHAVES PRIMÁRIAS E ESTRANGEIRAS PARA GARANTIR A INTEGRIDADE DOS DADOS E OS RELACIONAMENTOS ENTRE AS ENTIDADES.

2. Inserção de Dados de Exemplo

Para testar as consultas e atualizações, foram inseridos dados de exemplo em cada tabela. Por exemplo:

The image displays two screenshots of the MySQL Workbench interface, showing SQL queries for a database schema. The top screenshot shows the first part of the queries, and the bottom screenshot shows the second part.

Top Screenshot:

```
46 INSERT INTO Sincico (nome, telefone) VALUES
47 ('Carlos Silva', '11987654321'),
48 ('Ana Souza', '11912345678');
49
50 INSERT INTO Proprietario (nome, telefone) VALUES
51 ('João Pereira', '11911112222'),
52 ('Maria Oliveira', '11933334444'),
53 ('Pedro Santos', '11955556666');
54
55 INSERT INTO Condominio (nome, endereco, sindico_id) VALUES
56 ('Condominio 1', 'Rua das Flores, 123', 1),
57 ('Condominio 2', 'Avenida dos Lagos, 456', 2);
58
59 INSERT INTO Apartamento (numero, condominio_id) VALUES
60 ('A101', 1),
61 ('A201', 1),
62 ('A301', 1),
63 ('A401', 1),
64 ('A501', 1),
65 ('B101', 2),
66 ('B201', 2),
67 ('B301', 2),
68 ('B401', 2);
69
```

Bottom Screenshot:

```
70
71 INSERT INTO Garagem (numero, apartamento_id) VALUES
72 ('G01', 1),
73 ('G02', 2),
74 ('G03', 3),
75 ('G04', 4),
76 ('G05', 5),
77 ('G06', 6),
78 ('G07', 7),
79 ('G08', 8),
80 ('G09', 9),
81 ('G10', 10);
82
83 INSERT INTO Proprietario_Apartamento (proprietario_id, apartamento_id) VALUES
84 (1, 1),
85 (1, 2),
86 (2, 3),
87 (2, 4),
88 (3, 5),
89 (3, 6),
90 (1, 7),
91 (2, 8),
92 (3, 9),
93 (1, 10);
94
```

3. Tarefa 1: Relação de Condomínios com Nome, Endereço e Total de Apartamentos

Para atender a essa tarefa, foi desenvolvida a seguinte consulta SQL:

```

95 SELECT
96     c.nome AS Nome_Condominio,
97     c.endereco AS Endereco,
98     COUNT(a.id) AS Total_Apartamentos
99 FROM
100     Condominio c
101 LEFT JOIN
102     Apartamento a ON c.id = a.condominio_id
103 GROUP BY
104     c.id;

```

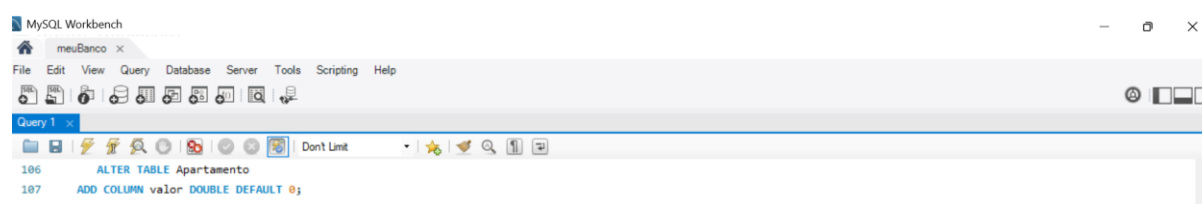
A consulta seleciona o nome e o endereço de cada condomínio.

Utiliza um **LEFT JOIN** para contar o número de apartamentos relacionados a cada condomínio.

O **GROUP BY** agrupa os resultados por condomínio, garantindo que o total de apartamentos seja calculado corretamente.

4. Tarefa 2: Alteração da Estrutura da Tabela Apartamento

Para adicionar o campo valor à tabela Apartamento, foi utilizado o seguinte comando SQL:



```

106 ALTER TABLE Apartamento
107 ADD COLUMN valor DOUBLE DEFAULT 0;

```

O comando **ALTER TABLE** adiciona uma nova coluna chamada valor à tabela Apartamento.

O tipo da coluna é **DOUBLE**, que permite armazenar valores monetários.

O **valor padrão** é definido como **0**, garantindo que todos os apartamentos tenham um valor inicial.

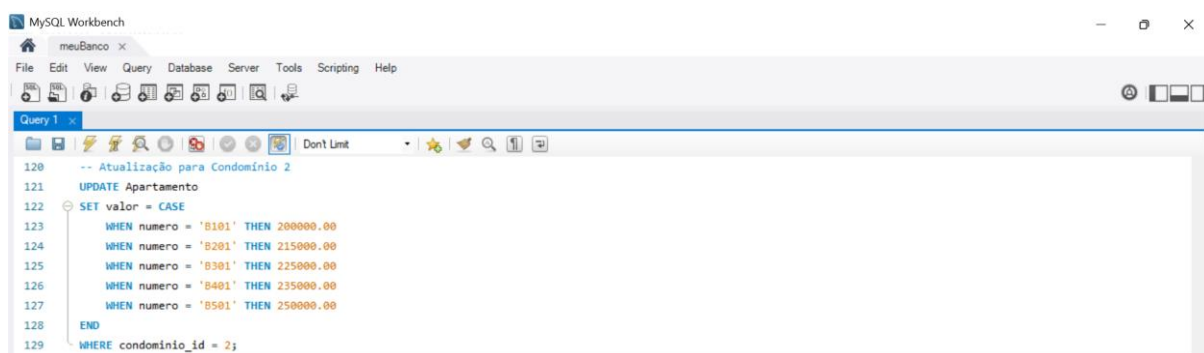
5. Tarefa 3: Atualização dos Valores dos Apartamentos

Para atualizar os valores dos apartamentos com base na tabela de preços fornecida, foram utilizados os seguintes comandos SQL:

```

109 -- Atualização para Condomínio 1
110 UPDATE Apartamento
111 SET valor = CASE
112     WHEN numero = 'A101' THEN 100000.00
113     WHEN numero = 'A201' THEN 115000.00
114     WHEN numero = 'A301' THEN 125000.00
115     WHEN numero = 'A401' THEN 135000.00
116     WHEN numero = 'A501' THEN 150000.00
117 END
118 WHERE condominio_id = 1;
119
120 -- Atualização para Condomínio 2
121 UPDATE Apartamento
122 SET valor = CASE
123     WHEN numero = 'B101' THEN 200000.00
124     WHEN numero = 'B201' THEN 215000.00
125     WHEN numero = 'B301' THEN 225000.00
126     WHEN numero = 'B401' THEN 235000.00
127     WHEN numero = 'B501' THEN 250000.00
128 END
129 WHERE condominio_id = 2;

```



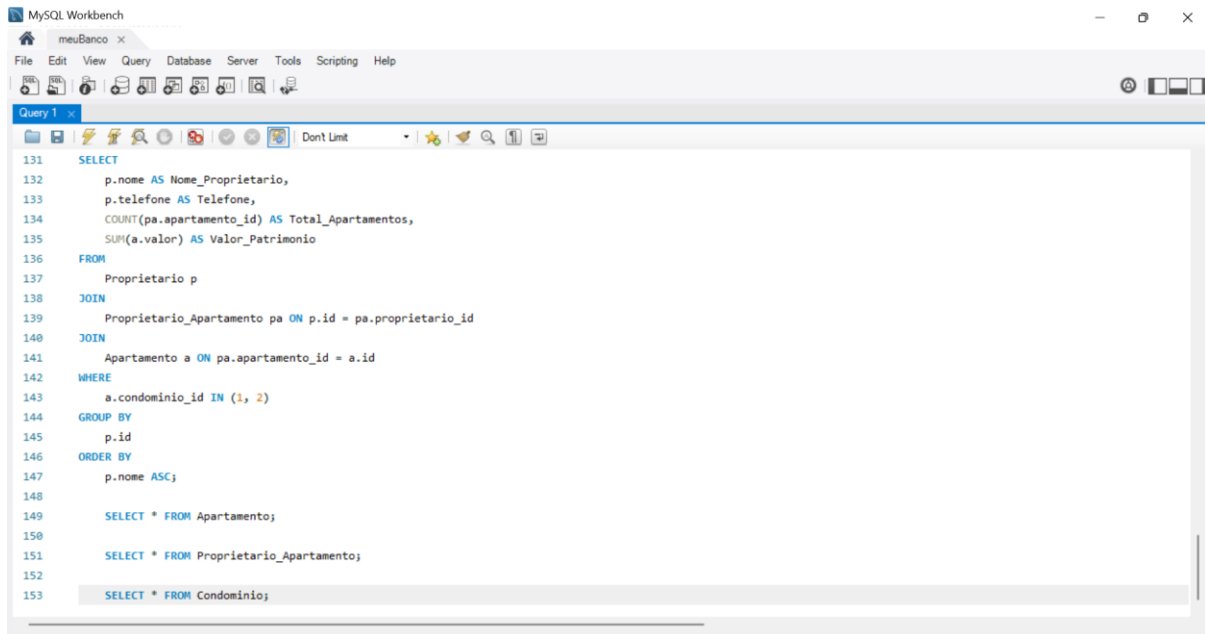
O comando **UPDATE** modifica os valores dos apartamentos com base no número do apartamento e no condomínio ao qual pertencem.

A cláusula **CASE** permite definir valores diferentes para cada apartamento.

A cláusula **WHERE** garante que apenas os apartamentos dos condomínios 1 e 2 sejam atualizados.

6. Tarefa 4: Relação de Proprietários com Nome, Telefone, Total de Apartamentos e Valor Total do Patrimônio

Para atender a essa tarefa, foi desenvolvida a seguinte consulta SQL:

A screenshot of the MySQL Workbench application window. The title bar says 'MySQL Workbench' and 'meuBanco'. The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar has icons for various database operations. The main editor area shows a SQL query with line numbers 131 to 153. The query is a complex SELECT statement with JOINs, WHERE, GROUP BY, and ORDER BY clauses. It also includes three separate SELECT statements at the bottom.

```
131 SELECT
132     p.nome AS Nome_Proprietario,
133     p.telefone AS Telefone,
134     COUNT(pa.apartamento_id) AS Total_Apartamentos,
135     SUM(a.valor) AS Valor_Patrimonio
136 FROM
137     Proprietario p
138 JOIN
139     Proprietario_Apartamento pa ON p.id = pa.proprietario_id
140 JOIN
141     Apartamento a ON pa.apartamento_id = a.id
142 WHERE
143     a.condominio_id IN (1, 2)
144 GROUP BY
145     p.id
146 ORDER BY
147     p.nome ASC;
148
149 SELECT * FROM Apartamentos;
150
151 SELECT * FROM Proprietario_Apartamento;
152
153 SELECT * FROM Condominio;
```

A consulta seleciona o nome e o telefone de cada proprietário.

Utiliza **JOIN** para relacionar as tabelas **Proprietario**, **Proprietario_Apartamento** e **Apartamento**.

A função **COUNT** conta o número de apartamentos que cada proprietário possui.

A função **SUM** calcula o valor total do patrimônio de cada proprietário, considerando apenas os apartamentos dos condomínios 1 e 2.

O **GROUP BY** agrupa os resultados por proprietário, e o **ORDER BY** ordena os resultados em ordem alfabética pelo nome do proprietário.