

Modelo Relacional

Tabela: clientes

- **id_cliente**: Identificador único do cliente.
- **nome**: Nome do cliente.
- **email**: Email do cliente (único).
- **telefone**: Número de telefone do cliente.
- **endereco**: Endereço do cliente.

Tabela: veiculos

- **id_veiculo**: Identificador único do veículo.
- **modelo**: Modelo do veículo.
- **tipo**: Tipo do veículo (Carro ou Moto).
- **preco_diario**: Preço de locação diário do veículo.

Tabela: locacoes

- **id_locacao**: Identificador único da locação.
- **id_cliente**: Referência para a tabela clientes (relacionamento com a tabela clientes).
- **id_veiculo**: Referência para a tabela veiculos (relacionamento com a tabela veiculos).
- **data_inicio**: Data de início da locação.
- **data_fim**: Data de fim da locação.
- **valor_total**: Valor total da locação.

Tabela: precos

- **id_preco**: Identificador único do preço.
- **id_veiculo**: Referência para a tabela veiculos.
- **preco_diario**: Preço diário de locação do veículo.

Justificativas de Modelagem

- **Relacionamento entre clientes e locações**: Um cliente pode realizar várias locações, mas cada locação pertence a um único cliente.
- **Relacionamento entre veículos e locações**: Cada locação está associada a um veículo, e um veículo pode ser locado várias vezes.
- **Tabela de preços**: A tabela de preços foi separada para permitir alterações no preço sem afetar diretamente a tabela de veículos.

Prints dos comandos SLQ:

```
94 ----- parte 4 -----
95 -- Buscar veículos do tipo 'Carro' com preço acima de 100 OU motos com preço abaixo de 70
96 v SELECT * FROM veiculos
97 WHERE (tipo = 'Carro' AND preco_diario > 100)
98      OR (tipo = 'Moto' AND preco_diario < 70);
```

Data Output Messages Notifications

	id_veiculo [PK] integer	modelo character varying (100)	tipo character varying (50)	preco_diario numeric (10,2)
1	2	Civic	Carro	120.00
2	3	NMAX 160	Moto	50.00
3	7	Bros 160	Moto	60.00
4	8	Onix	Carro	110.00
5	9	Lander 250	Moto	65.00

```
100 -- Buscar todos os veículos que não são do tipo 'Moto'
101 SELECT * FROM veiculos
102 WHERE NOT tipo = 'Moto';
103
```

Data Output Messages Notifications

≡+

▼

▼

SQL

	id_veiculo [PK] integer	modelo character varying (100)	tipo character varying (50)	preco_diario numeric (10,2)
1	1	Corolla	Carro	100.00
2	2	Civic	Carro	120.00
3	4	Fusca	Carro	80.00
4	6	HB20	Carro	90.00
5	8	Onix	Carro	110.00
6	10	Kwid	Carro	85.00

```
104 -- Buscar clientes que não moram na 'Rua A, 123'
105 SELECT * FROM clientes
106 WHERE endereco <> 'Rua A, 123';
107
```

Data Output Messages Notifications

≡+

▼

▼

SQL

	id_cliente [PK] integer	nome character varying (100)	email character varying (100)	telefone character varying (20)	endereco character varying (200)
1	2	Maria Oliveira	maria@exemplo.com	988888888	Rua B, 456
2	3	Carlos Souza	carlos@exemplo.com	977777777	Rua C, 789
3	4	Fernanda Costa	fernanda@exemplo.com	966666666	Rua D, 101
4	5	Paulo Almeida	paulo@exemplo.com	955555555	Rua E, 202
5	6	Ana Lima	ana@exemplo.com	944444444	Rua F, 303
6	7	Lucas Pereira	lucas@exemplo.com	933333333	Rua G, 404
7	8	Juliana Santos	juliana@exemplo.com	922222222	Rua H, 505
8	9	Renato Carvalho	renato@exemplo.com	911111111	Rua I, 606
9	10	Tatiane Rocha	tatiane@exemplo.com	900000000	Rua J, 707

```
108 -- Buscar clientes cujo nome começa com a letra 'J'
109 SELECT * FROM clientes
110 WHERE nome LIKE 'J%';
111
```

Data Output Messages Notifications

≡+

▼

▼

SQL

	id_cliente [PK] integer	nome character varying (100)	email character varying (100)	telefone character varying (20)	endereco character varying (200)
1	8	Juliana Santos	juliana@exemplo.com	922222222	Rua H, 505
2	1	João Silva	joao@exemplo.com	977777777	Rua A, 123

```

121
122 -- Listar todos os veículos ordenados pelo preço diário em ordem decrescente
123 v SELECT * FROM veiculos
124 ORDER BY preco_diario DESC;
125

```

Data Output Messages Notifications

SQL

	id_veiculo [PK] integer	modelo character varying (100)	tipo character varying (50)	preco_diario numeric (10,2)
1	2	Civic	Carro	120.00
2	8	Onix	Carro	110.00
3	1	Corolla	Carro	100.00
4	6	HB20	Carro	90.00
5	10	Kwid	Carro	85.00
6	4	Fusca	Carro	80.00
7	5	XRE 300	Moto	70.00
8	9	Lander 250	Moto	65.00
9	7	Bros 160	Moto	60.00
10	3	NMAX 160	Moto	50.00

```

116 -- Mostrar clientes que fizeram mais de 1 locação
117 v SELECT id_cliente, COUNT(*) AS total
118 FROM locacoes
119 GROUP BY id_cliente
120 HAVING COUNT(*) > 3;
121

```

Data Output Messages Notifications

SQL

id_cliente integer	total bigint
-----------------------	-----------------

```

112 -- Buscar clientes com IDs 1, 2 ou 3
113 v SELECT * FROM clientes
114 WHERE id_cliente IN (1, 2, 3);
115

```

Data Output Messages Notifications

SQL

	id_cliente [PK] integer	nome character varying (100)	email character varying (100)	telefone character varying (20)	endereco character varying (200)
1	2	Maria Oliveira	maria@exemplo.com	988888888	Rua B, 456
2	3	Carlos Souza	carlos@exemplo.com	977777777	Rua C, 789
3	1	João Silva	joao@exemplo.com	977777777	Rua A, 123

```

134 -- Buscar o menor valor de locação
135 ✓ SELECT MIN(valor_total) AS locacao_mais_barata
136 FROM locacoes;
137
138

```

Data Output Messages Notifications



	locacao_mais_barata numeric
1	200.00

```

129
130 -- Buscar o valor máximo de uma locação
131 ✓ SELECT MAX(valor_total) AS locacao_mais_cara
132 FROM locacoes;
133

```

Data Output Messages Notifications



	locacao_mais_cara numeric
1	480.00

```

126 -- Calcular o preço médio das locações
127 ✓ SELECT AVG(valor_total) AS media_locacoes
128 FROM locacoes;
129

```

Data Output Messages Notifications



	media_locacoes numeric
1	350.000000000000000000