

Fernanda Silva Bucheri RA: 135529  
Marcos Ferreira Villar Coelho RA: 135534  
Magno Luiz Gonçalves Melo RA: 133688

# **Implementação de um Banco de Dados de uma Rodoviária**

São José dos Campos - Brasil

Fevereiro de 2021



Fernanda Silva Bucheri RA: 135529  
Marcos Ferreira Villar Coelho RA: 135534  
Magno Luiz Gonçalves Melo RA: 133688

## **Implementação de um Banco de Dados de uma Rodoviária**

Trabalho apresentado à Universidade Federal  
de São Paulo como parte dos requisitos para  
aprovação na disciplina de Banco de Dados.

Docente: Prof<sup>a</sup>. Dr<sup>a</sup>.: Daniela Leal Musa

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Fevereiro de 2021

# Resumo

O trabalho desenvolvido é um banco de dados de uma rodoviária implementado na linguagem *SQL*. Neste projeto analisamos as necessidades de um banco de dados para o funcionamento de uma rodoviária, incluindo o funcionamento geral de uma rodoviária e seus clientes.

**Palavras-chave:** banco de dados. *SQL*. rodoviária.

# Lista de ilustrações

Figura 1 – Primeiro modelo relacional baseado na ideia do projeto pré-implementação.	
.....	12
Figura 2 – Modelo relacional final do projeto utilizado para implementação.	14

# Lista de tabelas

Tabela 1 – Resultado da consulta Listing 5.1 . . . . .	30
Tabela 2 – Resultado da consulta Listing 5.2 . . . . .	31
Tabela 3 – Resultado da consulta Listing 5.3 . . . . .	32
Tabela 4 – Resultado da consulta Listing 5.4 . . . . .	33
Tabela 5 – Resultado da consulta Listing 5.5 . . . . .	34
Tabela 6 – Resultado da consulta Listing 5.6 . . . . .	35
Tabela 7 – Resultado da consulta Listing 5.7 . . . . .	35
Tabela 8 – Resultado da consulta Listing 5.8 . . . . .	36
Tabela 9 – Resultado da consulta Listing 5.9 . . . . .	37
Tabela 10 – Resultado da consulta Listing 5.10 . . . . .	37
Tabela 11 – Resultado da consulta Listing 5.11 . . . . .	38
Tabela 12 – Resultado da consulta Listing 5.12 . . . . .	38

# Sumário

1	INTRODUÇÃO . . . . .	7
I	IDEALIZAÇÃO DO PROJETO	9
2	ELABORAÇÃO DO PROJETO . . . . .	11
2.1	Objetivo . . . . .	11
2.2	Modelos . . . . .	11
II	REALIZAÇÃO DO PROJETO	17
3	CRIAÇÃO DO BANCO DE DADOS . . . . .	19
4	CADASTRO DE INFORMAÇÕES . . . . .	23
III	RESULTADOS	27
5	CONSULTAS . . . . .	29
5.1	5 consultas com os operadores básicos de <i>SQL</i> . . . . .	29
5.1.1	Exibir o nome do funcionário, o número da equipe que ele pertence e qual a sua função . . . . .	29
5.1.2	Exibir o <i>ID</i> das passagens que foram compradas em 2020 na rodoviária São Carlos	31
5.1.3	Exibir o nome do funcionário, a função dele, a loja que ele trabalha e o segmento . . . . .	32
5.1.4	Exibir <i>ID</i> da passagem, nome do passageiro que possui essa passagem, origem e o destino da viagem e o nome do atendente que vendeu a passagem	33
5.1.5	Exibir a origem e destino da viagem, nome do motorista, <i>ID</i> do ônibus e o modelo do ônibus que realizou a viagem . . . . .	34
5.2	2 consultas com <i>LEFT JOIN</i> . . . . .	34
5.2.1	Exibir o id da passagem do passageiro . . . . .	34
5.2.2	Exibir o id da bagagem do passageiro . . . . .	35
5.3	5 consultas com os operadores ( <i>avg</i> , <i>sum</i> , etc.) usando <i>group by</i> , <i>having</i> e <i>order by</i> . . . . .	36
5.3.1	Busca pelas empresas cuja soma da capacidade de seus ônibus seja maior que 49 organizadas pelo id da empresa . . . . .	36

5.3.2	Busca pelo numero de lojas na rodoviária, por segmento, cujo número de lojas do ramo seja maior que 2. . . . .	36
5.3.3	Busca pelas rodoviárias cujo número de lojas maior que 1 . . . . .	37
5.3.4	Mostra a soma dos salários dos funcionários das lojas das rodoviárias onde a soma dos salários é acima de 1100. . . . .	37
5.3.5	Exibir o nome do passageiro(a) que possui mais que 1 bagagem . . . . .	38
<b>6</b>	<b>CÓDIGO COMPLETO . . . . .</b>	<b>39</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>49</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>51</b>



# 1 Introdução

Vivemos em uma era digital onde a necessidade de desenvolver um sistema elaborado de banco de dados e sua interação com aplicativos externos tem se tornado cada vez maior. Além de ser uma maneira para organizar as informações e torna-las acessíveis facilmente, os banco de dados também são utilizados para as operações internas da empresa com seus clientes.

O banco de dados é um conjunto de dados estruturados, normalmente armazenados *online* e gerenciados por um sistema dedicado. Juntando o banco de dados com o sistema de gerenciamento associados à aplicativos, são chamados de sistema de banco de dados, normalmente abreviado para banco de dados.<sup>1</sup>

---

<sup>1</sup> <<https://www.oracle.com/br/database/what-is-database/>> (ORACLE, 2021)



# Parte I

## Idealização do Projeto



## 2 Elaboração do projeto

### 2.1 Objetivo

"Deseja-se criar um banco de dados de uma empresa de passagens de ônibus. É necessário identificar os passageiros através de um código identificador, CPF e nome. Além disso, o passageiro possui bagagem e estas tem um código identificador e o código do passageiro a quem pertencem. Um passageiro pode ter uma ou mais passagens que possuem as seguintes informações: código identificador da passagem, código identificador da viagem, código identificador do passageiro que a comprou e número da poltrona. Um passageiro pode fazer uma ou mais viagens. Uma viagem possui uma origem e um destino e horário de saída e de chegada. Há uma empresa responsável pelo ônibus que será utilizado na viagem que possui código identificador, nome e CNPJ. Já o ônibus possui um modelo, quantidade de passageiros permitida por viagem e motorista. O motorista possui um código, nome e número da CNH. Além disso, a rodoviária tem uma equipe de funcionários para atendimento ao cliente que possui um código de identificação, o número de funcionários na equipe e quantidade de clientes atendidos por mês e uma equipe para manutenção interna, que tem um código de identificação, o número de funcionários na equipe e quantidade de manutenções realizadas por mês. Um funcionário está em apenas uma equipe e possui código identificador, nome, CPF, salário e escala de horário de trabalho. A rodoviária possui, ainda, lojas com um código identificador, nome, CNPJ e de qual segmento é e quantidade total de funcionários."

Dada a proposta acima, é necessário desenvolver os modelos *Entity-relationship* (Entidade Relacionamento - *ER*)<sup>1</sup> e relacional e implementar em *SQL*.

Além disto, deve-se demonstrar para cada consulta:

- enunciado para a consulta;
- implementação em *SQL*;
- resultado da consulta.

### 2.2 Modelos

O modelo ER é utilizado para estudar as associações entre entidades do mundo real e correlacioná-las entre si. São utilizadas tabelas formadas por palavras-chave do banco de

---

<sup>1</sup> <<https://beginnersbook.com/2015/04/e-r-model-in-dbms/>> (BEGINNERSBOOK, 2021)

dados e os seus relacionamentos indicam as conexões entre cada uma delas.<sup>2</sup>

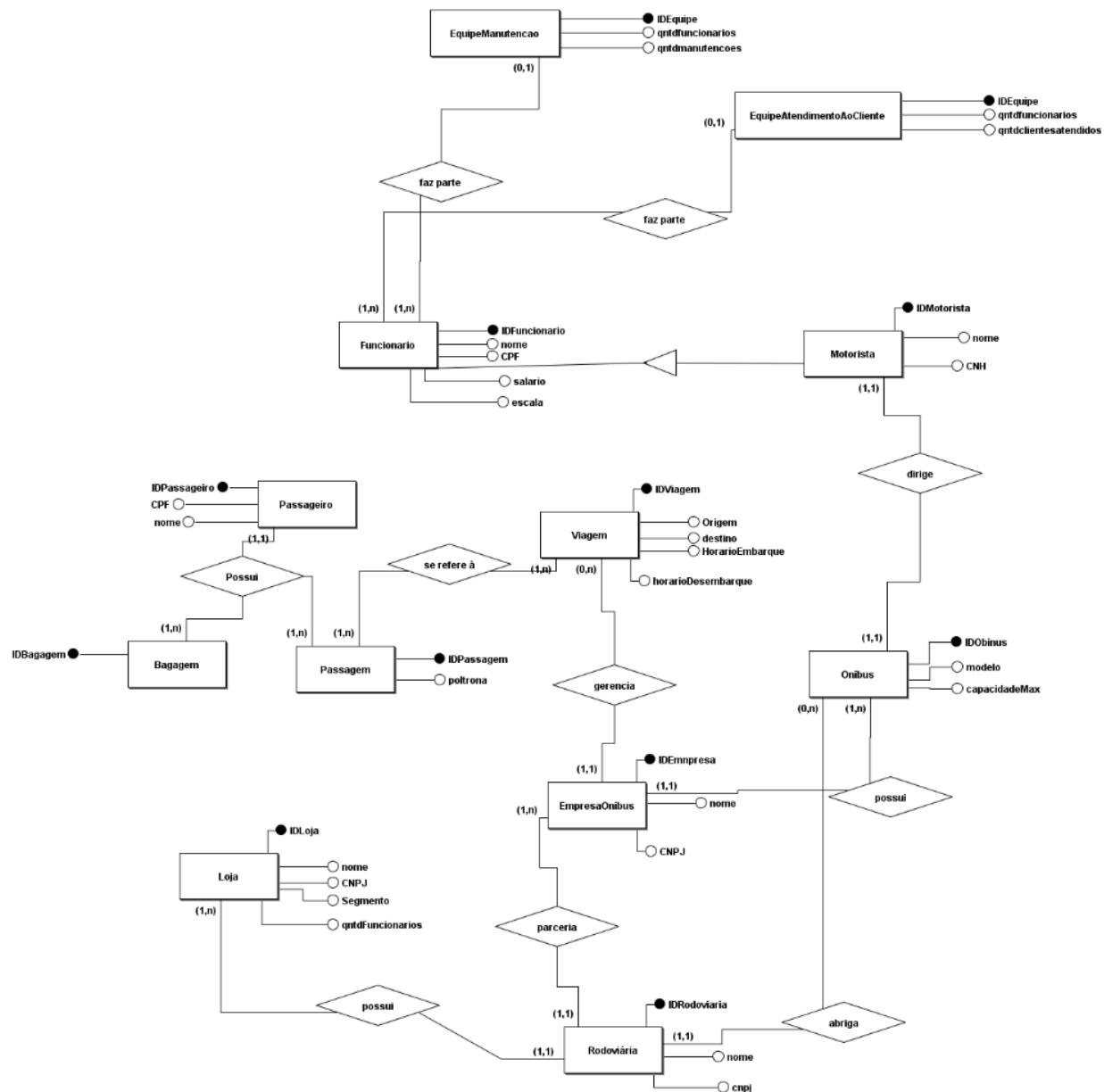


Figura 1 – Primeiro modelo relacional baseado na ideia do projeto pré-implementação.

1. Passageiro(IDPassageiro, CPF, nome)
2. Bagagem(IDBagagem, #IDPassageiro)
3. Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona)
4. Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDOnibus, #IDmotorista)

<sup>2</sup> <[https://www.tutorialspoint.com/dbms/er\\_model\\_basic\\_concepts.htm](https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm)> (TUTORIALSPPOINT, 2021)

5. EmpresaOnibus(IDEmpresa, nome, CNPJ)
6. Onibus(IDOnibus, modelo, capacidadeMax, #IDempresa)
7. Motorista(IDMotorista, nome, CNH)
8. EquipeAtendimentoAoCliente(IEquipe, qntdfuncionarios, qntdclientesatendidos)
9. EquipeManutencao(IEquipe, qntdfuncionarios, qntdmanutencoes)
10. Funcionario(IDFuncionario, #IEquipe, nome, CPF, salario, escala)
11. Loja(IDLoja, nome, CNPJ, segmento, qntdFuncinarios)
12. Rodoviária(IDRodoviaria, nome, CNPJ)

Inicialmente foram desenvolvidos o modelo da [Figura 1](#) as tabelas do [Item 12](#) antes do início da implementação em *SQL*.

Após a implementação em *SQL*, foram observadas as necessidades de alterar informações dos modelos do projeto. Logo, os modelos ficaram da forma mostradas pela [Figura 2](#) e [Item 14](#).

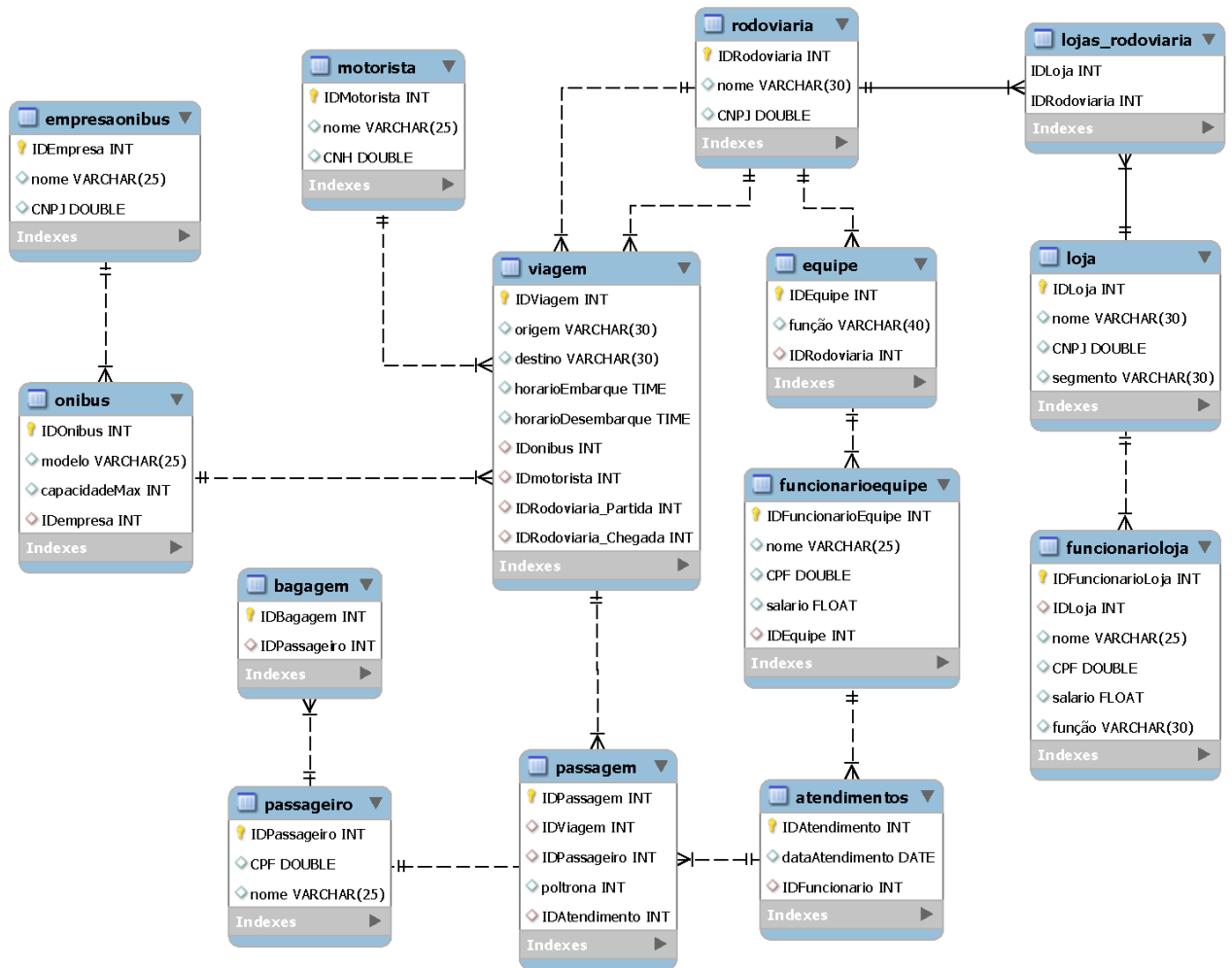


Figura 2 – Modelo relacional final do projeto utilizado para implementação.

1. Passageiro(IDPassageiro, CPF, nome)
2. Bagagem(IDBagagem, #IDPassageiro)
3. Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona, #IDAtendimento)
4. Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDonibus, #IDmotorista, #IDRodoviaria\_Partida, #IDRodoviaria\_Chegada)
5. EmpresaOnibus(IDEmpresa, nome, CNPJ)
6. Onibus(IDOnibus, modelo, capacidadeMax, #IDempresa)
7. Motorista(IDMotorista, nome, CNH)



- 
8. Equipe(IDEquipe, função, #IDRodoviaria)
  9. Loja(IDLoja, nome, CNPJ, segmento)
  10. Rodoviária(IDRodoviaria, nome, CNPJ)
  11. Lojas\_Rodoviaria(#IDLoja, #IDRodoviaria)
  12. FuncionarioLoja(IDFuncionarioLoja, #IDloja, nome, CPF, salario, função)
  13. FuncionarioEquipe(IDFuncionarioEquipe, nome, CPF, salario, #IDEquipe)
  14. Atendimentos(IDAtendimento, dataAtendimento, #IDFuncionario)



## Parte II

### Realização do projeto



### 3 Criação do Banco de Dados

Através da plataforma da *Oracle Corporation MySQL* e da linguagem *SQL*, foi realizada a criação do banco de acordo com a [Figura 2](#).

```

1 CREATE DATABASE RODOVIARIA;
2
3 /* Rodoviária(IDRodoviaria, nome, CNPJ) */
4 CREATE TABLE Rodoviaria (
5     IDRodoviaria INTEGER,
6     nome VARCHAR(30),
7     CNPJ DOUBLE,
8     PRIMARY KEY (IDRodoviaria)
9 );
10 /* Passageiro(IDPassageiro, CPF, nome) */
11 CREATE TABLE Passageiro (
12     IDPassageiro INTEGER,
13     CPF DOUBLE,
14     nome VARCHAR(25),
15     PRIMARY KEY (IDPassageiro)
16 );
17 /* Bagagem(IDBagagem, #IDPassageiro) */
18 CREATE TABLE Bagagem (
19     IDBagagem INTEGER,
20     IDPassageiro INTEGER,
21     PRIMARY KEY (IDBagagem),
22     FOREIGN KEY (IDPassageiro)
23         REFERENCES Passageiro (IDPassageiro)
24 );
25 /* EmpresaOnibus(IDEmpresa, nome, CNPJ) */
26 CREATE TABLE EmpresaOnibus (
27     IDEmpresa INTEGER,
28     nome VARCHAR(25),
29     CNPJ DOUBLE,
30     PRIMARY KEY (IDEmpresa)
31 );
32 /* Onibus(IDOnibus, modelo, capacidadeMax, #IDempresa) */
33 CREATE TABLE Onibus (
34     IDOnibus INTEGER,
35     modelo VARCHAR(25),
36     capacidadeMax INTEGER,
37     IDempresa INTEGER,
38     PRIMARY KEY (IDOnibus),
39     FOREIGN KEY (IDempresa)
40         REFERENCES EmpresaOnibus (IDEmpresa)
41 );
42 /* Motorista(IDMotorista, nome, CNH) */
43 CREATE TABLE Motorista (
44     IDMotorista INTEGER,
45     nome VARCHAR(25),
46     CNH DOUBLE,
47     PRIMARY KEY (IDMotorista)
48 );
49 /* Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDonibus,
50     #IDmotorista, #IDRodoviaria_Partida, #IDRodoviaria_Chegada) */
51 CREATE TABLE Viagem (

```

```

52     IDViagem INTEGER,
53     origem VARCHAR(30),
54     destino VARCHAR(30),
55     horarioEmbarque TIME,
56     horarioDesembarque TIME,
57     IDonibus INTEGER,
58     IDmotorista INTEGER,
59     IDRodoviaria_Partida INTEGER,
60     IDRodoviaria_Chegada INTEGER,
61     PRIMARY KEY (IDViagem),
62     FOREIGN KEY (IDonibus)
63         REFERENCES Onibus (IDonibus),
64     FOREIGN KEY (IDmotorista)
65         REFERENCES Motorista (IDmotorista),
66     FOREIGN KEY (IDRodoviaria_Partida)
67         REFERENCES Rodoviaria (IDRodoviaria),
68     FOREIGN KEY (IDRodoviaria_Chegada)
69         REFERENCES Rodoviaria (IDRodoviaria)
70 );
71 /* Loja(IDLoja, nome, CNPJ, segmento) */
72 CREATE TABLE Loja (
73     IDLoja INTEGER,
74     nome VARCHAR(30),
75     CNPJ DOUBLE,
76     segmento VARCHAR(30),
77     PRIMARY KEY (IDLoja)
78 );
79 /* Lojas_Rodoviaria(#IDLoja, #IDRodoviaria) */
80 CREATE TABLE Lojas_Rodoviaria (
81     IDLoja INTEGER,
82     IDRodoviaria INTEGER,
83     PRIMARY KEY (IDLoja , IDRodoviaria),
84     FOREIGN KEY (IDLoja)
85         REFERENCES Loja (IDLoja),
86     FOREIGN KEY (IDRodoviaria)
87         REFERENCES Rodoviaria (IDRodoviaria)
88 );
89 /* Equipe(IEquipe, função, #IDRodoviaria) */
90 CREATE TABLE Equipe (
91     IEquipe INTEGER,
92     função VARCHAR(40),
93     IDRodoviaria INTEGER,
94     PRIMARY KEY (IEquipe),
95     FOREIGN KEY (IDRodoviaria)
96         REFERENCES Rodoviaria (IDRodoviaria)
97 );
98 /* FuncionarioLoja(IDFuncionarioLoja, #IDloja, nome, CPF, salario, função) */
99 CREATE TABLE FuncionarioLoja (
100     IDFuncionarioLoja INTEGER,
101     IDLoja INTEGER,
102     nome VARCHAR(25),
103     CPF DOUBLE,
104     salario FLOAT,
105     função VARCHAR(30),
106     PRIMARY KEY (IDFuncionarioLoja),
107     FOREIGN KEY (IDLoja)
108         REFERENCES Loja (IDLoja)
109 );
110 /* FuncionarioEquipe(IDFuncionarioEquipe, nome, CPF, salario, #IEquipe) */
111 CREATE TABLE FuncionarioEquipe (

```

```
112     IDFuncionarioEquipe INTEGER,  
113     nome VARCHAR(25),  
114     CPF DOUBLE,  
115     salario FLOAT,  
116     IDEquipe INTEGER,  
117     PRIMARY KEY (IDFuncionarioEquipe),  
118     FOREIGN KEY (IDEquipe)  
119         REFERENCES Equipe (IDEquipe)  
120 );  
121 /* Atendimentos(IDAtendimento, dataAtendimento, #IDFuncionario) */  
122 CREATE TABLE Atendimentos (  
123     IDAtendimento INTEGER,  
124     dataAtendimento DATE,  
125     IDFuncionario INTEGER,  
126     PRIMARY KEY (IDAtendimento),  
127     FOREIGN KEY (IDFuncionario)  
128         REFERENCES FuncionarioEquipe (IDFuncionarioEquipe)  
129 );  
130 /* Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona, #IDAtendimento) */  
131 CREATE TABLE Passagem (  
132     IDPassagem INTEGER,  
133     IDViagem INTEGER,  
134     IDPassageiro INTEGER,  
135     poltrona INTEGER,  
136     IDAtendimento INTEGER,  
137     PRIMARY KEY (IDPassagem),  
138     FOREIGN KEY (IDViagem)  
139         REFERENCES Viagem (IDViagem),  
140     FOREIGN KEY (IDPassageiro)  
141         REFERENCES Passageiro (IDPassageiro),  
142     FOREIGN KEY (IDAtendimento)  
143         REFERENCES Atendimentos (IDAtendimento)
```

Listing 3.1 – Código em *SQL* do Banco de Dados





## 4 Cadastro de informações

Dado que o Banco de Dados agora está criado, foi inserido dados em cada tabela com valores aleatórios.

```

1  /* Rodoviária(IDRodoviaria, nome, CNPJ) */
2  INSERT INTO Rodoviaria VALUES(1, 'RodoviariaSP', 61007184751),
3      (2, 'RodoviariaSJC', 38236015844),
4      (3, 'RodoviariaCaragua', 43775503653),
5      (4, 'RodoviariaRJ', 08764288267),
6      (5, 'RodoviariaTaubate', 34145254005),
7      (6, 'RodoviariaJau', 50758625262),
8      (7, 'RodoviariaSãoCarlos', 47414266618),
9      (8, 'RodoviariaBauru', 36837884825),
10     (9, 'RodoviariaAraraquara', 68624250480),
11     (10, 'RodoviariaSantos', 17431182445);
12 /* Passageiro(IDPassageiro, CPF, nome) */
13 INSERT INTO Passageiro VALUES(1, 45461125720, 'Maria'),
14     (2, 55621067606, 'Lilian'),
15     (3, 42245256247, 'Gustavo'),
16     (4, 76788217266, 'Marcos'),
17     (5, 11170544787, 'Magno'),
18     (6, 28227750612, 'Melissa'),
19     (7, 60730724620, 'Fernanda'),
20     (8, 25550745515, 'Tiago'),
21     (9, 28344143780, 'Flávia'),
22     (10, 87666156440, 'Pedro'),
23     (11, 67014532135, 'Aaron');
24 /* Bagagem(IDBagagem, #IDPassageiro) */
25 INSERT INTO Bagagem VALUES(1, 1),
26     (2, 1),
27     (3, 2),
28     (4, 3),
29     (5, 4),
30     (6, 5),
31     (7, 6),
32     (8, 6),
33     (9, 7),
34     (10, 8),
35     (11, 8),
36     (12, 9),
37     (13, 10),
38     (14, 11),
39     (15, 11);
40 /* EmpresaOnibus(IDEmpresa, nome, CNPJ) */
41 INSERT INTO EmpresaOnibus VALUES(1, 'EmpresaUm', 38703781326),
42     (2, 'EmpresaDois', 71815874501),
43     (3, 'EmpresaTres', 24607003586),
44     (4, 'EmpresaQuatro', 37611384754),
45     (5, 'EmpresaCinco', 34503326463),
46     (6, 'EmpresaSeis', 41777761350),
47     (7, 'EmpresaSete', 51206117853),
48     (8, 'EmpresaOito', 13502370746),
49     (9, 'EmpresaNove', 55444688560),
50     (10, 'EmpresaDez', 03463527151);
51 /* Onibus(IDOnibus, modelo, capacidadeMax, #IDEmpresa) */

```

```

52 INSERT INTO Onibus VALUES(1, 'Mercedes Benz', 50, 1),
53      (2, 'Volkswagen', 60, 10),
54      (3, 'Marcopolo', 45, 7),
55      (4, 'Iveco', 50, 6),
56      (5, 'Volvo', 47, 5),
57      (6, 'Agrale', 50, 3),
58      (7, 'Scania', 60, 2),
59      (8, 'Jimbei', 55, 4),
60      (9, 'International', 57, 8),
61      (10, 'Versatile', 50, 9);
62 /* Motorista(IDMotorista, nome, CNH) */
63 INSERT INTO Motorista VALUES(1, 'Bernardo', 08442668378),
64      (2, 'Luis', 27820286075),
65      (3, 'Guilherme', 45806147888),
66      (4, 'Nathalia', 30188617372),
67      (5, 'Vitor', 61708774725),
68      (6, 'Luísa', 33514184155),
69      (7, 'Alberto', 42501358068),
70      (8, 'Bruna', 26563335726),
71      (9, 'Renata', 58735100121),
72      (10, 'Tereza', 64360646752);
73 /* Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDonibus, #
      IDmotorista, #IDRodoviaria_Partida,#IDRodoviaria_Chegada */
74 INSERT INTO Viagem VALUES(1, 'São Paulo', 'São José dos Campos', 140000, 153000, 1, 1, 1,
      2),
75      (2, 'Rio de Janeiro', 'São Paulo', 071500, 131500, 2, 2, 4, 1),
76      (3, 'Santos', 'Bauru', 160000, 231000, 3, 3, 10, 8),
77      (4, 'Taubate', 'Araraquara', 081000, 153000, 4, 4, 5, 9),
78      (5, 'Caraguatatuba', 'São Paulo', 064000, 094000, 5, 5, 3, 1),
79      (6, 'São Carlos', 'Araraquara', 151000, 160000, 6, 6, 7, 9),
80      (7, 'Jau', 'Bauru', 094000, 105000, 7, 7, 6, 8),
81      (8, 'Araraquara', 'Santos', 072000, 140000, 8, 8, 9, 10),
82      (9, 'São José dos Campos', 'Taubate', 094500, 104500, 9, 9, 2, 5),
83      (10, 'São Carlos', 'São Paulo', 125000, 170500, 10, 10, 7, 1);
84 /* Loja(IDLoja, nome, CNPJ, segmento) */
85 INSERT INTO Loja VALUES(1, 'LanchoneteUm', 18002343233, 'Alimentação'),
86      (2, 'LanchoneteDois', 00217685101, 'Alimentação'),
87      (3, 'LanchoneteTres', 24780088551, 'Alimentação'),
88      (4, 'LanchoneteQuatro', 74581772262, 'Alimentação'),
89      (5, 'LanchoneteCinco', 00865504573, 'Alimentação'),
90      (6, 'LanchoneteSeis', 76532126784, 'Alimentação'),
91      (7, 'LanchoneteSete', 18051327482, 'Alimentação'),
92      (8, 'LanchoneteOito', 61036438156, 'Alimentação'),
93      (9, 'LanchoneteNove', 57766571514, 'Alimentação'),
94      (10, 'LanchoneteDez', 63727065471, 'Alimentação'),
95      (11, 'LanchoneteOnze', 16514014087, 'Alimentação'),
96      (12, 'LanchoneteDoze', 83465705652, 'Alimentação');
97 /* Lojas_Rodoviaria (#IDLoja,#IDRodoviaria) */
98 INSERT INTO Lojas_Rodoviaria VALUES(1, 1),
99      (2, 2),
100      (3, 3),
101      (4, 4),
102      (5, 5),
103      (6, 6),
104      (7, 7),
105      (8, 8),
106      (9, 9),
107      (10, 10),
108      (11, 7),
109      (12, 6);

```

```

110 /* Equipe(IDEquipe, função, #IDRodoviaria) */
111 INSERT INTO Equipe VALUES(1, 'Atendimento ao cliente', 1),
112      (2, 'Atendimento ao cliente', 2),
113      (3, 'Atendimento ao cliente', 3),
114      (4, 'Atendimento ao cliente', 4),
115      (5, 'Atendimento ao cliente', 5),
116      (6, 'Atendimento ao cliente', 6),
117      (7, 'Atendimento ao cliente', 7),
118      (8, 'Atendimento ao cliente', 8),
119      (9, 'Atendimento ao cliente', 9),
120      (10, 'Atendimento ao cliente', 10),
121      (11, 'Limpeza', 1),
122      (12, 'Limpeza', 2),
123      (13, 'Limpeza', 3),
124      (14, 'Limpeza', 4),
125      (15, 'Limpeza', 5),
126      (16, 'Limpeza', 6),
127      (17, 'Limpeza', 7),
128      (18, 'Limpeza', 8),
129      (19, 'Limpeza', 9),
130      (20, 'Limpeza', 10);
131 /* FuncionarioLoja (IDFuncionarioLoja, #IDloja, nome, CPF, salario, função) */
132 INSERT INTO FuncionarioLoja VALUES(1, 1, 'Olivia', 20223047248, 1000.00, 'Atendimento'),
133      (2, 2, 'Nubia', 58533683060, 1000.00, 'Atendimento'),
134      (3, 3, 'Gabriel', 51311016782, 1000.00, 'Atendimento'),
135      (4, 4, 'Lucas', 31328070422, 1000.00, 'Atendimento'),
136      (5, 5, 'Renata', 86375044540, 1000.00, 'Atendimento'),
137      (6, 6, 'Helena', 46637308730, 1000.00, 'Atendimento'),
138      (7, 7, 'Bruna', 45522771060, 1000.00, 'Atendimento'),
139      (8, 8, 'Camilla', 46476845107, 1000.00, 'Atendimento'),
140      (9, 9, 'Marcela', 44804065580, 1000.00, 'Atendimento'),
141      (10, 10, 'Marcell', 08183175218, 1000.00, 'Atendimento'),
142      (11, 11, 'Rafael', 35032770208, 1000.00, 'Atendimento'),
143      (12, 12, 'Alice', 32831162370, 1000.00, 'Atendimento'),
144      (13, 1, 'Jonathan', 32845846804, 1000.00, 'Cozinheiro(a)'),
145      (14, 2, 'Gilberto', 43252507745, 1000.00, 'Cozinheiro(a)'),
146      (15, 3, 'Julio', 63682262646, 1000.00, 'Cozinheiro(a)'),
147      (16, 4, 'Alessandra', 62752451076, 1000.00, 'Cozinheiro(a)'),
148      (17, 5, 'Renato', 80756245125, 1000.00, 'Cozinheiro(a)'),
149      (18, 6, 'Luan', 14861663345, 1000.00, 'Cozinheiro(a)'),
150      (19, 7, 'Vanessa', 43770746713, 1000.00, 'Cozinheiro(a)'),
151      (20, 8, 'Veronica', 31215527400, 1000.00, 'Cozinheiro(a)'),
152      (21, 9, 'João', 21106188677, 1000.00, 'Cozinheiro(a)');
153 /* FuncionarioEquipe (IDFuncionarioEquipe, nome, CPF, salario, #IDEquipe) */
154 INSERT INTO FuncionarioEquipe VALUES(1, 'Suellen', 10123755624, 1000.00, 1),
155      (2, 'Lucas', 87762730387, 1000.00, 2),
156      (3, 'Nathan', 46832020722, 1000.00, 3),
157      (4, 'Flávia', 00645316503, 1000.00, 4),
158      (5, 'Ellen', 26857576746, 1000.00, 5),
159      (6, 'Roberta', 81010634255, 1000.00, 6),
160      (7, 'Tiago', 72672440468, 1000.00, 7),
161      (8, 'Mateus', 67020520734, 1000.00, 8),
162      (9, 'Julio', 10235820204, 1000.00, 9),
163      (10, 'Marlene', 57205804167, 1000.00, 10),
164      (11, 'Caio', 24536843280, 1000.00, 11),
165      (12, 'Priscila', 87483286734, 1000.00, 12),
166      (13, 'Nadir', 28262863783, 1000.00, 13),
167      (14, 'Roberto', 10780063544, 1000.00, 14),
168      (15, 'Luis', 40843384153, 1000.00, 15),
169      (16, 'Renata', 37245025684, 1000.00, 16),

```

```
170      (17, 'Rebeca', 62767816401, 1000.00, 17),
171      (18, 'Mario', 41552576521, 1000.00, 18),
172      (19, 'Vagner', 62804161040, 1000.00, 19),
173      (20, 'Gabriela', 48445021814, 1000.00, 20);
174 /* Atendimentos (IDAtendimento, dataAtendimento, #IDFuncionario) */
175 INSERT INTO Atendimentos VALUES(1, '2020-11-01', 1),
176      (2, '2020-11-03', 2),
177      (3, '2020-11-04', 3),
178      (4, '2020-11-05', 4),
179      (5, '2020-10-12', 5),
180      (6, '2020-10-19', 6),
181      (7, '2020-10-20', 7),
182      (8, '2020-10-21', 8),
183      (9, '2020-10-31', 9),
184      (10, '2020-12-09', 10),
185      (11, '2020-12-15', 4),
186      (12, '2020-12-18', 4),
187      (13, '2021-01-09', 1),
188      (14, '2021-01-17', 3),
189      (15, '2021-01-21', 7),
190      (16, '2021-01-22', 8),
191      (17, '2021-02-09', 8),
192      (18, '2021-02-10', 7),
193      (19, '2021-02-17', 10);
194 /* Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona, #IDAtendimento) */
195 INSERT INTO Passagem VALUES(1, 1, 10, 5, 1),
196      (2, 2, 9, 8, 2),
197      (3, 3, 8, 9, 3),
198      (4, 4, 7, 5, 4),
199      (5, 5, 6, 1, 5),
200      (6, 6, 5, 15, 6),
201      (7, 7, 4, 7, 7),
202      (8, 8, 3, 12, 8),
203      (9, 9, 2, 4, 9),
204      (10, 10, 1, 18, 10);
```

Listing 4.1 – Inserção de dados no Banco de Dados

## Parte III

### Resultados



## 5 Consultas

Como pré-requisito exigido no projeto, as consultas devem ser formuladas e implementadas. Portanto, foram realizadas a criação de algumas consultas.

Temos:

- 5 consultas com os operadores básicos de *SQL*, as quais devem conter junção com, no mínimo, duas tabelas.
- 2 consultas com *LEFT JOIN*
- 5 consultas com os operadores (*avg*, *sum*, etc.) usando *group by*, *having* e *order by*

### 5.1 5 consultas com os operadores básicos de *SQL*

Aqui temos os códigos e as consultas relacionadas às 5 consultas com os operadores básicos de *SQL*.

#### 5.1.1 Exibir o nome do funcionário, o número da equipe que ele pertence e qual a sua função

```
1  /* Exibir o nome do funcionário, o número da equipe que ele pertence e qual a sua função
   */
2  SELECT
3      nome, FuncionarioEquipe.IDEquipe, função
4  FROM
5      FuncionarioEquipe,
6      Equipe
7  WHERE
8      FuncionarioEquipe.IDEquipe = Equipe.IDEquipe;
```

Listing 5.1 – *Query* em *SQL*

Realizando a consulta [Listing 5.1](#) obtemos como resposta a [Tabela 1](#).

Tabela 1 – Resultado da consulta [Listing 5.1](#)

nome	IDequipe	função
Suellen	1	Atendimento ao cliente
Lucas	2	Atendimento ao cliente
Nathan	3	Atendimento ao cliente
Flávia	4	Atendimento ao cliente
Ellen	5	Atendimento ao cliente
Roberta	6	Atendimento ao cliente
Tiago	7	Atendimento ao cliente
Mateus	8	Atendimento ao cliente
Julio	9	Atendimento ao cliente
Marlene	10	Atendimento ao cliente
Caio	11	Limpeza
Priscila	12	Limpeza
Nadir	13	Limpeza
Roberto	14	Limpeza
Luis	15	Limpeza
Renata	16	Limpeza
Rebeca	17	Limpeza
Mario	18	Limpeza
Vagner	19	Limpeza
Gabriela	20	Limpeza

Resultado com as informações cadastradas no Banco de Dados.



### 5.1.2 Exibir o *ID* das passagens que foram compradas em 2020 na rodoviaria-SãoCarlos

```
1  /* Exibir o ID das passagens que foram compradas em 2020 na rodoviariaSãoCarlos */
2  SELECT
3      IDPassagem
4  FROM
5      Passagem ,
6      Viagem ,
7      Atendimentos
8  WHERE
9      Passagem.IDViagem = Viagem.IDViagem
10     AND Viagem.IDRodoviaria_Partida = 7
11     AND Atendimentos.IDAtendimento = Passagem.IDAtendimento
12     AND YEAR(Atendimentos.dataAtendimento) = '2020';
```

Listing 5.2 – *Query* em *SQL*

Tabela 2 – Resultado da consulta [Listing 5.2](#)

IDPassagem
6
10

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta [Listing 5.2](#) obtemos como resposta a [Tabela 2](#).

### 5.1.3 Exibir o nome do funcionário, a função dele, a loja que ele trabalha e o segmento

```

1  /* Exibir o nome do funcionário, a função dele, a loja que ele trabalha e o segmento*/
2  SELECT
3      FuncionarioLoja.nome,
4      FuncionarioLoja.função,
5      Loja.nome,
6      Loja.segmento
7  FROM
8      FuncionarioLoja,
9      Loja
10 WHERE
11     FuncionarioLoja.IDloja = Loja.IDLoja;

```

Listing 5.3 – Query em SQL

Tabela 3 – Resultado da consulta Listing 5.3

Nome	Função	Nome	Segmento
Olivia	Atendimento	LanchoneteUm	Alimentação
Jonathan	Cozinheiro(a)	LanchoneteUm	Alimentação
Nubia	Atendimento	LanchoneteDois	Alimentação
Gilberto	Cozinheiro(a)	LanchoneteDois	Alimentação
Gabriel	Atendimento	LanchoneteTres	Alimentação
Julio	Cozinheiro(a)	LanchoneteTres	Alimentação
Lucas	Atendimento	LanchoneteQuatro	Alimentação
Alessandra	Cozinheiro(a)	LanchoneteQuatro	Alimentação
Renata	Atendimento	LanchoneteCinco	Alimentação
Renato	Cozinheiro(a)	LanchoneteCinco	Alimentação
Helena	Atendimento	LanchoneteSeis	Alimentação
Luan	Cozinheiro(a)	LanchoneteSeis	Alimentação
Bruna	Atendimento	LanchoneteSete	Alimentação
Vanessa	Cozinheiro(a)	LanchoneteSete	Alimentação
Camilla	Atendimento	LanchoneteOito	Alimentação
Veronica	Cozinheiro(a)	LanchoneteOito	Alimentação
Marcela	Atendimento	LanchoneteNove	Alimentação
João	Cozinheiro(a)	LanchoneteNove	Alimentação
Marcell	Atendimento	LanchoneteDez	Alimentação
Rafael	Atendimento	LanchoneteOnze	Alimentação
Alice	Atendimento	LanchoneteDoze	Alimentação

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.3 obtemos como resposta a Tabela 3.

#### 5.1.4 Exibir *ID* da passagem, nome do passageiro que possui essa passagem, origem e o destino da viagem e o nome do atendente que vendeu a passagem

```
1  /* Exibir ID da passagem, nome do passageiro que possui essa passagem, origem e o destino
   da viagem e o nome do atendente que vendeu a passagem
2  Serão consideradas apenas as passagens com ID entre 3 e 7*/
3  SELECT
4      Passagem.IDPassagem,
5      Passageiro.nome,
6      Viagem.origem,
7      Viagem.destino,
8      FuncionarioEquipe.nome
9  FROM
10     Passagem,
11     Viagem,
12     Passageiro,
13     Atendimentos,
14     FuncionarioEquipe
15 WHERE
16     Passagem.IDViagem = Viagem.IDViagem
17     AND Passagem.IDPassageiro = Passageiro.IDPassageiro
18     AND Passagem.IDAtendimento = Atendimentos.IDAtendimento
19     AND Atendimentos.IDFuncionario = FuncionarioEquipe.IDFuncionarioEquipe
20     AND Passagem.IDPassagem BETWEEN '3' AND '7';
```

Listing 5.4 – Query em SQL

Tabela 4 – Resultado da consulta Listing 5.4

IDPassagem	nome	origem	destino	nome
3	Tiago	Santos	Bauru	Nathan
4	Fernanda	Taubate	Araraquara	Flávia
5	Melissa	Caraguatatuba	São Paulo	Ellen
6	Magno	São Carlos	Araraquara	Roberta
7	Marcos	Jau	Bauru	Tiago

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.4 obtemos como resposta a Tabela 4.

### 5.1.5 Exibir a origem e destino da viagem, nome do motorista, *ID* do ônibus e o modelo do ônibus que realizou a viagem

```

1  /* Exibir a origem e destino* da viagem, nome do motorista, ID do ônibus e o modelo do ô
   nibus que realizou a viagem
2  *Serão considerados apenas as viagens que tem como destino a cidade de São Paulo */
3  SELECT
4      Viagem.Origem,
5      Viagem.Destino,
6      Motorista.nome,
7      Onibus.IDOnibus,
8      Onibus.modelo
9  FROM
10     Viagem,
11     Motorista,
12     Onibus
13  WHERE
14     Viagem.destino LIKE 'São Paulo'
15     AND Viagem.IDMotorista = Motorista.IDMotorista
16     AND Viagem.IDOnibus = Onibus.IDOnibus;

```

Listing 5.5 – *Query* em *SQL*

Tabela 5 – Resultado da consulta Listing 5.5

Origem	Destino	nome	IDOnibus	modelo
Rio de Janeiro	São Paulo	Luis	2	Volkswagen
Caraguatatuba	São Paulo	Vitor	5	Volvo
São Carlos	São Paulo	Tereza	10	Versatile

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.5 obtemos como resposta a Tabela 5.

## 5.2 2 consultas com *LEFT JOIN*

Aqui temos os códigos e as consultas relacionadas às 2 consultas com *LEFT JOIN*.

### 5.2.1 Exibir o id da passagem do passageiro

```

1  /* Exibir o id da passagem do passageiro */
2  SELECT
3      nome, IDViagem
4  FROM
5      passageiro
6      LEFT JOIN
7      passagem ON passageiro.IDPassageiro = passagem.IDPassageiro
8  ORDER BY passageiro.nome;

```

Listing 5.6 – *Query* em *SQL*

Realizando a consulta Listing 5.6 obtemos como resposta a Tabela 6.

Tabela 6 – Resultado da consulta [Listing 5.6](#)

nome	IDViagem
Aaron	NULL
Fernanda	4
Flávia	2
Gustavo	8
Lilian	9
Magno	6
Marcos	7
Maria	10
Melissa	5
Pedro	1
Tiago	3

Resultado com as informações cadastradas no Banco de Dados.

### 5.2.2 Exibir o id da bagagem do passageiro

```

1  /* Exibir o id da bagagem do passageiro */
2  SELECT
3      nome, IDBagagem
4  FROM
5      passageiro
6      LEFT JOIN
7      bagagem ON passageiro.IDPassageiro = bagagem.IDPassageiro
8  ORDER BY passageiro.nome;
```

Listing 5.7 – Query em SQL

Tabela 7 – Resultado da consulta [Listing 5.7](#)

nome	IDBagagem
Aaron	14
Aaron	15
Fernanda	9
Flávia	12
Gustavo	4
Lilian	3
Magno	6
Marcos	5
Maria	1
Maria	2
Melissa	7
Melissa	8
Pedro	13
Tiago	10
Tiago	11

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta [Listing 5.7](#) obtemos como resposta a [Tabela 7](#).

### 5.3 5 consultas com os operadores (*avg*, *sum*, etc.) usando *group by*, *having* e *order by*

5 consultas com os operadores (*avg*, *sum*, etc.) usando *group by*, *having* e *order by*

#### 5.3.1 Busca pelas empresas cuja soma da capacidade de seus ônibus seja maior que 49 organizadas pelo id da empresa

```

1  /* Busca pelas empresas cuja soma da capacidade de seus ônibus seja maior que 49
2  organizadas pelo id da empresa. */
3  SELECT
4      empresaonibus.nome, SUM(capacidadeMax)
5  FROM
6      onibus,
7      empresaonibus
8  WHERE
9      empresaonibus.IDEmpresa = onibus.IDEmpresa
10 GROUP BY empresaonibus.IDEmpresa
11 HAVING SUM(capacidadeMax) > 49
12 ORDER BY empresaonibus.IDEmpresa;
```

Listing 5.8 – Query em SQL

Tabela 8 – Resultado da consulta [Listing 5.8](#)

nome	SUM(capacidadeMax)
EmpresaUm	50
EmpresaDois	60
EmpresaTres	50
EmpresaQuatro	55
EmpresaSeis	50
EmpresaOito	57
EmpresaNove	50
EmpresaDez	60

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta [Listing 5.8](#) obtemos como resposta a [Tabela 8](#).

#### 5.3.2 Busca pelo numero de lojas na rodoviária, por segmento, cujo número de lojas do ramo seja maior que 2.

```

1  /* Busca pelo numero de lojas na rodoviária, por segmento, cujo número de lojas do ramo
2  seja maior que 2. */
3  SELECT
4      COUNT(segmento)
5  FROM
```

```

6      loja
7  GROUP BY segmento
8  HAVING COUNT(segmento) > 2
9  ORDER BY segmento;

```

Listing 5.9 – Query em SQL

Tabela 9 – Resultado da consulta Listing 5.9

<u>count(segmento)</u>
12

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.9 obtemos como resposta a Tabela 9.

### 5.3.3 Busca pelas rodoviárias cujo número de lojas maior que 1

```

1  /* Busca pelas rodoviárias cujo número de lojas maior que 1 */
2  SELECT
3      rodoviaria.IDRodoviaria, rodoviaria.nome, COUNT(IDLoja)
4  FROM
5      lojas_rodoviaria,
6      rodoviaria
7  WHERE
8      lojas_rodoviaria.IDRodoviaria = rodoviaria.IDRodoviaria
9  GROUP BY rodoviaria.IDRodoviaria
10 HAVING COUNT(IDLoja) > 1
11 ORDER BY rodoviaria.IDRodoviaria;

```

Listing 5.10 – Query em SQL

Tabela 10 – Resultado da consulta Listing 5.10

IDRodoviaria	nome	count(IDLoja)
6	RodoviariaJau	2
7	RodoviariaSãoCarlos	2

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.10 obtemos como resposta a Tabela 10.

### 5.3.4 Mostra a soma dos salários dos funcionários das lojas das rodoviárias onde a soma dos salários é acima de 1100.

```

1  /* Mostra a soma dos salários dos funcionários das lojas das rodoviárias onde
2  a soma dos salários é acima de 1100. */
3  SELECT
4      rodoviaria.IDRodoviaria, rodoviaria.nome, SUM(salario)
5  FROM
6      funcionario_loja,
7      lojas_rodoviaria,
8      rodoviaria

```

```

9 WHERE
10     funcionarioloja.IDLoja = lojas_rodoviaria.IDLoja
11     AND lojas_rodoviaria.IDRodoviaria = rodoviaria.IDRodoviaria
12 GROUP BY rodoviaria.IDRodoviaria
13 HAVING SUM(salario) > 1100
14 ORDER BY rodoviaria.IDRodoviaria;

```

Listing 5.11 – Query em SQL

Tabela 11 – Resultado da consulta Listing 5.11

IDRodoviaria	nome	sum(salario)
1	RodoviariaSP	2000
2	RodoviariaSJC	2000
3	RodoviariaCaragua	2000
4	RodoviariaRJ	2000
5	RodoviariaTaubate	2000
6	RodoviariaJau	3000
7	RodoviariaSãoCarlos	3000
8	RodoviariaBauru	2000
9	RodoviariaAraraquara	2000

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.11 obtemos como resposta a Tabela 11.

### 5.3.5 Exibir o nome do passageiro(a) que possui mais que 1 bagagem

```

1 /* Exibir o nome do passageiro(a) que possui mais que 1 bagagem */
2 SELECT
3     nome
4 FROM
5     Passageiro,
6     Bagagem
7 WHERE
8     Bagagem.IDPassageiro = Passageiro.IDPassageiro
9 GROUP BY nome
10 HAVING COUNT(Bagagem.IDBagagem) > 1
11 ORDER BY nome;

```

Listing 5.12 – Query em SQL

Tabela 12 – Resultado da consulta Listing 5.12

nome
Aaron
Maria
Melissa
Tiago

Resultado com as informações cadastradas no Banco de Dados.

Realizando a consulta Listing 5.12 obtemos como resposta a Tabela 12.



## 6 Código completo

Dado todas os códigos fragmentados apresentados nos capítulos anteriores, temos o seguinte código completo abaixo.

```

1 CREATE DATABASE RODOVIARIA;
2
3 /* Rodoviária(IDRodoviaria, nome, CNPJ) */
4 CREATE TABLE Rodoviaria (
5     IDRodoviaria INTEGER,
6     nome VARCHAR(30),
7     CNPJ DOUBLE,
8     PRIMARY KEY (IDRodoviaria)
9 );
10 /* Passageiro(IDPassageiro, CPF, nome) */
11 CREATE TABLE Passageiro (
12     IDPassageiro INTEGER,
13     CPF DOUBLE,
14     nome VARCHAR(25),
15     PRIMARY KEY (IDPassageiro)
16 );
17 /* Bagagem(IDBagagem, #IDPassageiro) */
18 CREATE TABLE Bagagem (
19     IDBagagem INTEGER,
20     IDPassageiro INTEGER,
21     PRIMARY KEY (IDBagagem),
22     FOREIGN KEY (IDPassageiro)
23         REFERENCES Passageiro (IDPassageiro)
24 );
25 /* EmpresaOnibus(IDEmpresa, nome, CNPJ) */
26 CREATE TABLE EmpresaOnibus (
27     IDEmpresa INTEGER,
28     nome VARCHAR(25),
29     CNPJ DOUBLE,
30     PRIMARY KEY (IDEmpresa)
31 );
32 /* Onibus(IDOnibus, modelo, capacidadeMax, #IDEmpresa) */
33 CREATE TABLE Onibus (
34     IDOnibus INTEGER,
35     modelo VARCHAR(25),
36     capacidadeMax INTEGER,
37     IDEmpresa INTEGER,
38     PRIMARY KEY (IDOnibus),
39     FOREIGN KEY (IDEmpresa)
40         REFERENCES EmpresaOnibus (IDEmpresa)
41 );
42 /* Motorista(IDMotorista, nome, CNH) */
43 CREATE TABLE Motorista (
44     IDMotorista INTEGER,
45     nome VARCHAR(25),
46     CNH DOUBLE,
47     PRIMARY KEY (IDMotorista)
48 );
49 /* Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDonibus,
50     #IDmotorista, #IDRodoviaria_Partida, #IDRodoviaria_Chegada) */
51 CREATE TABLE Viagem (

```

```

52     IDViagem INTEGER,
53     origem VARCHAR(30),
54     destino VARCHAR(30),
55     horarioEmbarque TIME,
56     horarioDesembarque TIME,
57     IDonibus INTEGER,
58     IDmotorista INTEGER,
59     IDRodoviaria_Partida INTEGER,
60     IDRodoviaria_Chegada INTEGER,
61     PRIMARY KEY (IDViagem),
62     FOREIGN KEY (IDonibus)
63         REFERENCES Onibus (IDonibus),
64     FOREIGN KEY (IDmotorista)
65         REFERENCES Motorista (IDmotorista),
66     FOREIGN KEY (IDRodoviaria_Partida)
67         REFERENCES Rodoviaria (IDRodoviaria),
68     FOREIGN KEY (IDRodoviaria_Chegada)
69         REFERENCES Rodoviaria (IDRodoviaria)
70 );
71 /* Loja(IDLoja, nome, CNPJ, segmento) */
72 CREATE TABLE Loja (
73     IDLoja INTEGER,
74     nome VARCHAR(30),
75     CNPJ DOUBLE,
76     segmento VARCHAR(30),
77     PRIMARY KEY (IDLoja)
78 );
79 /* Lojas_Rodoviaria(#IDLoja, #IDRodoviaria) */
80 CREATE TABLE Lojas_Rodoviaria (
81     IDLoja INTEGER,
82     IDRodoviaria INTEGER,
83     PRIMARY KEY (IDLoja , IDRodoviaria),
84     FOREIGN KEY (IDLoja)
85         REFERENCES Loja (IDLoja),
86     FOREIGN KEY (IDRodoviaria)
87         REFERENCES Rodoviaria (IDRodoviaria)
88 );
89 /* Equipe(IEquipe, função, #IDRodoviaria) */
90 CREATE TABLE Equipe (
91     IEquipe INTEGER,
92     função VARCHAR(40),
93     IDRodoviaria INTEGER,
94     PRIMARY KEY (IEquipe),
95     FOREIGN KEY (IDRodoviaria)
96         REFERENCES Rodoviaria (IDRodoviaria)
97 );
98 /* FuncionarioLoja(IDFuncionarioLoja, #IDloja, nome, CPF, salario, função) */
99 CREATE TABLE FuncionarioLoja (
100     IDFuncionarioLoja INTEGER,
101     IDLoja INTEGER,
102     nome VARCHAR(25),
103     CPF DOUBLE,
104     salario FLOAT,
105     função VARCHAR(30),
106     PRIMARY KEY (IDFuncionarioLoja),
107     FOREIGN KEY (IDLoja)
108         REFERENCES Loja (IDLoja)
109 );
110 /* FuncionarioEquipe(IDFuncionarioEquipe, nome, CPF, salario, #IEquipe) */
111 CREATE TABLE FuncionarioEquipe (

```

```

112     IDFuncionarioEquipe INTEGER,
113     nome VARCHAR(25),
114     CPF DOUBLE,
115     salario FLOAT,
116     IDEquipe INTEGER,
117     PRIMARY KEY (IDFuncionarioEquipe),
118     FOREIGN KEY (IDEquipe)
119         REFERENCES Equipe (IDEquipe)
120 );
121 /* Atendimentos(IDAtendimento, dataAtendimento, #IDFuncionario) */
122 CREATE TABLE Atendimentos (
123     IDAtendimento INTEGER,
124     dataAtendimento DATE,
125     IDFuncionario INTEGER,
126     PRIMARY KEY (IDAtendimento),
127     FOREIGN KEY (IDFuncionario)
128         REFERENCES FuncionarioEquipe (IDFuncionarioEquipe)
129 );
130 /* Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona, #IDAtendimento) */
131 CREATE TABLE Passagem (
132     IDPassagem INTEGER,
133     IDViagem INTEGER,
134     IDPassageiro INTEGER,
135     poltrona INTEGER,
136     IDAtendimento INTEGER,
137     PRIMARY KEY (IDPassagem),
138     FOREIGN KEY (IDViagem)
139         REFERENCES Viagem (IDViagem),
140     FOREIGN KEY (IDPassageiro)
141         REFERENCES Passageiro (IDPassageiro),
142     FOREIGN KEY (IDAtendimento)
143         REFERENCES Atendimentos (IDAtendimento)
144 );
145 /*
146     *****
147     */ /* Cadastro */
148 /* Rodoviária(IDRodoviaria, nome, CNPJ) */
149 INSERT INTO Rodoviaria VALUES(1, 'RodoviariaSP', 61007184751),
150     (2, 'RodoviariaSJC', 38236015844),
151     (3, 'RodoviariaCaragua', 43775503653),
152     (4, 'RodoviariaRJ', 08764288267),
153     (5, 'RodoviariaTaubate', 34145254005),
154     (6, 'RodoviariaJau', 50758625262),
155     (7, 'RodoviariaSãoCarlos', 47414266618),
156     (8, 'RodoviariaBauru', 36837884825),
157     (9, 'RodoviariaAraraquara', 68624250480),
158     (10, 'RodoviariaSantos', 17431182445);
159 /* Passageiro(IDPassageiro, CPF, nome) */
160 INSERT INTO Passageiro VALUES(1, 45461125720, 'Maria'),
161     (2, 55621067606, 'Lilian'),
162     (3, 42245256247, 'Gustavo'),
163     (4, 76788217266, 'Marcos'),
164     (5, 11170544787, 'Magno'),
165     (6, 28227750612, 'Melissa'),
166     (7, 60730724620, 'Fernanda'),
167     (8, 25550745515, 'Tiago'),
168     (9, 28344143780, 'Flávia'),
169     (10, 87666156440, 'Pedro'),
170     (11, 67014532135, 'Aaron');
171 /* Bagagem(IDBagagem, #IDPassageiro) */

```

```

170 INSERT INTO Bagagem VALUES(1, 1),
171      (2, 1),
172      (3, 2),
173      (4, 3),
174      (5, 4),
175      (6, 5),
176      (7, 6),
177      (8, 6),
178      (9, 7),
179      (10, 8),
180      (11, 8),
181      (12, 9),
182      (13, 10),
183      (14, 11),
184      (15, 11);
185 /* EmpresaOnibus(IDEmpresa, nome, CNPJ) */
186 INSERT INTO EmpresaOnibus VALUES(1, 'EmpresaUm', 38703781326),
187      (2, 'EmpresaDois', 71815874501),
188      (3, 'EmpresaTres', 24607003586),
189      (4, 'EmpresaQuatro', 37611384754),
190      (5, 'EmpresaCinco', 34503326463),
191      (6, 'EmpresaSeis', 41777761350),
192      (7, 'EmpresaSete', 51206117853),
193      (8, 'EmpresaOito', 13502370746),
194      (9, 'EmpresaNove', 55444688560),
195      (10, 'EmpresaDez', 03463527151);
196 /* Onibus(IDOnibus, modelo, capacidadeMax, #IDempresa) */
197 INSERT INTO Onibus VALUES(1, 'Mercedes Benz', 50, 1),
198      (2, 'Volkswagen', 60, 10),
199      (3, 'Marcopolo', 45, 7),
200      (4, 'Iveco', 50, 6),
201      (5, 'Volvo', 47, 5),
202      (6, 'Agrale', 50, 3),
203      (7, 'Scania', 60, 2),
204      (8, 'Jimbei', 55, 4),
205      (9, 'International', 57, 8),
206      (10, 'Versatile', 50, 9);
207 /* Motorista(IDMotorista, nome, CNH) */
208 INSERT INTO Motorista VALUES(1, 'Bernardo', 08442668378),
209      (2, 'Luis', 27820286075),
210      (3, 'Guilherme', 45806147888),
211      (4, 'Nathalia', 30188617372),
212      (5, 'Vitor', 61708774725),
213      (6, 'Luísa', 33514184155),
214      (7, 'Alberto', 42501358068),
215      (8, 'Bruna', 26563335726),
216      (9, 'Renata', 58735100121),
217      (10, 'Tereza', 64360646752);
218 /* Viagem(IDViagem, origem, destino, horarioEmbarque, horarioDesembarque, #IDonibus, #
      IDmotorista, #IDRodoviaria_Partida, #IDRodoviaria_Chegada */
219 INSERT INTO Viagem VALUES(1, 'São Paulo', 'São José dos Campos', 140000, 153000, 1, 1, 1,
220      2),
221      (2, 'Rio de Janeiro', 'São Paulo', 071500, 131500, 2, 2, 4, 1),
222      (3, 'Santos', 'Bauru', 160000, 231000, 3, 3, 10, 8),
223      (4, 'Taubate', 'Araraquara', 081000, 153000, 4, 4, 5, 9),
224      (5, 'Caraguatatuba', 'São Paulo', 064000, 094000, 5, 5, 3, 1),
225      (6, 'São Carlos', 'Araraquara', 151000, 160000, 6, 6, 7, 9),
226      (7, 'Jau', 'Bauru', 094000, 105000, 7, 7, 6, 8),
227      (8, 'Araraquara', 'Santos', 072000, 140000, 8, 8, 9, 10),
228      (9, 'São José dos Campos', 'Taubate', 094500, 104500, 9, 9, 2, 5),

```

```

228         (10, 'São Carlos', 'São Paulo', 125000, 170500, 10, 10, 7, 1);
229 /* Loja(IDLoja, nome, CNPJ, segmento) */
230 INSERT INTO Loja VALUES(1, 'LanchoneteUm', 18002343233, 'Alimentação'),
231         (2, 'LanchoneteDois', 00217685101, 'Alimentação'),
232         (3, 'LanchoneteTres', 24780088551, 'Alimentação'),
233         (4, 'LanchoneteQuatro', 74581772262, 'Alimentação'),
234         (5, 'LanchoneteCinco', 00865504573, 'Alimentação'),
235         (6, 'LanchoneteSeis', 76532126784, 'Alimentação'),
236         (7, 'LanchoneteSete', 18051327482, 'Alimentação'),
237         (8, 'LanchoneteOito', 61036438156, 'Alimentação'),
238         (9, 'LanchoneteNove', 57766571514, 'Alimentação'),
239         (10, 'LanchoneteDez', 63727065471, 'Alimentação'),
240         (11, 'LanchoneteOnze', 16514014087, 'Alimentação'),
241         (12, 'LanchoneteDoze', 83465705652, 'Alimentação');
242 /* Lojas_Rodoviaria (#IDLoja,#IDRodoviaria) */
243 INSERT INTO Lojas_Rodoviaria VALUES(1, 1),
244         (2, 2),
245         (3, 3),
246         (4, 4),
247         (5, 5),
248         (6, 6),
249         (7, 7),
250         (8, 8),
251         (9, 9),
252         (10, 10),
253         (11, 7),
254         (12, 6);
255 /* Equipe(IDEquipe, função, #IDRodoviaria) */
256 INSERT INTO Equipe VALUES(1, 'Atendimento ao cliente', 1),
257         (2, 'Atendimento ao cliente', 2),
258         (3, 'Atendimento ao cliente', 3),
259         (4, 'Atendimento ao cliente', 4),
260         (5, 'Atendimento ao cliente', 5),
261         (6, 'Atendimento ao cliente', 6),
262         (7, 'Atendimento ao cliente', 7),
263         (8, 'Atendimento ao cliente', 8),
264         (9, 'Atendimento ao cliente', 9),
265         (10, 'Atendimento ao cliente', 10),
266         (11, 'Limpeza', 1),
267         (12, 'Limpeza', 2),
268         (13, 'Limpeza', 3),
269         (14, 'Limpeza', 4),
270         (15, 'Limpeza', 5),
271         (16, 'Limpeza', 6),
272         (17, 'Limpeza', 7),
273         (18, 'Limpeza', 8),
274         (19, 'Limpeza', 9),
275         (20, 'Limpeza', 10);
276 /* FuncionarioLoja (IDFuncionarioLoja, #IDloja, nome, CPF, salario, função) */
277 INSERT INTO FuncionarioLoja VALUES(1, 1, 'Olivia', 20223047248, 1000.00, 'Atendimento'),
278         (2, 2, 'Nubia', 58533683060, 1000.00, 'Atendimento'),
279         (3, 3, 'Gabriel', 51311016782, 1000.00, 'Atendimento'),
280         (4, 4, 'Lucas', 31328070422, 1000.00, 'Atendimento'),
281         (5, 5, 'Renata', 86375044540, 1000.00, 'Atendimento'),
282         (6, 6, 'Helena', 46637308730, 1000.00, 'Atendimento'),
283         (7, 7, 'Bruna', 45522771060, 1000.00, 'Atendimento'),
284         (8, 8, 'Camilla', 46476845107, 1000.00, 'Atendimento'),
285         (9, 9, 'Marcela', 44804065580, 1000.00, 'Atendimento'),
286         (10, 10, 'Marcell', 08183175218, 1000.00, 'Atendimento'),
287         (11, 11, 'Rafael', 35032770208, 1000.00, 'Atendimento'),

```

```

288      (12, 12, 'Alice', 32831162370, 1000.00, 'Atendimento'),
289      (13, 1, 'Jonathan', 32845846804, 1000.00, 'Cozinheiro(a)'),
290      (14, 2, 'Gilberto', 43252507745, 1000.00, 'Cozinheiro(a)'),
291      (15, 3, 'Julio', 63682262646, 1000.00, 'Cozinheiro(a)'),
292      (16, 4, 'Alessandra', 62752451076, 1000.00, 'Cozinheiro(a)'),
293      (17, 5, 'Renato', 80756245125, 1000.00, 'Cozinheiro(a)'),
294      (18, 6, 'Luan', 14861663345, 1000.00, 'Cozinheiro(a)'),
295      (19, 7, 'Vanessa', 43770746713, 1000.00, 'Cozinheiro(a)'),
296      (20, 8, 'Veronica', 31215527400, 1000.00, 'Cozinheiro(a)'),
297      (21, 9, 'João', 21106188677, 1000.00, 'Cozinheiro(a)');
298  /* FuncionarioEquipe (IDFuncionarioEquipe, nome, CPF, salario, #IDEquipe) */
299  INSERT INTO FuncionarioEquipe VALUES(1, 'Suellen', 10123755624, 1000.00, 1),
300      (2, 'Lucas', 87762730387, 1000.00, 2),
301      (3, 'Nathan', 46832020722, 1000.00, 3),
302      (4, 'Flávia', 00645316503, 1000.00, 4),
303      (5, 'Ellen', 26857576746, 1000.00, 5),
304      (6, 'Roberta', 81010634255, 1000.00, 6),
305      (7, 'Tiago', 72672440468, 1000.00, 7),
306      (8, 'Mateus', 67020520734, 1000.00, 8),
307      (9, 'Julio', 10235820204, 1000.00, 9),
308      (10, 'Marlene', 57205804167, 1000.00, 10),
309      (11, 'Caio', 24536843280, 1000.00, 11),
310      (12, 'Priscila', 87483286734, 1000.00, 12),
311      (13, 'Nadir', 28262863783, 1000.00, 13),
312      (14, 'Roberto', 10780063544, 1000.00, 14),
313      (15, 'Luis', 40843384153, 1000.00, 15),
314      (16, 'Renata', 37245025684, 1000.00, 16),
315      (17, 'Rebeca', 62767816401, 1000.00, 17),
316      (18, 'Mario', 41552576521, 1000.00, 18),
317      (19, 'Vagner', 62804161040, 1000.00, 19),
318      (20, 'Gabriela', 48445021814, 1000.00, 20);
319  /* Atendimentos (IDAtendimento, dataAtendimento, #IDFuncionario) */
320  INSERT INTO Atendimentos VALUES(1, '2020-11-01', 1),
321      (2, '2020-11-03', 2),
322      (3, '2020-11-04', 3),
323      (4, '2020-11-05', 4),
324      (5, '2020-10-12', 5),
325      (6, '2020-10-19', 6),
326      (7, '2020-10-20', 7),
327      (8, '2020-10-21', 8),
328      (9, '2020-10-31', 9),
329      (10, '2020-12-09', 10),
330      (11, '2020-12-15', 4),
331      (12, '2020-12-18', 4),
332      (13, '2021-01-09', 1),
333      (14, '2021-01-17', 3),
334      (15, '2021-01-21', 7),
335      (16, '2021-01-22', 8),
336      (17, '2021-02-09', 8),
337      (18, '2021-02-10', 7),
338      (19, '2021-02-17', 10);
339  /* Passagem(IDPassagem, #IDViagem, #IDPassageiro, poltrona, #IDAtendimento) */
340  INSERT INTO Passagem VALUES(1, 1, 10, 5, 1),
341      (2, 2, 9, 8, 2),
342      (3, 3, 8, 9, 3),
343      (4, 4, 7, 5, 4),
344      (5, 5, 6, 1, 5),
345      (6, 6, 5, 15, 6),
346      (7, 7, 4, 7, 7),
347      (8, 8, 3, 12, 8),

```

```

348         (9, 9, 2, 4, 9),
349         (10, 10, 1, 18, 10);
350 /*****
351  */
352 Devem ser formuladas e implementadas:
353 5 consultas com os operadores básicos de SQL, as quais devem conter junção com, no mínimo
    , duas tabelas.
354 2 consultas com LEFT JOIN 5 consultas com os operadores (avg, sum, etc.) usando group by,
    having e order by */
355 /* Exibir o nome do funcionário, o número da equipe que ele pertence e qual a sua função
    */
356 SELECT
357     nome, FuncionarioEquipe.IDEquipe, função
358 FROM
359     FuncionarioEquipe,
360     Equipe
361 WHERE
362     FuncionarioEquipe.IDEquipe = Equipe.IDEquipe;
363
364 /* Exibir o ID das passagens que foram compradas em 2020 na rodoviariaSãoCarlos */
365 SELECT
366     IDPassagem
367 FROM
368     Passagem,
369     Viagem,
370     Atendimentos
371 WHERE
372     Passagem.IDViagem = Viagem.IDViagem
373     AND Viagem.IDRodoviaria_Partida = 7
374     AND Atendimentos.IDAtendimento = Passagem.IDAtendimento
375     AND YEAR(Atendimentos.dataAtendimento) = '2020';
376
377 /* Exibir o nome do funcionário, a função dele, a loja que ele trabalha e o segmento*/
378 SELECT
379     FuncionarioLoja.nome,
380     FuncionarioLoja.função,
381     Loja.nome,
382     Loja.segmento
383 FROM
384     FuncionarioLoja,
385     Loja
386 WHERE
387     FuncionarioLoja.IDloja = Loja.IDLoja;
388
389 /* Exibir ID da passagem, nome do passageiro que possui essa passagem, origem e o destino
    da viagem e o nome do atendente que vendeu a passagem
    Serão consideradas apenas as passagens com ID entre 3 e 7*/
390
391 SELECT
392     Passagem.IDPassagem,
393     Passageiro.nome,
394     Viagem.origem,
395     Viagem.destino,
396     FuncionarioEquipe.nome
397 FROM
398     Passagem,
399     Viagem,
400     Passageiro,
401     Atendimentos,
402     FuncionarioEquipe
403 WHERE

```

```

404     Passagem.IDViagem = Viagem.IDViagem
405     AND Passagem.IDPassageiro = Passageiro.IDPassageiro
406     AND Passagem.IDAtendimento = Atendimentos.IDAtendimento
407     AND Atendimentos.IDFuncionario = FuncionarioEquipe.IDFuncionarioEquipe
408     AND Passagem.IDPassagem BETWEEN '3' AND '7';
409
410 /* Exibir a origem e destino* da viagem, nome do motorista, ID do ônibus e o modelo do ô
    nibus que realizou a viagem
411 *Serão considerados apenas as viagens que tem como destino a cidade de São Paulo */
412 SELECT
413     Viagem.Origem,
414     Viagem.Destino,
415     Motorista.nome,
416     Onibus.IDOnibus,
417     Onibus.modelo
418 FROM
419     Viagem,
420     Motorista,
421     Onibus
422 WHERE
423     Viagem.destino LIKE 'São Paulo'
424     AND Viagem.IDMotorista = Motorista.IDMotorista
425     AND Viagem.IDOnibus = Onibus.IDOnibus;
426
427 /* Exibir o id da passagem do passageiro */
428 SELECT
429     nome, IDViagem
430 FROM
431     passageiro
432     LEFT JOIN
433     passagem ON passageiro.IDPassageiro = passagem.IDPassageiro
434 ORDER BY passageiro.nome;
435
436 /* Exibir o id da bagagem do passageiro */
437 SELECT
438     nome, IDBagagem
439 FROM
440     passageiro
441     LEFT JOIN
442     bagagem ON passageiro.IDPassageiro = bagagem.IDPassageiro
443 ORDER BY passageiro.nome;
444
445 /* Busca pelas empresas cuja soma da capacidade de seus ônibus seja maior que 49
446 organizadas pelo id da empresa. */
447 SELECT
448     empresaonibus.nome, SUM(capacidadeMax)
449 FROM
450     onibus,
451     empresaonibus
452 WHERE
453     empresaonibus.IDEmpresa = onibus.IDEmpresa
454 GROUP BY empresaonibus.IDEmpresa
455 HAVING SUM(capacidadeMax) > 49
456 ORDER BY empresaonibus.IDEmpresa;
457
458
459 /* Busca pelo numero de lojas na rodoviária, por segmento, cujo número de lojas do ramo
460 seja maior que 2. */
461 SELECT
462     COUNT(segmento)

```



```

463 FROM
464     loja
465 GROUP BY segmento
466 HAVING COUNT(segmento) > 2
467 ORDER BY segmento;
468
469 /* Busca pelas rodoviárias cujo número de lojas maior que 1 */
470 SELECT
471     rodoviaria.IDRodoviaria, rodoviaria.nome, COUNT(IDLoja)
472 FROM
473     lojas_rodoviaria,
474     rodoviaria
475 WHERE
476     lojas_rodoviaria.IDRodoviaria = rodoviaria.IDRodoviaria
477 GROUP BY rodoviaria.IDRodoviaria
478 HAVING COUNT(IDLoja) > 1
479 ORDER BY rodoviaria.IDRodoviaria;
480
481 /* Mostra a soma dos salários dos funcionários das lojas das rodoviárias onde
482 a soma dos salários é acima de 1100. */
483 SELECT
484     rodoviaria.IDRodoviaria, rodoviaria.nome, SUM(salario)
485 FROM
486     funcionarioloja,
487     lojas_rodoviaria,
488     rodoviaria
489 WHERE
490     funcionarioloja.IDLoja = lojas_rodoviaria.IDLoja
491     AND lojas_rodoviaria.IDRodoviaria = rodoviaria.IDRodoviaria
492 GROUP BY rodoviaria.IDRodoviaria
493 HAVING SUM(salario) > 1100
494 ORDER BY rodoviaria.IDRodoviaria;
495
496 /* Exibir o nome do passageiro(a) que possui mais que 1 bagagem */
497 SELECT
498     nome
499 FROM
500     Passageiro,
501     Bagagem
502 WHERE
503     Bagagem.IDPassageiro = Passageiro.IDPassageiro
504 GROUP BY nome
505 HAVING COUNT(Bagagem.IDBagagem) > 1
506 ORDER BY nome;

```

Listing 6.1 – Código completo em *SQL*



## 7 Conclusão

Nossa expectativa com este projeto era visualizar a relação entre os modelos e planejamento do banco de dados com sua atual implementação na linguagem escolhida *SQL* e sua importância. Ficou constatado como um modelo ER e relacional é necessário para identificar as principais configurações do banco de dados, com suas respectivas relações e entidades.

Desta forma concluímos que, através da construção do modelo *ER*, a criação do banco de dados é favorecida. Logo, foi possível sim modelar, criar e adicionar informações no banco de dados através da linguagem *SQL* além de também ter sido realizadas consultas nas tabelas, atendendo os pré-requisitos impostos no projeto.



# Referências

BEGINNERSBOOK. *Entity Relationship Diagram – ER Diagram in DBMS*. 2021. Disponível em: <<https://beginnersbook.com/2015/04/e-r-model-in-dbms/>>. Acesso em: 26 fev 2021. Citado na página 11.

ORACLE. *O Que É um Banco de Dados?* 2021. Disponível em: <<https://www.oracle.com/br/database/what-is-database/>>. Acesso em: 26 fev 2021. Citado na página 7.

TUTORIALSPPOINT. *ER Model - Basic Concepts*. 2021. Disponível em: <[https://www.tutorialspoint.com/dbms/er\\_model\\_basic\\_concepts.htm](https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm)>. Acesso em: 26 fev 2021. Citado na página 12.