

**PROJETO EM DESENVOLVIMENTO DE SOFTWARE**

**São Paulo, SP**

**2023**

**UNINOVE - Universidade Nove de Julho**

**ALUNO 1 - RILLDER KAD SOUZA DE ARAÚJO RA: 923111530**

**ALUNO 2 - MATEUS NEVES OLIVEIRA RA: 923105795**

**ALUNO 3 - JOYCE GOMES RA : 923110588**

**ALUNO 4 - RICHARD DE OLIVEIRA RA : 923106027**

**ALUNO 5 - WELLINGTON FERNANDES MARQUES RA: 923106026**

**ALUNO 6 - GUILHERME TIMPANI SANTOS RA:923100626**

**ALUNO 7 - RAIMUNDO FILHO MOTA NASCIMENTO RA: 322202389**

**ALUNO 8 – LEANDRO HENRIQUE GUIMARÃES DE SOUZA RA:922115825**

**ALUNO 9 – IVAMBERTO SOARES DE LIMA JÚNIOR RA : 922208550**

**PROJETO DE MODELAGEM SQL E PROGRAMACÃO C ALUGACAR**

|  |  |
| --- | --- |
|  | Projeto apresentado no curso de Bacharelado em Ciência da Computação referente à disciplina Projeto em Desenvolvimento de Software. |

**Orientador: Prof. Marcos Alexandruk**

**São Paulo, SP**

**2023**

**1. INTRODUÇÃO**

**Razão Social**

**ALUGACAR**

**Nome Fantasia da empresa (fictícia)**

**ALUGACAR LTDA**

**Nome do Projeto**

Projeto de modelagem SQL e programação C Alugacar.

**Objetivo**

A AlugaCar é uma empresa de aluguel de carros criada por um grupo de universitários que, em 2004 decidiu, com o objetivo de arrecadar dinheiro para pagar a faculdade alugar os próprios carros ao invés de emprestar os veículos, a ideia partiu da atual CEO da empresa que teve a ideia e motivou seus amigos a colaborarem com o crescimento do negócio.

Conforme a procura pelos serviços da empresa foi crescendo, o grupo optou por ampliar os negócios visando crescimento no público consumidor. Desta forma aumentaram a frota de carros da empresa juntamente com o capital, abrindo sua primeira loja física em 2008 onde com o aumento do mercado de automóveis, foi possível para a empresa crescer mais ainda, juntos, os agora sócios da empresa começaram a aplicar novas inovações.

A empresa pode ser encontrada em diversas capitais e cidades turísticas visando satisfazer a demanda dos serviços e buscando adequar-se às necessidades dos clientes com valores acessíveis e veículos que podem atender as mais diversas demandas.

Entre os serviços oferecidos, além do atendimento personalizado que permite entrega e retirada dos veículos em hotéis, aeroportos e rodoviárias por exemplo, a empresa oferece assistência 24h e a possibilidade de personalização do serviço, tornando os veículos mais acessíveis para as necessidades de cada cliente, procurando atender turistas, viajantes a trabalho, e pessoas que utilizam um carro casualmente e por curtos períodos de tempo, buscando oferecer um serviço de qualidade e personalizado para que estes clientes se fidelizem e voltem a fechar negócios em muitas outras ocasiões.

Além disso, com a AlugaCar, os clientes terão a garantia de um serviço de alta qualidade e personalizado para atender às suas necessidades de aluguel de carro. Com uma grande variedade de veículos disponíveis, preços competitivos e um atendimento ao cliente excepcional, a AlugaCar se torna uma referência no mercado de aluguel de carros.

**Justificativa (Por que?)**

# MISSÃO, VISÃO E VALORES DA EMPRESA

# MISSÃO

Ser líder em aluguéis de carros inovadores e tecnológicos, com a ideia de suprir os clientes em todos os estados do Brasil, sendo a melhor locadora de automóveis, visando a satisfação do usuário e melhor experiência.

## VISÃO

Ser a locadora onde há diversidade de valores e estilos, visando todo público, e que sua experiência com cada locação seja única.

## VALORES

Honestidade, Respeito, Segurança, Tecnologia, Facilidade e Agilidade.

**Público-alvo (Quem utilizará o software?)**

A infraestrutura de dados é a espinha dorsal de qualquer operação moderna, e o Banco de Dados SQL da AlugaCar é desenvolvido para oferecer alta performance e confiabilidade. Armazenando e organizando dados de forma eficiente, o SQL proporciona acesso rápido e seguro às informações cruciais para a tomada de decisões estratégicas.

A gestão da frota, histórico de clientes, transações financeiras e outras operações cruciais são gerenciadas com precisão através do nosso banco de dados, garantindo consistência e integridade dos dados em tempo real.

Benefícios para a Empresa e seus Colaboradores:

Eficiência Operacional: Reduza o tempo gasto em tarefas administrativas, permitindo que a equipe foque em atividades mais estratégicas.

Personalização do Atendimento: O sistema permite personalizar serviços de acordo com as preferências e necessidades específicas dos clientes, promovendo uma experiência única.

Controle da Frota: Acompanhe em tempo real a disponibilidade dos veículos, status da manutenção e otimize a alocação de recursos.

Tomada de Decisões Embasada: Com acesso fácil a dados precisos, a equipe pode tomar decisões mais informadas, contribuindo para o crescimento sustentável da empresa.

Em resumo, o Sistema Integrado AlugaCar e o Banco de Dados SQL e programação C são ferramentas fundamentais para impulsionar a eficiência operacional e garantir a excelência no atendimento ao cliente. Na busca contínua pela inovação, a AlugaCar reafirma seu compromisso em oferecer soluções que elevem o padrão do setor de aluguel de carros.

**2. CÓDIGO FONTE DA LINGUAGEM C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Cliente {

int id;

char nome[50];

char celular[20];

char cpf[15];

char senha[50]; // Adicionando campo senha

};

struct Funcionario {

int id;

char nome[50];

char cargo[50];

char cpf[15];

};

struct Fornecedor {

int id;

char nome[50];

char cnpj[20];

};

struct Carro {

int id;

char marca[50];

char modelo[50];

char ano[5];

};

struct ProgramData {

struct Cliente clientes[100];

int contadorClientes;

struct Funcionario funcionarios[100];

int contadorFuncionarios;

struct Fornecedor fornecedores[100];

int contadorFornecedores;

struct Carro carros[100];

int contadorCarros;

};

int realizarLogin(struct ProgramData \*data) {

char cpf[15];

char senha[50];

printf("Digite o CPF: ");

scanf(" %[^\n]", cpf);

printf("Digite a senha: ");

scanf(" %[^\n]", senha);

for (int i = 0; i < data->contadorClientes; i++) {

// Verifica se o CPF e a senha coincidem

if (strcmp(cpf, data->clientes[i].cpf) == 0 &&

strcmp(senha, data->clientes[i].senha) == 0) {

return i; // Retorna o índice do cliente autenticado

}

}

// Se não encontrar correspondência, retorna -1

return -1;

}

void cadastrarCliente(struct ProgramData \*data);

int realizarLogin(struct ProgramData \*data);

void cadastrarCliente(struct ProgramData \*data) {

printf("Cadastro de Cliente:\n");

printf("Digite o nome do cliente: ");

scanf(" %[^\n]", data->clientes[data->contadorClientes].nome);

printf("Digite o celular do cliente: ");

scanf(" %[^\n]", data->clientes[data->contadorClientes].celular);

printf("Digite o CPF do cliente: ");

scanf(" %[^\n]", data->clientes[data->contadorClientes].cpf);

// Adicionando a opção de cadastrar senha

printf("Digite a senha para o cliente: ");

scanf(" %[^\n]", data->clientes[data->contadorClientes].senha);

data->clientes[data->contadorClientes].id = data->contadorClientes + 1;

data->contadorClientes++;

printf("Cliente cadastrado com sucesso! ID: %d\n", data->clientes[data->contadorClientes - 1].id);

}

void cadastrarFuncionario(struct ProgramData \*data) {

printf("Cadastro de Funcionário:\n");

printf("Digite o nome do funcionário: ");

scanf(" %[^\n]", data->funcionarios[data->contadorFuncionarios].nome);

printf("Digite o cargo do funcionário: ");

scanf(" %[^\n]", data->funcionarios[data->contadorFuncionarios].cargo);

printf("Digite o CPF do funcionário: ");

scanf(" %[^\n]", data->funcionarios[data->contadorFuncionarios].cpf);

data->funcionarios[data->contadorFuncionarios].id = data->contadorFuncionarios + 1;

data->contadorFuncionarios++;

printf("Funcionário cadastrado com sucesso! ID: %d\n", data->funcionarios[data->contadorFuncionarios - 1].id);

}

void cadastrarFornecedor(struct ProgramData \*data) {

printf("Cadastro de Fornecedor:\n");

printf("Digite o nome do fornecedor: ");

scanf(" %[^\n]", data->fornecedores[data->contadorFornecedores].nome);

printf("Digite o CNPJ do fornecedor: ");

scanf(" %[^\n]", data->fornecedores[data->contadorFornecedores].cnpj);

data->fornecedores[data->contadorFornecedores].id = data->contadorFornecedores + 1;

data->contadorFornecedores++;

printf("Fornecedor cadastrado com sucesso! ID: %d\n", data->fornecedores[data->contadorFornecedores - 1].id);

}

void cadastrarCarro(struct ProgramData \*data) {

printf("Cadastro de Carro:\n");

printf("Digite a marca do carro: ");

scanf(" %[^\n]", data->carros[data->contadorCarros].marca);

printf("Digite o modelo do carro: ");

scanf(" %[^\n]", data->carros[data->contadorCarros].modelo);

printf("Digite o ano do carro: ");

scanf(" %[^\n]", data->carros[data->contadorCarros].ano);

data->carros[data->contadorCarros].id = data->contadorCarros + 1;

data->contadorCarros++;

printf("Carro cadastrado com sucesso! ID: %d\n", data->carros[data->contadorCarros - 1].id);

}

void editarCliente(struct ProgramData \*data) {

int id;

printf("Digite o ID do cliente que deseja editar: ");

scanf("%d", &id);

if (id <= 0 || id > data->contadorClientes) {

printf("ID inválido. Tente novamente.\n");

return;

}

printf("Editar Cliente ID: %d\n", id);

printf("Digite o novo nome do cliente: ");

scanf(" %[^\n]", data->clientes[id - 1].nome);

printf("Digite o novo celular do cliente: ");

scanf(" %[^\n]", data->clientes[id - 1].celular);

printf("Digite o novo CPF do cliente: ");

scanf(" %[^\n]", data->clientes[id - 1].cpf);

printf("Cliente editado com sucesso!\n");

}

void editarFuncionario(struct ProgramData \*data) {

int id;

printf("Digite o ID do funcionário que deseja editar: ");

scanf("%d", &id);

if (id <= 0 || id > data->contadorFuncionarios) {

printf("ID inválido. Tente novamente.\n");

return;

}

printf("Editar Funcionário ID: %d\n", id);

printf("Digite o novo nome do funcionário: ");

scanf(" %[^\n]", data->funcionarios[id - 1].nome);

printf("Digite o novo cargo do funcionário: ");

scanf(" %[^\n]", data->funcionarios[id - 1].cargo);

printf("Digite o novo CPF do funcionário: ");

scanf(" %[^\n]", data->funcionarios[id - 1].cpf);

printf("Funcionário editado com sucesso!\n");

}

void editarCarro(struct ProgramData \*data) {

int id;

printf("Digite o ID do carro que deseja editar: ");

scanf("%d", &id);

if (id <= 0 || id > data->contadorCarros) {

printf("ID inválido. Tente novamente.\n");

return;

}

printf("Editar Carro ID: %d\n", id);

printf("Digite a nova marca do carro: ");

scanf(" %[^\n]", data->carros[id - 1].marca);

printf("Digite o novo modelo do carro: ");

scanf(" %[^\n]", data->carros[id - 1].modelo);

printf("Digite o novo ano do carro: ");

scanf(" %[^\n]", data->carros[id - 1].ano);

printf("Carro editado com sucesso!\n");

}

void editarFornecedores(struct ProgramData \*data) {

int id;

printf("Digite o ID do Fornecedor que deseja editar: ");

scanf("%d", &id);

if (id <= 0 || id > data->contadorFornecedores) {

printf("ID inválido. Tente novamente.\n");

return;

}

printf("Editar Fornecedor ID: %d\n", id);

printf("Editar o nome do fornecedor: ");

scanf(" %[^\n]", data->fornecedores[id - 1].nome);

printf("Editar o CNPJ do fornecedor: ");

scanf(" %[^\n]", data->fornecedores[id - 1].cnpj);

}

void listarClientes(struct ProgramData \*data) {

if (data->contadorClientes == 0) {

printf("Nenhum cliente cadastrado.\n");

} else {

printf("Lista de Clientes:\n");

for (int i = 0; i < data->contadorClientes; i++) {

printf("ID: %d\n", data->clientes[i].id);

printf("Nome: %s\n", data->clientes[i].nome);

printf("Celular: %s\n", data->clientes[i].celular);

printf("CPF: %s\n", data->clientes[i].cpf);

printf("\n");

}

}

}

void listarFuncionarios(struct ProgramData \*data) {

if (data->contadorFuncionarios == 0) {

printf("Nenhum funcionário cadastrado.\n");

} else {

printf("Lista de Funcionários:\n");

for (int i = 0; i < data->contadorFuncionarios; i++) {

printf("ID: %d\n", data->funcionarios[i].id);

printf("Nome: %s\n", data->funcionarios[i].nome);

printf("Cargo: %s\n", data->funcionarios[i].cargo);

printf("CPF: %s\n", data->funcionarios[i].cpf);

printf("\n");

}

}

}

void listarFornecedores(struct ProgramData \*data) {

if (data->contadorFornecedores == 0) {

printf("Nenhum fornecedor cadastrado.\n");

} else {

printf("Lista de Fornecedores:\n");

for (int i = 0; i < data->contadorFornecedores; i++) {

printf("ID: %d\n", data->fornecedores[i].id);

printf("Nome: %s\n", data->fornecedores[i].nome);

printf("CNPJ: %s\n", data->fornecedores[i].cnpj);

printf("\n");

}

}

}

void listarCarros(struct ProgramData \*data) {

if (data->contadorCarros == 0) {

printf("Nenhum carro cadastrado.\n");

} else {

printf("Lista de Carros:\n");

for (int i = 0; i < data->contadorCarros; i++) {

printf("ID: %d\n", data->carros[i].id);

printf("Marca: %s\n", data->carros[i].marca);

printf("Modelo: %s\n", data->carros[i].modelo);

printf("Ano: %s\n", data->carros[i].ano);

printf("\n");

}

}

}

int main() {

int escolha;

int tipoUsuario;

int clienteLogado = -1;

struct ProgramData programData;

programData.contadorClientes = 0;

programData.contadorFuncionarios = 0;

programData.contadorFornecedores = 0;

voltarInicio:

do {

printf("Digite o tipo de acesso: \n");

printf("0 - Para acessar como usuário \n");

printf("1 - Para acessar como administrador \n");

scanf("%d", &tipoUsuario);

if (tipoUsuario != 0 && tipoUsuario != 1) {

printf("Tipo de usuário inválido. Tente novamente.\n");

continue;

}

if (tipoUsuario == 1) {

int senha;

do {

printf("Digite a senha de administrador: ");

scanf("%d", &senha);

if (senha != 1020) {

printf("Senha incorreta. Tente novamente.\n");

} else {

break;

}

} while (senha != 1020);

}

else {

printf("Menu:\n");

printf("1. Cadastrar Cliente\n");

printf("2. Realizar Login\n");

printf("3. Sair\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

cadastrarCliente(&programData);

goto voltarInicio;

case 2:

clienteLogado = realizarLogin(&programData);

if (clienteLogado != -1) {

printf("Login bem-sucedido! Bem-vindo, %s.\n",programData.clientes[clienteLogado].nome);

printf("Esta tela ainda está em desenvolvimento\n");

goto voltarInicio;

} else {

printf("Login falhou. Verifique CPF e senha.\n");

goto voltarInicio;

}

case 3:

// Adicionando um break para encerrar o loop do usuário

goto voltarInicio;

default:

printf("Opção inválida. Tente novamente.\n");

goto voltarInicio;

}

}

do {

printf("1. Cadastrar \n");

printf("2. Listar \n");

printf("3. Editar \n");

printf("4. Sair\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

printf("1. Cadastrar Carro\n");

printf("2. Cadastrar Clientes\n");

printf("3. Cadastrar Funcionarios\n");

printf("4. Cadastrar Fornecedor\n");

printf("5. Sair\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

cadastrarCarro(&programData);

break;

case 2:

cadastrarCliente(&programData);

break;

case 3:

cadastrarFuncionario(&programData);

break;

case 4:

cadastrarFornecedor(&programData);

break;

case 5:

// Adicionando um break para encerrar o loop do usuário

goto voltarInicio;

default:

// Adicione um break aqui

break;

}

break;

case 2:

printf("1. Listar Carro\n");

printf("2. Listar Clientes\n");

printf("3. Listar Funcionários\n");

printf("4. Listar Fornecedores\n");

printf("5. Sair\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

listarCarros(&programData);

break;

case 2:

listarClientes(&programData);

break;

case 3:

listarFuncionarios(&programData);

break;

case 4:

listarFornecedores(&programData);

break;

case 5:

// Adicionando um break para encerrar o loop do usuário

goto voltarInicio;

default:

printf("Opção inválida. Tente novamente.\n");

// Adicione um break aqui

break;

}

break;

case 3:

printf("1. Editar Carro\n");

printf("2. Editar Clientes\n");

printf("3. Editar Funcionarios\n");

printf("4. Editar Fornecedor\n");

printf("5. Sair\n");

printf("Escolha uma opção: ");

scanf("%d", &escolha);

switch (escolha) {

case 1:

editarCarro(&programData);

break;

case 2:

editarCliente(&programData);

break;

case 3:

editarFuncionario(&programData);

break;

case 4:

editarFornecedores(&programData);

break;

case 5:

// Adicionando um break para encerrar o loop do usuário

goto voltarInicio;

default:

// Adicione um break aqui

break;

}

break;

case 4:

// Adicionando um break para encerrar o loop do usuário

goto voltarInicio;

default:

printf("Opção inválida. Tente novamente.\n");

// Adicione um break aqui

break;

}

} while (1);

} while (1);

return 0;

}

**Observações:**

senha administrador é 1020

senha cliente os próprios clientes podem cadastrar a senha

**3. DIAGRAMA ENTIDADE RELACIONAMENTO**

**MODELO CONCEITUAL**

Figura 1 Modelo conceitual – Utilizado BR Modelo.

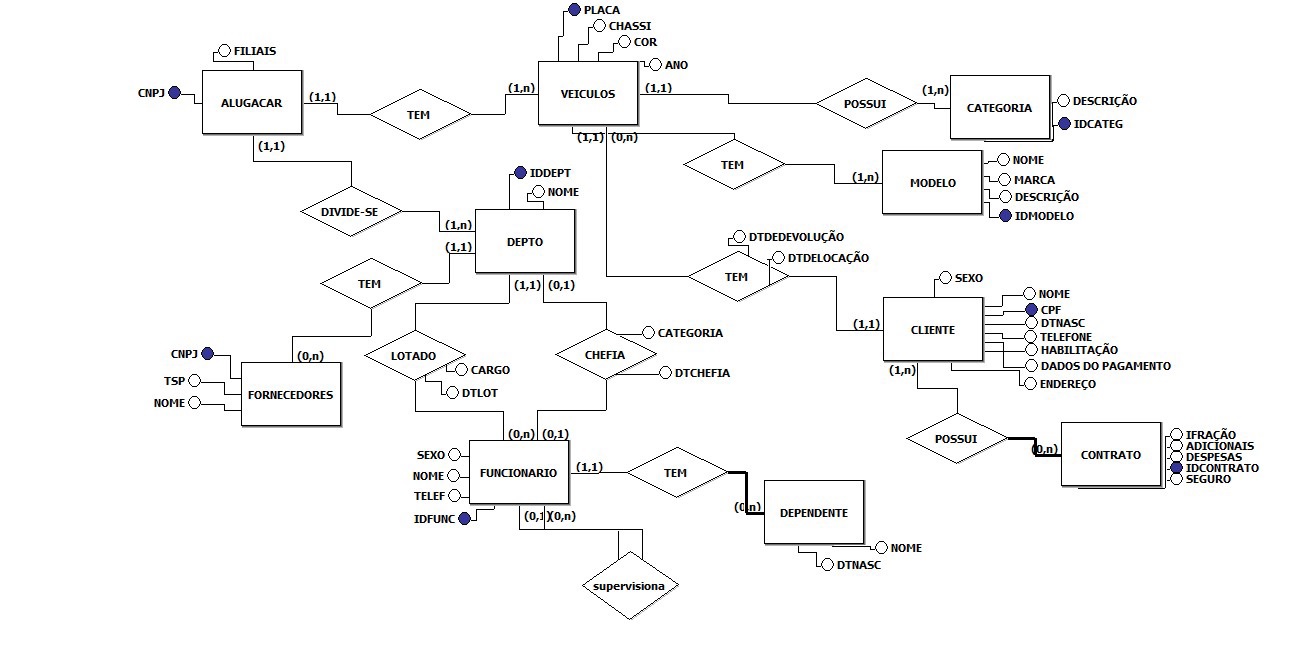
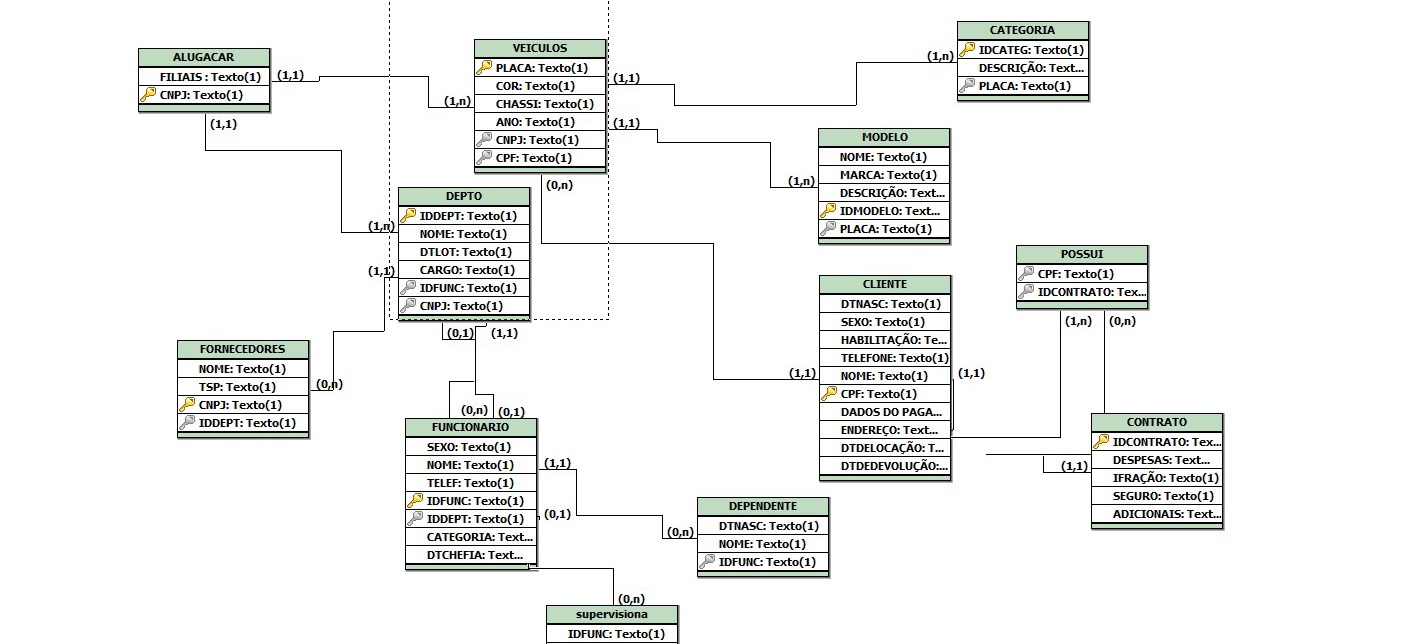
****Fonte: De autoria própria.

Figura 2 Modelo conceitual – Utilizado BR Modelo.



Fonte: De autoria própria.

**4. SCRIPT DE CRIAÇÃO DAS TABELAS E INSERÇÃO DE DADOS**

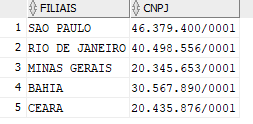
**MODELO RACIONAL**

UTILIZANDO O SQL ORACLE.

PRINTS DAS TABELAS

* ALUGACAR

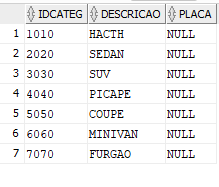
Figura 3 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* CATEGORIA

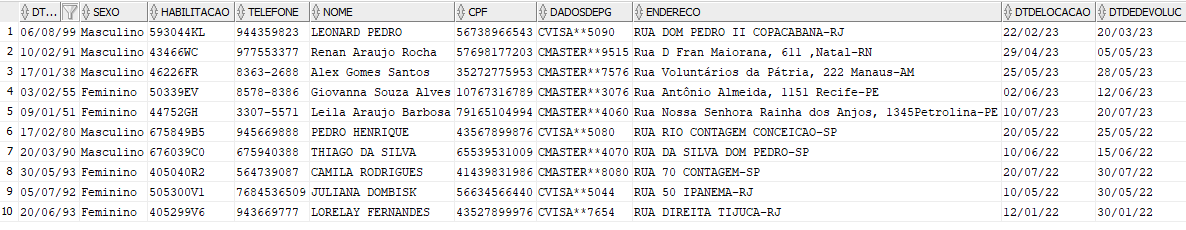
Figura 4 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* CLIENTE

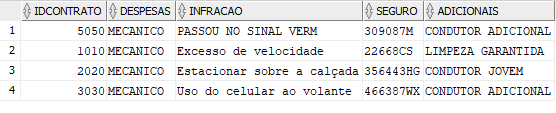
Figura 5 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* CONTRATO

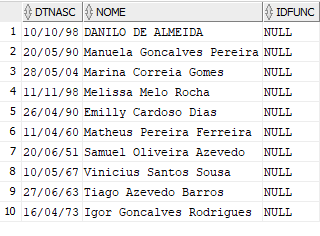
Figura 6 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* DEPENDENTE

Figura 7 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* DEPARTAMENTO

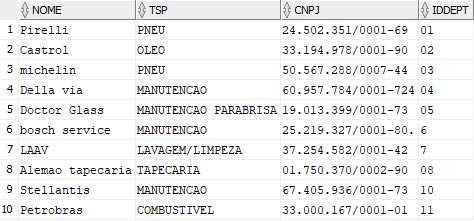
Figura 8 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* FORNECEDORES

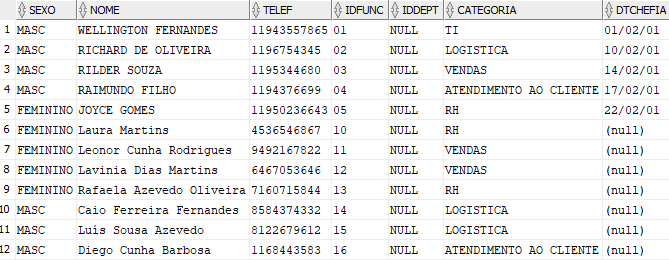
Figura 9 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

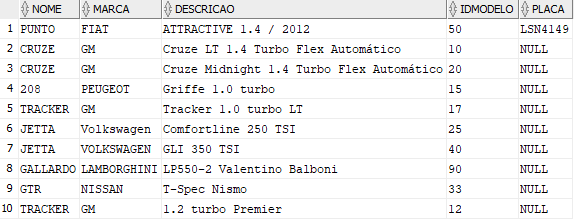
* FUNCIONARIO

Figura 10 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

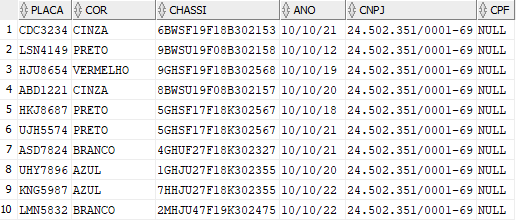
* MODELO
* Figura 11 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

* VEICULOS

Figura 12 Modelo conceitual – Utilizado SQL Developer XE213.



Fonte: De autoria própria.

**SCRIPT**

CREATE TABLE Cliente

(

DTNASC DATE,

SEXO VARCHAR2(10),

HABILITACAO VARCHAR2(20),

TELEFONE VARCHAR2(20),

NOME VARCHAR2(90),

CPF VARCHAR2(50) CONSTRAINT Clientepk\_PK Primary Key,

DADOSDEPG VARCHAR2(50),

ENDERECO VARCHAR2(200),

DTDELOCACAO DATE,

DTDEDEVOLUC DATE

)

CREATE TABLE Alugacar

(

FILIAIS VARCHAR2(50),

CNPJ VARCHAR2(50) CONSTRAINT alugacar\_CNPJ\_PK Primary Key

)

CREATE TABLE Contrato

(

IDCONTRATO NUMBER(20) CONSTRAINT IDCONTRATOPK\_PK Primary Key,

DESPESAS VARCHAR2(50),

INFRACAO VARCHAR2(80),

SEGURO VARCHAR2(80),

ADICIONAIS VARCHAR (100)

)

CREATE TABLE Veiculos

(

PLACA VARCHAR2(50) CONSTRAINT IDveiculosPK\_PK Primary Key,

COR VARCHAR2(60),

CHASSI VARCHAR2(70),

ANO DATE,

CNPJ VARCHAR2(50) CONSTRAINT Veiculos\_CNPJ\_FK References Alugacar,

CPF VARCHAR2(50),

CONSTRAINT Veiculos\_CPF\_FK FOREIGN KEY (CPF) References Cliente

)

CREATE TABLE DEPTO

(

IDDEPT VARCHAR2(50) CONSTRAINT DeptoPK\_PK Primary Key,

NOME VARCHAR2(50),

DTLOT DATE,

CARGO VARCHAR2(90),

IDFUNC VARCHAR2(50),

CNPJ VARCHAR2(80),

CONSTRAINT IDFUNC\_FK FOREIGN KEY(IDFUNC) REFERENCES ALUGACAR,

CONSTRAINT CNPJ\_FK FOREIGN KEY(CNPJ) REFERENCES ALUGACAR

)

CREATE TABLE Fornecedores

(

NOME VARCHAR2(90),

TSP VARCHAR2(90),

CNPJ VARCHAR2(80) CONSTRAINT CNPJ\_PK Primary Key,

IDDEPT VARCHAR2(80),

CONSTRAINT IDDEPT\_FK FOREIGN KEY(IDDEPT) REFERENCES DEPTO

)

CREATE TABLE Funcionario

(

SEXO VARCHAR2(10),

NOME VARCHAR2(50),

TELEF VARCHAR2(50),

IDFUNC VARCHAR2(50) CONSTRAINT Funcionario\_IDFUNC\_PK Primary Key,

IDDEPT VARCHAR2(50) CONSTRAINT Funcionario\_IDDEPT\_FK FOREIGN KEY References Depto,

CATEGORIA VARCHAR2(50),

DTCHEFIA DATE

)

CREATE TABLE Categoria

(

IDCATEG VARCHAR2(20) CONSTRAINT Categoria\_IDCATEG\_PK Primary Key,

DESCRICAO VARCHAR2(100),

PLACA VARCHAR2(60) CONSTRAINT Categoria\_PLACA\_FK FOREIGN KEY References Veiculos

)

CREATE TABLE Modelo

(

NOME VARCHAR2(50),

MARCA VARCHAR2(50),

DESCRICAO VARCHAR2(100),

IDMODELO VARCHAR2 (50) CONSTRAINT Modelo\_IDMODELO\_PK Primary Key,

PLACA VARCHAR2(60) CONSTRAINT Modelo\_PLACA\_FK FOREIGN KEY References Veiculos

)

CREATE TABLE Dependente

(

DTNASC DATE,

NOME VARCHAR2(50),

IDFUNC VARCHAR2(50) CONSTRAINT Dependente\_IDFUNC\_FK FOREIGN KEY References Funcionario

)

insert into ALUGACAR

(FILIAIS,CNPJ)

Values

('BAHIA','30.567.890/0001');

insert into ALUGACAR

(FILIAIS,CNPJ)

Values

('SAO PAULO','46.379.400/0001');

insert into ALUGACAR

(FILIAIS,CNPJ)

Values

('RIO DE JANEIRO','40.498.556/0001');

insert into ALUGACAR

(FILIAIS,CNPJ)

Values

('MINAS GERAIS','20.345.653/0001');

insert into ALUGACAR

(FILIAIS,CNPJ)

Values

('CEARA','20.435.876/0001');

insert into CONTRATO

(IDCONTRATO,DESPESAS,INFRACAO,SEGURO,ADICIONAIS)

Values

('3030','MECANICO','Uso do celular ao volante','466387WX','CONDUTOR ADICIONAL');

insert into CONTRATO

(IDCONTRATO,DESPESAS,INFRACAO,SEGURO,ADICIONAIS)

Values

('5050','MECANICO','PASSOU NO SINAL VERM','309087M','CONDUTOR ADICIONAL');

insert into CONTRATO

(IDCONTRATO,DESPESAS,INFRACAO,SEGURO,ADICIONAIS)

Values

('1010','MECANICO','Excesso de velocidade','22668CS','LIMPEZA GARANTIDA');

insert into CONTRATO

(IDCONTRATO,DESPESAS,INFRACAO,SEGURO,ADICIONAIS)

Values

('2020','MECANICO','Estacionar sobre a calçada','356443HG','CONDUTOR JOVEM');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('17/02/1980','Masculino','675849B5','945669888','PEDRO HENRIQUE','43567899876','CVISA\*\*5080','RUA RIO CONTAGEM CONCEICAO-SP','20/05/2022','25/05/2022');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('20/03/1990','Masculino','676039C0','675940388','THIAGO DA SILVA','65539531009','CMASTER\*\*4070','RUA DA SILVA DOM PEDRO-SP','10/06/2022','15/06/2022');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('30/05/1993','Feminino','405040R2','564739087','CAMILA RODRIGUES','41439831986','CMASTER\*\*8080','RUA 70 CONTAGEM-SP','20/07/2022','30/07/2022');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('05/07/1992','Feminino','505300V1','7684536509','JULIANA DOMBISK','56634566440','CVISA\*\*5044','RUA 50 IPANEMA-RJ','10/05/2022','30/05/2022');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('20/06/1993','Feminino','405299V6','943669777','LORELAY FERNANDES','43527899976','CVISA\*\*7654','RUA DIREITA TIJUCA-RJ','12/01/2022','30/01/2022');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('09/01/1951','Feminino','44752GH','3307-5571','Leila Araujo Barbosa','79165104994','CMASTER\*\*4060','Rua Nossa Senhora Rainha dos Anjos, 1345

Petrolina-PE','10/07/2023','20/07/2023');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('03/02/1955','Feminino','50339EV','8578-8386','Giovanna Souza Alves','10767316789','CMASTER\*\*3076','Rua Antônio Almeida, 1151 Recife-PE','02/06/2023','12/06/2023');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('17/01/1938','Masculino','46226FR','8363-2688','Alex Gomes Santos','35272775953','CMASTER\*\*7576','Rua Voluntários da Pátria, 222 Manaus-AM','25/05/2023','28/05/2023');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('10/02/1991','Masculino','43466WC','977553377','Renan Araujo Rocha','57698177203','CMASTER\*\*9515','Rua D Fran Maiorana, 611 ,Natal-RN','29/04/2023','05/05/2023');

insert into Cliente

(DTNASC, SEXO ,HABILITACAO, TELEFONE, NOME, CPF, DADOSDEPG, ENDERECO, DTDELOCACAO, DTDEDEVOLUC)

Values

('06/08/1999','Masculino','593044KL','944359823','LEONARD PEDRO','56738966543','CVISA\*\*5090','RUA DOM PEDRO II COPACABANA-RJ','22/02/2023','20/03/2023');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('LSN4149','PRETO','9BWSU19F08B302158','10/10/2012','24.502.351/0001-69','42567855643');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('ABD1221','CINZA','8BWSU19F08B302157','10/10/2020','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('CDC3234','CINZA','6BWSF19F18B302153','10/10/2021','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('HJU8654',VERMELHO,'9GHSF19F18B302568','10/10/2019','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('HKJ8687','PRETO','5GHSF17F18K302567','10/10/2018','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('UJH5574','PRETO','5GHSF17F18K302567','10/10/2021','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('ASD7824','BRANCO','4GHUF27F18K302327','10/10/2021','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('UHY7896',AZUL,'1GHJU27F18K302355','10/10/2020','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('KNG5987',AZUL,'7HHJU27F18K302355','10/10/2022','24.502.351/0001-69','NULL');

insert into veiculos

(PLACA,COR,CHASSI,ANO,CNPJ,CPF)

Values

('LMN5832',BRANCO,'2MHJU47F19K302475','10/10/2022','24.502.351/0001-69','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('16/04/1973','Igor Goncalves Rodrigues','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('27/06/1963','Tiago Azevedo Barros','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('10/05/1967','Vinicius Santos Sousa','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('20/06/1951','Samuel Oliveira Azevedo','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('11/04/1960','Matheus Pereira Ferreira','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('26/04/1990','Emilly Cardoso Dias','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('11/11/1998','Melissa Melo Rocha','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('28/05/2004','Marina Correia Gomes','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('20/05/1990','Manuela Goncalves Pereira','NULL');

insert into dependente

(DTNASC,NOME,IDFUNC)

Values

('10/10/1998','DANILO DE ALMEIDA','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('10','CONTAS A PAGAR','20/10/2018','RH','NULL','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('40','VENDAS VEICULO','20/10/2018','VENDAS','NULL','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('30','SUPORTE','10/10/2020','TI','NULL','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('20','PROGRAMACAO','10/10/2019','TI','NULL','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('05','ATENDIMENTO','10/10/2019','ATENDIMENTO AO CLIENTE','NULL','NULL');

insert into DEPTO

(IDDEPT,NOME,DTLOT,CARGO,IDFUNC,CNPJ)

Values

('05','ATENDIMENTO','10/10/2019','LOGISTICA','NULL','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('PUNTO','FIAT','ATTRACTIVE 1.4 / 2012','50','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('CRUZE','GM','Cruze LT 1.4 Turbo Flex Automático','10','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('CRUZE','GM','Cruze Midnight 1.4 Turbo Flex Automático','20','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('208','PEUGEOT','Griffe 1.0 turbo','15','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('TRACKER','GM','Tracker 1.0 turbo LT','17','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('TRACKER','GM','1.2 turbo Premier','12','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('JETTA','Volkswagen','Comfortline 250 TSI','25','NULL')

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('JETTA','VOLKSWAGEN','GLI 350 TSI','40','NULL')

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('GALLARDO','LAMBORGHINI','LP550-2 Valentino Balboni','90','NULL');

insert into modelo

(NOME,MARCA,DESCRICAO,IDMODELO,PLACA)

Values

('GTR','NISSAN','T-Spec Nismo','33','NULL');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('LAAV ','LAVAGEM/LIMPEZA','37.254.582/0001-42','7');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('bosch service ','MANUTENCAO','25.219.327/0001-80.','6');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Doctor Glass ','MANUTENCAO PARABRISA','19.013.399/0001-73','05');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Della via ','MANUTENCAO','

60.957.784/0001-724','04');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('michelin ','PNEU','50.567.288/0007-44','03');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Pirelli','PNEU','24.502.351/0001-69','01');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Castrol','oleo','33.194.978/0001-90','02');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Petrobras','COMBUSTIVEL','33.000.167/0001-01','11');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Stellantis','MANUTENCAO','67.405.936/0001-73','10');

insert into fornecedores

(NOME,TSP,CNPJ,IDDEPT)

Values

('Alemao tapecaria','TAPECARIA','01.750.370/0002-90','08');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('1010','HACTH','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('2020','SEDAN','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('3030','SUV','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('4040','PICAPE','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('5050','COUPE','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('6060','MINIVAN','NULL');

insert into CATEGORIA

(IDCATEG,DESCRICAO,PLACA)

Values

('7070','FURGAO','NULL');

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','WELLINGTON FERNANDES','11943557865','01','NULL','TI','01/02/2001');

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','RICHARD DE OLIVEIRA','1196754345','02','NULL','LOGISTICA','10/02/2001');

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','Diego Cunha Barbosa','1168443583','16','NULL','ATENDIMENTO AO CLIENTE',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','Luís Sousa Azevedo','8122679612','15','NULL','LOGISTICA',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','Caio Ferreira Fernandes','8584374332','14','NULL','LOGISTICA',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('FEMININO','Rafaela Azevedo Oliveira','7160715844','13','NULL','RH',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('FEMININO','Lavinia Dias Martins','6467053646','12','NULL','VENDAS',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('FEMININO','Leonor Cunha Rodrigues','9492167822','11','NULL','VENDAS',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('FEMININO','Laura Martins','4536546867','10','NULL','RH',NULL);

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('FEMININO','JOYCE GOMES','11950236643','05','NULL','RH','22/02/2001');

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','RAIMUNDO FILHO','1194376699','04','NULL','ATENDIMENTO AO CLIENTE','17/02/2001');

insert into funcionario

(SEXO,NOME,TELEF,IDFUNC,IDDEPT,CATEGORIA,DTCHEFIA)

Values

('MASC','RILDER SOUZA','1195344680','03','NULL','VENDAS','14/02/2001');

Commit;