IFSUL - Instituto Federal Sul-Rio-Grandense

Campus Passo Fundo

Trabalho de Banco de Dados

**MERCADO**

***Rafael Albuquerque de Paula e***

***Nikoli de Azeredo Alves***

Passo Fundo

2023

**MERCADO**

Resumo:

O projeto desenvolvido é sobre um mercado que vende produtos variados, desde limpeza, alimentos, carnes, etc… , estes produtos são fornecidos por diversos fornecedores, é um mercado grande que possui diversas seções nele como hortifruti, açougue, padaria, etc… para manter armazenado tantos produtos esse mercado possui um grande estoque além de precisar de vários funcionários para manter o estabelecimento em funcionamento e cada um possui um cargo formando uma hierarquia.

Descrição do Sistema:

O sistema deve manter o armazenamento de alguns dados de clientes que frequentam o estabelecimento, como nome, cpf, número, email se este possuir e a cidade onde reside, dados dos fornecedores que fornecem os produtos, como cnpj, nome fantasia, endereço, número para contato, email, dados dos produtos que estão disponíveis para compra, como nome, marca, preço de cada produto e a data de fabricação e validade, dados sobre as compras que foram realizadas, com dados como a data, valor e hora que a compra foi realizada, além de na compra haver uma ligação entre o cliente e os produtos, e registros em relação ao estoque, que é responsável por armazenar a quantidade de cada produto. O mercado também deve armazenar os dados dos funcionários, como nome, cpf, e-mail, cidade em que ele reside, a seção onde ele trabalha e o seu cargo, além de ter os dados sobre as seções que o mercado possui e os dados de cargos, com descrição e salário para cada cargo. Além de armazenar dados sobre as cidades que são utilizados nos dados dos clientes e funcionários.

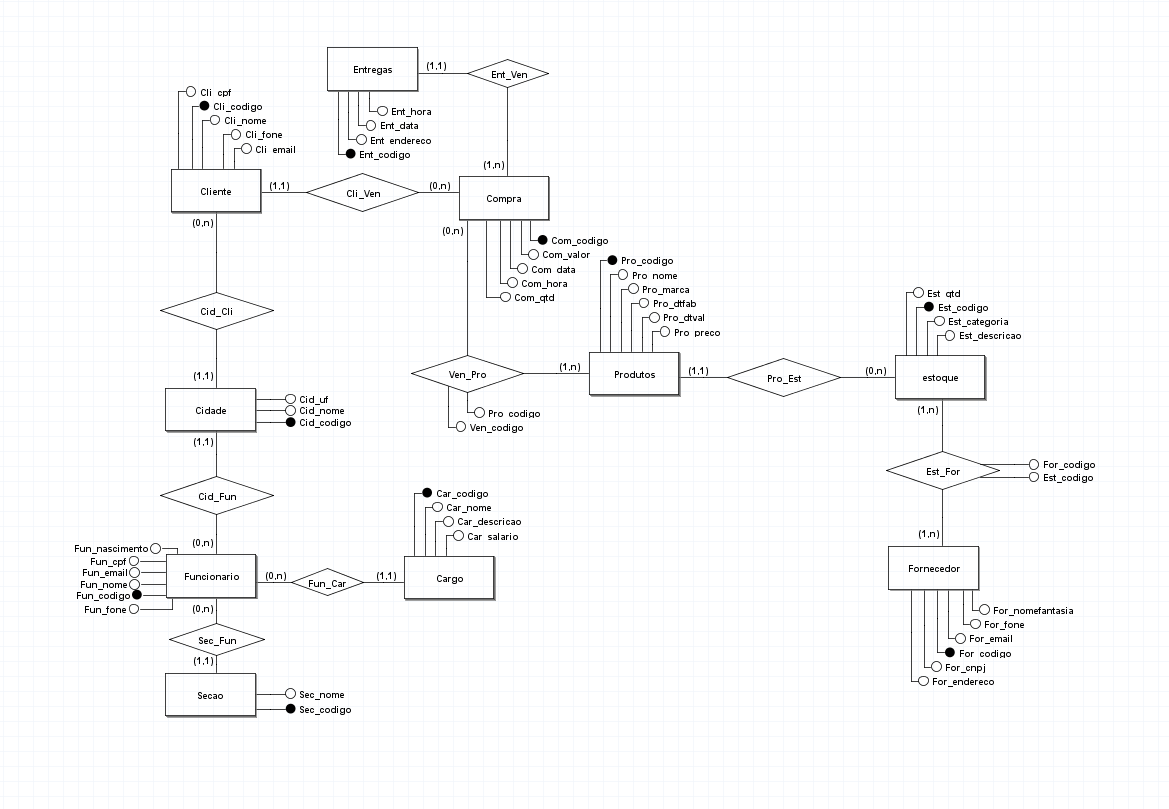
Requisitos Funcionais:

* cadastro de clientes;
* cadastro de produtos;
* registro de vendas;
* registro de estoque;
* cadastro e gerenciamento dos fornecedores;
* registro de entrega de produtos;
* cadastro e gerenciamento de funcionários;
* gerenciamento de cargos;
* sistema de controle de postos de serviço;

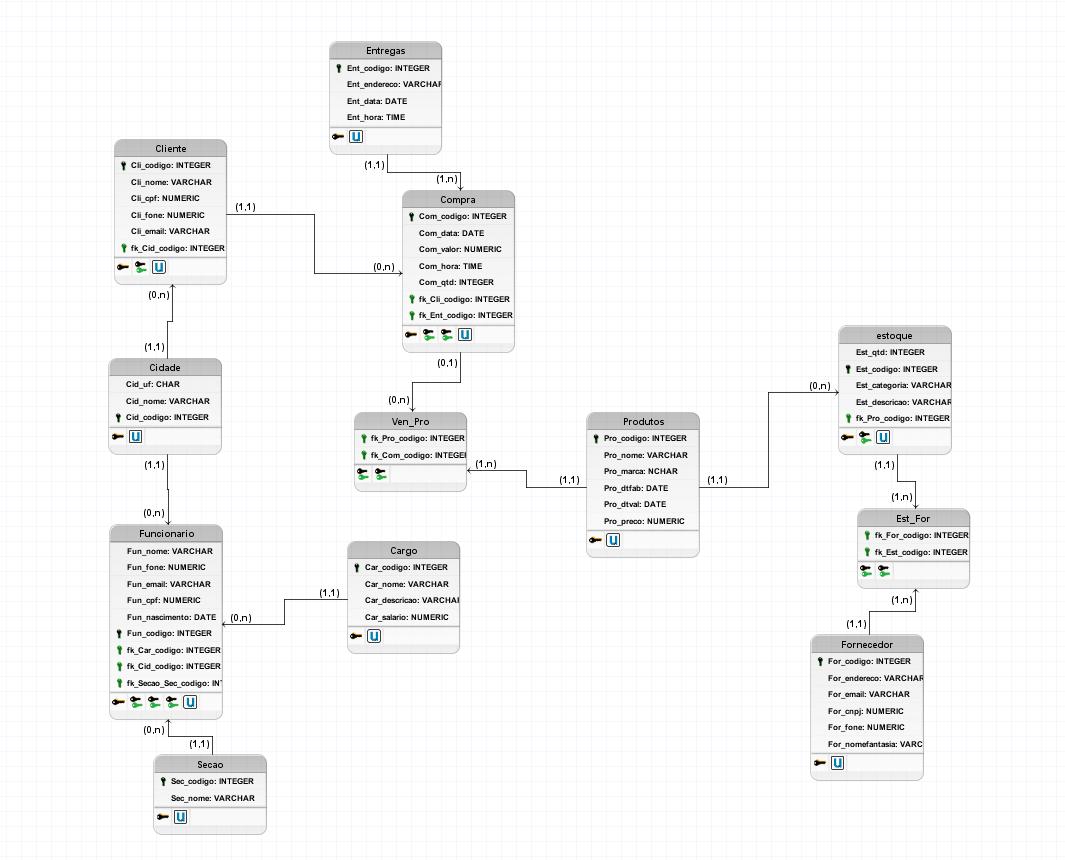
Requisitos Não-Funcionais:

* segurança contra usuários não autorizados;
* otimização e uso com fluidez;
* multiplataforma;
* proteção contra redundâncias;
* uso por mais de um usuário ao mesmo tempo;
* proteção contra cadastro duplicado;
* sistema de login;

***Modelo Conceitual***

******

***Modelo Lógico***

******

***Estrutura Física***

CREATE TABLE Cliente (

Cli\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Cli\_nome VARCHAR(100) NOT NULL,

Cli\_cpf NUMERIC(11) NOT NULL,

Cli\_fone VARCHAR(13) NOT NULL,

Cli\_email VARCHAR(100) NULL,

fk\_Cid\_codigo INTEGER NOT NULL

);

CREATE TABLE Produtos (

Pro\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Pro\_nome VARCHAR(100) NOT NULL,

Pro\_marca VARCHAR(50) NOT NULL,

Pro\_dtfab DATE NOT NULL,

Pro\_dtval DATE NOT NULL,

Pro\_preco NUMERIC(5,2) NOT NULL

);

CREATE TABLE Compra (

Com\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Com\_data DATE NOT NULL,

Com\_valor NUMERIC(5,2) NOT NULL,

Com\_hora TIME NOT NULL,

fk\_Cli\_codigo INTEGER NOT NULL,

fk\_Ent\_codigo INTEGER NOT NULL,

Com\_qtd INTEGER NOT NULL

);

CREATE TABLE estoque (

Est\_qtd INTEGER NOT NULL,

Est\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

fk\_Pro\_codigo INTEGER NOT NULL,

Est\_categoria VARCHAR(50) NOT NULL,

Est\_descricao VARCHAR(100) NOT NULL

);

CREATE TABLE Fornecedor (

For\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

For\_endereco VARCHAR(100) NOT NULL,

For\_email VARCHAR(100) NULL,

For\_cnpj VARCHAR(14) NOT NULL,

For\_fone VARCHAR(13) NOT NULL,

For\_nomefantasia VARCHAR(100) NOT NULL

);

CREATE TABLE Entregas (

Ent\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Ent\_endereco VARCHAR(100) NOT NULL,

Ent\_data DATE NOT NULL,

Ent\_hora TIME NOT NULL

);

CREATE TABLE Funcionario (

Fun\_nome VARCHAR(100) NOT NULL,

Fun\_fone VARCHAR(13) NOT NULL,

Fun\_email VARCHAR(100) NULL,

Fun\_cpf NUMERIC(11) NOT NULL,

Fun\_nascimento DATE NOT NULL,

Fun\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

fk\_Car\_codigo INTEGER NOT NULL,

fk\_Cid\_codigo INTEGER NOT NULL,

fk\_Sec\_codigo INTEGER NOT NULL

);

CREATE TABLE Cargo (

Car\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Car\_nome VARCHAR(100) NOT NULL,

Car\_descricao VARCHAR(200) NOT NULL,

Car\_salario NUMERIC(10,2) NOT NULL

);

CREATE TABLE Secao (

Sec\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL,

Sec\_nome VARCHAR(50) NOT NULL

);

CREATE TABLE Cidade (

Cid\_uf CHAR(2) NOT NULL,

Cid\_nome VARCHAR(100) NOT NULL,

Cid\_codigo SERIAL PRIMARY KEY UNIQUE NOT NULL

);

CREATE TABLE Est\_For (

fk\_For\_codigo INTEGER NOT NULL,

fk\_Est\_codigo INTEGER NOT NULL

);

CREATE TABLE Com\_Pro (

fk\_Pro\_codigo INTEGER NOT NULL,

fk\_Com\_codigo INTEGER NOT NULL

);

ALTER TABLE Cliente ADD CONSTRAINT FK\_Cliente\_2

FOREIGN KEY (fk\_Cid\_codigo)

REFERENCES Cidade (Cid\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Compra ADD CONSTRAINT FK\_Compra\_2

FOREIGN KEY (fk\_Cli\_codigo)

REFERENCES Cliente (Cli\_codigo)

ON DELETE CASCADE;

ALTER TABLE Compra ADD CONSTRAINT FK\_Compra\_3

FOREIGN KEY (fk\_Ent\_codigo)

REFERENCES Entregas (Ent\_codigo)

ON DELETE RESTRICT;

ALTER TABLE estoque ADD CONSTRAINT FK\_estoque\_2

FOREIGN KEY (fk\_Pro\_codigo)

REFERENCES Produtos (Pro\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Funcionario ADD CONSTRAINT FK\_Funcionario\_2

FOREIGN KEY (fk\_Car\_codigo)

REFERENCES Cargo (Car\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Funcionario ADD CONSTRAINT FK\_Funcionario\_3

FOREIGN KEY (fk\_Cid\_codigo)

REFERENCES Cidade (Cid\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Funcionario ADD CONSTRAINT FK\_Funcionario\_4

FOREIGN KEY (fk\_Sec\_codigo)

REFERENCES Secao (Sec\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Est\_For ADD CONSTRAINT FK\_Est\_For\_1

FOREIGN KEY (fk\_For\_codigo)

REFERENCES Fornecedor (For\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Est\_For ADD CONSTRAINT FK\_Est\_For\_2

FOREIGN KEY (fk\_Est\_codigo)

REFERENCES Estoque (Est\_codigo)

ON DELETE SET NULL;

ALTER TABLE Com\_Pro ADD CONSTRAINT FK\_Ven\_Pro\_1

FOREIGN KEY (fk\_Pro\_codigo)

REFERENCES Produtos (Pro\_codigo)

ON DELETE RESTRICT;

ALTER TABLE Com\_Pro ADD CONSTRAINT FK\_Ven\_Pro\_2

FOREIGN KEY (fk\_Com\_codigo)

REFERENCES Compra (Com\_codigo)

ON DELETE SET NULL;

***Inserções:***

insert into cargo(car\_nome, car\_descricao, car\_salario)

values('Empacotador', 'Empacota as compras feitas pelos clientes na área do caixa', 1300),

('Supervisor', 'É responsável pelo treinamento e auxilio dos funcionários nas vendas', 3500),

('Gerente', 'Cuida do gerenciamento das mercadorias reportando o que acontece com cada produto', 5000),

('Caixa', 'Registrar as compras do cliente na caixa registradora', 2000),

('Açougueiro', 'Corta as carnes, realiza a pesagem delas e cuida do açougue', 2600);

insert into cidade(cid\_nome, cid\_uf)

values('Passo Fundo', 'RS'),

('Lagoa Vermelha', 'RS'),

('São Paulo', 'SP'),

('Rio de Janeiro', 'RJ'),

('Belo Horizonte', 'MG');

insert into cliente(cli\_nome, cli\_cpf, cli\_fone, cli\_email, fk\_cid\_codigo)

values('Rafael Albuquerque de Paula', '77379777941','54964191523', 'rafaeldepaula@gmail.com', 2),

('Nikoli de Azeredo Alves', '69234949215', '54987838341', 'nikolialves@gmail.com', 1),

('Pedro Lucas Barboza Chagas', '16820485613', '11998678864', 'pedrochagas@gmail.com', 3),

('Tarcísio Luiz Harres de Almeida', '78125717841', '21969256260', 'tarcisioalmeida@gmail.com', 4),

('Carlo Ferrari', '19956823802', '54992358723', 'carlosalberto@gmail.com', 2),

('Fulano de Tal', '21764919694', '22997348478', 'fulanodetal@gmail.com', 1),

('Augusto Mendes Gonzatto', '84621346480', '31928028028', 'augustogonzatto@gmail.com', 5);

insert into produtos(pro\_nome, pro\_marca, pro\_dtFab, pro\_dtVal, pro\_preco)

values('Sabão em pó 2,2Kg', 'OMO', '2023-06-24', '2023-12-27', 28.32),

('Barra de chocolate', 'Nestlé', '2023-03-02', '2023-05-17', 4.99),

('Alcatra', 'Maturatta', '2023-08-05', '2023-09-24', 21.99),

('Arroz 5Kg','Prato Fino', '2023-04-22', '2023-10-12', 26.90),

('Refrigerante 2L', 'Coca-Cola', '2023-07-15', '2023-10-01', 11.99);

insert into secao(sec\_nome)

values('Hortifruti'),

('Açougue'),

('Padaria'),

('Cereais'),

('Produtos de limpeza');

insert into estoque(est\_qtd, fk\_pro\_codigo, est\_categoria, est\_descricao)

values(500, 1, 'Limpeza', 'Sabão em pó OMO de 2.2Kg removedor de manchas.'),

(50, 3, 'Carne', 'Alcatra Matturatta de corte nobre'),

(200, 2, 'Doce', 'Barra de chocolate nestle ao leite e meio amargo'),

(40, 4, 'Grão', 'Saco de 5kg de arroz branco prato fino'),

(145, 5, 'Refrigerante', 'Coca-Cola 2L normal gelada');

insert into funcionario(fun\_nome, fun\_fone, fun\_email,fun\_cpf, fun\_nascimento, fk\_car\_codigo, fk\_cid\_codigo, fk\_sec\_codigo)

values('Denílson José Seidel', '549231254154', 'denilsonSeidel@gmail.com', '64691864865', '1993-03-14', 2, 2, 1),

('Robson Brum Guerra', '11961724186', 'robsonguerra@gmail.com', '65165891365', '1973-09-18', 3, 3, 3),

('Edimara Sartori', '21944781408', 'edimarasartori@gmail.com', '36685618658', '1997-10-10', 5, 5, 2),

('Adriano Makux de Paula', '54938172597', 'adrianopaula@gmail.com', '87125660739', '1999-05-04', 4, 4, 5),

('Anúbis Graciela Rosetto', '31989562865', 'anubisrosetto@gmail.com', '96539865282', '1983-11-09', 1, 1, 4);

insert into fornecedor(for\_endereco, for\_email, for\_cnpj, for\_fone, for\_nomeFantasia)

values('Rua do urubu, 762', 'cooperativagraos@gmail.com', '29863582069078', '54978263586', 'Cooperativa Grãos'),

('Av. Afonso Pena, 300', 'megaohortifruti@gmail.com', '29658265826073', '54982652060', 'Megão Hortifruti'),

('Av. das Flores, 142', 'padariadascarnes@gmail.com', '32083570250072', '54937509237', 'Padaria das Carnes'),

('Rua fontana, 556', 'quimicalimpeza@gmail.com', '23856823658231', '54910927597', 'Química Limpeza'),

('Rua granjeira, 223', 'granjagalinaceo@gmail.com', '26543786000141', '54928375602', 'Granjeira Galináceo');

insert into entregas(ent\_endereco, ent\_data, ent\_hora)

values('Rua José Bilibio, 211', '2023-09-23', '10:45:00'),

('Rua morom, 67', '2023-07-14', '14:00:00'),

('Rua Araujo Ribeiro', '2023-10-30', '17:50:00'),

('Rua Paternon, 482', '2023-02-28', '09:24:00'),

('Rua Santos Andrade, 501', '2022-11-25', '18:30:00');

insert into compra(com\_data, com\_valor, com\_hora, com\_qtd, fk\_cli\_codigo, fk\_ent\_codigo)

values('2023-09-22', 150, '19:15:00', 5, 3, 1),

('2023-07-13', 37.90, '22:00:00', 8, 7, 2),

('2023-10-30', 354.99,'15:15:00', 16, 1, 3),

('2023-02-28', 247, '08:00:00', 9, 2, 4),

('2023-09-22', 90, '10:20:19', 4, 6, 4),

('2023-05-25', 78.80, '12:00:00', 7, 4, 5);

insert into com\_pro(fk\_pro\_codigo, fk\_com\_codigo)

values(1, 3),

(2, 2),

(3, 4),

(4, 5),

(5, 1);

insert into est\_for(fk\_for\_codigo, fk\_est\_codigo)

values(1, 4),

(3, 3),

(4, 1),

(2, 5),

(5, 2);

***Consultas:***

***1 – Listar a quantidade de funcionário em cada cargo cadastrado, listando somente os***

***cargos que possuem ao menos 1 funcionário.***

*select car\_nome as cargo, count(fk\_car\_codigo) as funcionarios from cargo c, funcionario f*

*where f.fk\_car\_codigo=c.car\_codigo*

*group by fk\_car\_codigo, car\_nome*

***2 – Listar o nome do funcionário, descrição do cargo e nome da cidades dos***

***funcionários que são do signo de touro.***

*select fun\_nome as funcionario, car\_descricao as descricao\_cargo, cid\_nome as cidade*

*from funcionario f, cargo cg, cidade c*

*where (extract(day from fun\_nascimento)>=21 and extract(month from fun\_nascimento)=4*

*or extract(day from fun\_nascimento)<=20 and extract(month from fun\_nascimento)=5)*

*and cg.car\_codigo=f.fk\_car\_codigo and c.cid\_codigo=f.fk\_cid\_codigo*

***3 – Listar todos os produtos que estão vencidos, ou seja, data de validade inferior a***

***data atual.***

*select \* from produtos*

*where pro\_dtval < current\_date*

***4 – Listar o cpf do cliente, do funcionário e o cnpj do fornecedor tudo em uma única***

***coluna.***

*select cli\_cpf as identificacao from cliente*

*union*

*select fun\_cpf as identificacao from funcionario*

*union*

*select cast(for\_cnpj as numeric(14)) as identificacao from fornecedor*

***5 – Listar o total de vendas realizadas na semana de número 21 do ano de 2023***

*select sum(com\_valor) as Total\_vendas from compra*

*where extract(week from com\_data)=21 and extract(year from com\_data)=2023*

***6 – Listar os produtos cadastrados, exibindo o preço normal e o preço promocional***

***com 5% de desconto***

*select pro\_nome as produto, pro\_preco as preco, (pro\_preco-(pro\_preco\*0.05)) as valor\_promocional*

*from produtos*

***7 – Criar uma visão para listar todos os produtos que foram fornecidos pelo***

***fornecedor cujo cnpj é igual 26543786000141***

*create view fornecedor\_cnpj as*

*select pro\_nome as produto, for\_nomefantasia as fornecedor, for\_cnpj as cnpj*

*from produtos p, fornecedor f, estoque e, est\_for ef*

*where p.pro\_codigo=e.fk\_pro\_codigo and e.est\_codigo=ef.fk\_est\_codigo and f.for\_codigo=ef.fk\_for\_codigo*

*and f.for\_cnpj='26543786000141'*

***8 – Listar todos os clientes que residem em cidades do “RS” e o nome inicial pela***

***letra “C” e termina pela letra “O”, listando todas as iniciais maiúscula e o restante***

***minúsculo.***

*select substring(upper(cli\_nome) from 1 for 1) || substring(lower(cli\_nome) from 2 for position(' ' in cli\_nome) -1)*

*|| substring(upper(cli\_nome) from position(' ' in cli\_nome) for 2) || substring(lower(cli\_nome) from position(' ' in cli\_nome) + 2 for length(cli\_nome)) as cliente, cid\_nome || '/' || cid\_uf as cidade\_uf*

*from cidade c, cliente cl*

*where c.cid\_codigo = cl.fk\_cid\_codigo and substring(lower(cli\_nome) from 1 for 1) = 'c' and upper(cid\_uf) = 'RS'*

*and substring(lower(cli\_nome) from (position(' ' in cli\_nome) - 1) for 1) = 'o'*

***9 – Criar uma visão para listar todas as compras efetuadas pelo cliente “Fulano de***

***Tal” mostrando a data, valor de cada compra e o dia da semana por extenso***

***(segunda-feira, terça-feira, etc...)***

*create view Compras\_fulanoDeTal as*

*select cli\_nome as cliente, com\_data as data, com\_valor as preço, case extract(dow from com\_data)*

*when '0' then 'domingo'*

*when '1' then 'segunda'*

*when '2' then 'terça'*

*when '3' then 'quarta'*

*when '4' then 'quinta'*

*when '5' then 'sexta'*

*when '6' then 'sábado'*

*else 'Data inválida'*

*end from cliente c, compra co where co.fk\_cli\_codigo = c.cli\_codigo and lower(cli\_nome) = 'fulano de tal'*

***10 – Criar uma visão para listar todos os funcionários que atuam na seção “Hortifruti”***

***e o salário é maior ou igual a média dos salários de todos os funcionários.***

*create view hortifruiti\_funcionarios as*

*select fun\_nome as funcionario, sec\_nome as seção, car\_salario as salário, (select avg(car\_salario) from cargo)*

*from funcionario f, secao s, cargo c*

*where s.sec\_codigo = f.fk\_sec\_codigo and f.fk\_car\_codigo = c.car\_codigo and car\_salario >= (select avg(car\_salario) from cargo)*

*and lower(sec\_nome) = 'hortifruti'*