Banco de Dados I
Universidade do Estado do Rio de Janeiro
Instituto de Matemática e Estatística
Faculdade de Ciência da Computação

Rede de Farmácias

MINI-MUNDO

É necessário gerenciar uma rede de farmácias. Ela possui funcionários, consumidores e produtos, que podem ou não estar em estoque.

Cada franquia da farmácia tem os seguintes dados relevantes: ID, endereço e telefone.

Esta farmácia trabalha com estoque e com lotes. Caso um produto esteja em estoque, este produto é incluído nos pedidos dos consumidores. Caso contrário, o produto não está disponível para venda, e é requisitado ao centro de distribuição através de um lote. É obrigatório uma farmácia ter um estoque: não pode haver falta de produtos que constem em seu catálogo.

Cada produto tem como dados relevantes: ID, nome, princípio ativo, data de compra, validade, e se há este produto em estoque.

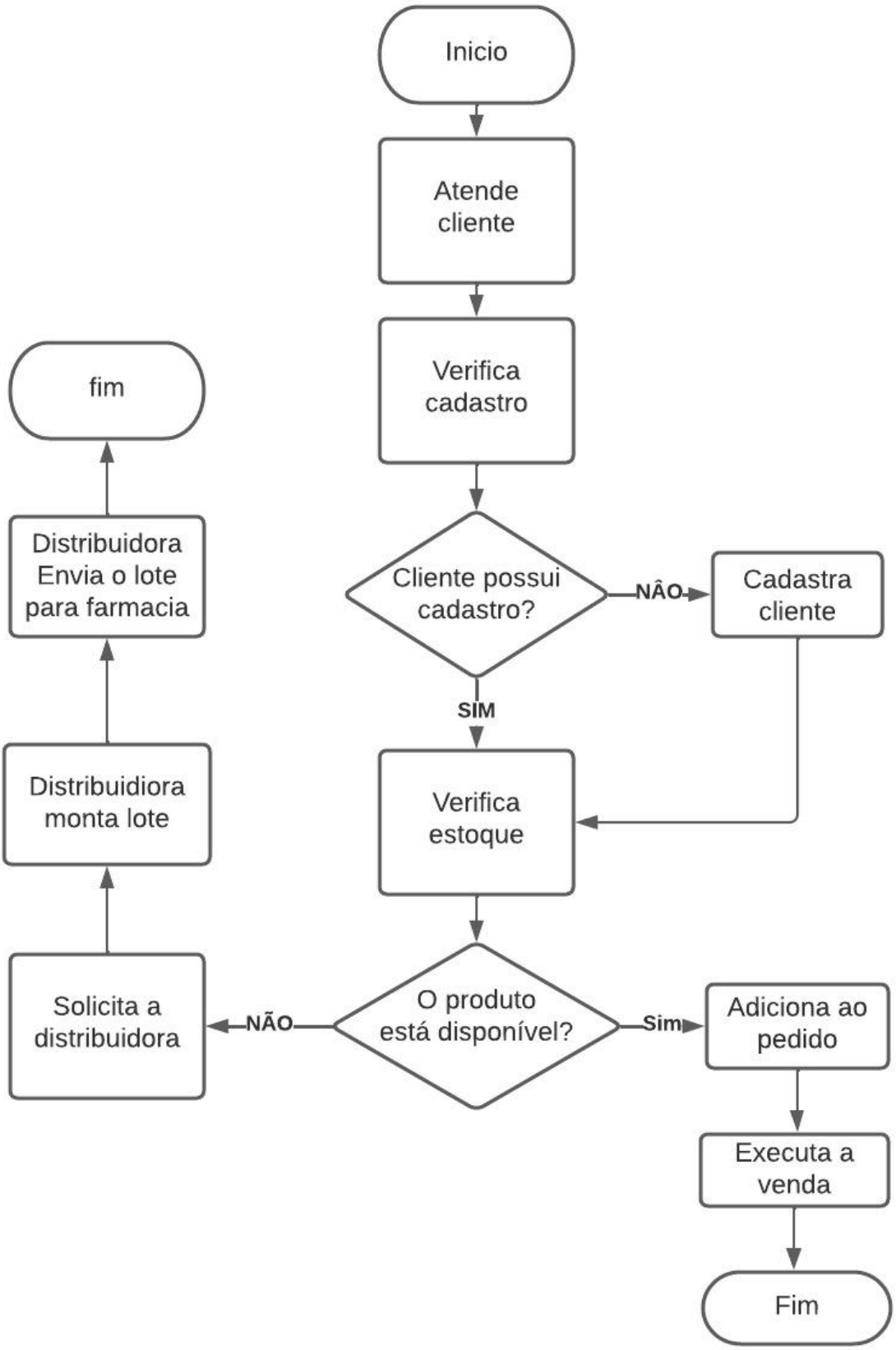
Os funcionários têm como dados relevantes: ID, nome, NIF, franquia em que trabalha, e cargo. O cargo, por sua vez, possui descrição, remuneração, privilégios e funcionários com os quais estão associados. Os privilégios de um determinado cargo envolvem, por exemplo, permissão para alterar dados no sistema, como por exemplo, preços e estoque dos produtos. O status do funcionário no sistema pode ser, a princípio, ativo ou inativo, mas pode incluir outros, como por exemplo, férias.

Cada consumidor cadastrado na rede tem: ID, nome, NIF, telefone, e-mail, endereço. Estes consumidores fazem pedidos, que têm como dados relevantes: ID, valor total e consumidor com o qual está associado. Como o produto deve estar em estoque, é também associado a este.

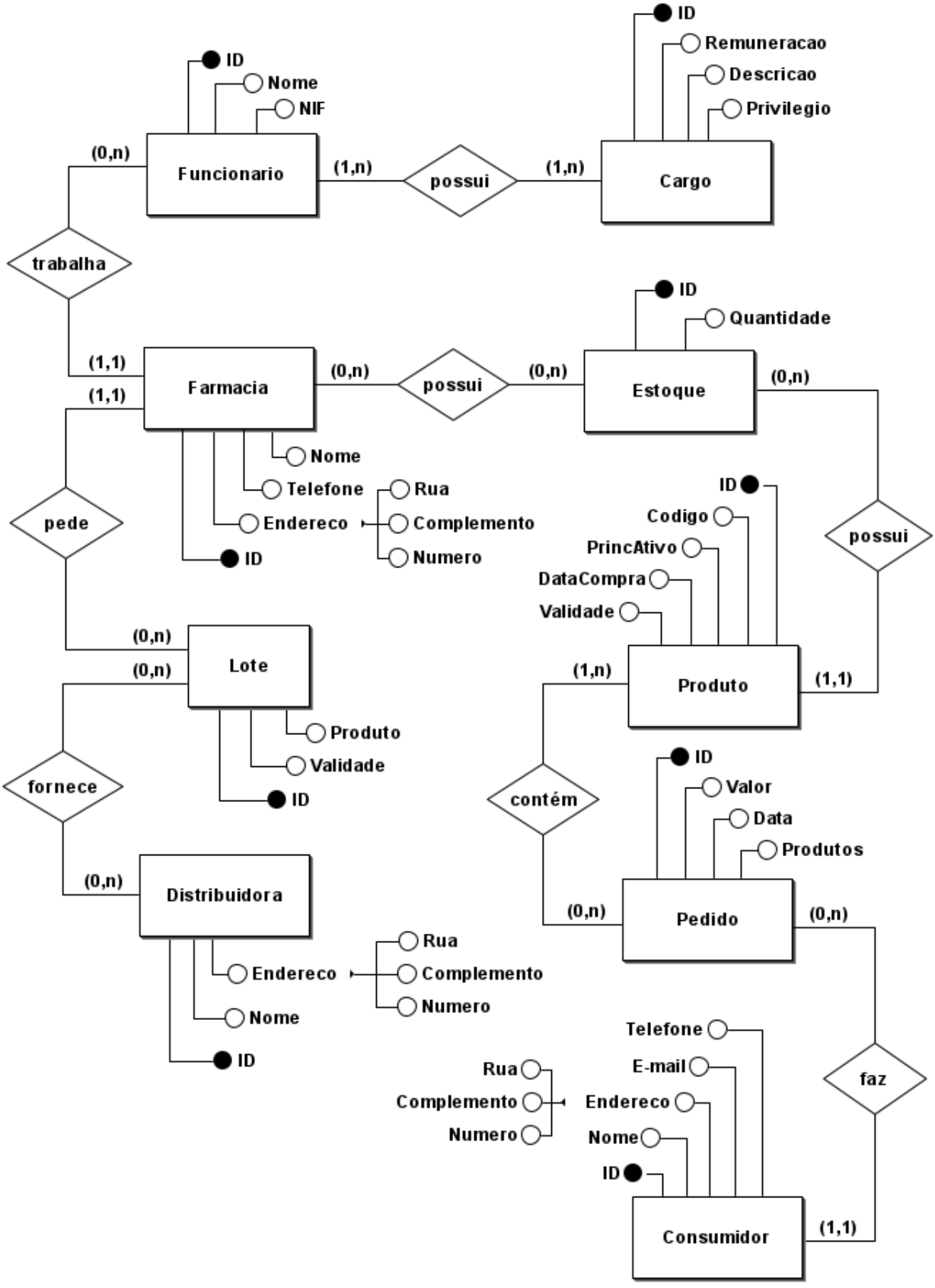
OBJETIVOS

1. Auxiliar os gerentes da rede de farmácias em sua administração;
2. Auxiliar os funcionários na busca por produtos e clientes;
3. Gerir o estoque de produtos da farmácia, e requisitar o distribuidor sempre que houver falta de determinados produtos;

FLUXOGRAMA


Inicio
Atende cliente
Verifica cadastro
Cliente possui cadastro?
NÂO
Cadastra cliente
SIM
Verifica estoque
O produto está disponível?
Sim
Adiciona ao pedido
Executa a venda
Fim
NÃO
Solicita a distribuidora
Distribuidiora monta lote
Distribuidora Envia o lote para farmacia
fim

MODELO DE ENTIDADES E RELACIONAMENTOS



ESQUEMA RELACIONAL

Funcionario (**ID**, Nome, NIF, ***Cargo***, ***Farmacia***)
Funcionario [Cargo] ➜ Cargo [ID]
Funcionario [Farmacia] ➜ Farmacia [ID]

Cargo (**ID**, Remuneracao, Descricao, Privilegio)

Farmacia (**ID**, Rua, Complemento, Numero, Telefone, Nome)

Estoque (**ID**, Quantidade, ***Farmacia***)
Estoque [Farmacia] ➜ Farmacia [ID]

Lote (**ID**, ***Farmacia***, ***Distribuidora***, Validade)
Lote [Farmacia] ➜ Farmacia [ID]
Lote [Distribuidora] ➜ Distribuidora [ID]

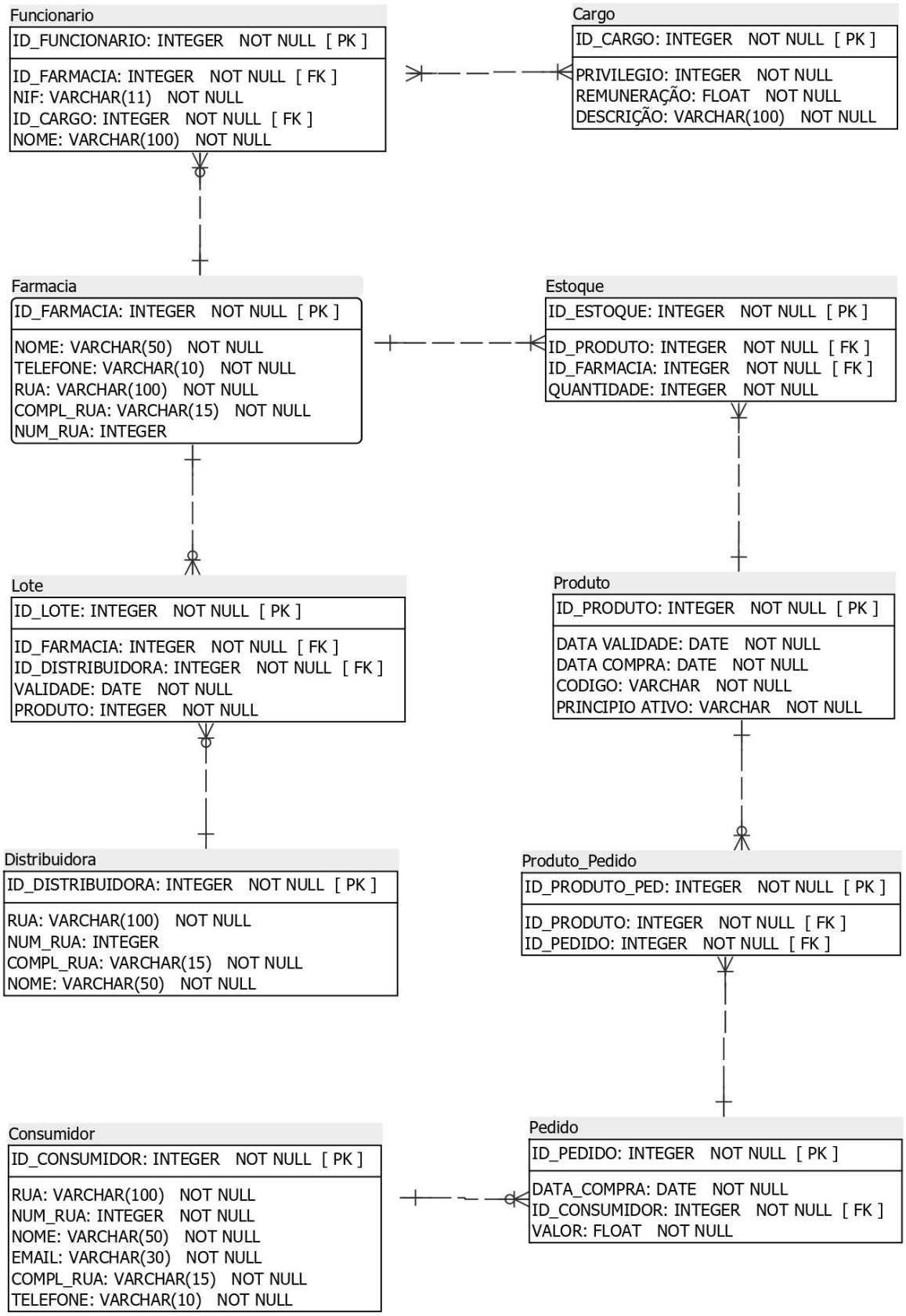
Produto (**ID**, Nome, PrincAtivo, DataCompra, Validade, ***Estoque***)
Produto [Estoque] ➜ Estoque [Quantidade]

Distribuidora (**ID**, Nome, Rua, Complemento, Numero)

Pedido (**ID**, ***Consumidor***, Valor, Data, ***Produtos***)
Pedido [Consumidor] ➜ Consumidor [ID]
Pedido [Produtos] ➜ Produto [ID]

Consumidor (**ID**, Nome, NIF, Telefone, Email, Rua, Complemento, Numero)

DIAGRAMA LÓGICO



FORMA NORMAL

Dada a tabela **Pedido**:

> Pedido (**ID**, ***Consumidor***, Valor, Data, ***Produtos***)
> Pedido [Consumidor] ➜ Consumidor [ID]
> Pedido [Produtos] ➜ Produto [ID]

Vemos que o atributo **Produtos** é multivalorado, ou seja, não está na primeira forma normal. Resolveremos isto com uma nova tabela **Produto_Pedido**, com as seguintes características:

> Produto_Pedido (**ID,** ***Produto***, ***Pedido***)
> Produto_Pedido [Produto] ➜ Produto [ID]
> Produto_Pedido [Pedido] ➜ Pedido[ID]
>
> Pedido (**ID**, ***Consumidor***, Valor, Data)
> Pedido [Consumidor] ➜ Consumidor [ID]

Após isso vemos que a tabela se encontra na segunda e na terceira forma normal.

RESTRIÇÕES DE INTEGRIDADE

1. Um pedido deve ter, no mínimo, um produto;
2. Um produto só pode ser incluído em um pedido se houver estoque deste;
3. Um lote só pode ser requisitado para um produto se não houver estoque deste;
4. Uma distribuidora e uma filial da farmácia não podem ter o mesmo endereço;
5. Uma filial da farmácia não pode ter dois funcionários no cargo de gerente;
6. Caso um funcionário também seja um consumidor, ele deve ter um cadastro separado;
7. Um funcionário da rede tem desconto de 15% no valor de seus pedidos;

CONSULTAS AO BANCO DE DADOS

1. **CRUD:** funcionário, cargo, consumidor, produto, filial, distribuidora, pedido, lote;
2. **Query:** montar um pedido à requisição de um consumidor;
3. **Query:** montar um lote para a distribuidora à requisição de uma filial da farmácia;
4. **Relação** de produtos em estoque em uma certa filial, ordenada por data de compra;
5. **Relação** de funcionários de toda a rede, agrupados por filial e cargo;
6. **Levantamento** de lotes requisitados a uma distribuidora, organizados por data;
7. **Faturamento** do dia, através dos valores dos pedidos feitos nas últimas 24 horas;
8. **Cruzar** funcionários que também são consumidores, de forma a conceder-lhes desconto;

ÁLGEBRA RELACIONAL

1. N/A;
2. N/A;
3. N/A;
4. Data γ (π Estoque > 0, Farmacia = “Foo” (σ Produtos))
5. Farmacia, Cargo γ (Funcionario ⋈ Farmacia);
6. Data γ (π Distribuidora = “Foo” (σ Lote));
7. Valor γ (π Data = “Foo” (σ Pedido));
8. π NIF Funcionario ∩ π NIF Consumidor;
   Obs.: “Foo” é um nome genérico

TRADUÇÃO PARA SQL

1. INSERT SELECT UPDATE DELETE Funcionário, Cargo, Consumidor, Produto, Farmacia, Distribuidora, Pedido, Lote;

   INSERT INTO "G4_Farmacia"."funcionario"(id_farmacia,nif,id_cargo,nome) VALUES (3,'02346456088',5,'Francisco Zero Ponto Clnco');

   INSERT INTO "G4_Farmacia"."consumidor"(rua,num_rua,compl_rua,telefone,email,nome) values ('Rua Major Freitas',88,'fundos','21998784567','lobatoo@hotmail.com','Lucas Lobat0');

   ALTER TABLE "G4_Farmacia"."produto_pedido" DROP COLUMN id_produto ;

   INSERT INTO "G4_Farmacia"."produto" (data_validade,data_compra,codigo,principio_ativo) VALUES ('2024-12-25','2022-04-20','26878989799','Hidroxicloroquina');

   UPDATE "G4_Farmacia"."consumidor" set nome='Lucas Lobato' WHERE nome='Lucas Lobat0'
2. INSERT INTO Pedido (ID, Consumidor, Valor, Data, Produtos) VALUES (“ID”, “Consumidor”, “Valor”, “Data”, “Produtos”);
3. INSERT INTO Lote (ID, Farmacia, Distribuidora, Validade) VALUES (“ID”, “Farmacia”, “Distribuidora”, “Validade”);
4. SELECT * FROM Produtos WHERE Estoque > 0;
5. SELECT * FROM Funcionario, Farmacia GROUP BY Farmacia, Cargo;
6. SELECT * FROM Lote WHERE Distribuidora = “Foo” ORDER BY Data;
7. SELECT * FROM Pedido WHERE Data = “Foo”, SUM Valor;
8. SELECT NIF FROM Funcionario INTERSECT SELECT NIF FROM Consumidor;

ESQUEMA FÍSICO

```
CREATE DATABASE G4_Farmacia;

DROP TABLE IF EXISTS Consumidor;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Consumidor_seq;

CREATE TABLE Consumidor(
id INT  DEFAULT NEXTVAL ('Consumidor_seq') PRIMARY KEY,
num_rua INT NOT NULL,
rua VARCHAR(100) NOT NULL,
compl_rua VARCHAR(30) NOT NULL,
nome VARCHAR(50) NOT NULL,
telefone VARCHAR(10) NOT NULL,
email VARCHAR(50) NOT NULL
);

DROP TABLE IF EXISTS pedido;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Pedido_seq;

CREATE TABLE Pedido(
id INT  DEFAULT NEXTVAL ('Pedido_seq') PRIMARY KEY,
produtos INT NOT NULL,
data_compra DATE NOT NULL,
id_consumidor INT NOT NULL,
valor DOUBLE PRECISION NOT NULL
);

DROP TABLE IF EXISTS Produto_pedido;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Produto_pedido_seq;

CREATE TABLE Produto_pedido(
id INT  DEFAULT NEXTVAL ('Produto_pedido_seq') PRIMARY KEY,
id_produto INT NOT NULL,
id_pedido INT NOT NULL
);

DROP TABLE IF EXISTS produto;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Produto_seq;

CREATE TABLE Produto(
id INT  DEFAULT NEXTVAL ('Produto_seq') PRIMARY KEY,
data_validade DATE NOT NULL,
data_compra DATE NOT NULL,
codigo VARCHAR(10) NOT NULL,
principio_ativo VARCHAR(50) NOT NULL
```

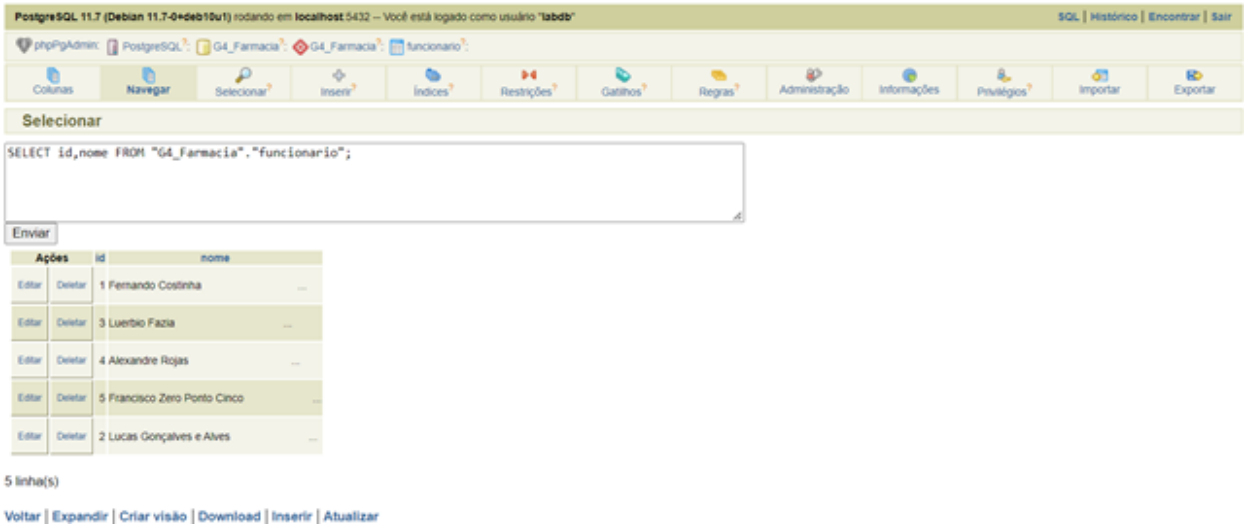
```
);

DROP TABLE IF EXISTS Estoque;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Estoque_seq;

CREATE TABLE Estoque(
id INT  DEFAULT NEXTVAL ('Estoque_seq') PRIMARY KEY,
id_produto INT NOT NULL,
id_farmacia INT NOT NULL,
quantidade INT NOT NULL
);

DROP TABLE IF EXISTS Farmacia;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Farmacia_seq;

CREATE TABLE Farmacia(
id INT DEFAULT NEXTVAL ('Farmacia_seq') PRIMARY KEY,
nome VARCHAR(50) NOT NULL,
telefone VARCHAR(10) NOT NULL,
rua VARCHAR(100) NOT NULL,
compl_rua VARCHAR(15) NOT NULL,
email VARCHAR(50) NOT NULL,
num_rua int
);

DROP TABLE IF EXISTS Lote;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Lote_seq;

CREATE TABLE Lote(
id INT  DEFAULT NEXTVAL ('Lote_seq') PRIMARY KEY,
id_farmacia INT NOT NULL,
id_distribuidora INT NOT NULL,
validade DATE NOT NULL,
produto INT NOT NULL
);

DROP TABLE IF EXISTS Distribuidora;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Distribuidora_seq;

CREATE TABLE Distribuidora(
id INT  DEFAULT NEXTVAL ('Distribuidora_seq') PRIMARY KEY,
num_rua int ,
compl_rua VARCHAR(15) NOT NULL,
nome VARCHAR(50) NOT NULL,
rua VARCHAR(100) NOT NULL
);
```

```
DROP TABLE IF EXISTS Funcionario;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Funcionario_seq;

CREATE TABLE Funcionario(

id INT  DEFAULT NEXTVAL ('Funcionario_seq') PRIMARY KEY,
id_farmacia INT NOT NULL,
nif VARCHAR(11) NOT NULL,
nome VARCHAR(100) NOT NULL,
id_cargo INT NOT NULL
);

DROP TABLE IF EXISTS Cargo;
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE SEQUENCE Cargo_seq;

CREATE TABLE Cargo(
id INT  DEFAULT NEXTVAL ('Cargo_seq') PRIMARY KEY,
privilegio int NOT NULL,
remuneração double precision NOT NULL,
descricao VARCHAR(100) NOT NULL
);

ALTER TABLE Pedido
ADD CONSTRAINT FK_consumidor_Pedido
FOREIGN KEY (id_consumidor) REFERENCES Consumidor(id);

ALTER TABLE Produto_pedido
ADD CONSTRAINT FK_pedido_produto_pedido
FOREIGN KEY (id_pedido) REFERENCES Pedido(id);

ALTER TABLE Produto_pedido
ADD CONSTRAINT FK_produto_produto_pedido
FOREIGN KEY (id_produto) REFERENCES Produto(id);

ALTER TABLE Estoque
ADD CONSTRAINT FK_produto_estoque
FOREIGN KEY (id_produto) REFERENCES Produto(id);

ALTER TABLE Estoque
ADD CONSTRAINT FK_farmacia_estoque
FOREIGN KEY (id_farmacia) REFERENCES Farmacia(id);

ALTER TABLE Lote
ADD CONSTRAINT FK_farmacia_lote
FOREIGN KEY (id_farmacia) REFERENCES Farmacia(id);

ALTER TABLE Lote
```

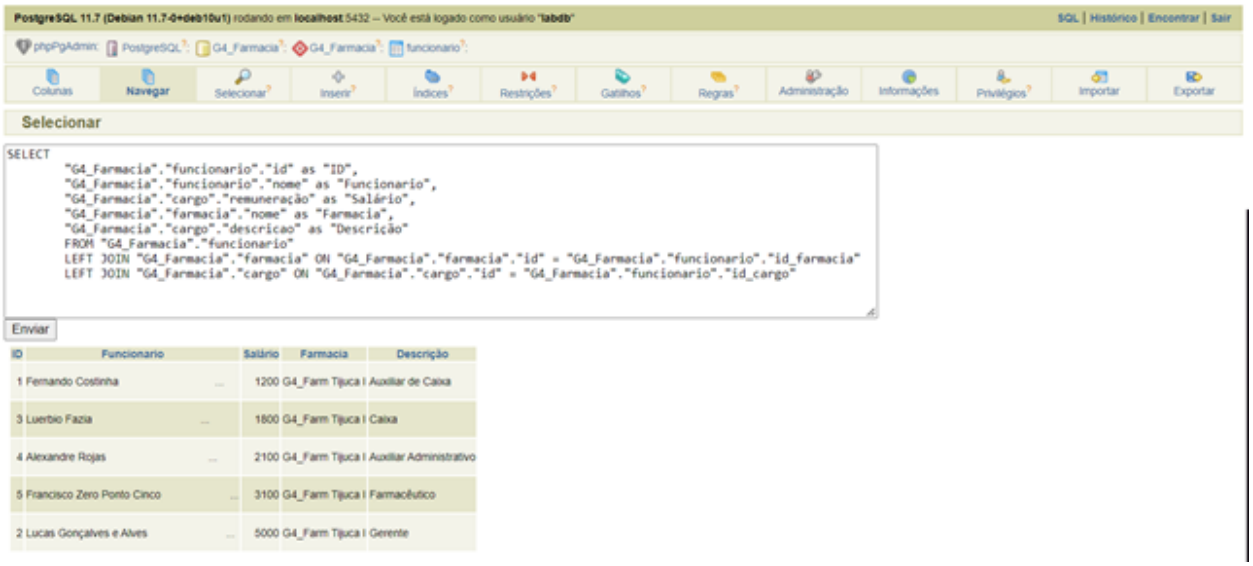
```
ADD CONSTRAINT FK_distribuidora_lote
FOREIGN KEY (id_distribuidora) REFERENCES Distribuidora(id);

ALTER TABLE Funcionario
ADD CONSTRAINT FK_cargo_funcionario
FOREIGN KEY (id_cargo) REFERENCES Cargo(id);

ALTER TABLE Funcionario
ADD CONSTRAINT FK_farmcia_funcionario
FOREIGN KEY (id_farmacia) REFERENCES Farmacia(id);
```

CONSULTAS EM SQL

1. **Consulta simples:**
   Visualizar todos os funcionários da farmácia

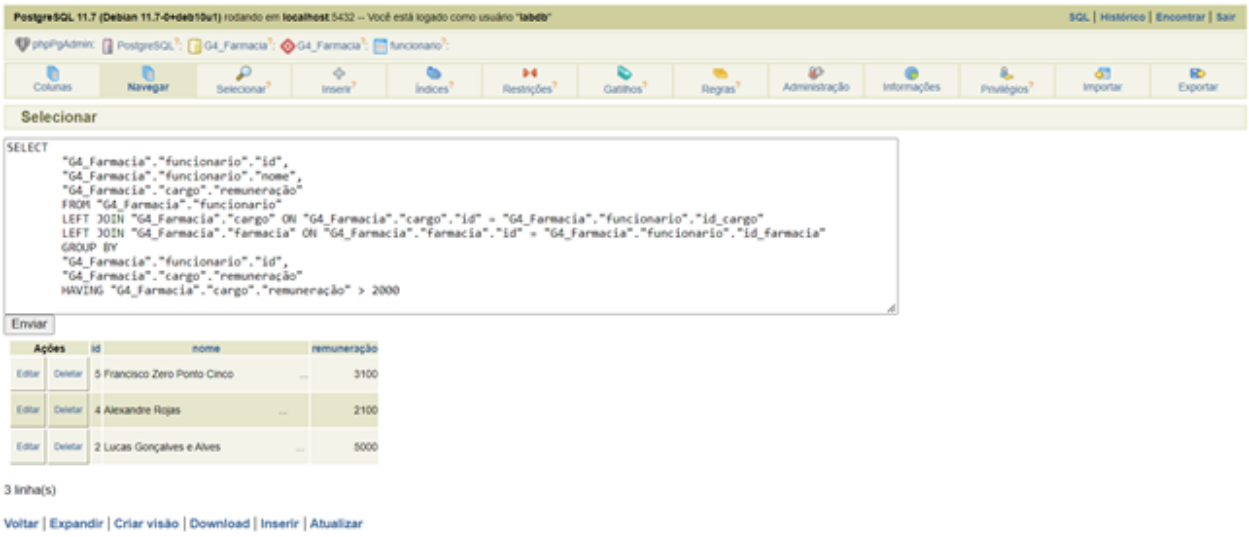


2. **Junção de três tabelas:**
   Visualizar todos os funcionários, assim como seus respectivos salários, onde trabalham e a função que exercem.



3. **Agrupamento por agregação:**
   Consultar o gasto total com os salários dos funcionários.

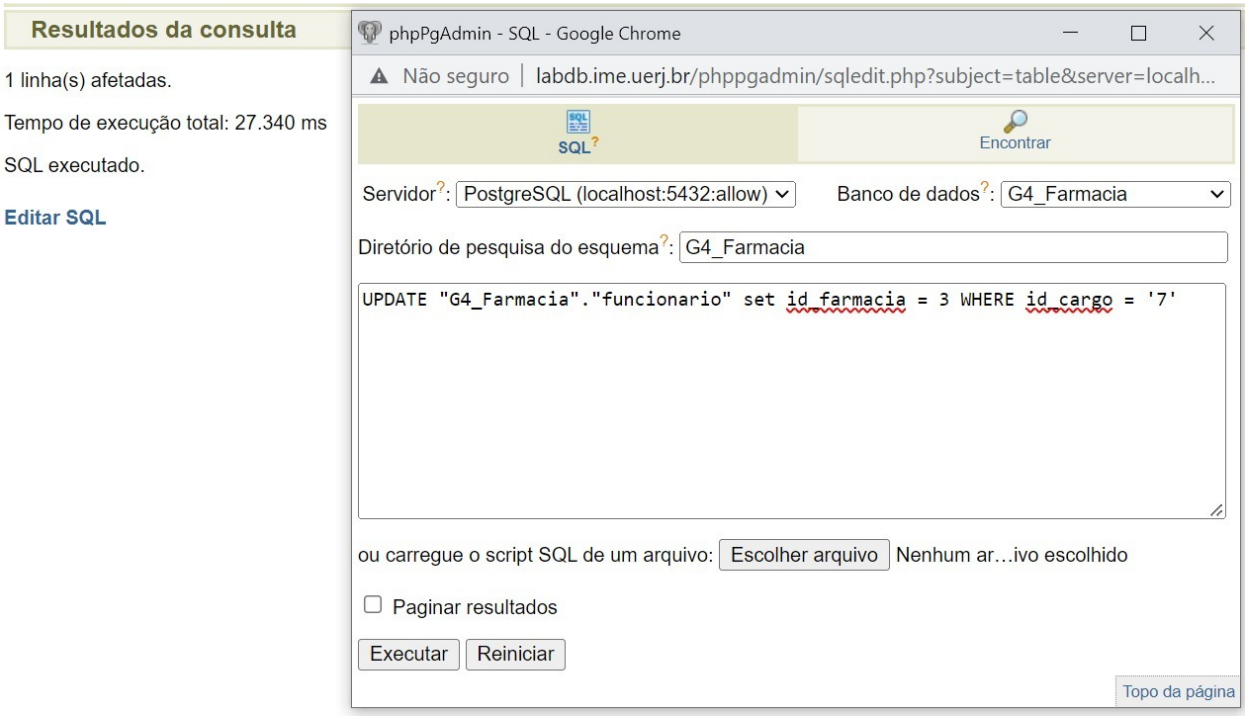
4. **Restrição sobre agrupamento:**
   Listar todos os funcionários com salário maior que dois mil reais.



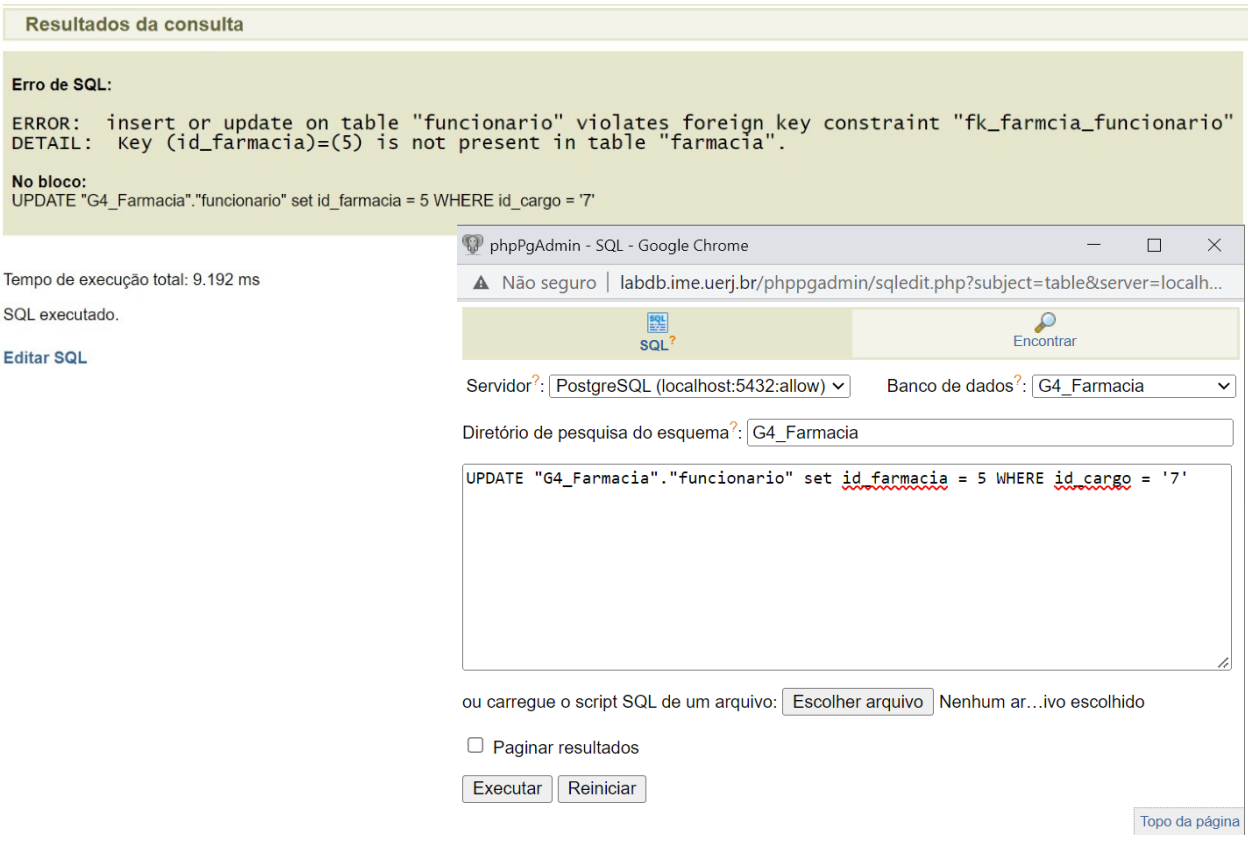
TESTES DE CHAVE ESTRANGEIRA

Vamos transferir um funcionário gerente de uma filial para outra.

Sucesso: filial existente.



Erro: filial inexistente.



TESTES DE REMOÇÃO DE DADOS

Um lote trouxe no banco de dados um tipo de produto que não pertence à distribuidora. Isso deve ter ocorrido devido a um problema na implementação. Portanto deve ser removida.

```
SELECT lt.id,lt.id_farmacia,dr.nome,pr.principio_ativo as distribuidora FROM "G4_Farmacia"."lote" lt
INNER JOIN "G4_Farmacia"."distribuidora" dr on dr.id=lt.id_distribuidora INNER JOIN
"G4_Farmacia"."produto" pr on pr.id=lt.produto;
```

Fazer a consulta

Ações	id	id_farmacia	nome	distribuidora
Editar Deletar	1	3	Insul Manipulada	insulina glargina
Editar Deletar	2	1	Insul Manipulada	insulina glargina
Editar Deletar	3	2	Insul Manipulada	insulina glargina
Editar Deletar	4	2	Colgate SA	Enxaguante Bucal
Editar Deletar	5	1	Pirulas Rocha	Enxaguante Bucal
Editar Deletar	6	2	Pirulas Rocha	Hidroxicloroquina

DELETE FROM "G4_Farmacia"."lote" where id=5;

Assim teremos novos valores.

```
SELECT lt.id,lt.id_farmacia,dr.nome,pr.principio_ativo as distribuidora FROM "G4_Farmacia"."lote" lt
INNER JOIN "G4_Farmacia"."distribuidora" dr on dr.id=lt.id_distribuidora INNER JOIN
"G4_Farmacia"."produto" pr on pr.id=lt.produto;
```

Fazer a consulta

Ações	id	id_farmacia	nome	distribuidora
Editar Deletar	1	3	Insul Manipulada	insulina glargina
Editar Deletar	2	1	Insul Manipulada	insulina glargina
Editar Deletar	3	2	Insul Manipulada	insulina glargina
Editar Deletar	4	2	Colgate SA	Enxaguante Bucal
Editar Deletar	6	2	Pirulas Rocha	Hidroxicloroquina

5 linha(s)