

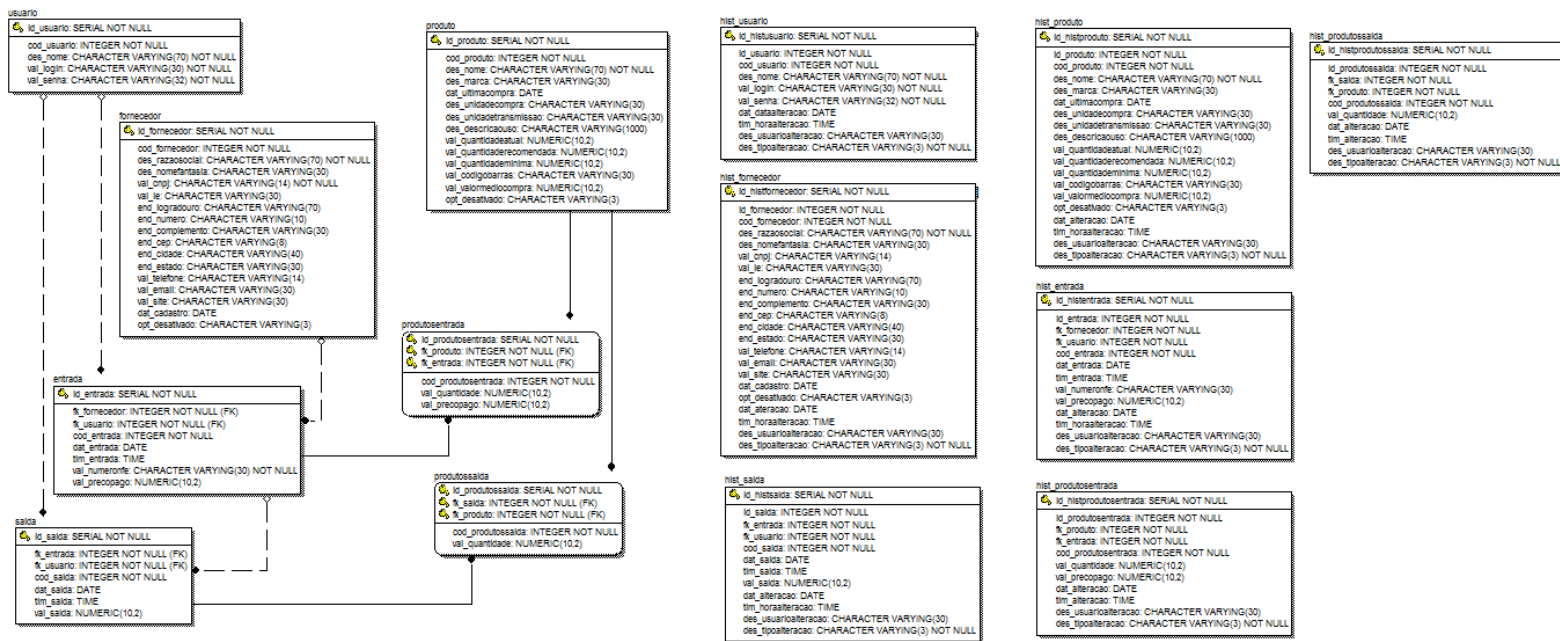
Projeto do Banco de Dados

Membros do grupo: Alex, Thiago, Paulo, Rodrigo

Tema do trabalho: Sistema de compras para minimercado

SGBD escolhido: Postgres sql

1. Diagrama de Entidade Relacionamento:



2. Script de criação do banco de dados

```
CREATE DATABASE buckstar WITH OWNER usudba;
```

```
CREATE SCHEMA scmbucstar;
```

```
SET SCHEMA scmbuckstar;
```

```
CREATE TABLE entrada (
    id_entrada CHARACTER VARYING(30) NOT NULL,
    cod_entrada INTEGER NOT NULL,
    fk_fornecedor INTEGER NOT NULL,
    dat_entrada DATE,
    tim_entrada TIME,
    val_numeronfe CHARACTER VARYING(30),
    fk_usuario INTEGER NOT NULL
);
```

```
CREATE UNIQUE INDEX idx_entrada ON entrada(id_entrada);
```

```
ALTER TABLE entrada
    ADD PRIMARY KEY (id_entrada);
```

```
CREATE TABLE fornecedor (
    id_fornecedor SERIAL NOT NULL,
    cod_fornecedor INTEGER NOT NULL,
    des_razaosocial CHARACTER VARYING(70) NOT NULL,
```

```

        des_nomefantasia CHARACTER VARYING(30),
        val_cnpj          CHARACTER VARYING(14),
        val_ie            CHARACTER VARYING(30),
        end_logradouro    CHARACTER VARYING(70),
        end_numero        CHARACTER VARYING(10),
        end_complemento   CHARACTER VARYING(30),
        end_cep           CHARACTER VARYING(8),
        end_cidade        CHARACTER VARYING(40),
        end_estado        CHARACTER VARYING(30),
        val_telefone      CHARACTER VARYING(14),
        val_email          CHARACTER VARYING(30),
        val_site          CHARACTER VARYING(30),
        dat_cadastro      DATE,
        opt_desativado    CHARACTER VARYING(3)
    );

CREATE UNIQUE INDEX idx_fornecedor ON fornecedor(id_fornecedor);

```

```

ALTER TABLE fornecedor
    ADD PRIMARY KEY (id_fornecedor);

```

```

CREATE TABLE hist_entrada (
    id_histentrada SERIAL NOT NULL,
    cod_entrada    INTEGER NOT NULL,
    fk_fornecedor  INTEGER NOT NULL,
    dat_entrada    DATE,
    tim_entrada    TIME,
    val_numeronfe  CHARACTER VARYING(30),
    fk_usuario     INTEGER NOT NULL,
    id_entrada     INTEGER NOT NULL,
    dat_dataalteracao DATE,
    tim_horaalteracao TIME,
    des_usuarioalteracao CHARACTER VARYING(30),
    des_tipoalteracao CHARACTER VARYING(3)
);

CREATE UNIQUE INDEX idx_hist_entrada ON hist_entrada(id_histentrada);

```

```

ALTER TABLE hist_entrada
    ADD PRIMARY KEY (id_histentrada);

```

```

CREATE TABLE hist_fornecedor (
    id_histfornecedor SERIAL NOT NULL,
    cod_fornecedor     INTEGER NOT NULL,
    des_razaosocial    CHARACTER VARYING(70) NOT NULL,
    des_nomefantasia   CHARACTER VARYING(30),
    val_cnpj           CHARACTER VARYING(14),
    val_ie             CHARACTER VARYING(30),
    end_logradouro     CHARACTER VARYING(70),
    end_numero         CHARACTER VARYING(10),
    end_complemento    CHARACTER VARYING(30),
    end_cep            CHARACTER VARYING(8),
    end_cidade         CHARACTER VARYING(40),

```

```

        end_estado      CHARACTER VARYING(30),
        val_telefone     CHARACTER VARYING(14),
        val_email        CHARACTER VARYING(30),
        val_site         CHARACTER VARYING(30),
        dat_cadastro     DATE,
        opt_desativado   CHARACTER VARYING(3),
        id_fornecedor    INTEGER NOT NULL,
        dat_dataalteracao DATE,
        tim_horaalteracao TIME,
        des_usuarioalteracao CHARACTER VARYING(30),
        des_tipoalteracao CHARACTER VARYING(3)
    );

CREATE UNIQUE INDEX idx_hist_fornecedor ON hist_fornecedor(id_histfornecedor);

ALTER TABLE hist_fornecedor
    ADD PRIMARY KEY (id_histfornecedor);

CREATE TABLE hist_produto (
    id_histproduto      SERIAL NOT NULL,
    cod_produto         INTEGER NOT NULL,
    des_nome            CHARACTER VARYING(30) NOT NULL,
    des_marca           CHARACTER VARYING(30),
    dat_ultimacompra    DATE,
    des_unidadecompra   CHARACTER VARYING(30),
    des_unidadetransmissao CHARACTER VARYING(30),
    des_descricaouso    CHARACTER VARYING(1000),
    val_quantidadeatual NUMERIC(10,2),
    val_quantidaderecomendada NUMERIC(10,2),
    val_quantidademinima NUMERIC(10,2),
    val_codigobarras    CHARACTER VARYING(30),
    val_valormediacompra NUMERIC(10,2),
    opt_desativado      CHARACTER VARYING(3),
    fk_ultimofornecedor INTEGER,
    id_produto          INTEGER NOT NULL,
    dat_dataalteracao   DATE,
    tim_horaalteracao   TIME,
    des_usuarioalteracao CHARACTER VARYING(30),
    des_tipoalteracao   CHARACTER VARYING(3)
);

CREATE UNIQUE INDEX idx_hist_produto ON hist_produto(id_histproduto);

ALTER TABLE hist_produto
    ADD PRIMARY KEY (id_histproduto);

CREATE TABLE hist_saida (
    id_histsaida        SERIAL NOT NULL,
    cod_saida           INTEGER NOT NULL,
    id_saida            INTEGER NOT NULL,
    fk_usuario          INTEGER NOT NULL,
    dat_dataalteracao   DATE,
    tim_horaalteracao   TIME,
    des_usuarioalteracao CHARACTER VARYING(30),
    des_tipoalteracao   CHARACTER VARYING(3)
);

```

);

CREATE UNIQUE INDEX idx_hist_saida ON hist_saida(id_histsaida);

ALTER TABLE hist_saida
ADD PRIMARY KEY (id_histsaida);

CREATE TABLE hist_usuario (
id_histusuario SERIAL NOT NULL,
cod_usuario INTEGER NOT NULL,
des_nome CHARACTER VARYING(70) NOT NULL,
val_login CHARACTER VARYING(30) NOT NULL,
val_senha CHARACTER VARYING(32) NOT NULL,
id_usuario INTEGER NOT NULL,
des_tipoalteracao CHARACTER VARYING(3) NOT NULL,
dat_dataalteracao DATE,
tim_horaalteracao TIME,
des_usuarioalteracao CHARACTER VARYING(30)
);

CREATE UNIQUE INDEX idx_hist_usuario ON hist_usuario(id_histusuario);

ALTER TABLE hist_usuario
ADD PRIMARY KEY (id_histusuario);

CREATE TABLE produto (
id_produto SERIAL NOT NULL,
cod_produto INTEGER NOT NULL,
des_nome CHARACTER VARYING(30) NOT NULL,
des_marca CHARACTER VARYING(30),
dat_ultimacompra DATE,
des_unidadecompra CHARACTER VARYING(30),
des_unidadetransmissao CHARACTER VARYING(30),
des_descricao CHARACTER VARYING(1000),
val_quantidadeatual NUMERIC(10,2),
val_quantidaderecomendada NUMERIC(10,2),
val_quantidademinima NUMERIC(10,2),
val_codigobarras CHARACTER VARYING(30),
val_valormediacompra NUMERIC(10,2),
opt_desativado CHARACTER VARYING(3),
fk_ultimofofornecedor INTEGER
);

CREATE UNIQUE INDEX idx_produto ON produto(id_produto);

ALTER TABLE produto
ADD PRIMARY KEY (id_produto);

CREATE TABLE produtosentrada (
id_produtosentrada SERIAL NOT NULL,
id_produto INTEGER NOT NULL,
id_entrada INTEGER NOT NULL,
val_quantidade NUMERIC(10,2),
val_precopago NUMERIC(10,2),

```
cod_produtosentrada INTEGER NOT NULL
);
```

```
CREATE UNIQUE INDEX idx_produtosentrada ON produtosentrada(id_produtosentrada, id_produto, id_entrada);
```

```
ALTER TABLE produtosentrada
ADD PRIMARY KEY (id_produtosentrada, id_produto, id_entrada);
```

```
CREATE TABLE produtossaida (
    id_produtossaida CHARACTER VARYING(30) NOT NULL,
    cod_produtossaida INTEGER NOT NULL,
    id_saida CHARACTER VARYING(30) NOT NULL,
    id_produto SERIAL NOT NULL,
    val_quantidade NUMERIC(10,2)
);
```

```
CREATE UNIQUE INDEX idx_produtossaida ON produtossaida(id_produtossaida, id_saida, id_produto);
```

```
ALTER TABLE produtossaida
ADD PRIMARY KEY (id_produtossaida, id_saida, id_produto);
```

```
CREATE TABLE saida (
    id_saida CHARACTER VARYING(30) NOT NULL,
    cod_saida INTEGER NOT NULL,
    fk_entrada INTEGER NOT NULL,
    fk_usuario INTEGER NOT NULL
);
```

```
CREATE UNIQUE INDEX idx_saida ON saida(id_saida);
```

```
ALTER TABLE saida
ADD PRIMARY KEY (id_saida);
```

```
CREATE TABLE usuario (
    id_usuario SERIAL NOT NULL,
    cod_usuario INTEGER NOT NULL,
    des_nome CHARACTER VARYING(70) NOT NULL,
    val_login CHARACTER VARYING(30) NOT NULL,
    val_senha CHARACTER VARYING(32) NOT NULL
);
```

```
CREATE UNIQUE INDEX idx_usuario ON usuario(id_usuario);
```

```
ALTER TABLE usuario
ADD PRIMARY KEY (id_usuario);
```

```
ALTER TABLE entrada
ADD FOREIGN KEY (fk_usuario)
REFERENCES usuario
ON DELETE SET NULL
ON UPDATE SET NULL;
```

```
ALTER TABLE entrada
  ADD FOREIGN KEY (fk_fornecedor)
    REFERENCES fornecedor
    ON DELETE SET NULL
    ON UPDATE SET NULL;
```

```
ALTER TABLE produto
  ADD FOREIGN KEY (fk_ultimofornecedor)
    REFERENCES fornecedor
    ON DELETE SET NULL
    ON UPDATE SET NULL;
```

```
ALTER TABLE produtosentrada
  ADD FOREIGN KEY (id_entrada)
    REFERENCES entrada;
```

```
ALTER TABLE produtosentrada
  ADD FOREIGN KEY (id_produto)
    REFERENCES produto;
```

```
ALTER TABLE produtossaida
  ADD FOREIGN KEY (id_produto)
    REFERENCES produto;
```

```
ALTER TABLE produtossaida
  ADD FOREIGN KEY (id_saida)
    REFERENCES saida;
```

```
ALTER TABLE saida
  ADD FOREIGN KEY (fk_entrada)
    REFERENCES entrada
    ON DELETE SET NULL
    ON UPDATE SET NULL;
```

```
CREATE OR REPLACE FUNCTION hist_usuarioDEL() RETURNS TRIGGER AS
$$
BEGIN
    INSERT INTO hist_usuario(id_usuariohist, id_usuario, cod_usuario, des_nome, val_login,
val_senha, des_tipoalteracao, dat_dataalteracao, tim_horaalteracao, des_usuarioalteracao)
VALUES(DEFAULT, OLD.id_usuario, OLD.cod_usuario, OLD.des_nome, OLD.val_login, OLD.val_senha,
'DEL', NOW(), NOW(), CURRENT_USER);
    RETURN OLD;
END;
$$
LANGUAGE 'plpgsql'
CREATE TRIGGER ins_histusuarioDEL BEFORE DELETE ON usuario
FOR EACH ROW
EXECUTE PROCEDURE hist_usuarioDEL();
CREATE OR REPLACE FUNCTION hist_clienteDEL() RETURNS TRIGGER AS
```

```

$$
BEGIN
    INSERT INTO hist_cliente(id_clientehist, id_cliente, cod_cliente, des_razaosocial,
des_nomefantasia, val_cnpj, val_inscricaoestadual, dat_datadecadastro, opt_clientedesativado,
opt_clientemanutencao, end_logradouro, end_numero, end_complemento, end_bairro, end_cep,
end_cidade, end_uf, des_contato, val_fone, val_email, fk_vendedor, fk_representante,
dat_dataalteracao, tim_horaalteracao, des_usuarioalteracao, des_tipoalteracao) VALUES(DEFAULT,
OLD.id_cliente, OLD.cod_cliente, OLD.des_razaosocial, OLD.des_nomefantasia, OLD.val_cnpj,
OLD.val_inscricaoestadual, OLD.dat_datadecadastro, OLD.opt_clientedesativado,
OLD.opt_clientemanutencao, OLD.end_logradouro, OLD.end_numero, OLD.end_complemento,
OLD.end_bairro, OLD.end_cep, OLD.end_cidade, OLD.end_uf, OLD.des_contato, OLD.val_fone,
OLD.val_email, OLD.id_clientehist, OLD.fk_vendedor, OLD.fk_representante, 'DEL', NOW(), NOW(),
CURRENT_USER);
    RETURN OLD;
END;
$$
LANGUAGE 'plpgsql'

```

3. Descrição dos grupos de usuários

Haverá somente dois tipos de grupos no sistema: Administradores e Comuns.

O grupo Administradores poderá visualizar e alterar todas as tabelas.

O grupo Comum poderá visualizar e alterar todas as tabelas, exceto as tabelas de histórico e a de usuário.

Haverá três usuários no sistema: usudba, usuadm e usucli.

O usuário usudba será o usuário que terá acesso a todos os recursos do banco (Superuser) e pertencerá ao grupo Administradores, herdando suas permissões.

O usuário usuadm pertencerá ao grupo Administradores e herdando suas permissões, esse usuário terá acesso às tabelas do grupo.

O usuário usucli pertencerá ao grupo Comum, herdará suas permissões e terá acesso apenas de consulta na tabela usuario.

4. Código de criação dos grupos usuários e permissões.

```

--> Grupos
CREATE ROLE Administradores;
CREATE ROLE Comuns;

--> Usuários
CREATE USER usuadm PASSWORD 'adM8ucks74r' IN ROLE
Administradores;
CREATE USER usucli PASSWORD 'Cl13n73*' IN ROLE Comuns;
CREATE USER usudba PASSWORD 'DBA8uck*74r' SUPERUSER CREATEDB
CREATEROLE IN ROLE Administradores;

--> Removendo todas as permissões
REVOKE ALL PRIVILEGES ON DATABASE buckstar FROM GROUP public;
REVOKE ALL PRIVILEGES ON DATABASE buckstar FROM GROUP Comuns;

--> Concedendo as permissões
GRANT USAGE ON SCHEMA scmbuckstar TO usuadm, usucli, usudba;
GRANT ALL ON TABLE entrada, fornecedor, produto,
produtosentrada, produtossaida, saida TO Comuns;
GRANT SELECT ON TABLE usuario TO usucli;

```

```
GRANT ALL ON TABLE entrada, fornecedor, produto,  
produtosentrada, produtossaida, saida, usuario, hist_entrada,  
hist_fornecedor, hist_produto, hist_saida, hist_usuario TO  
Administradores;
```