**Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof.Luciano B. Fernandes

**TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

**Projeto de banco de dados para um sistema de** CRM para sistema de Varejo (Magazines)

Isis Costa Bez Birolo [@Isiscostabb](https://github.com/isiscostabb)

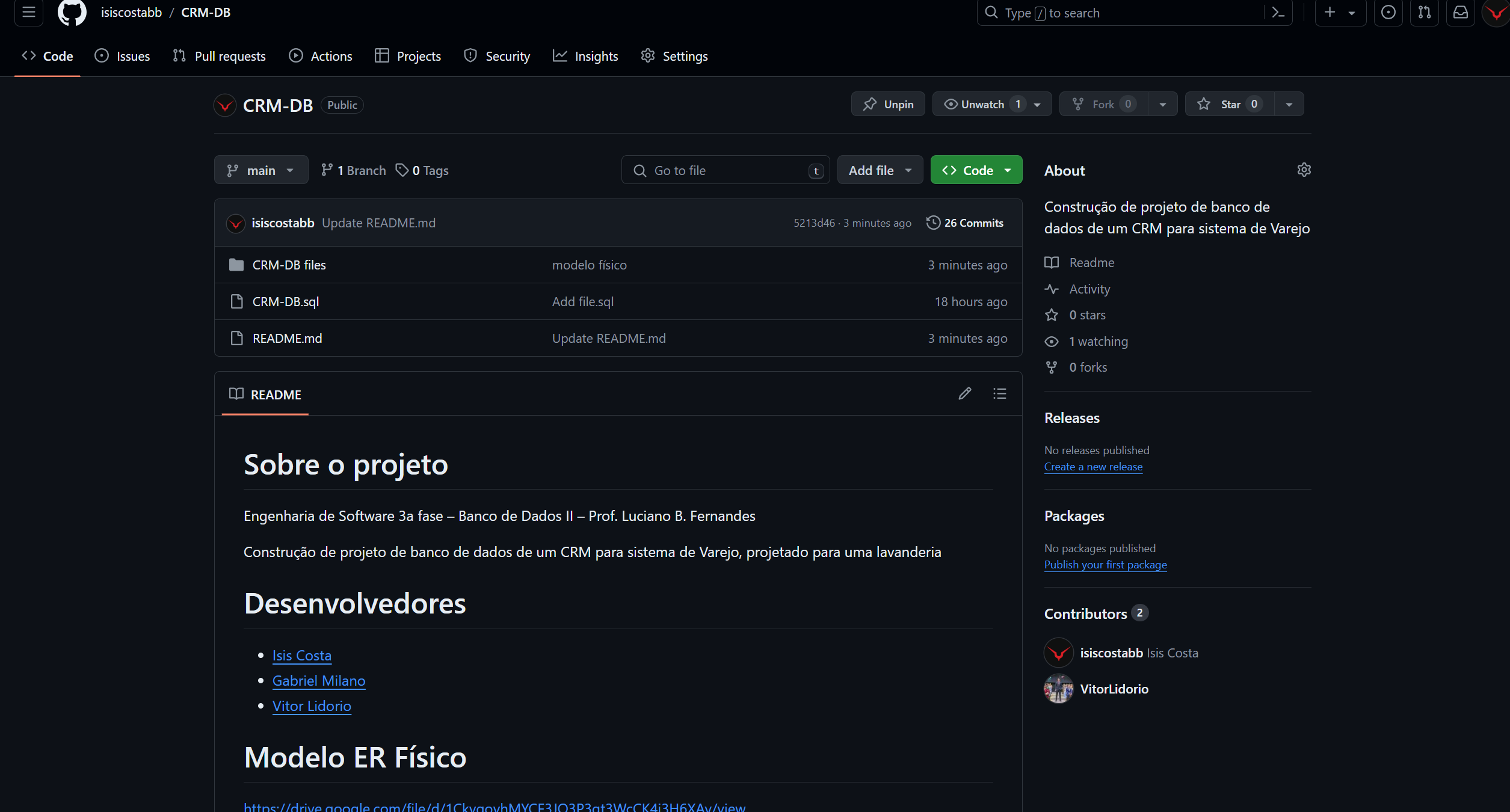
Gabriel Milano Alves [@gabrielmilano](https://github.com/gabrielmilano)

Vitor Ribeiro Lidorio [@vitorLidorio](https://github.com/VitorLidorio)

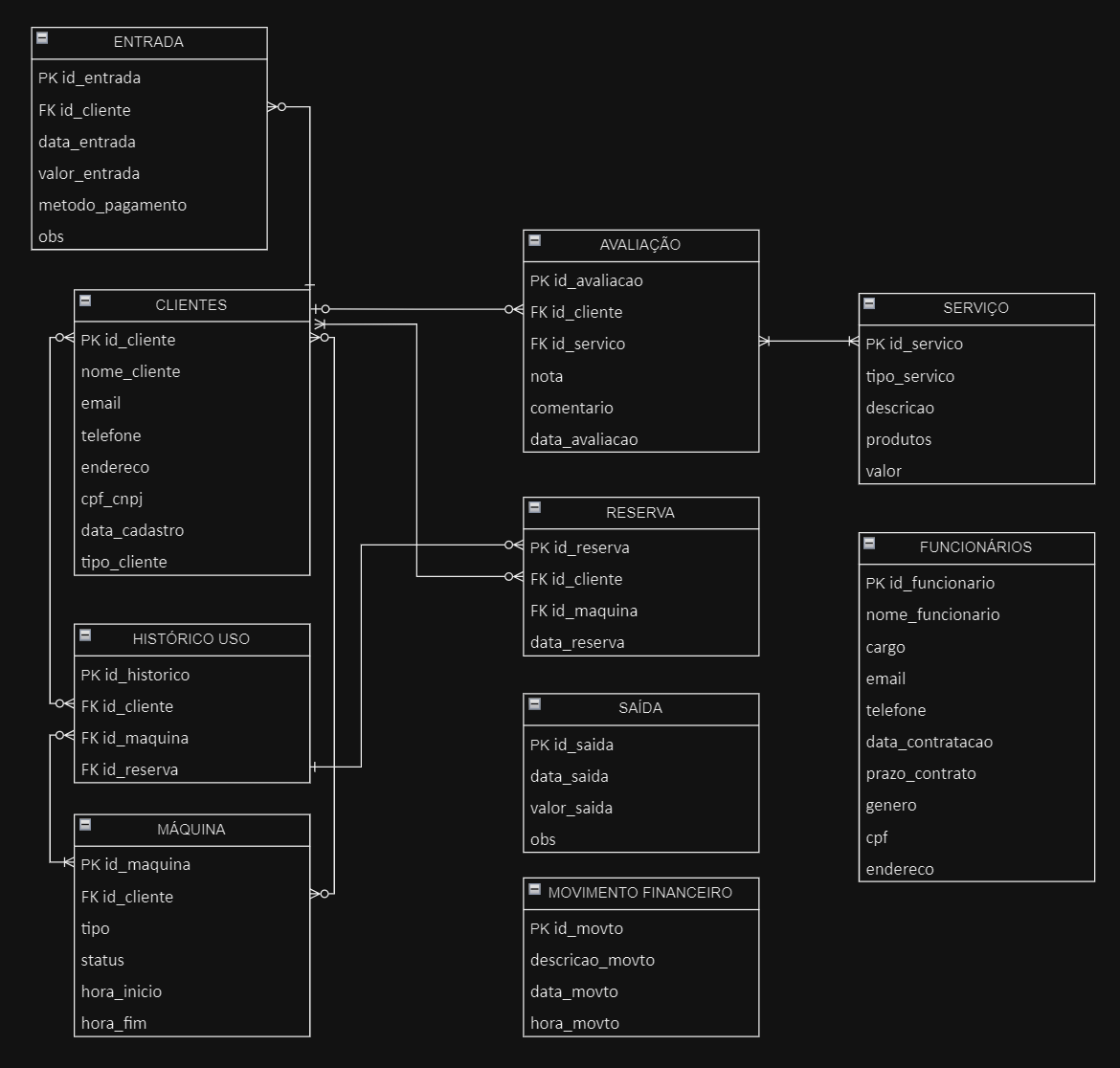
Criciúma, 10/06/2024

**URL do projeto no GitHub**

<https://github.com/isiscostabb/CRM-DB>



**Modelo ER Físico**



**Dicionário de Dados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ATRIBUTOS** | | | | | | | |
| **Avaliacao** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_avaliacao | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador da avaliação |
| id\_cliente |  | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do cliente |
| id\_servico | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do serviço |
| nota | | numeric | 0 -10 | NOT NULL |  |  | Nota referente ao serviço |
| comentario | | text | sem limite | NOT NULL |  |  | Comentário do cliente |
| data\_avaliacao | | date | data | NOT NULL |  |  | Data em que a avaliação foi realizada |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Clientes** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_cliente | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do cliente |
| nome\_cliente | | character varying | 0 - 30 | NOT NULL |  |  | Nome do cliente |
| email | | character varying | 0 - 64 | NOT NULL |  |  | Email do cliente |
| telefone | | character varying | 0 - 30 | NOT NULL |  |  | Telefone do cliente |
| endereço | | character varying | 0 - 30 | NOT NULL |  |  | Endereço do cliente |
| cpf\_cnpj | | character varying | 0 - 20 | NOT NULL |  |  | Cpf ou Cnpj do cliente |
| data\_cadastro | | date | data | NOT NULL |  |  | Data que o cadastro do cliente foi efetuado |
| tipo\_cliente | | character varying | 0 - 50 | NOT NULL |  |  | Tipo de cliente |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Entrada** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_entrada | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador da entrada |
| id\_cliente | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do cliente |
| data\_entrada | | date | data | NOT NULL |  |  | Data em que a entrada foi efetuada |
| valor\_entrada | | double precision | double precision | NOT NULL |  |  | Valor da entrada |
| metodo\_pagamento | | character varying | 0 -15 | NOT NULL |  |  | Metodo escolhido para realizar o pagamento |
| obs | | text | sem limite | NOT NULL |  |  | Campo para observações |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Funcionarios** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_funcionario | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do funcionário |
| nome\_funcionario | | character varying | 0 - 30 | NOT NULL |  |  | Nome do funcionário |
| cargo | | character varying | 0 -30 | NOT NULL |  |  | Cargo do funcionário |
| email | | character varying | 0 - 64 | NOT NULL |  |  | Email do funcionário |
| telefone | | character varying | 0 -30 | NOT NULL |  |  | Telefone do funcionário |
| data\_contratacao | | date | data | NOT NULL |  |  | Data em que o funcionário foi contratado |
| prazo\_contrato | | character varying | 0 - 30 | NOT NULL |  |  | Prazo de contrato do funcionário |
| genero | | character varying | 0 -15 | NOT NULL |  |  | Genero do funcionário |
| cpf | | character varying | 0 -20 | NOT NULL |  |  | Cpf do funcionário |
| endereco | | character varying | 0 -30 | NOT NULL |  |  | Endereço do funcionário |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Maquina** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_maquina | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do pedido |
| id\_cliente | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do cliente |
| tipo |  | character varying | 0 - 30 | NOT NULL |  |  | Tipo de máquina |
| status |  | character varying | 0 -30 | NOT NULL |  |  | Status da máquina |
| hora\_inicio | | time | hora | NOT NULL |  |  | Horário de incio de uso da máquina |
| hora\_fim | | time | hora | NOT NULL |  |  | Horário que termina o uso da máquina |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Reserva** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_reserva | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador da entrada |
| id\_cliente |  | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do cliente |
| id\_maquina |  | int | 1 - sem limite | NOT NULL |  | X | Código de identificador da máquina |
| data\_reserva |  | date | data | NOT NULL |  |  | Data escolhida para a reserva da máquina |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Saida** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_saida | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador da saída |
| data\_saida |  | date | data | NOT NULL |  |  | Data da saída |
| valor\_saida | | double precision | double precision | NOT NULL |  |  | Valor da saída |
| obs |  | text | sem limite | NOT NULL |  |  | Campo para observações |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **Servico** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_servico | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do serviço |
| tipo\_servico | | character varying | 0 -30 | NOT NULL |  |  | Tipo de serviço |
| descricao | | character varying | 0 - 30 | NOT NULL |  |  | Descrição do serviço |
| produtos | | character varying | 0 -30 | NOT NULL |  |  | Produtos usados no serviço |
| valor | | double precision | double precision | NOT NULL |  |  | Valor do serviço |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **historicoUso** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_historico | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do histórico de uso |
| id\_cliente | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador do cliente |
| id\_maquina | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador da máquina |
| id\_reserva | | int | 1 - sem limite | NOT NULL |  | X | Código de identificador da reserva |
|  | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **ATRIBUTOS** | | | | | | | |
| **movimento\_financeiro** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| id\_movto | | int | 1 - sem limite | NOT NULL | X |  | Código de identificador do movimento financeiro |
| descricao\_movto | | character varying | 0 - 5 | NOT NULL |  |  | Descrição sobre o movimento financeiro |
| data\_movto | | date | data | NOT NULL |  |  | Data do movimento financeiro |
| hora\_movto | | time | hora | NOT NULL |  |  | Horário do movimento financeiro |
|  |  |  |  |  |  |  |  |
| **INDÍCES** | | | | | | | |
| **Nome do indíce** | | **Clustered** | **Non Clustered** | **Unique** | **Colunas** |  |  |
| idx\_gin\_saida | |  | X |  | Saida(obs) |  |  |
| idx\_gin\_entrada | |  | X |  | Entrada(obs) |  |  |
| idx\_servico | |  | X |  | Servico(tipo\_servico) |  |  |
| idx\_gin\_avaliacao | |  | X |  | Avaliacao(comentario) |  |  |

**Índices**

CREATE EXTENSION pg\_trgm;

CREATE INDEX idx\_gin\_saida ON Saida USING gin (obs gin\_trgm\_ops);

CREATE INDEX idx\_gin\_entrada ON Entrada USING gin (obs gin\_trgm\_ops);

CREATE INDEX idx\_gin\_avaliacao ON Avaliacao USING gin (comentario gin\_trgm\_ops);

CREATE INDEX idx\_servico ON Servico (tipo\_servico);

**Script dos comandos DDL para criação do Banco de dados**

-- Table: public.avaliacao

-- DROP TABLE IF EXISTS public.avaliacao;

CREATE TABLE IF NOT EXISTS public.avaliacao

(

id\_avaliacao integer NOT NULL,

id\_cliente integer NOT NULL,

id\_servico integer NOT NULL,

nota numeric(10,0) NOT NULL,

comentario text COLLATE pg\_catalog."default" NOT NULL,

data\_avaliacao date NOT NULL,

CONSTRAINT avaliacao\_pkey PRIMARY KEY (id\_avaliacao),

CONSTRAINT id\_cliente\_fk FOREIGN KEY (id\_cliente)

REFERENCES public.clientes (id\_cliente) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_servico\_fk FOREIGN KEY (id\_servico)

REFERENCES public.servico (id\_servico) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.avaliacao

OWNER to postgres;

REVOKE ALL ON TABLE public.avaliacao FROM estagiario;

REVOKE ALL ON TABLE public.avaliacao FROM funcionario;

REVOKE ALL ON TABLE public.avaliacao FROM gerente;

GRANT SELECT ON TABLE public.avaliacao TO estagiario;

GRANT UPDATE, SELECT, INSERT ON TABLE public.avaliacao TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.avaliacao TO gerente;

GRANT ALL ON TABLE public.avaliacao TO postgres;

-- Index: idx\_gin\_avaliacao

-- DROP INDEX IF EXISTS public.idx\_gin\_avaliacao;

CREATE INDEX IF NOT EXISTS idx\_gin\_avaliacao

ON public.avaliacao USING gin

(comentario COLLATE pg\_catalog."default" gin\_trgm\_ops)

TABLESPACE pg\_default;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.clientes

-- DROP TABLE IF EXISTS public.clientes;

CREATE TABLE IF NOT EXISTS public.clientes

(

id\_cliente integer NOT NULL,

nome\_cliente character varying(30) COLLATE pg\_catalog."default" NOT NULL,

email character varying(64) COLLATE pg\_catalog."default" NOT NULL,

telefone character varying(30) COLLATE pg\_catalog."default" NOT NULL,

endereco character varying(30) COLLATE pg\_catalog."default" NOT NULL,

cpf\_cnpj character varying(20) COLLATE pg\_catalog."default" NOT NULL,

data\_cadastro date NOT NULL,

tipo\_cliente character varying(50) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT clientes\_pkey PRIMARY KEY (id\_cliente)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.clientes

OWNER to postgres;

REVOKE ALL ON TABLE public.clientes FROM estagiario;

REVOKE ALL ON TABLE public.clientes FROM funcionario;

REVOKE ALL ON TABLE public.clientes FROM gerente;

GRANT SELECT ON TABLE public.clientes TO estagiario WITH GRANT OPTION;

GRANT UPDATE, SELECT, INSERT ON TABLE public.clientes TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.clientes TO gerente;

GRANT ALL ON TABLE public.clientes TO postgres;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.entrada

-- DROP TABLE IF EXISTS public.entrada;

CREATE TABLE IF NOT EXISTS public.entrada

(

id\_entrada integer NOT NULL,

id\_cliente integer NOT NULL,

data\_entrada date NOT NULL,

valor\_entrada double precision NOT NULL,

metodo\_pagamento character varying(15) COLLATE pg\_catalog."default" NOT NULL,

obs text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT entrada\_pkey PRIMARY KEY (id\_entrada),

CONSTRAINT id\_cliente\_fk FOREIGN KEY (id\_cliente)

REFERENCES public.clientes (id\_cliente) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.entrada

OWNER to postgres;

REVOKE ALL ON TABLE public.entrada FROM funcionario;

REVOKE ALL ON TABLE public.entrada FROM gerente;

GRANT UPDATE, INSERT, SELECT ON TABLE public.entrada TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.entrada TO gerente;

GRANT ALL ON TABLE public.entrada TO postgres;

-- Index: idx\_gin\_entrada

-- DROP INDEX IF EXISTS public.idx\_gin\_entrada;

CREATE INDEX IF NOT EXISTS idx\_gin\_entrada

ON public.entrada USING gin

(obs COLLATE pg\_catalog."default" gin\_trgm\_ops)

TABLESPACE pg\_default;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.funcionarios

-- DROP TABLE IF EXISTS public.funcionarios;

CREATE TABLE IF NOT EXISTS public.funcionarios

(

id\_funcionario integer NOT NULL,

nome\_funcionario character varying(30) COLLATE pg\_catalog."default" NOT NULL,

cargo character varying(30) COLLATE pg\_catalog."default" NOT NULL,

email character varying(64) COLLATE pg\_catalog."default" NOT NULL,

telefone character varying(30) COLLATE pg\_catalog."default" NOT NULL,

data\_contratacao date NOT NULL,

prazo\_contrato character varying(30) COLLATE pg\_catalog."default" NOT NULL,

genero character varying(15) COLLATE pg\_catalog."default" NOT NULL,

cpf character varying(20) COLLATE pg\_catalog."default" NOT NULL,

endereco character varying(30) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT funcionarios\_pkey PRIMARY KEY (id\_funcionario)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.funcionarios

OWNER to postgres;

REVOKE ALL ON TABLE public.funcionarios FROM gerente;

GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE public.funcionarios TO gerente;

GRANT ALL ON TABLE public.funcionarios TO postgres;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.maquina

-- DROP TABLE IF EXISTS public.maquina;

CREATE TABLE IF NOT EXISTS public.maquina

(

"id\_maquina " integer NOT NULL,

id\_cliente integer NOT NULL,

tipo character varying(30) COLLATE pg\_catalog."default" NOT NULL,

status character varying(30) COLLATE pg\_catalog."default" NOT NULL,

hora\_inicio time with time zone NOT NULL,

hora\_fim time with time zone NOT NULL,

CONSTRAINT maquina\_pkey PRIMARY KEY ("id\_maquina "),

CONSTRAINT id\_cliente\_fk FOREIGN KEY (id\_cliente)

REFERENCES public.clientes (id\_cliente) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.maquina

OWNER to postgres;

REVOKE ALL ON TABLE public.maquina FROM estagiario;

REVOKE ALL ON TABLE public.maquina FROM funcionario;

REVOKE ALL ON TABLE public.maquina FROM gerente;

GRANT SELECT ON TABLE public.maquina TO estagiario;

GRANT UPDATE, SELECT, INSERT ON TABLE public.maquina TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.maquina TO gerente;

GRANT ALL ON TABLE public.maquina TO postgres;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.movimento\_financeiro

-- DROP TABLE IF EXISTS public.movimento\_financeiro;

CREATE TABLE IF NOT EXISTS public.movimento\_financeiro

(

id\_movto integer NOT NULL,

descricao\_movto character varying(5) COLLATE pg\_catalog."default" NOT NULL,

data\_movto date NOT NULL,

hora\_movto time with time zone NOT NULL,

CONSTRAINT movimento\_financeiro\_pkey PRIMARY KEY (id\_movto, descricao\_movto)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.movimento\_financeiro

OWNER to postgres;

REVOKE ALL ON TABLE public.movimento\_financeiro FROM gerente;

GRANT DELETE, INSERT, SELECT, UPDATE ON TABLE public.movimento\_financeiro TO gerente;

GRANT ALL ON TABLE public.movimento\_financeiro TO postgres;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.reserva

-- DROP TABLE IF EXISTS public.reserva;

CREATE TABLE IF NOT EXISTS public.reserva

(

id\_reserva integer NOT NULL,

id\_cliente integer NOT NULL,

id\_maquina integer NOT NULL,

data\_reserva date NOT NULL,

CONSTRAINT reserva\_pkey PRIMARY KEY (id\_reserva),

CONSTRAINT id\_cliente\_fk FOREIGN KEY (id\_cliente)

REFERENCES public.clientes (id\_cliente) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_maquina\_fk FOREIGN KEY (id\_maquina)

REFERENCES public.maquina ("id\_maquina ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.reserva

OWNER to postgres;

REVOKE ALL ON TABLE public.reserva FROM estagiario;

REVOKE ALL ON TABLE public.reserva FROM funcionario;

REVOKE ALL ON TABLE public.reserva FROM gerente;

GRANT SELECT ON TABLE public.reserva TO estagiario;

GRANT UPDATE, SELECT, INSERT ON TABLE public.reserva TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.reserva TO gerente;

GRANT ALL ON TABLE public.reserva TO postgres;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.saida

-- DROP TABLE IF EXISTS public.saida;

CREATE TABLE IF NOT EXISTS public.saida

(

id\_saida integer NOT NULL,

data\_saida date NOT NULL,

valor\_saida double precision NOT NULL,

obs text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT saida\_pkey PRIMARY KEY (id\_saida)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.saida

OWNER to postgres;

REVOKE ALL ON TABLE public.saida FROM funcionario;

REVOKE ALL ON TABLE public.saida FROM gerente;

GRANT UPDATE, INSERT, SELECT ON TABLE public.saida TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.saida TO gerente;

GRANT ALL ON TABLE public.saida TO postgres;

-- Index: idx\_gin\_saida

-- DROP INDEX IF EXISTS public.idx\_gin\_saida;

CREATE INDEX IF NOT EXISTS idx\_gin\_saida

ON public.saida USING gin

(obs COLLATE pg\_catalog."default" gin\_trgm\_ops)

TABLESPACE pg\_default;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.servico

-- DROP TABLE IF EXISTS public.servico;

CREATE TABLE IF NOT EXISTS public.servico

(

id\_servico integer NOT NULL,

tipo\_servico character varying(30) COLLATE pg\_catalog."default" NOT NULL,

descricao character varying(30) COLLATE pg\_catalog."default",

produtos character varying(30) COLLATE pg\_catalog."default",

valor double precision,

CONSTRAINT servico\_pkey PRIMARY KEY (id\_servico)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.servico

OWNER to postgres;

REVOKE ALL ON TABLE public.servico FROM estagiario;

REVOKE ALL ON TABLE public.servico FROM funcionario;

REVOKE ALL ON TABLE public.servico FROM gerente;

GRANT SELECT ON TABLE public.servico TO estagiario;

GRANT UPDATE, SELECT, INSERT ON TABLE public.servico TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.servico TO gerente;

GRANT ALL ON TABLE public.servico TO postgres;

-- Index: idx\_servico

-- DROP INDEX IF EXISTS public.idx\_servico;

CREATE INDEX IF NOT EXISTS idx\_servico

ON public.servico USING btree

(tipo\_servico COLLATE pg\_catalog."default" ASC NULLS LAST)

TABLESPACE pg\_default;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Table: public.historico\_uso

-- DROP TABLE IF EXISTS public.historico\_uso;

CREATE TABLE IF NOT EXISTS public.historico\_uso

(

id\_historico integer NOT NULL,

id\_cliente integer NOT NULL,

id\_maquina integer NOT NULL,

id\_reserva integer NOT NULL,

CONSTRAINT historico\_uso\_pkey PRIMARY KEY (id\_historico),

CONSTRAINT id\_cliente\_fk FOREIGN KEY (id\_cliente)

REFERENCES public.clientes (id\_cliente) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_maquina\_fk FOREIGN KEY (id\_maquina)

REFERENCES public.maquina ("id\_maquina ") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID,

CONSTRAINT id\_reserva\_fk FOREIGN KEY (id\_reserva)

REFERENCES public.reserva (id\_reserva) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.historico\_uso

OWNER to postgres;

REVOKE ALL ON TABLE public.historico\_uso FROM estagiario;

REVOKE ALL ON TABLE public.historico\_uso FROM funcionario;

REVOKE ALL ON TABLE public.historico\_uso FROM gerente;

GRANT SELECT ON TABLE public.historico\_uso TO estagiario;

GRANT UPDATE, SELECT, INSERT ON TABLE public.historico\_uso TO funcionario;

GRANT UPDATE, DELETE, INSERT, SELECT ON TABLE public.historico\_uso TO gerente;

GRANT ALL ON TABLE public.historico\_uso TO postgres;

**Script que popula as tabelas do Banco de dados**

**Tabela AVALIAÇÃO**

INSERT INTO Avaliacao

VALUES (1, 2, 2, 10, 'Adorei', '2024-06-01');

INSERT INTO Avaliacao

VALUES (2, 1, 3, 8, 'Bom', '2024-06-19');

INSERT INTO Avaliacao

VALUES (3, 3, 1, 3, 'Ruim', '2024-07-08');

**Tabela CLIENTES**

INSERT INTO Clientes

VALUES (1, 'Ana', 'Ana@gmail.com', '(48) 9999-9999', 'Criciuma', '123', '2024-06-01', 'pessoa');

INSERT INTO Clientes

VALUES (2, 'João', 'João@gmail.com', '(48) 8888- 8888, 'Criciuma', '321', '2024-06-06', 'pessoa');

INSERT INTO Clientes

VALUES (3, 'Moveis Pereira, 'Moveispereira@gmail.com', '(48) 7777- 7777, 'Criciuma', '0123', '2024-06-09', 'empresa');

**Tabela ENTRADA**

INSERT INTO Entrada

VALUES (1, 1, '2024-06-22', 100, 'Credito', '2x');

INSERT INTO Entrada

VALUES (2, 3, '2024-06-10', 1000, 'A vista', '1x');

**Tabela FUNCIONARIOS**

INSERT INTO Funcionarios

VALUES (1, 'Maicon', 'Estagiário', '[Maicon@gmail.com](mailto:Maicon@gmail.com)', '(48) 6666-6666', '2024-01-01', '2024-12-01', 'Masculino', '987', 'Tubarão');

**Tabela MAQUINA**

INSERT INTO Maquina

VALUES (1, 1, 'Secadora', 'Em uso', '13:30', '14:30');

INSERT INTO Maquina

VALUES (2, 1, 'Lavadora', 'Disponivel', '09:00', '10:00');

**Tabela RESERVA**

INSERT INTO Reserva

VALUES (1, 3, 1, '2024-07-01');

**Tabela SAIDA**

INSERT INTO Saida

VALUES (1, '2024-06-18', 1000, 'Urgencia');

**Tabela SERVIÇO**

INSERT INTO Servico

VALUES (1, 'Lavar roupa', 'Roupas escuras', 'Sabão em pó e Sabão liquido', 150);

**Tabela HISTÓRICO USO**

INSERT INTO **historicoUso**

VALUES (1, 3, 2, 1);

**Permissões e usuários**

create group gerente

create group funcionario

create group estagiario

select \* from pg\_group

create user gerente01 password '123456'

create user funcionario01 password '123456'

create user estagiario01 password '123456'

select \* from pg\_user

alter group gerente add user gerente01

alter group atendente add user funcionario01

alter group estagiario add user estagiario01

GRANT SELECT

ON avaliacao,clientes,historico\_uso,maquina,reserva,servico TO estagiario

GRANT SELECT, INSERT, UPDATE

ON avaliacao,clientes,historico\_uso,maquina,reserva,servico,entrada,saida TO funcionario

GRANT SELECT, INSERT, UPDATE, DELETE

ON ALL TABLES IN SCHEMA public TO gerente

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 4)**

**- Qual foi a quantidade de avaliações enviadas pelos clientes no último mês?**

SELECT id\_cliente, COUNT(\*) AS quantidade\_avaliacoes

FROM Avaliacao

WHERE EXTRACT(MONTH FROM data\_avaliacao) = 6

GROUP BY id\_cliente;

**- Qual foi a fatura total do último mês?**

SELECT SUM(valor\_entrada)

FROM Entrada

WHERE EXTRACT(MONTH FROM data\_entrada) = 6;

**- Existe algum funcionário que seu contrato irá vencer neste mês?**

SELECT \*

FROM Funcionarios

WHERE EXTRACT(MONTH FROM prazo\_contrato) = 6

**- Quais máquinas estão com o status disponíveis para o uso?**

SELECT \*

FROM Maquina

WHERE status = 'disponível';