

```
-- DROP SCHEMA public;
```

```
CREATE SCHEMA public AUTHORIZATION u132qjrgfkpaql;
```

```
-- DROP SEQUENCE alimento_id_seq;
```

```
CREATE SEQUENCE alimento_id_seq
```

```
    INCREMENT BY 1
```

```
    MINVALUE 1
```

```
    MAXVALUE 2147483647
```

```
    START 1
```

```
    CACHE 1
```

```
    NO CYCLE;
```

```
-- DROP SEQUENCE categoria_alimento_id_seq;
```

```
CREATE SEQUENCE categoria_alimento_id_seq
```

```
    INCREMENT BY 1
```

```
    MINVALUE 1
```

```
    MAXVALUE 2147483647
```

```
    START 1
```

```
    CACHE 1
```

```
    NO CYCLE;
```

```
-- DROP SEQUENCE doacao_id_doacao_seq;
```

```
CREATE SEQUENCE doacao_id_doacao_seq
```

```
    INCREMENT BY 1
```

```
    MINVALUE 1
```

```
    MAXVALUE 2147483647
```

```
    START 1
```

```
    CACHE 1
```

```
    NO CYCLE;
```

```
-- DROP SEQUENCE doador_id_seq;
```

```
CREATE SEQUENCE doador_id_seq
```

```
INCREMENT BY 1
MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1
NO CYCLE;
-- DROP SEQUENCE endereco_id_seq;

CREATE SEQUENCE endereco_id_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1
NO CYCLE;
-- DROP SEQUENCE entidade_recebedora_id_seq;

CREATE SEQUENCE entidade_recebedora_id_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1
NO CYCLE;
-- DROP SEQUENCE entrega_id_seq;

CREATE SEQUENCE entrega_id_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 2147483647
START 1
CACHE 1
NO CYCLE;
-- DROP SEQUENCE item_pedido_id_seq;
```

```
CREATE SEQUENCE item_pedido_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 2147483647
    START 1
    CACHE 1
    NO CYCLE;
-- DROP SEQUENCE local_coleta_id_seq;
```

```
CREATE SEQUENCE local_coleta_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 2147483647
    START 1
    CACHE 1
    NO CYCLE;
-- DROP SEQUENCE pedido_id_pedido_seq;
```

```
CREATE SEQUENCE pedido_id_pedido_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 2147483647
    START 1
    CACHE 1
    NO CYCLE;
-- DROP SEQUENCE regioao_id_seq;
```

```
CREATE SEQUENCE regioao_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 2147483647
    START 1
    CACHE 1
```

```

        NO CYCLE;
-- DROP SEQUENCE status_pedido_id_seq;

CREATE SEQUENCE status_pedido_id_seq
    INCREMENT BY 1
    MINVALUE 1
    MAXVALUE 2147483647
    START 1
    CACHE 1
    NO CYCLE;-- public.categoria_alimento definição

-- Drop table

-- DROP TABLE categoria_alimento;

CREATE TABLE categoria_alimento (
    id serial4 NOT NULL,
    nome varchar(100) NOT NULL,
    descricao text NULL,
    CONSTRAINT categoria_alimento_nome_key UNIQUE (nome),
    CONSTRAINT categoria_alimento_pkey PRIMARY KEY (id)
);
CREATE INDEX idx_categoria_nome ON public.categoria_alimento USING
btree (nome);

-- Table Triggers

create trigger trigger_atualizar_nome_categoria after
update
    on
        public.categoria_alimento    for each row execute function
atualizar_nome_categoria();

```

```

-- public.doador definição

-- Drop table

-- DROP TABLE doador;

CREATE TABLE doador (
    id serial4 NOT NULL,
    nome varchar(255) NOT NULL,
    email varchar(255) NOT NULL,
    telefone varchar(20) NOT NULL,
    tipo varchar(50) NULL,
    CONSTRAINT doador_email_key UNIQUE (email),
    CONSTRAINT doador_pkey PRIMARY KEY (id),
    CONSTRAINT doador_tipo_check CHECK (((tipo)::text = ANY
((ARRAY['Pessoa Física'::character varying, 'Restaurante'::character varying,
'ONG'::character varying, 'Voluntário'::character varying, 'Voluntario'::character
varying]))::text[])))
);
CREATE INDEX idx_doador_email ON public.doador USING btree (email);

-- Table Triggers

create trigger trigger_atualizar_nome_doador after
update
on
public.doador for each row execute function atualizar_nome_doador();

-- public.regiao definição

-- Drop table

-- DROP TABLE regiao;

```

```

CREATE TABLE regiao (
    id serial4 NOT NULL,
    nome varchar(100) NOT NULL,
    descricao text NULL,
    CONSTRAINT regiao_nome_key UNIQUE (nome),
    CONSTRAINT regiao_pkey PRIMARY KEY (id)
);

CREATE INDEX idx_regiao_nome ON public.regiao USING btree (nome);

-- public.status_pedido definição

-- Drop table

-- DROP TABLE status_pedido;

CREATE TABLE status_pedido (
    id serial4 NOT NULL,
    status varchar(20) NULL,
    CONSTRAINT status_pedido_pkey PRIMARY KEY (id),
    CONSTRAINT status_pedido_status_check CHECK (((status)::text =
ANY ((ARRAY['Pendente'::character_varying, 'Aprovado'::character_varying,
'Rejeitado'::character_varying, 'Entregue'::character_varying])::text[]))),
    CONSTRAINT status_pedido_status_key UNIQUE (status)
);

CREATE INDEX idx_status_pedido ON public.status_pedido USING btree
(status);

-- Table Triggers

create trigger trigger_atualizar_status_pedido after
update
on

```

```
public.status_pedido for each row execute function atualizar_status_pedido();
```

```
-- public.alimento definição
```

```
-- Drop table
```

```
-- DROP TABLE alimento;
```

```
CREATE TABLE alimento (  
    id serial4 NOT NULL,  
    nome varchar(255) NOT NULL,  
    quantidade int4 NOT NULL,  
    data_validade date NULL,  
    tipo varchar(50) NULL,  
    categoria_id int4 NULL,  
    nome_categoria varchar(255) NULL,  
    CONSTRAINT alimento_data_validade_check CHECK ((data_validade  
>= CURRENT_DATE)),  
    CONSTRAINT alimento_pkey PRIMARY KEY (id),  
    CONSTRAINT alimento_quantidade_check CHECK ((quantidade >= 0)),  
    CONSTRAINT alimento_tipo_check CHECK (((tipo)::text = ANY  
(ARRAY[('Frutas'::character varying)::text, ('Legumes'::character varying)::text,  
('Marmitta'::character varying)::text, ('Condimentos'::character varying)::text,  
('Cereais'::character varying)::text, ('Grãos'::character varying)::text, ('Pães'::character  
varying)::text, ('Massas'::character varying)::text, ('Carnes'::character varying)::text,  
('Peixes'::character varying)::text, ('Laticínios'::character varying)::text,  
('Doces'::character varying)::text, ('Sorvetes'::character varying)::text, ('Café'::character  
varying)::text, ('Chás'::character varying)::text, ('Frutos do mar'::character varying)::text,  
('Snacks'::character varying)::text, ('Manteigas'::character varying)::text,  
('Salgados'::character varying)::text, ('Ovos'::character varying)::text]))),  
    CONSTRAINT fk_alimento_categoria FOREIGN KEY (categoria_id)  
REFERENCES categoria_alimento(id) ON DELETE SET NULL  
);
```

```
CREATE INDEX idx_alimento_categoria_id ON public.alimento USING btree
(categoria_id);
```

```
CREATE INDEX idx_alimento_data_validade ON public.alimento USING
btree (data_validade);
```

```
CREATE INDEX idx_alimento_nome ON public.alimento USING btree
(nome);
```

```
CREATE INDEX idx_alimento_quantidade ON public.alimento USING btree
(quantidade);
```

```
-- public.endereco definição
```

```
-- Drop table
```

```
-- DROP TABLE endereco;
```

```
CREATE TABLE endereco (
    id serial4 NOT NULL,
    logradouro varchar(255) NOT NULL,
    numero varchar(20) NOT NULL,
    bairro varchar(100) NOT NULL,
    cidade varchar(100) NOT NULL,
    cep varchar(10) NOT NULL,
    complemento varchar(255) NULL,
    regiao_id int4 NULL,
    CONSTRAINT endereco_pkey PRIMARY KEY (id),
    CONSTRAINT fk_endereco_regiao FOREIGN KEY (regiao_id)
REFERENCES regiao(id) ON DELETE SET NULL
);
CREATE INDEX idx_endereco_cep ON public.endereco USING btree (cep);
```

```
-- public.entidade_recebedora definição
```



```

-- Drop table

-- DROP TABLE entidade_recebedora;

CREATE TABLE entidade_recebedora (
    id serial4 NOT NULL,
    nome varchar(255) NOT NULL,
    cnpj varchar(14) NOT NULL,
    contato varchar(255) NOT NULL,
    status varchar(20) NULL,
    regiao_id int4 NULL,
    CONSTRAINT entidade_recebedora_cnpj_key UNIQUE (cnpj),
    CONSTRAINT entidade_recebedora_pkey PRIMARY KEY (id),
    CONSTRAINT entidade_recebedora_status_check CHECK
(((status)::text = ANY ((ARRAY['ativo'::character_varying, 'inativo'::character_varying])::text[]))),
    CONSTRAINT fk_entidade_recebedora_regiao FOREIGN KEY
(regiao_id) REFERENCES regiao(id) ON DELETE SET NULL
);

CREATE INDEX idx_entidade_cnpj ON public.entidade_recebedora USING
btree (cnpj);

CREATE INDEX idx_entidade_nome ON public.entidade_recebedora USING
btree (nome);

-- Table Triggers

create trigger trigger_atualizar_nome_ong after
update
on
public.entidade_recebedora for each row execute function
atualizar_nome_ong();

-- public.local_coleta definição

```

```

-- Drop table

-- DROP TABLE local_coleta;

CREATE TABLE local_coleta (
    id serial4 NOT NULL,
    endereco_id int4 NULL,
    horario_funcionamento varchar(100) NOT NULL,
    tipo varchar(20) NULL,
    regiao_id int4 NULL,
    CONSTRAINT local_coleta_pkey PRIMARY KEY (id),
    CONSTRAINT local_coleta_tipo_check CHECK (((tipo)::text = ANY
((ARRAY['pedido'::character varying, 'doação'::character varying])::text[]))),
    CONSTRAINT fk_local_coleta_regiao FOREIGN KEY (regiao_id)
REFERENCES regiao(id) ON DELETE SET NULL,
    CONSTRAINT local_coleta_endereco_id_fkey FOREIGN KEY
(endereco_id) REFERENCES endereco(id) ON DELETE CASCADE
);
CREATE INDEX idx_local_coleta_endereco_id ON public.local_coleta
USING btree (endereco_id);
CREATE INDEX idx_local_coleta_horario ON public.local_coleta USING
btree (horario_funcionamento);
CREATE INDEX idx_local_coleta_tipo ON public.local_coleta USING btree
(tipo);

-- public.pedido definição

-- Drop table

-- DROP TABLE pedido;

CREATE TABLE pedido (

```

```

        id_pedido serial4 NOT NULL,
        data_pedido date NOT NULL,
        ong_id int4 NULL,
        status_id int4 NULL,
        nome_ong varchar(255) NULL,
        status_pedido varchar(50) NULL,
        CONSTRAINT pedido_pkey PRIMARY KEY (id_pedido),
        CONSTRAINT fk_pedido_status FOREIGN KEY (status_id)
REFERENCES status_pedido(id) ON DELETE SET NULL,
        CONSTRAINT pedido_ong_id_fkey FOREIGN KEY (ong_id)
REFERENCES entidade_recebedora(id) ON DELETE CASCADE
    );
    CREATE INDEX idx_pedido_data ON public.pedido USING btree
(data_pedido);
    CREATE INDEX idx_pedido_ong_id ON public.pedido USING btree (ong_id);
    CREATE INDEX idx_pedido_status_id ON public.pedido USING btree
(status_id);

```

```
-- public.doacao definição
```

```
-- Drop table
```

```
-- DROP TABLE doacao;
```

```

CREATE TABLE doacao (
    id_doacao serial4 NOT NULL,
    data_doacao date NOT NULL,
    doador_id int4 NULL,
    alimento_id int4 NULL,
    local_coleta_id int4 NULL,
    nome_doador varchar(255) NULL,
    CONSTRAINT doacao_pkey PRIMARY KEY (id_doacao),

```

```

        CONSTRAINT doacao_alimento_id_fkey FOREIGN KEY (alimento_id)
REFERENCES alimento(id) ON DELETE CASCADE,
        CONSTRAINT doacao_doador_id_fkey FOREIGN KEY (doador_id)
REFERENCES doador(id) ON DELETE CASCADE,
        CONSTRAINT doacao_local_coleta_id_fkey FOREIGN KEY
(local_coleta_id) REFERENCES local_coleta(id) ON DELETE CASCADE
    );
    CREATE INDEX idx_doacao_alimento_id ON public.doacao USING btree
(alimento_id);
    CREATE INDEX idx_doacao_data ON public.doacao USING btree
(data_doacao);
    CREATE INDEX idx_doacao_doador_id ON public.doacao USING btree
(doador_id);
    CREATE INDEX idx_doacao_local_coleta_id ON public.doacao USING btree
(local_coleta_id);

```

```

-- public.entrega definição

```

```

-- Drop table

```

```

-- DROP TABLE entrega;

```

```

CREATE TABLE entrega (
    id serial4 NOT NULL,
    data_entrega date NOT NULL,
    pedido_id int4 NULL,
    CONSTRAINT entrega_pkey PRIMARY KEY (id),
    CONSTRAINT entrega_pedido_id_fkey FOREIGN KEY (pedido_id)
REFERENCES pedido(id_pedido) ON DELETE CASCADE
);
    CREATE INDEX idx_entrega_data ON public.entrega USING btree
(data_entrega);

```

```
CREATE INDEX idx_entrega_pedido_id ON public.entrega USING btree
(pedido_id);
```

```
-- public.item_pedido definição
```

```
-- Drop table
```

```
-- DROP TABLE item_pedido;
```

```
CREATE TABLE item_pedido (
    id serial4 NOT NULL,
    pedido_id int4 NULL,
    alimento_id int4 NULL,
    quantidade int4 NULL,
    CONSTRAINT item_pedido_pkey PRIMARY KEY (id),
    CONSTRAINT item_pedido_quantidade_check CHECK ((quantidade >
0)),
    CONSTRAINT item_pedido_alimento_id_fkey FOREIGN KEY
(alimento_id) REFERENCES alimento(id),
    CONSTRAINT item_pedido_pedido_id_fkey FOREIGN KEY
(pedido_id) REFERENCES pedido(id_pedido) ON DELETE CASCADE
);
CREATE INDEX idx_item_pedido_alimento_id ON public.item_pedido
USING btree (alimento_id);
CREATE INDEX idx_item_pedido_pedido_id ON public.item_pedido USING
btree (pedido_id);
CREATE INDEX idx_item_pedido_quantidade ON public.item_pedido USING
btree (quantidade);
```

```
-- public.pg_stat_statements fonte
```

```
CREATE OR REPLACE VIEW pg_stat_statements
```

```
AS SELECT userid,  
    dbid,  
    toplevel,  
    queryid,  
    query,  
    plans,  
    total_plan_time,  
    min_plan_time,  
    max_plan_time,  
    mean_plan_time,  
    stddev_plan_time,  
    calls,  
    total_exec_time,  
    min_exec_time,  
    max_exec_time,  
    mean_exec_time,  
    stddev_exec_time,  
    rows,  
    shared_blks_hit,  
    shared_blks_read,  
    shared_blks_dirtied,  
    shared_blks_written,  
    local_blks_hit,  
    local_blks_read,  
    local_blks_dirtied,  
    local_blks_written,  
    temp_blks_read,  
    temp_blks_written,  
    blk_read_time,  
    blk_write_time,  
    temp_blk_read_time,  
    temp_blk_write_time,  
    wal_records,  
    wal_fpi,
```

```
wal_bytes,  
jit_functions,  
jit_generation_time,  
jit_inlining_count,  
jit_inlining_time,  
jit_optimization_count,  
jit_optimization_time,  
jit_emission_count,  
jit_emission_time
```

```
FROM pg_stat_statements(true) pg_stat_statements(userid, dbid, toplevel,  
queryid, query, plans, total_plan_time, min_plan_time, max_plan_time,  
mean_plan_time, stddev_plan_time, calls, total_exec_time, min_exec_time,  
max_exec_time, mean_exec_time, stddev_exec_time, rows, shared_blks_hit,  
shared_blks_read, shared_blks_dirtied, shared_blks_written, local_blks_hit,  
local_blks_read, local_blks_dirtied, local_blks_written, temp_blks_read,  
temp_blks_written, blk_read_time, blk_write_time, temp_blk_read_time,  
temp_blk_write_time, wal_records, wal_fpi, wal_bytes, jit_functions,  
jit_generation_time, jit_inlining_count, jit_inlining_time, jit_optimization_count,  
jit_optimization_time, jit_emission_count, jit_emission_time);
```

```
-- public.pg_stat_statements_info fonte
```

```
CREATE OR REPLACE VIEW pg_stat_statements_info  
AS SELECT dealloc,  
stats_reset  
FROM pg_stat_statements_info() pg_stat_statements_info(dealloc,  
stats_reset);
```

```
-- DROP FUNCTION public.atualizar_nome_categoria();
```

```
CREATE OR REPLACE FUNCTION public.atualizar_nome_categoria()
```

```

RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN
    UPDATE public.alimento
    SET nome_categoria = NEW.nome
    WHERE categoria_id = NEW.id;
    RETURN NEW;
END;
$function$
;

-- DROP FUNCTION public.atualizar_nome_doador();

CREATE OR REPLACE FUNCTION public.atualizar_nome_doador()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN
    UPDATE public.doacao
    SET nome_doador = NEW.nome
    WHERE doador_id = NEW.id;
    RETURN NEW;
END;
$function$
;

-- DROP FUNCTION public.atualizar_nome_ong();

CREATE OR REPLACE FUNCTION public.atualizar_nome_ong()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN

```



```

UPDATE public.pedido
SET nome_ong = NEW.nome
WHERE ong_id = NEW.id;
RETURN NEW;
END;
$function$
;

```

```

-- DROP FUNCTION public.atualizar_status_pedido();

```

```

CREATE OR REPLACE FUNCTION public.atualizar_status_pedido()
RETURNS trigger
LANGUAGE plpgsql
AS $function$
BEGIN
    UPDATE public.pedido
    SET status_pedido = NEW.status
    WHERE status_id = NEW.id;
    RETURN NEW;
END;
$function$
;

```

```

-- DROP FUNCTION public.pg_stat_statements(in bool, out oid, out oid, out
bool, out int8, out text, out int8, out float8, out float8, out float8, out float8,
out int8, out float8, out float8, out float8, out float8, out float8, out int8, out int8, out int8,
out int8, out int8, out int8, out int8, out int8, out int8, out int8, out float8, out
float8, out float8, out float8, out int8, out int8, out numeric, out int8, out float8, out int8,
out float8, out int8, out float8, out int8, out float8);

```

```

CREATE OR REPLACE FUNCTION public.pg_stat_statements(showtext
boolean, OUT userid oid, OUT dbid oid, OUT toplevel boolean, OUT queryid bigint,
OUT query text, OUT plans bigint, OUT total_plan_time double precision, OUT
min_plan_time double precision, OUT max_plan_time double precision, OUT

```

mean\_plan\_time double precision, OUT stddev\_plan\_time double precision, OUT calls  
bigint, OUT total\_exec\_time double precision, OUT min\_exec\_time double precision,  
OUT max\_exec\_time double precision, OUT mean\_exec\_time double precision, OUT  
stddev\_exec\_time double precision, OUT rows bigint, OUT shared\_blks\_hit bigint, OUT  
shared\_blks\_read bigint, OUT shared\_blks\_dirtied bigint, OUT shared\_blks\_written  
bigint, OUT local\_blks\_hit bigint, OUT local\_blks\_read bigint, OUT local\_blks\_dirtied  
bigint, OUT local\_blks\_written bigint, OUT temp\_blks\_read bigint, OUT  
temp\_blks\_written bigint, OUT blk\_read\_time double precision, OUT blk\_write\_time  
double precision, OUT temp\_blk\_read\_time double precision, OUT  
temp\_blk\_write\_time double precision, OUT wal\_records bigint, OUT wal\_fpi bigint,  
OUT wal\_bytes numeric, OUT jit\_functions bigint, OUT jit\_generation\_time double  
precision, OUT jit\_inlining\_count bigint, OUT jit\_inlining\_time double precision, OUT  
jit\_optimization\_count bigint, OUT jit\_optimization\_time double precision, OUT  
jit\_emission\_count bigint, OUT jit\_emission\_time double precision)

RETURNS SETOF record

LANGUAGE c

PARALLEL SAFE STRICT

AS '\$libdir/pg\_stat\_statements', \$function\$pg\_stat\_statements\_1\_10\$function\$  
;

-- DROP FUNCTION public.pg\_stat\_statements\_info(out int8, out  
timestampz);

CREATE OR REPLACE FUNCTION public.pg\_stat\_statements\_info(OUT  
dealloc bigint, OUT stats\_reset timestamp with time zone)

RETURNS record

LANGUAGE c

PARALLEL SAFE STRICT

AS '\$libdir/pg\_stat\_statements', \$function\$pg\_stat\_statements\_info\$function\$  
;

-- DROP FUNCTION public.pg\_stat\_statements\_reset(oid, oid, int8);

```
CREATE OR REPLACE FUNCTION public.pg_stat_statements_reset(userid
oid DEFAULT 0, dbid oid DEFAULT 0, queryid bigint DEFAULT 0)
    RETURNS void
    LANGUAGE c
    PARALLEL SAFE STRICT
    AS                                '$libdir/pg_stat_statements',
    $function$pg_stat_statements_reset_1_7$function$
    ;
```