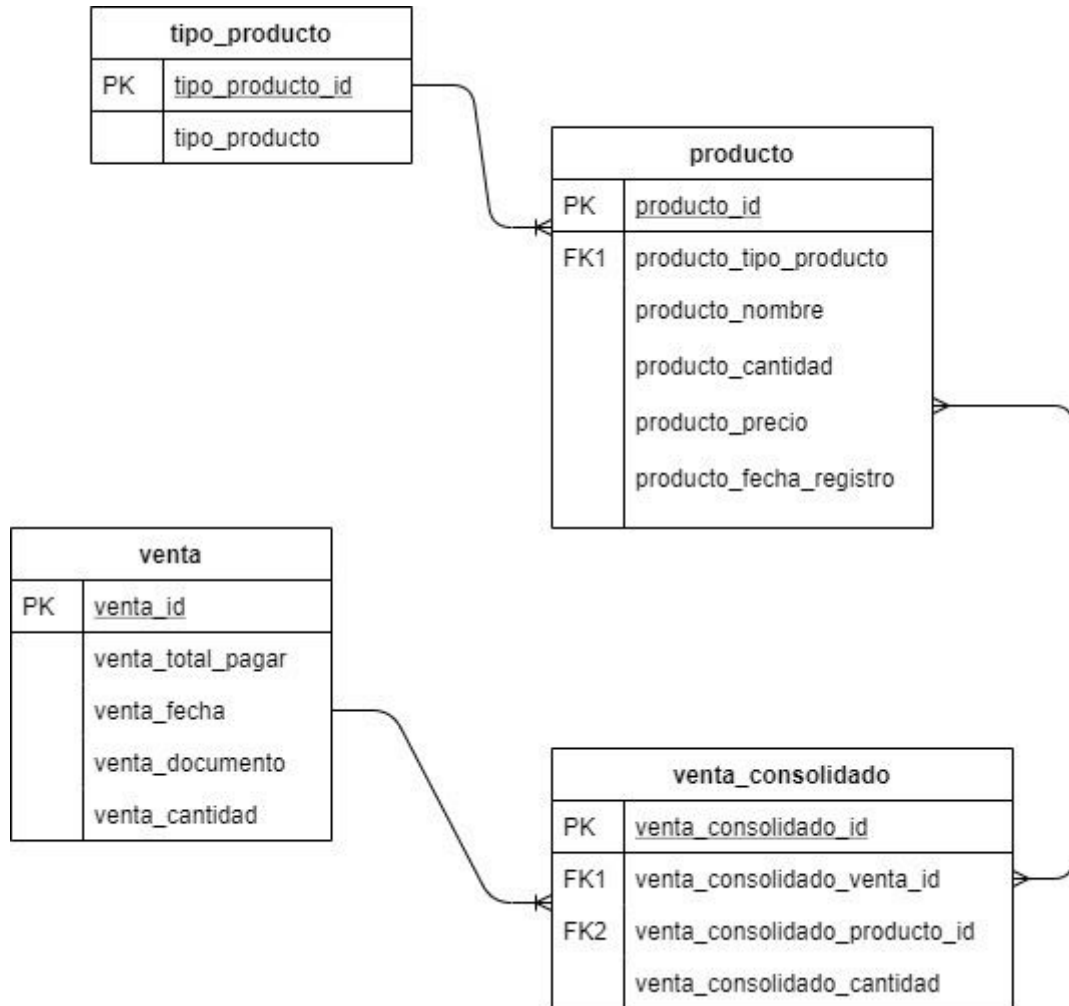


BASE DE DATOS

La base de datos usada para la prueba es una base de datos sql con postgresql.

1. Diagrama entidad relación:



2. Procedimientos en sql para crear la base de datos:

-- Database: cidenet

-- DROP DATABASE cidenet;

```
CREATE DATABASE cidenet
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Spanish_Colombia.1252'
  LC_CTYPE = 'Spanish_Colombia.1252'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;
```

3. Procedimiento sql para crear el schema "**kardex**":

-- SCHEMA: kardex

-- DROP SCHEMA kardex ;

```
CREATE SCHEMA kardex
  AUTHORIZATION postgres;
```

4. Procedimientos sql para crear las secuencias:

```
-----
CREATE SEQUENCE kardex.producto_sec
  INCREMENT 1
  START 1
  MINVALUE 1
  MAXVALUE 9999999999
  CACHE 1;
```

```
ALTER SEQUENCE kardex.producto_sec
  OWNER TO postgres;
```

```
-----
CREATE SEQUENCE kardex.producto_sec
  INCREMENT 1
  START 10
  MINVALUE 1
  MAXVALUE 9999999999
  CACHE 1;
```

```
ALTER SEQUENCE kardex.producto_sec
  OWNER TO postgres;
```

```
CREATE SEQUENCE kardex.venta_consolidado_sec
INCREMENT 1
START 1
MINVALUE 1
MAXVALUE 9999999999
CACHE 1;
```

```
ALTER SEQUENCE kardex.venta_consolidado_sec
OWNER TO postgres;
```

```
CREATE SEQUENCE kardex.venta_sec
INCREMENT 1
START 10
MINVALUE 1
MAXVALUE 999999999999999
CACHE 1;
```

```
ALTER SEQUENCE kardex.venta_sec
OWNER TO postgres;
```

5. Procedimiento sql para crear las tablas:

a. **Tipo producto:**

```
CREATE TABLE kardex.tipo_producto
(
    tipo_producto_id numeric(1000,0) NOT NULL,
    tipo_producto character varying(50) COLLATE pg_catalog."default",
    tipo_producto_fecha_creacion time without time zone,
    CONSTRAINT tipo_producto_pkey PRIMARY KEY (tipo_producto_id)
)

TABLESPACE pg_default;
```

```
ALTER TABLE kardex.tipo_producto
OWNER to postgres;
```

b. Producto:

```
CREATE TABLE kardex.producto
(
    producto_id numeric(1000,0) NOT NULL,
    producto_nombre character varying(100) COLLATE pg_catalog."default",
    producto_cantidad numeric(1000,0),
    producto_precio numeric(100,0),
    producto_tipo_producto numeric(100,0),
    producto_fecha_registro timestamp without time zone,
    CONSTRAINT producto_id PRIMARY KEY (producto_id),
    CONSTRAINT producto_producto_tipo_producto_fkey FOREIGN KEY
(producto_tipo_producto)
    REFERENCES kardex.tipo_producto (tipo_producto_id) MATCH
SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE kardex.producto
    OWNER to postgres;
```

c. Venta:

```
CREATE TABLE kardex.venta
(
    venta_id numeric(1000,0) NOT NULL,
    venta_total_pagar numeric(1000,0),
    venta_fecha timestamp without time zone,
    venta_documento character varying(15) COLLATE pg_catalog."default",
    venta_cantidad numeric(10,0),
    CONSTRAINT venta_id PRIMARY KEY (venta_id)
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE kardex.venta
    OWNER to postgres;
```

d. venta consolidado:

```
CREATE TABLE kardex.venta_consolidado
(
    venta_consolidado_id numeric(1000,0) NOT NULL,
    venta_consolidado_venta_id numeric(1000,0),
    venta_consolidado_producto_id numeric(1000,0),
    venta_consolidado_cantidad numeric(1000,0),
    CONSTRAINT venta_consolidado_pkey PRIMARY KEY (venta_consolidado_id),
    CONSTRAINT venta_consolidado_producto_id FOREIGN KEY
(venta_consolidado_producto_id)
    REFERENCES kardex.producto (producto_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT venta_consolidado_venta_id FOREIGN KEY
(venta_consolidado_venta_id)
    REFERENCES kardex.venta (venta_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE kardex.venta_consolidado
    OWNER to postgres;
```