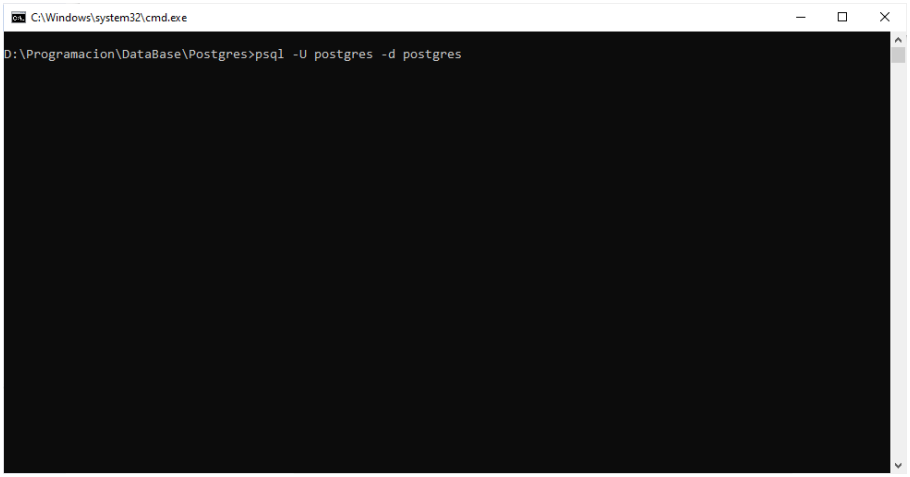
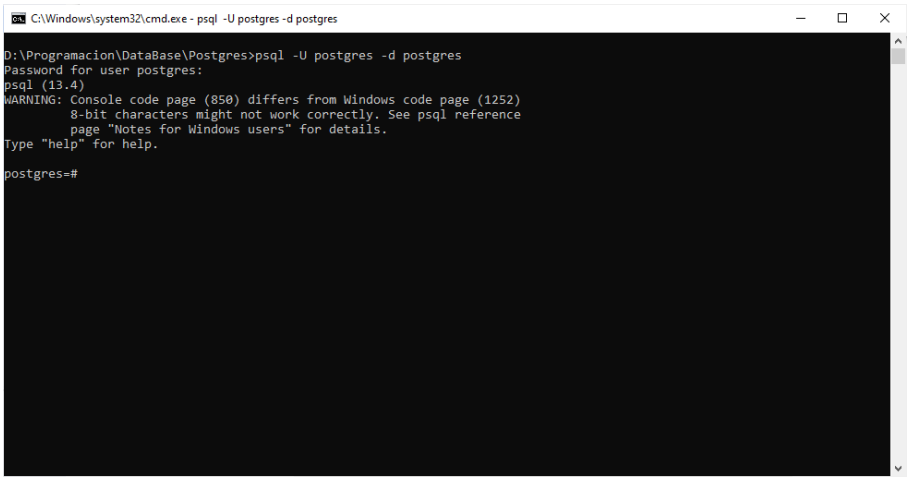
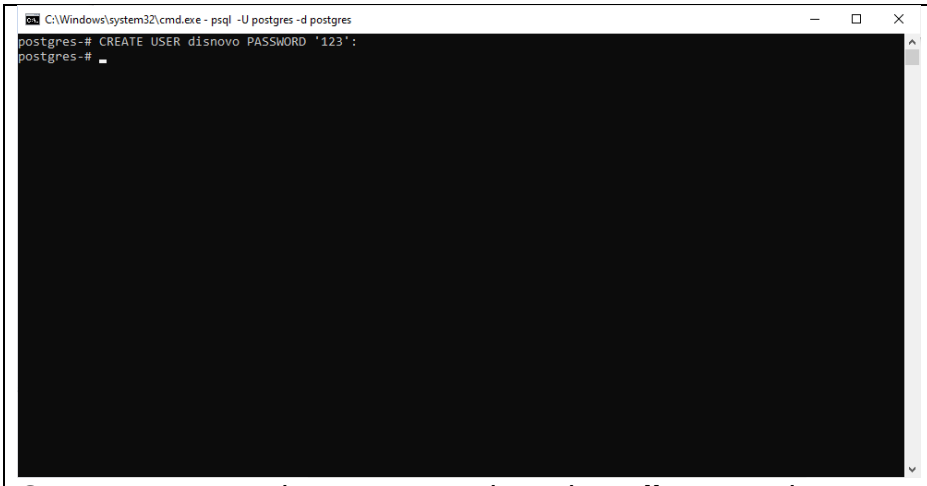
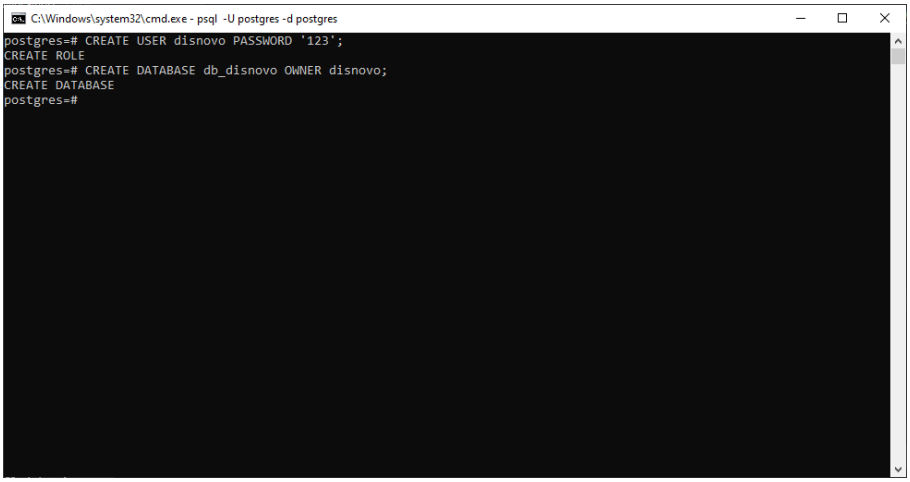
PROCESO DE EJECUCIÓN

OPERACIONES / PASOS /SUBPASOS	SEGURIDAD / MEDIO AMBIENTE / NORMAS -ESTANDARES
Preparación del campo de trabajo	
*Asumiendo que ya tenemos instalado **Postgres** en nuestro OS, en su defecto en un contenedor de Docker, seguiremos los siguientes pasos:*  Nos ubicamos dentro de nuestra terminal en la ubicación donde trabajaremos, en mi caso mi directorio será: **D:\Programacion\DataBase\Postgres>** Dentro de esta abrimos Postgres: **psql -U postgres -d postgres** donde nuestro usuario por defecto es **postgres** y nuestra base de datos por defecto también lleva el mismo nombre.  Ingresamos con nuestra contraseña	**ISO 14001: Gestión Ambiental**



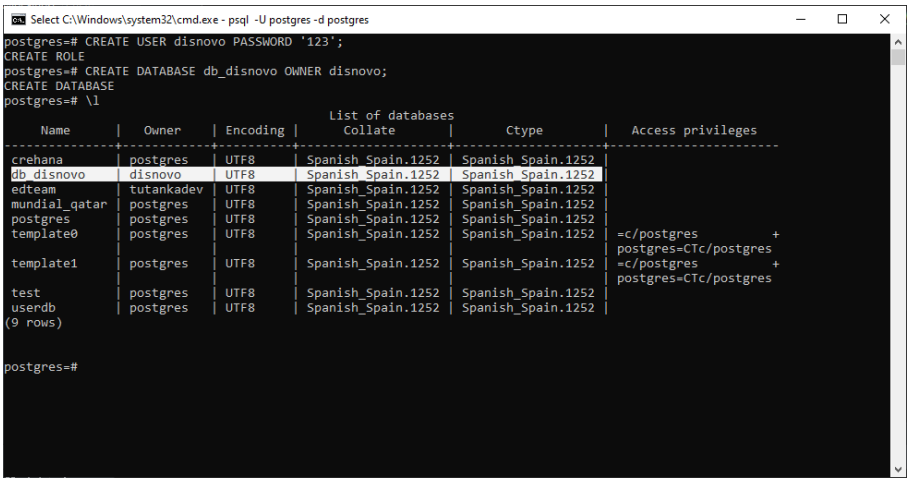
Creamos un usuario nuevo con el nombre: **disnovo** y la contraseña: '**123**'.

CREATE USER disnovo PASSWORD '123';



Creamos una base de datos con el nombre db_disnovo y le asignamos al usuario disnovo.

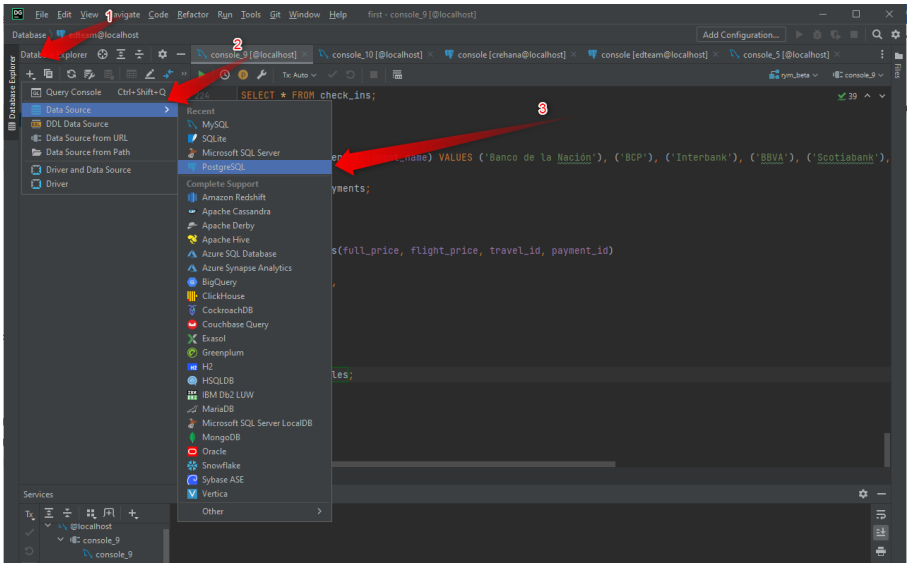
CREATE DATABASE db_disnovo OWNER disnovo;



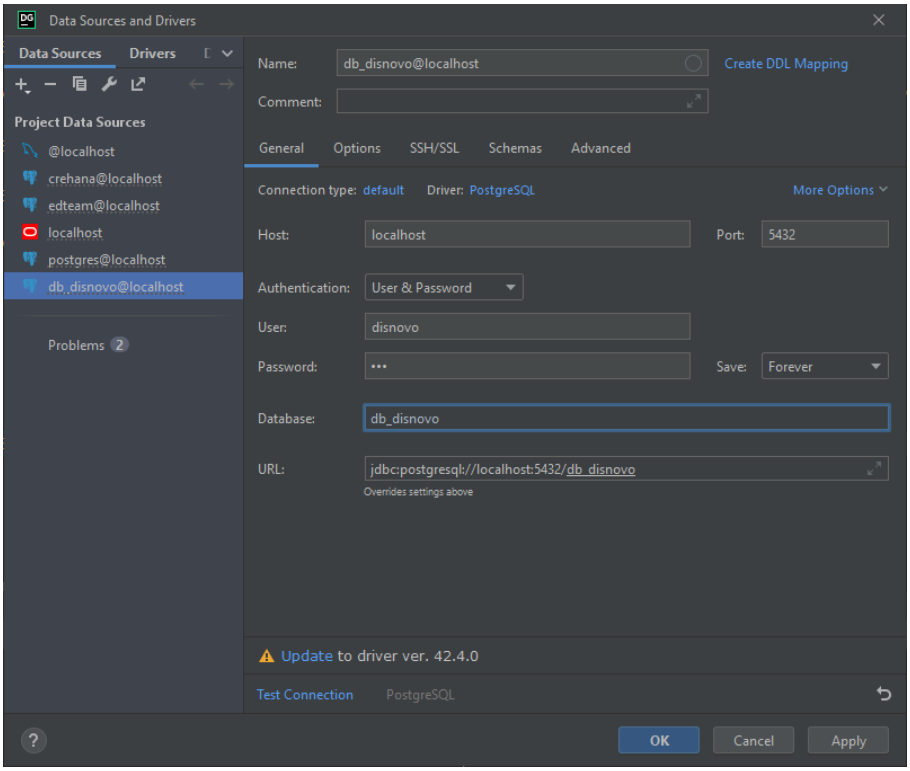
Ya se ubica dentro de Postgres.

- *Para cuestiones practicas y mejor rendimiento usaremos el software **DataGrip** para acelerar la creación de nuestros scripts para nuestra base de datos.*

Lo primero que hacemos es conectar nuestra base de datos postgres con el usuario disnovo en DataGrip



Agregamos una nueva base de datos.



Rellenamos los campos requeridos con la información de nuestra base de datos recién creada. Datagrip nos ubica dentro de nuestra base de datos **db_disnovo** con el usuario **disnovo**, y ya podemos empezar a trabajar

CREACION DE TABLAS

Empezamos con la creación de nuestras tablas de acuerdo a lo planteado:

CONTACT	Almacena los contactos de un usuario	
	id_contact	integer
	nickname_owner	varchar
	business_name	varchar
	tradename	varchar
	firstnames	varchar
	lastnames	varchar
	type_user	varchar
	description	varchar
	observations	varchar
	objective	varchar
	creation_date	data/time
	state	boolean
	erased	boolean

Tabal Contact

Para la creación de esta tabla usamos el siguiente script:

```
CREATE TABLE contacts(
    contact_id SERIAL PRIMARY KEY,
    nickname_owner VARCHAR(64) UNIQUE NOT NULL,
    busines_name VARCHAR(100) NOT NULL,
    trade_name VARCHAR(100) NOT NULL,
    firstname VARCHAR(100) NOT NULL,
    lastname VARCHAR(100) NOT NULL,
    type_user VARCHAR(100) NOT NULL,
    description VARCHAR(255),
    observation VARCHAR(255),
    objetive VARCHAR(100),
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    state BOOLEAN DEFAULT true,
    erased BOOLEAN DEFAULT false
);
```

PRODUCT	Almacena los productos de un usuario	
	id_product	INTEGER
	nickname_owner	varchar
	product_name	varchar
	product_title	varchar
	description	varchar
	observations	varchar
	stock_min	numeric
	stock_max	numeric
	price	numeric
Viene de product_type	id_product_type	integer
	nickname_owner_type	varchar
	nickname_creator	varchar
	creation_date	date/time
	state	boolean
	erased	boolean

Tabla Product

```
CREATE TABLE products(
    product_id SERIAL PRIMARY KEY,
    nickname_owner VARCHAR(64) UNIQUE,
    product_name VARCHAR(100) NOT NULL,
    product_title VARCHAR(100) NOT NULL,
    description VARCHAR(255),
    observation VARCHAR(255),
    stock_min NUMERIC NOT NULL,
    stock_max NUMERIC NOT NULL,
    price NUMERIC NOT NULL,
    product_type_id INT NOT NULL,
    nickname_owner_type VARCHAR(100) NOT NULL,
    nickname_creator VARCHAR(100) NOT NULL,
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    state BOOLEAN DEFAULT true,
    erased BOOLEAN DEFAULT false
);
```

TRANSACTION_TYPE	Almacena los tipos de transacción creados por el usuario	
	id_transation_type	integer
	nickname_owner	varchar
	transaction_type_name	varchar
	transaction_type_title	varchar
	issubtype	boolean
	description	varchar
	id_external	varchar
	nickname_creator	varchar
	creation_date	

Tabla Transaction Type

```
CREATE TABLE transaction_type(
    transaction_type_id SERIAL PRIMARY KEY,
    nickname_owner VARCHAR(100) UNIQUE,
    transaction_type_name VARCHAR(100) NOT NULL,
    transaction_type_title VARCHAR(100) NOT NULL,
    is_sub_type BOOLEAN NOT NULL,
    description VARCHAR(100),
    external_id INT NOT NULL,
    nickname_creator VARCHAR(100) NOT NULL,
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

TRANSACTION	Almacena las transacciones creadas por un usuario	
	id_transaction	integer
	nickname_owner	varchar
	transaction_name	varchar
	transaction_title	varchar
Viene de TRANSACTION_TYPE	id_transaction_type	integer
	nickname_owner_type	varchar
Viene de CONTACT	id_contacto	integer
	nickname_owner_contact	varchar
	start_date_full	date/time
	start_date	date
	start_year	smallint
	start_month	smallint
	start_week	smallint
	start_day	smallint
	start_time	time
	transaction_probability	numeric
	transaction_prevalue	numeric
	transaction_discount	numeric
	transaction_charge	numeric
	transaction_value	numeric
	transaction_taxpercentage	numeric
	transaction_tax	numeric
	transaction_total	numeric
	transaction_balance	numeric
	id_external	varchar
	commentary	varchar
	description	varchar
	observation	varchar
	objective	varchar
	objective_completed	boolean
	nickname_creator	varchar
	creation_date	date/time
	state	boolean

Tabla Transaction

```
CREATE TABLE transaction(
    transaction_id SERIAL PRIMARY KEY,
    nickname_owner VARCHAR(100) UNIQUE,
    transaction_name VARCHAR(100) NOT NULL,
    transaction_title VARCHAR(100) NOT NULL,
    transaction_type_id INT NOT NULL,
    nickname_owner_type VARCHAR(64) NOT NULL,
    contact_id INT NOT NULL,
    nickname_owner_contact VARCHAR(64) NOT NULL,
    start_date_full TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    start_date VARCHAR(32) NOT NULL,
    start_year SMALLINT NOT NULL,
    start_month SMALLINT NOT NULL,
    start_week SMALLINT NOT NULL,
    start_day SMALLINT NOT NULL,
    start_time SMALLINT NOT NULL,
    transaction_probability NUMERIC NOT NULL,
    transaction_prevalue NUMERIC NOT NULL,
    transaction_discount NUMERIC NOT NULL,
    transaction_charge NUMERIC NOT NULL,
    transaction_value NUMERIC NOT NULL,
    transaction_taxpercentage NUMERIC NOT NULL,
    transaction_tax NUMERIC NOT NULL,
    transaction_total NUMERIC NOT NULL,
    transaction_balance NUMERIC NOT NULL,
    external_id INT NOT NULL,
```

```
    commentary VARCHAR(255) NOT NULL,
    description VARCHAR(255) NOT NULL,
    observation VARCHAR(255) NOT NULL,
    objective VARCHAR(100) NOT NULL,
    objective_completed VARCHAR(255) NOT NULL,
    nickname_creator VARCHAR(64) NOT NULL,
    creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    state BOOLEAN DEFAULT true,
    erased BOOLEAN DEFAULT false

);
```

TRANSACTION_DETAIL	Detalles de las transacciones , es el producto que se comercializa en la transacción	
Viene de TRANSACTION	id_transaction	integer
	nickname_owner_transaction	varchar
VIENE DE PRODUCT	id_product	integer
	nickname_owner_product	varchar
	id_item	smallint
	price	numeric
	quantity	numeric
	transactiondetail_discount	numeric
	transactiondetail_discountpercentage	numeric
	transactiondetail_charge	numeric
	transactiondetail_chargepercentage	numeric
	transactiondetail_taxpercentage	numeric
	transactiondetail_tax	numeric
	transactiondetail_balance	numeric
	description	varchar
	commentary	varchar
	state	boolean
	erased	boolean

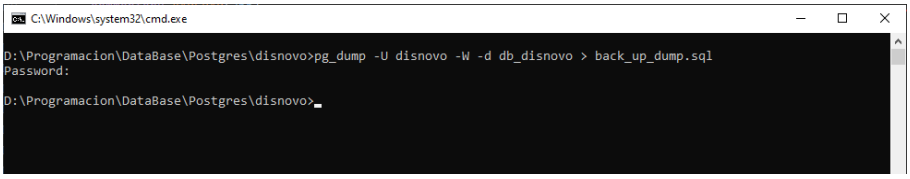
Tabla Transaction Detail

```
CREATE TABLE transaction_detail(
    transaction_dateil_id SERIAL PRIMARY KEY,
    nickname_owner_transaction VARCHAR(64) UNIQUE,
    id_product INT NOT NULL,
    nickname_owner_product VARCHAR(64) UNIQUE,
    id_item SMALLINT UNIQUE,
    price NUMERIC NOT NULL,
    quantity NUMERIC NOT NULL,
    transactiondetail_discount NUMERIC NOT NULL,
    transactiondetail_discountpercentage NUMERIC NOT
NULL,
    transactiondetail_charge NUMERIC NOT NULL,
    transactiondetail_chargepercentage NUMERIC NOT NULL,
    transactiondetail_taxpercentage NUMERIC NOT NULL,
    transactiondetail_tax NUMERIC NOT NULL,
    transactiondetail_balance NUMERIC NOT NULL,
    description VARCHAR(255),
    commentary VARCHAR(255),
    state BOOLEAN DEFAULT true,
    erased BOOLEAN DEFAULT false
);
```

PRODUCT_TYPE	almacena las categorías aplicables a los productos y/ servicios creados por un usuario	
	id_product_type	integer
	nickname_owner	varchar
	product_type_name	varchar
	product_type_title	
	id_external	
	nickname_creator	varchar
	creation_date	
	ispublic	boolean
	state	boolean
	erased	boolean

Tabla Product Type

```
CREATE TABLE product_type(
    product_type_id SERIAL PRIMARY KEY,
    nickname_owner VARCHAR(64) UNIQUE,
    product_type_name VARCHAR(64) NOT NULL,
    product_type_title  VARCHAR(64) NOT NULL,
   external_id VARCHAR(64) UNIQUE ,
   nickname_creator VARCHAR(64) NOT NULL,
   creation_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   is_public BOOLEAN NOT NULL,
   state BOOLEAN NOT NULL,
   erased BOOLEAN NOT NULL
);
```



Hacemos el primer BackUp y guardamos el scriot en el archivo **back_up_dump.sql**

pg_dump -U disnovo -W -d db_disnovo > back_up_dump.sql

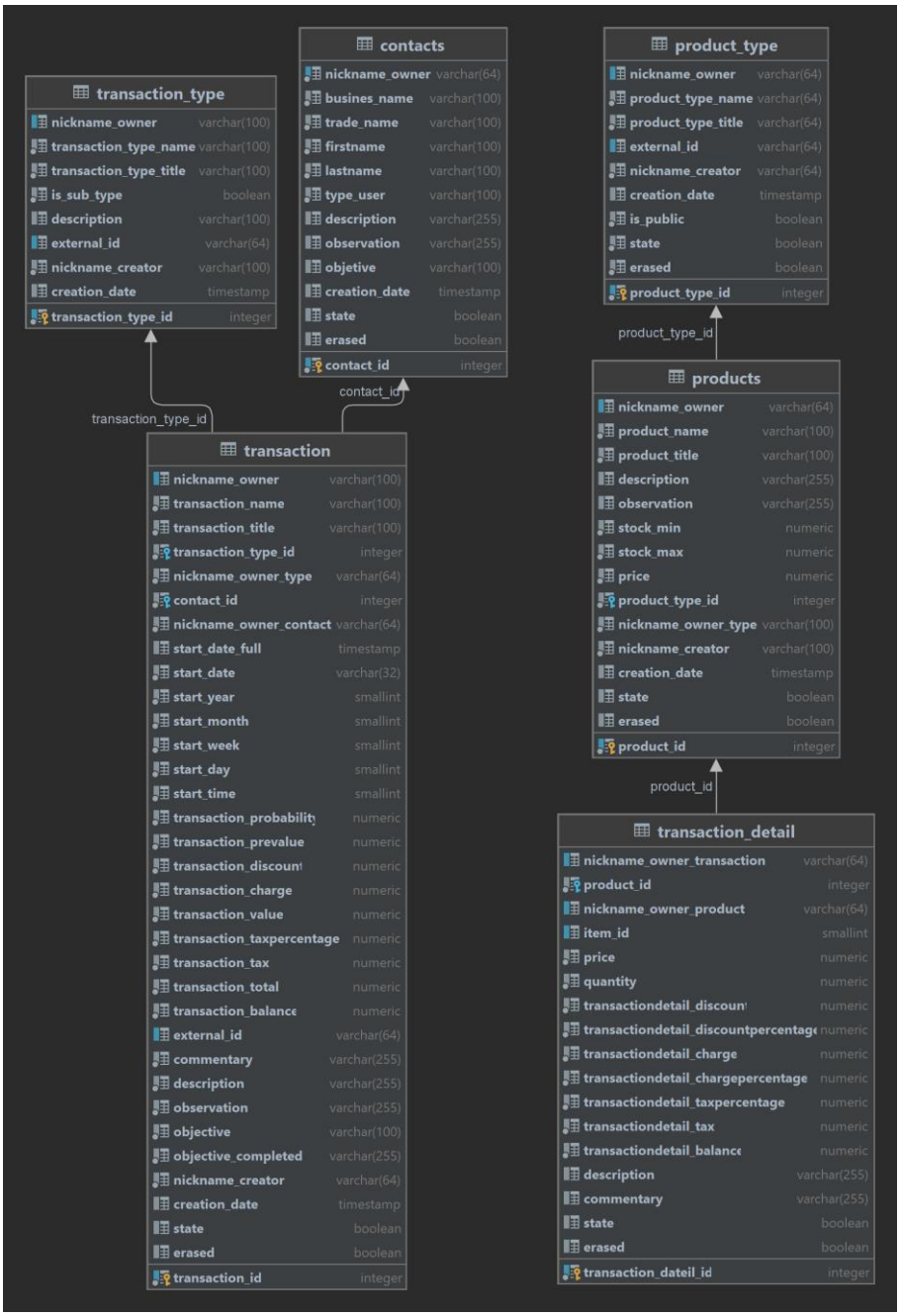
Añadimos las llaves foraneas:

```
ALTER TABLE products ADD CONSTRAINT fk_pti
    FOREIGN KEY (product_type_id)
    REFERENCES product_type(product_type_id);

ALTER TABLE transaction ADD CONSTRAINT fk_tti
    FOREIGN KEY (transaction_type_id)
    REFERENCES transaction_type(transaction_type_id);

ALTER TABLE transaction ADD CONSTRAINT fk_ci
    FOREIGN KEY (contact_id)
    REFERENCES contacts(contact_id);

ALTER TABLE transaction_detail ADD CONSTRAINT fk_pi
    FOREIGN KEY (product_id)
    REFERENCES products(product_id);
```

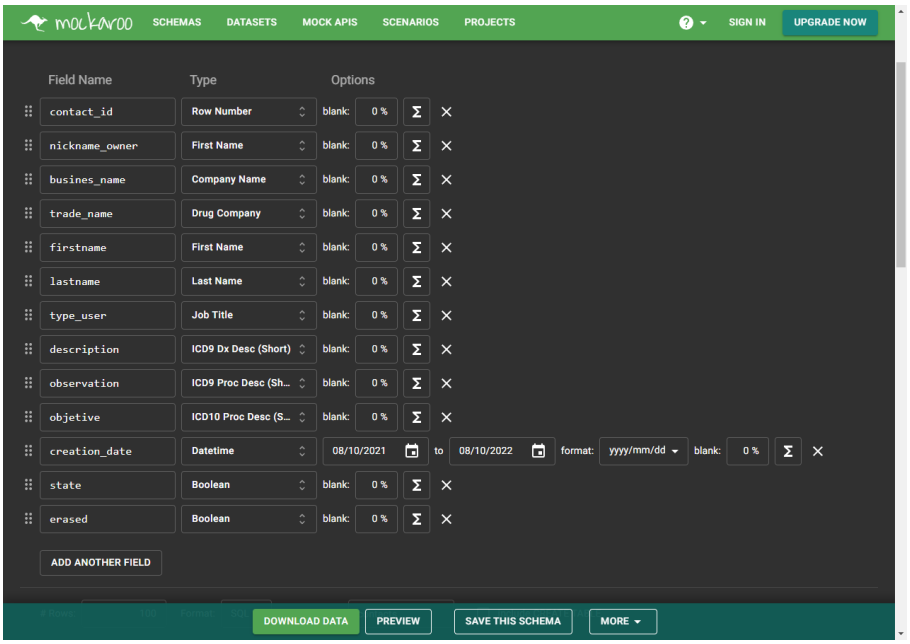


Asi queda nuestro diseño de tablas

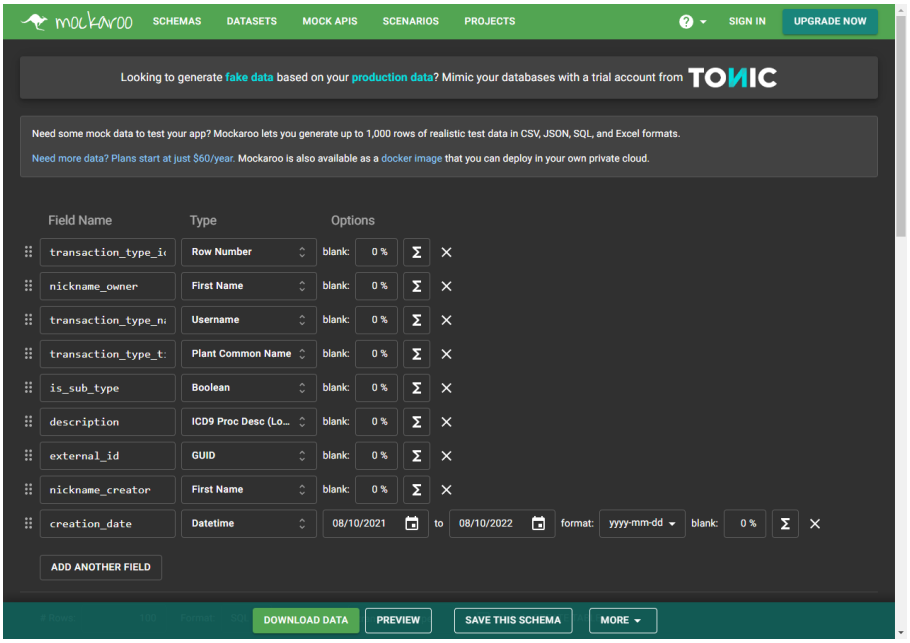
Est. Kenyi Julberth Hancco Quispe

Insertamos datos a nuestras tablas

Usaremos la herramienta que nos brinda mockaroo en su sitio web https://www.mockaroo.com/



Esta herramienta nos genera daos para nuestras tablas de acuerdo con lo que se le indica.



De esta manera generamos datos aleatorios para cada tabla