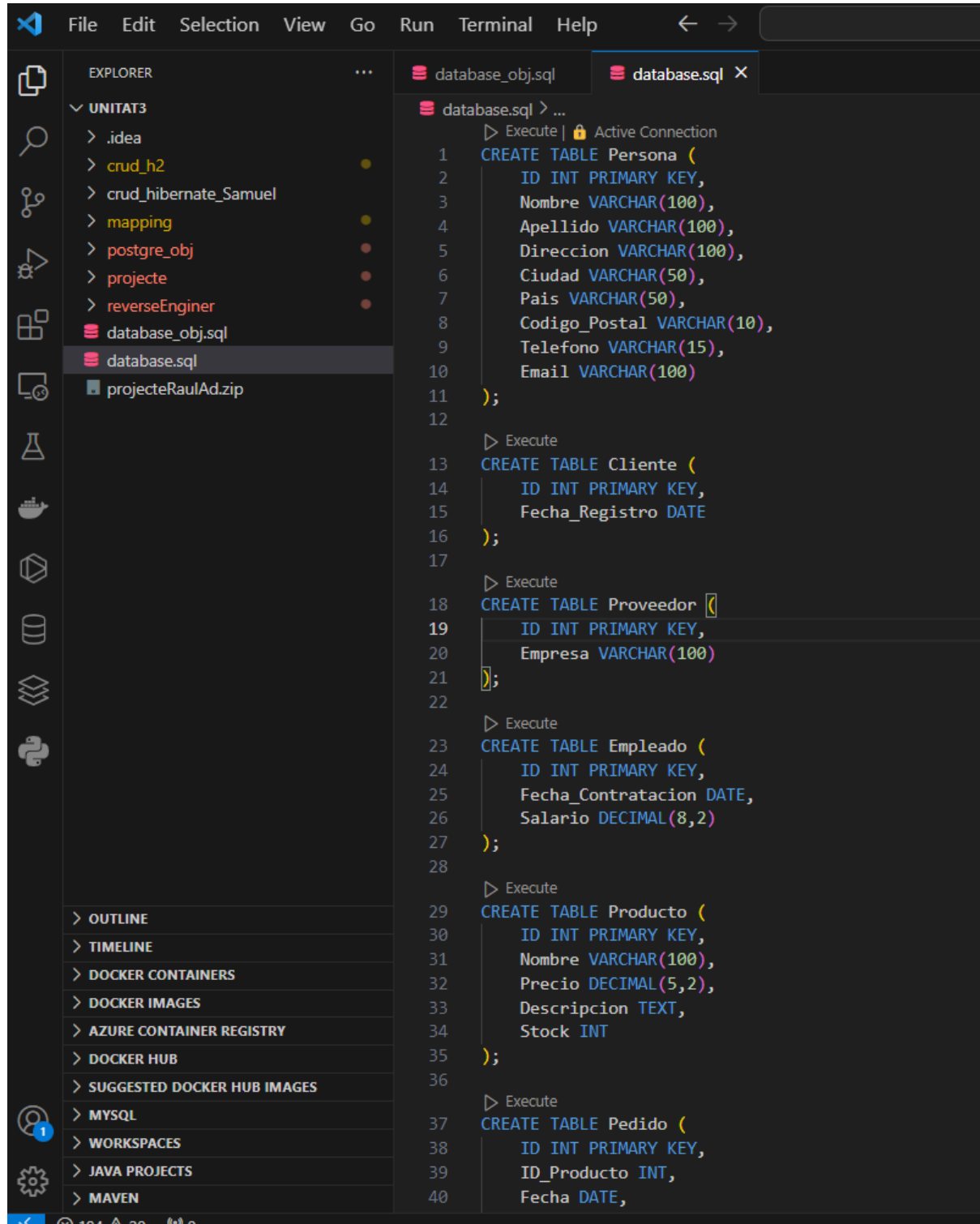


# Practica relational:object-relational

|   |   |
|---|---|
| Imagen del script .sql:   | 2 |
| Imagen del script .sql para postgre aplicando el uso de objeto-relacional | 3 |
| Cambios realizados:   | 3 |
| RESUMEN:  | 4 |

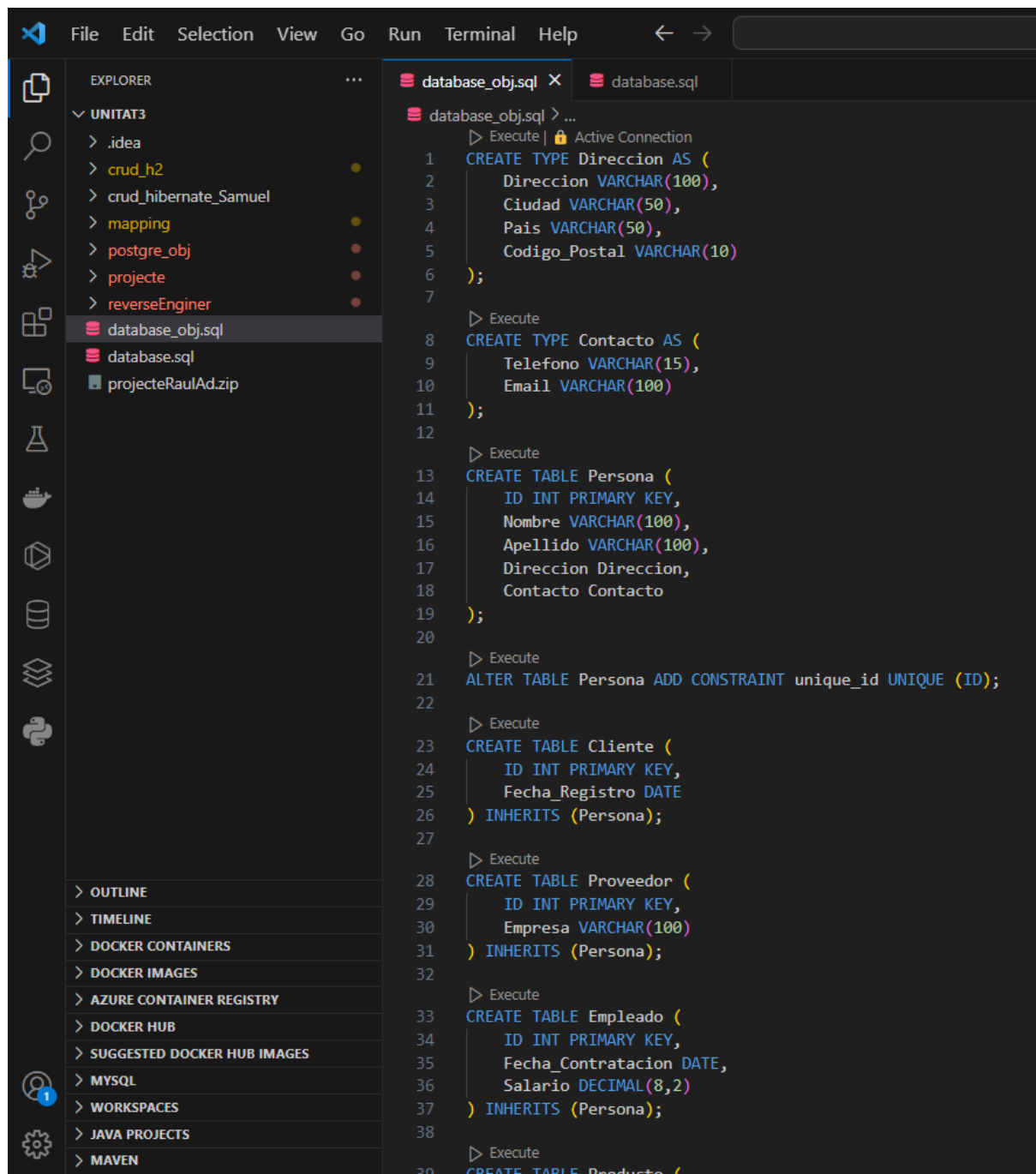
Imagen del script .sql:



The image shows a screenshot of an IDE (Integrated Development Environment) with a dark theme. The Explorer panel on the left shows a project structure under 'UNITAT3' with files like '.idea', 'crud\_h2', 'crud\_hibernate\_Samuel', 'mapping', 'postgre\_obj', 'projecte', 'reverseEngineer', 'database\_obj.sql', 'database.sql', and 'projecteRaulAd.zip'. The 'database.sql' file is selected. The main editor area shows the SQL script content, which is divided into sections by 'Execute' commands. The script defines several tables: 'Persona', 'Cliente', 'Proveedor', 'Empleado', 'Producto', and 'Pedido'. Each table definition includes its columns and data types, with some columns marked as primary keys.

```
1 CREATE TABLE Persona (  
2     ID INT PRIMARY KEY,  
3     Nombre VARCHAR(100),  
4     Apellido VARCHAR(100),  
5     Direccion VARCHAR(100),  
6     Ciudad VARCHAR(50),  
7     Pais VARCHAR(50),  
8     Codigo_Postal VARCHAR(10),  
9     Telefono VARCHAR(15),  
10    Email VARCHAR(100)  
11 );  
12  
13 Execute  
14 CREATE TABLE Cliente (  
15     ID INT PRIMARY KEY,  
16     Fecha_Registro DATE  
17 );  
18 Execute  
19 CREATE TABLE Proveedor (  
20     ID INT PRIMARY KEY,  
21     Empresa VARCHAR(100)  
22 );  
23 Execute  
24 CREATE TABLE Empleado (  
25     ID INT PRIMARY KEY,  
26     Fecha_Contratacion DATE,  
27     Salario DECIMAL(8,2)  
28 );  
29 Execute  
30 CREATE TABLE Producto (  
31     ID INT PRIMARY KEY,  
32     Nombre VARCHAR(100),  
33     Precio DECIMAL(5,2),  
34     Descripcion TEXT,  
35     Stock INT  
36 );  
37 Execute  
38 CREATE TABLE Pedido (  
39     ID INT PRIMARY KEY,  
40     ID_Producto INT,  
41     Fecha DATE,
```

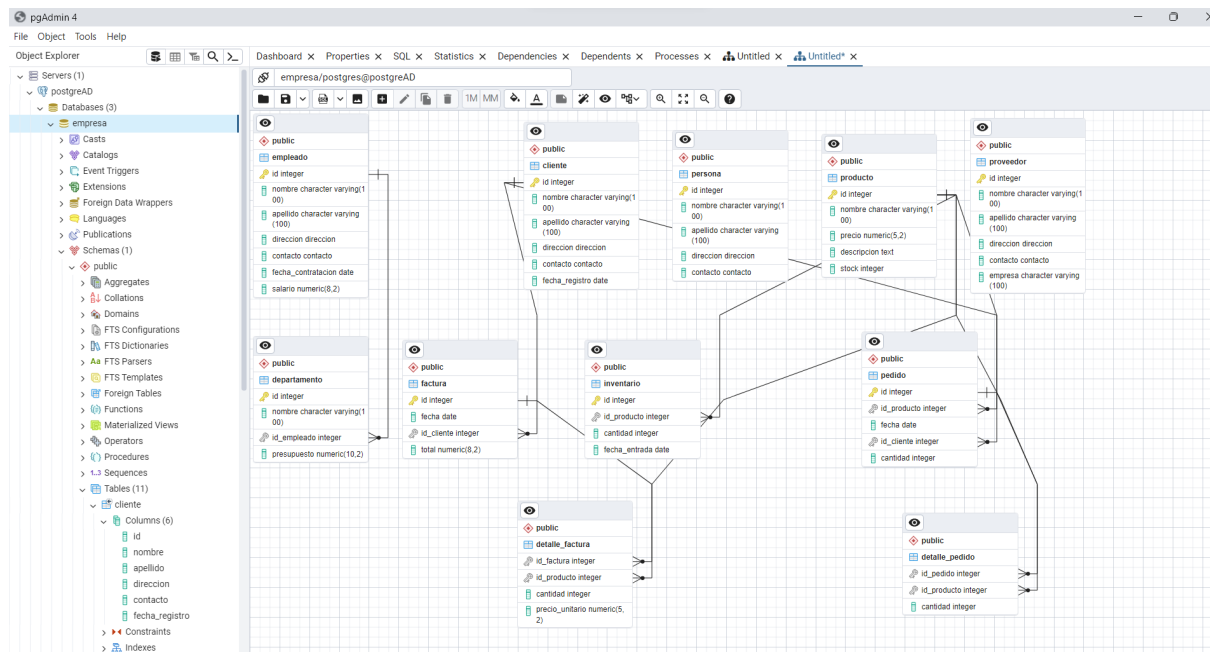
Imagen del script .sql para postgre aplicando el uso de objeto-relacional



The image shows a screenshot of an IDE (likely IntelliJ IDEA) with a PostgreSQL database script. The left sidebar shows the 'EXPLORER' view with a project structure including 'UNITAT3' and various files like '.idea', 'crud\_h2', 'crud\_hibernate\_Samuel', 'mapping', 'postgre\_obj', 'projecte', 'reverseEngineer', 'database\_obj.sql', 'database.sql', and 'projecteRaulAd.zip'. The main editor displays the 'database\_obj.sql' file, which contains SQL code for creating types, tables, and constraints.

```
database_obj.sql > ...
  > Execute | 🔒 Active Connection
1  CREATE TYPE Direccion AS (
2      Direccion VARCHAR(100),
3      Ciudad VARCHAR(50),
4      Pais VARCHAR(50),
5      Codigo_Postal VARCHAR(10)
6  );
7
  > Execute
8  CREATE TYPE Contacto AS (
9      Telefono VARCHAR(15),
10     Email VARCHAR(100)
11 );
12
  > Execute
13 CREATE TABLE Persona (
14     ID INT PRIMARY KEY,
15     Nombre VARCHAR(100),
16     Apellido VARCHAR(100),
17     Direccion Direccion,
18     Contacto Contacto
19 );
20
  > Execute
21 ALTER TABLE Persona ADD CONSTRAINT unique_id UNIQUE (ID);
22
  > Execute
23 CREATE TABLE Cliente (
24     ID INT PRIMARY KEY,
25     Fecha_Registro DATE
26 ) INHERITS (Persona);
27
  > Execute
28 CREATE TABLE Proveedor (
29     ID INT PRIMARY KEY,
30     Empresa VARCHAR(100)
31 ) INHERITS (Persona);
32
  > Execute
33 CREATE TABLE Empleado (
34     ID INT PRIMARY KEY,
35     Fecha_Contratacion DATE,
36     Salario DECIMAL(8,2)
37 ) INHERITS (Persona);
38
  > Execute
39 CREATE TABLE Producto (
```

E-R Final en PGAdmin



## Cambios realizados:

se han creado nuevos tipos, de los cuales van a contener atributos que luego vamos a implementar en otras tablas para que se hereden ede ellas. Estos nuevos tipos tienen datos de contacto y la direccion.

```
CREATE TYPE Direccion AS (
    Direccion VARCHAR(100),
    Ciudad VARCHAR(50),
    Pais VARCHAR(50),
    Codigo_Postal VARCHAR(10)
);
```

```
CREATE TYPE Contacto AS (
    Telefono VARCHAR(15),
    Email VARCHAR(100)
);
```

Se ha creado estos nuevos tipos, luego una nueva tabla persona, el cual tiene como atributos estos tipos : CREATE TABLE Persona (

```
    ID INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Apellido VARCHAR(100),
    Direccion Direccion,
    Contacto Contacto
);
```

de esta forma la tabla persona utiliza los tipos creados de direccion y Contacto para trabajar con sus atributos

```
ALTER TABLE Persona ADD CONSTRAINT unique_id UNIQUE (ID);
```

esta clausula es utilizada para evitar que los identificadores se repitan (spoiler, vengo del futuro).

y finalmente heredamos de persona, de esta forma evitamos atributos duplicados, que podemos heredar en la tabla cliente, empleado y proveedor. (se heredan todos los campos)

## RESUMEN:

De esta forma se consigue que una base de datos pueda ser más eficiente, y seria aún mucho más notoria en una base de datos con muchas tablas, que repiten atributos.