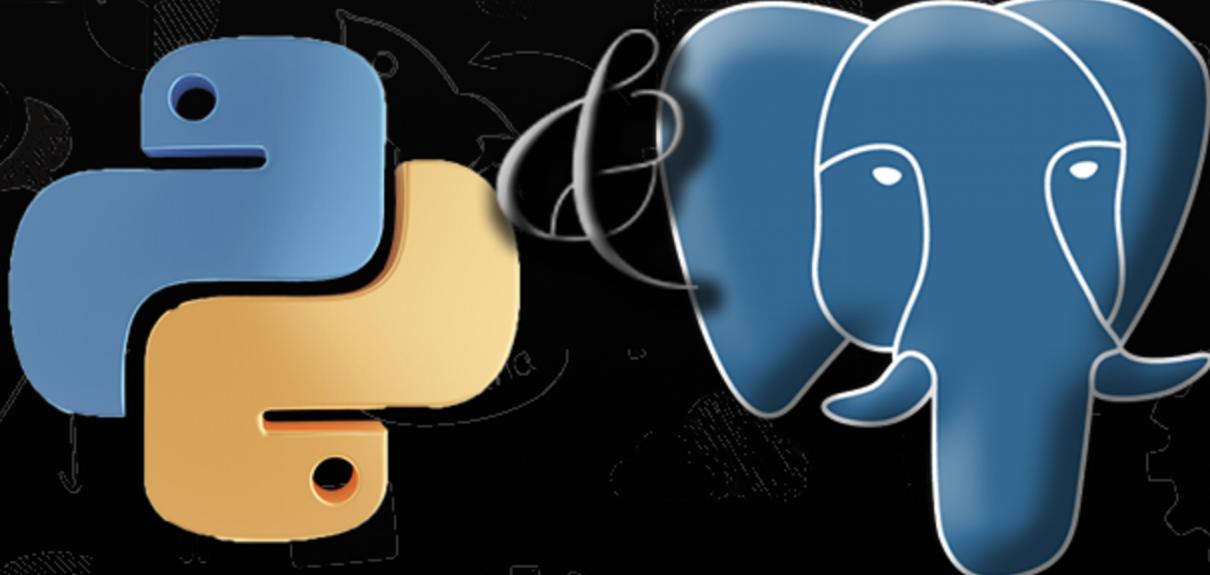
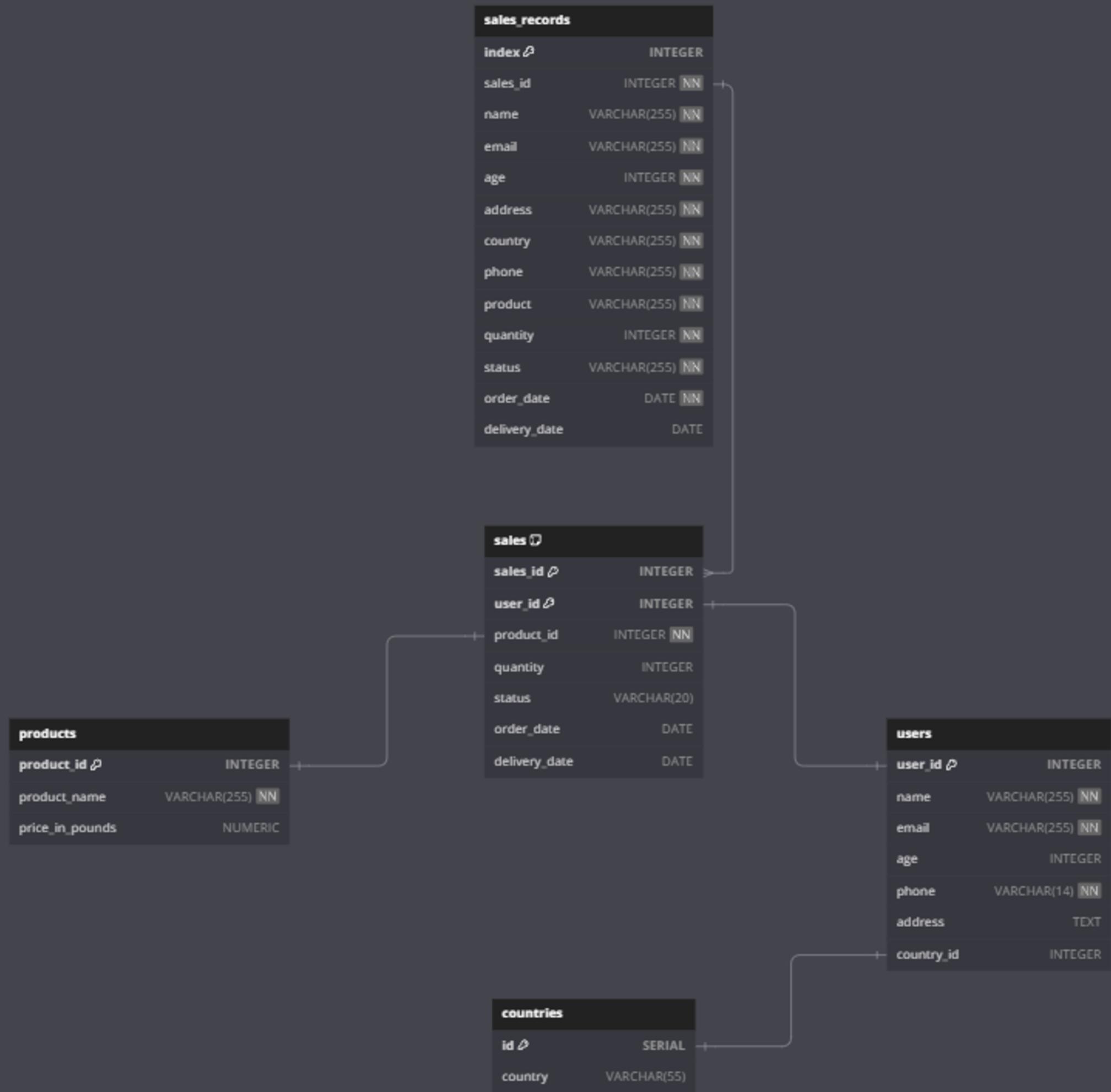


Generating Sales Data and Normalizing

Database With
Python & PostgreSQL.







```
1 from datetime import datetime as dd, timedelta as td
2 from faker import Faker
3 import pandas as pd
4 import random
5
6 seed = random.randint(5100, 5500)
7
8 fake = Faker()
9
10 fake.seed_instance(seed)
11
12 arr = []
13
14 column_names = ["sales_id", "name", "email", "age", "address", "country", "phone",
15                 "product", "price", "quantity", "status", "order_date", "delivery_date"]
16
17 price = {
18     "Iphone 12": 250,
19     "Iphone 12 mini": 225,
20     "Iphone 12 pro": 320,
21     "Iphone 12 pro max": 520,
22     "Iphone 13": 430,
23     "Iphone 13 pro": 540,
24     "Iphone 14": 870,
25     "Iphone 14 pro": 1050,
26     "Iphone 14 pro max": 1300,
27     "Iphone 15": 1150,
28     "Iphone 15 plus": 1280,
29     "Iphone 15 pro": 1400,
30     "Iphone 15 pro max": 1570,
31     "Lenovo PC": 750,
32     "PS4": 275,
33     "PS5": 400,
34     "Samsung S21": 285,
35     "Samsung S21+": 350,
36     "Samsung S22": 335,
37     "Samsung S22 Ultra": 480,
38     "Samsung S23 Ultra": 950,
39     "Samsung S23+": 670,
40     "Samsung S24": 880,
41     "Samsung S24 Ultra": 1200,
42     "Samsung S24+": 975
43 }
```



```
1 def get_price(product):
2     return price[product]
3
4
5
6 for _ in range(1000):
7
8     sales_id = random.randint(110000, 145000)
9     user_name = fake.name()
10    user_age = random.randint(20, 80)
11    user_email = f"{user_name.lower().replace(' ', '')} + str(user_age + random.randint(0,
10))}@{random.choice(['gmail', 'yahoo', 'outlook', 'hotmail'])}.com"
12    address = fake.address()
13    country = fake.country()
14    user_phone = f"{fake.country_calling_code()}{random.randint(700000000, 799999999)}"
15    product_name = random.choice(["Lenovo PC", "PS4", "PS5",
16                                    "Iphone 15", "Iphone 15 plus",
17                                    "Iphone 12", "Iphone 12 mini",
18                                    "Iphone 12 pro", "Iphone 12 pro max",
19                                    "Iphone 14 pro", "Iphone 14 pro max",
20                                    "Iphone 15 pro", "Iphone 15 pro max",
21                                    "Iphone 13", "Iphone 13 pro", "Iphone 14",
22                                    "Samsung S21", "Samsung S21+", "Samsung S22",
23                                    "Samsung S24", "Samsung S24+", "Samsung S24 Ultra", "Samsung S22 Ultra",
24                                    "Samsung S23+", "Samsung S23 Ultra"])
25
26    price = get_price(product_name)
27    quantity = random.randint(1, 3)
28    status = random.choice(["Pending", "Delivered", "Cancelled"])
29    payment_method = random.choice(["Cash", "Credit card", "PayPal", "Bank Transfer"])
30    order_date = fake.date_between(dd(2022, 1, 1), dd.now())
31    delivery_date = order_date + td(random.choice([1, 2, 3, 4]))
32
33    arr.append([sales_id, user_name, user_email, user_age, address, country,
34               user_phone, product_name, price, quantity, status, str(order_date),
35               str(delivery_date)])
36 df = pd.DataFrame(arr, columns = column_names)
```



```
1 import glob
2 import pandas as pd
3 import os
4 import json
5 from sqlalchemy import create_engine
6
7 # Declare path to the csv files
8 file_path = "data/csv"
9
10 # Select a list of all the files in the file path
11 files = glob.glob(f"{file_path}/*.csv")
12
13 # Initiates a dataframe
14 df = pd.DataFrame()
15
16 # loop through the list to convert each
17 # csv to dataframe and remove them
18 for csv_file in files:
19     temp_df = pd.read_csv(csv_file)
20     df = df.append(temp_df, ignore_index = True)
21     os.remove(csv_file)
22
23 # Stores the compiled csv file
24 df.to_csv("data/csv/sales_data.csv")
25
26 # get keys from json file
27 with open("keys.json") as file:
28     keys = json.load(file)
29
30 # Creates engine
31 engine = create_engine(f"{keys['provider']}+{keys['dialect']}://{{keys['username']}:{keys['password']}}@{{keys['host']}:{keys['port']}}/{{keys['database']}}")
32
33 # Saves to local sql database
34 df.to_sql(con = engine, name = "sales_records", index = False, if_exists = "append")
```

```
1 -- Create the primary sales_records table
2 CREATE TABLE sales_records (
3     index SERIAL PRIMARY KEY,
4     sales_id INTEGER NOT NULL,
5     name VARCHAR (255) NOT NULL,
6     email VARCHAR (255) NOT NULL,
7     age INTEGER NOT NULL,
8     address VARCHAR (255) NOT NULL,
9     country VARCHAR (255) NOT NULL,
10    phone VARCHAR (255) NOT NULL,
11    product VARCHAR (255) NOT NULL,
12    quantity INTEGER NOT NULL,
13    status VARCHAR (255) NOT NULL,
14    order_date DATE NOT NULL,
15    delivery_date DATE
16 );
17
18
19 -- Creating the users dimension table
20 CREATE TABLE users (
21     user_id INTEGER SERIAL PRIMARY KEY NOT NULL,
22     name VARCHAR (255) NOT NULL,
23     email VARCHAR (255) NOT NULL,
24     age INTEGER,
25     phone VARCHAR (25) NOT NULL,
26     address TEXT,
27     country VARCHAR
28 );
29
30
31 -- Creating the product dimension table
32 CREATE TABLE products (
33     product_id SERIAL PRIMARY KEY NOT NULL,
34     product_name VARCHAR (255) NOT NULL,
35     price NUMERIC
36 );
```



```
1 -- Creating the sales fact table
2 CREATE TABLE sales (
3     sales_id INTEGER NOT NULL,
4     user_id INTEGER NOT NULL,
5     product_id INTEGER NOT NULL,
6     quantity INTEGER,
7     status VARCHAR (20),
8     order_date DATE,
9     delivery_date DATE,
10    CONSTRAINT sales_user_pkey
11        PRIMARY KEY(sales_id, user_id),
12    CONSTRAINT sales_id_fkey
13        FOREIGN KEY(sales_id)
14            REFERENCES sales_records(sales_id),
15    CONSTRAINT user_id_fkey
16        FOREIGN KEY(user_id)
17            REFERENCES users(user_id),
18    CONSTRAINT fkey_product_id
19        FOREIGN KEY(product_id)
20            REFERENCES products(product_id)
21 );
22
23 CREATE TABLE users1 (
24     user_id SERIAL PRIMARY KEY NOT NULL,
25     name VARCHAR (255) NOT NULL,
26     email VARCHAR (255) NOT NULL,
27     age INTEGER,
28     phone VARCHAR (25) NOT NULL,
29     address TEXT,
30     country_id INTEGER,
31     CONSTRAINT country_id_fkey
32         FOREIGN KEY(country_id)
33             REFERENCES countries(id)
34 );
35
36
37 CREATE TABLE countries (
38     id SERIAL PRIMARY KEY,
39     country VARCHAR (55) NOT NULL
40 );
```