SQL Homework

Assessment

Scenario:

Kamu bekerja sebagai associate data analyst di sebuah perusahaan produk mainan. Kamu tengah memiliki waktu luang, jadi kamu ingin mengeksplorasi database untuk mendapat pemahaman yang lebih baik mengenai data perusahaan.

- Tabel yang menyimpan data penjualan adalah data yang terpisah berdasarkan tahun 2017 dan 2018.
- Setiap data penjualan tidak menyimpan harga produk. Harga produk disimpan di tabel products.
- Kamu ingin mendapatkan total penjualan tahun 2017 dan 2018 dalam sebuah query.

Dataset:

2

Kamu akan menggunakan table sales yang memuat data penjualan mainan suatu perusahaan di Meksiko. Tetapi datanya berbeda dengan data yang kita gunakan di Lesson 5, sehingga pastikan kamu mengikuti instruksi di bawah ya.

Bagaimana menyimpan data ke PGAdmin saya?

Pertama-tama, download <u>file Query (SQL)</u> yang dibutuhkan.

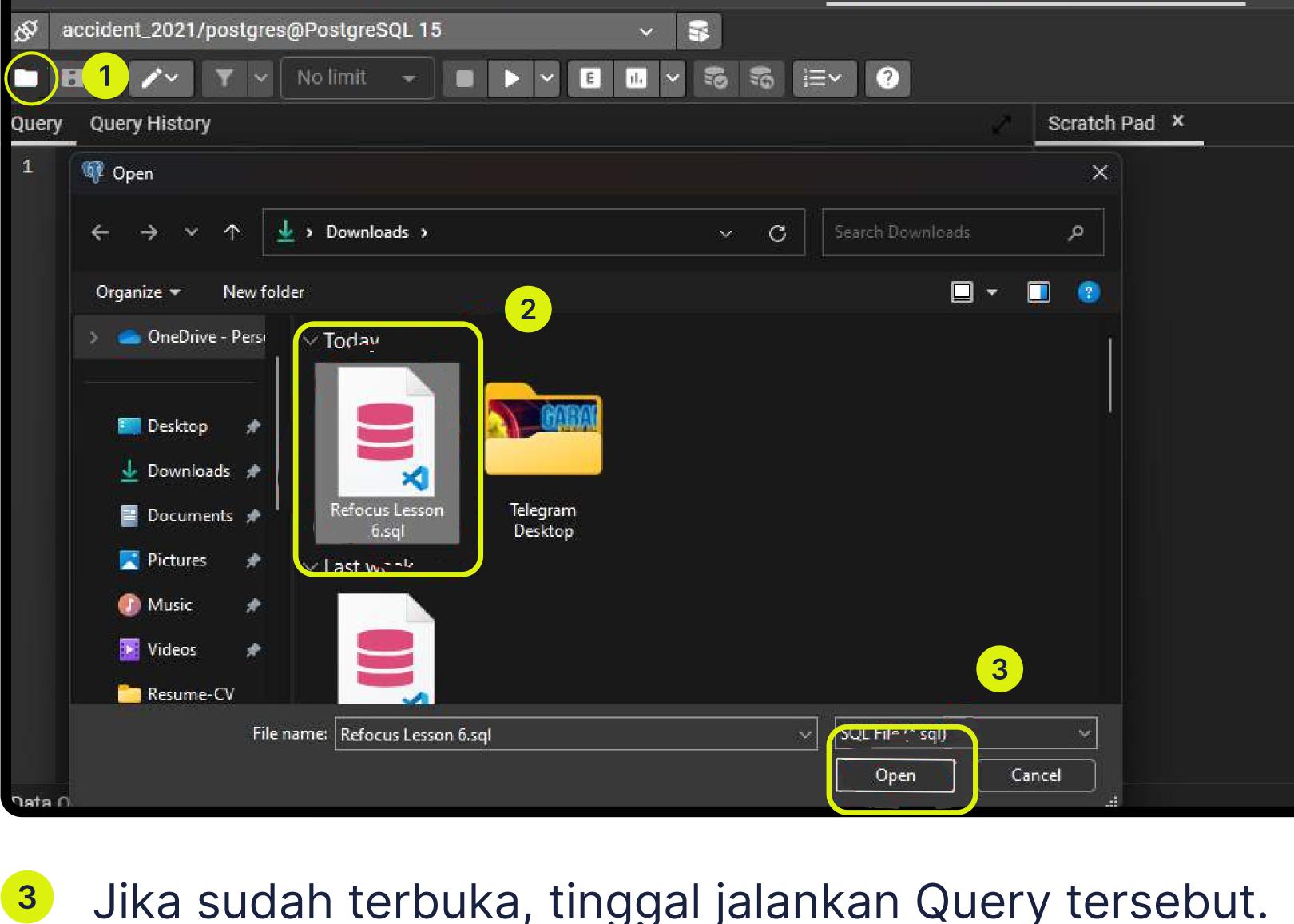
```
4
mexico_toy_sales.sql
                                   Открыть с помощью... 

        -- Table structure for sales_data_final
        DROP TABLE IF EXISTS "public". "sales";
        CREATE TABLE "public". "sales" (
          "sale id" int8,
          "date" date,
          "store id" int8,
          "product id" int8,
          "units" int8,
          "store_name" varchar(255) COLLATE "pg_catalog"."default",
          "store city" varchar(255) COLLATE "pg_catalog"."default",
          "store location" varchar (255) COLLATE "pg_catalog"."default",
          "store open date" date,
          "product name" varchar(255) COLLATE "pg_catalog"."default",
          "product category" varchar(255) COLLATE "pg catalog". "default",
          "product cost" float4,
          "product price" float4
```

Klik File nya lalu Open.

Dashboard Properties SQL Statistics Dependencies Dependents Processes accident_2021/postgres@PostgreSQL 15

Buka QueryTool pada PgAdmin, dan pilih logo folder atau Open File,



Query Query History

No limit.

E III

S S II

Scratch Pad

Scratch Pad

A

Dashboard Properties SQL Statistics Dependencies Dependents Processes 🖹 Refocus Lesson 6.sql

```
Table structure for england_region_north_east_covid
       DROP TABLE IF EXISTS "public". "england_region_north_east_covid";
      CREATE TABLE "public". "england_region_north_east_covid" (
        "date" date,
        "area_type" text COLLATE "pg_catalog"."default",
        "area_name" text COLLATE "pg_catalog"."default",
        "daily_cases" int4,
        "pcr_test" int4,
   10
        "new_first_dose" int4,
   11
        "new_second_dose" int4,
        "new_third_dose" int4
   13
   14
   15
   16
      -- Records of england_region_north_east_covid
      INSERT INTO "public"."england_region_north_east_covid" VALUES ('2023-01-25', '
      INSERT INTO "public". "england_region_north_east_covid" VALUES ('2023-01-24',
      INSERT INTO "public". "england_region_north_east_covid" VALUES ('2023-01-23', _!
      THESENT THIS "public" "contant courts could VALUES (!2022-01-22!
Data Output Messages Notifications
      Jalankan query.
 ■ B ∨ ✓ ▼ ∨ No limit ▼
                                 Query History
```

5 CREATE TABLE "public"."inventory" (
6 "store_id" int4,
7 "product_id" int4,
8 "stock_on_hand" int4
9)

> | FTS Dictionaries

> Image: Materialized Views

> b Operators

10

-- Table structure for inventory

DROP TABLE IF EXISTS "public". "inventory";

accident_2021/postgres@PostgreSQL 15

```
-- Records of inventory
     BEGIN;
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 1, 27);
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 2, 0);
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 3, 32);
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 4, 6);
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 5, 0);
      INSERT INTO "public"."inventory" ("store_id", "product_id", "stock_on_hand") VALUES (1, 6, 79);
Data Output Messages Notifications
NOTICE: table "inventory" does not exist, skipping
      table "products" does not exist, skipping
      table "sales_2017" does not exist, skipping
      table "sales_2018" does not exist, skipping
COMMIT
Query returned successfully in 4 secs 287 msec.
Total rows: 0 of 0
              Query complete 00:00:04.287
     Sekarang kamu sudah punya semua datanya di PGAdmin.
     Klik kanan di Schemas \rightarrow public \rightarrow table, dan klik Refresh.
                                                           "stor
    Schemas (2)
                                                           "prod

▼ ◆ public

                                                          "stoc
       > In Aggregates
       > 🔂 Collations
                                                  10
       > n Domains
                                                  11
         FTS Configurations
                                                  13
                                                            Reco
```

> Aa FTS Parsers
15 BEGIN;
> I FTS Templates
16 INSERT
17 INSERT
18 Functions
18 INSERT

Mes

Data Output

```
table
                                NOTICE:
     > ( Procedures
                                       table "p
                                NOTICE:
                                       table "
     > 1..3 Sequences
                                NOTICE:
                                       table "s
                                NOTICE:
     +able "s
       > mployee_pro
                       Create
       > == sales
                                        ned su
                       Refresh
     > ( Trigger Functions
                       Grant Wizard...
     > Types
                       Search Objects...
     > liews
                       PSQL Tool
    > 📀 trial
                       Query Tool
   Subscriptions
                                        0 of 0
    Sekarang kamu memiliki tabel inventory, products, sales_2017,
6
    sales_2018, dan stores.
    Tables (7)
       employee_promotion
           inventory
           products
        == sales
```

sales_2017

stores

tabel products.

sales_2017 dan sales_2018.

- Yang perlu kamu lakukan:

 1 JOIN tabel sales_2017 dengan tabel products dengan menggunakan key
- product_id.
 Hitung Tahun dan Nilai penjualan dari tabel sales_2017 dan products yang sudah di join. Tahun penjualan bisa didapatkan dari kolom date dari tabel
- sales_2017 menggunakan **EXTRACT** dan Nilai penjualan dapat didapakan dari kolom units di tabel sales_2017 dikalikan dengan product_price dari
- 3 Lakukan hal yang sama untuk tabel sales_2018 dan products.

Gunakan UNION ALL untuk menggabungkan total nilai penjualan dari tabel