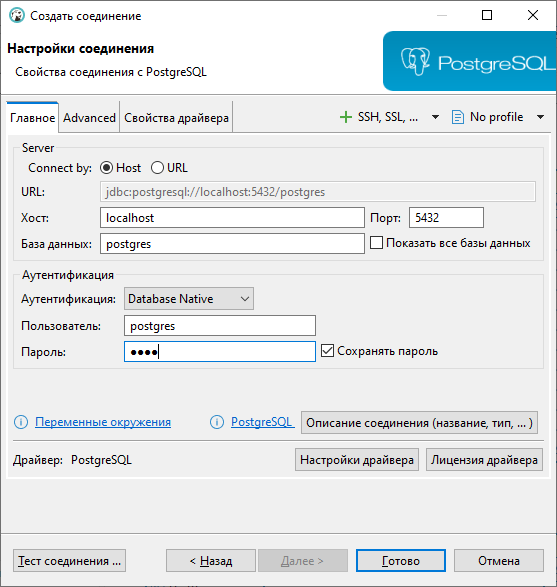
**1. Установка PostgreSQL и DBeaver**

1. **Скачать PostgreSQL**

**Скачать DBeaver**: https://dbeaver.io/download/  
После установки открой DBeaver → Database → New Connection → PostgreSQL  
Вводим свои данные. 

## 2. Создание таблиц

В Excel-файле **Superstore** есть три листа:**Orders**, **People,** **Returns**

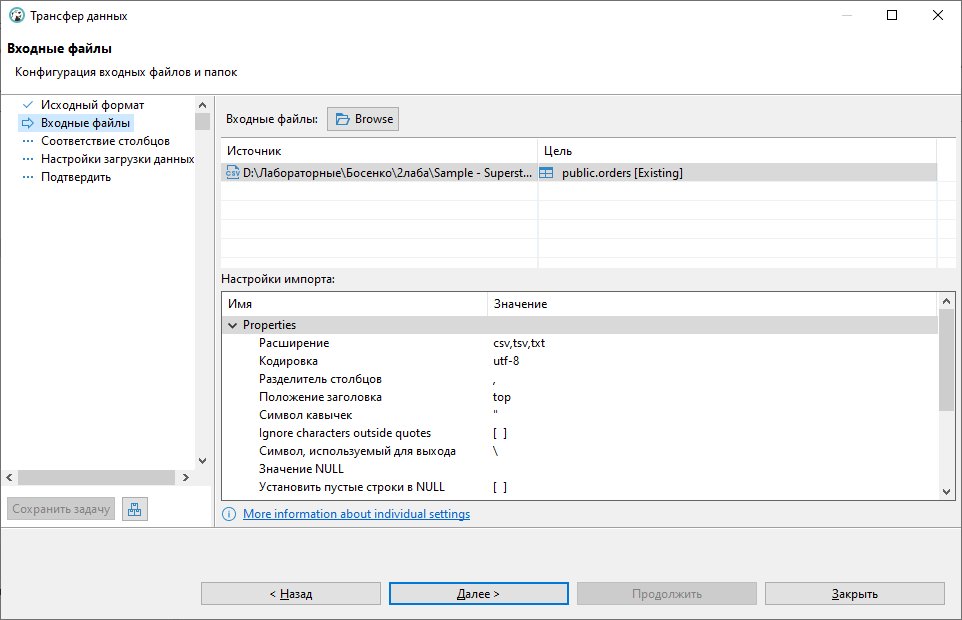
**Соответственно делаем три таблицы под них:**

|  |
| --- |
| CREATE TABLE orders (  row\_id SERIAL PRIMARY KEY,  order\_id VARCHAR(20),  order\_date DATE,  ship\_date DATE,  ship\_mode VARCHAR(50),  customer\_id VARCHAR(20),  customer\_name VARCHAR(100),  segment VARCHAR(50),  country VARCHAR(50),  city VARCHAR(50),  state VARCHAR(50),  postal\_code VARCHAR(20),  region VARCHAR(50),  product\_id VARCHAR(50),  category VARCHAR(50),  subcategory VARCHAR(50),  product\_name TEXT,  sales NUMERIC(10,2),  quantity INT,  discount NUMERIC(5,2),  profit NUMERIC(10,2)  );  CREATE TABLE people (  person VARCHAR(100),  region VARCHAR(50)  );  CREATE TABLE returns (  order\_id VARCHAR(20),  returned VARCHAR(10)  ); |

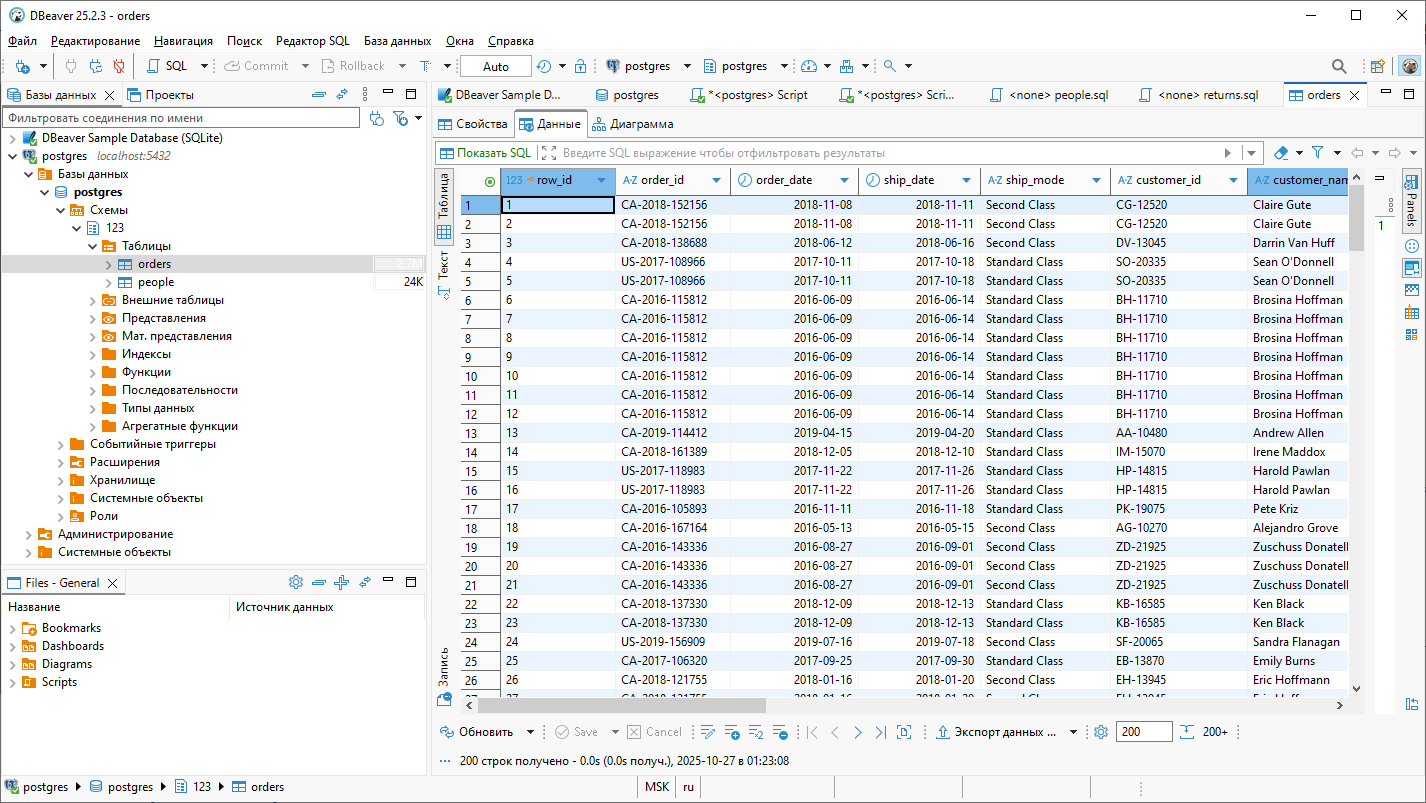
## 3. Загрузка данных из Excel

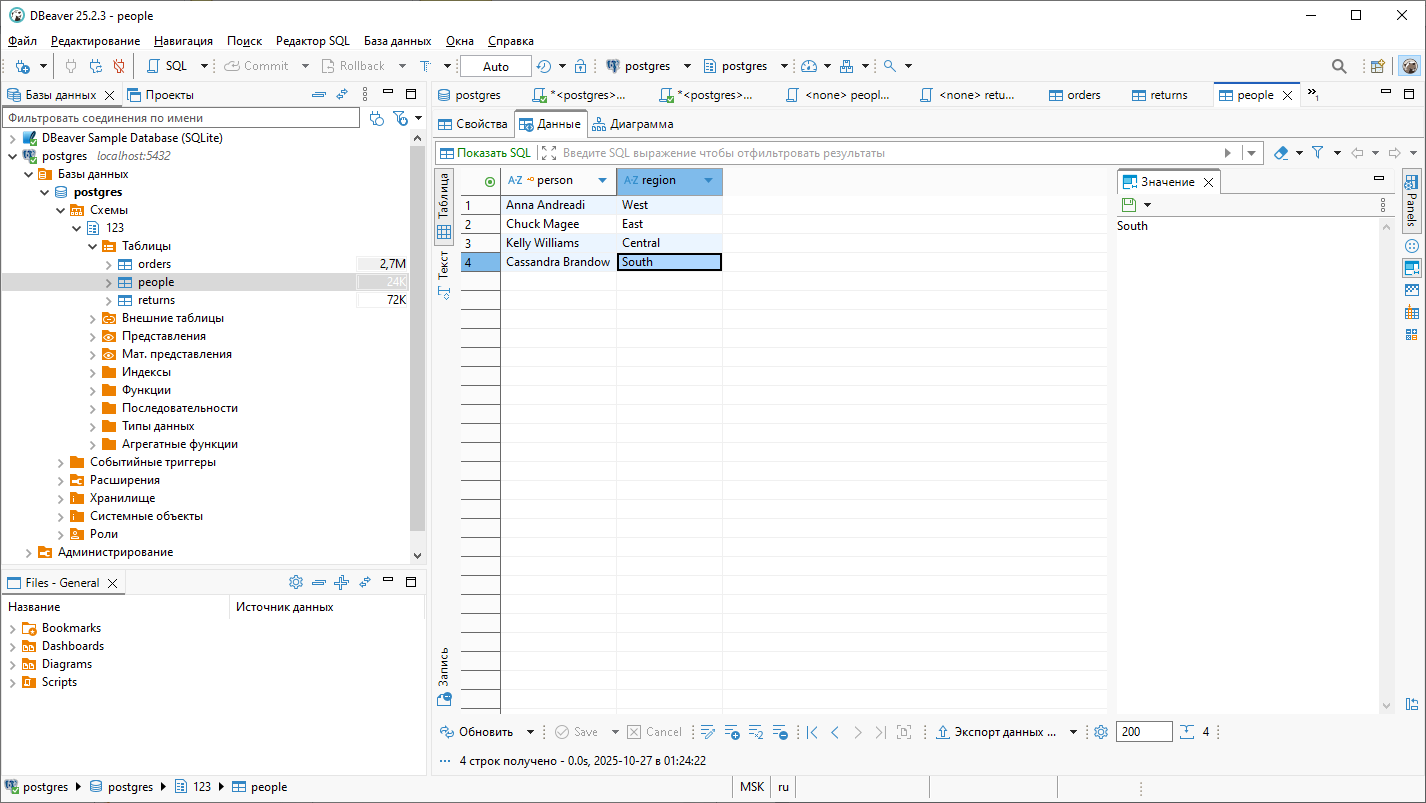
В DBeaver можно загрузить Excel напрямую:

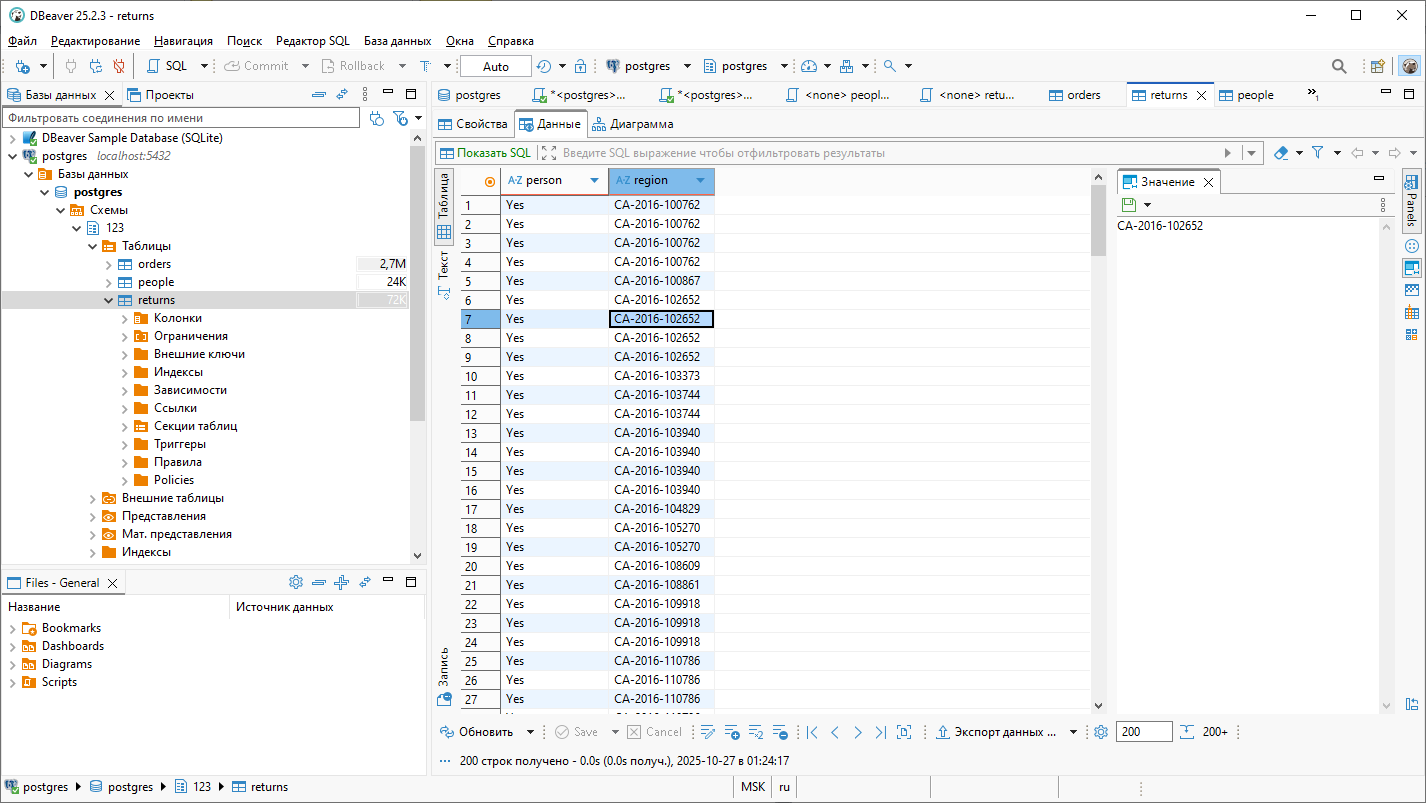
1. Откроем **мастера импорта данных в таблицу**
2. Выбираем таблицу
3. Источник: Excel или CSV
4. Укажи файл Sample - Superstore.xls и лист Orders
5. DBeaver сам сопоставит столбцы
6. Повторяем для листов **People** и **Returns**



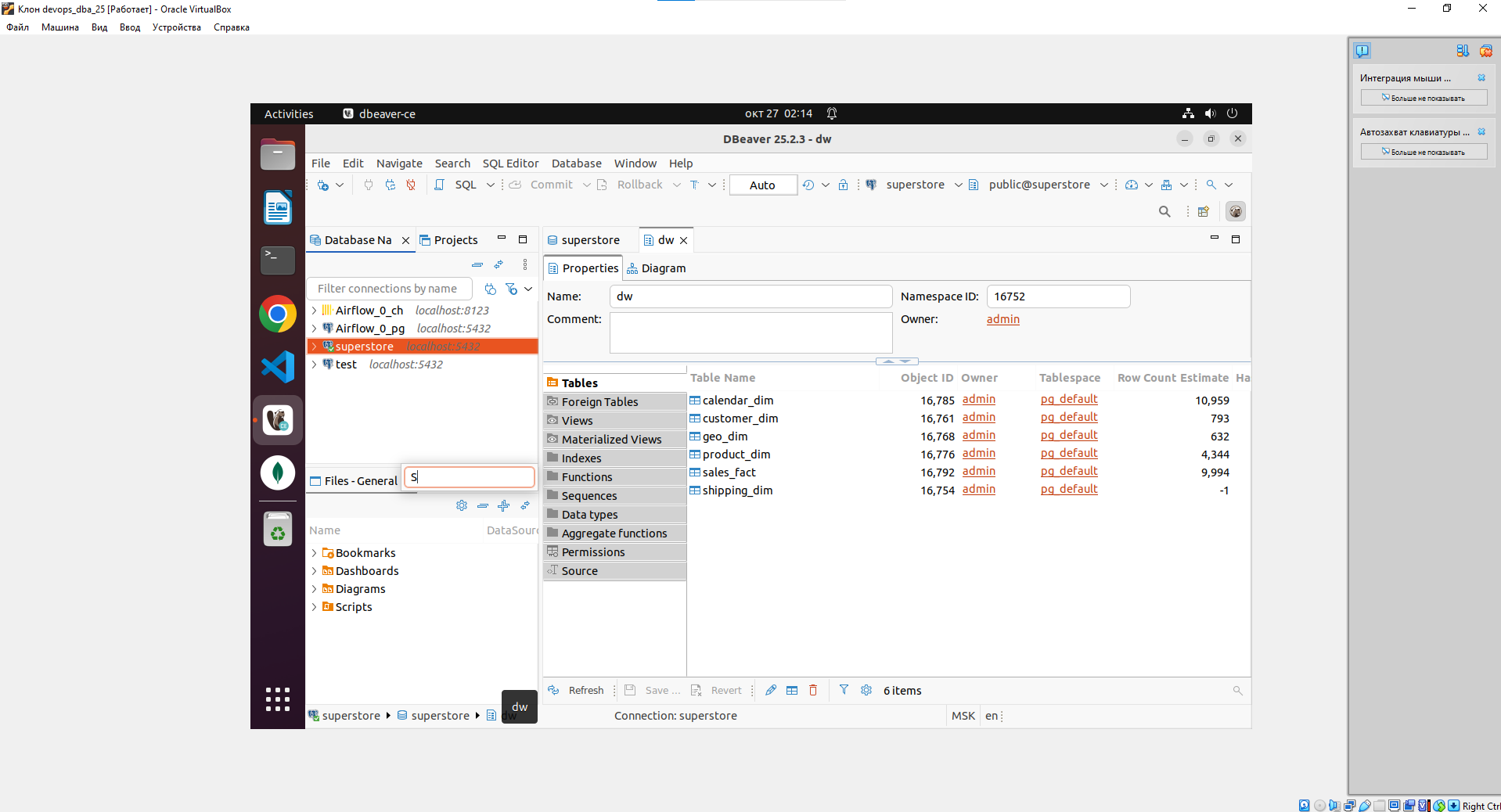
Имеем заполненные таблицы:







Также, установил виртуальную машину, но из за слабой производительности продолжил на своей ОС:



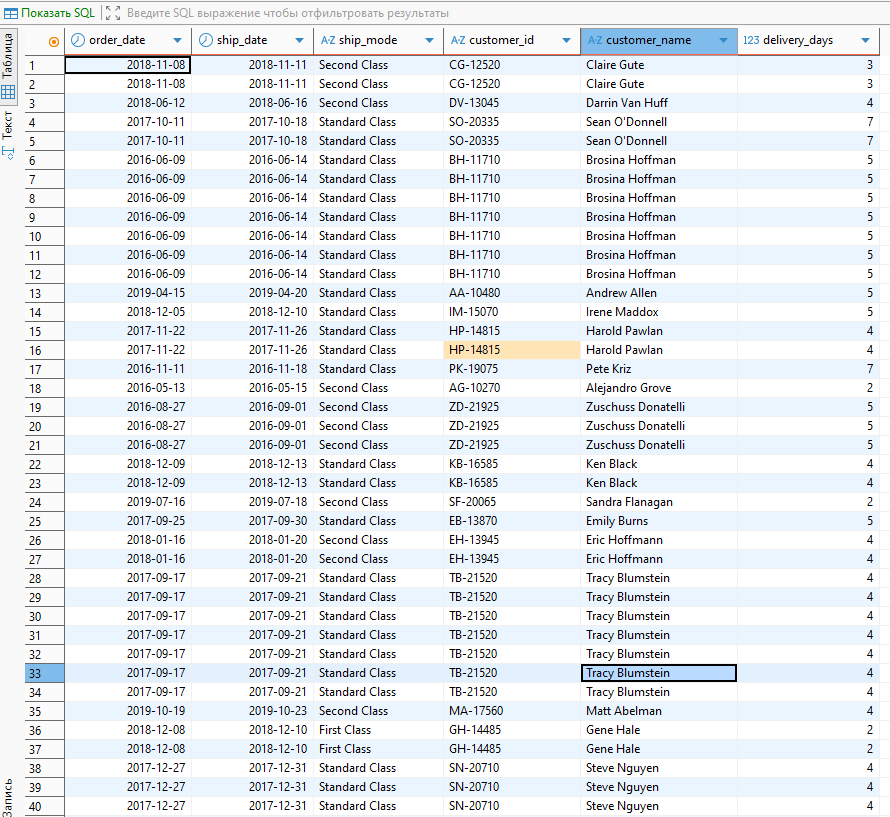
## 4. Индивидуальные задания



Был создан скрипт sql для создания представлений.

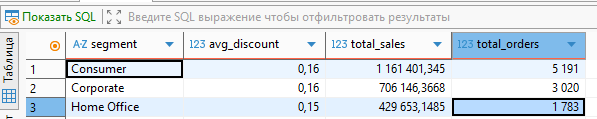
|  |
| --- |
| -- Задание 1 Создать представление по времени доставки  SET search\_path TO 123;  CREATE VIEW delivery\_time AS  SELECT  order\_id,  order\_date,  ship\_date,  ship\_mode,  customer\_id,  customer\_name,  (ship\_date - order\_date) AS delivery\_days  FROM orders;  --Задание 2 Рассчитать скидки по сегментам  CREATE VIEW discounts\_by\_segment AS  SELECT  segment,  ROUND(AVG(discount), 2) AS avg\_discount,  SUM(sales) AS total\_sales,  COUNT(\*) AS total\_orders  FROM orders  GROUP BY segment  ORDER BY avg\_discount DESC;  --Задание 3 Определить продажи по подкатегориям  CREATE VIEW sales\_by\_subcategory AS  SELECT  category,  subcategory,  SUM(sales) AS total\_sales,  SUM(profit) AS total\_profit,  COUNT(\*) AS num\_orders  FROM orders  GROUP BY category, subcategory  ORDER BY total\_sales DESC; |

**Задание 1 Представление по времени доставки:**



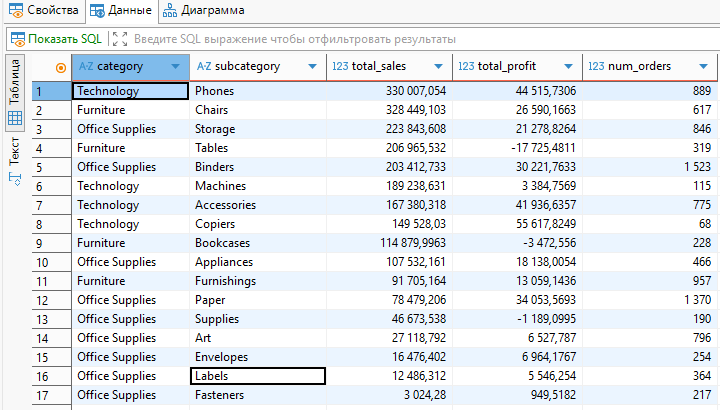
**Что делает:**  
– вычисляет разницу между ship\_date и order\_date (в днях)  
– сохраняет результат как представление delivery\_time

**Задание 2 расчет скидок по сегментам**



**Что делает:**  
– группирует данные по segment  
– считает среднюю скидку (AVG(discount))  
– суммарные продажи (SUM(sales))  
– количество заказов (COUNT(\*))

**Задание 3**

****

**Что делает:**  
– группирует товары по category и subcategory  
– считает общие продажи, прибыль и количество заказов  
– сортирует по продажам

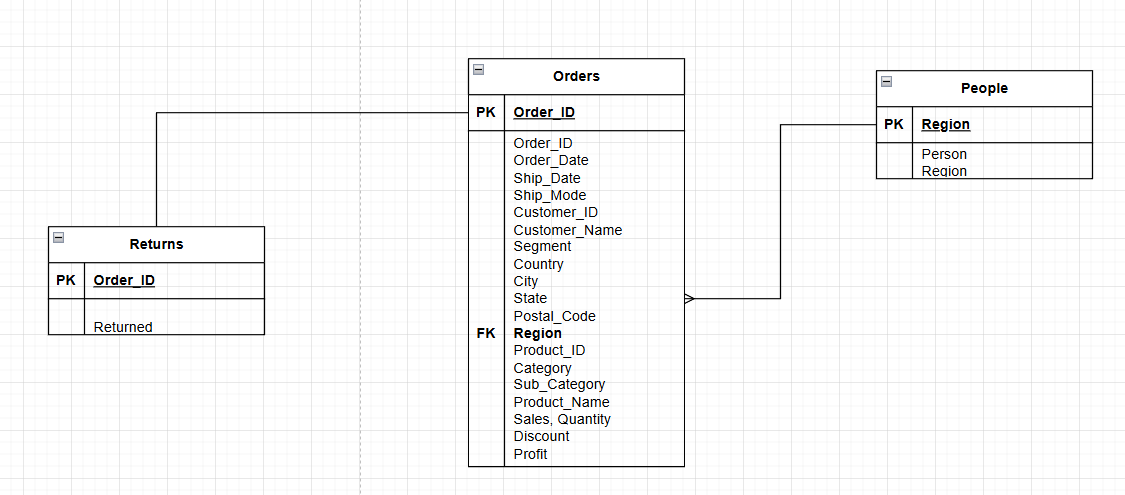
**Проверка результатов**

После создания представлений можно запросить данные:

|  |
| --- |
| SELECT \* FROM delivery\_time LIMIT 10;  SELECT \* FROM discounts\_by\_segment;  SELECT \* FROM sales\_by\_subcategory; |

## 5.ER-модель, DDL скрипты

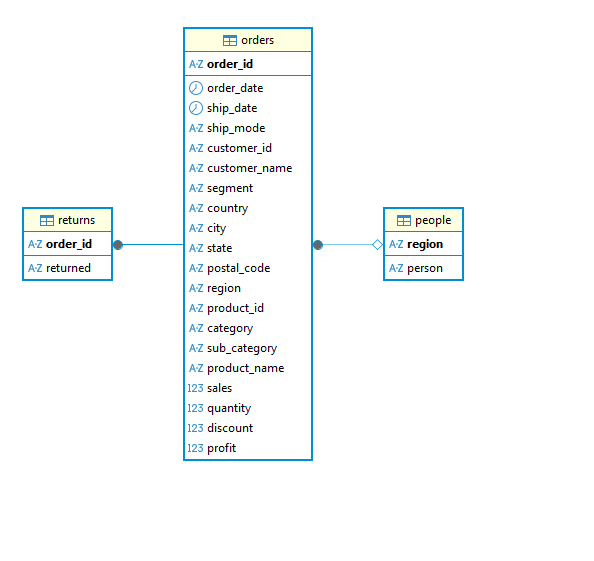
Была создана следующая ER модель через draw.io



Для генерации DDL в SQLDBM требуется бизнес-аккаунт, поэтому вставим следующий предположительный DDL код:

|  |
| --- |
| -- Удаляем таблицы, если они уже существуют  DROP TABLE IF EXISTS returns CASCADE;  DROP TABLE IF EXISTS orders CASCADE;  DROP TABLE IF EXISTS people CASCADE;  -- Таблица: people  CREATE TABLE people (  region VARCHAR(50) PRIMARY KEY,  person VARCHAR(100) NOT NULL  );  COMMENT ON TABLE people IS 'Список сотрудников, закреплённых за регионами';  COMMENT ON COLUMN people.region IS 'Название региона (уникальный ключ)';  COMMENT ON COLUMN people.person IS 'Имя сотрудника, ответственного за регион';  -- Таблица: orders  CREATE TABLE orders (  order\_id VARCHAR(20) PRIMARY KEY,  order\_date DATE NOT NULL,  ship\_date DATE,  ship\_mode VARCHAR(50),  customer\_id VARCHAR(20),  customer\_name VARCHAR(100),  segment VARCHAR(50),  country VARCHAR(50),  city VARCHAR(50),  state VARCHAR(50),  postal\_code VARCHAR(20),  region VARCHAR(50),  product\_id VARCHAR(50),  category VARCHAR(50),  sub\_category VARCHAR(50),  product\_name TEXT,  sales NUMERIC(10,2),  quantity INT,  discount NUMERIC(5,2),  profit NUMERIC(10,2),  CONSTRAINT fk\_orders\_region FOREIGN KEY (region)  REFERENCES people(region)  ON UPDATE CASCADE  ON DELETE SET NULL  );  COMMENT ON TABLE orders IS 'Таблица заказов Superstore';  COMMENT ON COLUMN orders.order\_id IS 'Уникальный идентификатор заказа';  COMMENT ON COLUMN orders.order\_date IS 'Дата размещения заказа';  COMMENT ON COLUMN orders.ship\_date IS 'Дата отгрузки заказа';  COMMENT ON COLUMN orders.ship\_mode IS 'Режим доставки';  COMMENT ON COLUMN orders.customer\_id IS 'Идентификатор клиента';  COMMENT ON COLUMN orders.customer\_name IS 'Имя клиента';  COMMENT ON COLUMN orders.segment IS 'Сегмент рынка (Consumer, Corporate и т.д.)';  COMMENT ON COLUMN orders.region IS 'Регион, связанный с таблицей people';  COMMENT ON COLUMN orders.sales IS 'Объем продаж';  COMMENT ON COLUMN orders.discount IS 'Применённая скидка';  COMMENT ON COLUMN orders.profit IS 'Полученная прибыль';  -- Индекс для ускорения фильтрации по региону  CREATE INDEX idx\_orders\_region ON orders(region);  -- Таблица: returns  CREATE TABLE returns (  order\_id VARCHAR(20) PRIMARY KEY,  returned VARCHAR(10) CHECK (returned IN ('Yes', 'No')),  CONSTRAINT fk\_returns\_order FOREIGN KEY (order\_id)  REFERENCES orders(order\_id)  ON UPDATE CASCADE  ON DELETE CASCADE  );  COMMENT ON TABLE returns IS 'Информация о возвратах заказов';  COMMENT ON COLUMN returns.order\_id IS 'Идентификатор заказа (FK → orders)';  COMMENT ON COLUMN returns.returned IS 'Статус возврата (Yes / No)';  COMMIT; |

По итогу получили:



Результаты проверок данных:

