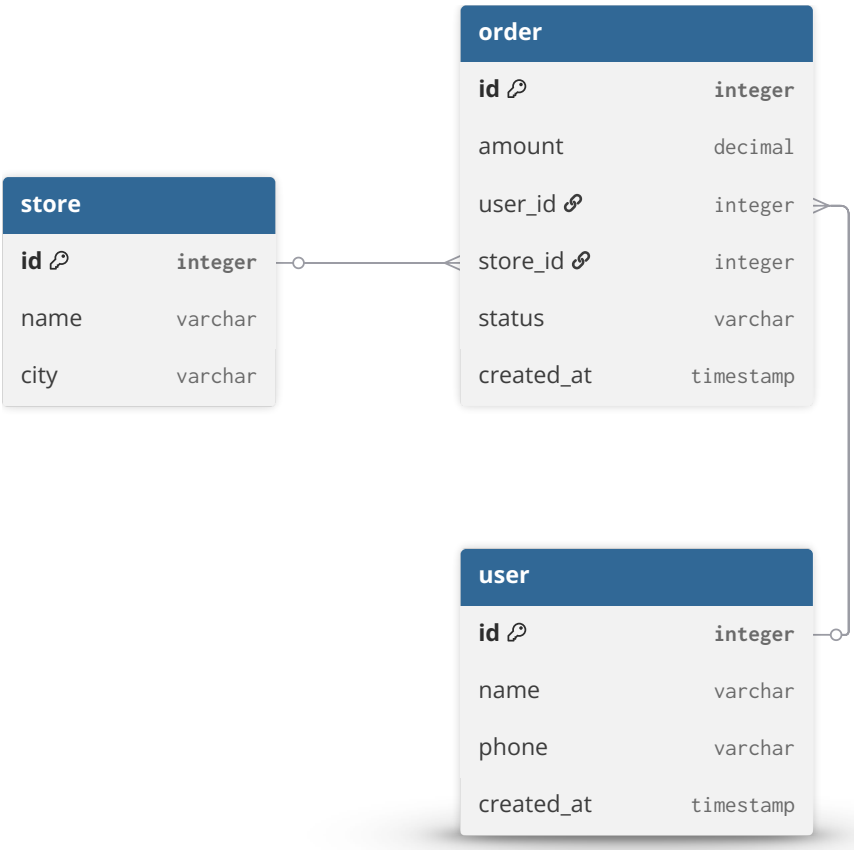
# Тестовое задание

В этом тестовом задании мы предлагаем вам реализовать простой ETL для расчета аналитического запроса над несколькими связанными источниками и оформить его в виде прототипа приложения, готового к запуску и тестированию с использованием контейнеризации.

## Контекст

Пользователи (**user**) оформляют заказы (**order**) в некоторых магазинах (**store**).
Упрощенная модель данных для этого процесса выглядит так:




---

**Задача**: необходимо для каждого города (**store.city**) определить топ-3 магазинов по общей сумме заказов (**order.amount**), сделанных пользователями с датой регистрации в **2025** году (**user.created_at**)

В результате мы ожидаем получить такую структуру, где **city** - название города (**store.city**), **store_name** - название магазина (**store.name**), а **target_amount** - целевая сумма:

| result | |
|---|---|
| city | varchar |
| store_name | varchar |
| target_amount | decimal |

## Инфраструктура и технологии

- Для чтения, записи и трансформации данных вы можете использовать **Apache Spark** (предпочтительно) + **Python, Scala или Java**

  Или же задействовать для вашего приложения только **Python** (возможно с привлечением библиотек **Pandas** или **Polars**)

- Приложение должно работать с **S3-совместимым** хранилищем (например, **Minio**)
  Входные данные ожидаются в виде **Parquet**-файлов (отдельных для store / order / user).
  Результат также записываем в **Parquet**

  Для тестирования вашего приложения вы можете использовать сформированные заранее датасеты или написать скрипт для их генерации.

- Приложение разворачивается в **Docker**, а ваш проект - содержит скрипт **docker-compose**, который запускает и оркестрирует все необходимые вашему приложению контейнеры, в т.ч. контейнер с S3-совместимым хранилищем

## Что мы оцениваем:

- Точность в следовании инструкции и требованиям
- Корректность работы вашего приложения
- Структуру и прозрачность кода и конфигурации
- При проверке вашего решения мы будем дополнительно использовать контрольные датасеты

## Будет плюсом, если вы:

- Используете ООП при определении модели данных и работе с ними
- Напишете тесты для вашего проекта
- Добавите логирование и/или сбор метрик выполнения с записью в файл (время работы, кол-во записей и т.п.)

## Инструкция по сдаче:

1. Создайте публичный репозиторий на Github или его аналоге
2. Загрузите все необходимые файлы, включая набор используемых вами тестовых данных и результат его обработки
3. Убедитесь, что ваш проект содержит **README** с кратким описанием его структуры, а также с инструкцией по запуску, и, желательно - по загрузке дополнительных данных на вход / экспорту результата
4. Пришлите ссылку на репозиторий