Итоговая домашняя работf Trino + PostgreSQL + MySQL + Iceberg + Визуализация

Вид задания: итоговая домашняя работа с проверкой преподавателем.

Преподаватель: Влад Шевченко.

Рекомендуемое время выполнения: 4–6 академических часов (180–270 минут) с учетом всех уровней: подключения, агрегации, визуализации и сохранения в Iceberg.

Формат сдачи: один Jupyter Notebook с названием final_homework.

Критерии приема работы

Работа допускается к оцениванию (и может получить от 1 до 10 баллов), если одновременно выполнены условия:

- Файл final_homework.ipynb загружен в систему сдачи и открывается без технических ошибок
- В ноутбуке есть осмысленное содержимое по всем трем уровням (подключения, агрегация, визуализация + Iceberg), а не пустой шаблон.

Работа считается **несданной** (0 баллов), если:

- Файл отсутствует, не открывается или по сути пуст (нет реализованных шагов задания)

Критерии оценивания

Оценка выставляется по совокупности реализации уровней, корректности шагов и качества оформления. Значения 9–10 баллов резервируются для заведомо выдающихся решений.

Базовые уровни (до 8 баллов)

1–3 балла.

- Частично реализован 1 уровень (подключения), есть подключение к Trino, но один из каталогов (PostgreSQL/MySQL) не проверен или проверка поверхностная.
- Агрегации, визуализация и Iceberg либо отсутствуют, либо только в виде заготовок

4–5 баллов.

- Полностью реализован 1 уровень: подключение к Trino и вывод схем/таблиц как минимум по одному каталогу (PostgreSQL или MySQL).
- 2 уровень реализован частично: есть хотя бы один агрегирующий запрос и загрузка результата в pandas DataFrame, но финальный DataFrame или логика агрегаций сделаны минимально.
- 3 уровень выполнен частично (например, один график без аккуратного оформления или пробная работа с Iceberg без полноценной таблицы)

6–8 баллов.

- 1 уровень выполнен полностью: подключение к Trino из ноутбука, вывод схем и таблиц и для PostgreSQL, и для MySQL.final_homework.pdf
- 2 уровень выполнен по заданию: минимум два осмысленных агрегирующих SQL-запроса через Trino, результаты загружены в pandas DataFrame, построен финальный агрегированный DataFrame.final_homework.pdf
- 3 уровень выполнен по заданию:
- минимум два графика на основе агрегированных данных, с подписями осей и заголовками;
- создана собственная схема в Iceberg (при отсутствии — создана) и Iceberg-таблица;
- данные сохранены в Iceberg, есть проверочный SQL-запрос через Trino к этой таблице
- Код структурирован, шаги логично прокомментированы.

Высокие уровни (9–10 баллов)

Оценки **9–10 баллов** ставятся редко и предполагают **существенное превышение требований**:

- Все требования на 6–8 баллов выполнены аккуратно и безошибочно.
- Плюс заметные улучшения, например:
- продуманная дополнительная аналитика (несколько вариантов агрегирующих запросов, сравнение PostgreSQL и MySQL, дополнительные метрики);
- более сложные, информативные визуализации с продуманным дизайном и пояснениями;
- использование дополнительных возможностей Iceberg (например, разделение по партициям, проверка версий/снапшотов) при условии, что это не противоречит установкам курса;
- очень хорошо организованный ноутбук (чистый код, понятная структура, качественные текстовые объяснения).

Преподаватель имеет право ограничить максимальный балл (например, не поднимать выше 8), если формально все шаги есть, но качество и глубина их реализации минимальны.

Цель работы

В одном Jupyter Notebook выполнить подключение к нескольким источникам данных через Trino, выполнить агрегирующие SQL-запросы и сохранить итоговые данные в Iceberg-таблицу.

Уровень 1 — Подключения

1. Подключиться к серверу Trino из Jupyter Notebook.

2. Проверить доступность каталога PostgreSQL в Trino: вывести список схем и таблиц.

3. Проверить доступность каталога MySQL в Trino: вывести список схем и таблиц.

Уровень 2 — Агрегация данных

1. Выбрать таблицы в PostgreSQL и/или MySQL, содержащие числовые или датовые поля.

2. Выполнить минимум два агрегирующих SQL-запроса через Trino (группировки, подсчёты, суммы, top-N и т. д.).

3. Загрузить результаты агрегирующих запросов в pandas DataFrame.

4. Сформировать финальный DataFrame для дальнейшей визуализации и сохранения.

Уровень 3 — Визуализация и сохранение данных в Iceberg

Визуализация:

1. Построить минимум два графика на основе агрегированных данных (например, график по датам и график распределения).
2. Оформить графики с подписями осей и заголовками.

Сохранение данных в Iceberg:

1. Создать свою схему в каталоге Iceberg (если её нет).
2. Создать Iceberg-таблицу для сохранения финального агрегированного DataFrame.
3. Сохранить агрегированные данные в эту таблицу.
4. Выполнить SQL-запрос через Trino для проверки содержимого таблицы.

Что сдавать

Один Jupyter Notebook с названием final_homework.ipynb, содержащий:
- подключение к Trino и вывод схем/таблиц PostgreSQL и MySQL;
- выполнение агрегирующих SQL-запросов;
- визуализацию данных;
- сохранение итогового набора данных в Iceberg и проверку таблицы.

Инструкции по выполнению (по шагам)

Уровень 1 — Подключения

Выполняется в final_homework.

- Подключиться к серверу Trino из Jupyter Notebook (настроить Python-клиент, параметры хоста/порта/каталога).
- Проверить доступность каталога PostgreSQL в Trino:
- вывести список схем;
- вывести список таблиц хотя бы в одной схеме (например, SHOW SCHEMAS, SHOW TABLES).
- Аналогично проверить доступность каталога MySQL: список схем и таблиц.

Уровень 2 — Агрегация данных

- Выбрать одну или несколько таблиц в PostgreSQL и/или MySQL, содержащие числовые или датовые поля (например, заказы, транзакции, метрики).
- Сформировать и выполнить **минимум два** агрегирующих SQL-запроса через Trino, например:
- группировки по дате, клиенту, категории;
- подсчет количества записей, суммирование, средние значения;
- top-N по какому-либо показателю.
- Загрузить результаты этих агрегирующих запросов в pandas DataFrame.
- На основе полученных DataFrame сформировать финальный агрегированный DataFrame, который будет использоваться и для визуализации, и для сохранения в Iceberg.

Уровень 3 — Визуализация и Iceberg

Визуализация:

- Построить **минимум два графика** по агрегированным данным, например:
- график по датам (линейный или столбчатый);
- график распределения (гистограмма, boxplot и т. п.).
- Оформить графики:
- задать осмысленные подписи осей;
- добавить заголовки, при необходимости легенду.

Сохранение данных в Iceberg: final_homework.pdf

- Создать собственную схему в каталоге Iceberg (если она еще не создана).
- Создать Iceberg-таблицу для финального агрегированного DataFrame (задать структуру полей).
- Сохранить агрегированные данные в эту таблицу (через Trino/SQL или доступный способ интеграции).
- Выполнить SQL-запрос через Trino к созданной Iceberg-таблице и вывести результат в ноутбуке для проверки содержимого.

Завершение и сдача

- Проверить, что final_homework.ipynb содержит:
- шаги подключения к Trino и вывод схем/таблиц PostgreSQL и MySQL;
- агрегирующие SQL-запросы и работу с pandas DataFrame;
- не менее двух оформленных графиков;
- создание схемы/таблицы Iceberg, запись данных и проверочный запрос.
- Убедиться, что все ячейки выполнены, а ноутбук сохраняется без ошибок.
- Загрузить файл в систему сдачи в установленный срок; при необходимости проверить, что файл открывается на стороне системы.