

Техническое задание: Сервис агрегации данных для аналитической платформы мерчантов

<input checked="" type="checkbox"/> Favorite	<input type="checkbox"/>
▼ Type	Business

1. Цель проекта

Разработать сервис для агрегации и предварительного расчета аналитических метрик из основной базы данных заказов маркетплейса, чтобы обеспечить быструю работу фронтенд-дашборда для мерчантов, где каждый мерчант должен иметь доступ только к своим агрегированным данным.

2. Исходные данные

- Существующая PostgreSQL база данных с детальными данными заказов всех мерчантов
- Готовый фронтенд с определенным набором метрик и отчетов
- Требуемые метрики для агрегации (на основе макета):
 - Общие показатели по заказам (принятые, выданные, отмененные, возвраты)
 - Финансовые метрики (оборот, средний чек)
 - ABC-анализ товаров
 - Динамика заказов по периодам
 - География заказов
 - Статистика по способам доставки и оплаты

- Клиентская аналитика

3. Функциональные требования

3.1 Основной функционал

- Создание и поддержка отдельной базы данных для агрегированных метрик
- Периодическое обновление агрегированных данных из основной БД
- Расчет всех необходимых метрик с группировкой по мерчантам
- API для получения агрегированных данных фронтендом
- Система разграничения доступа к данным по merchant_id

3.2 Детальные требования по агрегации

- Агрегация данных должна выполняться по следующим временным срезам:
 - Ежедневные метрики
 - Еженедельные метрики
 - Ежемесячные метрики
 - Квартальные метрики
- Необходимые группировки данных:
 - По мерчантам (основной идентификатор для разграничения доступа)
 - По товарам
 - По категориям товаров
 - По городам
 - По типам доставки
 - По способам оплаты

3.3 Требования к API

- REST API на FastAPI для получения агрегированных данных
- Эндпоинты должны поддерживать:
 - Аутентификацию и авторизацию мерчантов
 - Фильтрацию по датам
 - Фильтрацию по статусам заказов
 - Пагинацию
 - Сортировку

4. Нефункциональные требования

- Использование FastAPI для разработки API
- Документация API в формате OpenAPI/Swagger
- Время ответа API не более 500мс
- Обработка ошибок и повторных попыток при сбоях в процессе агрегации

5. Этапы разработки

1. Проектирование схемы базы данных для агрегированных метрик с учетом разделения данных по мерчантам
2. Разработка скриптов агрегации данных
3. Разработка API с системой авторизации
4. Тестирование производительности
5. Документирование

6. Результаты работы

- Исходный код сервиса
- SQL-скрипты создания структуры БД
- Документация API (Swagger)

- Инструкция по развертыванию

7. Дополнительные требования

- Архитектура должна предусматривать возможность горизонтального масштабирования
- Механизм обработки ошибок и повторных попыток при сбоях
- Оптимизация запросов для быстрой работы с большими объемами данных