

Аналитический пайплайн для мониторинга онлайн-магазина

Цель проекта:

Создать полноценную инфраструктуру аналитики для онлайн-магазина, включающую:

- хранение данных
- автоматическую загрузку и генерацию данных
- построение дашбордов
- систему алертов в реальном времени
- автоматическую отчетность с доставкой пользователю

Оглавление

Описание архитектуры системы.....	1
Проектирование базы данных	4
Описание дашборда.....	7
Описание alert'ов.....	9
Заключение	14
Примечания.....	15

Кузнецов Арсений

email: ababaan365@gmail.com

Phone: +7 (901)701-80-33

Telegram: @cantsmoke

GitHub: <https://github.com/cantsmoke>

Описание архитектуры системы

Разработанный проект представляет собой end-to-end аналитический пайплайн, обеспечивающий полный цикл работы с данными - от их генерации до представления пользователю в виде визуализаций, автоматических уведомлений и отчетности. В основе решения лежит реляционная база данных PostgreSQL, которая выступает центральным хранилищем информации о пользователях, заказах, просмотрах товаров и другой бизнес-активности. Данные в данное хранилище поступают искусственным образом: отдельные скрипты имитируют поведение клиентов в онлайн-магазине и записывают результаты в соответствующие таблицы. Такой подход позволяет воспроизводить характерные для электронной коммерции метрики (DAU, количество заказов, выручка, коэффициент конверсии) и в дальнейшем использовать их в аналитике.

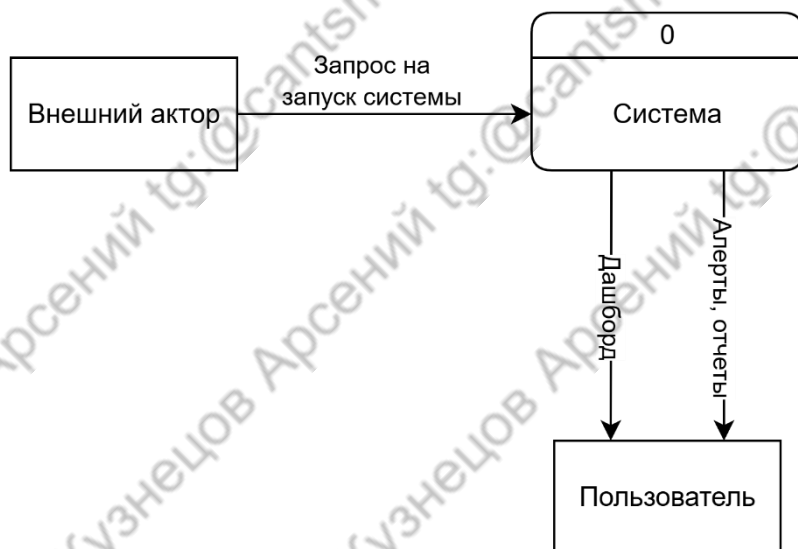


Рисунок 1. DFD-диаграмма на контекстном уровне

Собранные в базе данных сведения используются несколькими подсистемами. Во-первых, реализована интеграция с Power BI, что позволяет формировать динамические дашборды и визуализировать ключевые показатели за выбранные периоды. Во-вторых, в проект встроена система алертов: отдельный скрипт извлекает агрегированные метрики из PostgreSQL, сравнивает их с заданными пороговыми значениями и в случае отклонений отправляет уведомления через Telegram-бота. Дополнительно разработан модуль формирования регулярной отчетности, который создает PDF-документы с графиками и ключевыми метриками за неделю, сохраняет их в файловой системе и предоставляет возможность пользователю запросить отчет через Telegram.

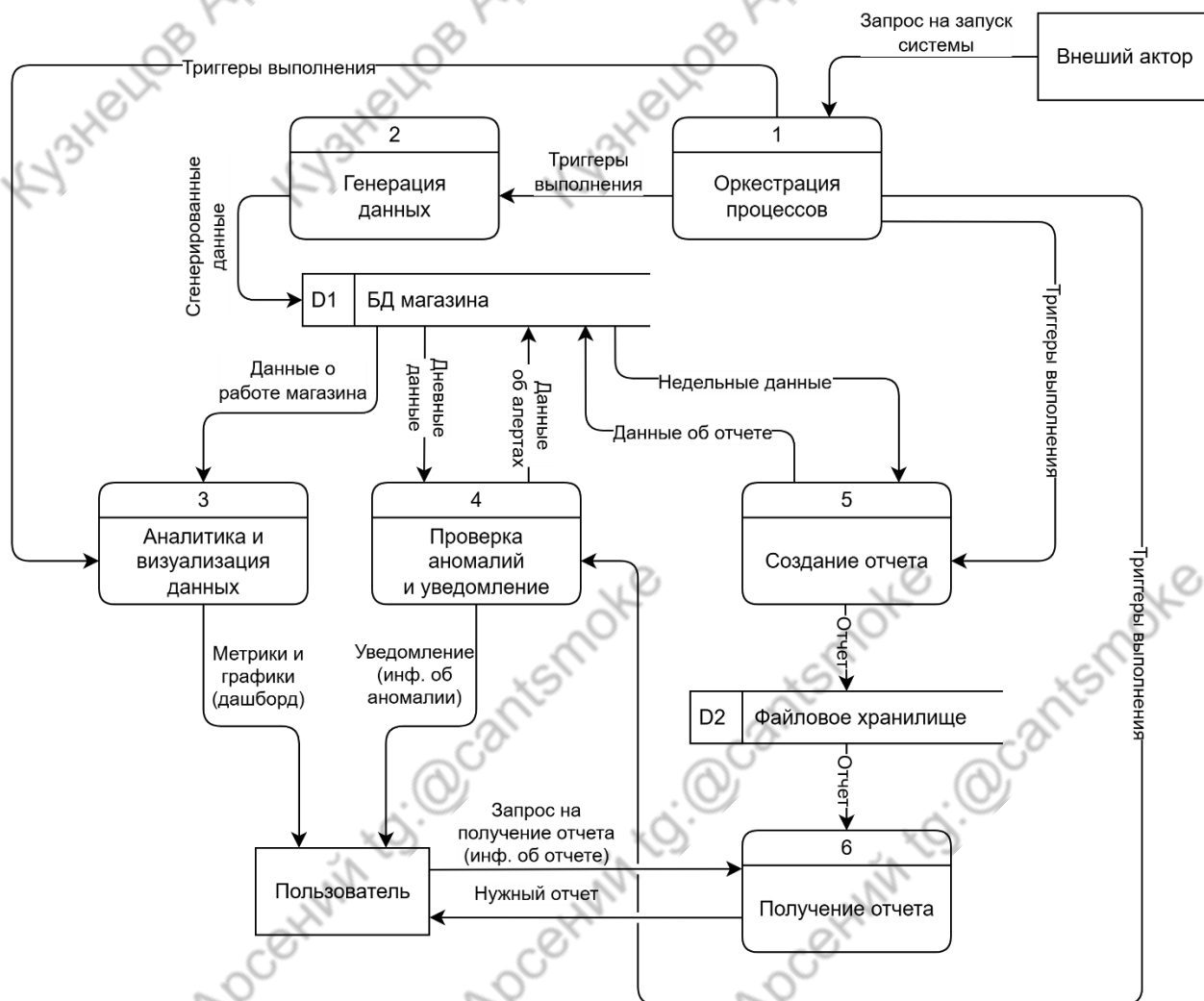


Рисунок 2. DFD-диаграмма на логическом уровне

Принцип работы оркестратора реализован посредством Apache Airflow, который выполняет роль управляющего механизма для скриптов ETL, проверки метрик и формирования отчетности. Оркестратор обеспечивает расписание запуска задач, где одна «модельная минута» соответствует одному дню симулированной активности. Каждый цикл состоит из последовательных шагов: сначала выполняется генерация новых данных за сутки, затем запускается процедура проверки метрик и отправки алертов, после чего (каждые семь шагов) активируется модуль генерации недельного отчета. Airflow управляет зависимостями между задачами, гарантируя, что формирование отчета возможно только после успешного завершения этапов генерации данных и проверки алертов. Такое построение пайплайна позволяет централизованно контролировать процессы, обеспечивать их воспроизводимость и исключать нарушения порядка выполнения.

В результате реализована автоматизированная и масштабируемая система, которая охватывает полный цикл обработки данных и обеспечивает аналитическую поддержку пользователей. Архитектура объединяет в себе базы данных, инструменты визуализации, механизмы уведомлений и модуль генерации отчетов в рамках одного аналитического контура.

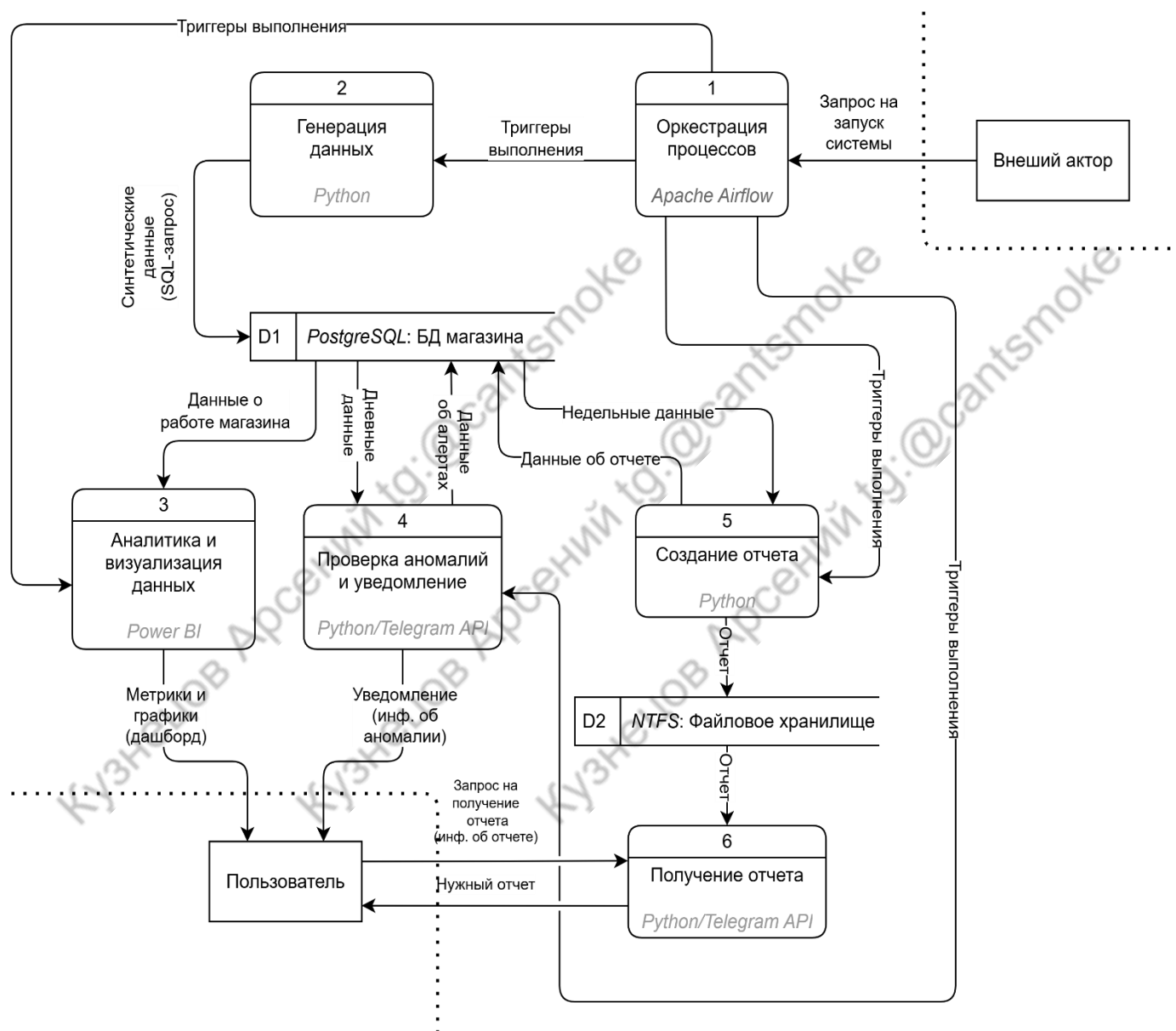


Рисунок 3. DFD-диаграмма на физическом уровне

Проектирование базы данных

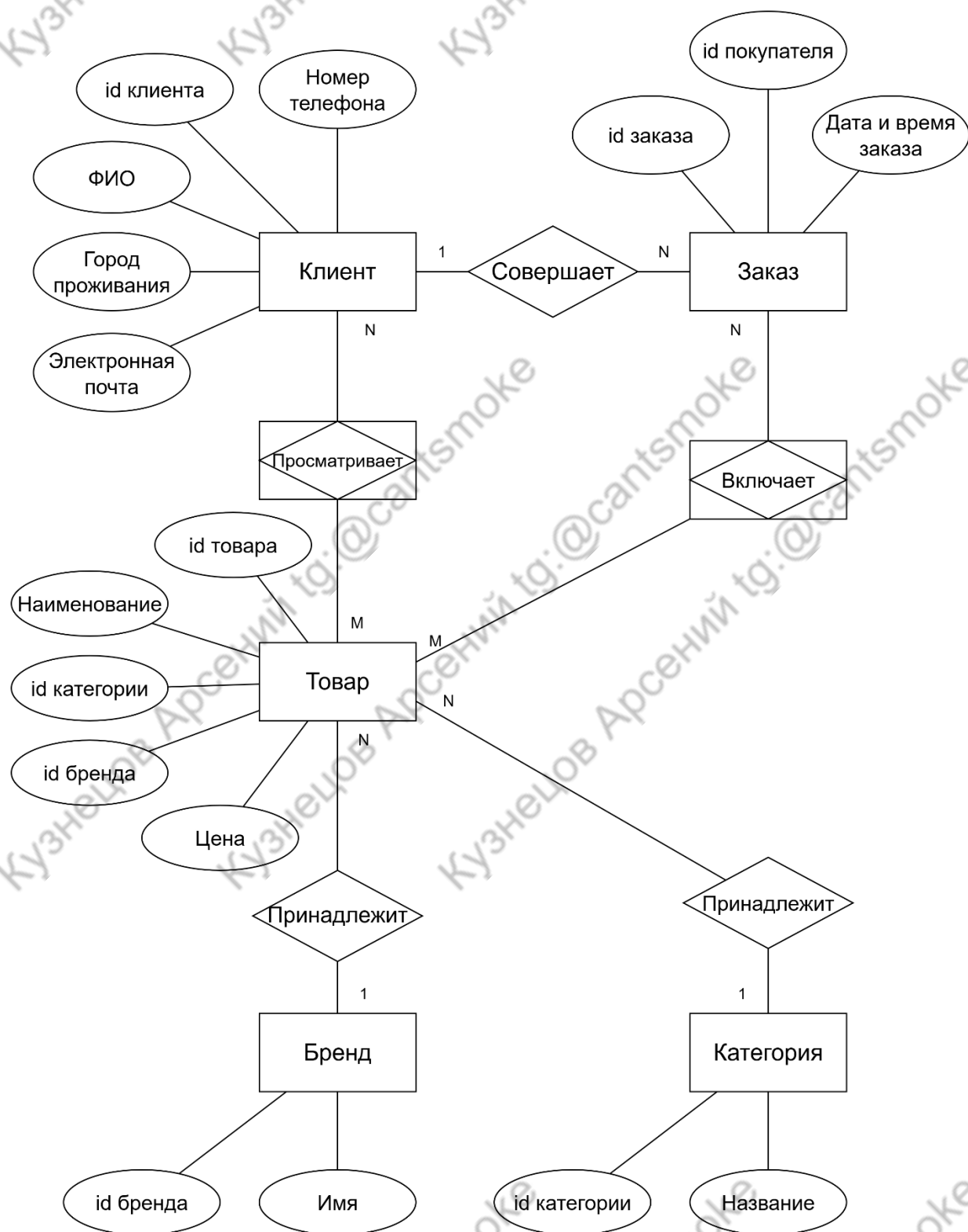


Рисунок 4. ER-диаграмма на концептуальном уровне

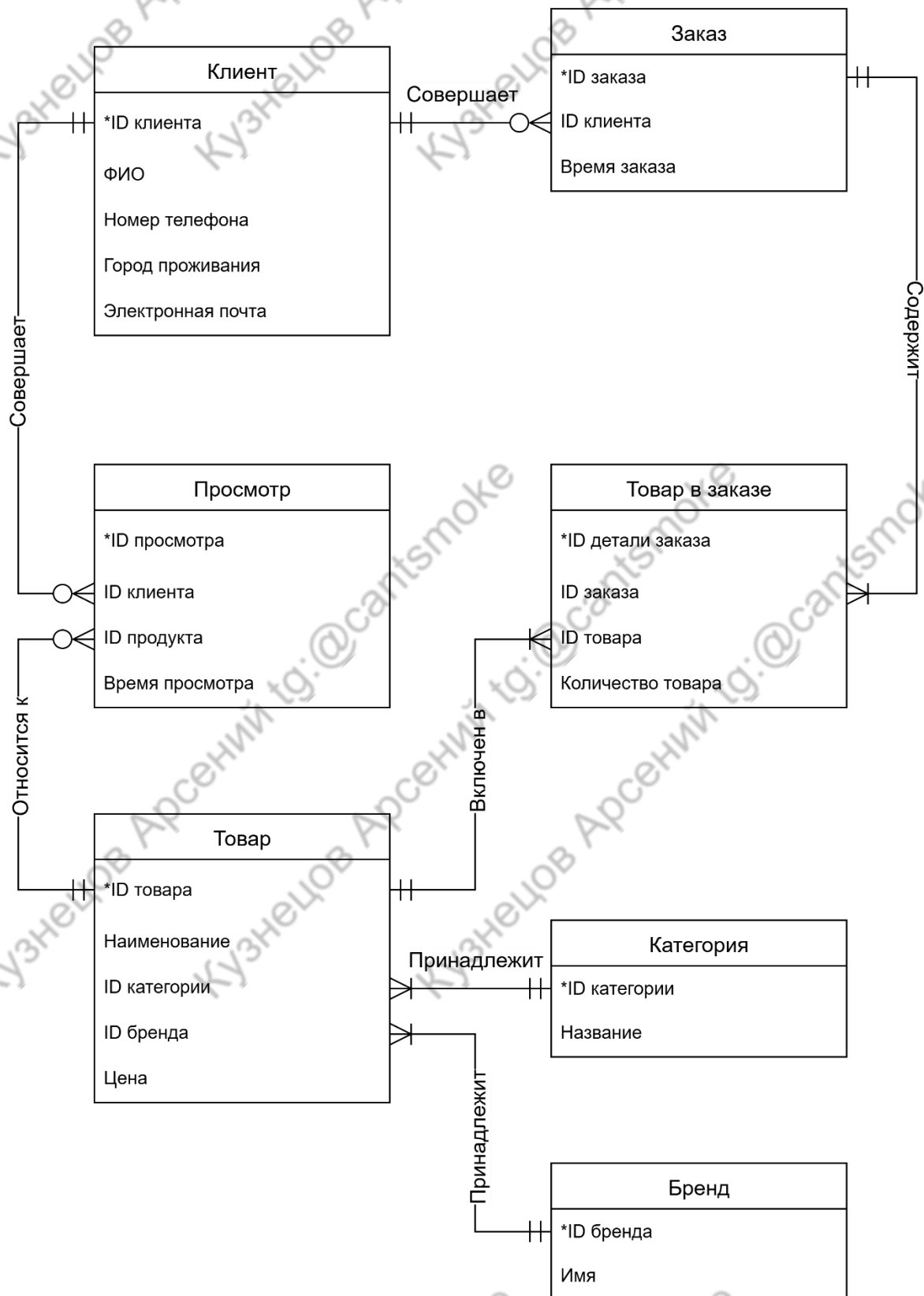


Рисунок 5. ER-диаграмма на логическом уровне

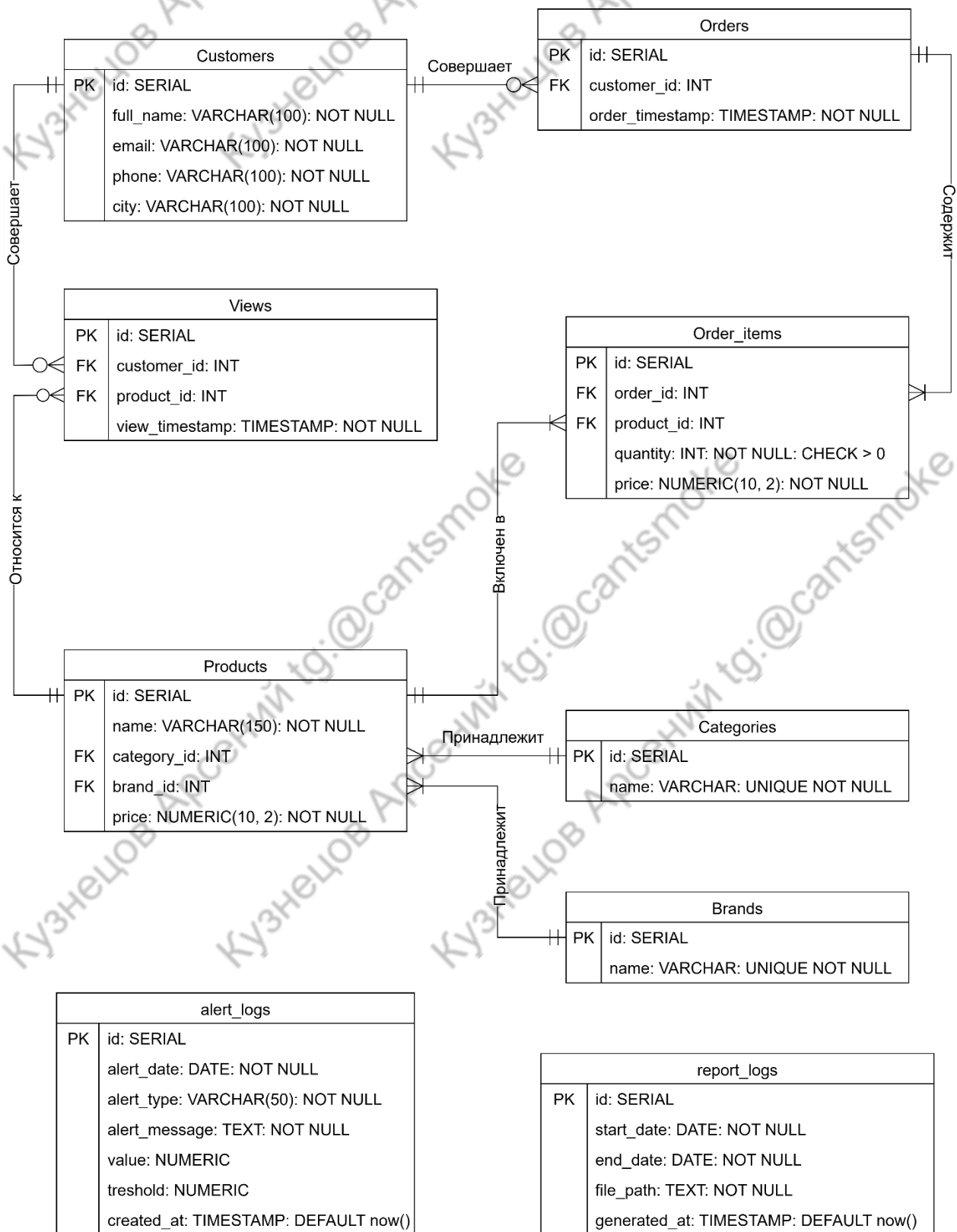


Рисунок 6. ER-диаграмма на физическом уровне

Описание дашборда

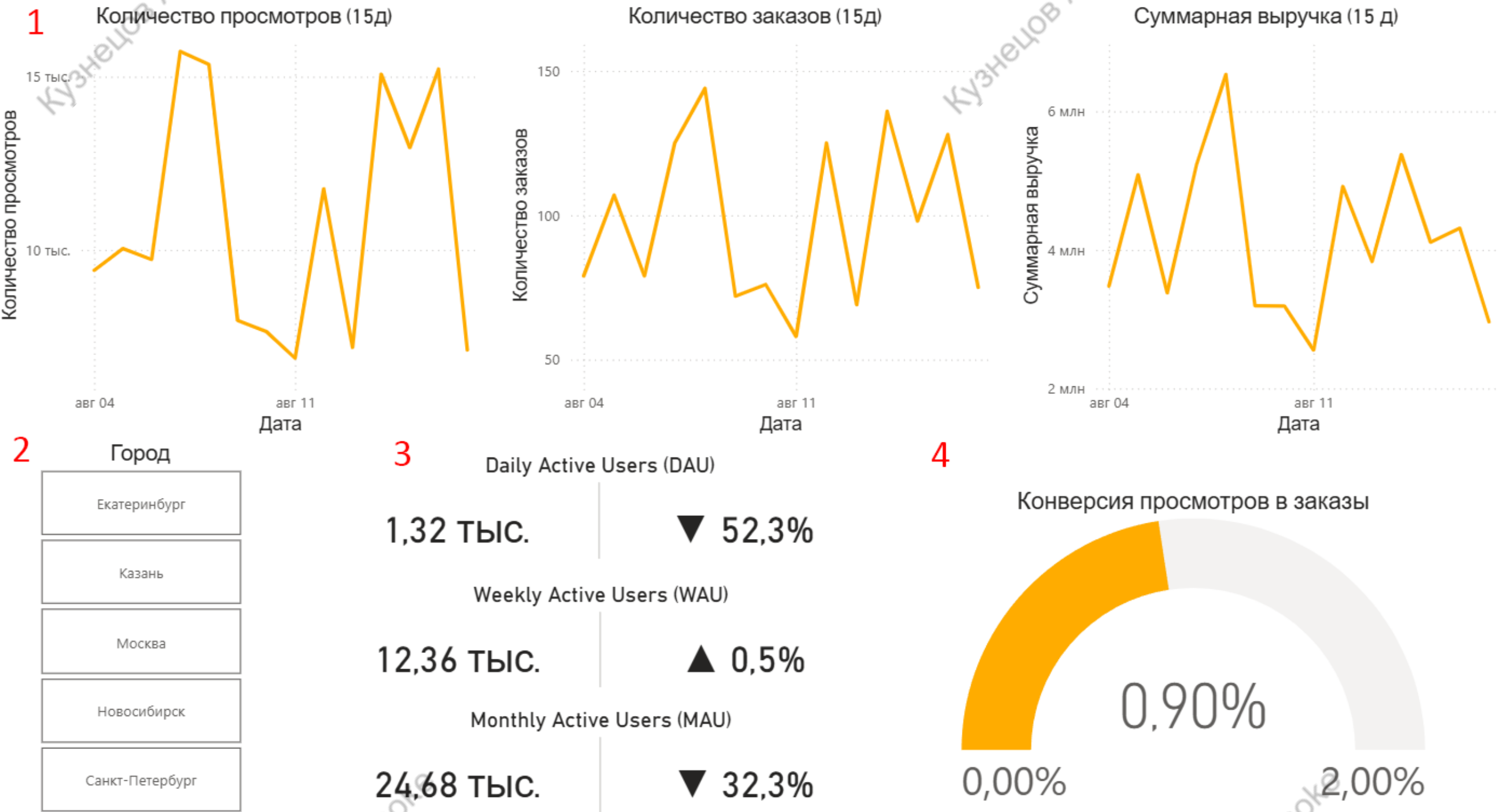


Рисунок 7. Первая страница дашборда

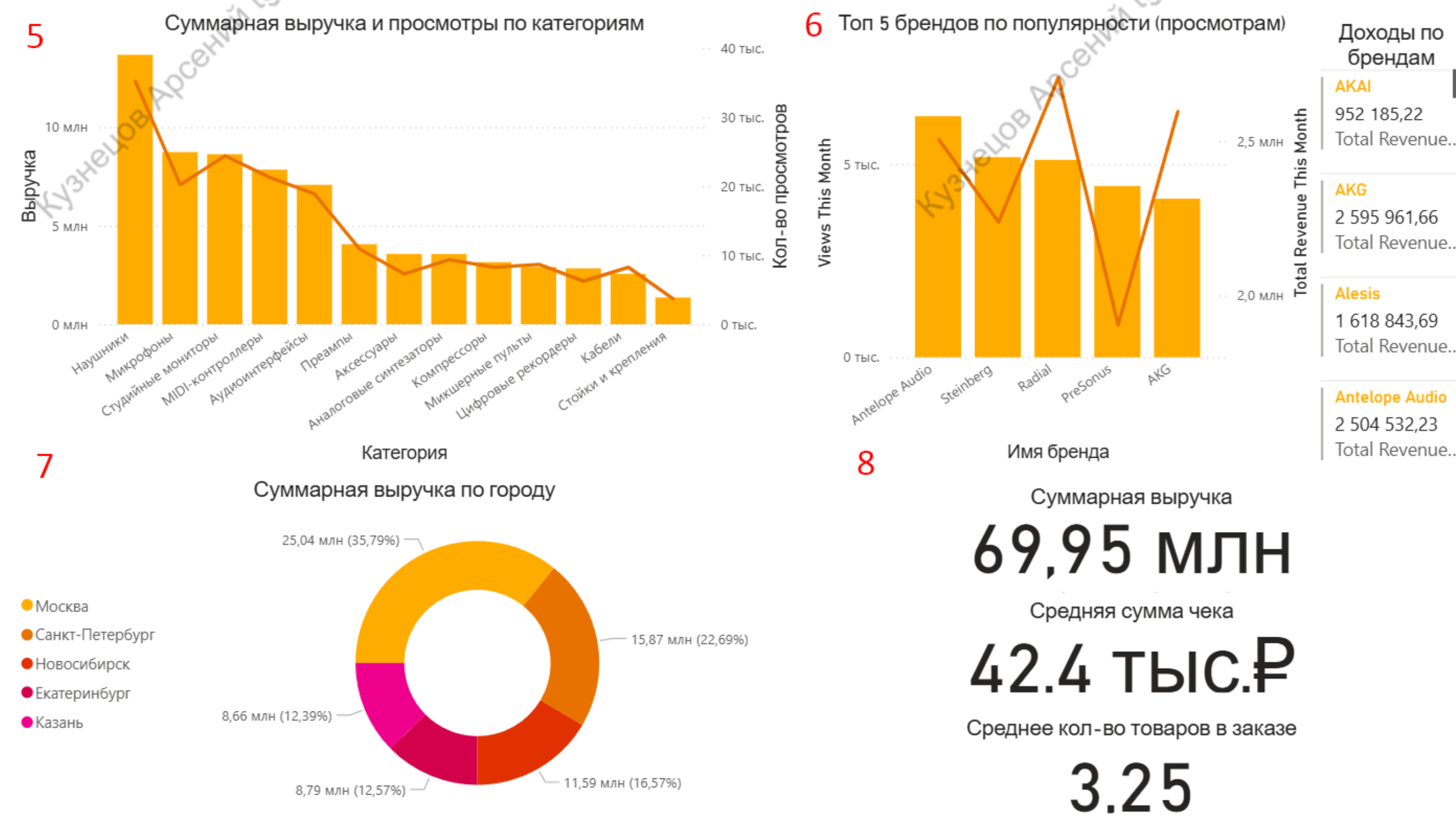


Рисунок 8. Вторая страница дашборда

Дашборд был разработан в Power BI и состоит из двух страниц. Для удобства описания и навигации информационные блоки на дашборде выделены красными номерами:

Блок №1 – Три линейных графика, предоставляющих пользователю информацию о том, какое кол-во просмотров, заказов и какая выручка была за день в период предыдущих 15 дней.

Блок №2 – Кнопки-фильтры, позволяющие отслеживать данные из блоков №1 и №4 по городам.

Блок №3 – Три ключевых метрики необходимых для анализа работы онлайн-магазина:

- 1. DAU – кол-во уникальных пользователей за день
- 2. WAU – кол-во уникальных пользователей за неделю
- 3. MAU – кол-во уникальных пользователей за месяц

Также есть лейблы, показывающие процентное изменение данных метрик по сравнению с предыдущим периодом.

Блок №4 – Коэффициент конверсии просмотров в заказы за текущий месяц.

Блок №5 – Столбчатая диаграмма, на которой можно отслеживать популярность категорий товаров, судя по выручке (столбцы) и просмотрам (линия) конкретных категорий. Данные формируются за текущий календарный месяц.

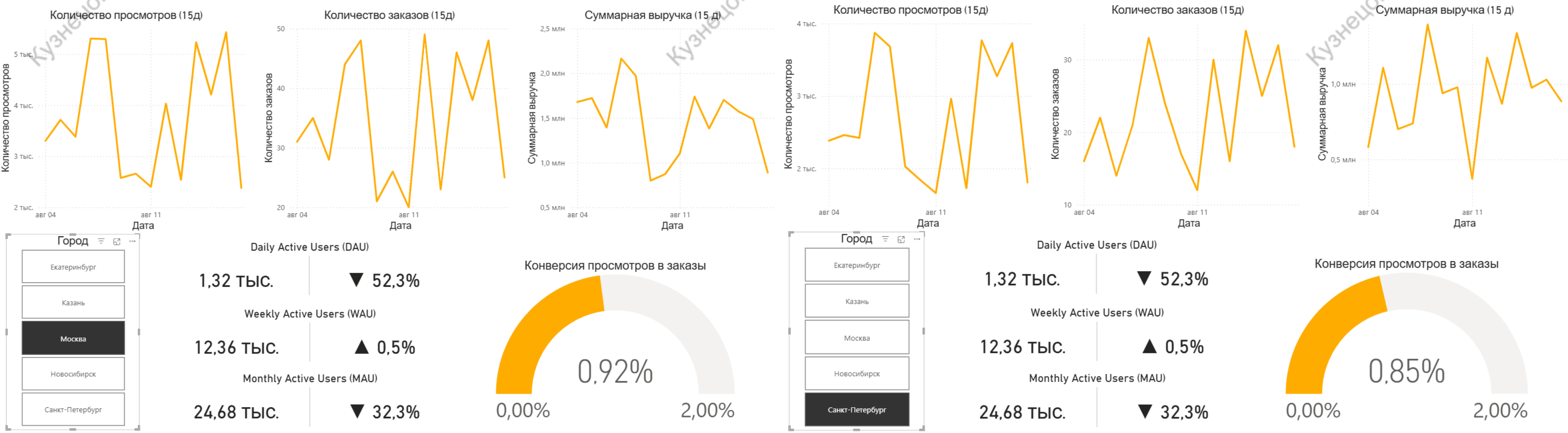
Блок №6 – Столбчатая диаграмма с информацией о топ-5 брендах по популярности (наибольшее кол-во просмотров - столбцы) и выручке от конкретного бренда (линия + скроллинговая панель). Данные формируются за текущий календарный месяц.

Блок №7 – Кольцевая диаграмма, показывающая выручку от каждого города функционирования магазина. Данные формируются за текущий календарный месяц.

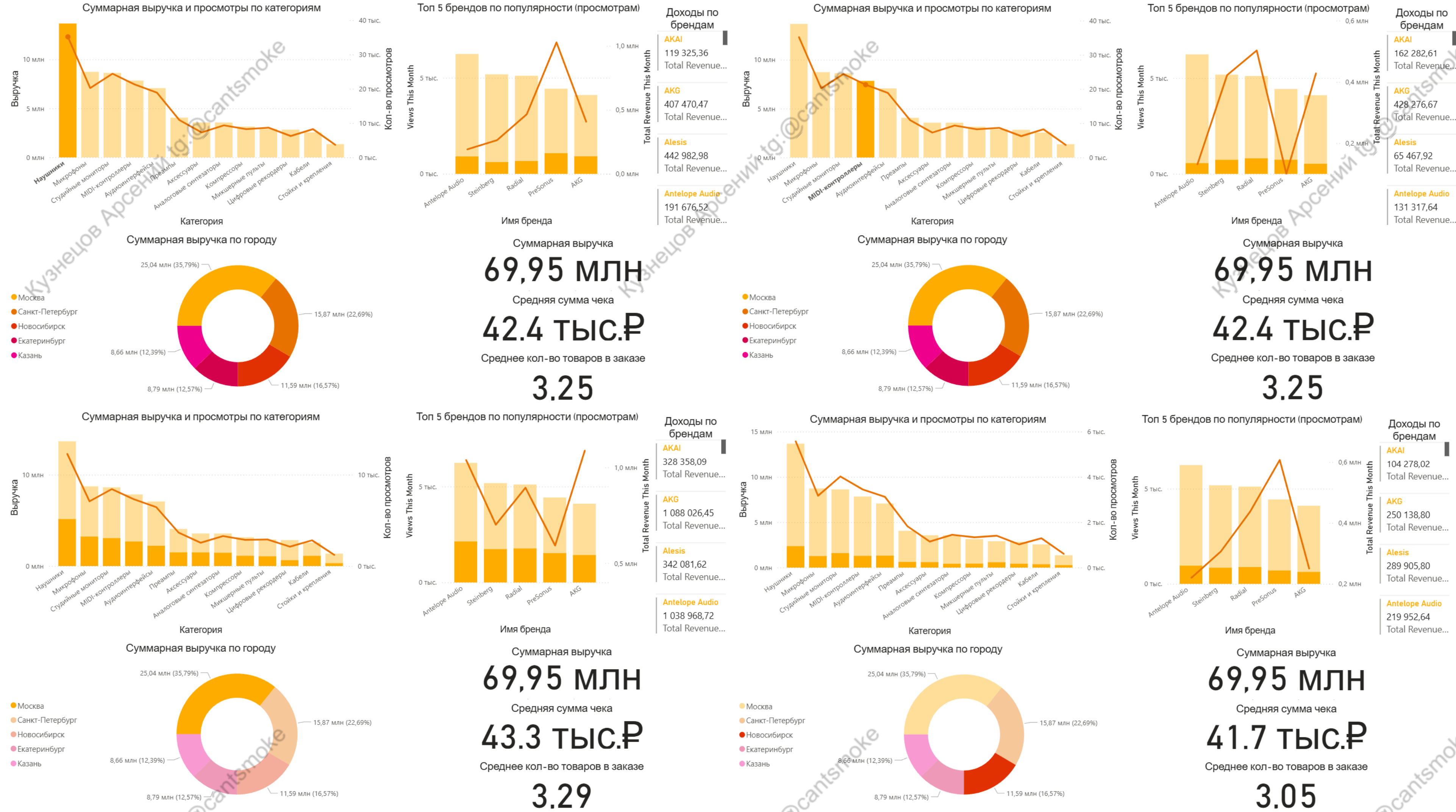
Блок №8 – Три лейбла с простыми метриками:

- 1. Суммарная выручка за текущий месяц
- 2. Средняя сумма чека в текущем месяце
- 3. Среднее кол-во товаров в заказе в текущем месяце

Примеры интерактивной филтрации по городам на первой странице:



Примеры интерактивной фильтрации по категориям, и городам на второй странице:



Описание alert'ов

Система алертов в рамках данного проекта представляет собой механизм автоматического мониторинга ключевых метрик онлайн-магазина и уведомления ответственных пользователей о выявленных аномалиях. Основная цель подсистемы - оперативное обнаружение критических отклонений в пользовательской активности и финансовых показателях, что позволяет своевременно принимать управленческие решения.

Алерты реализованы в виде Python-скрипта, который запускается по расписанию с помощью оркестратора Apache Airflow. На каждом шаге моделирования (одна минута соответствует одному дню симуляции) скрипт извлекает из базы данных PostgreSQL агрегированные показатели за сгенерированный день. Среди контролируемых метрик выделяются:

- DAU (Daily Active Users) - количество уникальных пользователей, совершивших активность за сутки;
- Conversion Rate (просмотр → заказ) - коэффициент конверсии из просмотров товаров в заказы;
- Orders - количество заказов, созданных за сутки;
- Revenue - суммарная выручка за день.

Для каждой метрики установлены пороговые значения. Например, DAU должен превышать 1250 пользователей, коэффициент конверсии - 0,8%, количество заказов - 60, а дневная выручка - 3 млн рублей. Если фактические показатели оказываются ниже установленных границ, генерируется предупреждающее сообщение. Все выявленные аномалии формируются в единый текстовый отчет, который автоматически отправляется в Telegram через бота.

Дополнительно предусмотрено логирование алертов: каждая сработавшая проверка фиксируется в таблице alert_logs базы данных. Запись включает дату проверки, тип метрики, текст уведомления, фактическое значение, порог и время фиксации. Это позволяет сохранять историю срабатываний для последующего анализа, построения отчетов или применения методов ретроспективной аналитики.

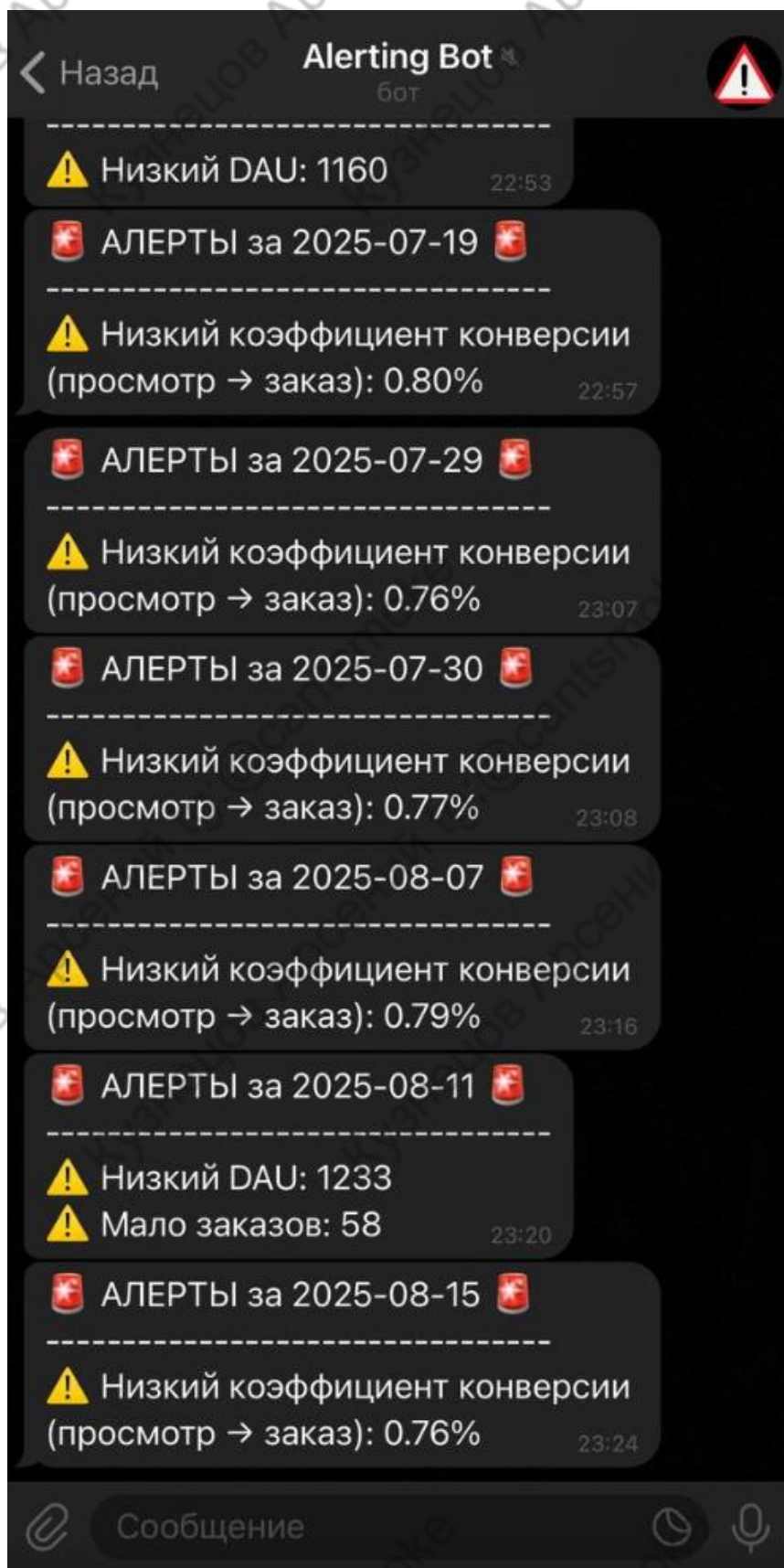


Рисунок 9. Примеры полученных алертов

Описание отчетов

Подсистема отчетности обеспечивает автоматизированное формирование аналитических отчетов о ключевых бизнес-показателях и их доставку заинтересованным пользователям. Основная задача отчетов заключается в предоставлении информации о состоянии онлайн-магазина за прошедший период в наглядной форме, что упрощает контроль динамики показателей и позволяет своевременно выявлять тенденции.

Отчеты реализованы в формате PDF и формируются Python-скриптом, который запускается оркестратором Apache Airflow раз в семь симулированных дней. Скрипт обращается к базе данных PostgreSQL и агрегирует данные за текущий недельный период. В отчет включаются следующие ключевые блоки:

- WAU (Weekly Active Users) - число уникальных пользователей за неделю и динамика относительно предыдущего периода.
- Total Revenue - суммарная недельная выручка.
- Orders (7d) - количество заказов по дням недели (линейный график).
- Revenue (7d) - динамика выручки по дням недели (линейный график).
- DAU (7d) - количество уникальных пользователей по дням (линейный график).
- Revenue by City - распределение недельной выручки по городам (столбчатая диаграмма).

Готовые отчеты сохраняются в файловой системе в специализированных папках, структурированных по периодам (например, 2025-08-01_to_2025-08-07). Каждому отчету присваивается уникальное имя, содержащее даты охвата.

Дополнительно реализована функция доставки отчетов по запросу через Telegram-бота. Пользователь может вызвать список доступных отчетов, выбрать нужный период и получить файл напрямую в мессенджере.

Таким образом, подсистема отчетности выполняет функцию централизованного инструмента аналитического контроля, интегрируясь с ETL-процессом и алертами, а также обеспечивая автоматизацию подготовки материалов для последующего анализа и принятия управленческих решений.

Пример сформированного отчета:

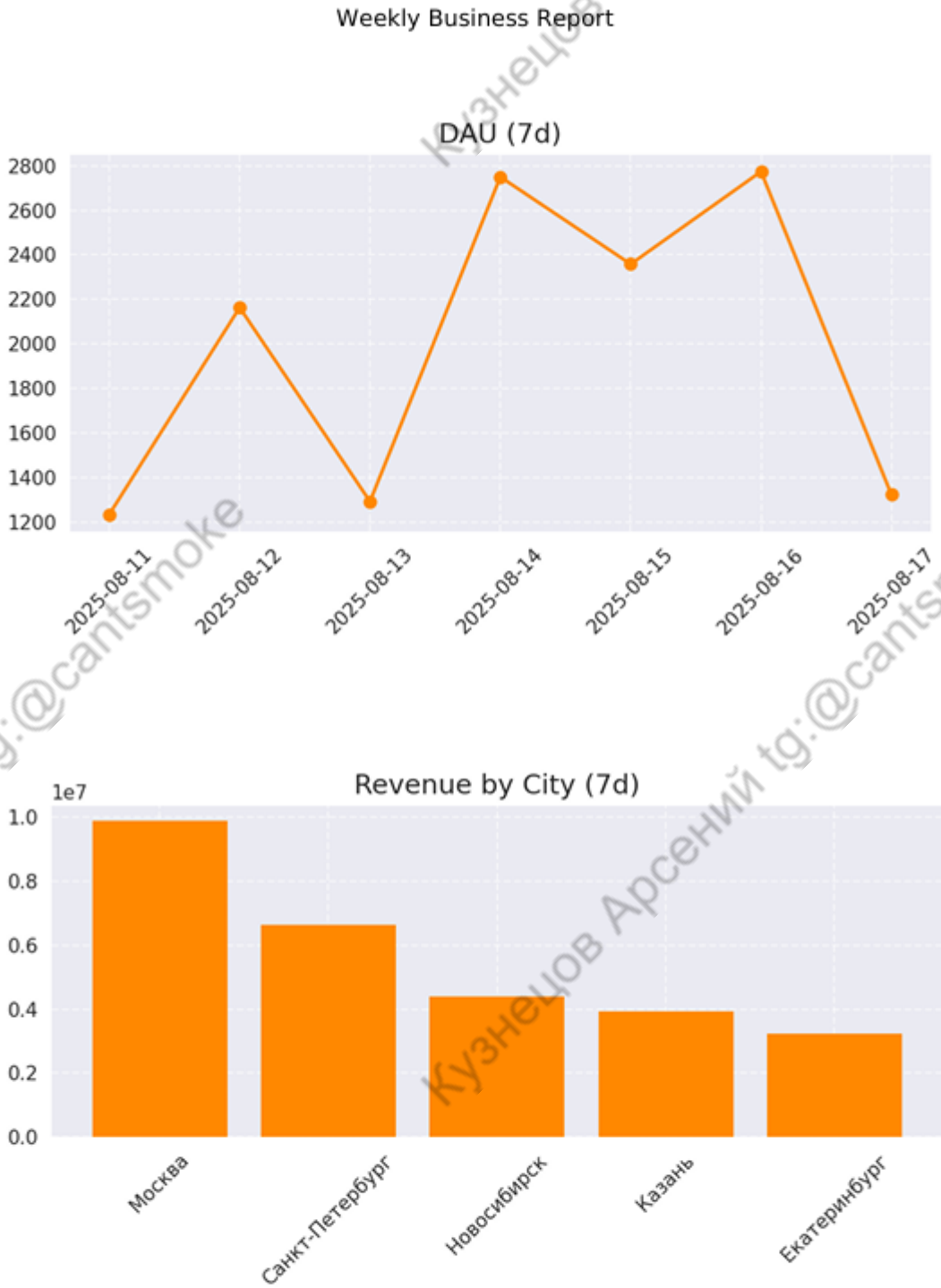


Рисунок 10. Первая (слева) и вторая (справа) страницы отчет

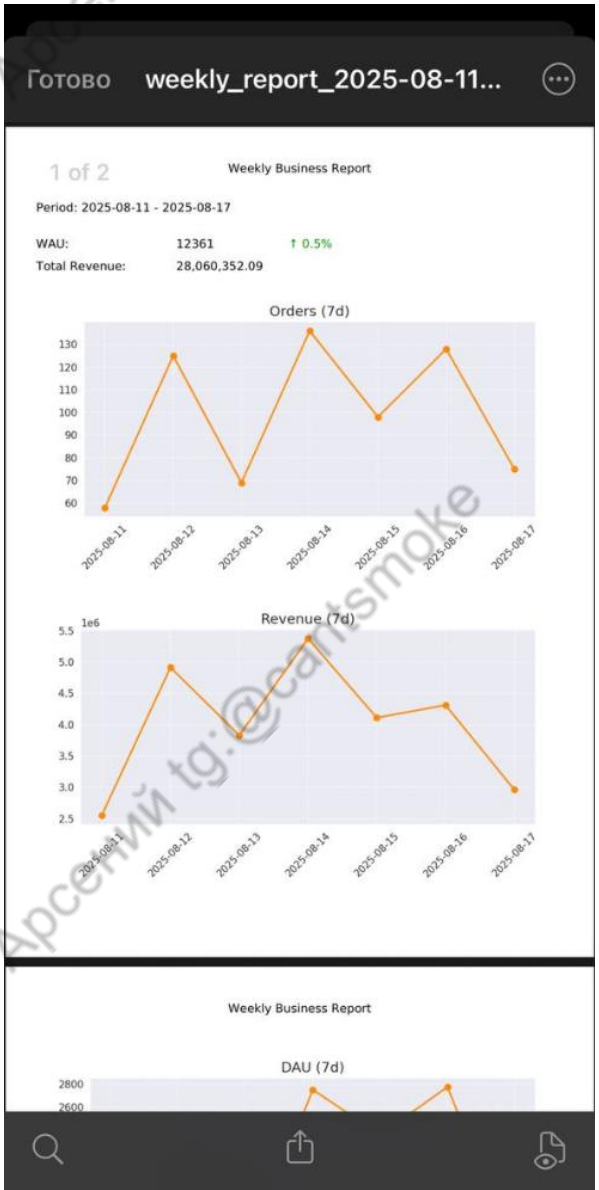
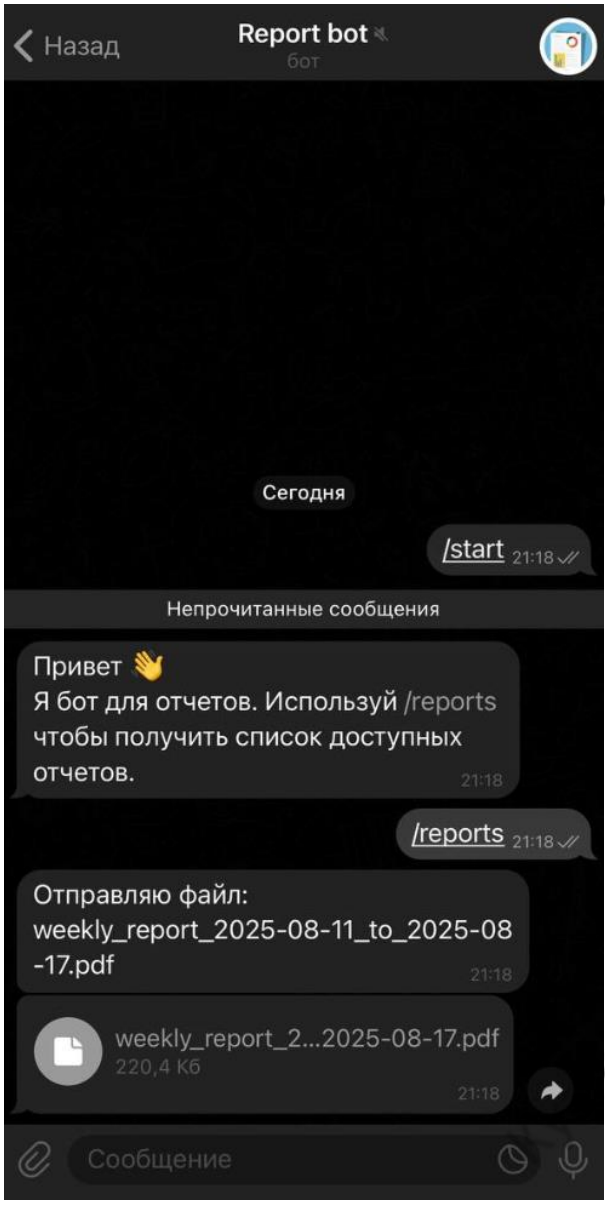
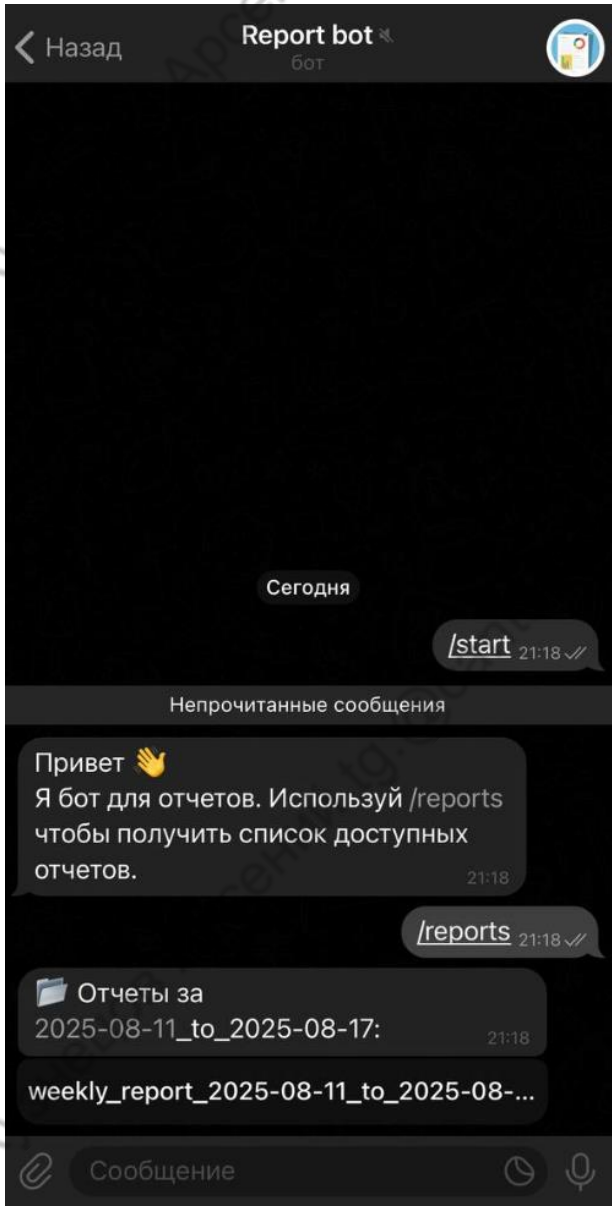
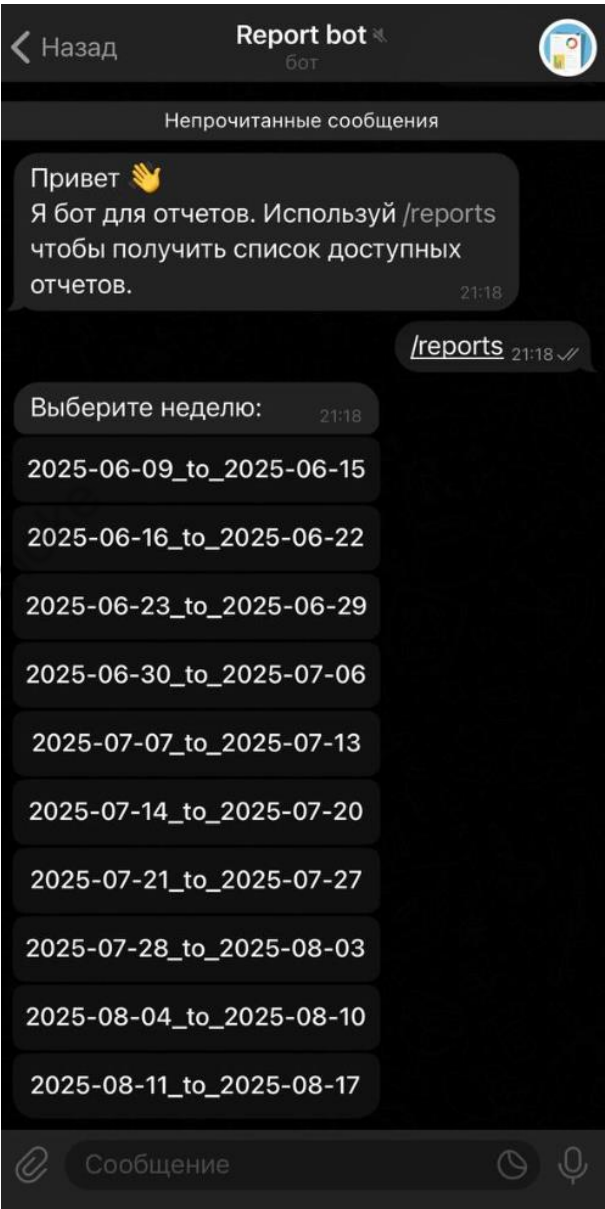


Рисунок 11. Этапы получения отчета через телеграм бота

Заключение

В рамках данного проекта был реализован end-to-end аналитический пайплайн, включающий в себя построение базы данных, генерацию синтетических данных, автоматизированные процессы ETL, построение дашбордов, систему алертов и подсистему отчетности. Благодаря оркестрации процессов через Apache Airflow удалось обеспечить строгое расписание выполнения задач, что позволило синхронизировать ежедневную генерацию данных, контроль ключевых метрик и формирование недельных аналитических отчетов.

Результатом стала комплексная система мониторинга активности пользователей и финансовых показателей онлайн-магазина. Интеграция с Telegram позволила оперативно доставлять оповещения и отчеты, а также реализовать удобный интерфейс для запроса отчетности в интерактивном формате.

Таким образом, проект продемонстрировал полный цикл построения аналитической системы - от моделирования и наполнения базы данных до визуализации и автоматизации контроля бизнес-метрик. Полученное решение можно рассматривать как универсальный шаблон для внедрения в реальных сценариях электронной коммерции, где важны прозрачность, оперативность и автоматизация аналитики.

Примечания

Ссылка на репозиторий проекта:

<https://github.com/cantsmoke/ecommerce-analytics-pipeline>