GeekBrains

**ДИПЛОМНАЯ РАБОТА**

**Автоматизированная работа с данными:   
интеграция APACHE AirFlow для обработки   
и анализа ежемесячных отчетов компании**

Программа: Разработчик

Специализация: Data Engineer

Кузина Полина Сергеевна

Нижний Новгород

2024

**Содержание**

Введение 1

ГЛАВА 1. ИССЛЕДОВАНИЕ актуальности ПРОЦЕССА   
ОБРАБОТКИ ДАННЫХ компании 2

1.1 Данные компании. Важность их обработки и анализа 2

1.2 Структура бизнес-процесса по обработке данных в компании 5

1.3 Современные методы автоматизации процесса обработки данных 6

1.4 Выводы по разделу 7

ГЛАВА 2. ПОДГОТОВКА К ИНТЕГРАЦИИ ИНСТРУМЕНТОВ АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ КОМПАНИИ 8

2.1 Анализ текущей ситуации обработки данных в компании 8

2.2 Дизайн обновленной структуры бизнес-процесса с применением инструментов автоматизированной обработки и анализа данных 11

2.3 Выбор инструментов автоматизации обработки и анализа данных 14

2.4 Выводы по разделу 23

ГЛАВА 3. ПРОЦЕСС НАСТРОЙКИ АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ КОМПАНИИ 24

3.1 Подготовка исходного файла реестра к дальнейшей работе 24

3.2 Загрузка подготовленного датасета в базу данных 31

3.3 Создание чат-бота в Telegram для удобного получения   
аналитических данных 34

3.4 Активация обработки и анализа данных через Apache AirFlow   
и интеграция отправки сообщений в Телеграмм 36

3.5 Подведение итогов 51

ЗАКЛЮЧЕНИЕ 52

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 54

ПРИЛОЖЕНИЯ 55

**Введение**

**Информация по проекту:** Дипломная работа представляет собой документ из 60 страниц, 48 рисунков и 3 приложений.

**Тема проекта:** Автоматизированная работа с данными: интеграция Apache AirFlow для обработки и анализа ежемесячных отчетов компании.

**Цель:** Оптимизировать структуру бизнес-процесса по обработке ежемесячных отчетов, настроить автоматизированную обработку данных и предоставление краткой аналитической информации напрямую руководителю.

**Какую проблему решает:** Автоматизация рутинных повторяющихся задач по аналитике данных позволить высвободить ценное рабочее время сотрудников, исключить ошибки в расчетах и наладить конфиденциальную передачу информации напрямую руководителю.

**Задачи:**

1. Изучить структуру бизнес-процесса.
2. Выявить ключевые этапы оптимизации и автоматизации процесса.
3. Изучить литературу и материалы, касающиеся темы.
4. Разработать методы решения поставленной задачи.
5. Внедрить необходимые инструменты.
6. Протестировать внедренные изменения.
7. Разработать рекомендации и предложения по дальнейшему поддержанию и улучшению процесса.

**Инструменты:** Anaconda, Jupiter Notebook, WSL Ubuntu, MobaXterm, Apache Airflow, IDLE, HeidiSQL, Python, SQL

**Состав команды:** Кузина Полина Сергеевна (Data Engineer)

**ГЛАВА 1. ИССЛЕДОВАНИЕ актуальности ПРОЦЕССА ОБРАБОТКИ ДАННЫХ компании**

1. **Данные компании. Важность их обработки и анализа**

Любая компания в текущее время сталкивается с необходимостью хранения, систематизации и анализа данных.

В компании могут быть различные виды данных, которые классифицируются по разным критериям. Ниже приведены основные виды данных, которые могут быть в компании:

1. Структурированные данные: это данные, организованные в определенном формате или структуре, обычно хранятся в базах данных. Например, это могут быть таблицы с клиентской информацией, финансовые отчеты, списки сотрудников и их контактной информацией и т.д.

2. Неструктурированные данные: это данные, которые не имеют фиксированной структуры и не могут быть легко организованы в таблицы или базы данных. Например, это могут быть текстовые документы, видео- и аудиофайлы, электронная почта, сообщения в социальных сетях и др.

3. Полуструктурированные данные: это данные, которые имеют частично определенную структуру, но не полностью соответствуют структурированным данным. Например, это могут быть форматы файлов XML, JSON, CSV, логи серверов и прочее.

4. Операционные данные: это данные, которые используются для повседневной операционной деятельности компании. Например, данные о заказах, продажах, поставках, клиентах, сотрудниках и т.д.

5. Аналитические данные: это данные, которые используются для анализа и принятия управленческих решений. Например, данные о ключевых показателях производительности, отчеты по финансовым показателям, данные о выполнении задач и проектов и т.д.

6. Персональные данные: это данные, относящиеся к конкретным лицам, такие как имена, адреса, телефоны, электронные адреса, данные о зарплате и т.д. Эти данные требуют особой защиты и обработки в соответствии с законодательством о защите персональных данных.

В зависимости от специфики бизнеса и отрасли, в которой работает компания, могут быть и другие виды данных, которые также могут быть важны для бизнеса.

Обработка данных является важным аспектом деятельности компаний и представляет собой процесс преобразования и анализа информации с целью получения ценных знаний и улучшения бизнес-процессов. Вот некоторые причины, по которым компании обрабатывают данные:

**Принятие управленческих решений:** Обработка данных помогает компании собирать, анализировать и интерпретировать информацию о своей деятельности, клиентах, рынке и конкурентах. На основе этих данных руководство может принимать обоснованные стратегические и операционные решения.

**Улучшение операционной эффективности:** Обработка данных позволяет автоматизировать бизнес-процессы, оптимизировать производственные процессы, управлять запасами, улучшать качество продукции и услуг, сокращать издержки и повышать производительность труда.

**Улучшение взаимодействия с клиентами:** Обработка данных о клиентах позволяет компаниям лучше понимать их потребности, предпочтения и поведение, персонализировать предложения, улучшить качество обслуживания, увеличить лояльность клиентов и увеличить выручку.

**Маркетинг и реклама:** Обработка данных помогает компаниям анализировать данные о рынке, целевой аудитории, эффективности рекламных кампаний, прогнозировать спрос, управлять маркетинговыми бюджетами, оптимизировать маркетинговые стратегии и увеличивать продажи.

**Соответствие законодательству:** Обработка данных включает в себя соблюдение законодательства о защите персональных данных, конфиденциальности информации, прав потребителей и других правовых норм, что позволяет компаниям избежать штрафов и судебных преследований.

Обработка данных помогает компаниям быть конкурентоспособными, адаптироваться к изменяющимся условиям рынка, улучшать качество продукции и услуг, повышать эффективность бизнес-процессов и улучшать отношения с клиентами.

1. **Структура бизнес-процесса по обработке данных в компании**

Можно выделить следующие этапы по работе с данными в компании:

**Сбор данных:** на этом этапе происходит сбор всех необходимых данных, которые потребны компании для дальнейшей обработки. Данные могут поступать из различных источников, таких как базы данных, сенсоры, интернет и т.д.

**Хранение данных:** после сбора данных они обычно хранятся в базе данных или хранилище данных. Это позволяет хранить данные в структурированном виде и легко получать к ним доступ при необходимости.

**Обработка данных:** на этом этапе происходит обработка данных с целью извлечения информации и знаний из них. Обработка может включать в себя различные операции, такие как фильтрация, агрегация, преобразование и т.д.

**Анализ данных:** после обработки данных происходит их анализ с целью выявления паттернов, трендов и закономерностей. Анализ данных позволяет принимать обоснованные решения на основе данных.

**Визуализация данных:** результаты анализа данных обычно визуализируются с помощью различных графических инструментов, таких как диаграммы, графики, дашборды и т.д. Визуализация помогает лучше интерпретировать данные и делиться результатами с другими сотрудниками.

**Принятие решений:** на основе обработанных и проанализированных данных руководство компании принимает решения по стратегическим вопросам, оптимизации бизнес-процессов, улучшению качества продукции или услуг и т.д. Все эти этапы взаимосвязаны и взаимодействуют друг с другом, обеспечивая эффективную работу бизнес-процесса по обработке данных компании. Важно также учитывать вопросы безопасности данных и их конфиденциальности на всех этапах обработки.

1. **Современные методы автоматизации процесса обработки данных**

В настоящее время существует множество технологий, которые позволяют автоматизировать обработку данных:

**Системы управления базами данных (СУБД):** СУБД, такие как MySQL, PostgreSQL, Oracle, Microsoft SQL Server и другие, предоставляют средства для хранения, управления и обработки данных.

**ETL (Extract, Transform, Load) инструменты:** Эти инструменты позволяют извлекать данные из различных источников (таких как базы данных, файлы, API), преобразовывать их в нужный формат и загружать в целевую базу данных или хранилище данных.

**Автоматизация бизнес-процессов (Business Process Automation, BPA) инструменты:** Эти инструменты позволяют автоматизировать повседневные бизнес-процессы, включая обработку данных, согласование документов, управление задачами и многое другое.

**Business Intelligence (BI) и аналитические инструменты:** Инструменты BI, такие как Tableau, Power BI, QlikView и другие, помогают визуализировать данные, создавать отчеты и дашборды для анализа информации и принятия решений.

**Машинное обучение и искусственный интеллект:** Технологии машинного обучения и искусственного интеллекта могут быть использованы для автоматизации обработки данных, создания прогнозов, классификации данных, распознавания образов и многое другое.

**Облачные вычисления и хранилища данных:** Облачные платформы, такие как AWS, Google Cloud Platform, Microsoft Azure и др., предлагают готовые инструменты и сервисы для обработки, хранения и анализа данных.

Эти технологии позволяют компаниям эффективно обрабатывать большие объемы данных, улучшать производительность и качество работы, сокращать временные затраты, повышать точность анализа и принятия решений.

1. **Выводы по разделу**

Автоматизированная обработка данных обладает множеством преимуществ, включая повышение эффективности и точности работы за счет исключения человеческих ошибок, ускорение процессов обработки и анализа данных, снижение затрат времени и ресурсов, увеличение производительности и оперативности в принятии решений, а также возможность работы с большими объемами данных и автоматическое обновление информации в режиме реального времени, что позволяет компаниям быть более конкурентоспособными и адаптивными на рынке.

Выбор технологии автоматизации обработки данных зависит от конкретных потребностей и целей компании.

Прежде всего, необходимо определить объем данных, их источники, виды обработки и анализа, а также требования к безопасности и скорости обработки.

Затем следует изучить доступные на рынке инструменты и платформы для автоматизации данных, провести их сравнительный анализ по критериям соответствия вашим требованиям, стоимости и сложности внедрения.

Важно также учитывать наличие поддержки и обучения со стороны поставщика, а также возможность масштабирования и интеграции с существующими системами компании.

После анализа всех этих факторов можно выбрать оптимальную технологию, которая наилучшим образом подойдет для автоматизации обработки данных.

**ГЛАВА 2. ПОДГОТОВКА К ИНТЕГРАЦИИ инструментов АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ КОМПАНИИ**

**2.1** **Анализ текущей ситуации обработки данных в компании**

В рамках дипломной работы я рассматриваю процесс обработки и анализа данных по реестру выполненных задач компании, оказывающей дизайнерские услуги крупному бизнесу.

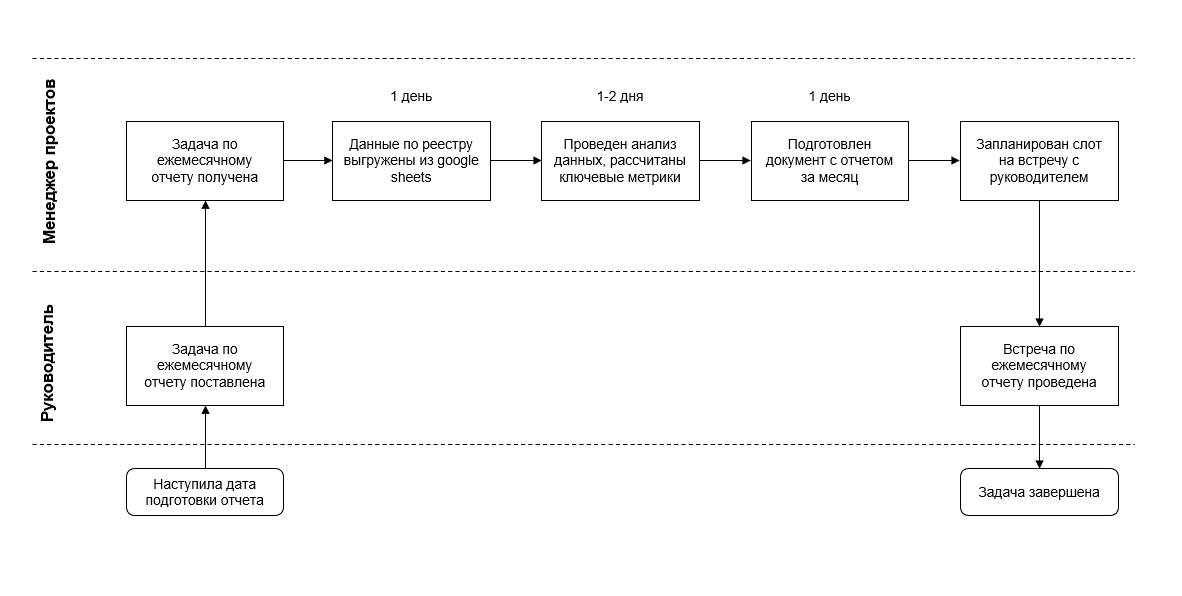
На данный момент в компании не применяются автоматизированные системы обработки и анализа данных. Структура бизнес-процесса в настоящее время выглядит следующим образом:

1. Данные по выполняемым задачам вручную заносятся в реестр, созданный в google sheets
2. По завершении месяца менеджеры вручную анализируют полученные результаты, выявляя следующие метрики:

* Выручка за период
* Суммарное количество задач
* Сумма среднего чека
* Среднее время выполнения задачи
* Средняя производительность

1. После анализа данных менеджеры лично общаются с руководителем и передают ему данные по итогам месяца.

Текущая схема бизнес-процесса компании представлена на Рисунке 1, а также в Приложении 1.



*Рисунок 1. Структура бизнес-процесса компании по обработке  
ежемесячных отчетов*

Такая организация бизнес-процесса может нести ряд рисков и негативно сказываться на качестве принимаемых решений. Вот несколько минусов ручной обработки данных:

**Ошибка человеческого фактора:** При ручной обработке данных возникает риск возникновения ошибок, так как оператор часто может допустить опечатки, упустить важные детали или неправильно интерпретировать данные.

**Низкая скорость обработки:** Ручная обработка требует большого количества времени и усилий, особенно при обработке больших объемов данных или повторяющихся операций.

**Ограниченная масштабируемость:** Ручная обработка данных не масштабируется хорошо и может стать проблематичной при увеличении объемов и сложности данных.

**Отсутствие стандартизации:** Без автоматизации процессов могут возникнуть проблемы с недостаточной стандартизацией данных и процессов, что затрудняет их обработку и анализ.

**Высокие затраты:** Ручная обработка данных требует большого количества человеческих ресурсов, что может привести к дополнительным расходам на оплату персонала.

**Недостаточная безопасность:** При ручной обработке данные могут быть подвержены утечкам или несанкционированному доступу, так как они могут храниться на не защищенных носителях или передаваться через небезопасные каналы связи.

Все эти недостатки могут быть преодолены с помощью автоматизации процессов обработки данных с использованием специальных технологий и инструментов.

**2.2 Дизайн обновленной структуры бизнес-процесса с применением инструментов автоматизированной обработки и анализа данных**

Для автоматизации процесса обработки и анализа данных применяем инструменты ETL.

ETL — это процесс, который используется для передачи данных из одной системы в другую, обычно из источника данных в базу данных для анализа и обработки.

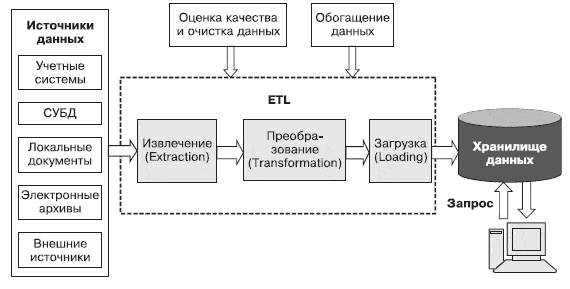
Этот процесс состоит из трех этапов:

- Извлечение (Extraction): Данные извлекаются из источника данных, например, из файла Excel, базы данных или API.

- Трансформация (Transformation): Данные преобразуются и очищаются для обеспечения их качества. Например, удаляются дубликаты, заполняются пропущенные значения или объединяются несколько таблиц данных.

- Загрузка (Loading): Преобразованные и очищенные данные загружаются в целевую базу данных или хранилище данных для последующего анализа и использования.

В данной ситуации идея применения ETL в том, чтобы обеспечить соответствие данных требованиям бизнеса, сделать данные доступными и простыми для анализа и использования. Этот процесс является важным шагом в обработке данных и помогает организациям принимать основанные на фактах решения.



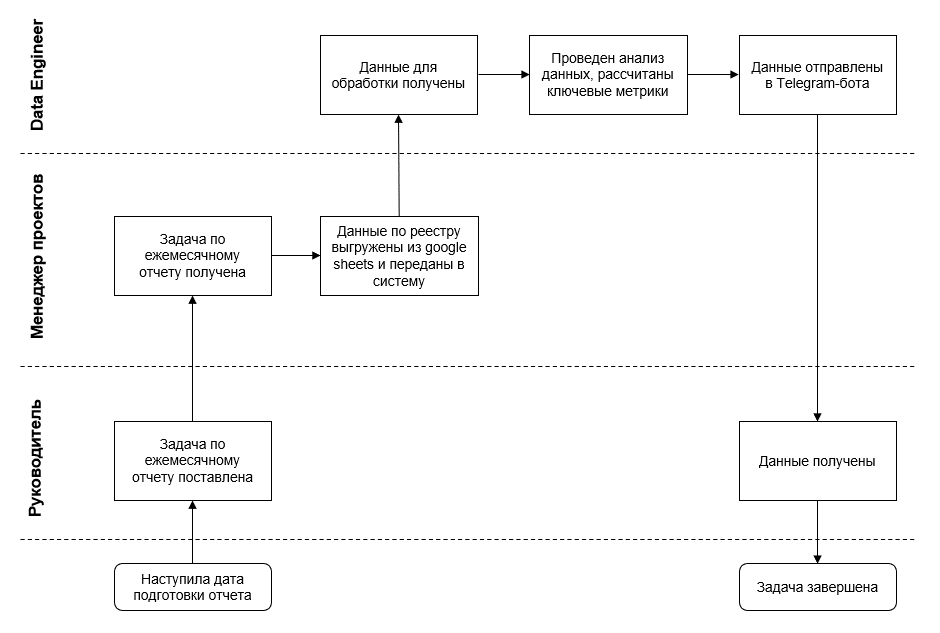
*Рисунок 2. Структура ETL процессов*

Далее подключаем инструменты аналитики и проводим анализ данных с фиксацией результата по необходимым метрикам. Подготовленные аналитические данные передаем заказчику в удобном виде.

На основе анализа текущей организации процесса обработки данных в компании предлагаю следующие обновления в структуре бизнес-процесса:

1. Извлечение (Extraction): Выгрузка данных из google sheets
2. Трансформация (Transformation): Автоматизированная подготовка данных к дальнейшему анализу
3. Загрузка (Loading): Загрузка подготовленных данных в систему управления рабочими процессами
4. Анализ: Анализируем данные по необходимым метрикам
5. Выгрузка отчета: Автоматизированная адресная передача аналитических данных

Обновленная схема бизнес-процесса компании представлена на Рисунке 3, а также в Приложении 2.



*Рисунок 3. Обновленная структура бизнес-процесса компании   
по обработке ежемесячных отчетов*

Обновленная структура бизнес-процесса позволит минимизировать риски, связанные с человеческим фактором, ускорить процесс обработки и анализа данных, а также сохранить конфиденциальность передаваемых аналитических отчетов.

К тому же данную систему можно будет масштабировать на другие бизнес-процессы компании, связанные с обработкой и анализом данных.

**2.3 Выбор инструментов автоматизации обработки и анализа данных**

Разберем подробно выбор инструментов для проведения каждого этапа автоматизации бизнес-процесса

**1. Извлечение и подготовка данных**

**Среда разработки:** Для работ на этом этапе выбрана среда Jupiter Notebook. Это среда разработки, которая позволяет:

* писать код в специальной среде с подсветкой синтаксиса, исправлением ошибок и другими возможностями IDE;
* запускать разные участки кода в произвольной последовательности или написанную программу целиком;
* загружать какие-то данные, обрабатывать и преобразовывать их, не затрагивая при этом другие участки программы;
* вставлять и выводить результаты, включая визуализацию, прямо посреди кода;
* делиться кодом с другими разработчиками и давать им общий доступ к проекту;
* организовывать командную работу, когда у каждого программиста — своя задача, связанная с другими;
* писать сопровождающий текст и оформлять документ так, чтобы он выглядел красиво и понятно.

**Язык программирования:** Для подготовки, обработки и последующего анализа данных выбран язык программирования Python.

Python — это интерпретируемый, высокоуровневый объектно-ориентированный язык общего назначения, используемый для разработки API, искусственного интеллекта, веб-разработки, интернета вещей и так далее.

Python предлагает множество библиотек, которые применяются на всех этапах анализа данных. Поэтому язык однозначно подходит для этих целей.

Для анализа данных в Jupyter Notebook можно использовать следующие библиотеки: pandas, numpy, matplotlib, seaborn.

**2. Загрузка данных**

Для дальнейшей работы с данными нам необходимо загрузить подготовленный датасет в базу данных.

**Среда разработки:** Для загрузки данных используем среду разработки IDLE.

IDLE - это интегрированная среда разработки (IDE) для языка программирования Python. IDLE предоставляет пользователю удобное программное обеспечение для написания, отладки и выполнения программ на Python.

IDLE включает в себя функции, такие как подсветка синтаксиса, автодополнение, отладчик, интерактивная оболочка Python и другие инструменты, которые облегчают разработку приложений на Python.

IDLE поставляется вместе с установкой Python и может быть использован как начинающими программистами, так и опытными разработчиками для создания разнообразных программ на Python.

**Язык программирования:** Для загрузки данных выбран язык программирования Python.

**База данных:** Для выбора базы данных для хранения и обработки датасетов необходимо учитывать несколько ключевых факторов, таких как тип данных, объем информации, скорость доступа, требования к безопасности и масштабируемость системы. Также важно учитывать стоимость реализации и поддержки базы данных, а также уровень опыта сотрудников при работе с конкретными технологиями.

В текущем проекте для хранения и последующего доступа к данным используется HeidiSQL

HeidiSQL — это бесплатный и открытый исходный код клиент базы данных, предназначенный для администрирования и управления различными базами данных, такими как MySQL, MariaDB, PostgreSQL и Microsoft SQL Server.

HeidiSQL предоставляет удобный графический интерфейс пользователя для выполнения различных операций с базами данных, таких как создание и изменение таблиц, выполнение SQL-запросов, экспорт и импорт данных, управление пользователями и привилегиями, а также выполнение административных задач.

Основные функции HeidiSQL включают поддержку множества типов баз данных, работу с несколькими соединениями одновременно, автоматическое заполнение и подсветку синтаксиса SQL, а также возможность сохранения истории запросов. HeidiSQL является популярным инструментом среди разработчиков и администраторов баз данных для управления и мониторинга баз данных.

**3. Автоматизированная обработка и анализ данных**

**Платформа обработки и мониторинга:** Для автоматизации процессов работы с данными выбран инструмент Apache AirFlow

Airflow — это платформа с открытым исходным кодом для разработки, планирования и мониторинга рабочих процессов. Эти рабочие процессы могут помочь вам перемещать данные, фильтровать их наборы, применять политики данных, манипулировать, отслеживать и даже вызывать микросервисы для запуска задач управления базой данных.

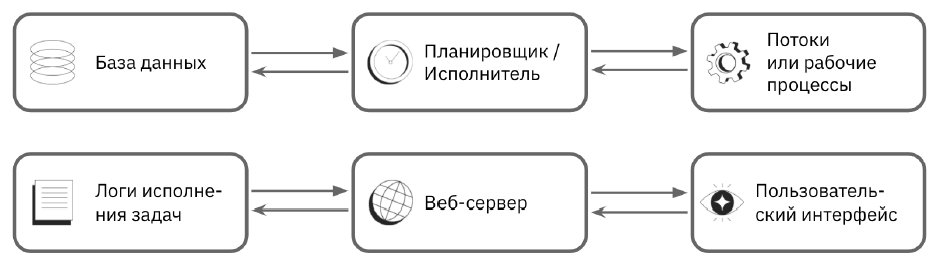
Преимущество Airflow по сравнению с другими похожими инструментами в том, что Airflow позволяет планировать и отслеживать рабочие процессы, а не просто создавать их.

Особенности Airflow:

* Динамический. DAG написаны на Python, что позволяет создавать динамические конвейеры.
* Расширяемый. Легко создавать собственные операторы, исполнители и библиотеки.
* Элегантный. DAG Airflow компактны и понятны. Параметризовать конвейеры можно с помощью шаблонов Jinja.
* Масштабируемый. Airflow модульный, использует очередь сообщений для управления исполнителями, которые выполняют задачи.

Преимущества Airflow

* Открытый исходный код. Следствие — низкая стоимость, инновации и поддержка сообщества.
* Широкая интеграция. Может использоваться в облачных провайдерах большой тройки — AWS, Azure и GCP.
* Пользовательский интерфейс. Позволяет пользователям легко отслеживать конвейеры и устранять неполадки.
* Программный подход. Используя Python, пользователи могут создавать инновационные расширения.
* Повторы. Многие конвейеры данных должны быть настроены для повторных попыток, в Airflow это встроено.
* Предупреждение. Пайплайны в Airflow могут быть на виду, даже когда системы не мониторятся.



*Рисунок 4. Архитектура Apache Airflow*

**Среда разработки:** Для активации доступа к Apache AirFlow на Windows использую WSL Ubuntu.

WSL (Windows Subsystem for Linux) - это среда выполнения Linux, предоставляемая операционной системой Windows. WSL позволяет запускать исполняемые файлы Linux напрямую в Windows без необходимости установки отдельной виртуальной машины или перезагрузки компьютера.

WSL обеспечивает совместимость с ядром Linux, что позволяет пользователям запускать большинство командной строки и приложений Linux непосредственно в Windows.

В качестве терминала использую MobaXterm.

MobaXterm — это многофункциональная программа для удаленного управления и администрирования систем, предлагающая широкий спектр инструментов и функций в одном удобном интерфейсе. Это инструмент, который сделает жизнь системных администраторов, разработчиков и инженеров значительно проще, обеспечивая доступ к удаленным серверам, работу с разнообразными протоколами и инструментами, и многие другие полезные функции.

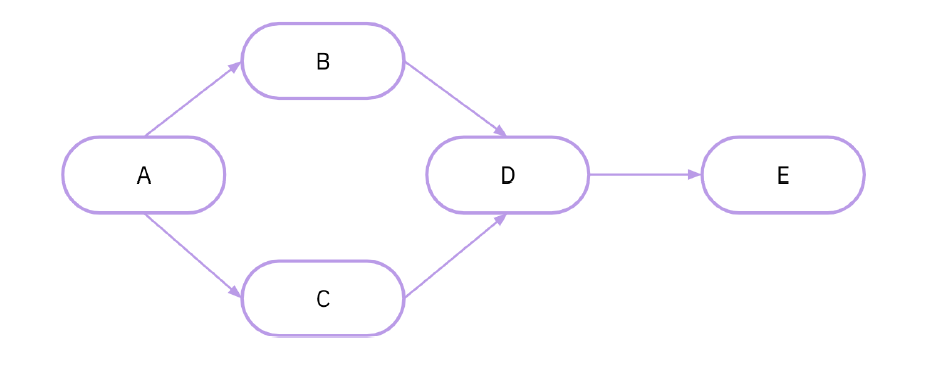
MobaXterm интегрирует в себя множество инструментов и протоколов, что делает его мощным и гибким решением для различных сценариев использования:

* Множество протоколов: Одной из главных особенностей MobaXterm является поддержка множества протоколов, таких как SSH, Telnet, RDP, VNC, FTP, SFTP и X11. Вы можете подключаться к разным типам серверов и устройств с использованием одного инструмента.
* Управление сеансами: MobaXterm позволяет создавать и организовывать сеансы удаленного подключения, облегчая вам работу с несколькими серверами одновременно. Вы можете легко управлять этими сеансами, переключаясь между ними и быстро копируя данные между ними.
* Интегрированный X-сервер: Если вам нужно запускать графические приложения на удаленном сервере, MobaXterm включает в себя встроенный X-сервер, который позволяет вам легко запускать графические программы с удаленных серверов на вашем локальном компьютере.
* Поддержка Unix-команд: MobaXterm включает в себя набор Unix-команд и утилит, что делает его отличным инструментом для администрирования Unix-подобных систем. Вы можете выполнять команды на удаленных серверах и управлять файлами с легкостью.
* Интегрированный терминал: Встроенный терминал MobaXterm обеспечивает множество полезных функций, таких как авто-завершение, подсветка синтаксиса, множество вкладок и возможность настройки внешнего вида.
* Переносимость и простота использования: MobaXterm представляет собой портативное приложение, которое не требует установки, что делает его идеальным выбором для ситуаций, когда у вас нет прав администратора на компьютере. Простой и интуитивный интерфейс делает его доступным для пользователей с разным уровнем опыта.

**Язык программирования:** Для написания операторов AirFlow выбран язык программирования Python.

Операторы Airflow — это команды, выполняемые асинхронным графом (DAG) каждый раз, когда срабатывает нужный триггер в процессе пайплайна.

DAG — это направленный ациклический граф, основа концепции Airflow. DAG состоит из узлов, соединенных направленными ребрами. Одна из самых мощных особенностей Airflow в том, что DAG написаны на Python. Этот факт помогает решить одну из самых больших проблем, с которыми сталкиваются инженеры по данным, — интеграцию различных систем.



*Рисунок 5. Направленный ациклический граф (DAG)*

Для написания запросов по работе данных выбран язык SQL.

SQL (сокращение от англ. Structured Query Language) — это язык запросов, который применяют для работы с базами данных, структурированными особым образом.

Главные задачи SQL — составлять запросы так, чтобы находить среди большого объёма информации ту, что нужна для конкретных целей, сортировать её, структурировать и представлять в наиболее простом и понятном виде.

С помощью SQL можно:

* собирать и хранить данные в виде таблиц;
* изменять их содержимое и структуру;
* объединять данные и выполнять вычисления;
* защищать и распределять доступ.

Язык SQL нужен разработчикам, тестировщикам, аналитикам данных, администраторам, маркетологам — всем тем, кому по работе нужно выгружать и обрабатывать большие объёмы данных.

**4. Автоматизированная адресная передача аналитических данных:**

Для отправки аналитических данных напрямую руководителю выбран Telegram-бот.

Apache AirFlow позволяет настроить передачу данных в Telegram с помощью модуля TelegramBotOperator.

Преимущества передачи данных из Apache AirFlow в Telegram:

**Моментальное уведомление:** передача данных из Apache Airflow в Telegram позволяет получать моментальные уведомления о статусе выполнения задач и работе планировщика, что помогает оперативно реагировать на любые проблемы или задержки.

**Удобство и доступность:** благодаря использованию мессенджера Telegram для передачи данных, информация всегда доступна и удобна в использовании. Пользователи могут получать уведомления прямо на своих устройствах и в любое время.

**Повышение эффективности и контроля:** с помощью передачи данных из Apache Airflow в Telegram можно легко контролировать выполнение задач и планировать следующие шаги, что помогает повысить эффективность работы и повысить качество результатов.

**Автоматизация процессов:** использование Telegram для передачи данных из Apache Airflow позволяет автоматизировать уведомления и контроль за работой, что снижает необходимость вручную отслеживать выполнение задач.

**Безопасность и конфиденциальность:** Telegram обеспечивает высокий уровень защиты данных, что позволяет передавать конфиденциальную информацию без риска её утечки или доступа третьих лиц. Настройка отправки сообщений по конкретному ID пользователя исключает получение информации третьими лицами, даже если будет перехвачена ссылка на сам Telegram-бот.

**2.4 Выводы по разделу**

В процессе работы была составлена новая структура бизнес-процесса, которая позволяет существенно сэкономить ценное рабочее время ключевых сотрудников, а также избежать ошибок и потери данных.

Также был проведен анализ и отбор инструментов, позволяющих решить поставленные задачи.

Для проведения всех необходимых этапов настройки автоматизированной обработки и анализа данных в компании необходимы следующие инструменты:

* Язык программирования Python
* Язык запросов SQL
* Среда Jupiter Notebook
* Среда IDLE
* База данных HeidiSQL
* Apache AirFlow
* Среда WSL Ubuntu
* Терминал MobaXterm
* Мессенджер Telegram

Применение этих инструментов позволит настроить необходимый функционал и решить поставленную задачу.

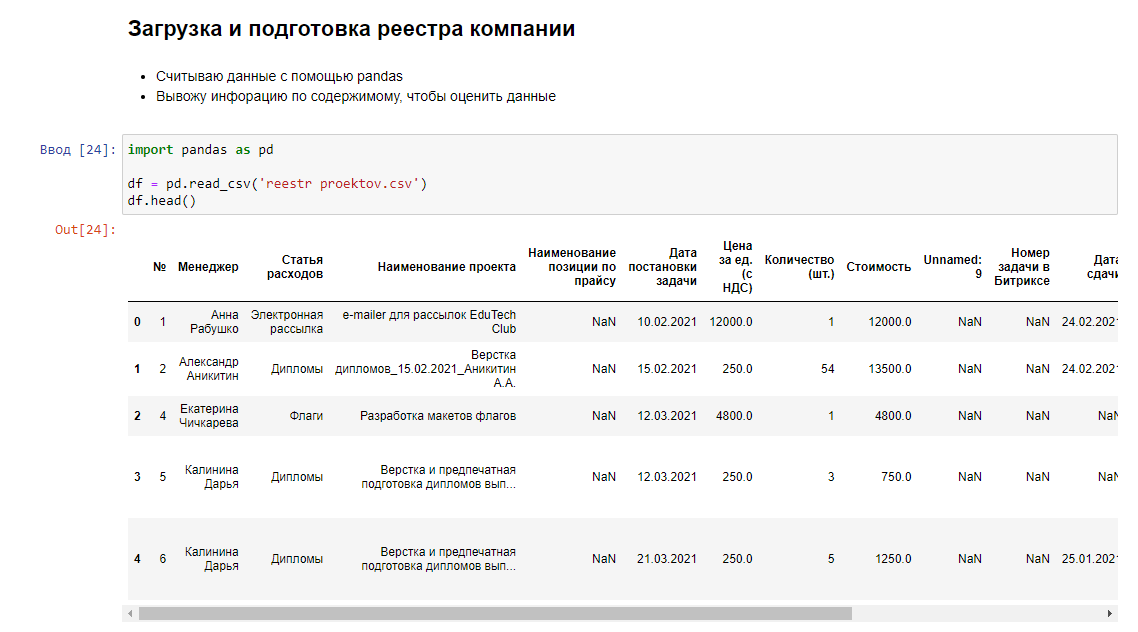
**ГЛАВА 3. процесс настройки АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ И АНАЛИЗА ДАННЫХ КОМПАНИИ**

**3.1** **Подготовка исходного файла реестра к дальнейшей работе**

Выполнение данного этапа проводится в среде Jupiter Notebook на языке программирования Python

**Шаг 1. Загрузка и первичный анализа датасета**

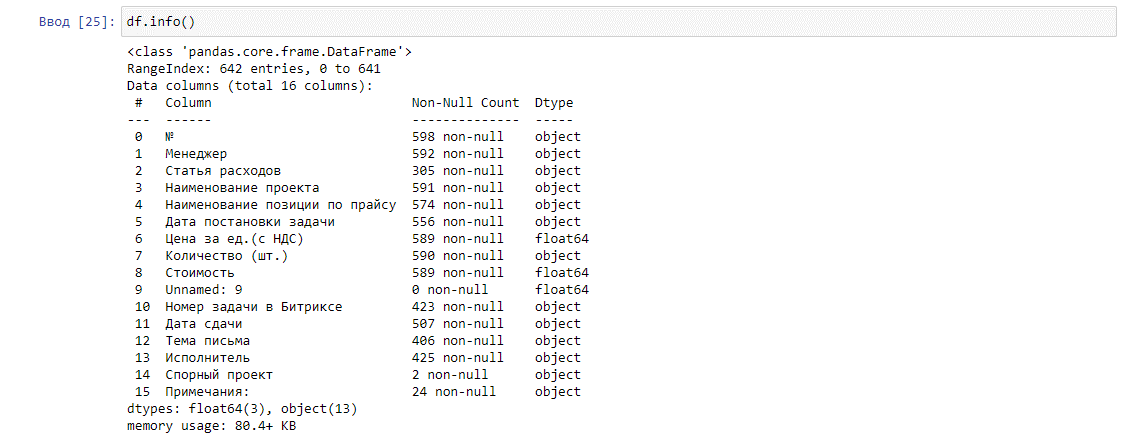
* Для работы с данными необходимо установить библиотеку pandas. Далее происходит считывание файла с данными по реестру задач компании, который был заранее выгружен из Google Sheets в формате csv.



*Рисунок 6. Загрузка датасета из файла*

Ссылка на исходный файл с данными «reestr proektov.csv» находится в Приложении 3.

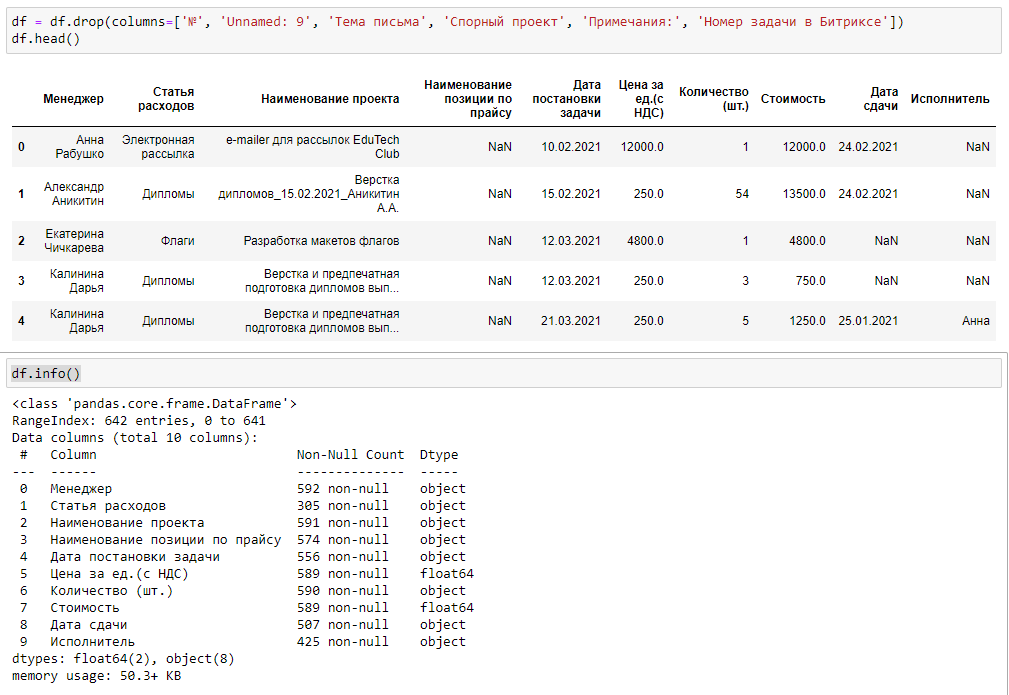
* Для первичного анализа содержимого загруженного датасета необходимо вывести информацию по содержимому, чтобы оценить данные.   
  Эта информация позволит нам оценить следующие параметры:
* размер датасета
* количество и наименование столбцов
* тип данных по столбцам
* объем заполненности данных



*Рисунок 7. Информация по загруженному датасету*

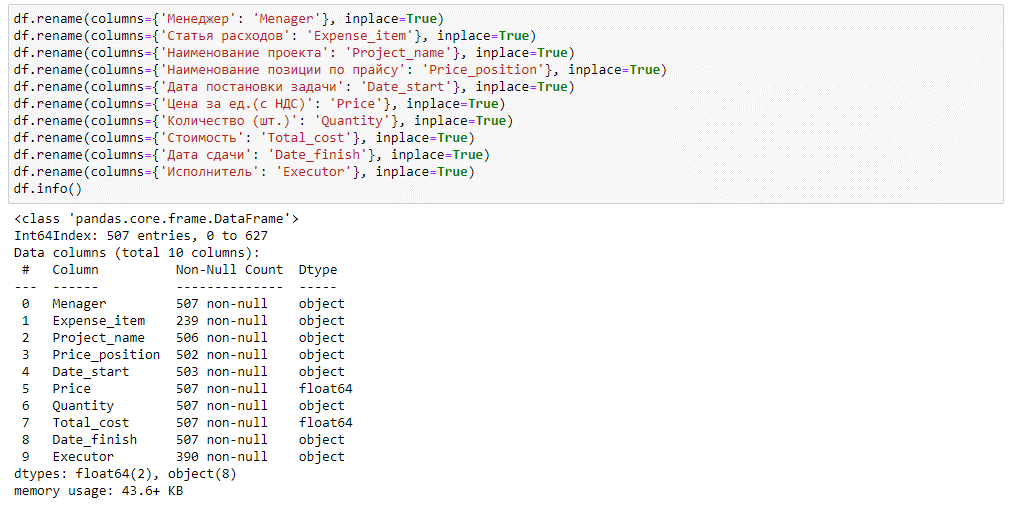
**Шаг 2. Предварительная обработка данных и очистка.**

* Первичных анализ структуры датасета показал, что присутствуют столбцы, которые не содержат ключевой информации для анализа, а также имеют большое количество пропусков данных. Это столбцы: '№', 'Unnamed: 9', 'Тема письма', 'Спорный проект', 'Примечания:', 'Номер задачи в Битриксе'. Эти столбцы необходимо удалить.   
  После удаления столбцов выводим обновленную информацию по датасету.



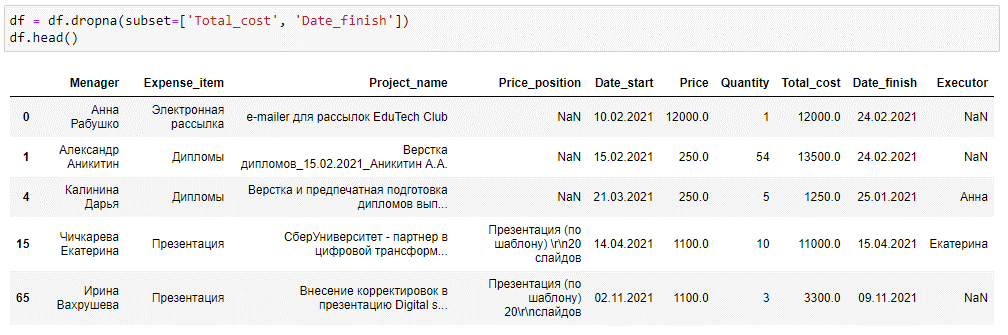
*Рисунок 8. Удаление столбцов в датасете*

* Для дальнейшего удобства обработки и считывании данных изменяю название всех столбцов на написание на английском языке.  
  После переименования столбцов вывожу обновленную информацию по датасету для проверки результата проводимых изменений.



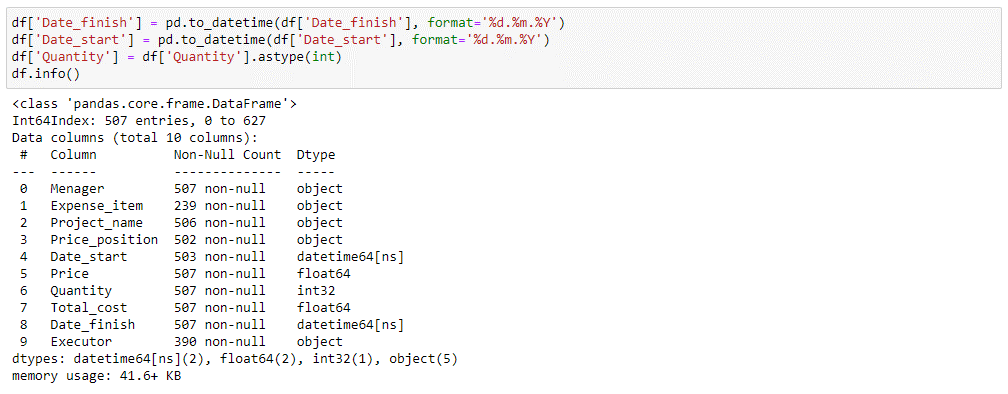
*Рисунок 9. Переименование столбцов в датасете*

* Удаляю строки, если в столбцах 'Total\_cost' и 'Date\_finish' отсутствуют значение т.к., вероятно, это не завершенные проекты (статистика, по которым нам пока не требуется) или пустые строки, не нужные для аналитики



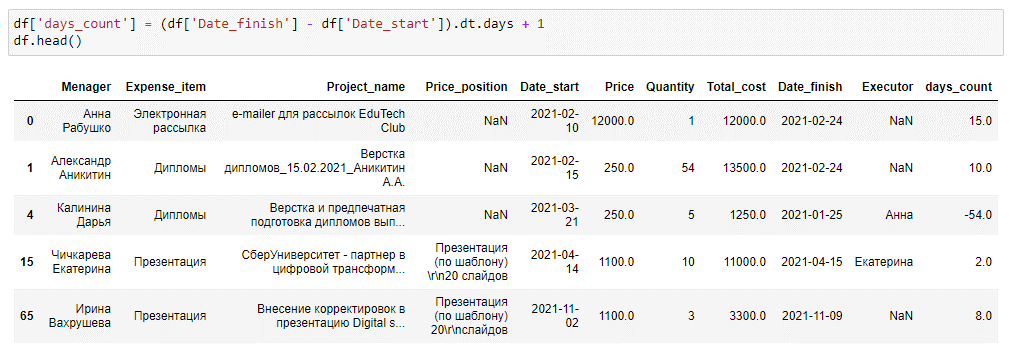
*Рисунок 10. Удаление строк с пропущенными значениями*

* Анализирую типы данных по столбцам и преобразую тип данных столбцов 'Date\_start' и 'Date\_finish' в дату, а 'Quantity' в число



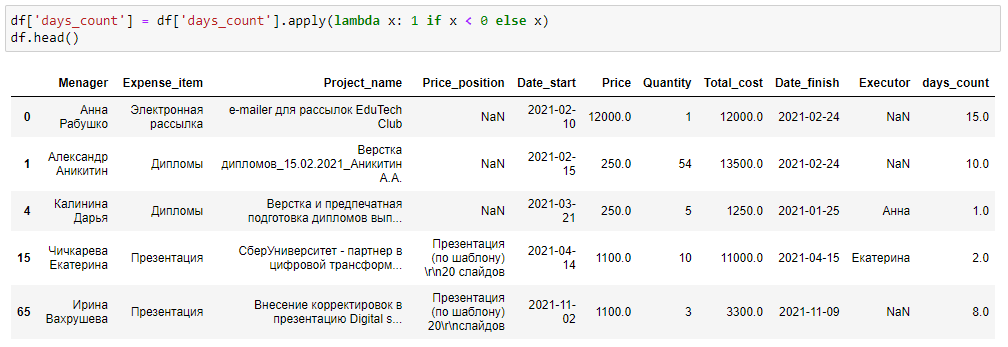
*Рисунок 11. Изменение типа данных столбцов*

* Для расчета длительности выполнения задач, добавляю новый столбец «days\_count», значение которого формируется как разница между датой окончания проекта и датой начала выполнения задачи. Поскольку день постановки задачи также считается днем выполнения: прибавляем +1 день ко всем расчетам



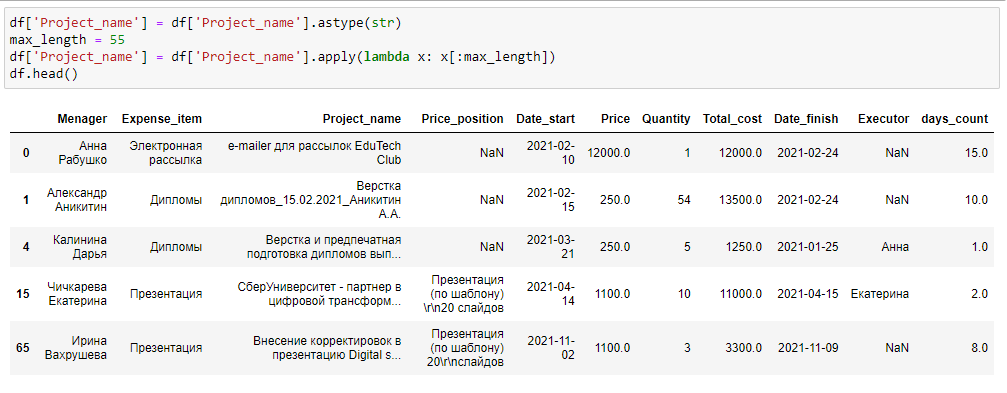
*Рисунок 12. Добавление нового признака в датасет*

* Поскольку не все значения в столбце «days\_count», имеют корректное значение, заменяю все отрицательные значения на 1



*Рисунок 13. Замена некорректных значений в датасете на константные*

* Столбец "Project\_name" имеет слишком длинный текст, это может мешать дальнейшей обработке информации. Преобразую тип данных в столбце в строковый и сокращаю количество текста в столбце до 55 символов.



*Рисунок 14. Сокращение длины текста в столбце*

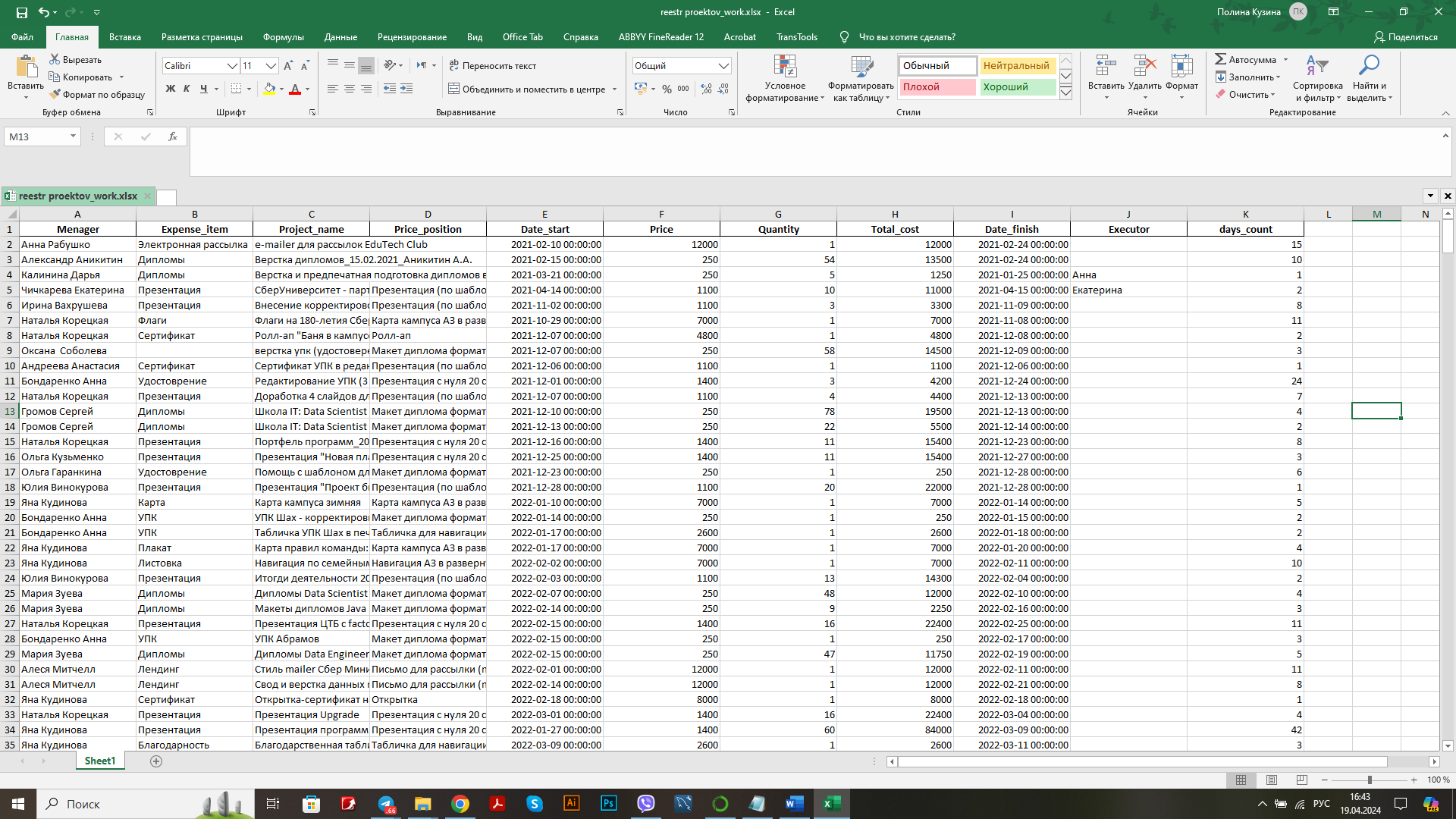
**Шаг 3. Выгрузка подготовленного документа для дальнейшей работы**



*Рисунок 15. Выгрузка подготовленного датасета в файл excel*

Ссылка на файл «Обработка реестра.ipynb» с выполняемым кодом и вычислениями находится в Приложении 3.

**Шаг 4. Файл «reestr proektov\_work.xlsx» успешно создан и готов для дальнейшей работы**



*Рисунок 16. Файл excel с подготовленными данными*

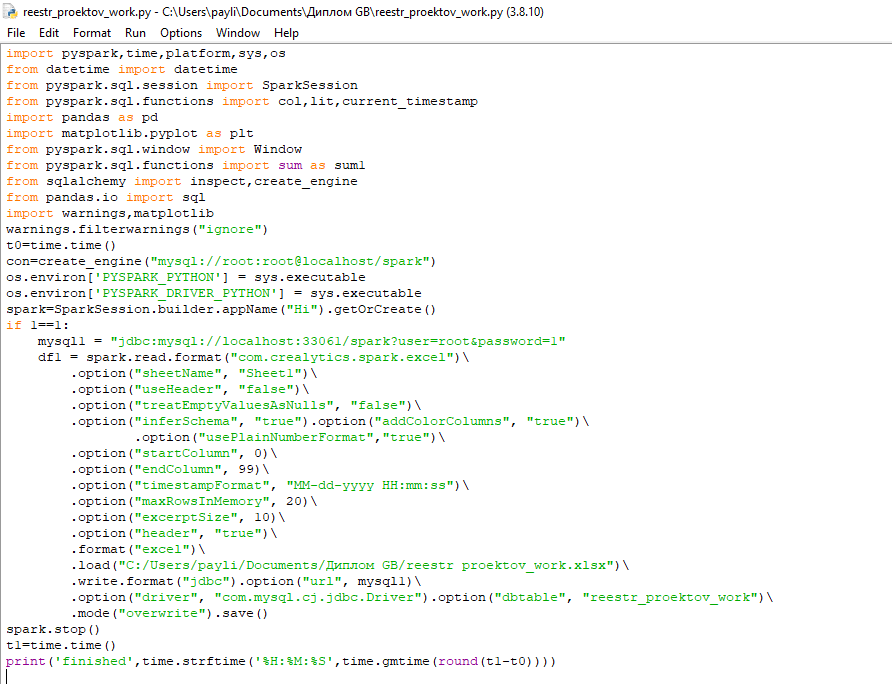
Ссылка получившуюся таблицу «reestr proektov\_work.xlsx» находится в Приложении 3.

**3.2 Загрузка подготовленного датасета в базу данных**

Выполнение данного этапа проводится в среде IDLE на языке программирования Python. Данные загружаются в базу HeidiSQL.

**Шаг 1. Написание кода на Python в IDLE**

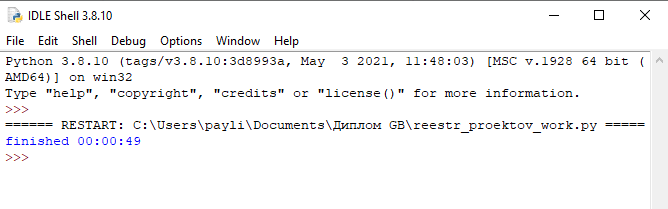
* Создаю файл «reestr\_proektov\_work.py», в котором прописываю код для считывания данных из файл «reestr proektov\_work.xlsx» и загрузки датасета в HeidiSQL. Подробное описание загрузки библиотек и написание кода представлено на Рисунке 17.



*Рисунок 17. Код на Python для считывания и загрузки датасета в базу данных*

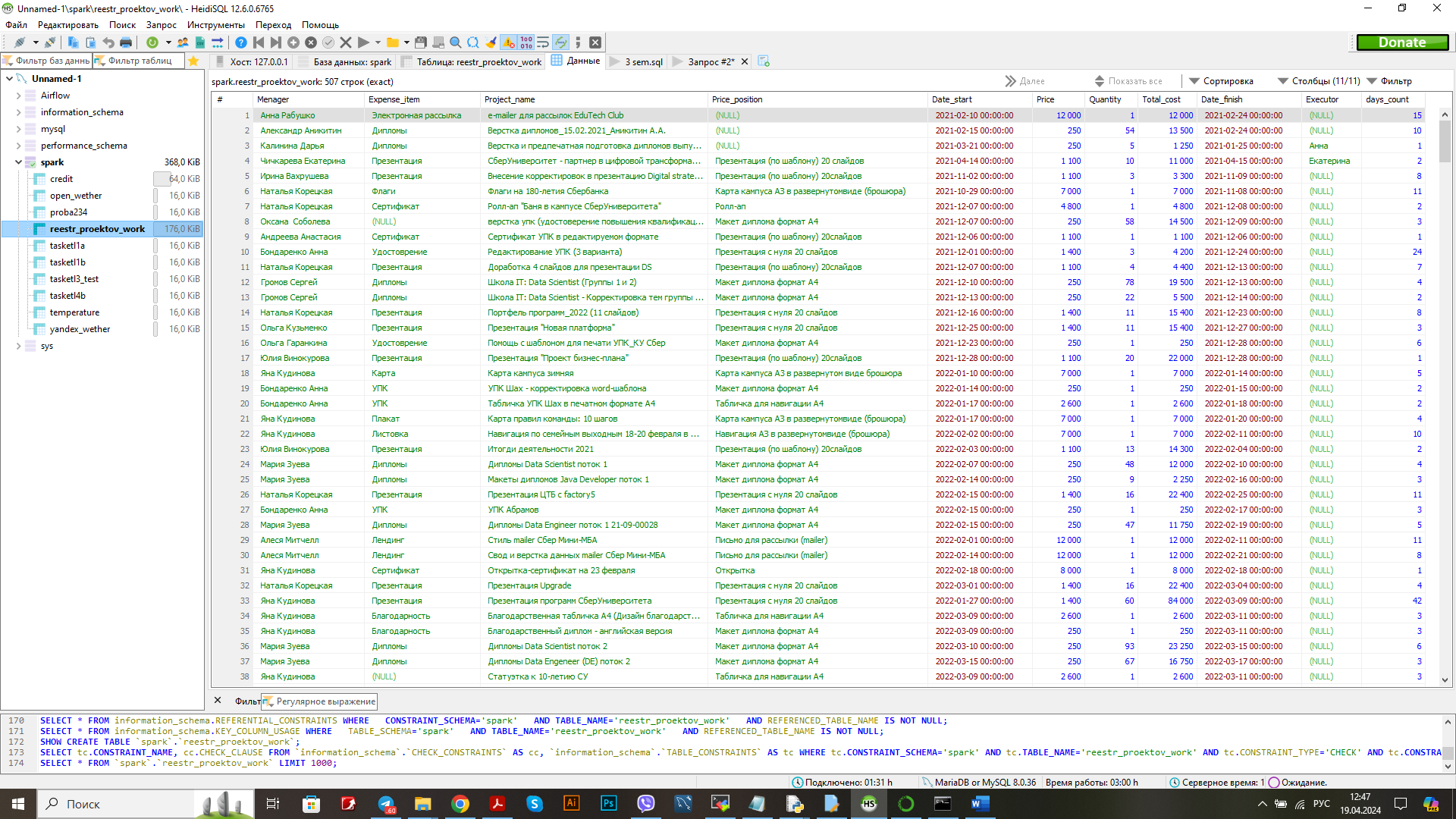
Ссылка на файл с кодом «reestr\_proektov\_work.py» находится в   
Приложении 3.

* Запускаю код, дожидаюсь его выполнения и проверяю наличие созданной таблицы в HeidiSQL



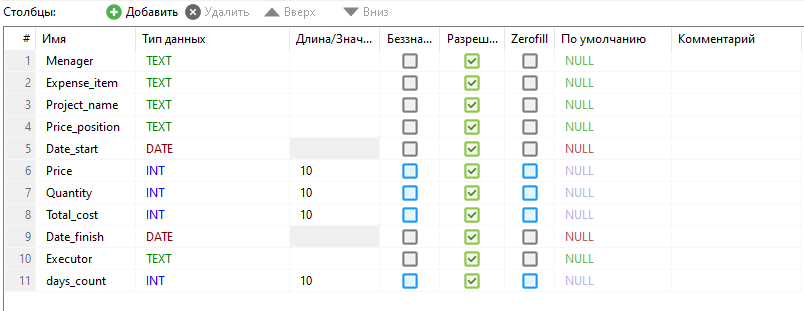
*Рисунок 18. Отслеживание работы кода в IDLE Shell*

* Датасет со всеми данными успешно перенесена в HeidiSQL и готова к дальнейшей работе



*Рисунок 19. База данных HeidiSQL с загруженным датасетом*

* Проверяю корректность типов данных по столбцам. Заменяю тип данных в столбцах «Date\_start» и «Date\_finish» на формат DATA, а в столбце «daya\_count» на числовой INT.



*Рисунок 20. Замена типов данных в столбцах*

* Таблица готова к дальнейшей работе

**3.3 Создание чат-бота в Telegram для удобного получения аналитических данных**

Выполнение данного этапа проводится в мессенджере Telegram.

**Шаг 1. Создание чат-бота**

* В телеграмм-канале BotFather <https://t.me/BotFather> создаю нового бота «reestr\_proektov\_bot» с помощью команд:

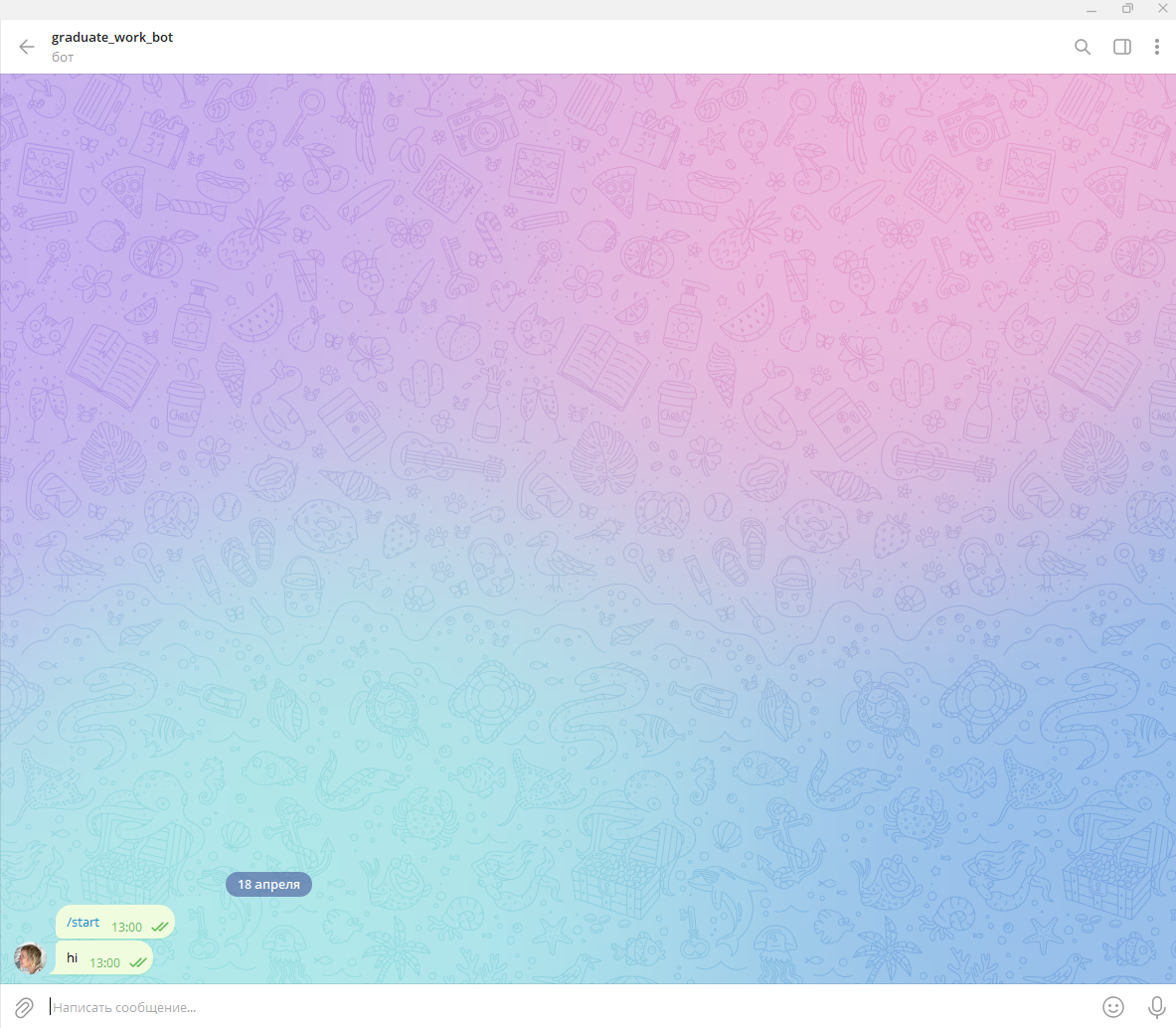
/start

/newbot

/tocken

/mybots

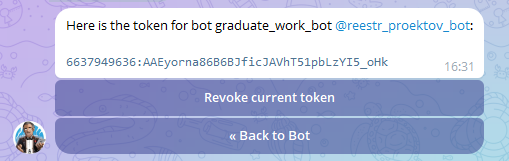
Ссылка на бот «reestr\_proektov\_bot» находится в Приложении 3.



*Рисунок 21. Созданный бот «reestr\_proektov\_bot»*

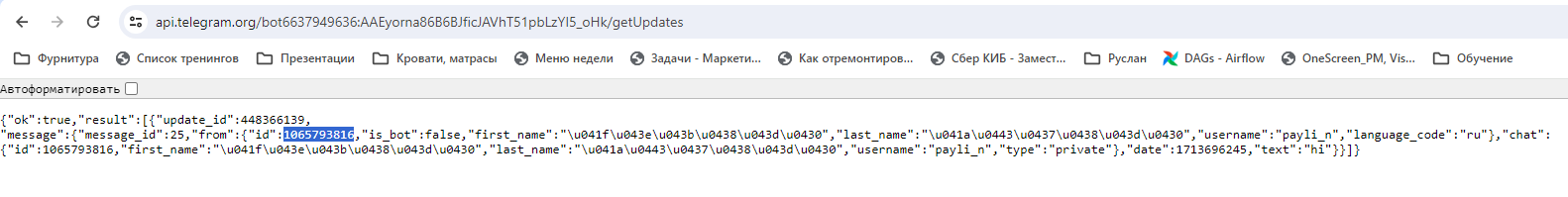
**Шаг 2. Получение персональных идентификационных данных бота**

* После того как бот создан, получаю и сохраняю персональный токен бота для дальнейшего подключения и передачи информации в бот.



*Рисунок 22. Токен для бота «reestr\_proektov\_bot»*

* Для получения ID пользователя, которому будут передаваться сообщения, выполняю следующие действия:
* Активирую бота, отправляя несколько сообщений с того аккаунта, на который в дальнейшем будут передаваться сообщения
* По ссылке https://api.telegram.org/bot6637949636:AAEyorna86B6BJficJAVhT51pbLzYI5\_oHk/getUpdates в браузере открываю информацию по отправленным сообщениям и копирую данные ID



*Рисунок 23. Получение ID пользователя Telegram*

**3.4 Активация обработки и анализа данных через Apache AirFlow и интеграция отправки сообщений в Телеграмм**

Выполнение данного этапа проводится с помощью Apache AirFlow в среде WSL Ubuntu в терминале MobaXterm.

Отправка сообщений осуществляется в мессенджере Telegram.

Для написания DAG файла используется язык программирования Python. Для написания запросов к базе данных используется язык SQL.

**Шаг 1. Установка рабочей среды, настройка запуска Apache AirFlow**

* Развертываю рабочую среду WSL Ubuntu (все программное обеспечение было установлено в рамках прохождения курса по ETL)
* Загружаю терминал MobaXterm и перехожу во вкладку WSL Ubuntu
* Последовательно ввожу команды в терминале, чтобы активировать виртуальную среду и запустить Airflow

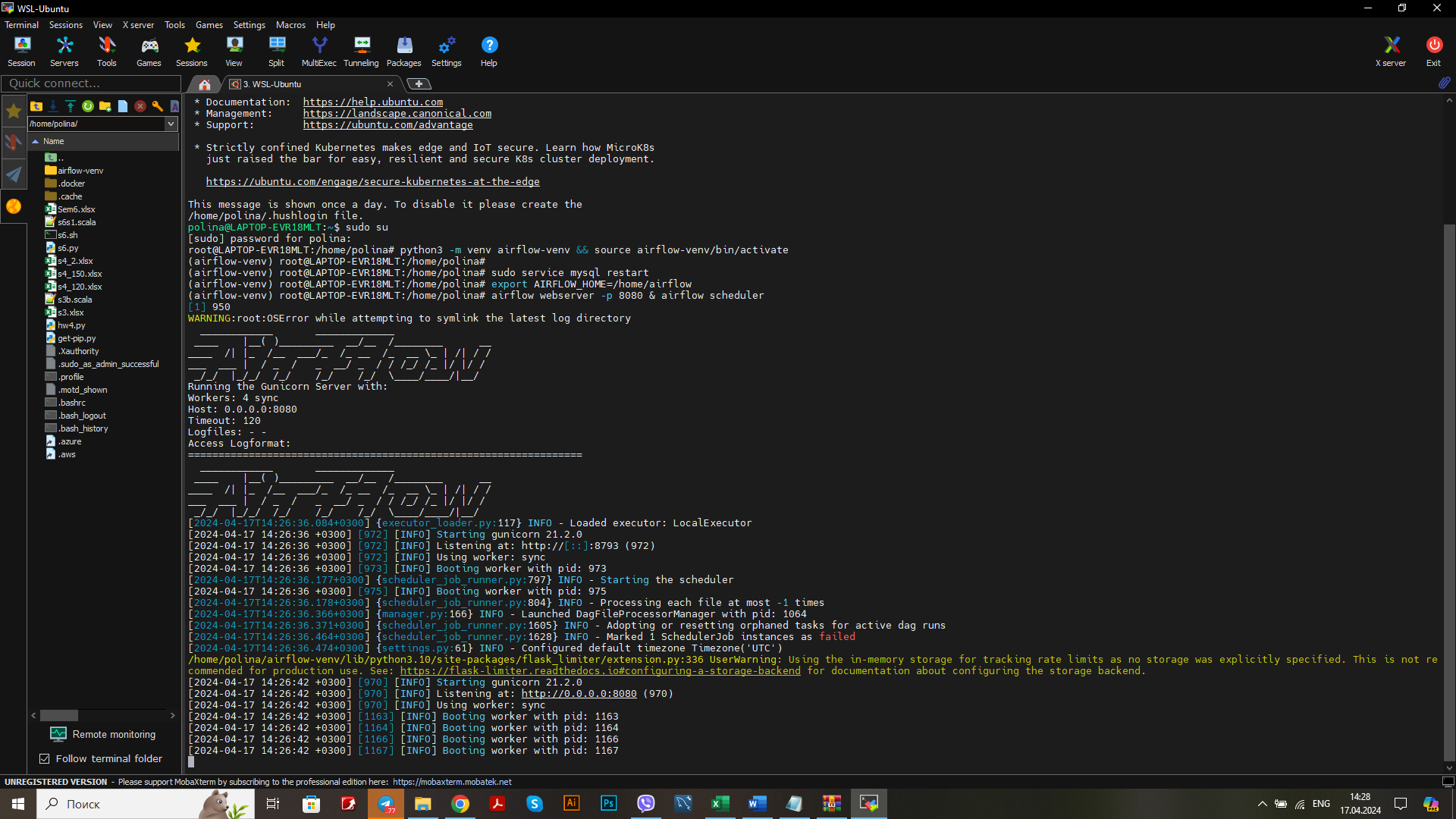
sudo su

python3 -m venv airflow-venv && source airflow-venv/bin/activate

sudo service mysql restart

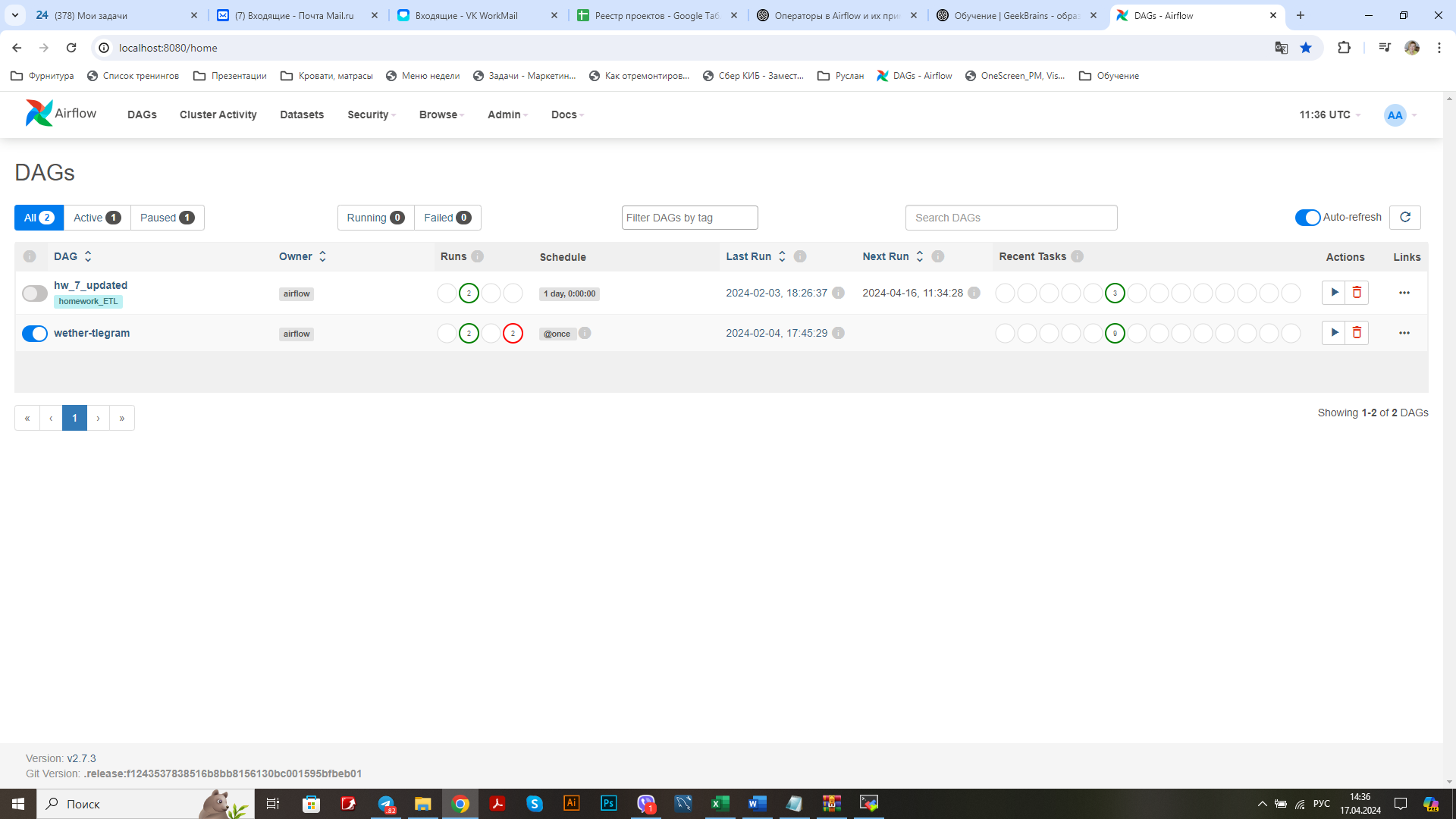
export AIRFLOW\_HOME=/home/airflow

airflow webserver -p 8080 & airflow scheduler



*Рисунок 24. Запуск Apache AirFlow через терминал MobaXterm*

* В браузере перехожу на порт http://localhost:8080/home и проверяю, запущен ли Apache AifFlow.



*Рисунок 25. Apache AirFlow готов к работе*

**Шаг 2. Установка необходимых библиотек**

* Для отправки сообщений в Telegram-бота в виртуальную среду устанавливаю библиотеку «apache-airflow-providers-telegram»

pip install apache-airflow-providers-telegram

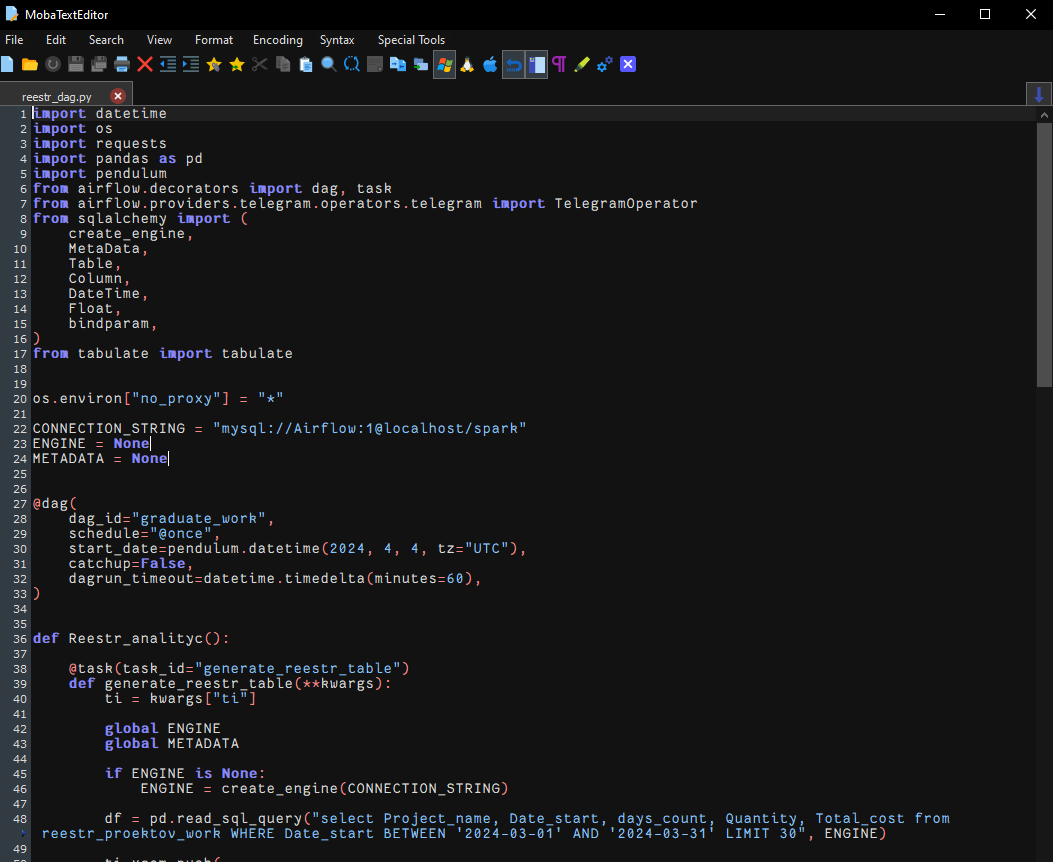
**Шаг 3. Создание DAG файла для отправки запросов и передачи данных в телеграмм-бота**

* Создаю dag файл «reestr\_dag.py» и отправляю его в папку dags в wsl ubunta

В файле прописываю:

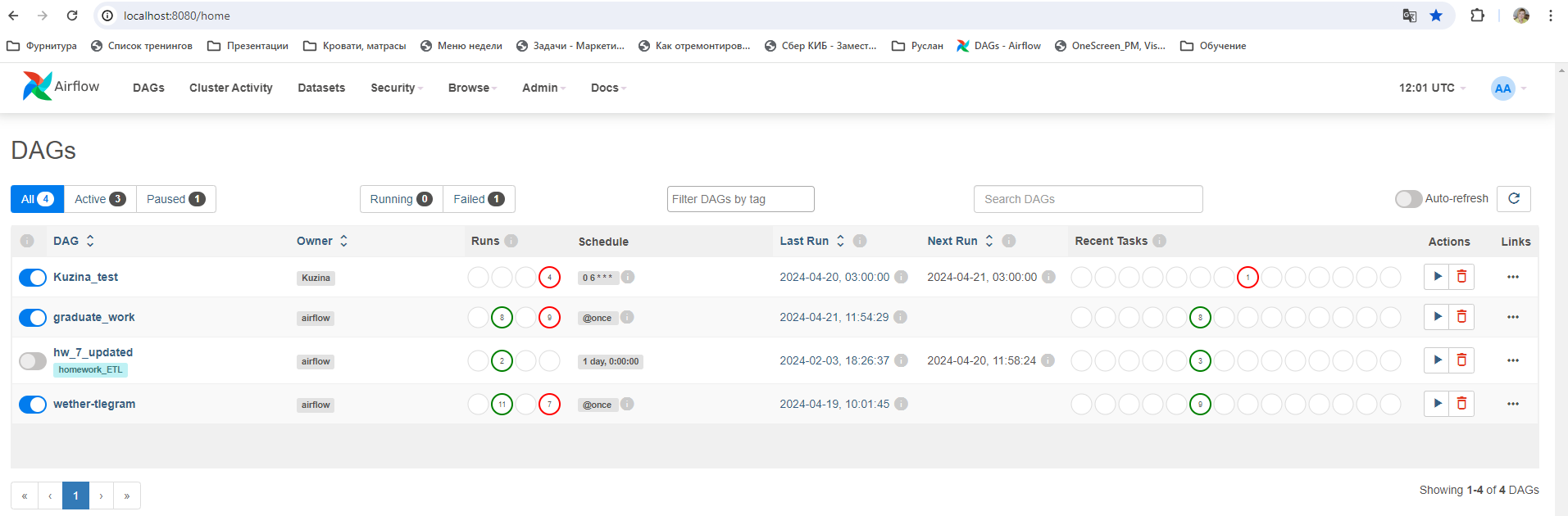
* Загрузку необходимых библиотек
* Подключение к базе данных HeidiSQL
* Название DAG, которое будет отображаться в AirFlow
* Необходимые функции и таски для выполнения поставленных задач по получению и обработке данных и передачи их в Telegram-бота. В функциях идет вычисление необходимых метрик.
* Последовательность выполнения тасков

Ссылка на dag файл «reestr\_dag.py» находится в Приложении 3.



*Рисунок 26. Фрагмент DAG файла*

* Проверяю, что в AirFlow появился созданный DAG

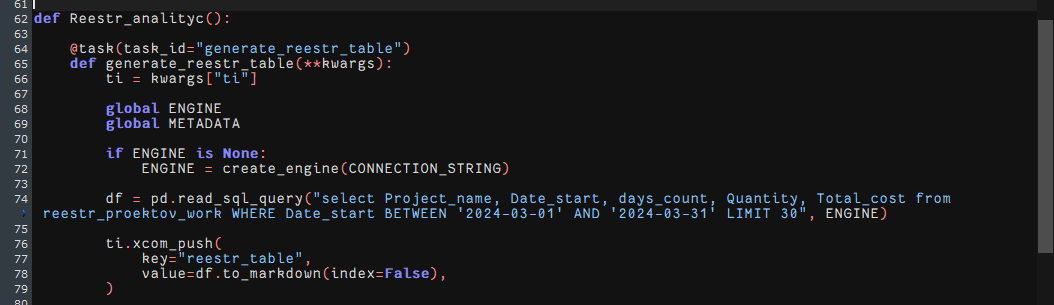


*Рисунок 27. Отображение DAG файла в AirFlow*

**Описание тасков в DAG файле:**

1. Task generate\_reestr\_table

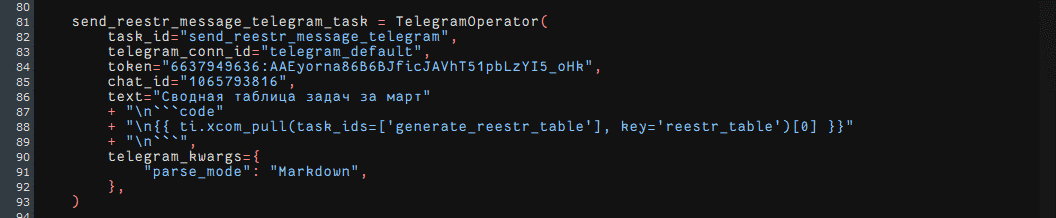
Создание таблицы из столбцов «Project\_name», «Date\_start», «days\_count», «Quantity», «Total\_cost» за указанный месяц.



*Рисунок 28. Фрагмент DAG файла - Task generate\_reestr\_table*

1. Task send\_reestr\_message\_telegram

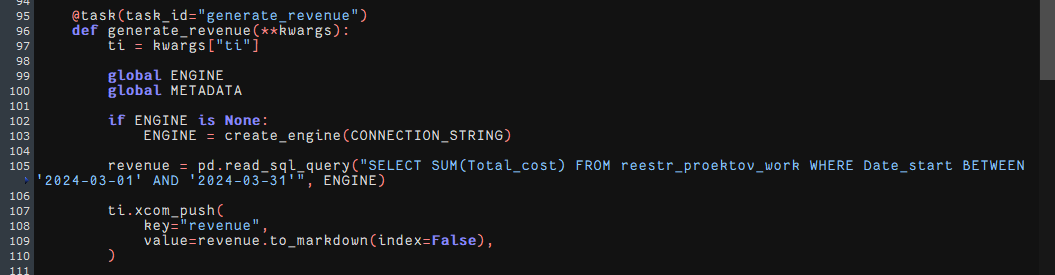
Отправка созданной в таске «generate\_reestr\_table» таблицы в Telegram-бота.



*Рисунок 29. Фрагмент DAG файла - Task send\_reestr\_message\_telegram*

1. Task generate\_revenue

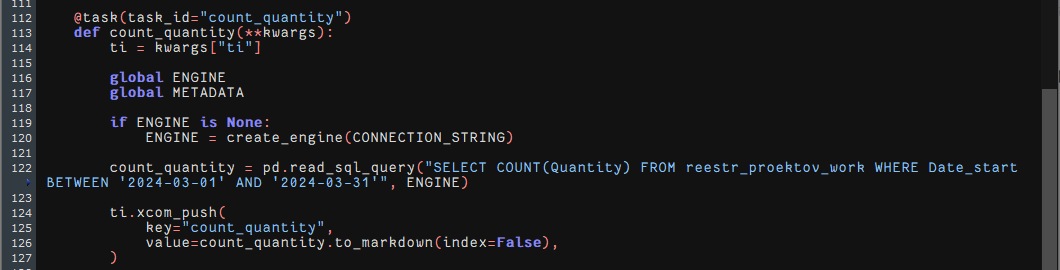
Расчет выручки за указанный период



*Рисунок 30. Фрагмент DAG файла - Task generate\_revenue*

1. Task count\_quantity

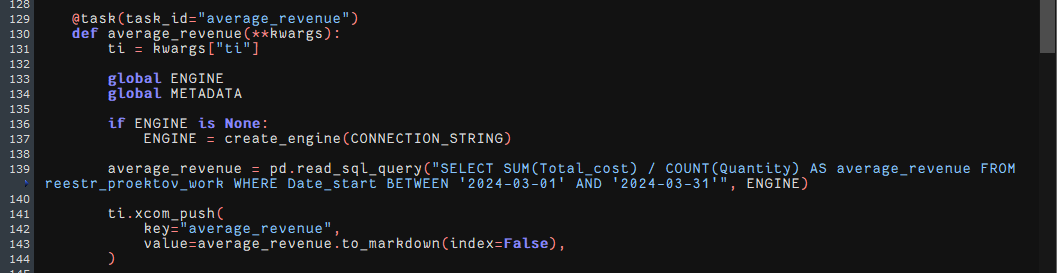
Расчет количества задач за указанный период



*Рисунок 31. Фрагмент DAG файла - Task count\_quantity*

1. Task average\_revenue

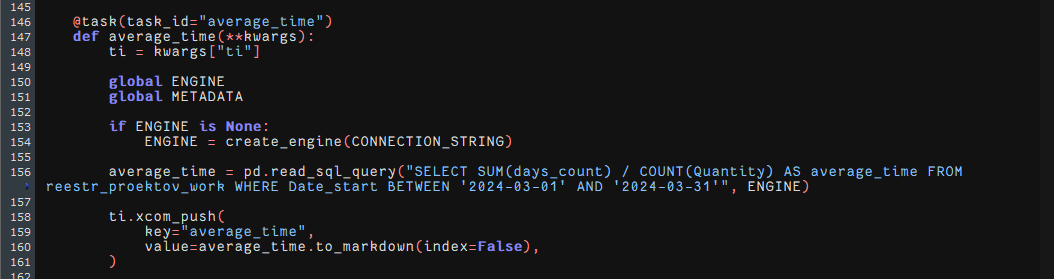
Расчет среднего чека за указанный период



*Рисунок 32. Фрагмент DAG файла - Task average\_revenue*

1. Task average\_time

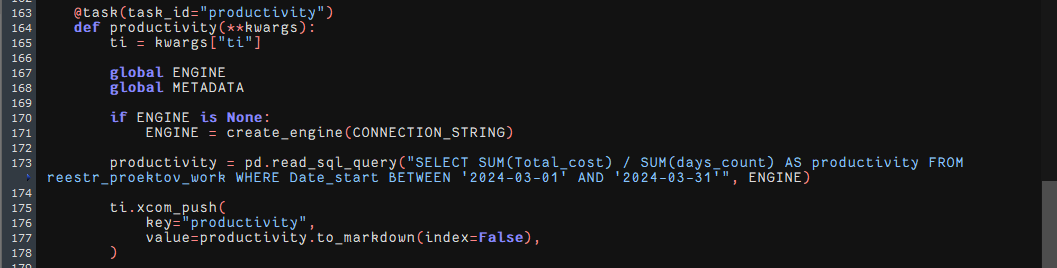
Расчет среднего времени выполнения одной задачи за указанный период



*Рисунок 33. Фрагмент DAG файла - Task average\_time*

1. Task productivity

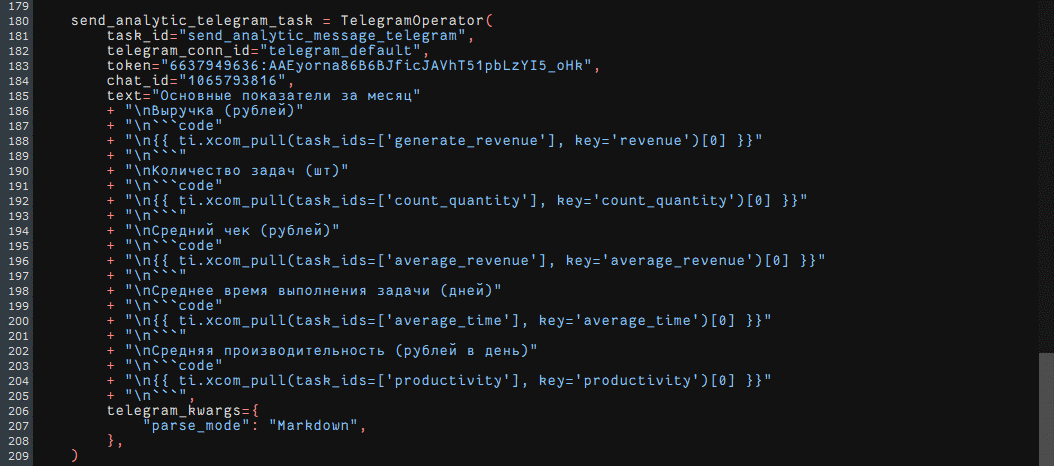
Расчет производительности труда за указанный период



*Рисунок 34. Фрагмент DAG файла - Task productivity*

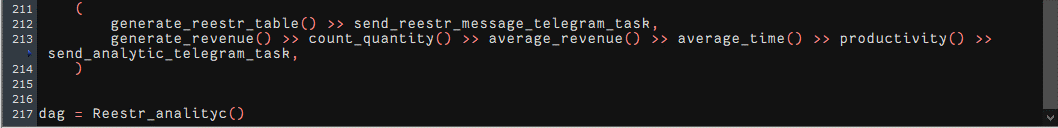
1. Task send\_analytic\_message\_telegram

Отправка расчетов по всем необходимым метрикам в Telegram-бота.



*Рисунок 35. Фрагмент DAG файла - Task send\_analytic\_message\_telegram*

1. Задаем последовательность вызова и исполнения функций в AirFlow

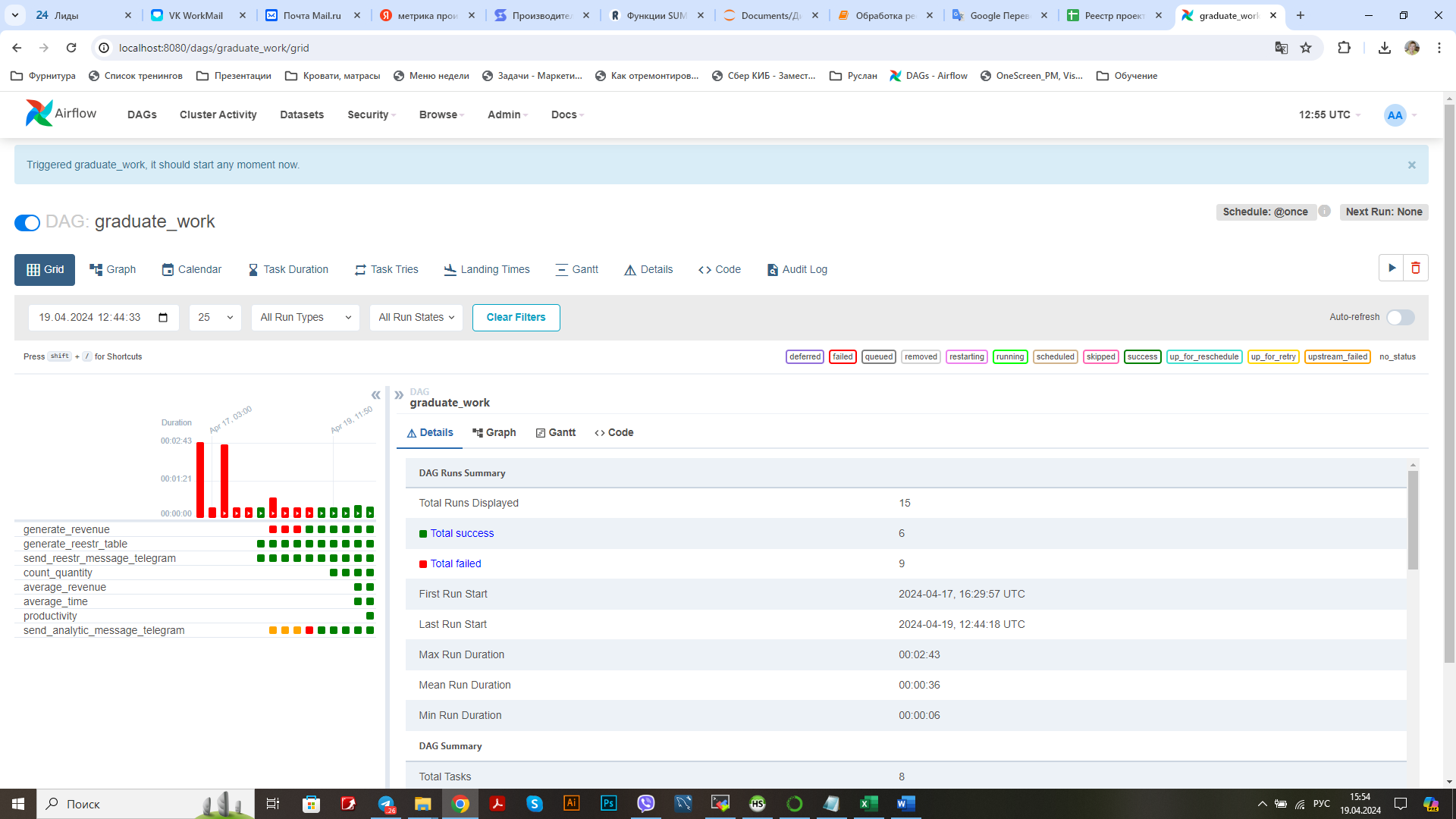


*Рисунок 36. Фрагмент DAG файла - последовательность вызова и исполнения функций в AirFlow*

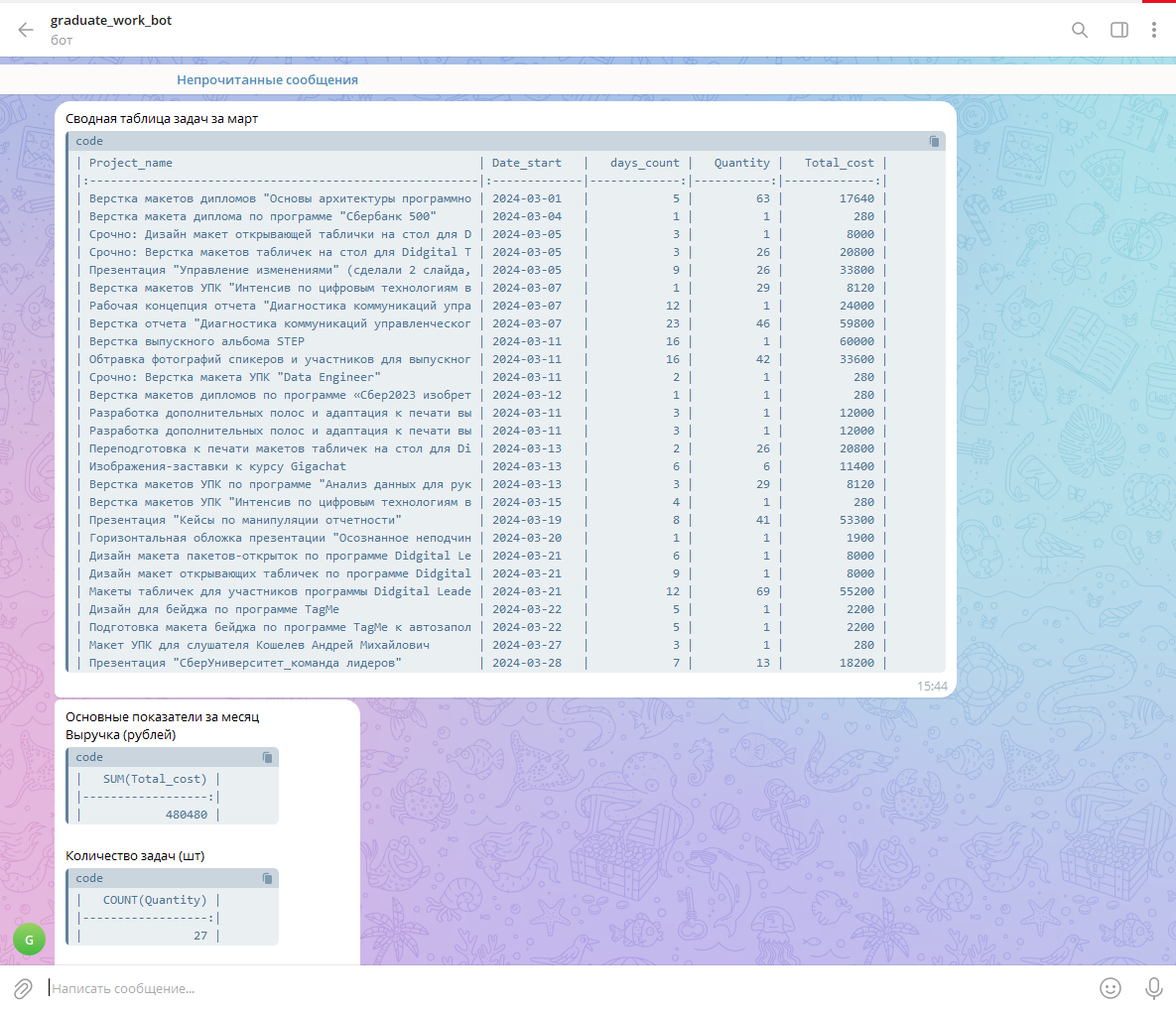
**Шаг 4. Запуск DAGа в AirFlow**

* Перехожу в Apache AirFlow запускаю работу созданного DAGа.  
  Слежу, чтобы все таски отработали без ошибок и в заданной последовательности.

Результатом корректной работы DAG можно считать отображение зеленых индикаторов напротив каждого таска в Apache AirFlow, а также выполненная отправка сообщений в Telegram-боте.



*Рисунок 37. Результат корректной работы DAG в Apache AirFlow*



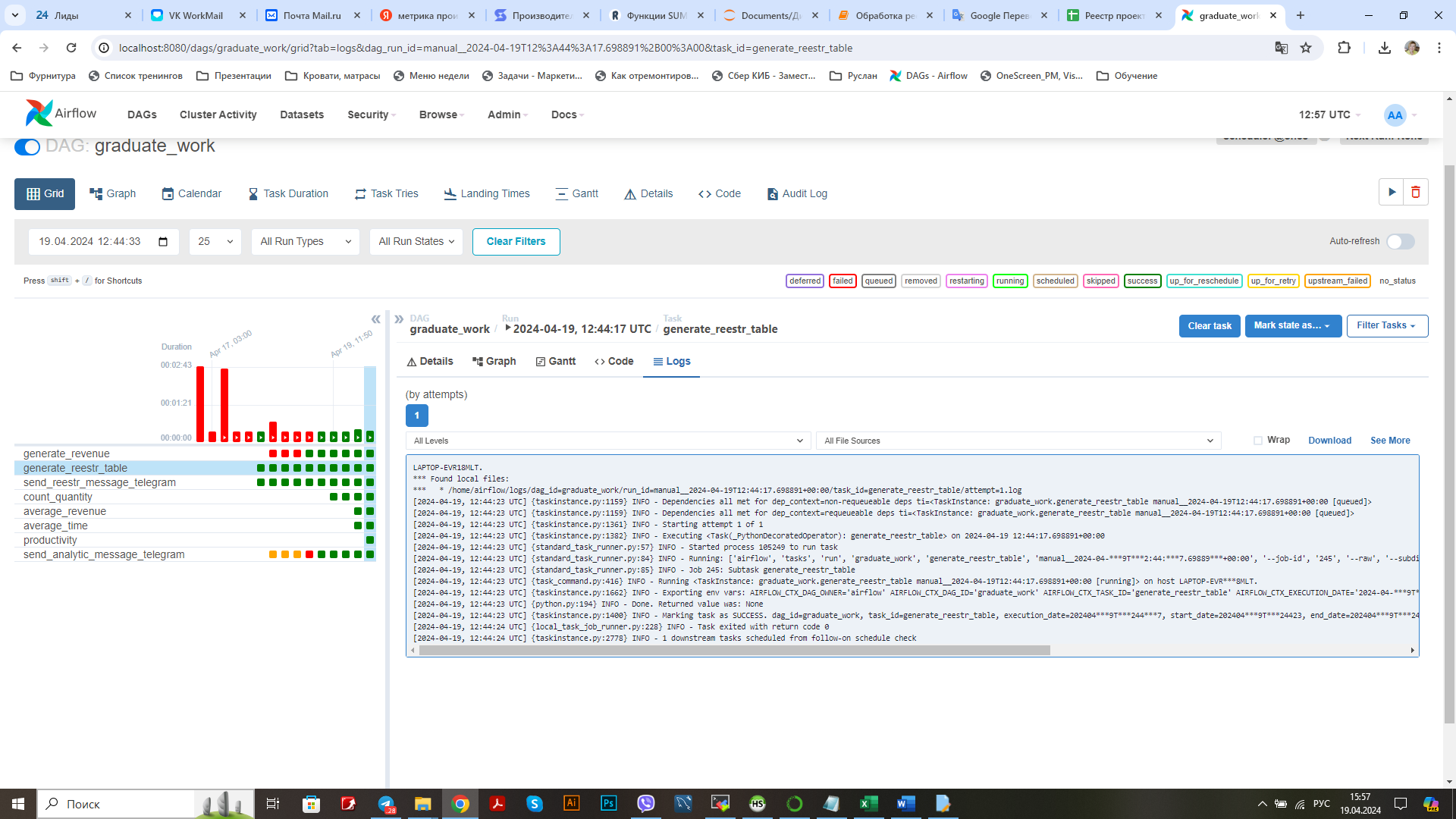
*Рисунок 38. Сообщение с таблицей из реестра в Telegram-боте*



*Рисунок 39. Сообщение с метриками из реестра в Telegram-боте*

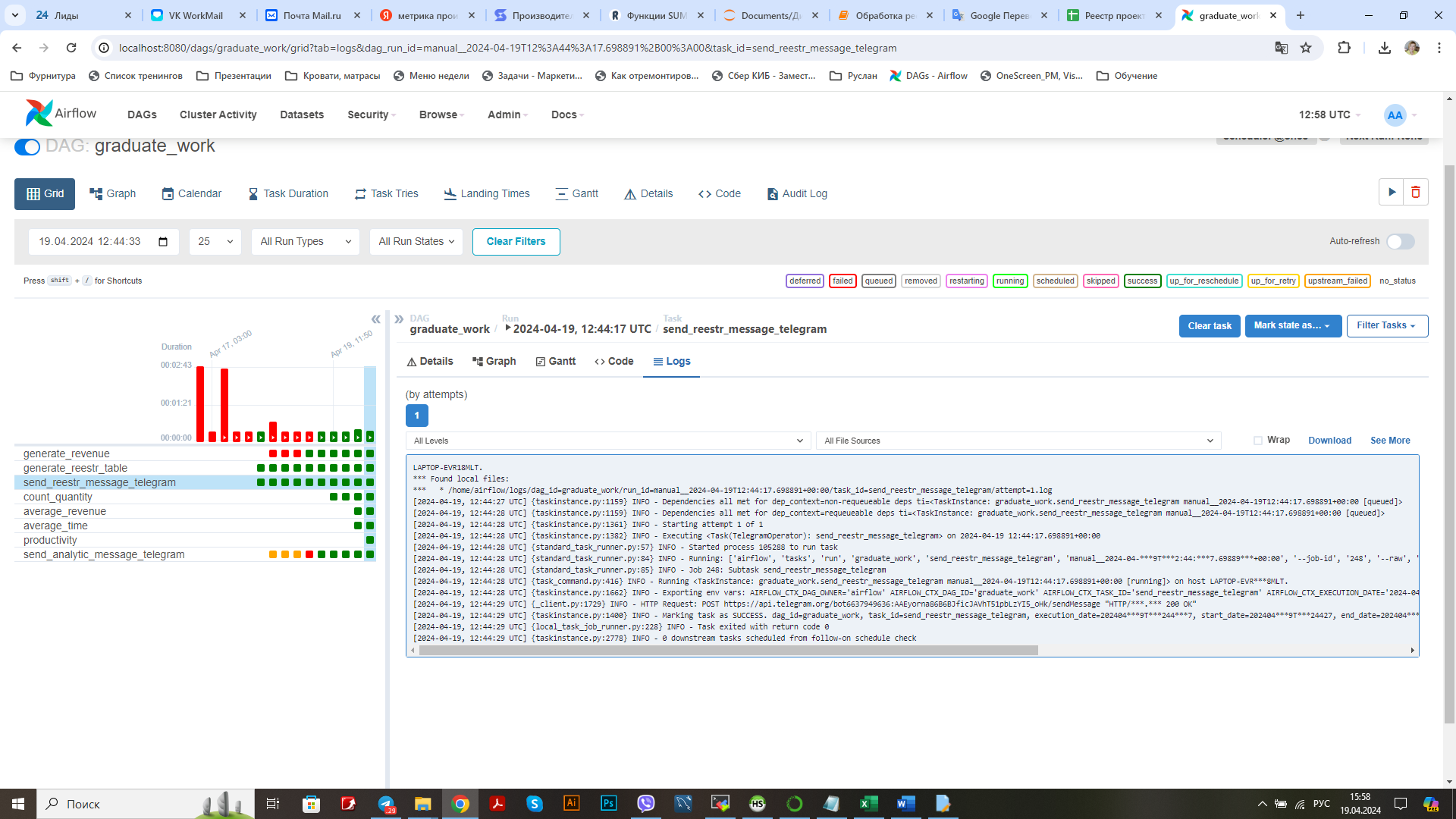
**Логи по каждому таску в Apache AirFlow:**

1. Task generate\_reestr\_table



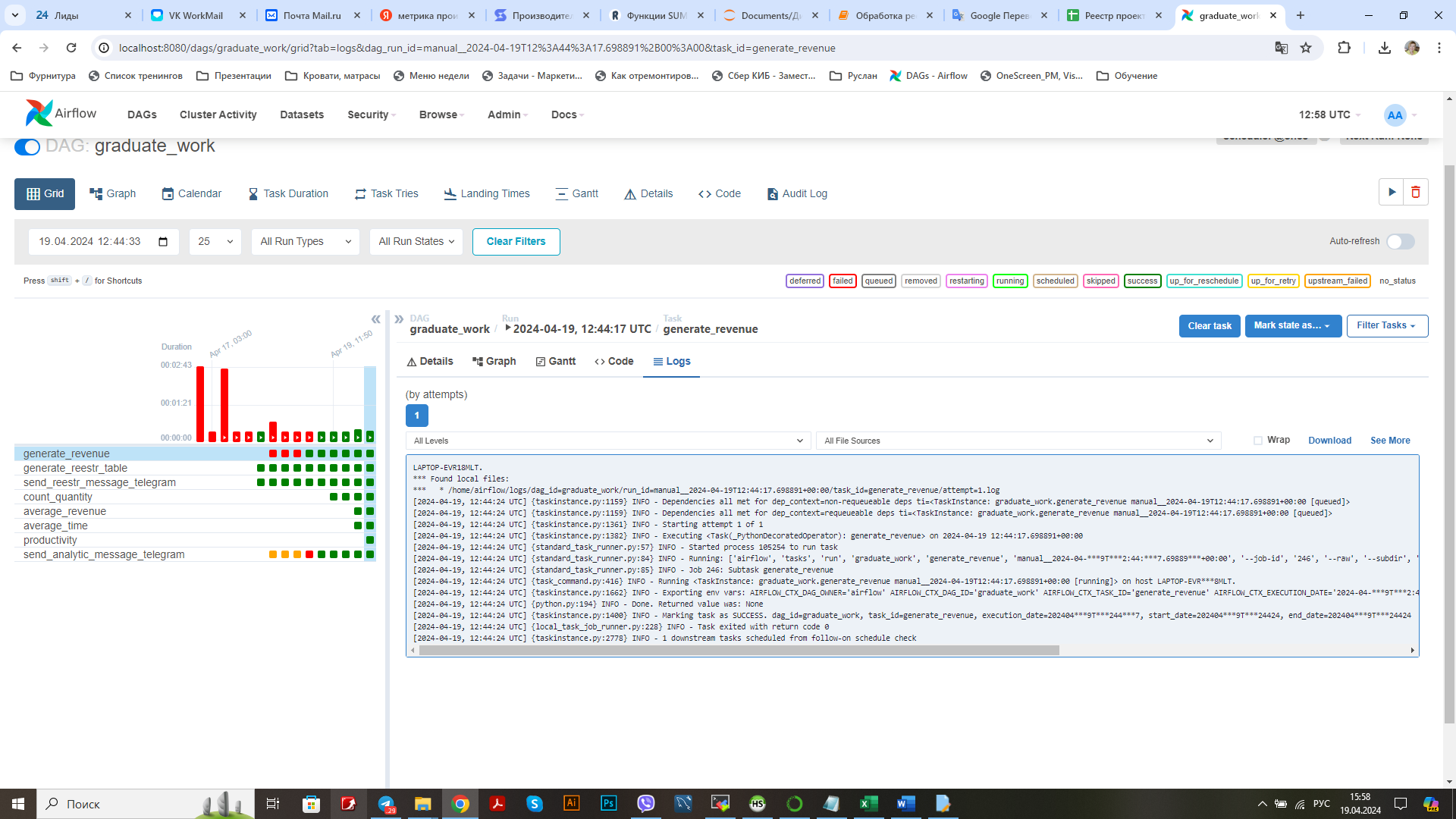
*Рисунок 40. Логи по Task generate\_reestr\_table*

1. Task send\_reestr\_message\_telegram



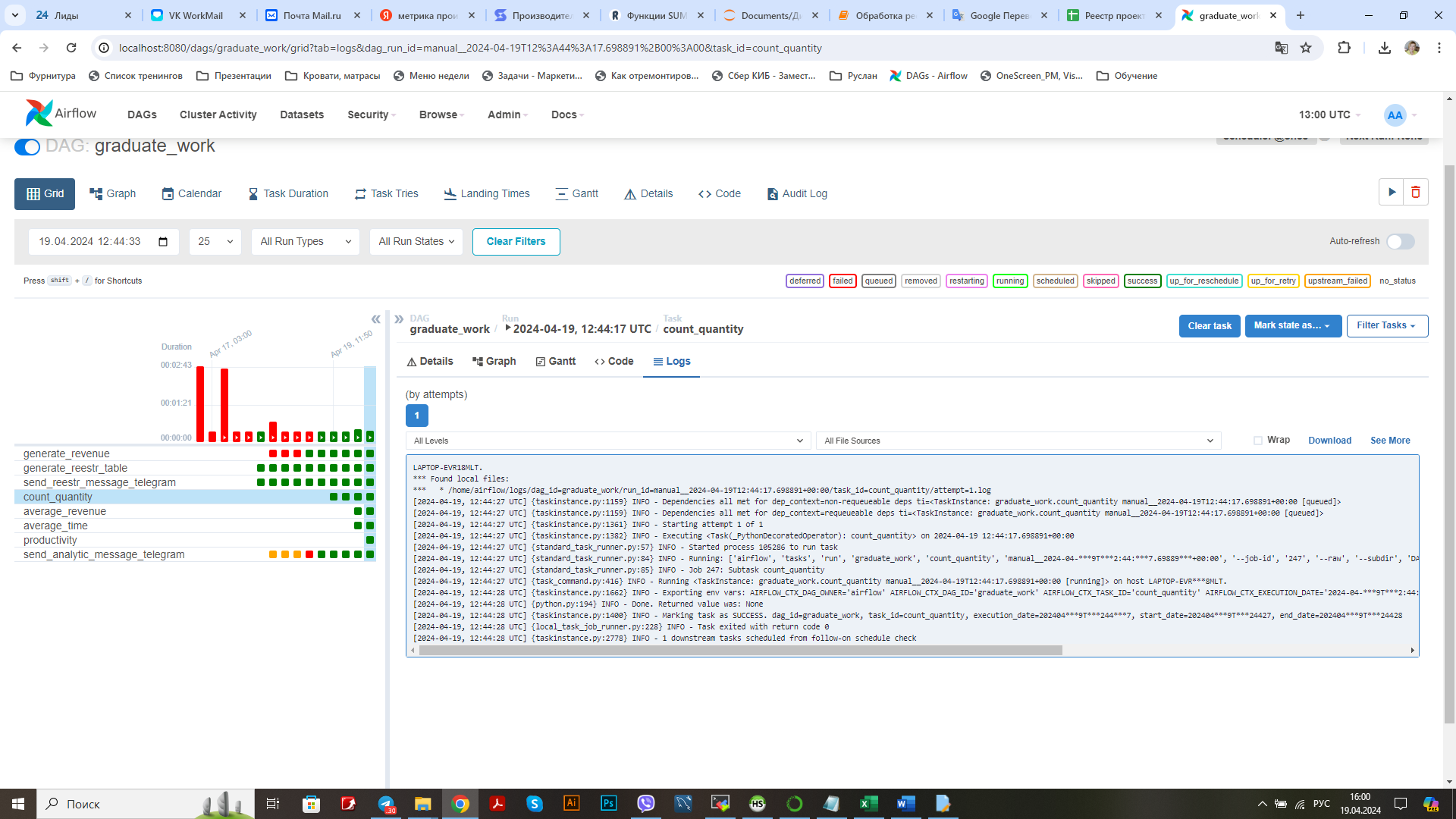
*Рисунок 41. Логи по Task send\_reestr\_message\_telegram*

1. Task generate\_revenue



*Рисунок 42. Логи по Task generate\_revenue*

1. Task count\_quantity



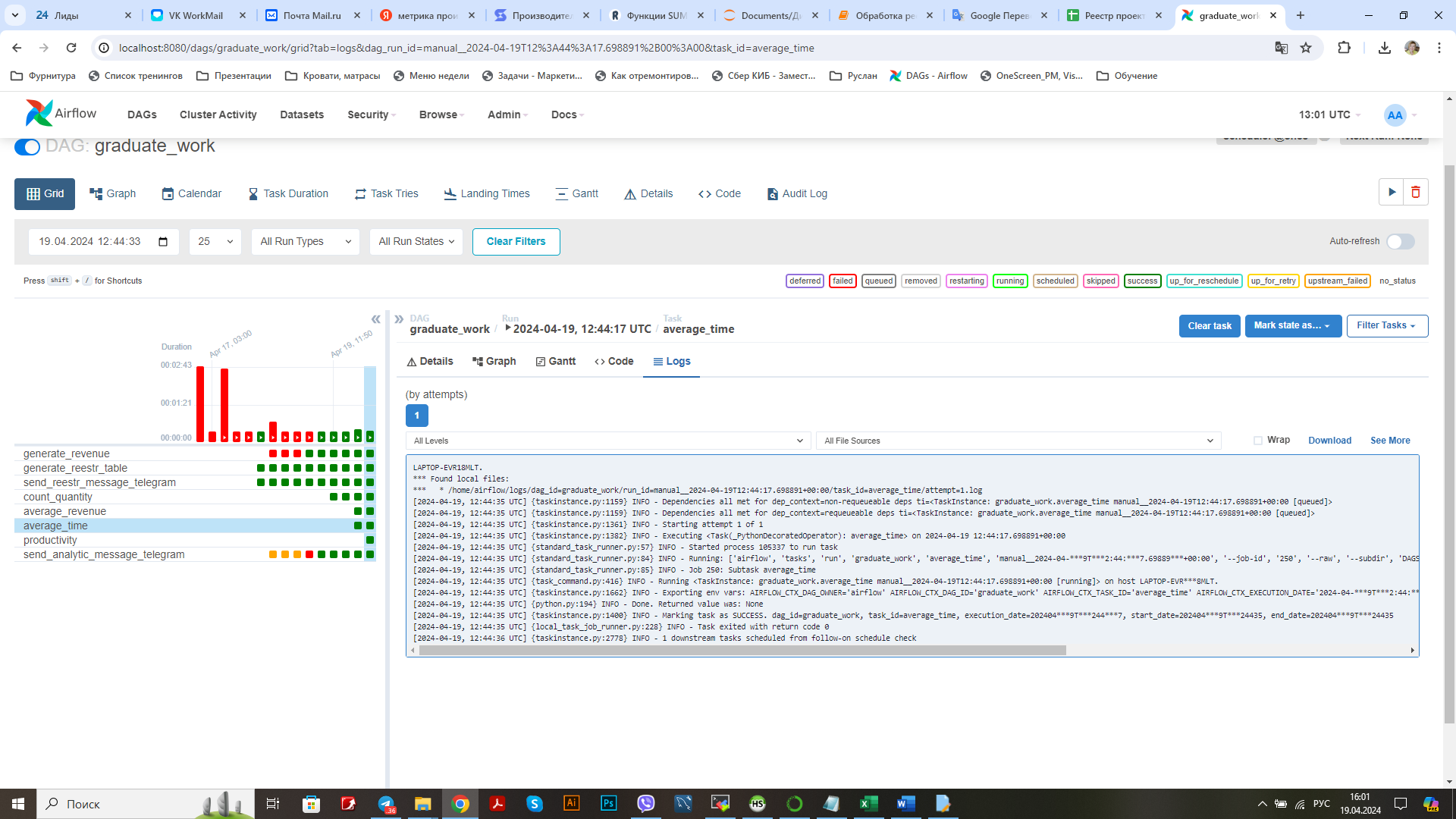
*Рисунок 43. Логи по Task count\_quantity*

1. Task average\_revenue



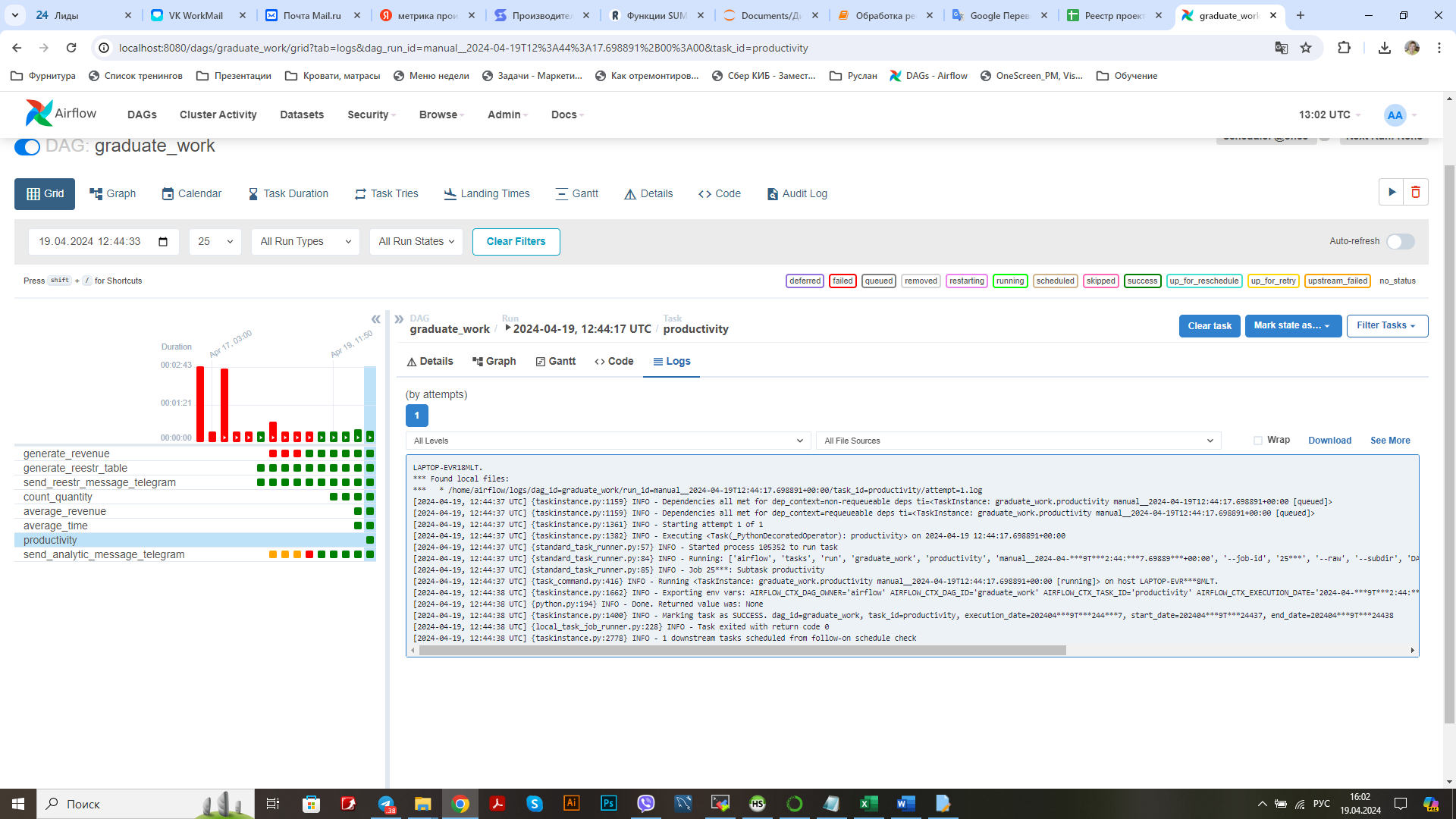
*Рисунок 44. Логи по Task average\_revenue*

1. Task average\_time



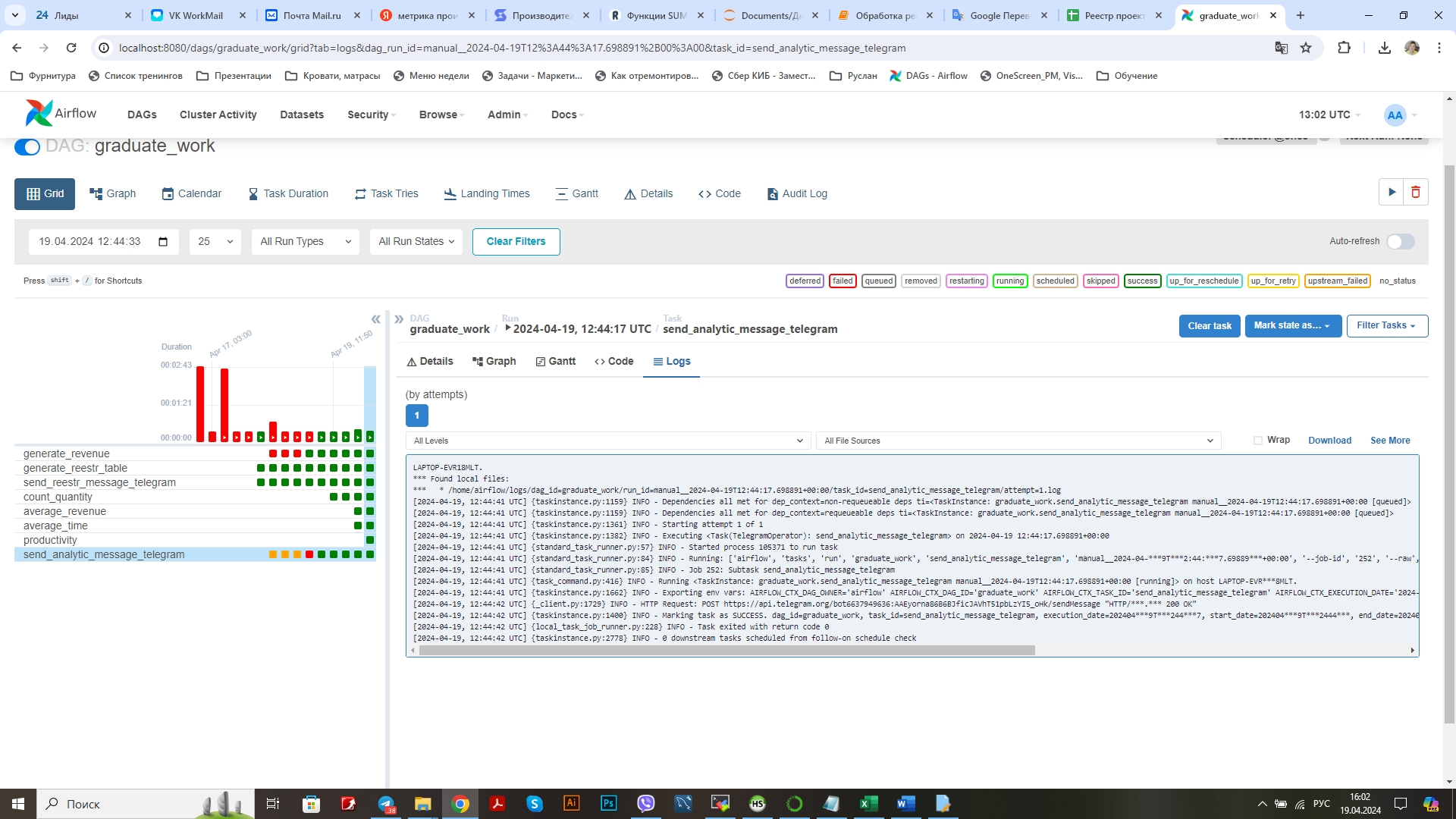
*Рисунок 45. Логи по Task average\_time*

1. Task productivity



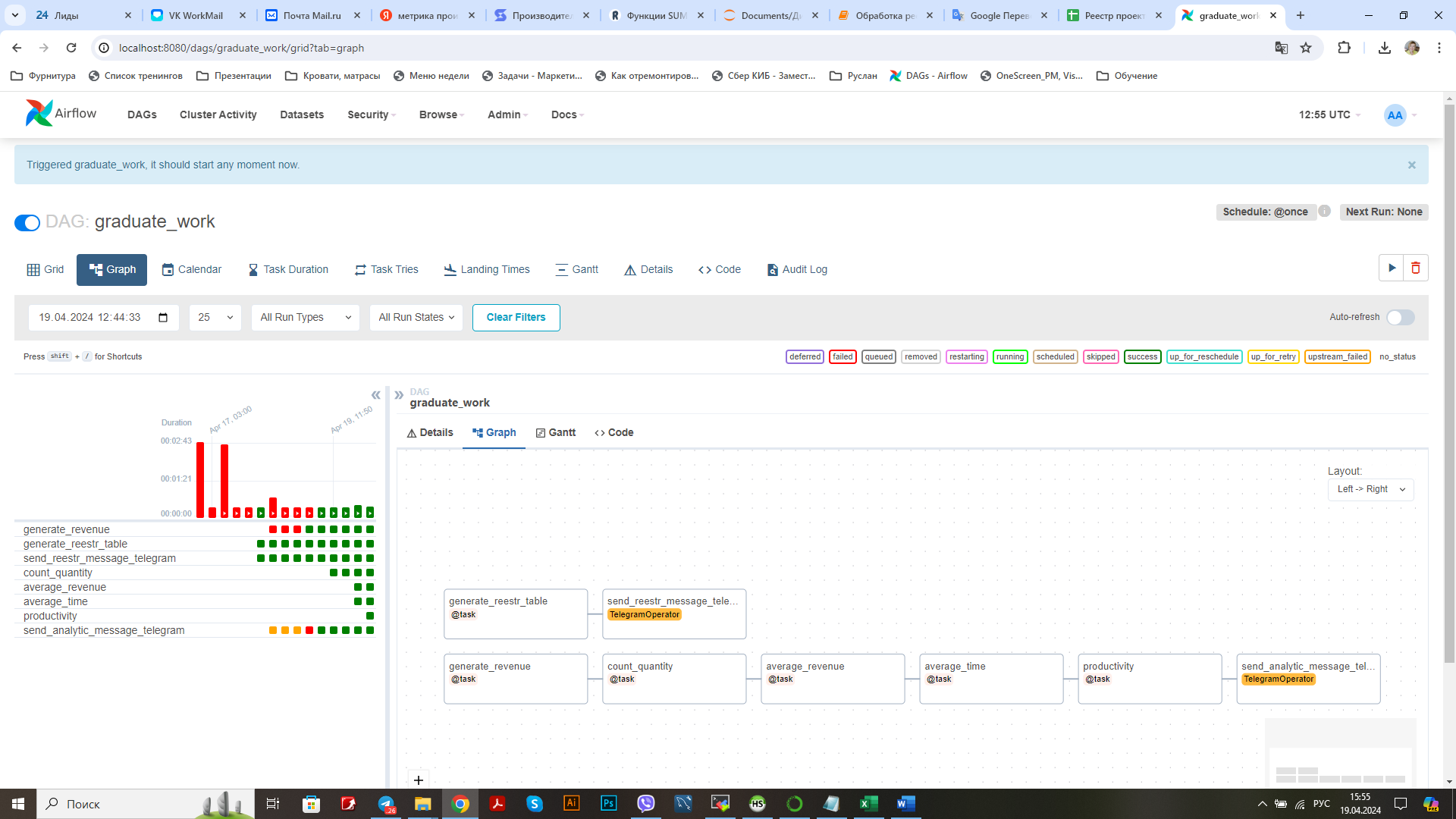
*Рисунок 46. Логи по Task productivity*

1. Task send\_analytic\_message\_telegram



*Рисунок 47. Логи по Task send\_analytic\_message\_telegram*

**Graph выполнения задач в Apache AirFlow:**



*Рисунок 48. Граф выполнения задач в AirFlow*

**3.5 Подведение итогов**

В результате выполнения всех этапов проекта удалось создать обновленную структуру выполнения бизнес-процесса с применением автоматизированной обработки и анализа ежемесячного реестра компании.

В ходе выполнения проекта было сделано следующее:

1. Выполнена подготовка датасета
2. Датасет перенесен в базу данных
3. Создано виртуальное окружение и запущен Apache AirFlow
4. Настроен доступ к базе данных через DAG в Apache AirFlow
5. Проведена обработка и анализ данных
6. Создан Telegram-бот
7. Настроена адресная отправка результатов анализа данных в Telegram-бот

Внедрение этих этапов позволяет решить поставленные задачи, а именно: оптимизировать структуру бизнес-процесса по обработке ежемесячных отчетов, настроить автоматизированную обработку данных и предоставление краткой аналитической информации напрямую руководителю. А также: высвободить ценное рабочее время сотрудников, исключить ошибки в расчетах и наладить конфиденциальную передачу данных.

**ЗАКЛЮЧЕНИЕ**

В данной дипломной работе на примере одного бизнес-процесса была продемонстрирована интеграция автоматизации процессов работы с данными. Внедрение этой оптимизации позволит компании решить ряд проблем, связанных с качеством, скоростью и конфиденциальность обработки данных.

Показанные инструменты можно масштабировать на другие бизнес-процессы компании, постепенно наладив полностью автоматизированную систему работы с данными.

В настоящее время во многих компаниях малого и среднего бизнеса присутствует ведение отчетности с помощью Excel или Google Sheets. Изложенные в работе шаги и инструменты автоматизации применимы для всех компаний, сталкивающиеся с проблемами, возникающими при ручной обработке данных.

Внедрение автоматизации процессов работы с данными не только повысит эффективность и качество работы компании, но также сэкономит время и ресурсы персонала, которые могут быть направлены на выполнение других задач. В результате, компания получит возможность значительно улучшить свою конкурентоспособность на рынке и повысить общую производительность своего бизнеса.

В качестве рекомендаций по доработке предложенных в проекте инструментов можно предложить следующие моменты:

* Настройку автоматизированной загрузки данных в датасет из Google Sheets
* Настройку выдачи информации по реестру через отправку запросов необходимых дат в Telegram-бот
* Настройку персонального расписания выдачи отчетов

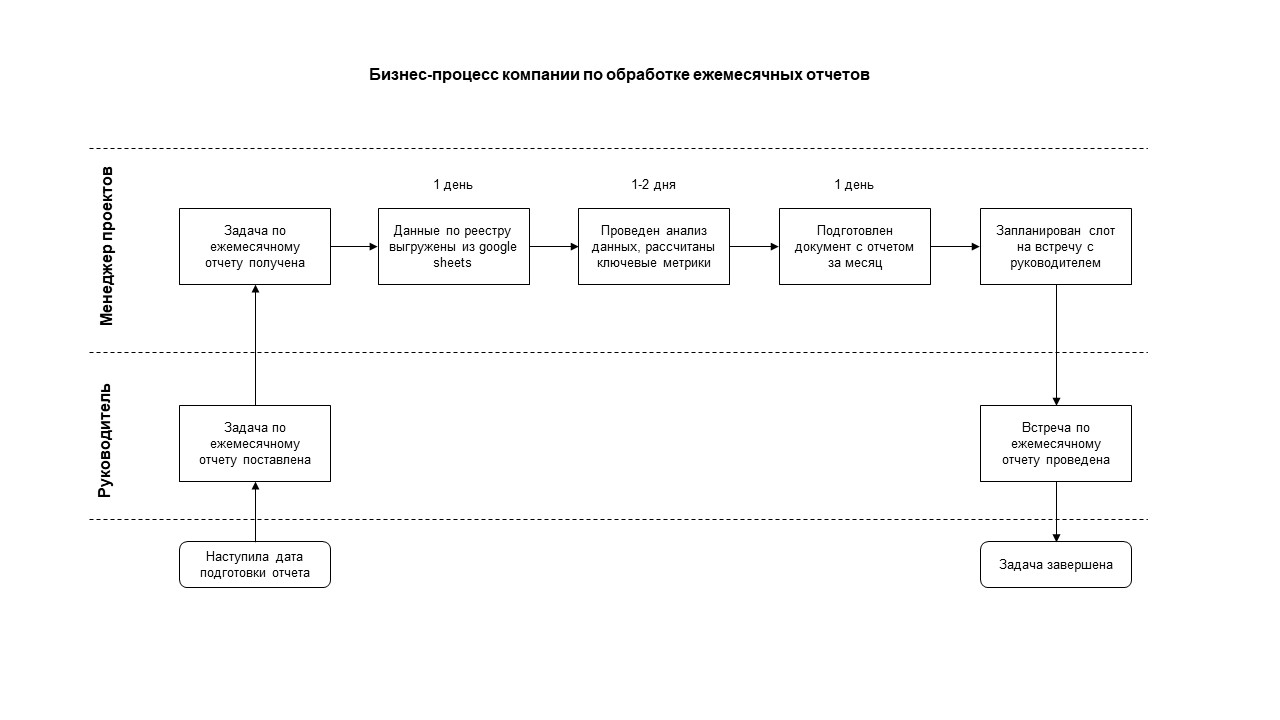
В заключении хочется отметить, что использование автоматизации процессов работы с данными также способствует улучшению аналитической отчетности компании и повышению прозрачности бизнес-процессов. Это позволит быстро и точно принимать стратегические решения, основанные на надежных данных, что в свою очередь поможет компании выйти на новый уровень развития и роста.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

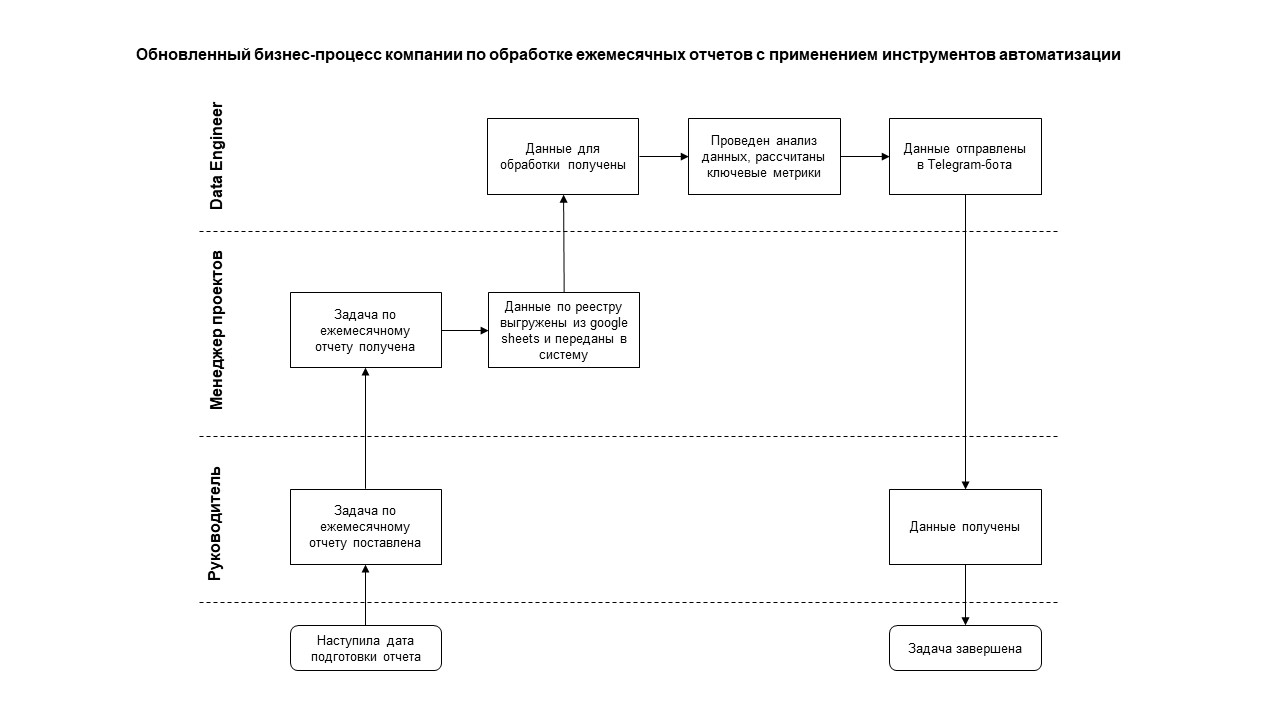
1. «Apache Airflow и конвейеры обработки данных», Харенслак Б., де Руйтер Дж. (пер. с англ. Д. А. Беликова)
2. «Data Science. Наука о данных с нуля», Джоэл Грас
3. «SQL для анализа данных. Расширенные методы преобразования данных для аналитики», Танимура Кэти
4. «Data Analysis Using SQL and Excel» («Анализ данных с использованием SQL и Excel»), Гордон С. Линофф
5. «Большие данные. Big Data. Учебник для вузов», А. Е. Журавлев, А. В. Макшанов, Л. Н. Тындыкарь
6. «Бизнес-процессы. Языки моделирования, методы, инструменты», Франк Шёнталер, Готфрид Фоссен, Андреас Обервайс, Томас Карле.
7. «Путь аналитика. Практическое руководство IT-специалиста», Андрей Перерва, Вера Иванова.
8. Статья «Необходимость и преимущества графического описания бизнес-процессов» https://www.businessstudio.ru/articles/article/neobkhodimost\_i\_preimushchestva\_graficheskogo\_opis/
9. «Основы Python для Data Science», Кеннеди Бермен
10. «Чистый код. Создание, анализ и рефакторинг» Роберт Мартин
11. «Python для всех: обработка данных с использованием Python 3» Чарльз Р. Северанс

**ПРИЛОЖЕНИЯ**

**Приложение 1**

****

**Приложение 2**

****

**Приложение 3**

**Ссылки на файлы, используемые в проекте:**

1. По ссылке <https://github.com/Polina-Kuzina/Graduate_work_DataEngineer.git> находятся файлы

* Файл «reestr proektov.csv»
* Файл «Обработка реестра.ipynb»
* Файл «reestr proektov\_work.xlsx»
* Файл «reestr\_proektov\_work.py»
* Файл «reestr\_dag.py»

2. По ссылке <https://t.me/reestr_proektov_bot> находится Telegram-бот «reestr\_proektov\_bot»