Система контроля деятельности персонала

Система контроля деятельности персонала (далее система) предназначена для сбора, хранения и обработки информации о составе структурного подразделения, а также его деятельности.

Вся система развернута в docker-контейнере (рис. 1).

Хранение данных осуществляется в БД – postgres 12.0 (порт 15432:5432) (рис. 2).

Чтение данных из БД возможна через pgAdmin4:6.8 (порт 5050:80) (рис. 3).

Автоматизация работы с данными осуществляется с помощью apache airflow 2.9.3 (порт 3000:3000) (рис. 4).

Отображение графиков осуществляется с помощью metabase 0.50.23 (порт 8080:8080) (рис. 5).

Все файлы должны лежать в папке /airflow/grant\_department (разработка велась для windows, поэтому полный путь - C:/airflow/grant\_department) (рис. 6).

Папка grant\_department содержит следующие папки:

1. backups – папка с бэкапами БД;
2. dags – папка с dag-файлами и конфигурационным json-файлом;
3. logs – папка логов airflow;
4. plugins – папка с плагинами airflow;
5. sql – папка и sql-скриптами для инициализации БД;
6. venv – папка, с виртуальным окружением python (нужна для разработки dag-файла);
7. work\_dara\_source – папка, с excel-файлами.

В случае запуска контейнера на linux необходимо поменять путь к папкам в docker-compose файле (например, C:/airflow/grant\_department –> /home/user/airflow/grant\_department)

Исходные данные

В качестве исходных данных используются следующие таблицы Excel:

1. Таблица employees (рис. 7) – предназначена для хранения актуальной информации об сотрудниках.

Таблица состоит из 4 обязательных для заполнения столбцов:

* «Имя» - содержит ФИО сотрудника;
* «Должность» - содержит актуальную должность сотрудника;
* «Табельный номер» - содержит уникальный номер, присваиваемый сотруднику, при трудоустройстве;
* «Дата изменения» - содержит последнюю дату изменения какого-либо из полей или дату приема на работу.

В случае увольнения сотрудника, соответствующая запись удаляется из таблицы employees.

1. Таблица standard (рис. 8) – предназначена для хранения актуальной информации о пунктах СТО, отображающих коэффициенты расчета трудоемкости.

Таблица состоит из 4 обязательных для заполнения столбцов:

* «Пункт стандарта» - содержит уникальный номер, согласно которому классифицируются работы;
* «Количество н/ч за единицу» - содержит актуальный коэффициент перевода производства единицы продукции в часы;
* «Текст пункта» - содержит текстовое описание данного пункта;
* «Дата обновления» - содержит последнюю дату изменения какого-либо из полей.

В случае удаления пункта СТО, соответствующая запись удаляется из таблицы standard.

Пункт стандарта 0, не должен впоследствии изменяться и должен всегда присутствовать в таблице, для обозначения работ, выполняемых вне данного стандарта.

1. Таблицы headsectors\_info\_xxx (рис. 9) – предназначены для хранения информации о выполненных работах.

Таблица состоит из 4 обязательных и 4 необязательных для заполнения столбцов:

* «Табельный номер» - обязательное поле, содержащее табельный номер сотрудника;
* «Имя» - обязательное поле, содержащее имя сотрудника (может отличаться от полного имени);
* «Дата» - обязательное поле, содержащее дату, выполняемой работы;
* «Описание работ» - обязательное поле, содержащее краткое описание выполненной работы;
* «Количество листов А4» - необязательное поле, содержащее фактическое количество листов А4 (в некоторых случаях единиц продукции), разработанных в рамках данной работы, за определенную дату. В случае, если поле не заполнено, в системе проставляется 0;
* «Количество часов» - необязательное поле, содержащее количество часов, в течение которых выполнялись указанные работы. В случае, если поле не заполнено, в системе проставляется 0;
* «Тема» - необязательное поле, содержащее название темы, в рамках которой выполнялась работа. В случае, если тема не указана, в системе проставляется «Вне темы»;
* «Пункт норматива» - необязательное поле, содержащее номер пункта СТО, по которому будет считаться трудоемкость. В случае, если пункт не указан, в системе проставляется пункт 0.

В случае, если пункт норматива – 0, то при расчете трудоемкости используется столбец «Количество часов», во всех остальных «Количество листов А4».

Excel таблицы должны лежать в папке airflow\grant\_department\works\_data\_source. В данной папке должны лежать 1 файл «employees», 1 файл «standard» и сколько угодно файлов, которые начинаются с «headsectors\_info» (рисунок 10).

Хранилище данных

В качестве хранилища данных используется БД postgres work\_info (рис. 11), с разделением данных по слоям:

1. stg-слой (рисунок 12) – слой, используемый для выгрузки данных в хранилище без каких-либо изменений, за исключением того, что все таблицы headsectors\_info сводятся в единую таблицу;
2. dds-слой (рисунок 13) – слой, в котором происходит хранение всех данных в таблицах, связанных между собой по модели данных «снежинка»;
3. cdm-слой (рисунок 14) – слой, в котором хранятся таблицы и изображения, используемые для построения отчетов.

stg-слой имеет в своем составе следующие таблицы:

1. employees – таблица для выгрузки данных из соответствующей таблицы excel;
2. standard – таблица для выгрузки данных из соответствующей таблицы excel;
3. headsectors\_info – таблица для выгрузки данных из всех соответствующих таблиц excel.

dds-слой имеет в своем составе следующие таблицы:

1. dm\_standard\_points – таблица для хранения коэффициентов расчета трудоемкости, построенная с механизмом SCD2 (колонки active\_from и active\_to). active\_to по умолчанию – NULL;
2. dm\_works\_info – таблица для хранения описания работ с ссылкой на пункт СТО, с учетом даты выполнения работы;
3. dm\_dates – таблица для хранения всех дат, используемый в системе;
4. dm\_topics – таблица для хранения всех тем, в рамках которых производились работы;
5. dm\_employees – таблица для хранения данных о сотрудниках, построенная с механизмом SCD2 (колонки active\_from и active\_to). active\_to по умолчанию – NULL;
6. fct\_works – таблица для хранения фактов выполненных работ, с рассчитанным количеством часов трудоемкости;
7. load\_info – таблица для хранения ключей загрузки, для отсутствия дублирования данных;
8. logs – таблица для хранения всех логов для данного слоя.

На каждую таблицу, за исключением таблицы logs, наложены триггеры на действия INSERT, UPDATE, DELETE, TRUNCATE, по которым производится логирование всех действий в данном слое.

cdm-слой имеет в своем составе следующие таблицы изображения:

1. full\_works\_last\_months – таблица, хранящая данные о суммарных наработках по каждому сотруднику за последние 3 месяца (включая текущий);
2. full\_works\_history - таблица, хранящая данные о суммарных наработках по каждому сотруднику за все периоды, предшествующие последним 3 месяцам;
3. load\_info - таблица для хранения ключей загрузки, для неиспользования одних и тех же данных;
4. position\_statistic – изображение, показывающее все изменения в составе сотрудников за год, начиная с момента использования изображения;
5. works\_this\_month – изображение, показывающее подробную информацию о работах за текущий месяц;
6. works\_previous\_month – изображение, показывающее подробную информацию о работах за предыдущий месяц;
7. works\_month\_before\_previous – изображение, показывающее подробную информацию о работах за позапрошлый месяц.

Скрипты по созданию каждого из слоев представлены в файлах:

1. create\_schemas.sql – скрипт создания схем;
2. create\_stg\_tables.sql – скрипт создания слоя stg;
3. create\_dds\_tables.sql – скрипт создания слоя dds;
4. create\_cdm\_tables.sql – скрипт создания слоя cdm;

Отображение данных

Визуализация данных осуществляется с помощью metabase. Вся отображаемая информация представлена в виде 2 dashboard:

1. «Статистика сотрудников» (рисунок 15) - dashboard, отображающий данные о приеме новых сотрудников, увольнении и переводе на новую должность старых сотрудников, которые осуществлялись за год, отсчитывая от текущей даты. Данный dashboard строится автоматически по ходу ETL - процесса;
2. «Результаты работ» (рисунок 16-19) - dashboard, отображающий данные о количестве часов, наработанных каждым сотрудников по отдельности, в текущем, предыдущем и позапрошлом месяце. Статистика ведется общая и по каждой системе по-отдельности. Данный dashboard строится автоматически по ходу ETL - процесса.

ETL – процесс

Реализация ETL-процесса представлена в dag-файле «employees\_data» в папке /airflow/grant\_department/dags.

ETL – процесс построен, при помощи apache airflow (рисунок 20).

Весь процесс представлен в виде следующих шагов:

1. backup\_database – создание backup-файла текущего состояния БД;
2. stg\_upload – группа задач, по выгрузке данных из excel-таблиц в stg-слой:
   1. stg\_standard\_upload – выгрузка данных из файла standard;
   2. stg\_headsectors\_info\_upload – выгрузка данных из файлов headsectors\_info;
   3. stg\_employees\_upload – выгрузка данных из файла employees;
3. dds\_upload – заполнение таблиц dds слоя на основе данных из stg:
   1. dm\_standart\_points\_update – заполнение таблицы dds.dm\_standart\_points;
   2. dm\_tables\_upload - группа задач, по заполнению таблиц измерений dds-слоя:
      1. dm\_works\_info\_update - заполнение таблицы dds.dm\_works\_info;
      2. dm\_dates\_update - заполнение таблицы dds.dm\_ dates;
      3. dm\_employees\_update - заполнение таблицы dds.dm\_ employees;
      4. dm\_topics\_update - заполнение таблицы dds.dm\_ topics;
   3. fct\_works\_update – заполнение таблицы фактов dds.fct\_works;
4. cdm\_full\_works\_fullfil – заполнение таблиц cdm.full\_works\_history, cdm.full\_works\_last\_months;
5. dashboards\_create – создание dashboards:
   1. dashboard\_full\_works\_create – создание dashboard «Результаты работ»;
   2. dashboard\_employees\_statistic\_create – создание dashboard «Статистика сотрудников».

Шаг backup\_database создает файл с помощью pg\_dump в папке /airflow/grant\_department/backups (рисунок 21).

В шаге stg\_headsectors\_info\_upload происходит заполнение пустых полей таблиц headsectors\_info\_xxx валидными данными (смотри раздел «Хранилище данных»).

На шагах dm\_standart\_points\_update и dm\_employees\_update, в случае удаления какой-либо строки (полное удаление пункта СТО или увольнение сотрудника), в соответствующей таблице измерений в поле active\_to проставляется дата запуска ETL-процесса.

В связи с тем, что количество тем работ, может меняться в течение одного месяца, реализован шаг dashboard\_full\_works\_create, который создает dashboard, состав вкладок, которого может меняться.

В системе представлена защита от повторного чтения данных (рисунок 22 - 23):

1. На шаге dm\_employees\_update из stg.employees берутся все данные только при первом запуске ETL-процесса, во время которого в таблицу dds.load\_info записывается дата последнего обновления данных пользователей из stg.employees (key=’employees\_last\_date’). При всех последующих чтениях stg\_employees будут учитываться данные только с датой изменения, которая больше записанной в dds.load\_info. В случае наличия подходящих данных данные в dds.load\_info обновляются;
2. На шаге dm\_standard\_update из stg.standard берутся все данные только при первом запуске ETL-процесса, во время которого в таблицу dds.load\_info записывается дата последнего обновления данных пунктов СТО из stg.standard (key=‘standard\_last\_date’). При всех последующих чтениях stg\_standard будут учитываться данные только с датой изменения, которая больше записанной в dds.load\_info. В случае наличия подходящих данных данные в dds.load\_info обновляются;
3. На шагах dm\_works\_info\_update и fct\_works\_update берутся все данные только при первом запуске ETL-процесса, во время которого в таблицу dds.load\_info записываются максимальные даты из stg.headsectors\_info для каждого табельного номера (key=’works\_info\_last\_dates’ и ’works\_last\_dates’). При всех последующих чтениях stg.headsectors будут учитываться данные только с датой изменения, которая больше записанной в dds.load\_info. В случае наличия подходящих данных данные в dds.load\_info обновляются;
4. На шаге cdm\_full\_works\_fullfil из dds.fct\_works берутся все данные только при первом запуске ETL-процесса, во время которого в таблицу cdm.load\_info записывается максимальный id записей (key=’full\_works\_last\_id’). При всех последующих чтениях dds.fct\_works будут учитываться данные только с id, который больше записанной в cdm.load\_info. В случае наличия подходящих данных данные в cdm.load\_info обновляются.

ETL-процесс настроен на автоматический запуск в 6:00 по нулевому меридиану (9:00 по Москве). В случае ошибки предусмотрены 3 повторные попытки запуска ETL-процесса с интервалом 5 секунд после неудачи.

Запрещается заполнять данные сотрудника в таблице headsectors\_info\_xxx за текущую дату, до запуска ETL-процесса для загрузки предыдущих данных, в связи с тем, что последующие данные по этому сотруднику за это число будут проигнорированы.

Первичная инициализация

При первом запуске необходимо проинициализировать систему. Для этого необходимо выполнить следующую последовательность шагов:

1. Развернуть docker-образ через docker-compose;
2. Проконтролировать доступ чтение/запись к новым папкам (airflow/grant\_department);
3. Файлы из папок dag и sql\_scripts скопировать в новые папки airflow/grant\_department/dags и airflow/grant\_department/sql соответственно;
4. В apache airflow должны появиться процессы – «init\_airflow», «init\_metabase», «init\_airflow», «init\_postgres» (рисунок 24);
5. В случае необходимости необходимо изменить содержимое файла variables.json, касающееся metabase (за исключением conn\_id, host, port);
6. Вручную запустить каждый из процессов инициализации.

Процесс инициализации airflow (рисунок 25) состоит из следующих шагов:

1. postgres\_connection\_create – создание подключения для postgres (BD\_INFO\_WORK);
2. metabse\_connection\_create – создание подключения для metabase (METABASE\_DASHBOARDS\_CONNECTION.

Процесс инициализации postgres (рисунок 26) состоит из следующих шагов:

1. create\_chemas – создание схем stg, dds, cdm;
2. create\_layers – создание таблиц, изображений, функций и триггеров в каждой из схем:
   1. stg – создание таблиц в схеме stg;
   2. dds – создание таблиц, функций и триггеров в схеме dds;
   3. cdm – создание таблиц и изображений в схеме cdm.

Процесс инициализации metabase (рисунок 27) состоит из следующих шагов:

1. metabase\_registration – создание root пользователя в metabase.

Приложение

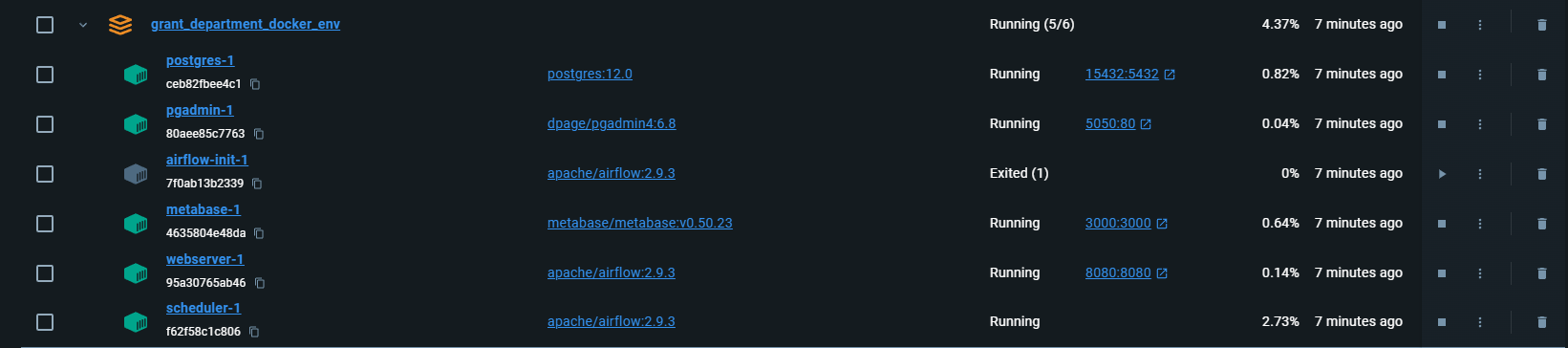


Рисунок 1 – docker-контейнер

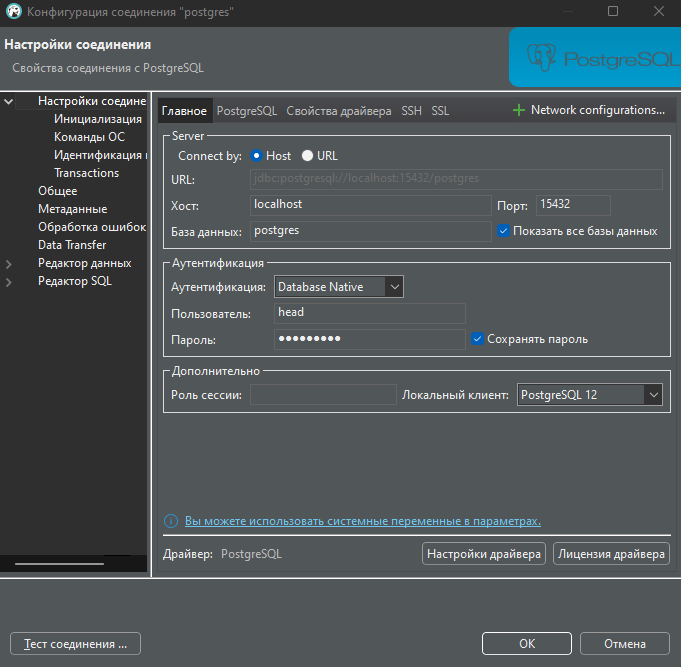


Рисунок 2 – Настройки соединения к БД

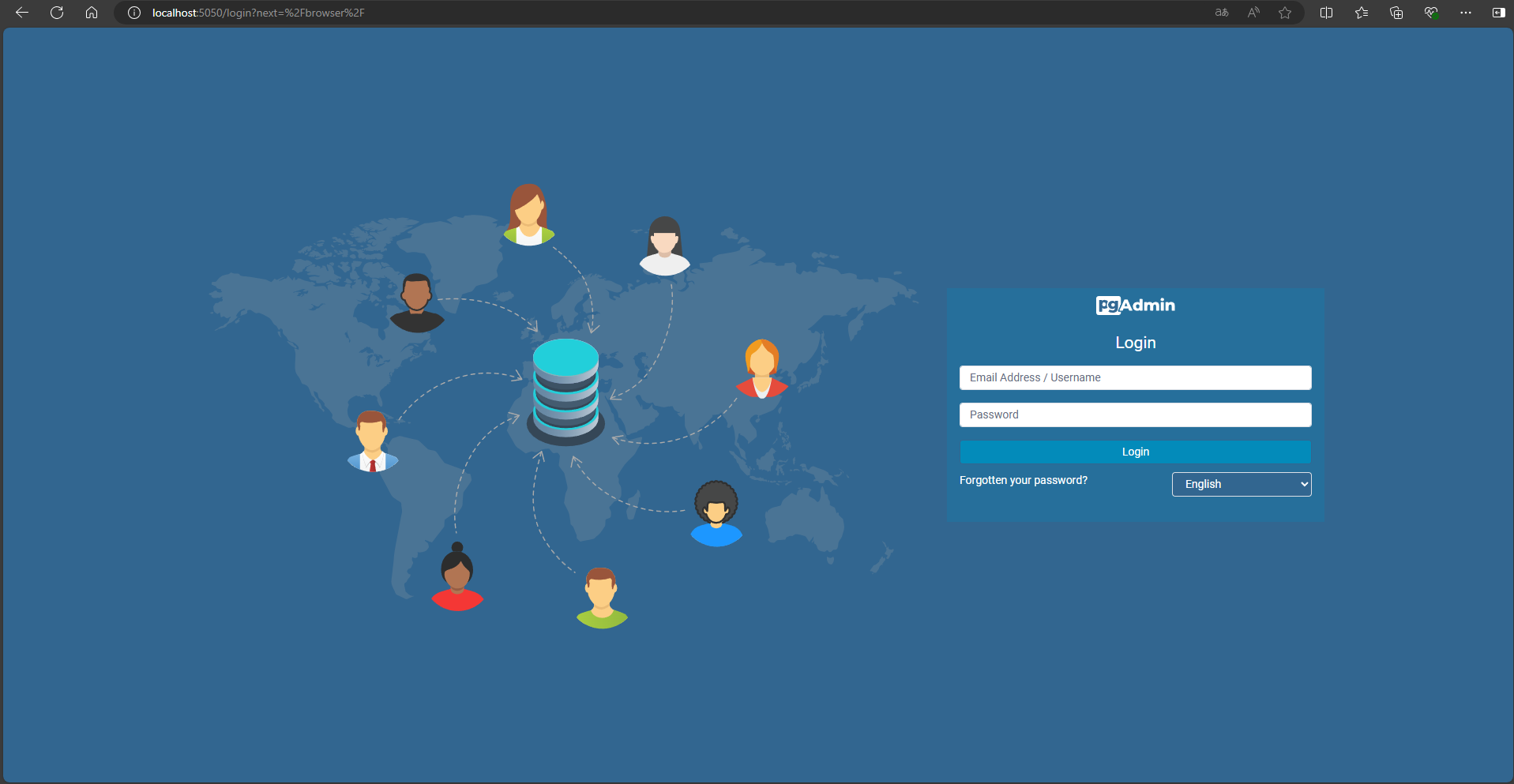


Рисунок 3 – Окно авторизации pgAdmin

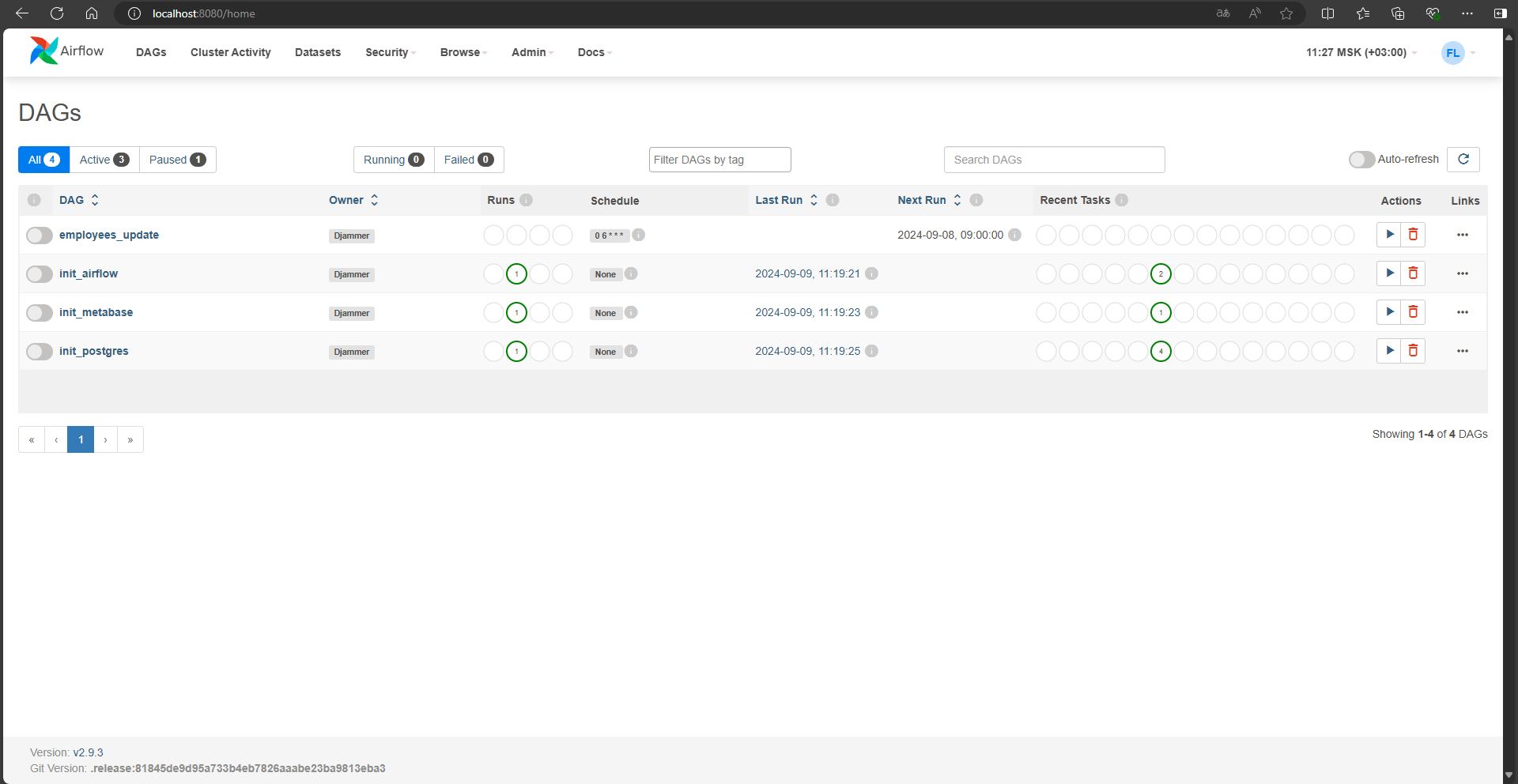


Рисунок 4 – Главная страница apache airflow

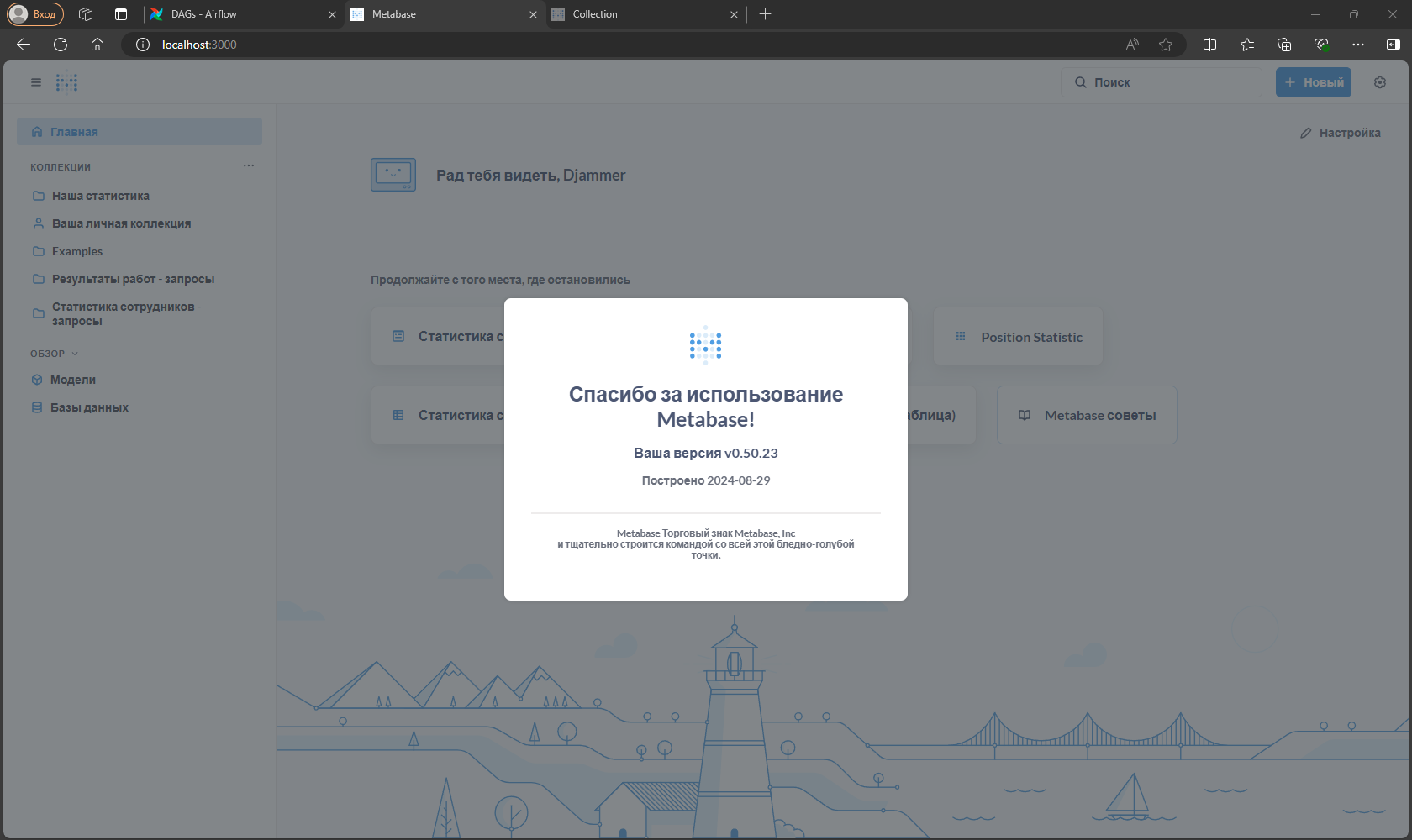


Рисунок 5 – Версия metabase

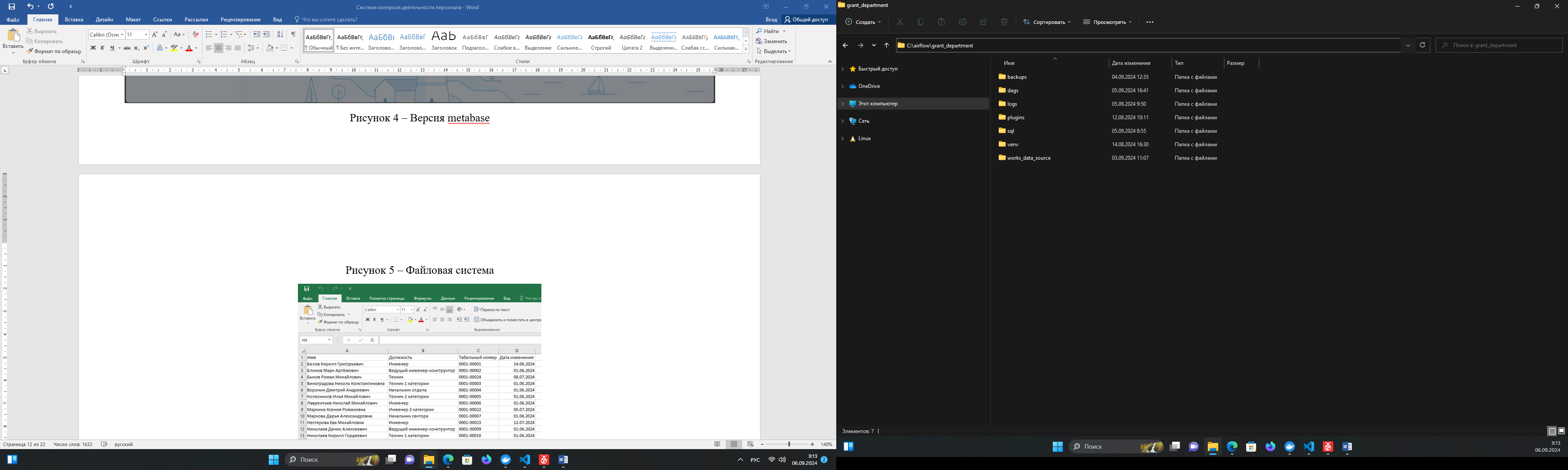


Рисунок 6 – Файловая система

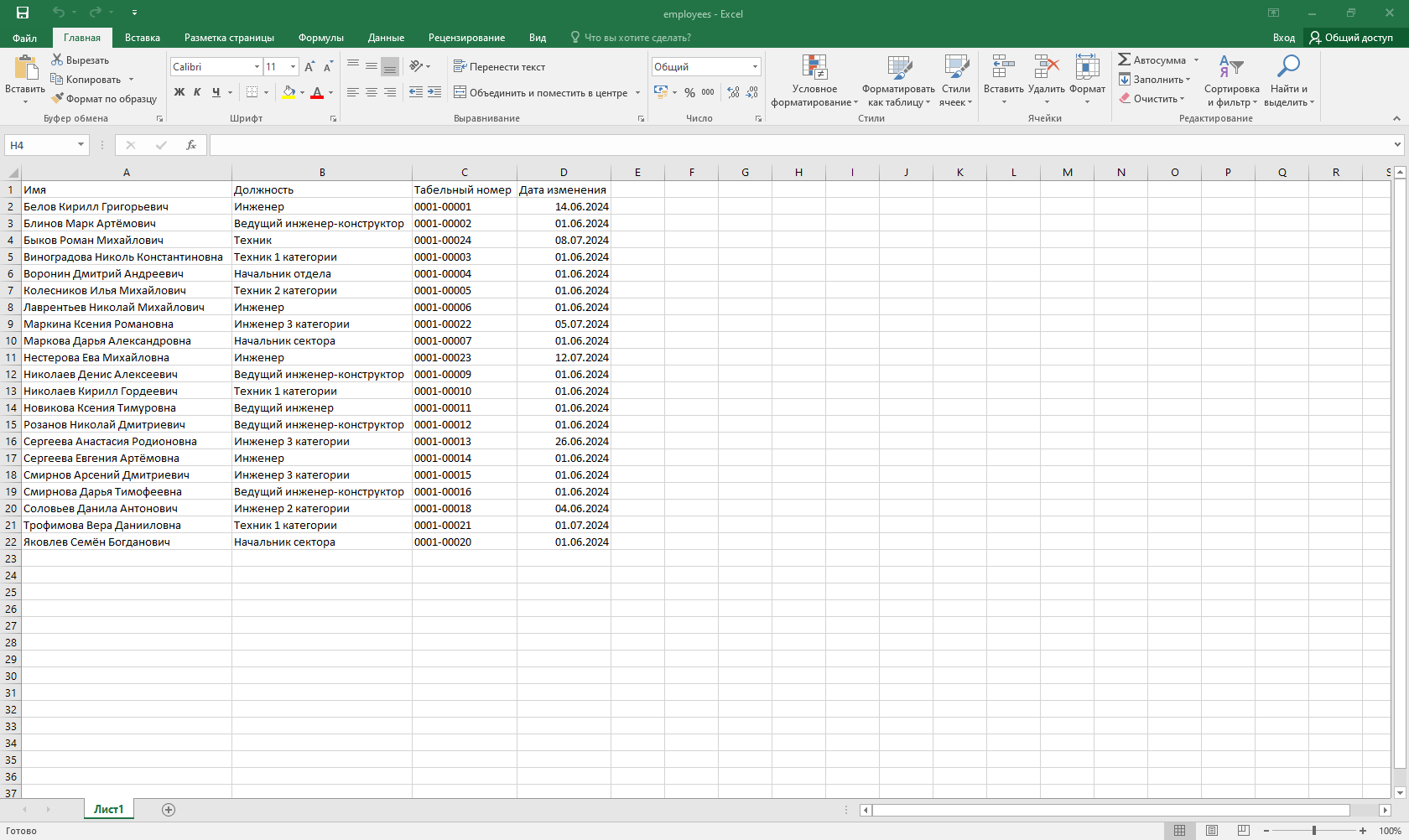


Рисунок 7 – таблица employees

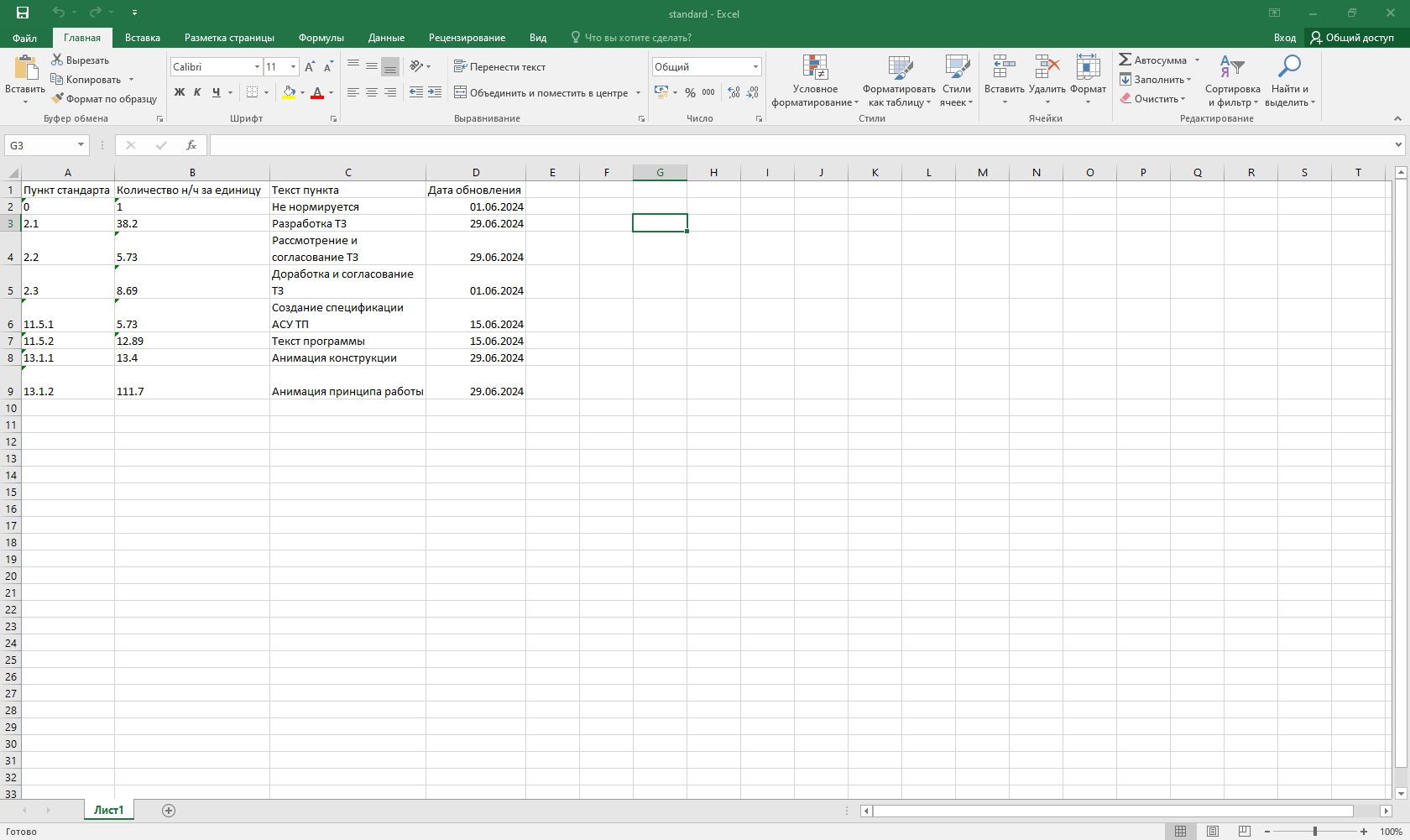


Рисунок 8 – таблица standard

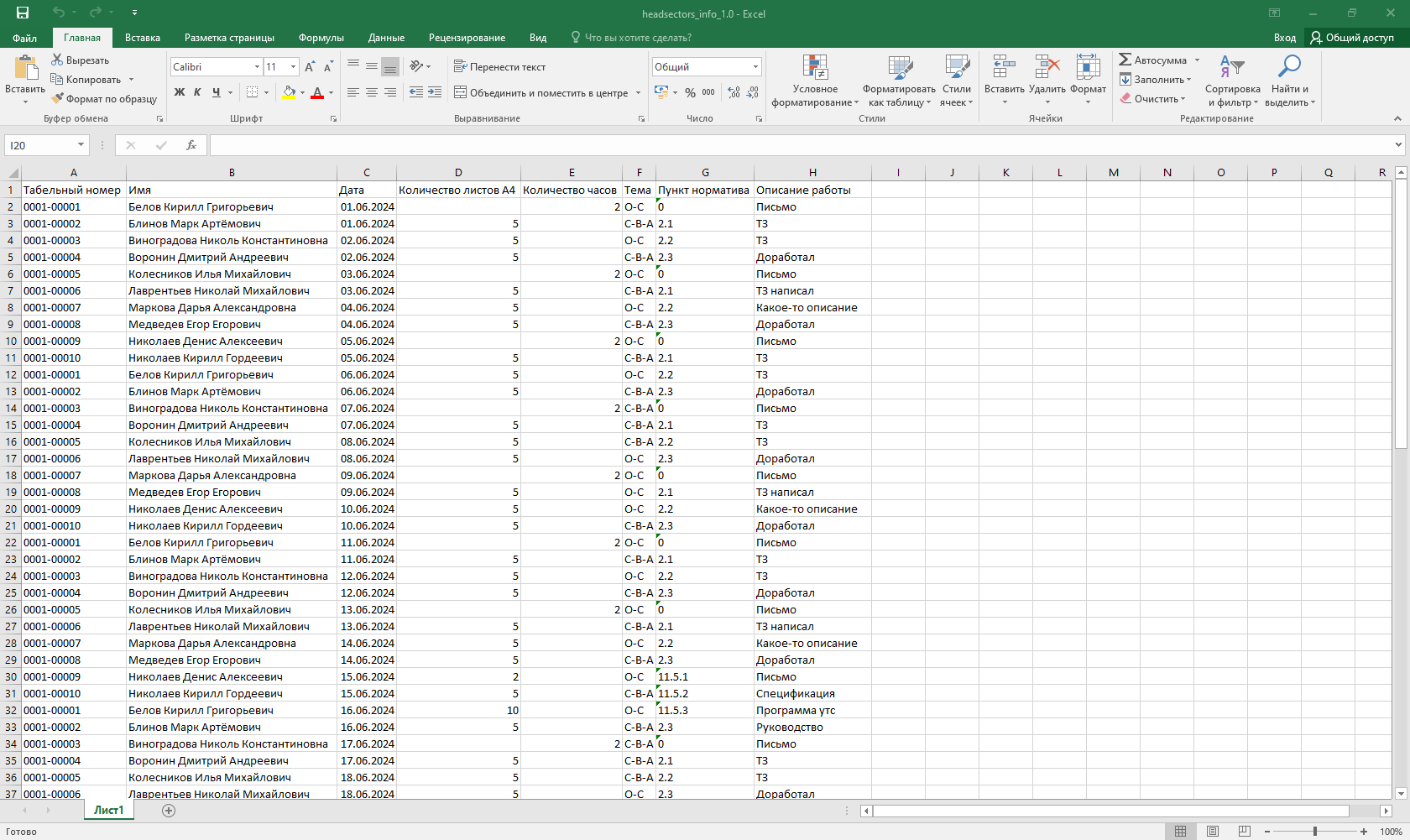


Рисунок 9 – таблица headsectors\_info

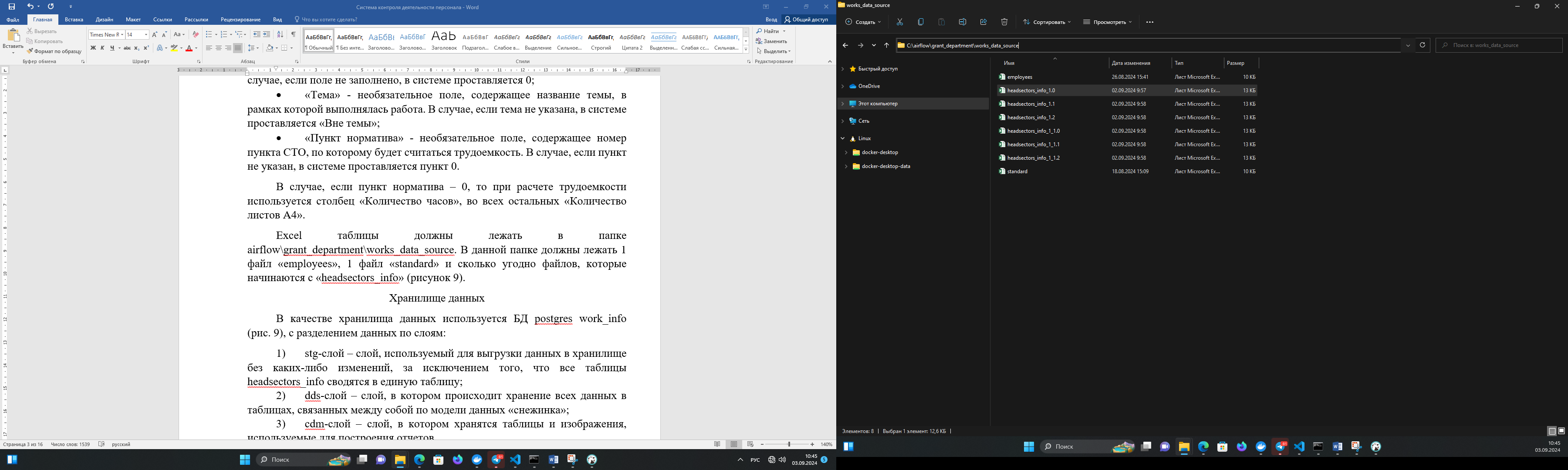


Рисунок 10 – папка works\_data\_source

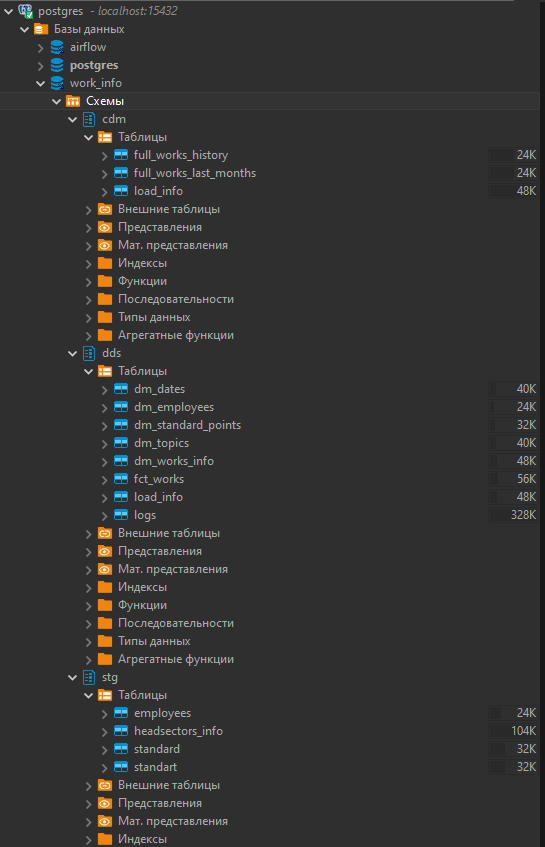


Рисунок 11 - БД work\_info

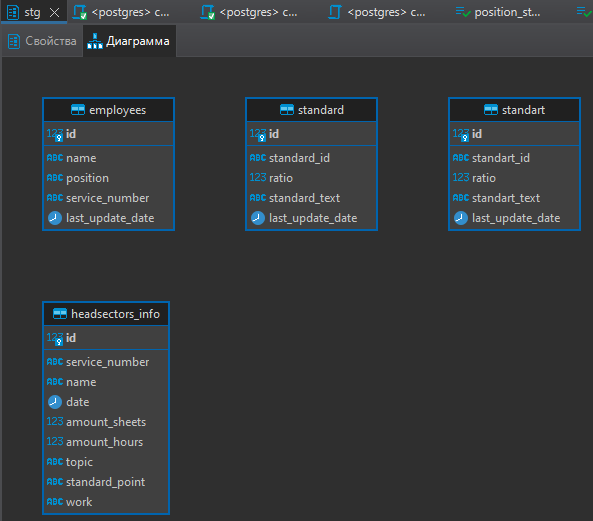


Рисунок 12 – stg – слой

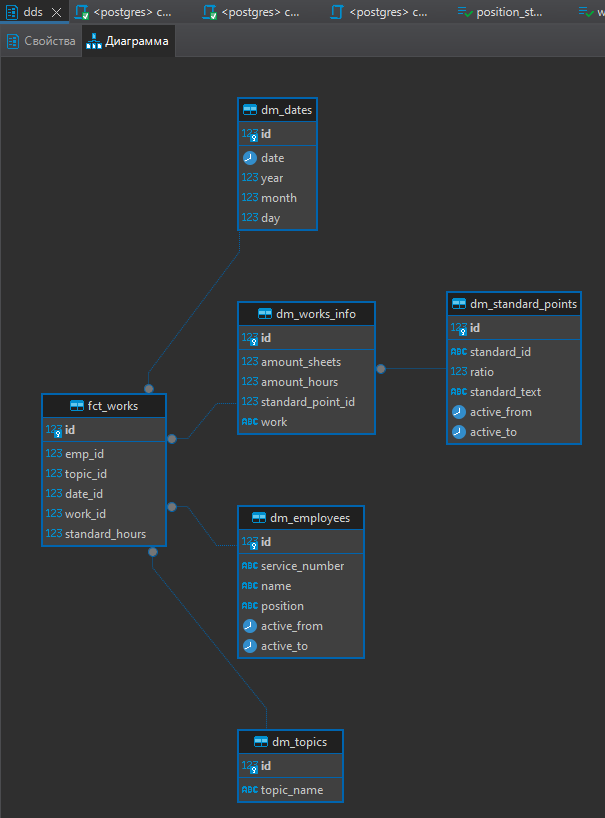


Рисунок 13 – dds – слой

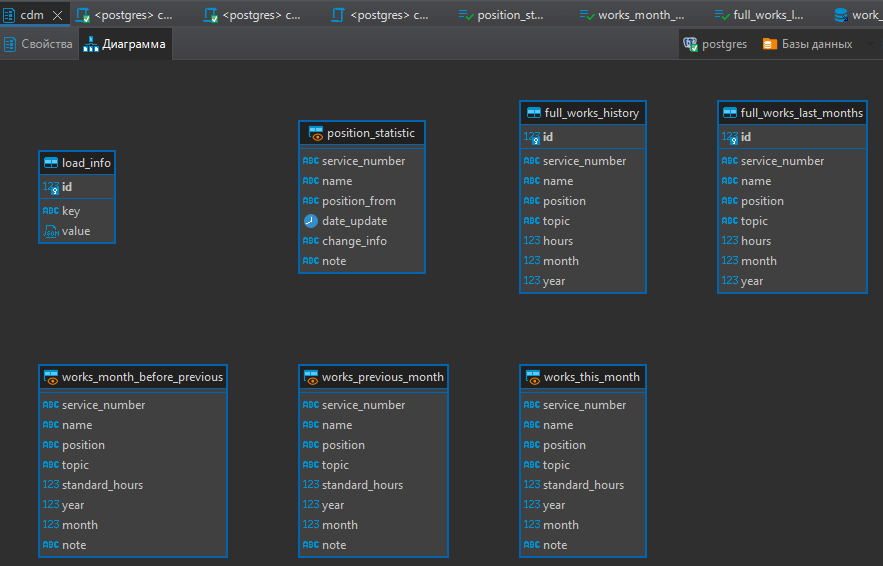


Рисунок 14 – cdm – слой

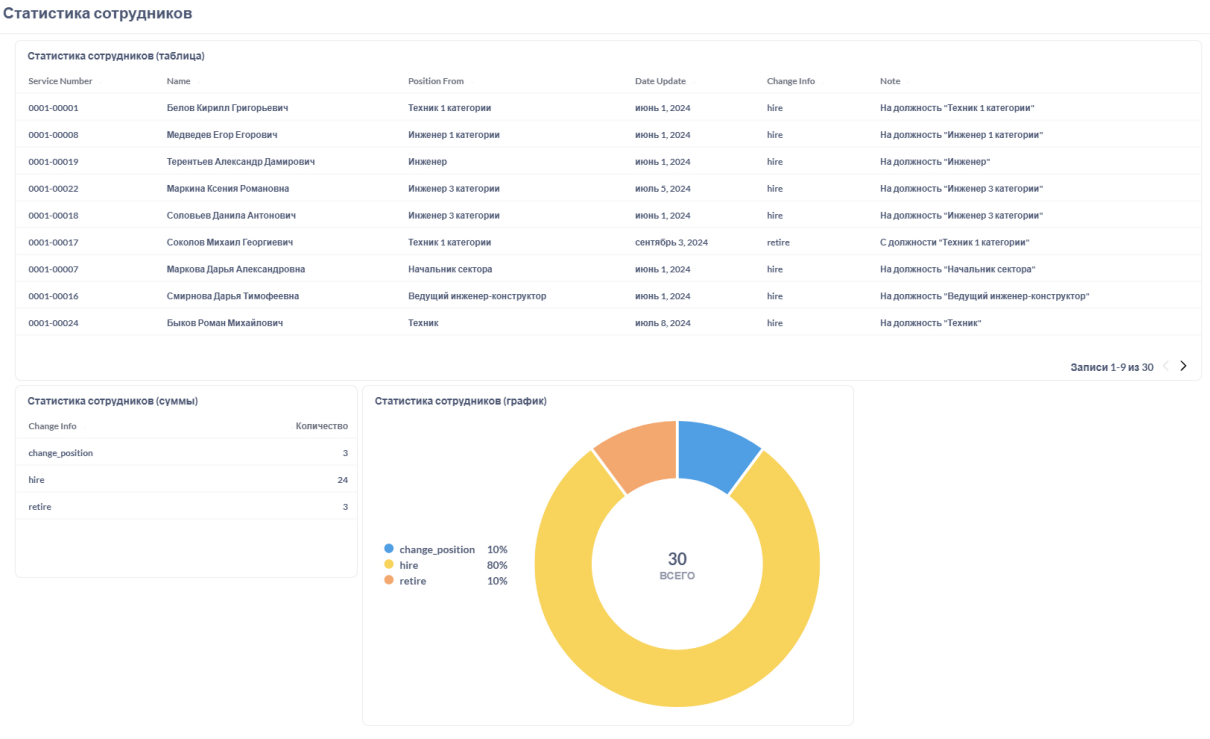


Рисунок 15 – dashboard «Статистика сотрудников»

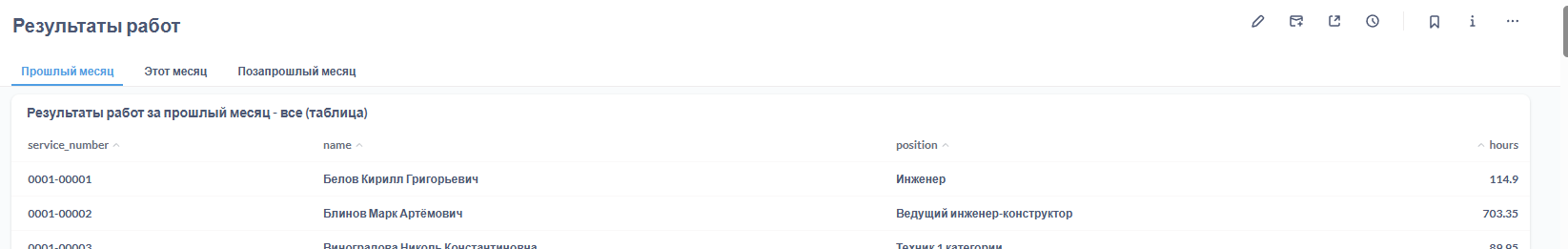


Рисунок 16 – dashboard «Результаты работ» - часть 1

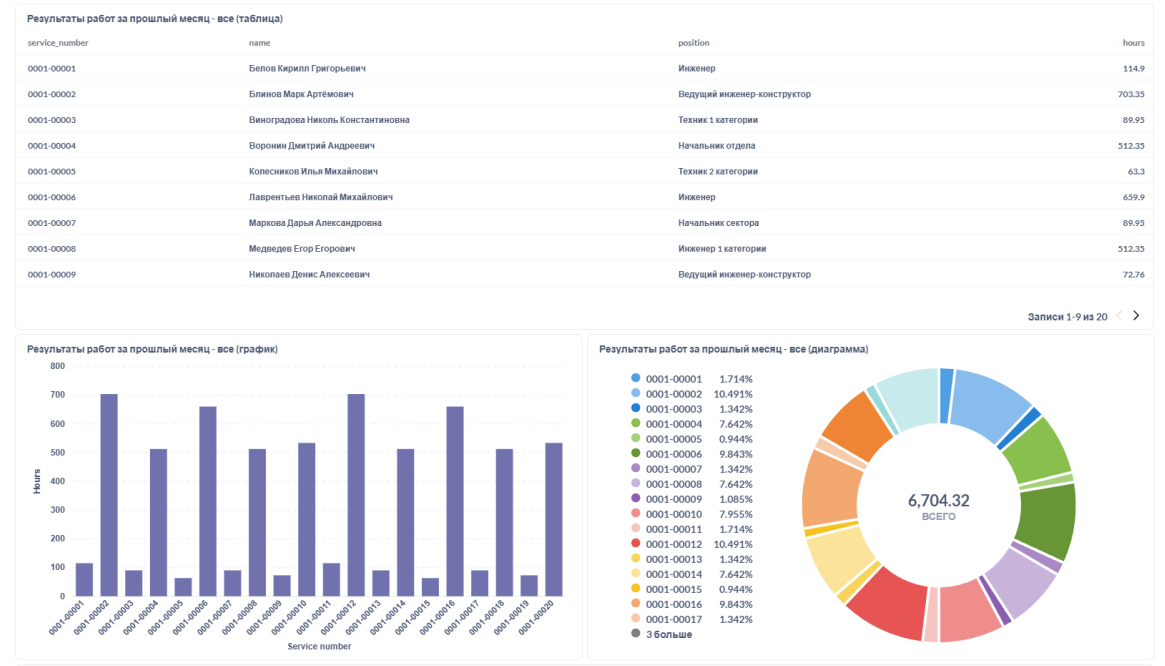


Рисунок 17 – dashboard «Результаты работ» - часть 2

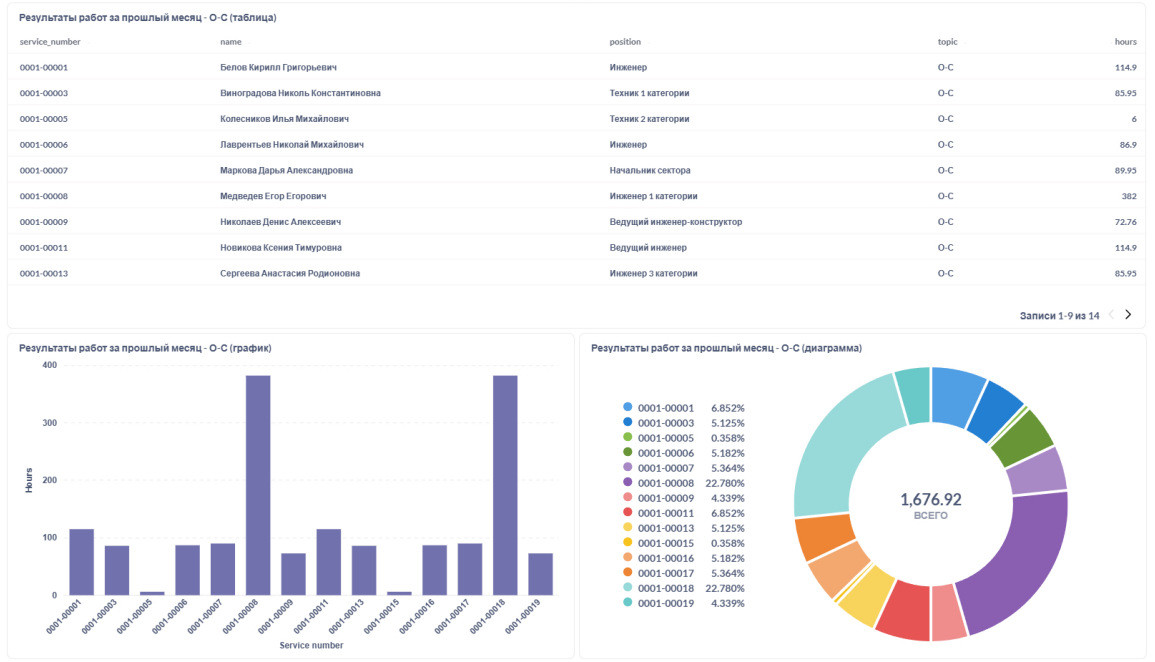


Рисунок 18 – dashboard «Результаты работ» - часть 3

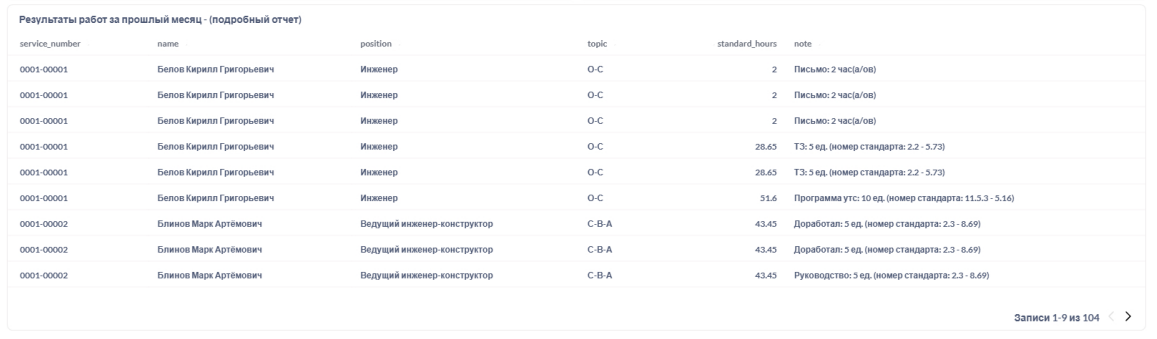


Рисунок 19 – dashboard «Результаты работ» - часть 4

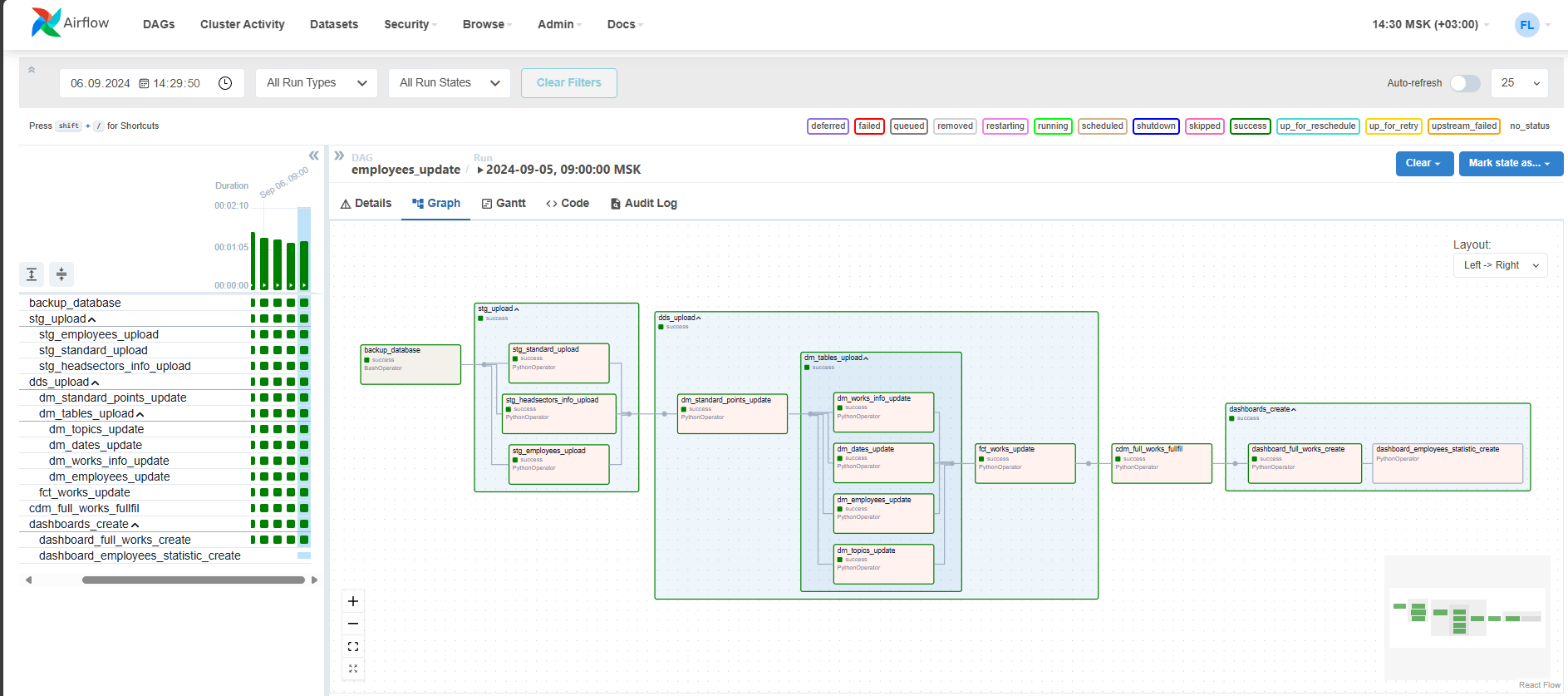


Рисунок 20 – ETL процесс

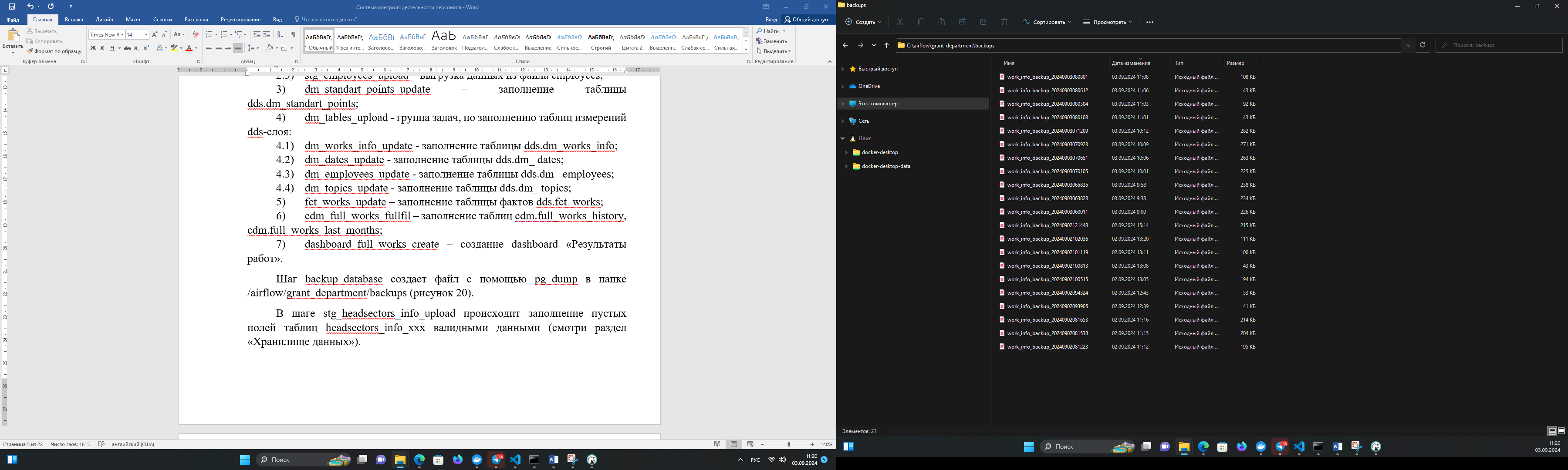


Рисунок 21 – backup – файлы

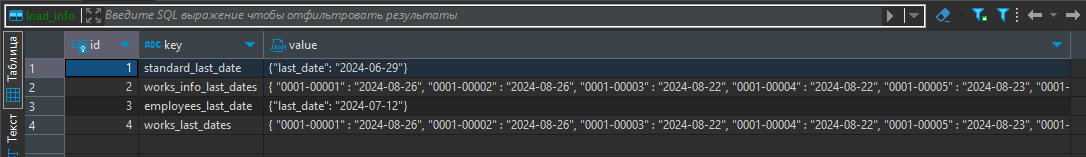


Рисунок 22 – Таблица dds.load\_info

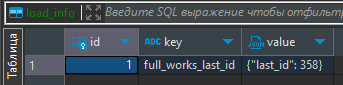


Рисунок 23 – Таблица cdm.load\_info

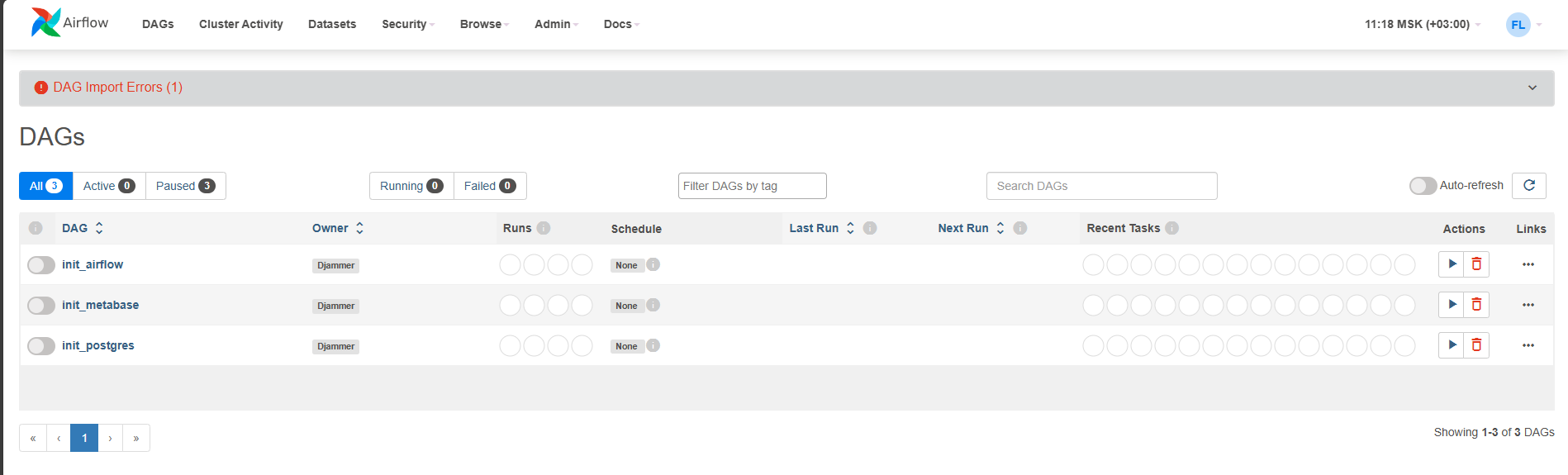


Рисунок 24 – dag-инициализации

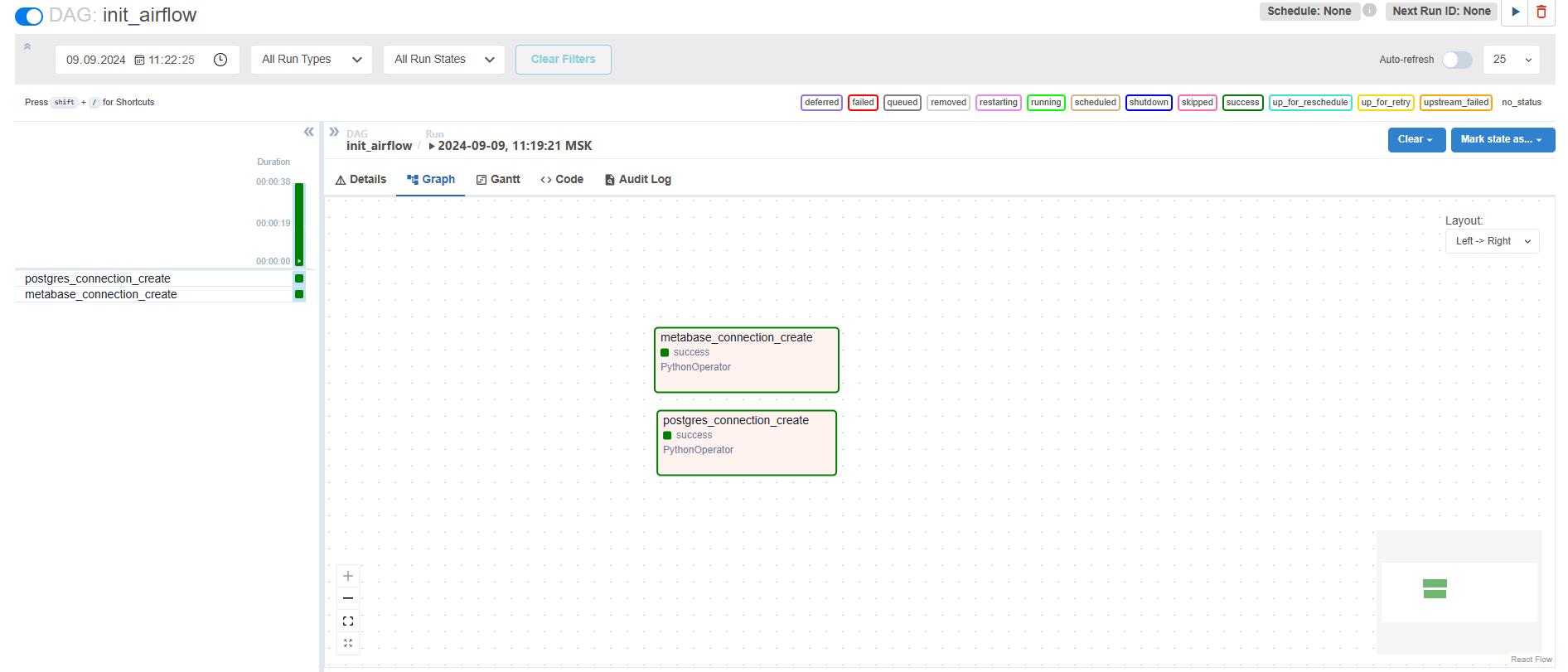
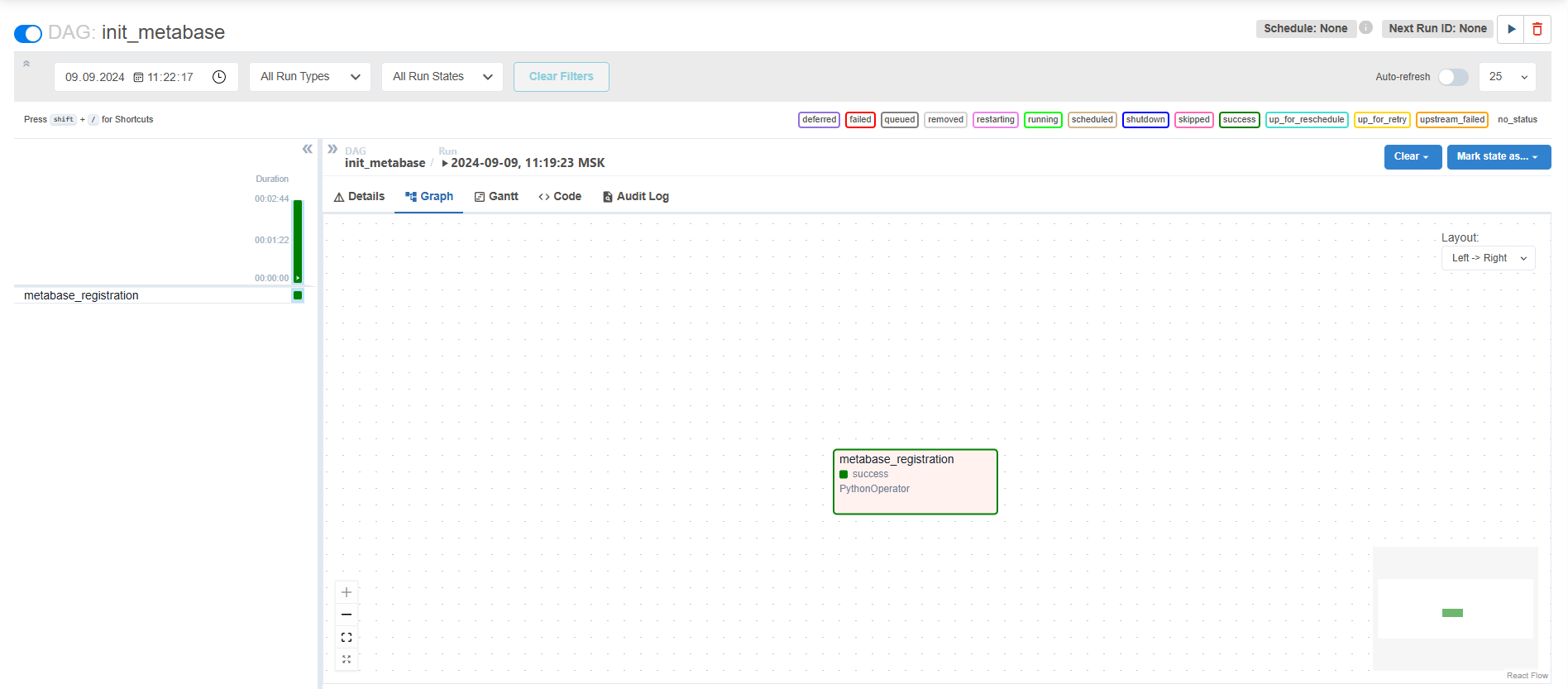


Рисунок 25 – dag init\_airflow

  
Рисунок 26 - dag init\_metabase

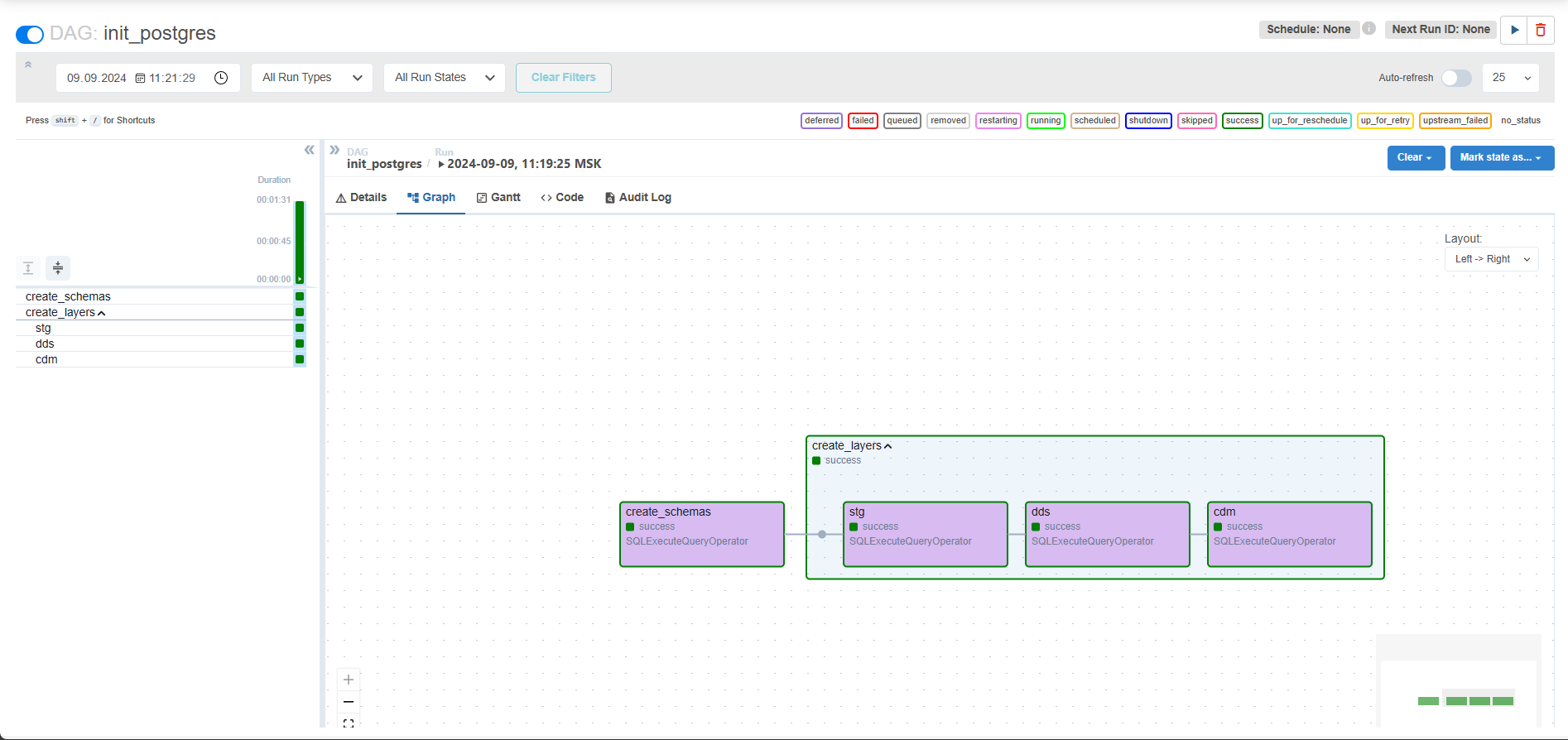


Рисунок 27 – dag init\_postgres