**АРХИТЕКТУРА РЕШЕНИЯ**

**Проект «Анализ профессиональных навыков сотрудников департамента аналитических решений»**

**Исполнители:**

Исаева Ирина Александровна, стажер

*(Ф.И.О., должность)*

Годзун Егор Андреевич, стажер

*(Ф.И.О., должность)*

**Дата:** 02.08.2024

Содержание

[1. Схема архитектуры решения 3](#_Toc173420928)

[2. Описание компонентов 4](#_Toc173420929)

[2.1. Management layer (слой управления) 4](#_Toc173420930)

[2.2. Описание ETL-потока 5](#_Toc173420931)

[2.3. Storage layer (слой хранения) 7](#_Toc173420932)

[2.4. Presentation layer (слой представления) 7](#_Toc173420933)

[3. Описание используемых сущностей 8](#_Toc173420934)

1. **Схема архитектуры решения**

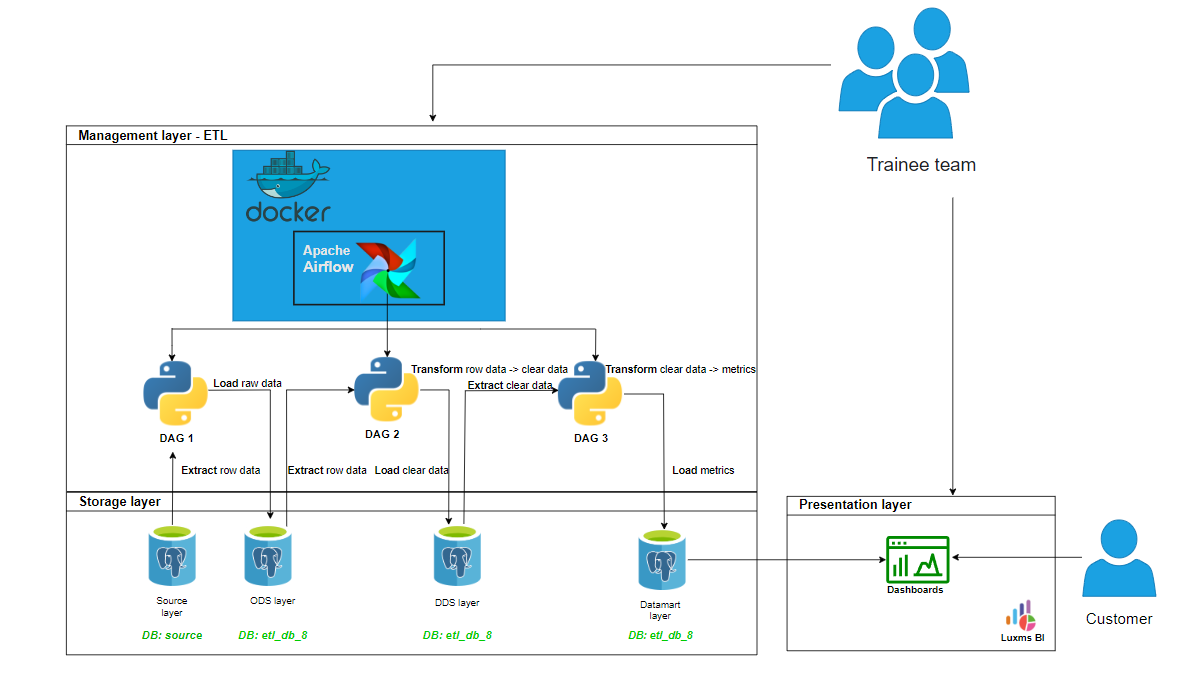
****

Рисунок 1 – Схема архитектуры решения

В схеме архитектуры решения, представленной на рисунке 1, определены следующие роли: команда стажеров (trainee team) и заказчик (customer). Для каждой из ролей предусмотрены свои компоненты.

Команда стажеров использует и имеет доступ к таким компонентам, как:

- Docker, осуществляющий контейнеризацию и управления компонентами;

- Airflow, применяемый для планирования выполнения файлов DAG;

- Файлы DAG (скрипты python), разрабатываемые для реализации процессов излечения, трансформации и загрузки данных;

- PostgreSQL, позволяющая получить доступ к хранимым данным в базах данных таких, как source и etl\_db\_8;

- Luxms BI, применяемый для создания дашбордов и генерации отчетности.

Заказчик использует Luxms BI для просмотра дашбордов, которые отображают ключевые метрики в понятном для заказчика виде.

1. **Описание компонентов**

Схема архитектуры решения, представленная в разделе 1, состоит из 3 слоев таких, как слой управления, хранения и представления данных. Используемое ПО с соответствующими версиями представлено в Таблице 1.

Таблица 1 - Используемое ПО и версии ПО

|  |  |
| --- | --- |
| **Наименование ПО** | **Версия ПО** |
| Docker Desktop | 26.0.0 |
| Airflow | 2.9.2 |
| PostgreSQL | 16.3 |
| Python | 3.11.9 |
| Luxms BI | 9.3 |

* 1. **Management layer (слой управления)**

В Docker Desktop поднят контейнер с Airflow, к которому примонтирована папка /**hr-project/dags** для хранения DAG’ов (скриптов python). Airflow оркестрирует ETL-поток на базе трех DAG’ов, с помощь которых реализуется извлечение (extract) «сырых» данных (raw data) из базы данных source в слой ODS в базу данных etl\_db\_8, извлечение «сырых» данных из слоя ODS, трансформация и загрузка «очищенных» данных (clear data) в слой DDS. После чего «очищенные» данные из слоя DDS извлекаются и трансформируются в метрики, и загружаются в слой DM. Параметры подключения к Airflow представлены в Таблице 2. Переменные Airflow представлены в Таблице 3.

Таблица 2 - Параметры подключения к Airflow

|  |  |
| --- | --- |
| **Параметр** | **Значение параметра** |
| URL веб-интерфейса Airflow | http://localhost:8080/ |
| Логин | airflow |
| Путь к файлам DAG | /hr-project/dags |
| Путь к исполняемым скриптам | /hr-project/dags/layers  /hr-project/dags/scripts  /hr-project/dags/sql\_scripts |

Таблица 3 -Переменные Airflow

|  |  |
| --- | --- |
| **Переменная** | **Значение переменной** |
| source\_conn\_id | source\_postgres\_stage |
| target\_conn\_id | target\_postgres\_ods |
| source\_schema\_ods | source\_data |
| targer\_schema\_ods | ods |
| dds\_to\_dm | {"source\_conn\_id": "etl\_db\_8",  "target\_conn\_id":"etl\_db\_8",  "source\_schema": "dds",  "target\_schema": "dm"} |

* 1. **Описание ETL-потока**

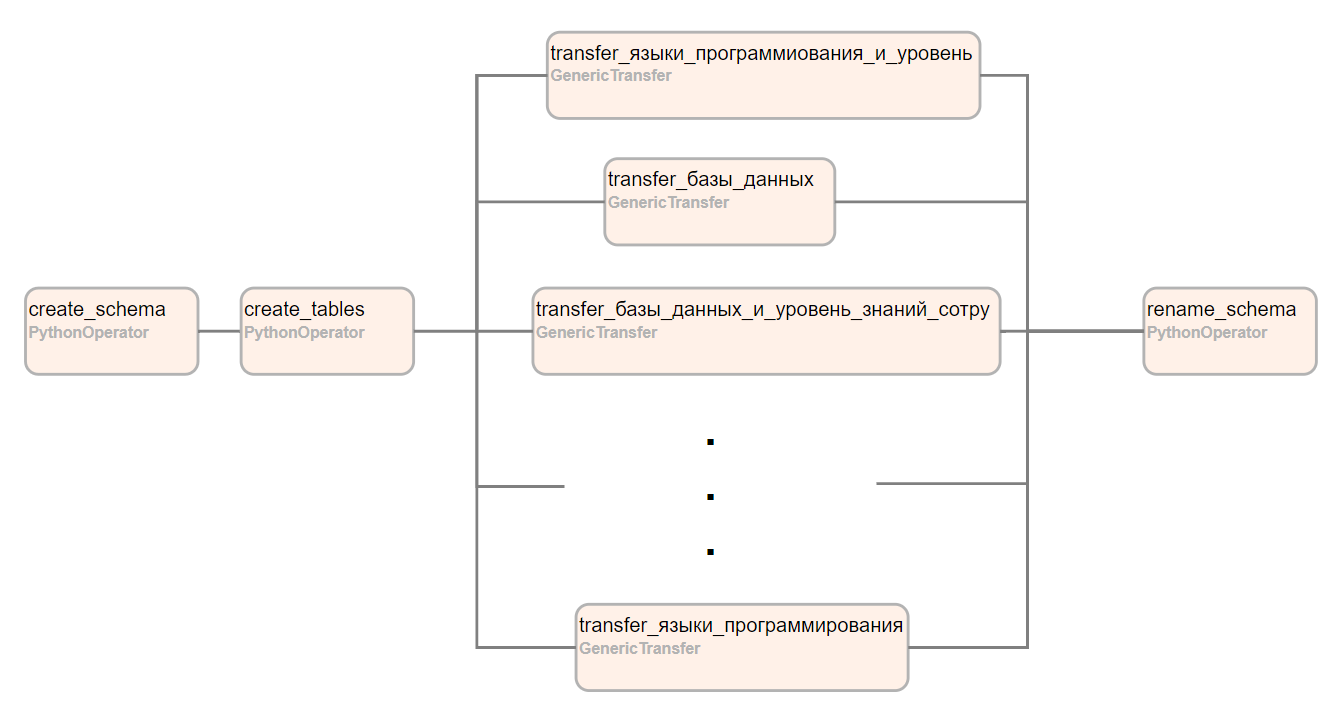


Рисунок 2 – Схема для DAG dag\_ods\_layer

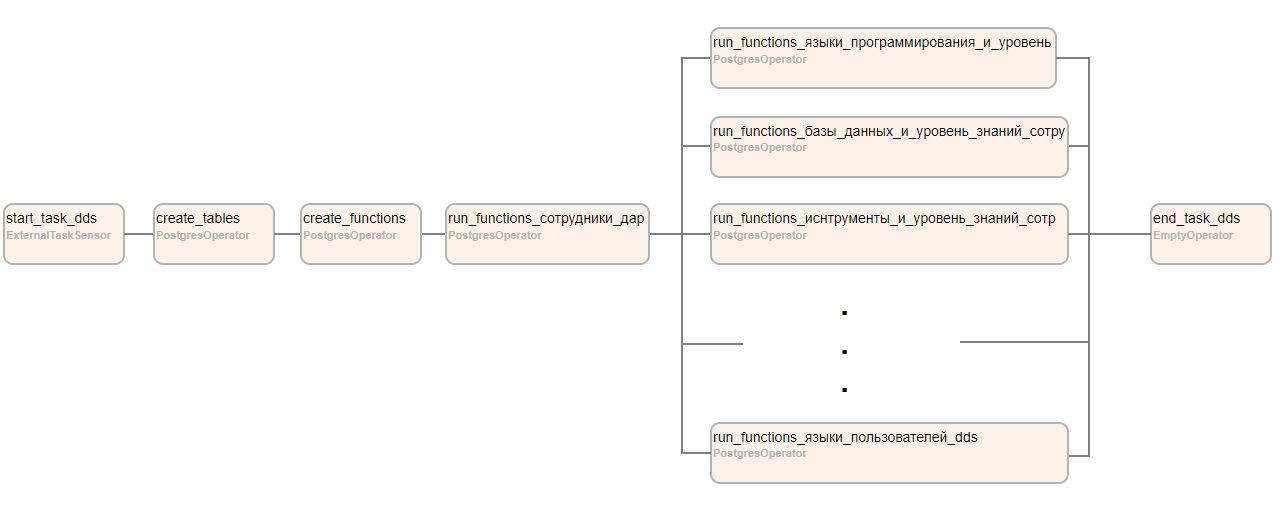


Рисунок 3 – Схема для DAG dag\_dds\_layer

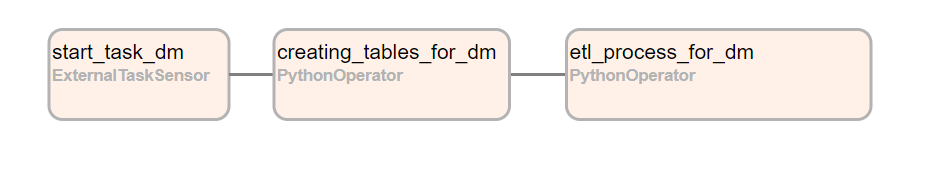


Рисунок 4 – Схема для DAG dag\_dm\_layer

Реализация ETL-потока осуществлена следующими DAG’ми Airflow, схемы которых представлены на рисунках 2 - 4:

- **dag\_ods\_layer.py –** DAG для Airflow, который осуществляет загрузку данных в слой ods из базы данных source (Source layer). DAG определяет задачи с помощью PythonOperator и GenericTransfer: create\_schema, create\_tables, transfer\_{название таблицы} и rename\_schema, которые выполняются последовательно. dag\_ods\_layer запускается ежедневно в 21:00 UTC.

- **dag\_dds\_layer.py –** DAG для Airflow, который осуществляет ETL-процесс из слоя ods в dds. DAG определяет задачи с помощью PostgresOperator, EmptyOperator и ExternalTaskSensor, который связывает DAGи dag\_ods\_layer и dag\_dds\_layer: start\_task\_dds, create\_tables, create\_functions, run\_functions\_сотрудники\_дар, run\_functions\_{название таблицы} и end\_task\_dds. dag\_dds\_layer запускается ежедневно в 21:00 UTC.

- **dag\_dm\_layer.py -** DAG для Airflow, который осуществляет ETL-процесс из слоя dds в dm. dags.py использует функцию из dag\_manager.py для получения данных из переменной Airflow, а также определяет задачи с помощью ExternalTaskSensor и PythonOperator: start\_task\_dm, creating\_tables\_for\_dm и etl\_process\_for\_dm, которые выполняются последовательно. DAG запускается ежедневно в 21:00 UTC.

Таким образом благодаря ExternalTaskSensor DAGи связаны в единую цепочку и выполняются друг за другом.

* 1. **Storage layer (слой хранения)**

Базы данных PostgreSQL source и etl\_db\_8 обеспечивают хранение данных. «Сырые» данные хранятся в базе данных source. «Сырые» и «очищенные» данных после трансформации хранятся в базе данных etl\_db\_8. Структура хранилищ данных представлена в Таблице 4.

Таблица 4 - Структура хранилищ данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Слой** | **Схема** | **База данных** | **Параметры подключения** | **Имя учетной записи** |
| Source layer | source\_data | source | хост: 10.82.0.4  порт: 5432 | etl\_user\_8 |
| ODS layer | ods | etl\_db\_8 |
| DDS layer | dds |
| DM layer | dm |

* 1. **Presentation layer (слой представления)**

Итоговая отчетность, представленная на рисунке 5, реализуется с помощью технологии Luxms BI и библиотеки React с дашбордами, построенными на основе данных из слоя DM.

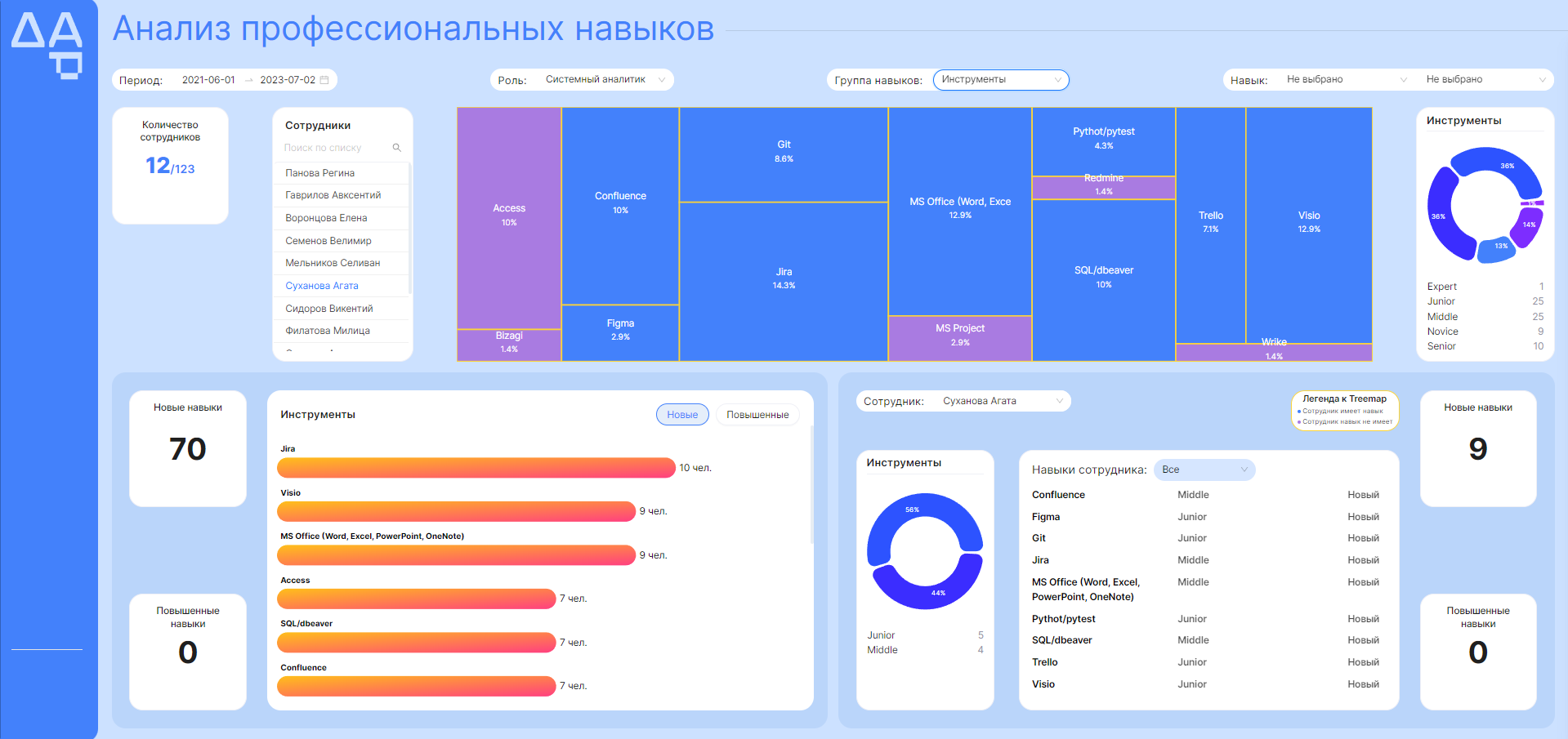


Рисунок 5 – Итоговая отчетность

1. **Описание используемых сущностей**

Описание используемых сущностей и соответствующих им схем представлено в Таблице 5.

Таблица 5 – Используемые сущности

|  |  |
| --- | --- |
| **Схема** | **Сущности** |
| source\_data | «базы\_данных», «базы\_данных\_и\_уровень\_знаний\_сотру», «инструменты», «инструменты\_и\_уровень\_знаний\_сотр», «образование\_пользователей», «опыт\_сотрудника\_в\_отраслях», «опыт\_сотрудника\_в\_предметных\_обла», «отрасли», «платформы», «платформы\_и\_уровень\_знаний\_сотруд», «предметная\_область», «резюмедар», «сертификаты\_пользователей», «сотрудники\_дар», «среды\_разработки», «среды\_разработки\_и\_уровень\_знаний\_», «технологии», «технологии\_и\_уровень\_знаний\_сотру», «типы\_систем», «типы\_систем\_и\_уровень\_знаний\_сотру», «уровень\_образования», «уровни\_владения\_ин», «уровни\_знаний», «уровни\_знаний\_в\_отрасли», «уровни\_знаний\_в\_предметной\_област», «фреймворки», «фреймворки\_и\_уровень\_знаний\_сотру», «языки», «языки\_пользователей», «языки\_программирования», «языки\_программирования\_и\_уровень» |
| ods | «базы\_данных», «базы\_данных\_и\_уровень\_знаний\_сотру», «инструменты», «инструменты\_и\_уровень\_знаний\_сотр», «образование\_пользователей», «опыт\_сотрудника\_в\_отраслях», «опыт\_сотрудника\_в\_предметных\_обла», «отрасли», «платформы», «платформы\_и\_уровень\_знаний\_сотруд», «предметная\_область», «резюмедар», «сертификаты\_пользователей», «сотрудники\_дар», «среды\_разработки», «среды\_разработки\_и\_уровень\_знаний\_», «технологии», «технологии\_и\_уровень\_знаний\_сотру», «типы\_систем», «типы\_систем\_и\_уровень\_знаний\_сотру», «уровень\_образования», «уровни\_владения\_ин», «уровни\_знаний», «уровни\_знаний\_в\_отрасли», «уровни\_знаний\_в\_предметной\_област», «фреймворки», «фреймворки\_и\_уровень\_знаний\_сотру», «языки», «языки\_пользователей», «языки\_программирования», «языки\_программирования\_и\_уровень» |

Продолжение таблицы 5

|  |  |
| --- | --- |
| **Схема** | **Сущности** |
| dds | «базы\_данных», «базы\_данных\_и\_уровень\_знаний\_сотру», «инструменты», «инструменты\_и\_уровень\_знаний\_сотр», «образование\_пользователей», «опыт\_сотрудника\_в\_отраслях», «опыт\_сотрудника\_в\_предметных\_обла», «отрасли», «платформы», «платформы\_и\_уровень\_знаний\_сотруд», «предметная\_область», «сертификаты\_пользователей», «сотрудники\_дар», «среды\_разработки», «среды\_разработки\_и\_уровень\_знаний\_», «технологии», «технологии\_и\_уровень\_знаний\_сотру», «типы\_систем», «типы\_систем\_и\_уровень\_знаний\_сотру», «уровень\_образования», «уровни\_владения\_ин», «уровни\_знаний», «уровни\_знаний\_в\_отрасли», «уровни\_знаний\_в\_предметной\_област», «фреймворки», «фреймворки\_и\_уровень\_знаний\_сотру», «языки», «языки\_пользователей», «языки\_программирования», «языки\_программирования\_и\_уровень» |
| dds\_er | «базы\_данных», «базы\_данных\_и\_уровень\_знаний\_сотру», «инструменты», «инструменты\_и\_уровень\_знаний\_сотр», «образование\_пользователей», «опыт\_сотрудника\_в\_отраслях», «опыт\_сотрудника\_в\_предметных\_обла», «отрасли», «платформы», «платформы\_и\_уровень\_знаний\_сотруд», «предметная\_область», «сертификаты\_пользователей», «среды\_разработки», «среды\_разработки\_и\_уровень\_знаний\_», «технологии», «технологии\_и\_уровень\_знаний\_сотру», «типы\_систем», «типы\_систем\_и\_уровень\_знаний\_сотру», «уровень\_образования», «уровни\_владения\_ин», «уровни\_знаний», «уровни\_знаний\_в\_отрасли», «уровни\_знаний\_в\_предметной\_област», «фреймворки», «фреймворки\_и\_уровень\_знаний\_сотру», «языки», «языки\_пользователей», «языки\_программирования», «языки\_программирования\_и\_уровень» |
| dm | «группы\_навыков\_и уровень знаний\_со», «сотрудники\_дар», «навыки», «группы\_навыков», «уровни\_знаний» |