Цель проекта

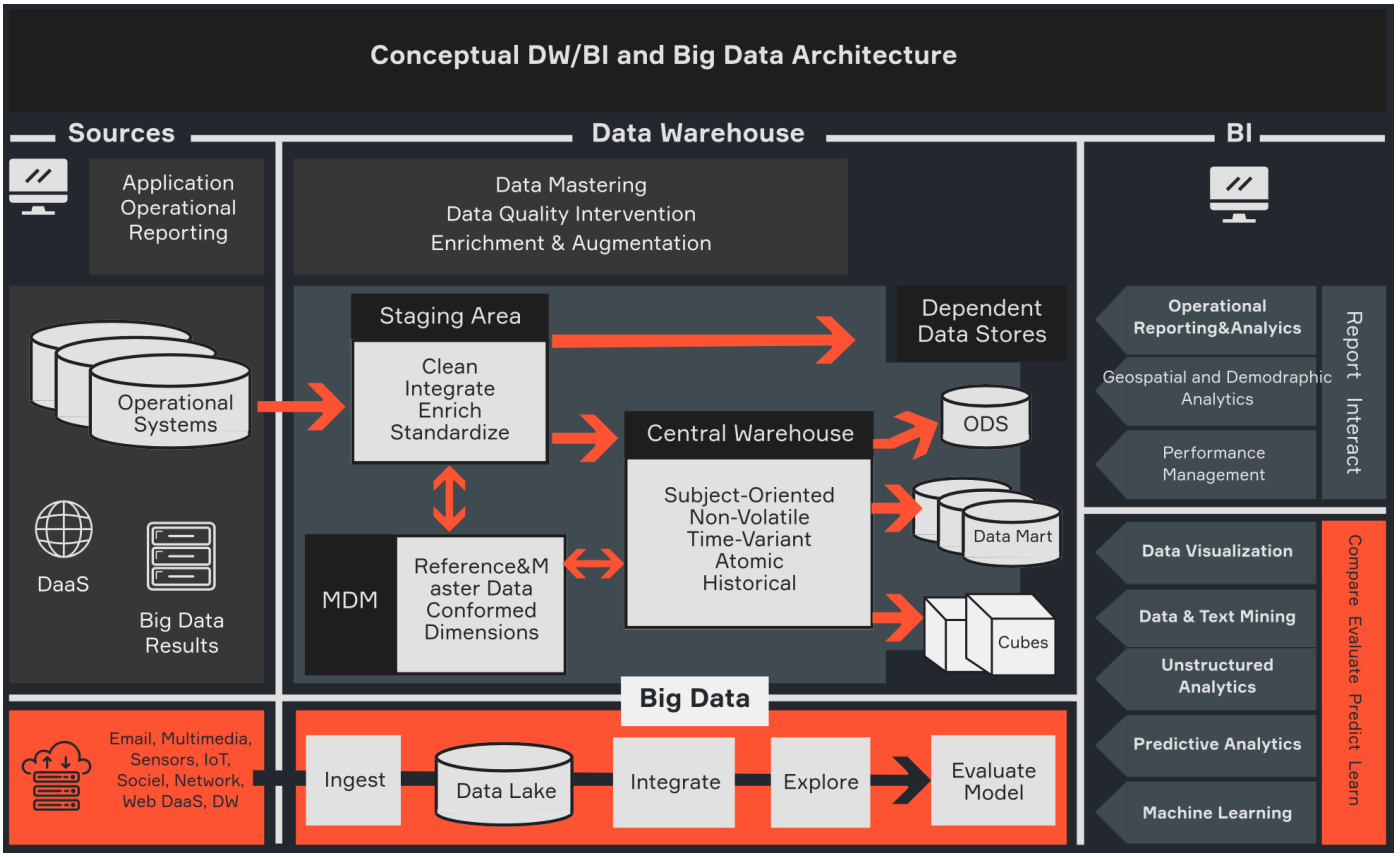
Получить представление о совместной работе DWH, Data Lake и Data Orchestrator на примере Greenplum \ Spark \ Airflow, приближенное к реальной работе Data Engineer

Состав проекта:

- пять запросов на Spark, объединенных в один DAG Airflow
- пять запросов на Greenplum, объединенных в один DAG Airflow

Образовательные результаты проекта:

- использование изученных инструментов в связке: Airflow, Spark, Greenplum
- понимание, как строятся пайплайны обработки данных в озере данных и DWH
- построение отчётов DE по ТЗ от заказчика



В качестве логической архитектуры в рамках проекта мы будем эмулировать классическую, представленную в DAMA DMBOK, разделенную на две части

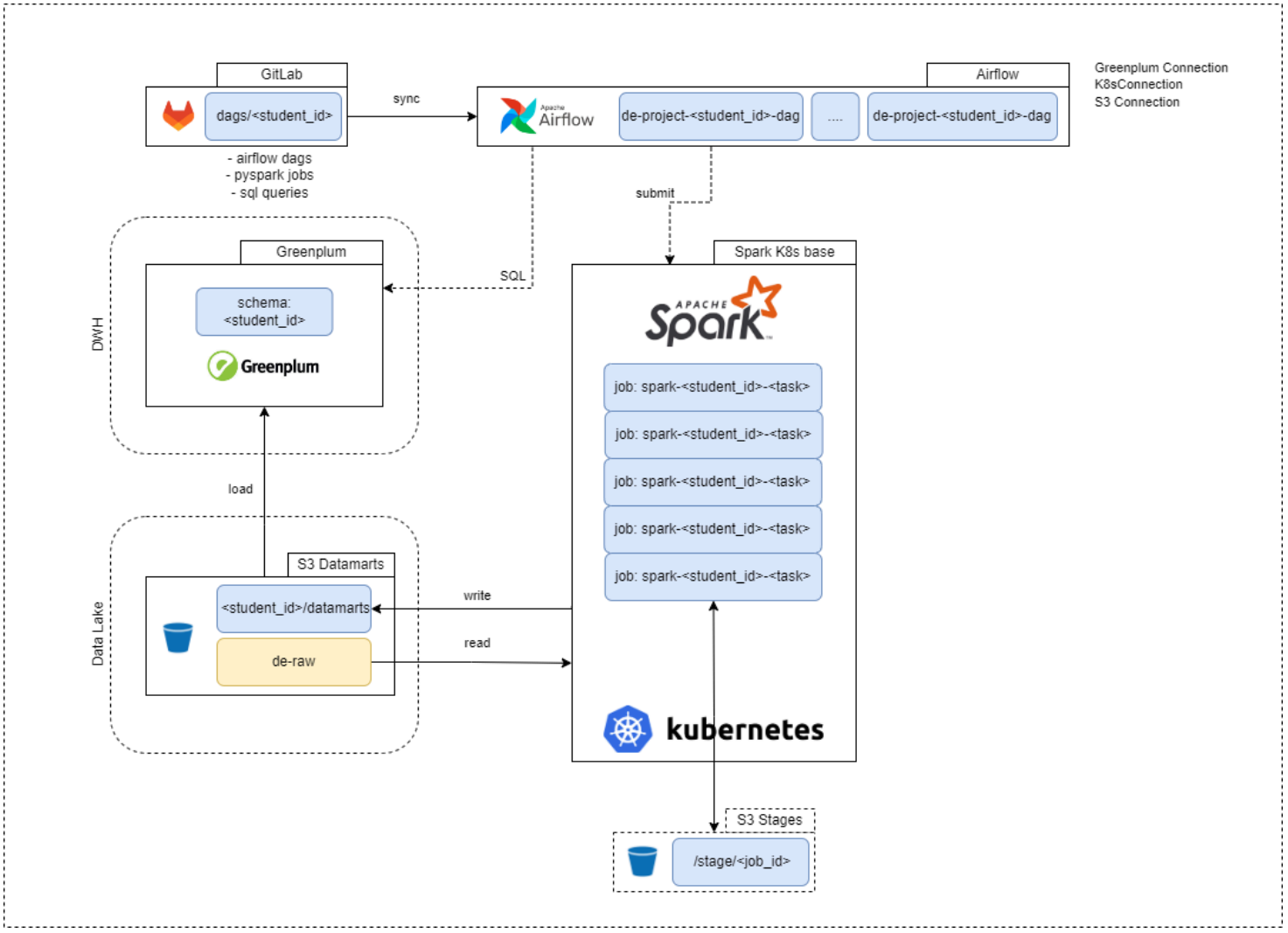
Аналитическое хранилище (англ. analytical data warehouse) и озеро данных (англ. data lake) - это две различные архитектуры данных и методы их хранения и использования.

Аналитическое хранилище - это база данных, специально оптимизированная для анализа больших объемов структурированных данных, которые были предварительно обработаны и преобразованы в определенный формат. В аналитических хранилищах данные обычно организованы в соответствии с определенной схемой (например, звезда или снежинка), и они могут охватывать даты до нескольких лет и содержать миллионы строк. Аналитическое хранилище может использоваться для проведения анализа данных, создания отчетов, прогнозирования и т.д.

Озеро данных, с другой стороны, является хранилищем данных, которое позволяет хранить структурированные и неструктурированные данные в их естественном формате без необходимости их предварительной обработки. Озеро данных может включать данные из различных источников и форматов, таких как социальные сети, системы почты и многие другие. Целью озера данных является сохранение данных в максимально широком и гибком формате для дальнейшего анализа, обработки или использования в различных приложениях.

Таким образом, основным отличием между аналитическим хранилищем и озером данных является формат и способ хранения данных. Аналитическое хранилище предназначено для хранения структурированных данных, которые были предварительно обработаны и преобразованы в определенный формат, тогда как озеро данных используется для хранения большого объема данных в естественном формате.

Техническая инфраструктура



В качестве технической инфраструктуры будут использованы следующие инструменты.

① Озеро данных на базе S3
② Аналитическое хранилище на базе Greenplum
③ Оркестратор потоков данных - Airflow
④ Кластер распределенных вычислений Spark (Kubernetes)
⑤ GitLab для хранения исходного кода

Интеграция между GP и Spark будет проходить через внешние таблицы с хранением в S3.

Репозиторий кода

Вам доступен репозиторий с кодом, куда вы должны будете разместить свое решение: https://git.lab.karpov.courses/de/de-project

Библиотеки

Все необходимые вам зависимости и не конфликтующие версии указаны в requirements.txt нашего репозитория. Настройте вашу среду разработки. При необходимости для установки выполните команду:

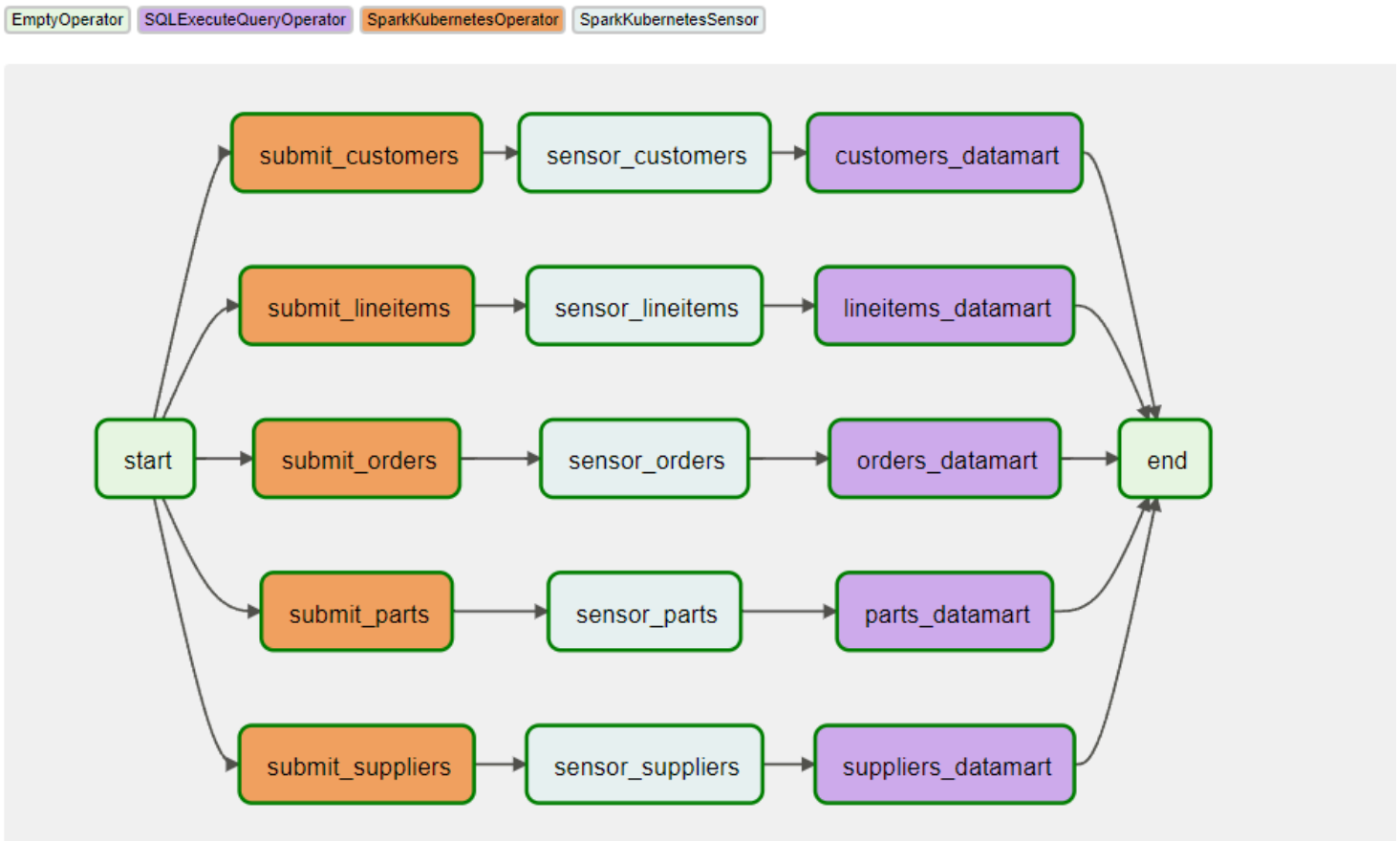
```
pip install requirements.txt
```

Итоговый вид DAG

Ниже представлен итоговый вид DAG'a который у нас должен получится.

Вам понадобятся следующие операторы и сенсоры:

1. airflow.operators.empty.EmptyOperator - для объединения задач
2. airflow.providers.cncf.kubernetes.operators.spark_kubernetes.SparkKubernetesOperator - для отправки Spark задач на кластер
3. airflow.providers.cncf.kubernetes.sensors.spark_kubernetes.SparkKubernetesSensor - для отслеживания статуса Spark задач
4. airflow.providers.common.sql.operators.sql.SQLExecuteQueryOperator - для выполнения запросов к Greenplum



Внимание

Группа операторов из пакета airflow.providers.cncf.kubernetes.sensors.spark_kubernetes.* являются специфичными и предназначены упростить запуск Spark задач в кластерном режиме на базе Kubernetes.
Помимо самого оператора вам может пригодится документация по созданию конфигураций запуска Spark задач. А так же пример, размещённый в GitLab:
https://git.lab.karpov.courses/de/de-project/-/tree/master/dags/a.savchenko

Создание инфраструктуры

Для создания инфраструктуры запустите шаг, в результате него будут созданы:

① Yandex Cloud S3 бакет и доступы к нему.
② Greenplume схема и пользователь к ней.

Инструменты проекта

На данной странице собраны ссылки на инструменты проекта для быстрого доступа.

GitLab

Создайте в репозитории папку с названием в виде вашего логина - <student_id> где будут содержаться ваши
DAG, Spark Jobs, SQL запросы. Используйте пример из GitLab.

Airflow

!!! Убедитесь что созданный вами DAG имеет dag_id следующего вида: **de-project-<student_id>-dag** !!!

В Airflow уже созданы нужные вам connections, пожалуйста используйте их:

greenplume_karpov - сконфигурированное подключение к Greenplum

kubernetes_karpov - сконфигурированное подключение к Kubernetes Cluster

S3 bucket: **de-raw** (бакет с содержанием исходных данных)

Данный бакет доступен вам только на чтение.

S3 bucket: <**student_id**> (бакет для ваших данных)

Ключи для доступа к бакетам выдаются на последнем шаге раздела "Техническая инфраструктура"

Greenplum

Ваш логин/пароль и схема создаются и выдаются на последнем шаге раздела "Техническая инфраструктура"

host: greenplum.karpov.courses
port: 6432

Описание данных

В проекте мы с вами будем использовать данные предоставляемые из Теста TPC-H

Исходные данные с которыми мы будем работать размещены в S3 бакете - de-project (Yandex.Cloud)

TPC-H - это стандартный бенчмарк для тестирования производительности систем управления реляционными базами данных (RDBMS). Он состоит из набора запросов, которые используются для оценки производительности обработки запросов на чтение больших объемов данных из базы данных.

TPC-H был разработан Transaction Processing Performance Council (TPC) - некоммерческой организацией, которая занимается разработкой стандартов и методов для тестирования производительности систем обработки транзакций. Бенчмарк состоит из 22 запросов, которые покрывают широкий диапазон типов запросов, таких как агрегирование, слияние таблиц и фильтрация данных.

TPC-H используется для тестирования производительности RDBMS на больших объемах данных, таких как решения хранилища данных и системы аналитики. Он помогает выявлять узкие места в производительности системы и оптимизировать её работу при работе с большими объёмами данных. Результаты тестирования TPC-H используются для сравнения производительности разных систем RDBMS.

Набор таблиц, используемых в тесте, включает в себя следующие таблицы:

Таблица	Описание
customer	информация о заказчиках
lineitem	информация о позициях в заказах
nation	информация о нациях/странах
orders	информация о заказах
parts	информация о деталях/запчастях
partsupp	информация о поставках деталей/запчастей
region	информация о регионах
supplier	информация о поставщиках

Каждая из этих таблиц содержит различные поля, описывающие соответствующие сущности. Таблицы взаимосвязаны и связаны между собой по ключевым полям, таким образом, что можно выполнять сложные запросы, которые связывают данные из нескольких таблиц.

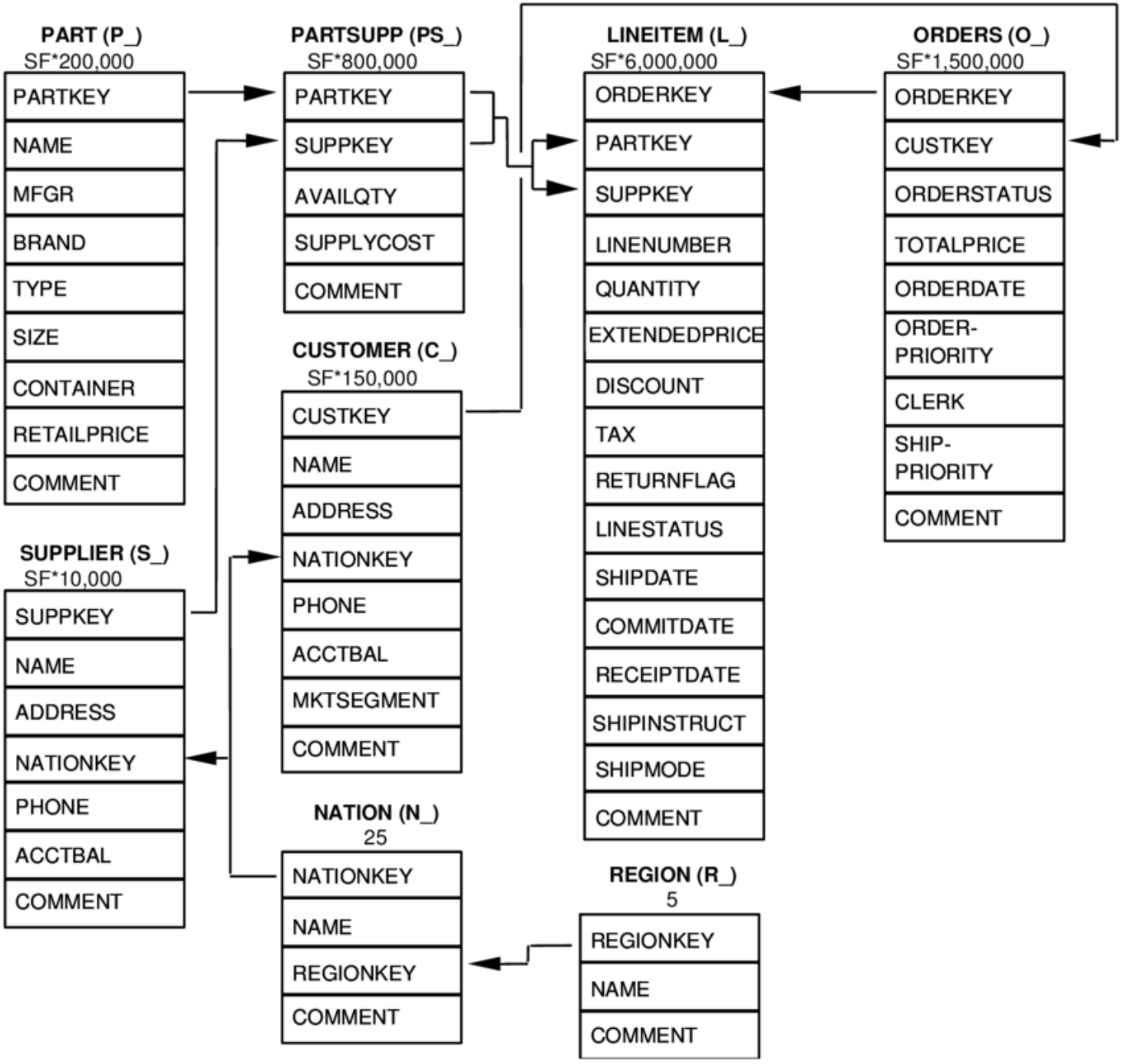


Таблица **region** в тесте TPC-H содержит всего два поля:

Поле	Тип	Описание
R_REGIONKEY	Integer	уникальный идентификатор региона, целочисленный тип данных
R_NAME	String	название региона, текстовый тип данных
R_COMMENT	String	коментарий

Эта таблица используется в тесте для хранения информации о регионах, к которым могут быть привязаны поставщики и потребители товаров, заказчики и т.д. Ключевое поле r_regionkey может использоваться для связи информации об этом регионе с другой таблицей, например, с таблицей supplier, где указывается регион, в котором зарегистрирован поставщик. Поле r_name просто содержит название региона в текстовом формате.
Таблица **nation** в тесте TPC-H содержит следующие поля:

Поле	Тип	Описание

N_NATIONKEY	Integer	уникальный идентификатор нации, целочисленный тип данных
N_NAME	String	название нации, текстовый тип данных
N_REGIONKEY	Integer	идентификатор региона, к которому относится данная нация, целочисленный тип данных
N_COMMENT	String	дополнительная информация о нации, текстовый тип данных

Эта таблица используется в тесте для хранения информации о нациях или странах, которые принадлежат определенным регионам. Ключевое поле n_nationkey может использоваться для связи информации об этой нации с другой таблицей, например, с таблицей customer, где указывается нация, к которой относится заказчик. Поле n_regionkey связывает эту нацию с регионом, к которому она принадлежит. Поле n_name просто содержит название нации в текстовом формате, а поле n_comment может содержать любую дополнительную информацию о нации.

В таблице **customer** теста TPC-H содержатся следующие поля:

Поле	**Тип**	**Описание**
C_CUSTKEY	Integer	Уникальный идентификатор клиента
C_NAME	String	Имя клиента
C_ADDRESS	String	Адрес клиента
C_NATIONKEY	Integer	Уникальный идентификатор нации, к которой принадлежит клиент
C_PHONE	String	Телефон клиента
C_ACCTBAL	Float	Остаток на счете клиента
C_MKTSEGMENT	String	Рыночный сегмент клиента
C_COMMENT	String	Комментарий

Таблица содержит информацию о клиентах, которые используют услуги компании, участвующей в тесте TPC-H.

В таблице **supplier** теста TPC-H содержатся следующие поля:

Поле	**Тип**	**Описание**
S_SUPPKEY	Integer	Уникальный идентификатор поставщика
S_NAME	String	Название поставщика
S_ADDRESS	String	Адрес поставщика
S_NATIONKEY	Integer	Уникальный идентификатор нации, к которой принадлежит поставщик
S_PHONE	String	Телефон поставщика
S_ACCTBAL	Float	Остаток на счете поставщика

S_COMMENT	String	Рыночный сегмент клиента

Таблица содержит информацию о поставщиках, которые предоставляют товары и услуги компании,
участвующей в тесте TPC-H.

В таблице **orders** теста TPC-H содержатся следующие поля:

Поле	Тип	Описание
O_ORDERKEY	Integer	Уникальный идентификатор заказа
O_CUSTKEY	Integer	Уникальный идентификатор клиента, который разместил заказ
O_ORDERSTATUS	String	Статус заказа
O_TOTALPRICE	Float	Общая стоимость заказа
O_ORDERDATE	String	Дата размещения заказа (YYYY-MM-DD)
O_ORDERPRIORITY	String	Приоритет заказа
O_CLERK	String	Имя сотрудника, который обработал заказ
O_SHIPPRIORITY	Integer	Приоритет доставки заказа
O_COMMENT	String	Комментарий

Таблица содержит информацию о заказах, которые размещают клиенты в компании, участвующей в тесте TPC-H.

В таблице **lineitem** теста TPC-H содержатся следующие поля:

Поле	Тип	Описание
L_ORDERKEY	Integer	Уникальный идентификатор заказа, к которому относится строка заказа
L_PARTKEY	Integer	Уникальный идентификатор товара, который был заказан в данной строке заказа
L_SUPPKEY	Integer	Уникальный идентификатор поставщика товара, который был заказан в данной строке заказа
L_LINENUMBER	Integer	Уникальный номер строки/позиции в заказе
L_QUANTITY	Integer	Количество заказанного товара
L_EXTENDEDPRICE	Float	Расширенная цена (QUANTITY * цена единицы товара)
L_DISCOUNT	Float	Скидка, применяемая к данной строке заказа
L_TAX	Float	Налог, применяемый к данной строке заказа
L_RETURNFLAG	String	Флаг возврата товара
L_LINESTATUS	String	Статус строки заказа
L_SHIPDATE	String	Дата отгрузки заказа

L_COMMITDATE	String	Дата подтверждения заказа
L_RECEIPTDATE	String	Дата получения заказа
L_SHIPINSTRUCT	String	Инструкции для отгрузки
L_SHIPMODE	String	Способ доставки
L_COMMENT	String	Комментарий

Таблица содержит информацию о каждой строке заказа, которая содержит информацию о заказанных товарах и их характеристиках.

В таблице **part** содержатся следующие поля:

Поле	**Тип**	**Описание**
P_PARTKEY	Integer	Уникальный идентификатор товара
P_NAME	String	Описание товара
P_MFGR	String	Название производителя товара
P_BRAND	String	Бренд товара
P_TYPE	String	Тип товара
P_SIZE	Integer	Количество товара на складе
P_CONTAINER	String	Тип контейнера товара
P_RETAILPRICE	Float	Розничная цена товара
P_COMMENT	String	Комментарий

Таблица содержит информацию о товарах, которые размещены на складах в компании, участвующей в тесте TPC-H.

В таблице **partsupp** теста TPC-H содержатся следующие поля:

Поле	**Тип**	**Описание**
PS_PARTKEY	Integer	Уникальный идентификатор товара
PS_SUPPKEY	Integer	Уникальный идентификатор поставщика товара
PS_AVAILQTY	Integer	Доступное количество товара на складе
PS_SUPPLYCOST	Float	Стоимость поставки товара от поставщика
PS_COMMENT	String	Комментарий

Таблица **partsupp** содержит информацию о поставках товаров от поставщиков, которые продаются в компании, участвующей в тесте TPC-H. Каждая запись в таблице связывает товар из таблицы **part** с поставщиком из таблицы **supplier** и указывает доступное количество товара на складе поставщика и его стоимость поставки.

Постановка задачи

Необходимо создать pipeline обработки данных (DAG Airflow), который реализует все запросы из последующих шагов.

Git

Создайте в репозитории папку с названием в виде вашего логина - <STUDENT_ID> где будут содержаться ваши DAG, Spark Jobs, SQL запросы. Используйте пример из GitLab.

Требования к Airflow DAG

1. DAG не имеет расписания (schedule_interval=None)
2. каждая Spark задача выполняется в своей таске (отдельный submit)
3. все таски выполняются параллельно
4. таски называются идентично отчетам

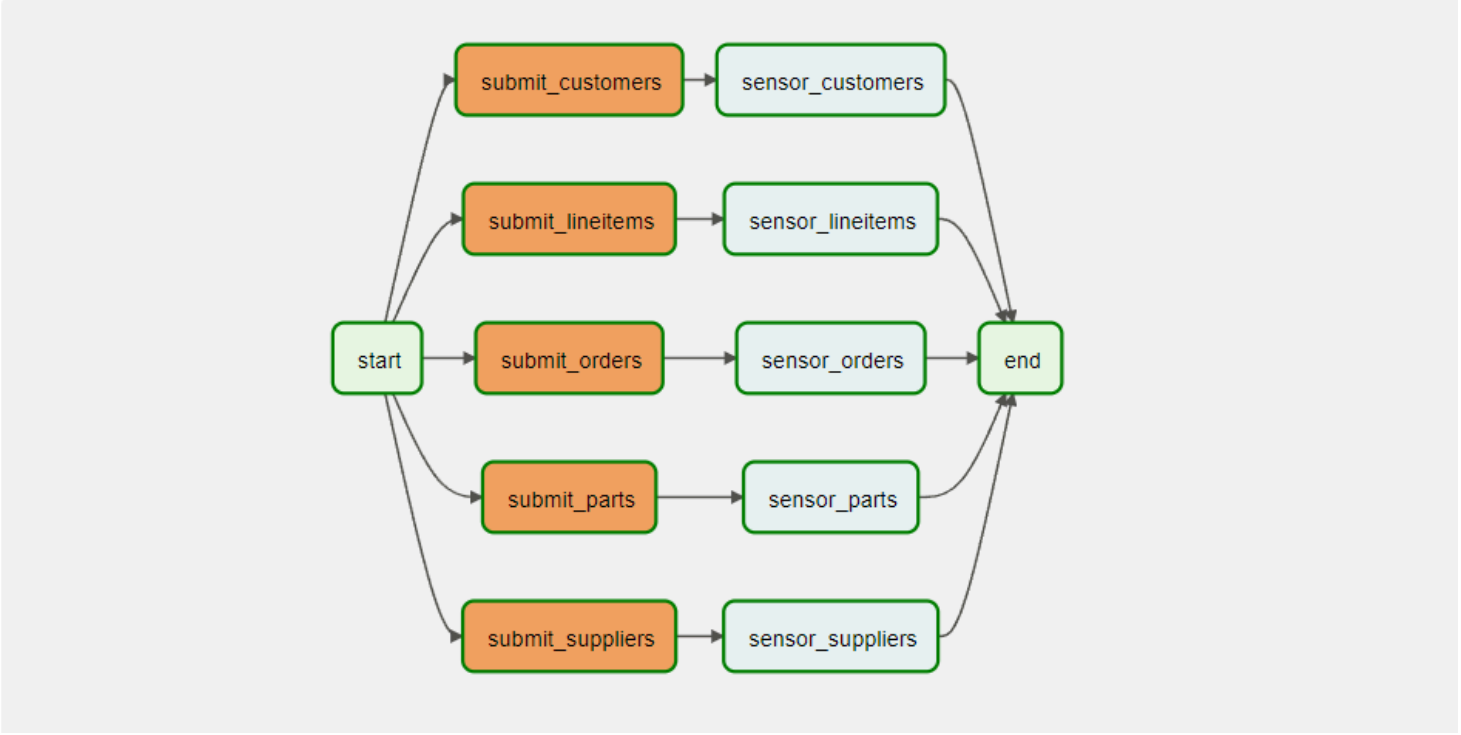
!!! Убедитесь что созданный вами DAG имеет dag_id следующего вида: **de-project-<STUDENT_ID>-dag** !!!

В Airflow уже созданы нужные вам connections, пожалуйста используйте их:

kubernetes_karpov - сконфигурированное подключение к Kubernetes Cluster

По завершению данного шага ваш DAG должен выглядеть примерно так:

EmptyOperator SparkKubernetesOperator SparkKubernetesSensor



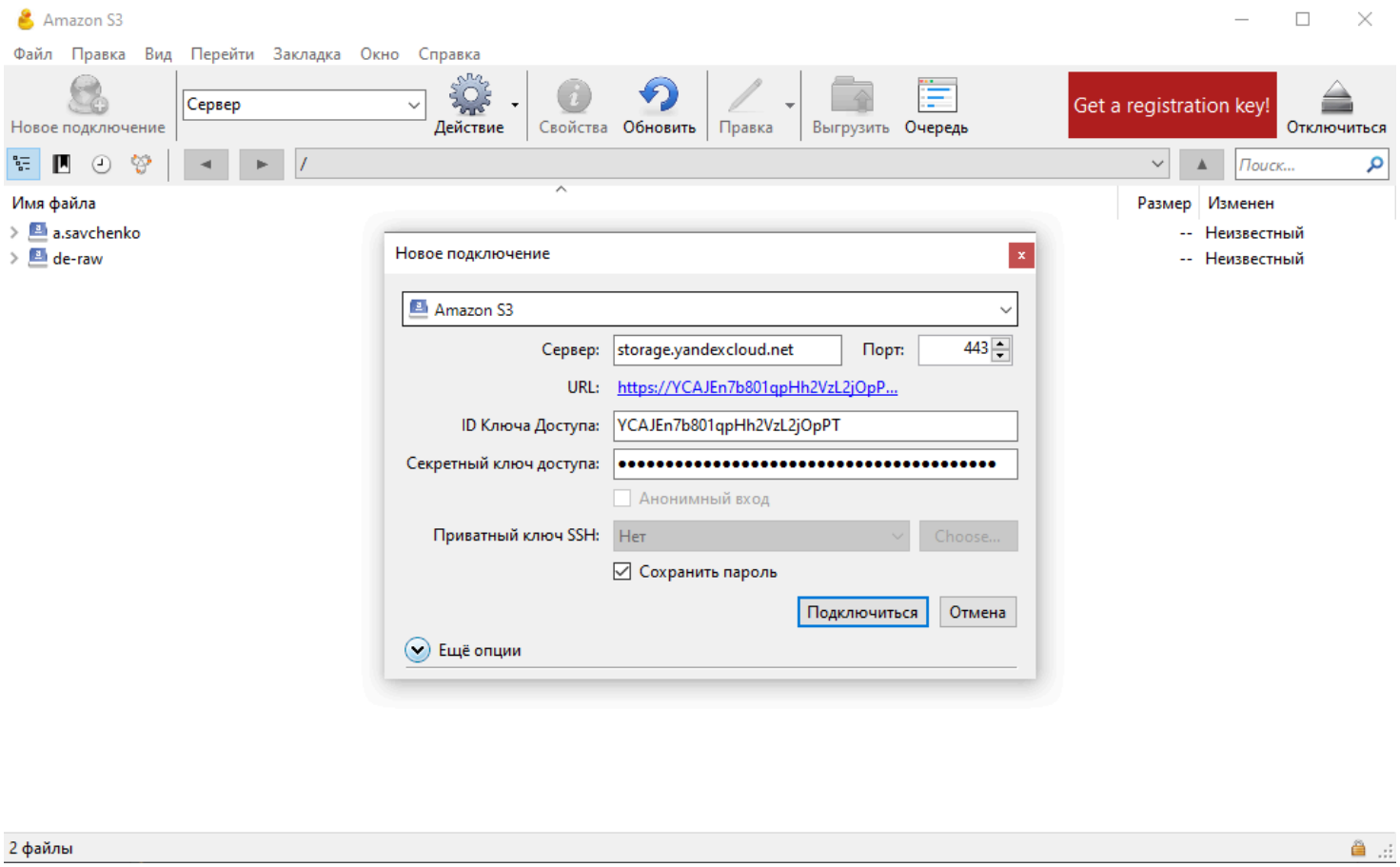
Вам понадобятся следующие операторы и сенсоры:

1. airflow.operators.empty.EmptyOperator - для объединения задач
2. airflow.providers.cncf.kubernetes.operators.spark_kubernetes.SparkKubernetesOperator - для отправки Spark задач на кластер
3. airflow.providers.cncf.kubernetes.sensors.spark_kubernetes.SparkKubernetesSensor - для отслеживания статуса Spark задач

Доступ к S3

Для доступа к S3 хранилищу и просмотру его содержимого используйте CyberDuck и ключи которые вы получили на шаге создания инфраструктуры под вас.

Настройка подключения



Используя CyberDuck вы можете скачать исходные данные для отладки.

Конфигурация SparkKubernetesOperator

Для запуска вашей Spark задачи, оператору требуется указать пути до самой py-spark-job, а так же до конфигурационного файла задачи, где описаны параметры запуска. Для нашего проекта используйте следующий пример для отчета **orders**.
Некоторые параметры вам нужно заменить на свои, для минимизации проблем в общем окружении избегайте создания задач с одним и тем же metadata.name.

User guide https://github.com/GoogleCloudPlatform/spark-on-k8s-operator/blob/master/docs/user-guide.md apiVersion: sparkoperator.k8s.io/v1beta2 kind: SparkApplication metadata: name: spark-job-a-savchenko-orders #!!!Тут имя для вашей Spark Job!!! spec: type: Python pythonVersion: "3" mode: cluster image: itayb/spark:3.1.1-hadoop-3.2.0-aws # Образ на котором базируются наши Spark Driver/Executers imagePullPolicy: Always

mainApplicationFile: "local:///de-project/dags/a.savchenko/orders-spark-job.py" #!!!Тут путь к вашему файлу Spark Job!!! hadoopConf: # Дополнительные параметры для доступа в S3 "fs.s3a.endpoint": "HTTPS://STORAGE.YANDEXCLOUD.NET" "fs.s3a.region": "ru-central1" "fs.s3a.access.key": "YCAJEn7b801qpHh2VzL2jOpPT" "fs.s3a.secret.key": "YCMi7T9HQljkaaFOcZIeVkGEWjtf8P39jzH1w9K6" sparkVersion: "3.1.1" timeToLiveSeconds: 30 restartPolicy: type: Never volumes: - name: git-repo emptyDir: sizeLimit: 1Gi - name: ssh-key secret: secretName: ssh-key defaultMode: 256 driver: # Конфигурация Spark Driver javaOptions: "-Divy.cache.dir=/tmp -Divy.home=/tmp" volumeMounts: - name: "git-repo" mountPath: /de-project - name: ssh-key mountPath: /tmp/ssh initContainers: # Инициализация контейнера - клонирование проекта с вашим исходным кодом - name: git-clone image: alpine/git:2.40.1 env: - name: GIT_SSH_COMMAND value: "ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no" command: ['sh', '-c', 'git clone --depth=1 --single-branch git@git.lab.karpov.courses:de/de-project.git /de-project'] volumeMounts: - name: git-repo mountPath: /de-project - name: ssh-key mountPath: /root/.ssh cores: 1 coreLimit: "1200m" memory: "1024m" labels: version: 3.1.1 serviceAccount: spark-driver executor: # Конфигурация Spark Executor image: itayb/spark:3.1.1-hadoop-3.2.0-aws cores: 1 instances: 2 memory: "1024m" labels: version: 3.1.1

Подробнее про конфигурирование SparkKubernetesOperator.

Обработка данных для отчета по LineItems

Описание задачи

Необходимо построить отчет по данным о позициях в заказе (lineitems) содержащий сводную информацию по позициям каждого заказа, когда либо совершенного в системе, группировать данные необходимо по идентификатору заказа.
Используйте сортировку по ключу заказа на возрастание.

Итоговый формат

В отчете должны присутствовать следующие колонки:

Поле	Тип	Описание
L_ORDERKEY	bigint	уникальный идентификатор заказа
count	int	число позиций/вещей в заказе
sum_extendprice	float	сумма заказа (сумма всех позиций по L_EXTENDEDPRICE)
mean_discount	float	медиана скидок по позициям (L_DISCOUNT)
mean_tax	float	средний налог в заказе между позициями
delivery_days	float	среднее время доставки всех позиций заказа (в днях) с момента заказа (L_RECEIPTDATE) до момента доставки (L_SHIPDATE)
A_return_flags	int	количество заказов с флагом L_RETURNFLAG == 'A'
R_return_flags	int	количество заказов с флагом L_RETURNFLAG == 'R'
N_return_flags	int	количество заказов с флагом L_RETURNFLAG == 'N'

Пример результата

	L_ORDERKEY	count	sum_extendprice	mean_discount	mean_tax	delivery_days	A_return_flags	R_return_flags	N_return_flags
0	1	6	195298.34	0.081667	0.036667	8.333333	0	0	6
1	2	1	63066.32	0.000000	0.050000	5.000000	0	0	1
2	3	6	287551.64	0.061667	0.025000	15.833333	3	3	0
3	4	1	39819.00	0.030000	0.080000	8.000000	0	0	1
4	5	3	125431.30	0.056667	0.050000	13.666667	1	2	0
5	6	1	53697.73	0.080000	0.030000	5.000000	1	0	0
6	7	7	298268.24	0.065714	0.040000	15.714286	0	0	7
7	32	6	130065.56	0.056667	0.036667	13.833333	0	0	6
8	33	4	132757.38	0.062500	0.030000	11.250000	3	1	0
9	34	3	54144.49	0.033333	0.063333	7.666667	0	0	3

Исходные данные

Используйте данные:

s3a://de-raw/lineitem

Итоговое расположение и формат

Разместите отчет в формате parquet по пути:

s3a://{USER_BUCKET}/lineitems_report

Обработка данных для отчета по Orders

Описание задачи

Необходимо построить отчет по данным о заказах (orders) содержащий сводную информацию по заказам в разрезе месяца, страны откуда был отправлен заказ, а так же приоритета выполняемого заказа.
Используйте сортировку по названию страны и приоритета заказа на возрастание.

Итоговый формат

В отчете должны присутствовать следующие колонки:

Поле	Тип	Описание
O_MONTH	str	год и месяц заказа (O_ORDERDATE) в формате YYYY-MM
N_NAME	str	название страны заказа
O_ORDERPRIORITY	str	приоритет заказа (O_ORDERPRIORITY)
orders_count	int	число заказов
avg_order_price	float	средняя цена в заказе (O_TOTALPRICE)
sum_order_price	float	сумма заказов (O_TOTALPRICE)
min_order_price	float	минимальная цена в заказе (O_TOTALPRICE)
max_order_price	float	максимальная цена в заказе (O_TOTALPRICE)
f_order_status	int	количество заказов с флагом O_ORDERSTATUS == 'F'
o_order_status	int	количество заказов с флагом O_ORDERSTATUS == 'O'
p_order_status	int	количество заказов с флагом O_ORDERSTATUS == 'P'

Пример результата

	O_MONTH	N_NAME	O_ORDERPRIORITY	orders_count	avg_order_price	sum_order_price	min_order_price	max_order_price	f_order_status	o_order_status	p_order_status
0	1998-06	ALGERIA	1-URGENT	1478	150074.397131	2.218100e+08	1273.52	438182.55	0	1478	0
1	1993-11	ALGERIA	1-URGENT	1469	147758.320933	2.170570e+08	980.53	465727.14	1469	0	0
2	1997-12	ALGERIA	1-URGENT	1450	147321.924200	2.136168e+08	1327.28	420579.94	0	1450	0
3	1997-08	ALGERIA	1-URGENT	1524	149620.194895	2.280212e+08	1200.14	413478.84	0	1524	0
4	1996-09	ALGERIA	1-URGENT	1445	149719.759370	2.163451e+08	1186.38	438319.82	0	1445	0
5	1993-07	ALGERIA	1-URGENT	1584	151124.154836	2.393807e+08	967.67	448893.77	1584	0	0
6	1997-04	ALGERIA	1-URGENT	1513	152199.063966	2.302772e+08	946.66	425953.97	0	1513	0
7	1996-07	ALGERIA	1-URGENT	1555	152636.028482	2.373490e+08	1185.22	432717.74	0	1555	0
8	1994-09	ALGERIA	1-URGENT	1486	152202.593742	2.261731e+08	1294.59	489337.27	1486	0	0
9	1992-03	ALGERIA	1-URGENT	1550	153026.210832	2.371906e+08	1015.69	506306.38	1550	0	0

Исходные данные

Используйте данные:

s3a://de-raw/orders s3a://de-raw/customer s3a://de-raw/nation

Итоговое расположение и формат

Разместите отчет в формате parquet по пути:

s3a://{USER_BUCKET}/orders_report

Обработка данных для отчета по Customers

Описание задачи

Необходимо построить отчет по данным о клиентах (customers) содержащий сводную информацию по заказам в разрезе месяца, страны откуда был отправлен заказ, а так же приоритета выполняемого заказа.
Используйте сортировку по названию страны (N_NAME) и приоритета заказа (C_MKTSEGMENT) на возрастание.

Итоговый формат

В отчете должны присутствовать следующие колонки:

Поле	Тип	Описание
R_NAME	str	название региона
N_NAME	str	название страны
C_MKTSEGMENT	str	маркетинговый сегмент
unique_customers_count	int	число уникальных заказов
avg_acctbal	float	средний остаток клиента (C_ACCTBAL)
mean_acctbal	float	медианный остаток клиента (C_ACCTBAL)
min_acctbal	float	минимальный остаток клиента (C_ACCTBAL)
max_acctbal	float	максимальный остаток клиента (C_ACCTBAL)

Пример результата

	R_NAME	N_NAME	C_MKTSEGMENT	unique_customers_count	avg_acctbal	mean_acctbal	min_acctbal	max_acctbal
0	AFRICA	ALGERIA	AUTOMOBILE	11890	4489.457164	4489.457164	-999.35	9998.88
1	AFRICA	ALGERIA	MACHINERY	11990	4471.501088	4471.501088	-999.46	9998.69
2	AFRICA	ALGERIA	HOUSEHOLD	12142	4489.274984	4489.274984	-998.98	9998.86
3	AFRICA	ALGERIA	FURNITURE	11826	4480.853280	4480.853280	-999.89	9999.01
4	AFRICA	ALGERIA	BUILDING	12068	4494.172725	4494.172725	-999.29	9998.43
5	AFRICA	ETHIOPIA	HOUSEHOLD	11967	4520.779158	4520.779158	-998.61	9999.44
6	AFRICA	ETHIOPIA	BUILDING	12197	4476.442034	4476.442034	-999.38	9999.59
7	AFRICA	ETHIOPIA	AUTOMOBILE	12185	4490.287638	4490.287638	-999.01	9999.34
8	AFRICA	ETHIOPIA	MACHINERY	12039	4541.589341	4541.589341	-999.24	9999.48
9	AFRICA	ETHIOPIA	FURNITURE	12083	4532.438614	4532.438614	-999.29	9998.72

Исходные данные

Используйте данные:

s3a://de-raw/customer s3a://de-raw/nation s3a://de-raw/region

Итоговое расположение и формат

Разместите отчет в формате parquet по пути:

s3a://{USER_BUCKET}/customers_report

Обработка данных для отчета по Suppliers

Описание задачи

Необходимо построить отчет по данным о поставщиках(suppliers) содержащий сводную информацию в разрезе страны и региона поставщика.
Используйте сортировку по названию страны (N_NAME) и региона (R_NAME) на возрастание.

Итоговый формат

В отчете должны присутствовать следующие колонки:

Поле	Тип	Описание
R_NAME	str	название региона
N_NAME	str	название страны
unique_supplers_count	int	число уникальных поставщиков в разрезе страны и региона
avg_acctbal	float	средний остаток поставщика (S_ACCTBAL)
mean_acctbal	float	медианный остаток поставщика (S_ACCTBAL)
min_acctbal	float	минимальный остаток поставщика (S_ACCTBAL)
max_acctbal	float	максимальный остаток поставщика (S_ACCTBAL)

Пример результата

	R_NAME	N_NAME	unique_supplers_count	avg_acctbal	mean_acctbal	min_acctbal	max_acctbal
0	AFRICA	ALGERIA	3934	4486.125092	4486.125092	-998.44	9999.01
1	AFRICA	ETHIOPIA	3945	4516.879161	4516.879161	-998.16	9997.98
2	AFRICA	KENYA	4044	4372.236041	4372.236041	-999.89	9992.70
3	AFRICA	MOROCCO	3990	4557.683301	4557.683301	-998.69	9995.38
4	AFRICA	MOZAMBIQUE	3924	4529.804755	4529.804755	-997.74	9997.48
5	AMERICA	ARGENTINA	4007	4487.078780	4487.078780	-995.61	9998.69
6	AMERICA	BRAZIL	3995	4537.748093	4537.748093	-996.42	9994.95
7	AMERICA	CANADA	4054	4574.455072	4574.455072	-997.13	9999.93
8	AMERICA	PERU	3991	4565.430396	4565.430396	-997.61	9999.01
9	AMERICA	UNITED STATES	4004	4548.567440	4548.567440	-999.38	9997.03

Исходные данные

Используйте данные:

s3a://de-raw/supplier s3a://de-raw/nation s3a://de-raw/region

Итоговое расположение и формат

Разместите отчет в формате parquet по пути:

s3a://{USER_BUCKET}/suppliers_report

Обработка данных для отчета по Part

Описание задачи

Необходимо построить отчет по данным о грузоперевозках (part) содержащий сводную информацию в разрезе страны поставки (N_NAME), типа поставки (P_TYPE) и типа контейнера (P_CONTAINER).
Используйте сортировку по названию страны (N_NAME) , типа поставки (P_TYPE) и типа контейнера (P_CONTAINER) на возрастание.

Итоговый формат

В отчете должны присутствовать следующие колонки:

Поле	Тип	Описание
N_NAME	str	название региона
P_TYPE	str	название страны
P_CONTAINER	str	тип контейнера
parts_count	int	число поставок

avg_retailprice	float	средняя розничная цена (P_RETAILPRICE)
size	int	суммарный размер (P_SIZE)
mean_retailprice	float	медианная розничная цена(P_RETAILPRICE)
min_retailprice	float	минимальная розничная цена(P_RETAILPRICE)
max_retailprice	float	максимальная розничная цена(P_RETAILPRICE)
avg_supplycost	float	средняя стоимость поставки (PS_SUPPLYCOST)
mean_supplycost	float	медианная стоимость поставки (PS_SUPPLYCOST)
min_supplycost	float	минимальная стоимость поставки (PS_SUPPLYCOST)
max_supplycost	float	максимальная стоимость поставки (PS_SUPPLYCOST)

Пример результата

	N_NAME	P_TYPE	P_CONTAINER	parts_count	avg_retailprice	size	mean_retailprice	min_retailprice	max_retailprice	avg_supplycost	mean_supplycost	min_supplycost	max_supplycost
0	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO BAG	41	1449.093864	1119	1449.093864	978.02	1958.94	485.485000	485.485000	4.43	936.80
1	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO BOX	53	1507.499649	1358	1507.499649	944.95	2014.88	508.132982	508.132982	25.15	988.16
2	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO CAN	58	1579.811746	1711	1579.811746	1056.05	2032.95	567.961587	567.961587	65.59	970.84
3	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO CASE	48	1499.149412	1069	1499.149412	953.04	2062.96	467.613725	467.613725	12.14	993.42
4	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO DRUM	55	1567.289508	1448	1567.289508	1077.15	1967.85	564.523279	564.523279	16.42	988.82
5	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO JAR	47	1498.489600	1109	1498.489600	955.03	1939.87	574.084800	574.084800	74.38	997.37
6	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO PACK	52	1491.001579	1470	1491.001579	978.03	2068.91	507.515965	507.515965	4.81	996.95
7	ALGERIA	ECONOMY ANODIZED BRASS	JUMBO PKG	51	1449.401538	1362	1449.401538	951.01	2001.93	438.965385	438.965385	33.03	952.80
8	ALGERIA	ECONOMY ANODIZED BRASS	LG BAG	59	1500.864844	1791	1500.864844	1018.07	2061.94	508.347656	508.347656	14.93	988.33
9	ALGERIA	ECONOMY ANODIZED BRASS	LG BOX	46	1506.698750	1199	1506.698750	1020.97	2062.93	537.926042	537.926042	16.51	983.42

Исходные данные

Используйте данные:

s3a://de-raw/part s3a://de-raw/partsupp s3a://de-raw/supplier s3a://de-raw/nation

Итоговое расположение и формат

Разместите отчет в формате parquet по пути:

s3a://{USER_BUCKET}/parts_report

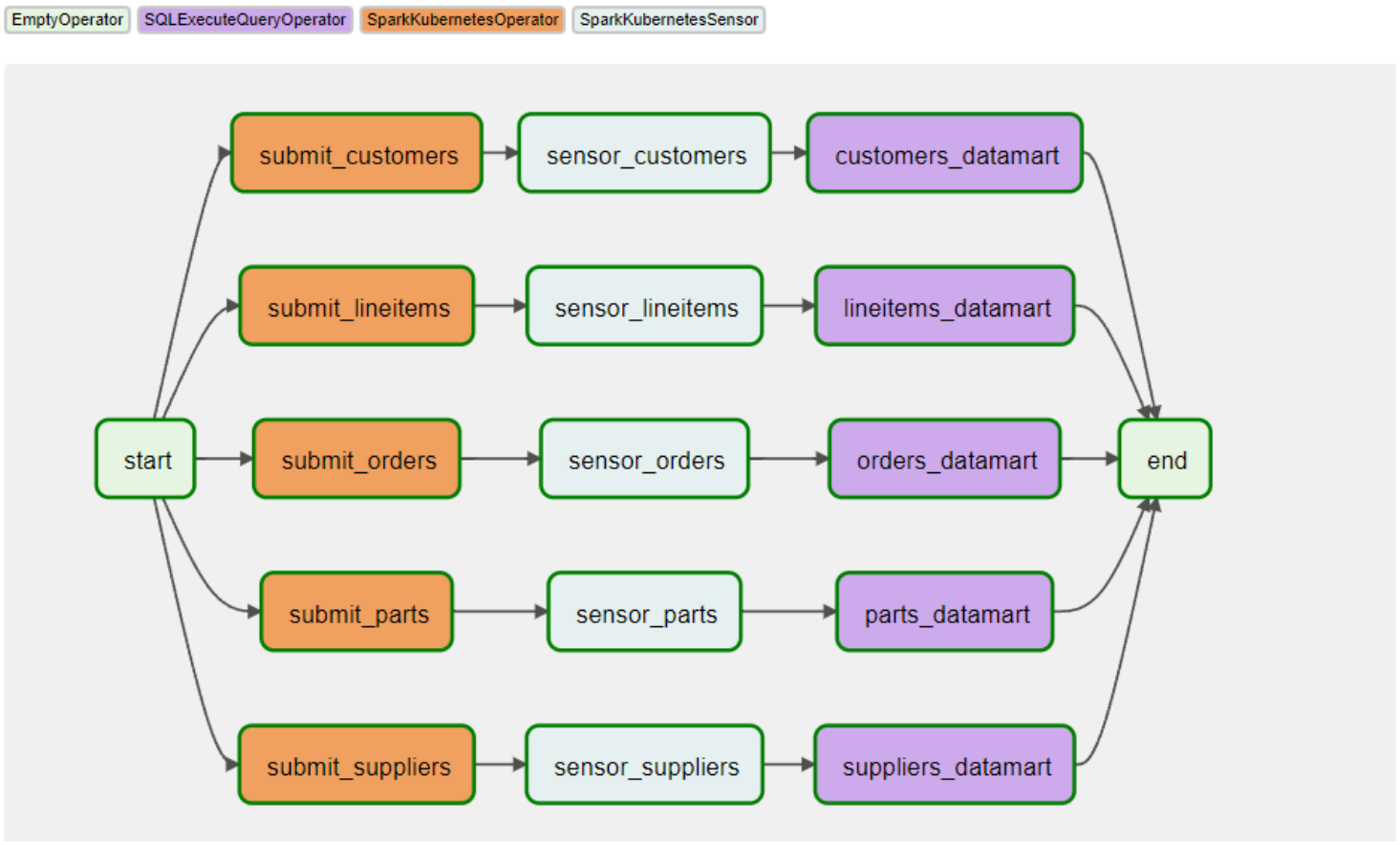
Постановка задачи

Необходимо расширить созданный вами ранее pipeline обработки данных (DAG Airflow), который реализует все запросы из последующих шагов для создания витрины в Greenplum хранилище над вашими отчетами в S3.

① Все таблицы должны создаваться в вашей схеме.
② Создание таблиц должно происходить с помощью оператора SQLExecuteQueryOperator.
③ Каждый оператор должен вызываться последовательно после построения отчета и сохранения его в вашем S3.
④ В Операторе используйте соединение connection_id = greenplume_karpov
⑤ Описание требуемых структур таблиц представлено в последующих шагах.
⑥ Создание таблиц должно быть идемпотентным.

Не забудьте зафиксировать все изменения в нашем GitLab репозитории!

Итоговый DAG'a который у нас должен получится:



Вид итогового Airflow DAG

Greenplum PXF

Для витрин мы будем использовать **Greenplum PXF** - данная технология позволит создавать нам внешние таблицы в Greenplum над нашими отчетами из внешнего хранилища S3 (Подробнее, Документация)

Пример запроса с конфигурацией:

CREATE EXTERNAL TABLE my_schema.employee(ID BIGINT, NAME TEXT) LOCATION ('pxf://s3bucket/data_path?PROFILE=s3:parquet&SERVER=default') ON ALL FORMAT 'CUSTOM' (FORMATTER='pxfwritable_import') ENCODING 'UTF8';

Строка LOCATION описывает размещение данных над которыми мы строим представление согласно создаваемой таблицы.
В строке 'pxf://s3bucket/data_path?PROFILE=s3:parquet&SERVER=default' указан:

① pxf - применяемая технология
② s3bucket - S3 бакет в облаке (Yandex Cloud)
③ data_path - путь к файлам с данными
④ PROFILE=s3:parquet - формат данных и протокол интеграции
⑤ SERVER=default - конфигурация соединения, здесь скрыты настройки доступа к S3, указан region, endpoint, access_key, secret_key

Витрина LineItems

Описание задачи

Добавьте шаг в DAG Airflow для создания внешней таблицы над отчетом LineItems.

Таблица должна называться lineitems_report и размещена в схеме студента.

s3a://{USER_BUCKET}/lineitems_report

Итоговый формат

Название: <student_schema>.lineitems

Формат витрины:

Поле	Тип (GP)	Описание
L_ORDERKEY	BIGINT	уникальный идентификатор заказа
count	BIGINT	число позиций/вещей в заказе
sum_extendprice	FLOAT8	сумма заказа (сумма всех позиций по L_EXTENDEDPRICE)
mean_discount	FLOAT8	медиана скидок по позициям (L_DISCOUNT)
mean_tax	FLOAT8	средний налог в заказе между позициями
delivery_days	FLOAT8	среднее время доставки всех позиций заказа (в днях) с момента заказа (L_RECEIPTDATE) до момента доставки (L_SHIPDATE)
A_return_flags	BIGINT	количество заказов с флагом L_RETURNFLAG == 'A'
R_return_flags	BIGINT	количество заказов с флагом L_RETURNFLAG == 'R'
N_return_flags	BIGINT	количество заказов с флагом L_RETURNFLAG == 'N'

Витрина Orders

Описание задачи

Добавьте шаг в DAG Airflow для создания внешней таблицы над отчетом Orders.
Таблица должна называться orders_report и размещена в схеме студента.

s3a://{USER_BUCKET}/orders_report

Итоговый формат

Название: <student_schema>.orders

Формат витрины:

Поле	Тип (GP)	Описание
O_MONTH	TEXT	год и месяц заказа (O_ORDERDATE) в формате YYYY-MM

N_NAME	TEXT	название страны заказа
O_ORDERPRIORITY	TEXT	приоритет заказа (O_ORDERPRIORITY)
orders_count	BIGINT	число заказов
avg_order_price	FLOAT8	средняя цена в заказе (O_TOTALPRICE)
sum_order_price	FLOAT8	сумма заказов (O_TOTALPRICE)
min_order_price	FLOAT8	минимальная цена в заказе (O_TOTALPRICE)
max_order_price	FLOAT8	максимальная цена в заказе (O_TOTALPRICE)
f_order_status	BIGINT	количество заказов с флагом O_ORDERSTATUS == 'F'
o_order_status	BIGINT	количество заказов с флагом O_ORDERSTATUS == 'O'
p_order_status	BIGINT	количество заказов с флагом O_ORDERSTATUS == 'P'

Витрина Customers

Описание задачи

Добавьте шаг в DAG Airflow для создания внешней таблицы над отчетом Customers. Таблица должна называться customers_report и размещена в схеме студента.

s3a://{USER_BUCKET}/customers_report

Итоговый формат

Название: <student_schema>.customers

Формат витрины:

Поле	**Тип (GP)**	**Описание**
R_NAME	TEXT	название региона
N_NAME	TEXT	название страны
C_MKTSEGMENT	TEXT	маркетинговый сегмент
unique_customers_count	BIGINT	число уникальных заказов
avg_acctbal	FLOAT8	средний остаток клиента (C_ACCTBAL)
mean_acctbal	FLOAT8	медианный остаток клиента (C_ACCTBAL)
min_acctbal	FLOAT8	минимальный остаток клиента (C_ACCTBAL)
max_acctbal	FLOAT8	максимальный остаток клиента (C_ACCTBAL)

Витрина Suppliers

Описание задачи

Добавьте шаг в DAG Airflow для создания внешней таблицы над отчетом Suppliers. Таблица должна называться suppliers_report и размещена в схеме студента.

s3a://{USER_BUCKET}/suppliers_report

Итоговый формат

Название: <student_schema>.suppliers

Формат витрины:

Поле	Тип (GP)	Описание
R_NAME	TEXT	название региона
N_NAME	TEXT	название страны
unique_supplers_count	BIGINT	число уникальных поставщиков в разрезе страны и региона
avg_acctbal	FLOAT8	средний остаток поставщика (S_ACCTBAL)
mean_acctbal	FLOAT8	медианный остаток поставщика (S_ACCTBAL)
min_acctbal	FLOAT8	минимальный остаток поставщика (S_ACCTBAL)
max_acctbal	FLOAT8	максимальный остаток поставщика (S_ACCTBAL)

Витрина Part

Добавьте шаг в DAG Airflow для создания внешней таблицы над отчетом Part.
Таблица должна называться parts_report и размещена в схеме студента.

s3a://{USER_BUCKET}/parts_report

Итоговый формат

Название: <student_schema>.parts

Формат витрины:

Поле	Тип (GP)	Описание
N_NAME	TEXT	название региона
P_TYPE	TEXT	название страны
P_CONTAINER	TEXT	тип контейнера
parts_count	BIGINT	число поставок
avg_retailprice	FLOAT8	средняя розничная цена (P_RETAILPRICE)
size	BIGINT	суммарный размер (P_SIZE)
mean_retailprice	FLOAT8	медианная розничная цена(P_RETAILPRICE)
min_retailprice	FLOAT8	минимальная розничная цена(P_RETAILPRICE)
max_retailprice	FLOAT8	максимальная розничная цена(P_RETAILPRICE)
avg_supplycost	FLOAT8	средняя стоимость поставки (PS_SUPPLYCOST)
mean_supplycost	FLOAT8	медианная стоимость поставки (PS_SUPPLYCOST)

min_supplycost	FLOAT8	минимальная стоимость поставки (PS_SUPPLYCOST)
max_supplycost	FLOAT8	максимальная стоимость поставки (PS_SUPPLYCOST)

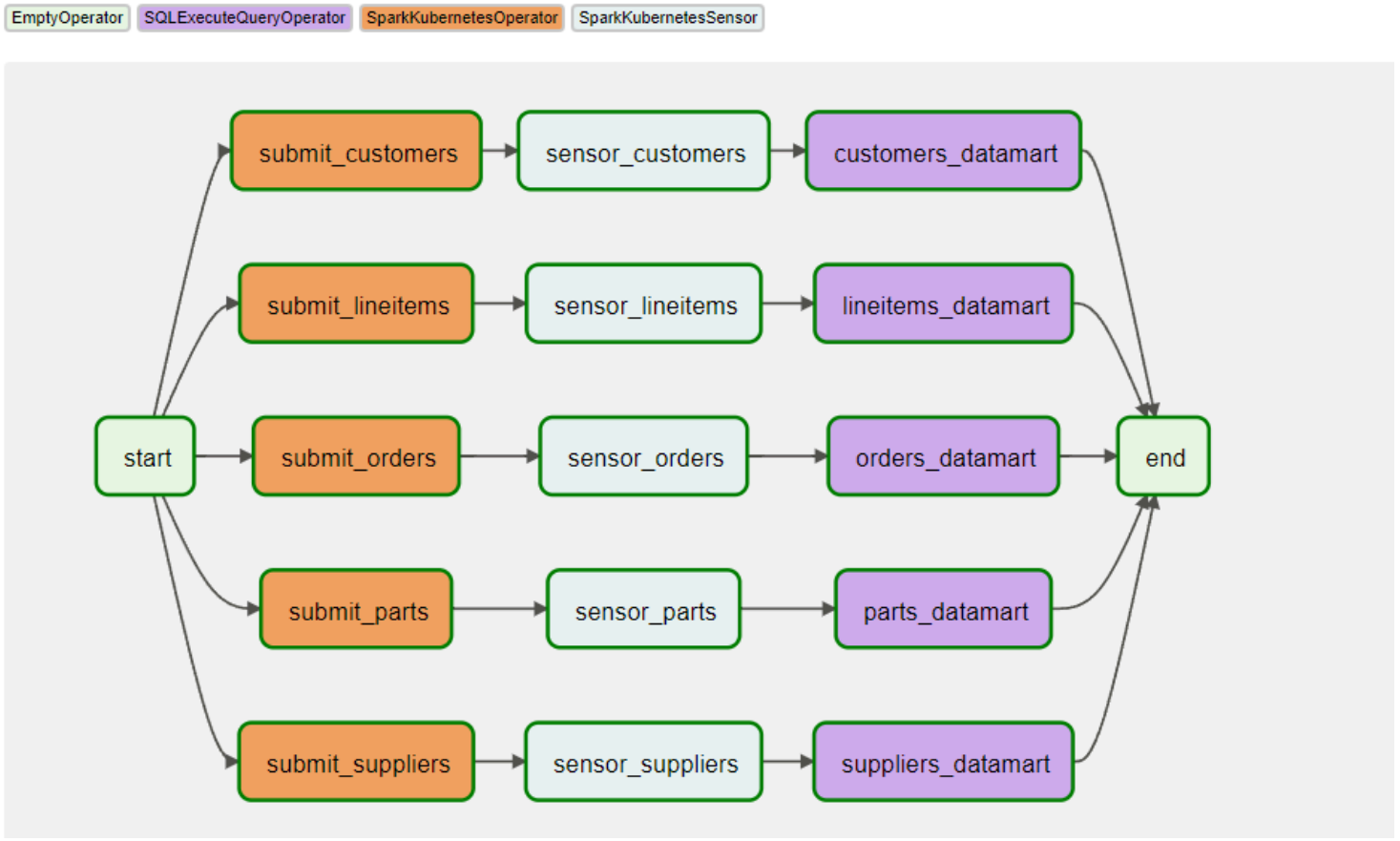
Оценка работы

Критерии оценки согласно требованиям описанным в заданиях предыдущих разделов.

Убедитесь что последний запуск вашего Airflow DAG завершен успешно!

Для корректной проверки результатов, ваш DAG должен иметь успешный последний запуск. Пожалуйста не используйте запуск по расписанию, запустите ваш DAG в ручную, если у вас нет успешного запуска. Если у вас есть шаги завершившиеся с ошибкой, помните, вы можете перезапустить именно нужный шаг, а не весь DAG.

Вы можете запускать проверку решения на любой стадии выполнения задач!



Раздел	Задача	Баллы
Airflow	У вашего Airflow DAG последний запуск (DagRun) является успешным (status=SUCCESS)	5
Airflow	У вашего Airflow DAG присутствую шаги с запуском Spark-задач отчетов. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	1x5
Паиплайны озера данных (Spark+S3)	В вашем S3 бакете обнаружен отчет с правильной схемой данных. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	1x5

Паиплайны озера данных (Spark+S3)	В вашем S3 бакете обнаружен отчет с правильным содержанием данных. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	4x5
Airflow	У вашего Airflow DAG присутствую шаги с запуском построения представлений в Greenplum. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	1x5
Паиплайны витрин данных (S3+Greenplum)	В вашей схеме Greenplum обнаружено представление для отчета с правильной схемой данных. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	1X5
Паиплайны витрин данных (S3+Greenplum)	В вашей схеме Greenplum обнаружено представление для отчета с правильным содержанием данных. Бал дается за каждый из 5 отчетов (customers_report, orders_report, suppliers_report, parts_report, lineitems_report)	3X5