Работа с Hive

1. Подготовка к работе

Подготовьте файл `Работа с Hive - отчёт.docx`. Помещайте в него выполняемые команды и вставляйте скриншоты выводимых на экран результатов.

После выполнения предыдущих заданий по работе с кластером Hadoop у Вас уже должен быть скачан и распакован архив `bigd_course_hadoop.zip`. Если нет - скачайте и распакуйте его.

Перейдите в консоли в папку `bigd_course_hadoop` и, используя имеющийся в ней конфигурационный файл, поднимите кластер Hadoop и дождитесь, когда все его узлы перейдут в статус `healthy`.

Если по окончании выполнения предыдущих заданий Вы удалили тома кластера Hadoop, заново скопируйте содержимое папки `files` в файловую систему узла имён, в папку `/home/user`.

Перейдите в файловую систему узла имён

```
docker exec -it namenode bash
```

Создайте в HDFS следующие папки (возможно папка `/user` уже создана при выполнении предыдущих заданий; создайте недостающие):

```
/user/hive/warehouse
```

2. Создание внутренней таблицы, управляемой Hive

Для работы с Hive мы будем использовать оболочку командой строки `beeline`. Чтобы начать работать с `beeline` можно сначала перейти в файловую систему Hive-сервера `docker exec -it hiveserver2 bash` (этой командой мы запустили оболочку командной строки `bash`) и затем выполнить команду `beeline`. Но удобнее вызвать `beeline` напрямую, из командой строки хоста:

```
docker exec -it hiveserver2 beeline -u 'jdbc:hive2://hiveserver2:10000/'
```

Здесь `beeline` получает параметр `-u` в котором записан URL базы данных с указанием используемого драйвера (у нас это `jdbc:hive2`).

Работа в `beeline` аналогична работе с консольными утилитами других SQL-баз данных, таких как `mysql` для MySQL и `psql` для PostgeSQL.

Изучите структуру запроса ниже.

```
CREATE TABLE test_table1 (
    id INT,
    name STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'hdfs://namenode:9000//user/hive/warehouse/test_table1';
```

Запишите в отчёт ответы на следующие вопросы:

- Какого типа будет таблица: управляемая Hive (Managed Table) или внешняя (External)?
- Какие столбцы будут в таблице, какой у них тип данных?
- В каком формате будет сохранены данные?
- Какой символ будет использован в качестве разделителя данных в строках?
- В какой файловой системе и по какому адресу будут храниться файлы данных?

Выполните запрос.

Выполните команду просмотра имеющихся таблиц. Убедитесь, что таблица создана:

```
show tables;
```

Поместите в отчёт скриншот вывода этой команды.

Поместите в созданную таблицу данные при помощи следующего запроса:

```
INSERT INTO test_table1 VALUES (1, 'Alice'), (2, 'Bob'), (3, 'Jane'), (4,
'Paul');
```

Напишите и выполните запрос, который выводит все записи таблицы `test_table1`. Вы должны увидеть таблицу с записанными в неё данными. Поместите скриншот в отчёт.

Созданная таблица управляется Hive - он имеет возможность создавать и уничтожать файлы данных, отображаемые в этой таблице. Что в этом убедиться, создадим копию таблицы `test_table1`, посмотрим на расположение фалов обеих таблиц, а потом удалим вторую и снове проверим расположение файлов.

Для создания копии таблицы выполните запрос

```
CREATE TABLE test_table1_copy
LOCATION 'hdfs://namenode:9000//user/hive/warehouse/test_table1_copy'
AS SELECT * FROM 'test_table1';
```

Выйдите из консоли `beeline` и перейдите в файловую систему узла имён. При помощи команд HDFS `-ls -R` выведите на экран файлы в хранилище Hive (в папке `warehouse`). Поместите скриншот в отчёт.

Используя команду HDFS `-cat` выведите на экран содержимое файлов, которые хранят данные таблиц `test_table1` и `test_table1_copy`. Идентичны ли эти данные? Поместите скриншоты в отчёт.

Вернитесь в консоль `beeline` (выполните в консоли хоста команду, приведённую выше) и удалите вторую таблицу при помощи следующей команды

```
drop table test_table1_copy;
```

Убедитесь, что теперь имеется одна таблица:

```
show tables;
```

Снова перейдите в файловую систему узла имён, выведите на экран файлы в хранилище Hive и поместите скриншот в отчёт. Что изменилось после удаления второй таблицы?

3. Создание внешней таблицы

Мы собираемся создать внешнюю таблицу на основе существующего файла `exams.csv`. Hive не умеет создавать таблицы на основе отдельных файлов. При создании внешней таблицы нужно указать папку, все файлы который буду отображаться в таблице.

Найдите файлы `exams.csv` и `exams2.csv` (либо в файловой системе хоста, в папке `files`, либо в локальной файловой системе узла имён, в папке `/home/user`) и поместите их содержание в отчёт.

Скопируйте `exams.csv` и `exams2.csv` из папки хоста в локальную файловую систему узла имён, в папку `/home/user`, если их там ещё нет.

Находясь в файловой системе узла имён, создайте в HDFS папку `/user/data/exams` и скопируйте туда файл `exams.csv`. Скопируйте только этот файл, второй файл `exams2.csv` пока не трогайте!

Перейдите в консоль `beeline`. Изучите устройство следующего запроса:

```
CREATE EXTERNAL TABLE exams (
    year INT,
    math FLOAT,
    compsc FLOAT,
    phys FLOAT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LOCATION 'hdfs://namenode:9000/user/data/exams'
TBLPROPERTIES ("skip.header.line.count"="1");
```

Запишите в отчёт ответы на следующие вопросы:

- Какого типа будет таблица: управляемая Hive (Managed Table) или внешняя (External)?
- Какие столбцы будут в таблице, какой у них тип данных?
- В какой файловой системе и по какому адресу находятся файлы данных?
- Сколько строк из начала файла будет пропущено при чтении?

Выведите на экран список имеющихся таблиц. Убедитесь, что новая таблица создана.

Напишите запрос, который выводит на экран содержание таблицы `exams`. Поместите скриншот в отчёт. Сравните его с содержанием исходного файла `exams.csv`.

4. Обновление и добавление данных во внешней таблице

Перейдите в файловую систему узла имён, запустите при помощи консольной команды `mc` файловый менеджер Midnight Commander.

Перейдите в MC в папку `/user/data/exams`, откройте при помощи клавиши F4 на редактирование файл `exams.csv`.

Допишите в него строку `2025,100,100,100` и сохраните сделанные изменения.

Выйдите из MC, нажав клавищу F10.

Скопируйте обновлённый файл `exams.csv` в HDFS с заменой уже имеющейся там старой версии файла. Для этого нужно использовать комнду `-put` с дополнительно опцией `-f` (от слова force что в данном случае означает принудительно).

```
hdfs dfs -put -f /home/user/exams.csv /user/data/exams/
```

Перейдите в `beeline` и выполните запрос, который выводит на экран содержание таблицы `exams`. Убедитесь, что в таблице появилась строка 2025 года. Сделайте и поместите в отчёт скриншот.

Перейдите в файловую систему узла имён и скопируйте в HDFS, в папку данных таблицы exams файл `exams2.csv`.

Используя команду HDFS `-ls` убедитесь, что в папке таблицы `exams` теперь находятся файлы `exams.csv` и `exams2.csv`.

Перейдите в `beeline` и снова выполните запрос, выводящий на экран таблицу `exams`. Убедитесь, что теперь в таблицу помещено содержание обоих файлов. Сделайте скриншот и поместите его в отчёт.

После того как выполнение задания окончено, выйдите из консоли контейнера, с которым работали и остановите кластер:

```
docker compose down
```

Для очистки долговременной памяти своего компьютера, удалите тома файловых систем узлов кластера командой ниже:

```
docker volume prune --all
```

Образы контейнеров также занимают довольно много места. Для полной очистки долговременной памяти от сохранённых образом используйте команду

```
 docker system prune --all
```

ВАЖНО: это приведёт к полному удалению всех образов и всех томов. Если есть такая возможность, не выполните полную очистку до конца семестра.