Методы анализа данных

Практическая работа №2

Работа с Apache Sqoop

# Исследование данных с помощью Sqoop для РСУБД (MariaDB)

В этой практической работе вам предстоит импортировать и экспортировать таблицы из РСУБД в HDFS (Hadoop Distributed Filesystem), используя Sqoop.

Импортирование таблицы в HDFS с использованием Sqoop.

Начнем с получения информации о базах данных и таблицах в них перед импортированием таблиц в HDFS.

## В терминале авторизуйтесь в MariaDB и выберите базу данных labs.

БД: labs

$ mysql --user=student --password=student labs

Замечание: Если вы не введете ничего в качестве пароля (password), то он будет запрошен:

$ mysql -u student -p labs

Введите "student".

## Если авторизация прошла успешно, то приглашение для ввода сменится на "MariaDB [labs]>". Введите команду для проверки того, какие базы данных доступны.

MariaDB> show databases;

텍스트이(가) 표시된 사진

자동 생성된 설명

Рисунок 1. Список баз данных в MariaDB

## Затем введите команду для просмотра таблиц в базе labs:

텍스트이(가) 표시된 사진

자동 생성된 설명

Рисунок 2. Список названий таблиц в БД labs

На экране отображены таблицы authors и posts.

Эти таблицы мы будем импортировать и экспортировать через команды Sqoop.

MariaDB> desc authors;

MariaDB> describe posts;

Замечание: Команды desc и describe выполняют одно и то же.

텍스트이(가) 표시된 사진

자동 생성된 설명

Рисунок 3. Структура таблиц authors и posts

## Посмотрим на структуру таблиц authors и post, а также на несколько записей оттуда.

MariaDB> SELECT id, first\_name, last\_name, email, added

FROM authors limit 5;

## Для выхода из MariaDB напишите quit и нажмите Enter.

MariaDB> quit

## Для получения помощи по базовым командам sqoop запустите следующую команду:

$sqoop help

## Для получения детализированной информации по каждой подкоманде, введите ее название после help. Например, для получения справки по команде import, выполните следующую команду:

$sqoop help import

## Просмотр списка баз данных в MariaDB и таблиц в базе данных labs осуществляется следующей командой:

﻿$sqoop list-databases --connect jdbc:mysql://localhost --username student --password student

Результат выполнения этой команды совпадает с тем, что мы видели на рисунке 1.

Замечание: альтернативой использованию аргумента --password является использование ключа -P (заглавная буква) и ввод пароля при запросе. При этом вводимые символы будут невидимы.

﻿$sqoop list-tables --connect jdbc:mysql://localhost/labs --username student -P

В результате выполнения этой команды будут выведены таблицы authors и posts.

## Импортируем все таблицы в базе labs используя команду import-all-tables.

﻿$sqoop import-all-tables --connect jdbc:mysql://localhost/labs \

--username student --password student

Замечание: в Sqoop хоть и есть команда import-all-table, на практике она редко используется.

Причина в том, что эта команда за раз выполняет много задач. Постарайтесь по возможности ее больше не использовать.

В реальных средах обычно сотни баз данных с тысячами таблиц в каждой, поэтому используйте эту команду только для тестов.

Обычно импортирование даже одной таблицы занимает много времени. В большинстве случаев используйте команду import.

## Выполните эту команду для получения таблицы posts из базы labs и сохранения последней в HDFS:

$ ﻿sqoop import --connect jdbc:mysql://localhost/labs \

--username student --password student --table posts

После выполнения этой команды создастся директория posts в домашней директории HDFS /user/student, а данные сохранятся как показано ниже:

텍스트이(가) 표시된 사진

자동 생성된 설명

Рисунок 4. Список постов в HDFS

## Создаем целевую директорию в HDFS для импортирования данных в нее.

$hdfs dfs -mkdir /mywarehouse

## Импортируем таблицу authors и сохраняем ее в директорию HDFS, которую мы создали выше, используя ‘,’ для разделения полей.

Замечание: аргумент --field-terminated-by ‘,’ используется для задания запятой в качестве разделителя полей в HDFS файле. Если вы хотите работать с Hive или Spark, то лучше использовать ‘\t’ вместо ‘,’.

$ sqoop import --connect jdbc:mysql://localhost/labs \

--username student --password student \

--table authors --fields-terminated-by ',' \

--target-dir /mywarehouse/authors

## Для проверки воспользуемся командой hdfs по целевой директории.

$ ﻿hdfs dfs -ls /mywarehouse/authors

$ ﻿hdfs dfs -cat /mywarehouse/authors/part-m-00000

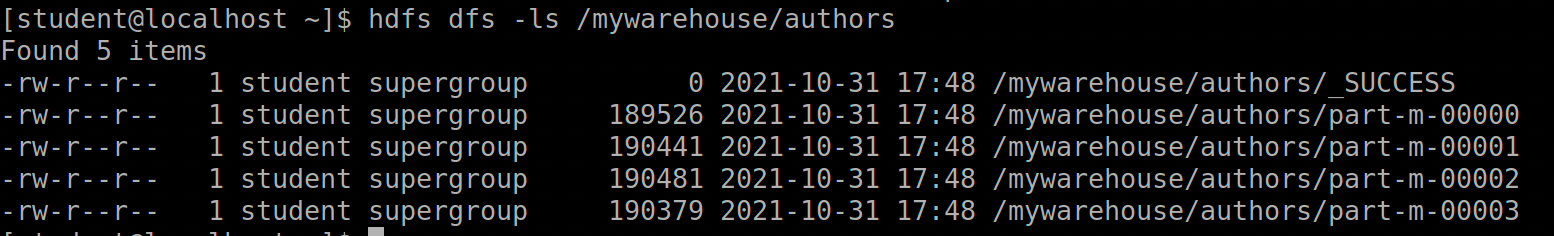


Рисунок 5. Список файлов в /mywarehouse/authors

Если запустить команду cat, вы сможете увидеть, что каждая строка данных отделена от соседней знаком ‘,’, что отличается от того, что было в файле posts (там были табуляции) в HDFS.

## Импортирование определенных столбцов осуществляется указанием аргумента --columns для папки authors в домашней директории HDFS. Импортируемые столбцы: first\_name, last\_name, email

$ sqoop import --connect jdbc:mysql://localhost/labs --username student --password student --table authors --fields-terminated-by '\t' --columns "first\_name, last\_name, email"

텍스트이(가) 표시된 사진

자동 생성된 설명

Рисунок 6. Результаты команды Sqoop

## Импортируем только подходящие строки с помощью --where. Для примера импортируем строки из таблицы authors, где first\_name указано, как ‘Dorthy’.

$ sqoop import --connect jdbc:mysql://localhost/test --username student --password student --table authors --fields-terminated-by '\t' --where "first\_name='Dorthy'" --target-dir authors\_Dorthy

Замечание: вывод работ (job) Hadoop сохраняется в один или более файлов разделов (partition). Обычно создаются 4 файла, а результаты запроса помещаются в произвольный файл.

## Импортируем таблицу, используя другой формат файла, вместо простого текстового. Импортируем таблицу authors в файл формата Parquet:

﻿$sqoop import --connect jdbc:mysql://localhost/labs --username student --password student --table authors --target-dir /mywarehouse/authors\_parquet --as-parquetfile

## Смотрим результат при помощи отображения содержимого целевой директории HDFS:

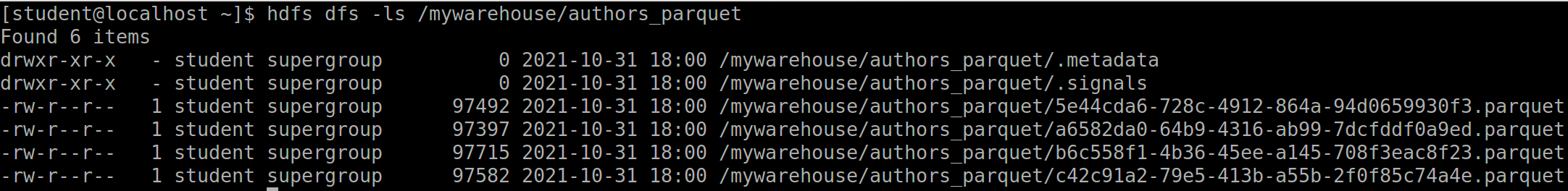
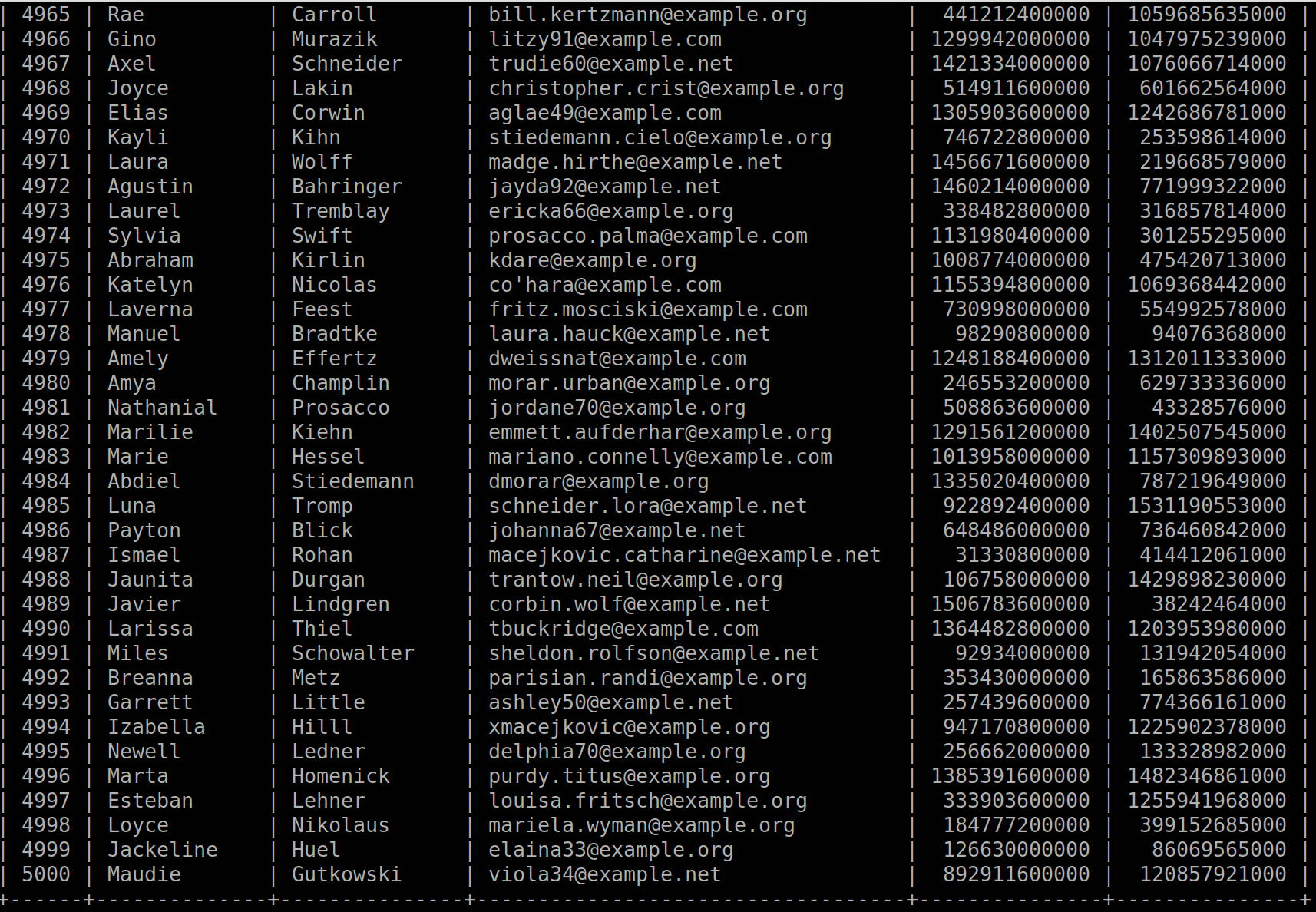


Figure 7. Список файлов Parquet

Каждому файлу Parquet дано уникальное имя, например, 5e44cda6-728c-4912-864a-94d0659930f3.parquet. Данные в этом формате нельзя просмотреть напрямую, так как формат двоичный. Для просмотра записей в таких файлах используйте команду parquet-tools show. Сначала нужно получить файл parquet на локальный узел, а затем запустить команду parqet-tools:

﻿$ hdfs dfs -get ﻿﻿/mywarehouse/authors\_parquet/5e44cda6-728c-4912-864a-94d0659930f3.parquet

$ parquet-tools show ﻿5e44cda6-728c-4912-864a-94d0659930f3.parquet



## Импортируем таблицу authors, используя сжатие (аргумент --compress или -z):

$ sqoop import --connect jdbc:mysql://localhost/labs --username student --password student --table authors --target-dir /mywarehouse/authors\_compressed --compress

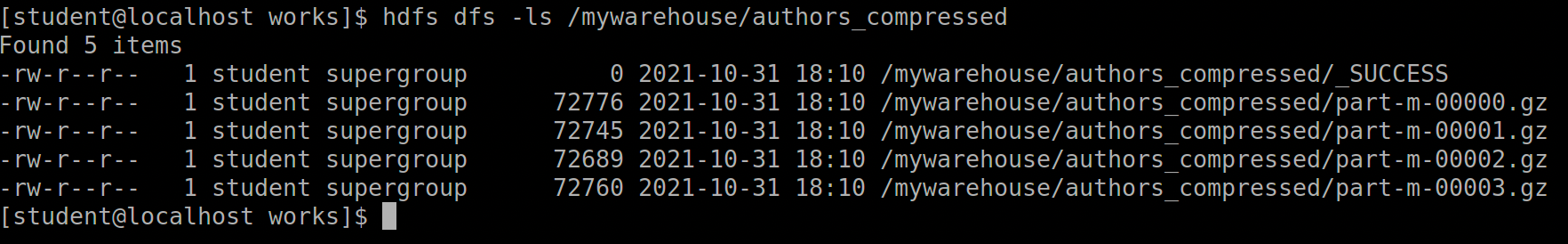


Рисунок 8. Список сжатых файлов

## Импортируем строки, где first\_name указано, как «Dorthy» (мы уже делали это в шаге 15) и сохраняем результат в папку dorthy в домашней директории HDFS:

﻿$ sqoop import --connect jdbc:mysql://localhost/labs --username student --password student --table authors --fields-terminated-by '\t' --where "first\_name='Dorthy'" --target-dir dorthy

Экспортируем сохраненную папку dorthy в качестве таблицы в базу данных labs:

﻿$sqoop export --connect jdbc:mysql://localhost/labs --username student --password student --table authors\_export --fields-terminated-by '\t' --export-dir dorthy

## Посмотрим содержимое экспортированных записей в MariaDB:

텍스트, 전자기기, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

Рисунок 9. Экспортированные записи в MariaDB

## Создайте целевую директорию в HDFS для импортирования данных таблицы из базы labs (/tmp/mylabs).

### Из таблицы posts импортируйте первичный ключ, заголовки постов, дату публикации в директорию HDFS /tmp/mylabs/posts\_info. Сохраните файл в текстовом формате с табуляциями в качестве разделителей.

Подсказка: вам необходимо выяснить названия нужных столбцов, используя команды sqoop, а не напрямую в MariaDB.

### Теперь сохраните то же самое, только в формате parquet со сжатием snappy в папке /tmp/mylabs/posts\_compressed.

### Из окна терминала отобразите несколько записей из тех, что вы только что проимпортировали.

### Импортируйте столбцы id, first\_name, last\_name, birthdate из таблицы authors в домашнюю директорию HDFS. Файл сохраните в текстовом формате с табуляциями в качестве разделителей.

Подсказка: если директория authors уже существует в домашней директории HDFS, удалите ее и только потом выполните команду import.

### Импортируйте и сохраните в директории /tmp/mylabs/posts\_NotN те посты, название которых не NULL. Импортируйте только следующие столбцы: id, title, content. Сохраните файл в формате parquet, а для сжатия используйте кодек snappy.

Подсказка: выражение SQL для проверки того, что значение в столбце не NULL: "название\_столбца is not NULL"

SQL будет объяснен на лекции.