|  |
| --- |
| АЛЬФА-БАНК |
| Impala user guide |
| Введение в работу с Impala для новых пользователей |
|  |
|  |
|  |

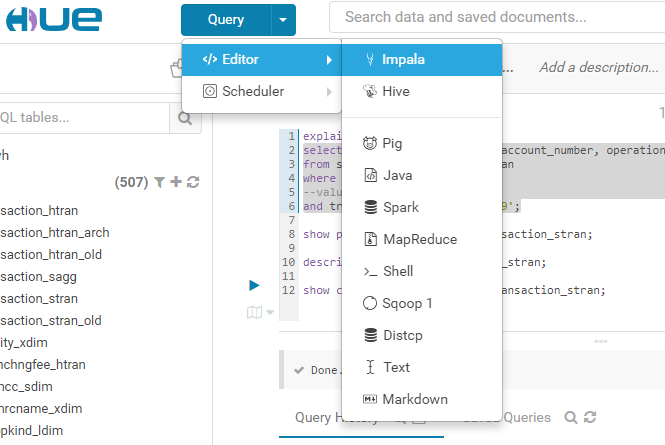
# История изменений документа

|  |  |  |  |
| --- | --- | --- | --- |
| ***Версия*** | ***Дата*** | ***Описание изменений*** | ***Автор*** |
| 0.1 | 23.07.2018 | Первоначальная версия | Кабанов В. |
| 0.2 | 12.11.2018 | Добавлен раздел «Как писать производительные запросы». Исправлены опечатки. Увеличен масштаб рисунков. | Кабанов В. |

# Введение

В документе описаны особенности и ограничения работы с Impala, в частности в сравнении с Hive. Также описаны случаи, когда следует применять Impala и наиболее часто задаваемые вопросы.

# Начало работы

Запросы в Impala проще всего писать через приложение Hue. Для этого в вкладке Query нужно выбрать Editor > Impala.

Через Impala можно посылать запросы к таблицам Hive (при условии совместимости форматов, см. ниже), можно создавать свои таблицы, в том числе внешние.

# Область применения

Impala – это высокопроизводительный движок SQL запросов к базам данных на основе файловой системы Hadoop.

Impala подходит для

* Запросов с фильтрацией по конкретному ключу сущности, например, «вывести все транзакции клиента XYZ»
* Запросов с группировками и аналитическими функциями
* ETL

# Особенности работы с Impala

## Как писать (производительные) запросы на Impala?

### Партицирование

Партицирование – это разделение логически одной таблицы на физически раздельные файлы данных. При создании таблицы выбирается поле, по которому она будет партицирована. Это поле называется «ключом партицирования». Если в запросе к партицированной есть фильтр по ключу партицирования, то база данных до выполнения запроса может понять, какие файлы ей необходимо прочитать. **Использование партицирования является одним из главных способов оптимизации запросов на Impala.**

Примеры

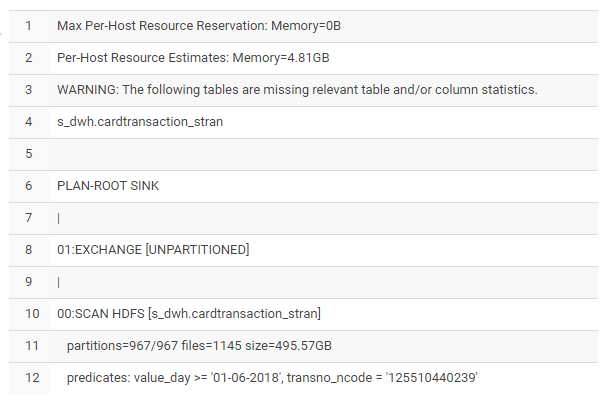
*Примечание:* в примерах ниже для анализа запросов используется команда EXPLAIN <query>. Команда показывает план выполнения запроса, его трудоемкость и ресурсоемкость. Рекомендуется выполнять EXPLAIN **перед** запуском объемных запросов.

Запрос с фильтрацией по дате

select \* from s\_dwh.cardtransaction\_stran  
where value\_day>='01-06-2018'  
and transno\_ncode='125510440239';

Вывод команды EXPLAIN

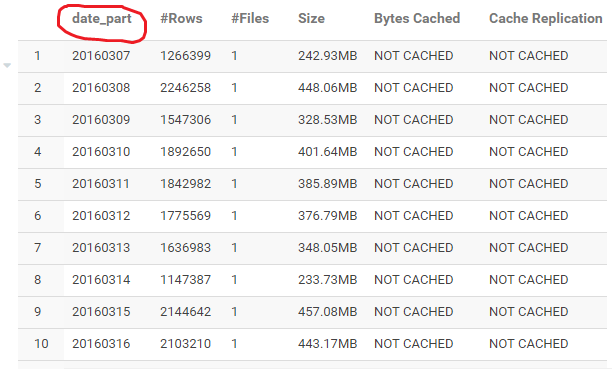
EXPLAIN select \* from s\_dwh.cardtransaction\_stran  
where value\_day>='01-06-2018'  
and transno\_ncode='125510440239';



Согласно 11 строчке, для выполнения запроса Impala просканирует 1145 файлов общим размером 495 гигабайт. Посмотрим, можно ли уменьшить этот объем.

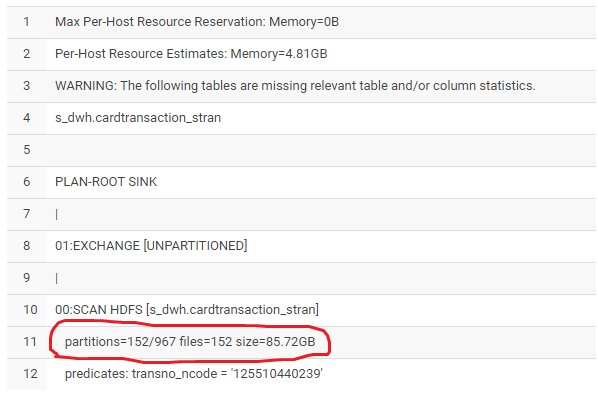
Для просмотра информации о партицировании таблицы нужно отдать команду SHOW PARTITIONS <table\_name>.

SHOW PARTITIONS s\_dwh.cardtransaction\_stran;



Первый столбец в выводе команды – это ключ партицирования таблицы, в нашем случае date\_part. Посмотрим, как изменится план запроса:

EXPLAIN select \* from s\_dwh.cardtransaction\_stran  
where date\_part>=20180601  
and transno\_ncode='125510440239';



Как видно, объем чтения уменьшился почти в 6 раз. Время выполнения запроса без фильтра по ключу партицирования – **48 минут.** Время выполнения запроса с фильтром по ключу партицирования – **14 минут**.

*Примечание:* как правило, партицируются большие фактовые таблицы. Ключом партицирования в этом случае выбирается бизнес-дата, ассоциированная с фактом (отчетная дата, дата срез или совершения операции). **Важно**: **ключ партицирования таблиц в Impala, как правило, отличается от ключа в DWH.** Обычно, ключ партицирования таблиц в Impala называется date\_part.

### Колоночный формат хранения данных

Хотя Impala поддерживает разные форматы хранения данных (см. параграф [Работа с файловыми форматами](#_Работа_с_файловыми)), данные, реплицируемые из ядра и витрин DWH, хранятся в формате Parquet. Parquet является колоночным форматом хранения данных, то есть данные таблицы группируются **сначала по столбцам, потом по строчкам.**

Для пользователя это значит, что выбор только конкретных столбцов из таблицы пропорционально уменьшает объем чтений – потому что Impala читает только ту часть файла Paruqet, которая содержит запрашиваемые столбцы. Таким образом, при работе с Impala следует приучать себя к тому, чтобы запрашивать из таблицы только нужные для анализа столбцы (не писать «select \* »).

В качестве примера рассмотрим запрос к таблице s\_dwh.cardtransaction\_stran из предыдущего пункта. После учета партицирования мы получили запрос

select \* from s\_dwh.cardtransaction\_stran  
where date\_part>=20180601  
and transno\_ncode='125510440239';

который работал **14 минут** и, согласно результатам выполнения, прочитал **79GB.** Предположим, что для нашей задачи нас интересуют PIN клиента, дата совершения операции, номер счета и сумма операции. Тогда правильный запрос примет вид

select client\_pin, value\_day, account\_number, operation\_cur\_amt  
from s\_dwh.cardtransaction\_stran  
where date\_part>=20180601  
and transno\_ncode='125510440239';

Данный запрос работал **1 минуту** и прочитал **11GB** данных.

Итого, учитывая факторы партицирования и колоночного хранения, мы смогли получить ускорение запроса в **48 раз.**

Примечание: статистика по результатам выполнения запросов доступна по адресу [https://bda31node03:7183/cmf/services/17/queries](https://bda31node03:7183/cmf/services/17/queries" \t "_blank) (для кластера BDA31) или по адресу Cloudera Manager -> Impala -> Queries для других кластеров.

## Работа с файловыми форматами

Impala умеет работать только с файловыми форматами из таблицы ниже. Если файловый формат не указан в таблице, Impala не может с ним работать. **В частности, Impala не может работать с форматом ORC.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **File Type** | **Format** | **Codecs** | **Impala Can CREATE?** | **Impala Can INSERT?** |
| [Parquet](https://impala.apache.org/docs/build/html/topics/impala_parquet.html#parquet) | Structured | Snappy, gzip; currently Snappy by default | Yes. | Yes: CREATE TABLE, INSERT, LOAD DATA, and query. |
| [Text](https://impala.apache.org/docs/build/html/topics/impala_txtfile.html#txtfile) | Unstructured | LZO, gzip, bzip2, Snappy | Yes. For CREATE TABLE with no STORED AS clause, the default file format is uncompressed text, with values separated by ASCII 0x01 characters (typically represented as Ctrl-A). | Yes: CREATE TABLE, INSERT, LOAD DATA, and query. If LZO compression is used, you must create the table and load data in Hive. If other kinds of compression are used, you must load data through LOAD DATA, Hive, or manually in HDFS. |
| [Avro](https://impala.apache.org/docs/build/html/topics/impala_avro.html#avro) | Structured | Snappy, gzip, deflate, bzip2 | Yes, in Impala 1.4.0 and higher. Before that, create the table using Hive. | No. Import data by using LOAD DATA on data files already in the right format, or use INSERT in Hive followed by REFRESHtable\_name in Impala. |
| [RCFile](https://impala.apache.org/docs/build/html/topics/impala_rcfile.html#rcfile) | Structured | Snappy, gzip, deflate, bzip2 | Yes. | No. Import data by using LOAD DATA on data files already in the right format, or use INSERT in Hive followed by REFRESHtable\_name in Impala. |
| [SequenceFile](https://impala.apache.org/docs/build/html/topics/impala_seqfile.html#seqfile) | Structured | Snappy, gzip, deflate, bzip2 | Yes. | No. Import data by using LOAD DATA on data files already in the right format, or use INSERT in Hive followed by REFRESHtable\_name in Impala. |

Источник: https://impala.apache.org/docs/build/html/topics/impala\_file\_formats.html

## Работа с метаданными и статистикой

**Статистика** в Impala, как и в других базах данных, это информация о кардинальности таблицы и значениях столбцов, которая нужна для выбора оптимального плана запроса. В Impala есть 2 режима сбора статистики – обычный и инкрементальный. Инкрементальный режим работает по набору партиций, его нужно использовать для больших партицированных таблиц. Эти режимы взаимоисключающие, то есть, если вы запускаете COMPUTE INCREMENTAL STATS по таблице, для которой посчитана обычная статистика, перед этим нужно отдать команду DROP STATS (и наоборот).

Команды выглядят следующим образом. Для сбора обычной статистики:

COMPUTE STATS ACCOUNT\_HDIM;

Для сбора инкрементальной статистики (возможны другие типы условий для выбора партиций):

COMPUTE INCREMENTAL STATS card\_transaction (date\_part BETWEEN 20170101 and 20170601);

COMPUTE INCREMENTAL STATS, вообще говоря, работает дольше COMPUTE STATS **на том же объеме данных**, поэтому его следует использовать для таблиц, в которых меняется ограниченный объем партиций.

Согласно документации Impala, следует обновлять статистику после добавления в таблицу более 30% строк от первоначального объема. **Выявлять несвежую статистику можно 2 способами: командами SHOW TABLE STATS или SHOW PARTITIONS (тогда значение поля #Rows=-1) или просмотром вывода EXPLAIN (в шапке плана будет предупреждение).**

**Метаданные** в Impala – это информация о том, в каких файлах и блоках HDFS хранятся данные таблицы. Эта информация в первую очередь нужна для корректной работы запроса, без этого запрос может упасть, если попытается прочитать несуществующий файл. Кроме того, Impala использует эту информацию для оптимизации выполнения запросов – чтобы сохранять локальность задачи при распределении запроса по нодам.

Действия пользователя для обновления метаданных нужны только в том случае, если таблицы менялись в другом приложении, например, в Hive. То есть влияние всех своих действий на метаданные таблиц Impala учитывает сама.

Для обновления метаданных используют 2 команды:

INVALIDATE METADATA BALANCE\_SSTAT

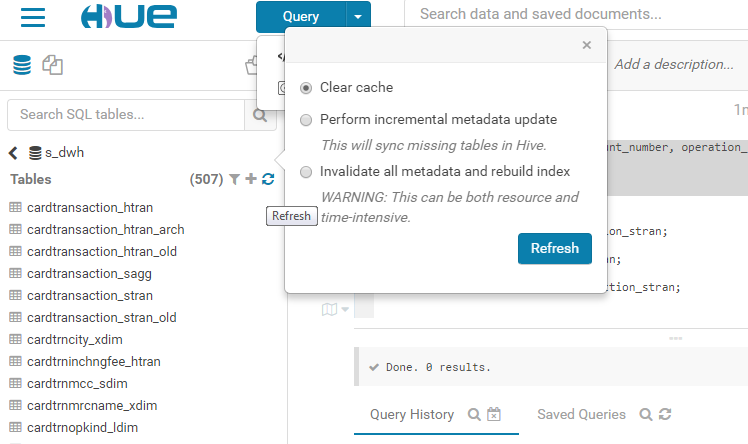
Сбрасывает все метаданные таблицы, так, что следующее обращение к ней вызывает их полный пересбор. Более трудоемкая операция, нужно применять **только если таблицу создали в Hive**, чтобы сделать таблицу доступной для Impala.

REFRESH BALANCE\_SSTAT;

Операцию нужно выполнять, если поменялся состав файлов таблицы.

Команду INVALIDATE METADATA можно также отдавать к сервису Impala целиком – это инвалидирует все метаданные. Отдать команду к отдельной базе нельзя.

Команду REFRESH можно выполнить нажав на соответствующую иконку в Tables Browser-е в левом окне



## Возможности Impala в части функций SQL

Impala обладает всеми стандартными возможностями SQL. По сравнению с Hive отсутствуют некоторые специальные возможности: нет функций для работы с XML и JSON. Impala поддерживает nested types но только на чтение (сделать insert в таблицу с nested types нельзя) и только в формате Parquet.

Возможности Impala части UDF, UDAF:

* Impala поддерживает как нативные UDF, написанные на C++, так и UDF на Java, в частности, при некоторых условиях можно импортировать Hive UDF
* согласно документации, функции на C++ работают на несколько порядков быстрее
* возможно создавать UDF и UDAF (то есть агрегационные), нельзя создавать UDTF (табличные) или оконные
* Impala не поддерживает вложенные типы данных в пользовательских функциях (ни в качестве входных, ни в качестве выходных параметров)
* Impala не поддерживает timestamp, char и varchar в качестве входных или выходных параметров пользовательских функций
* условия, при которых можно импортировать Hive UDF
  + можно импортировать только скалярные Hive UDF (UDAF нельзя)
  + все типы данных в Hive UDF должны поддерживаться Impala

# Часто встречающиеся вопросы

* Ошибка вида AnalysisException: Failed to load metadata for table: 'p\_datalake.balance\_agr' CAUSED BY: TableLoadingException: Unrecognized table type for table: p\_datalake.balance\_agr

- формат таблицы не является допустимым для Impala

* Ошибка вида Disk I/O error: Failed to open HDFS file hdfs://bda31/user/u\_m0weu/db/operation\_tran/date\_part=20120313/000000\_0 Error(2): No such file or directory Root cause: RemoteException: File does not exist: /user/u\_m0weu/db/operation\_tran/date\_part=20120313/000000\_0

- файловая структура таблицы изменилась, нужно выполнить refresh