**Создать переменную**

You need to use the special hiveconf for variable substitution. e.g.

hive> set CURRENT\_DATE='2012-09-16';

hive> select \* from foo where day >= ${hiveconf:CURRENT\_DATE}

similarly, you could pass on command line:

% hive -hiveconf CURRENT\_DATE='2012-09-16' -f test.hql

**Показать скрипт создания указанной таблицы**

show create table ess\_bdm\_kfl.v\_prodrecret;

**Показать план выполнения запроса**

explain select \* from v\_prodrecret;

**Запросить первые 5 строк таблицы с добавлением псевдостолбцов: смещение блока внутри файла и файл, в котором хранится строка**

select BLOCK OFFSET INSIDE FILE, INPUT FILE NAME, p.\* from ess\_bdm\_kfl\_aux.d\_gregor\_big p limit 5; -

Создание структуры таблицы для секционированной таблицы

|  |
| --- |
| CREATE TABLE internal\_pkap\_kksb.RGP\_VERTICAL\_MARKING\_PART (cust\_id double, ogrn string, inn string, kpp string, cust\_name string, program\_code\_id double,  attribute\_code\_id double, website\_attribute\_name string, plan\_attribute\_value string, fact\_attribute\_value string, commentary string, curr\_data string)  PARTITIONED BY (SYSTEM\_CODE STRING); |

**Вставка в секционированную таблицу**

|  |
| --- |
| INSERT OVERWRITE TABLE internal\_pkap\_kksb.RGP\_VERTICAL\_MARKING\_PART PARTITION(SYSTEM\_CODE)  SELECT cust\_id, ogrn, inn, kpp, cust\_name, program\_code\_id, attribute\_code\_id, website\_attribute\_name,  plan\_attribute\_value, fact\_attribute\_value, commentary, curr\_data, SYSTEM\_CODE  from internal\_pkap\_kksb.rgp\_vertical\_marking; |

**Сброс патриций в Hive**

|  |
| --- |
| set hive.msck.path.validation=ignore;  MSCK REPAIR TABLE ИМЯ\_ТАБЛИЦЫ; |

**При этом параметре корректно отображаются строки с символом переноса строки**

|  |
| --- |
| set hive.query.result.fileformat=SequenceFile |

При этом параметре корректно отрабатывает запрос с JOIN

|  |
| --- |
| set hive.auto.convert.join=false; |

**Запускаем под spark**

|  |
| --- |
| set hive.execution.engine = spark;  set hive.vectorized.execution.enabled = true;  set hive.vectorized.execution.reduce.enabled = true;  set spark.executor.memory = 20g;  set spark.serializer = org.apache.spark.serializer.KryoSerializer;  set spark.executor.instances = 20;  set spark.yarn.executor.cores = 5; |

Выставляем параметры чтобы партицировать таблицу

|  |
| --- |
| set hive.exec.reducers.max=5000;  set mapreduce.map.memory.mb=10480;  set mapreduce.reduce.memory.mb=10480;  set mapred.map.child.java.opts=-Xmx10g;  set mapreduce.map.java.opts=-Xmx10g;  set mapred.reduce.slowstart.completed.maps=0.80;  set tasktracker.http.threads=80;  set hive.enforce.bucketing = true;  set hive.exec.dynamic.partition=true;  set hive.exec.dynamic.partition.mode=nonstrict;  set hive.exec.max.dynamic.partitions=10000;  set hive.exec.max.dynamic.partitions.pernode=10000;  set hive.auto.convert.join=false; |

**Из bigint в дату**

|  |
| --- |
| from\_unixtime(cast(report\_date/1000 as bigint),'yyyy-MM-dd') |

**Из bigint корректно в date Impala**

|  |
| --- |
| --Запрос дает разный результат в Hive и Impala  --Дело в том, что Impala в отличии от Hive не преобразует автоматически (без настройки) GMT в GMT+3, делаем это сами.  select 1469998800000,  from\_unixtime(cast(1469998800000/1000 as bigint),'yyyy-MM-dd HH:mm:ss'),  from\_utc\_timestamp(from\_unixtime(cast(1469998800000/1000 as bigint),'yyyy-MM-dd HH:mm:ss'), 'EEST') --из GMT0 в GMT+3 |

**Милисекунды**

|  |
| --- |
| -- Строка должна быть в стандартном формате yyyy-MM-dd hh:mm:ss.SSS  -- Умножить на 1000 для перевода секунд (double) в милисекунды (bigint)  select double(timestamp('2019-02-27 10:08:11.464'))\*1000  --1551251291464  select to\_utc\_timestamp(1551251291464, 'UTC') -- если передать целое число, оно будет обработано как милисекунды, дробное - как секунды  --2019-02-27 10:08:11.464    -- Если формат строки отличается от стандартного yyyy-MM-dd hh:mm:ss.SSS  -- его можно преобразовать в стандартный регулярным выражением:  select timestamp((regexp\_replace('2019.02.27 10:08:11.464', '(\\d{4})\\.(\\d{2})\\.(\\d{2}) (\\d{2})\\:(\\d{2})\\:(\\d{2})\\.(\\d{3})', '$1-$2-$3 $4:$5:$6.$7')))  --2019-02-27 10:08:11.464 |

**Календарь**

|  |
| --- |
| create table team\_business\_services.bibs\_coa\_bal\_gregor\_dt stored as parquet as  select cast(date\_add('2014-07-01', a.pos) as timestamp) as gregor\_dt  from ( select posexplode(split(repeat(',', day(last\_day(date'2014-07-01')) - 1), ',')) ) a; |

Если запрос падает с ошибкой "physical memory limit"

|  |
| --- |
| set mapred.max.split.size=512000000;  set mapreduce.input.fileinputformat.split.maxsize=512000000; |

**Unpivot**

|  |
| --- |
| -- преобразование из множества колонок-атрибутов в key-value хранилище  SELECT  t1.bvd\_id,  t1.quarter,  t2.key,  t2.value  FROM  sbx\_t\_sbi.leadgen\_index\_by\_1\_2\_3\_ilina t1  LATERAL VIEW  explode(  MAP(  'receiver\_bank\_name', receiver\_bank\_name,  'aggregated\_amount\_to\_russia', aggregated\_amount\_to\_russia,  'incoming\_aggregated\_volume', incoming\_aggregated\_volume,  'number\_of\_aggregated\_outgoing\_transactions', number\_of\_aggregated\_outgoing\_transactions  )  ) t2 AS KEY, VALUE |

**Pivot**

|  |
| --- |
| -- из key-value хранилища сделать много колонок-атрибутов  select  b.id,  b.code,  concat\_ws('',b.p) as p,  concat\_ws('',b.q) as q,  concat\_ws('',b.r) as r,  concat\_ws('',b.t) as t  from  (  select id, code,  collect\_list(a.group\_map['p']) as p,  collect\_list(a.group\_map['q']) as q,  collect\_list(a.group\_map['r']) as r,  collect\_list(a.group\_map['t']) as t  from (  select  id,  code,  map(key, value) as group\_map  from  test\_sample  ) a  group by  a.id,  a.code  ) b; |

Работа со структурой

|  |
| --- |
| create table t  (  i int  ,s1 struct<id:int,birthday:date,fname:string>  ,s2 struct<id:int,lname:string>  )  ;    insert into t  select 1  ,named\_struct('id',333,'birthday',date '1941-10-13','fname','Paul')  ,named\_struct('id',444,'lname','Simon')  ;    insert into t  select 2  ,named\_struct('id',777,'birthday',date '1941-11-05','fname','Art')  ,named\_struct('id',888,'lname','Garfunkel')  ;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  select \* from t  ;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  +-----+---------------------------------------------------+--------------------------------+  | t.i | t.s1 | t.s2 |  +-----+---------------------------------------------------+--------------------------------+  | 1 | {"id":333,"birthday":"1941-10-13","fname":"Paul"} | {"id":444,"lname":"Simon"} |  | 2 | {"id":777,"birthday":"1941-11-05","fname":"Art"} | {"id":888,"lname":"Garfunkel"} |  +-----+---------------------------------------------------+--------------------------------+  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  select i  ,i1.\*  ,i2.\*    from t  lateral view inline (array(s1)) i1  lateral view inline (array(s2)) i2  ;  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  +---+-------+-------------+----------+-------+-----------+  | i | i1.id | i1.birthday | i1.fname | i2.id | i2.lname |  +---+-------+-------------+----------+-------+-----------+  | 1 | 333 | 1941-10-13 | Paul | 444 | Simon |  | 2 | 777 | 1941-11-05 | Art | 888 | Garfunkel |  +---+-------+-------------+----------+-------+-----------+    -- доступ к данным в структуре    select eg.inn, eg.svnaimul.naimulsokr, eg.svnaimul.naimulpoln,  eg.svadresul.adresrf.region.tipregion,eg.svadresul.adresrf.region.NaimRegion,  eg.svadresul.adresrf.indeks, eg.svadresul.adresrf.gorod.TipGorod, eg.svadresul.adresrf.gorod.NaimGorod,  eg.svadresul.adresrf.ulica.TipUlica, eg.svadresul.adresrf.ulica.NaimUlica, eg.svadresul.adresrf.Dom,  eg.svadresul.adresrf.Korpus, eg.svadresul.adresrf.Kvart  from egrul EG  join fns\_inn4 FI  where eg.inn = FI.inn |

**Встать в другую очередь**

|  |
| --- |
| set mapred.job.queue.name=root.internal----; |