# **ЗАДАНИЕ**

Поработать с Cassandra через консоль. Протестировать инсерты, селекты с разными ключами. Работать в keyspace lesson7. Там можно создать свои таблички.  
Когда cassandra станет понятна, поработать с ней через Spark. Проверить пушит ли спарк фильтры в касандру.

# **Cassandra в консоли**

Запускаем mobaXterm

Создаем сессию с подключеним к worker-2

Выполняем команды:

*/cassandra/bin/cassandra #(находясь на worker2 - запускает базу, если она упала)*

*/cassandra/bin/cqlsh 10.0.0.18*

Создаем схему и таблицу

CREATE KEYSPACE mmingalov06

WITH REPLICATION = {

'class' : 'SimpleStrategy', 'replication\_factor' : 1 } ;

CREATE TABLE mmingalov06.animals

(id int,

name text,

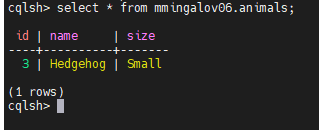
size text,

primary key (id));

*insert into mmingalov06.animals (id, name, size)*

*values (3, 'Hedgehog', 'Small');*

*select \* from mmingalov06.animals;*



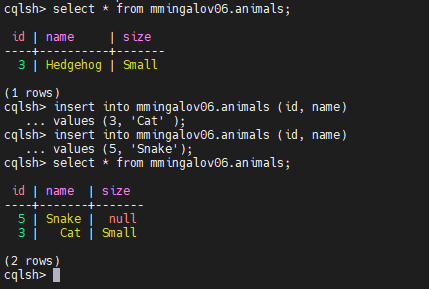
*#апдейт записи 3*

*insert into mmingalov06.animals (id, name)*

*values (3, 'Cat' );*

*insert into mmingalov06.animals (id, name)*

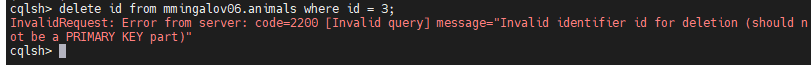
*values (5, 'Snake');*



Мы видим, что UPDATE произошел. Вернее, Cassandra выдала в Select оба значения по ключу 3, но вывела нам правильное значение (последнее, обновленное)

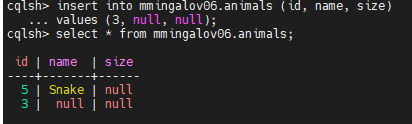
*#не сработает*

*delete id from mmingalov06.animals where id = 3;*



*insert into mmingalov06.animals (id, name, size)*

*values (3, null, null);*



# **Cassandra и pyspark**

Открываем отдельную вкладку

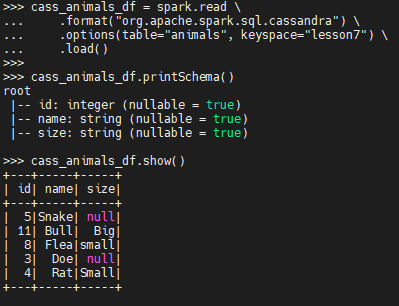
Запускаем команды:

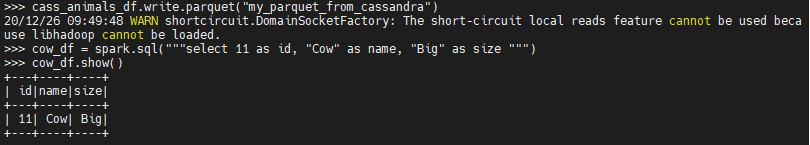
*export SPARK\_KAFKA\_VERSION=0.10*

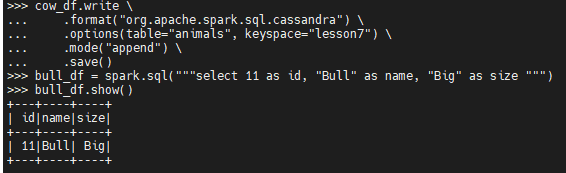
*/spark2.4/bin/pyspark --packages org.apache.spark:spark-sql-kafka-0-10\_2.11:2.4.5,com.datastax.spark:spark-cassandra-connector\_2.11:2.4.2 --driver-memory 512m --driver-cores 1 --master local[1]*

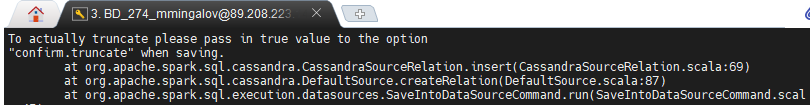
**from** pyspark.sql **import** SparkSession  
**from** pyspark.sql.types **import** StructType, StringType, IntegerType, TimestampType  
**from** pyspark.sql **import** functions **as** F

**def** explain(self, extended=**True**):  
 **if** extended:  
 print(self.\_jdf.queryExecution().toString())  
 **else**:  
 print(self.\_jdf.queryExecution().simpleString())  
  
cass\_animals\_df = spark.read \  
 .format(**"org.apache.spark.sql.cassandra"**) \  
 .options(table=**"animals"**, keyspace=**"lesson7"**) \  
 .load()  
  
cass\_animals\_df.printSchema()  
cass\_animals\_df.show()

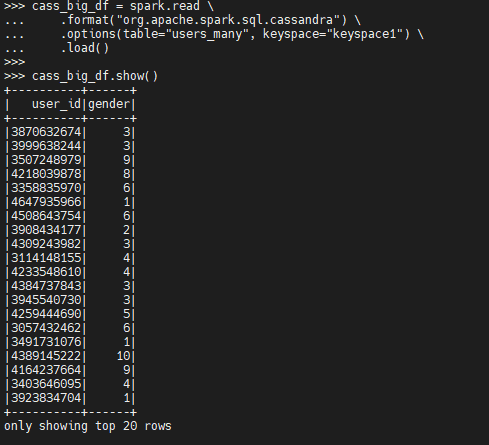
  
  
cass\_animals\_df.write.parquet(**"my\_parquet\_from\_cassandra"**)  
  
  
cow\_df = spark.sql(**"""select 11 as id, "Cow" as name, "Big" as size """**)  
cow\_df.show()

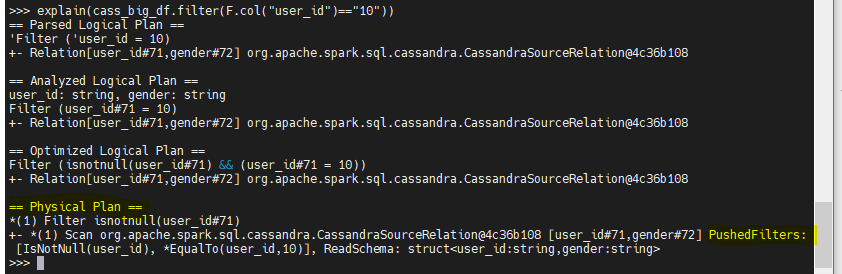
  
  
cow\_df.write \  
 .format(**"org.apache.spark.sql.cassandra"**) \  
 .options(table=**"animals"**, keyspace=**"lesson7"**) \  
 .mode(**"append"**) \  
 .save()  
  
*#айди такой же, имя другое*bull\_df = spark.sql(**"""select 11 as id, "Bull" as name, "Big" as size """**)  
bull\_df.show()

  
  
bull\_df.write \  
 .format(**"org.apache.spark.sql.cassandra"**) \  
 .options(table=**"animals"**, keyspace=**"lesson7"**) \  
 .mode(**"append"**) \  
 .save()

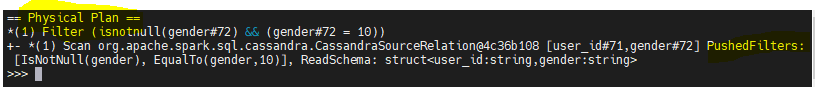
**OVERWRITE**  
*#выдаст предупреждение и попросит дополнительный параметр из за overwrite*bull\_df.write \  
 .format(**"org.apache.spark.sql.cassandra"**) \  
 .options(table=**"animals"**, keyspace=**"lesson7"**) \  
 .mode(**"overwrite"**) \  
 .save()  
  


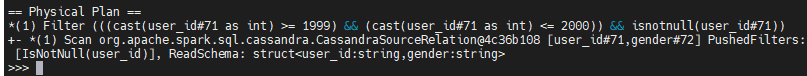
*#теперь читаем большой большой датасет по ключу*cass\_big\_df = spark.read \  
 .format(**"org.apache.spark.sql.cassandra"**) \  
 .options(table=**"users\_many"**, keyspace=**"keyspace1"**) \  
 .load()  
  
cass\_big\_df.show()



*#выполняем запрос попадая в ключ*cass\_big\_df.filter(F.col(**"user\_id"**)==**"10"**).show()  
*#выполняем запрос не попадая в ключ*cass\_big\_df.filter(F.col(**"gender"**)==**"1"**).show() *#быстро из за предела в 20*cass\_big\_df.filter(F.col(**"gender"**)==**"1"**).count() *#медленно  
  
#Наблюдаем на pushedFilter в PhysicalPlan*explain(cass\_big\_df.filter(F.col(**"user\_id"**)==**"10"**))  


explain(cass\_big\_df.filter(F.col(**"gender"**)==**"10"**))

  
  
*#between не передается в pushdown*cass\_big\_df.createOrReplaceTempView(**"cass\_df"**)  
sql\_select = spark.sql(**"""  
select \*   
from cass\_df  
where user\_id between 1999 and 2000  
"""**)  
*#проверяем, что user id не попал в pushedFilter*explain(sql\_select)



sql\_select.show() *#медленно  
  
#in передается в pushdown*sql\_select = spark.sql(**"""  
select \*   
from cass\_df  
where user\_id in (3884632855,3562535987)  
"""**)  
  
*#проверяем, что user id попал в pushedFilter*explain(sql\_select)

  
sql\_select.show() *#быстро*

