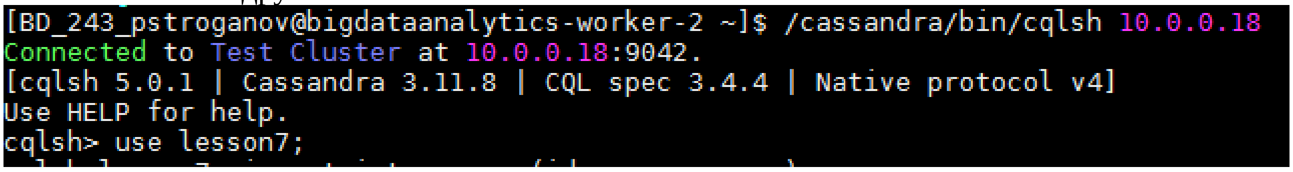
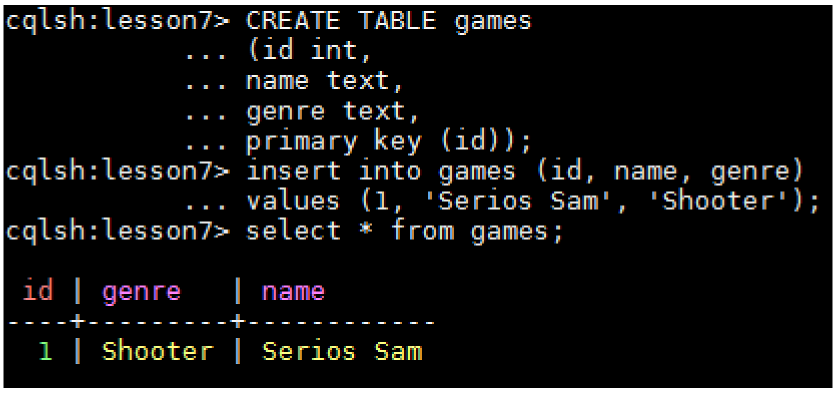
1. Зашел в кассандру.



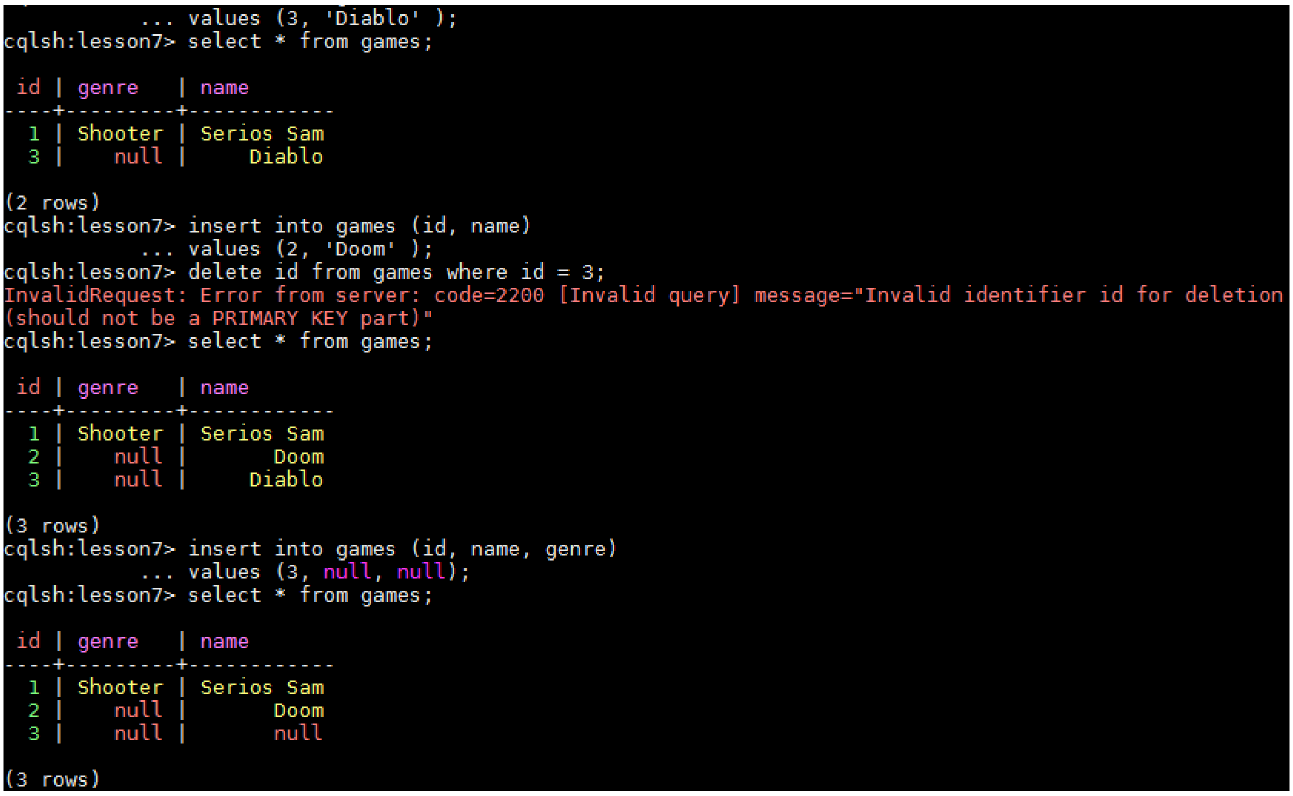
```
[BD_243_pstroganov@bigdataanalytics-worker-2 ~]$ /cassandra/bin/cqlsh 10.0.0.18
Connected to Test Cluster at 10.0.0.18:9042.
[cqlsh 5.0.1 | Cassandra 3.11.8 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use lesson7;
```

2. Повторил что делали на уроке.



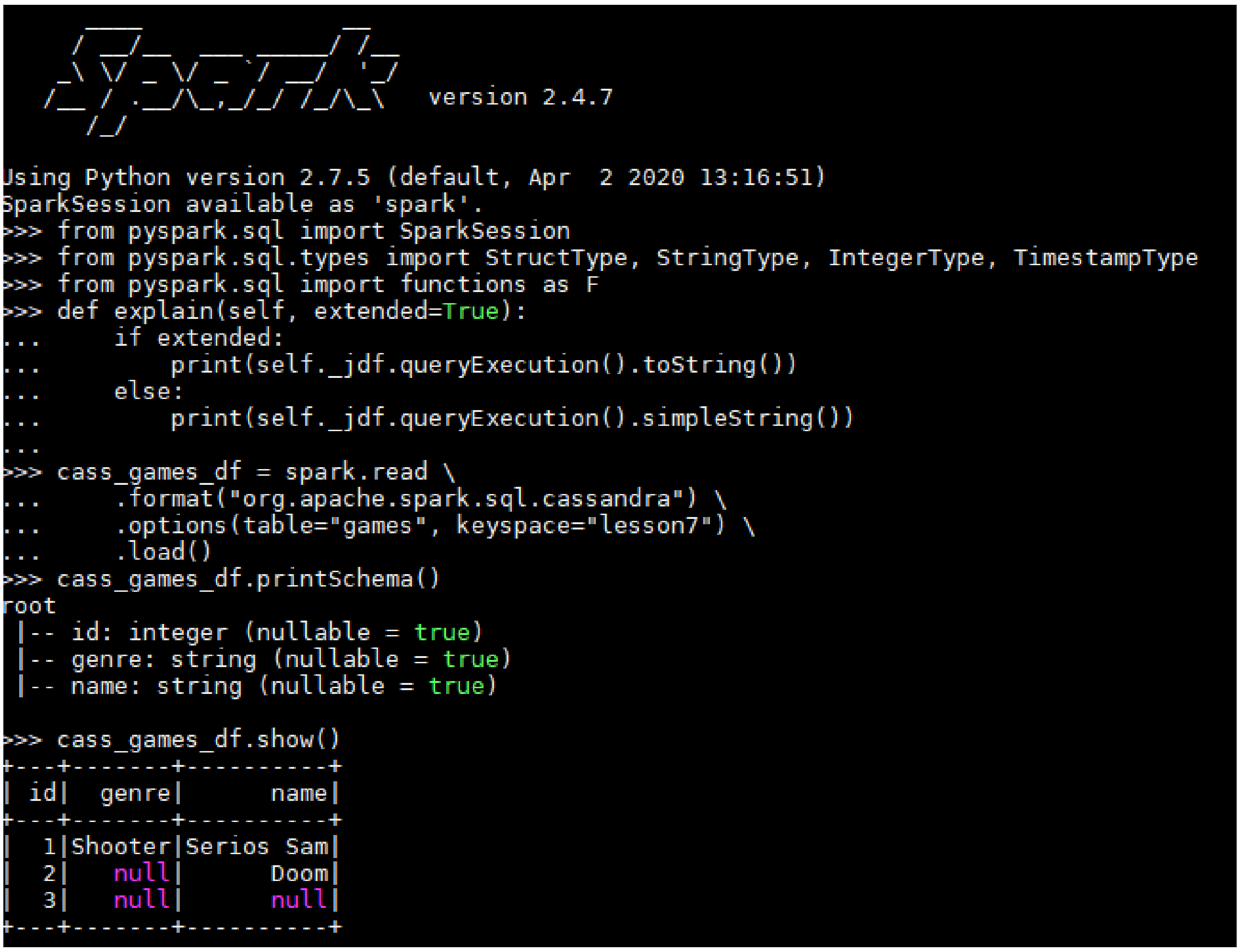
```
cqlsh:lesson7> CREATE TABLE games
           ... (id int,
           ... name text,
           ... genre text,
           ... primary key (id));
cqlsh:lesson7> insert into games (id, name, genre)
           ... values (1, 'Serios Sam', 'Shooter');
cqlsh:lesson7> select * from games;

 id | genre   | name
----+---------+------------
  1 | Shooter | Serios Sam
```

Через cqlsh:

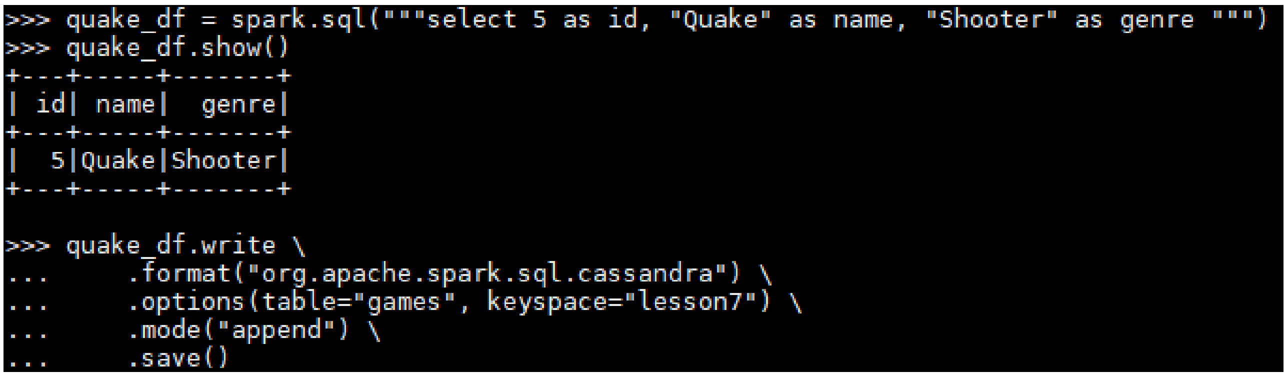


```
           ... values (3, 'Diablo' );
cqlsh:lesson7> select * from games;

 id | genre   | name
----+---------+------------
  1 | Shooter | Serios Sam
  3 |    null |     Diablo

(2 rows)
cqlsh:lesson7> insert into games (id, name)
           ... values (2, 'Doom' );
cqlsh:lesson7> delete id from games where id = 3;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid identifier id for deletion
(should not be a PRIMARY KEY part)"
cqlsh:lesson7> select * from games;

 id | genre   | name
----+---------+------------
  1 | Shooter | Serios Sam
  2 |    null |       Doom
  3 |    null |     Diablo

(3 rows)
cqlsh:lesson7> insert into games (id, name, genre)
           ... values (3, null, null);
cqlsh:lesson7> select * from games;

 id | genre   | name
----+---------+------------
  1 | Shooter | Serios Sam
  2 |    null |       Doom
  3 |    null |       null

(3 rows)
```

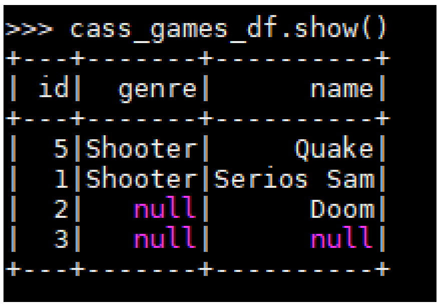
Далее из спарка:



```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.7
      /_/

Using Python version 2.7.5 (default, Apr  2 2020 13:16:51)
SparkSession available as 'spark'.
>>> 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_games_df = spark.read \
...     .format("org.apache.spark.sql.cassandra") \
...     .options(table="games", keyspace="lesson7") \
...     .load()
>>> cass_games_df.printSchema()
root
 |-- id: integer (nullable = true)
 |-- genre: string (nullable = true)
 |-- name: string (nullable = true)

>>> cass_games_df.show()
+---+-------+----------+
| id|  genre|      name|
+---+-------+----------+
|  1|Shooter|Serios Sam|
|  2|   null|      Doom|
|  3|   null|      null|
+---+-------+----------+
```

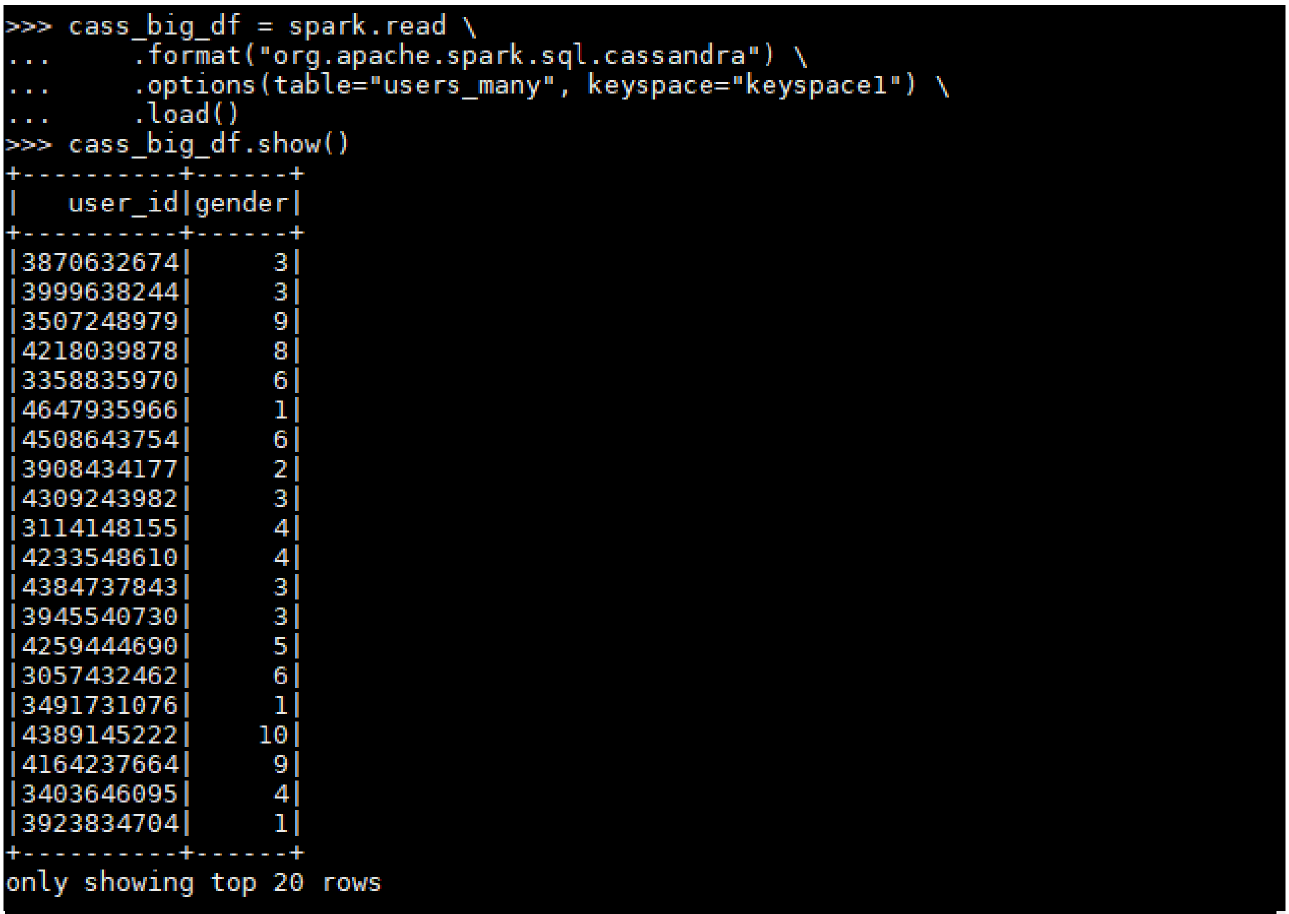


```
>>> quake_df = spark.sql("""select 5 as id, "Quake" as name, "Shooter" as genre """)
>>> quake_df.show()
+---+-----+-------+
| id| name|  genre|
+---+-----+-------+
|  5|Quake|Shooter|
+---+-----+-------+

>>> quake_df.write \
...     .format("org.apache.spark.sql.cassandra") \
...     .options(table="games", keyspace="lesson7") \
...     .mode("append") \
...     .save()
```

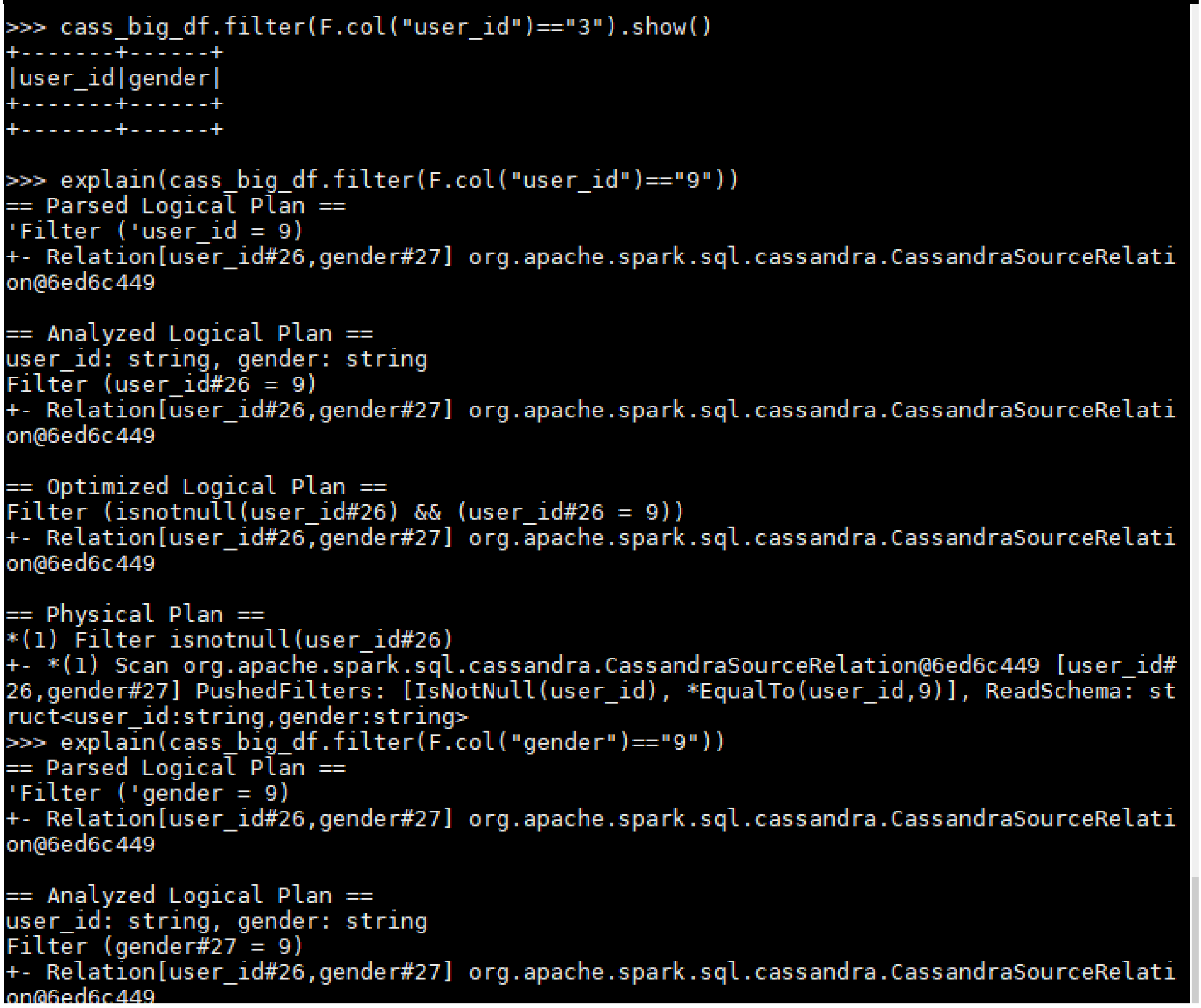


```
>>> cass_games_df.show()
+---+-------+----------+
| id|  genre|      name|
+---+-------+----------+
|  5|Shooter|     Quake|
|  1|Shooter|Serios Sam|
|  2|   null|      Doom|
|  3|   null|      null|
+---+-------+----------+
```

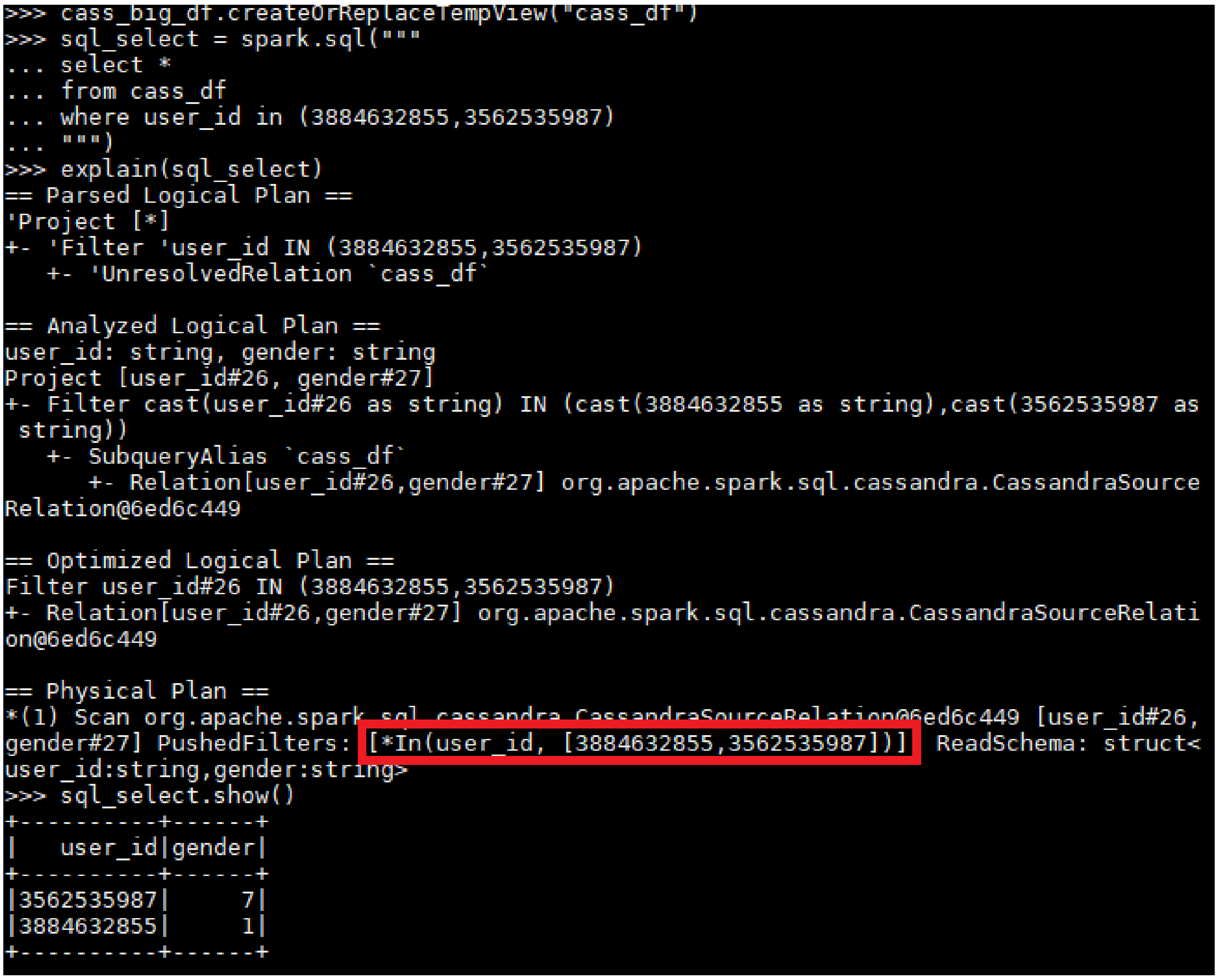
Планы запросов и фильтры:



```
>>> cass_big_df = spark.read \
...     .format("org.apache.spark.sql.cassandra") \
...     .options(table="users_many", keyspace="keyspace1") \
...     .load()
>>> cass_big_df.show()
+----------+------+
|   user_id|gender|
+----------+------+
|3870632674|     3|
|3999638244|     3|
|3507248979|     9|
|4218039878|     8|
|3358835970|     6|
|4647935966|     1|
|4508643754|     6|
|3908434177|     2|
|4309243982|     3|
|3114148155|     4|
|4233548610|     4|
|4384737843|     3|
|3945540730|     3|
|4259444690|     5|
|3057432462|     6|
|3491731076|     1|
|4389145222|    10|
|4164237664|     9|
|3403646095|     4|
|3923834704|     1|
+----------+------+
only showing top 20 rows
```



```
>>> cass_big_df.filter(F.col("user_id")=="3").show()
+-------+------+
|user_id|gender|
+-------+------+
+-------+------+

>>> explain(cass_big_df.filter(F.col("user_id")=="9"))
== Parsed Logical Plan ==
'Filter ('user_id = 9)
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449

== Analyzed Logical Plan ==
user_id: string, gender: string
Filter (user_id#26 = 9)
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449

== Optimized Logical Plan ==
Filter (isnotnull(user_id#26) && (user_id#26 = 9))
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449

== Physical Plan ==
*(1) Filter isnotnull(user_id#26)
+- *(1) Scan org.apache.spark.sql.cassandra.CassandraSourceRelation@6ed6c449 [user_id#
26,gender#27] PushedFilters: [IsNotNull(user_id), *EqualTo(user_id,9)], ReadSchema: st
ruct<user_id:string,gender:string>
>>> explain(cass_big_df.filter(F.col("gender")=="9"))
== Parsed Logical Plan ==
'Filter ('gender = 9)
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449

== Analyzed Logical Plan ==
user_id: string, gender: string
Filter (gender#27 = 9)
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449
```



```
>>> cass_big_df.createOrReplaceTempView("cass_df")
>>> sql_select = spark.sql("""
... select *
... from cass_df
... where user_id in (3884632855,3562535987)
... """)
>>> explain(sql_select)
== Parsed Logical Plan ==
'Project [*]
+- 'Filter 'user_id IN (3884632855,3562535987)
   +- 'UnresolvedRelation `cass_df`

== Analyzed Logical Plan ==
user_id: string, gender: string
Project [user_id#26, gender#27]
+- Filter cast(user_id#26 as string) IN (cast(3884632855 as string),cast(3562535987 as
 string))
   +- SubqueryAlias `cass_df`
      +- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSource
Relation@6ed6c449

== Optimized Logical Plan ==
Filter user_id#26 IN (3884632855,3562535987)
+- Relation[user_id#26,gender#27] org.apache.spark.sql.cassandra.CassandraSourceRelati
on@6ed6c449

== Physical Plan ==
*(1) Scan org.apache.spark.sql.cassandra.CassandraSourceRelation@6ed6c449 [user_id#26,
gender#27] PushedFilters: [*In(user_id, [3884632855,3562535987])], ReadSchema: struct<
user_id:string,gender:string>
>>> sql_select.show()
+----------+------+
|   user_id|gender|
+----------+------+
|3562535987|     7|
|3884632855|     1|
+----------+------+
```