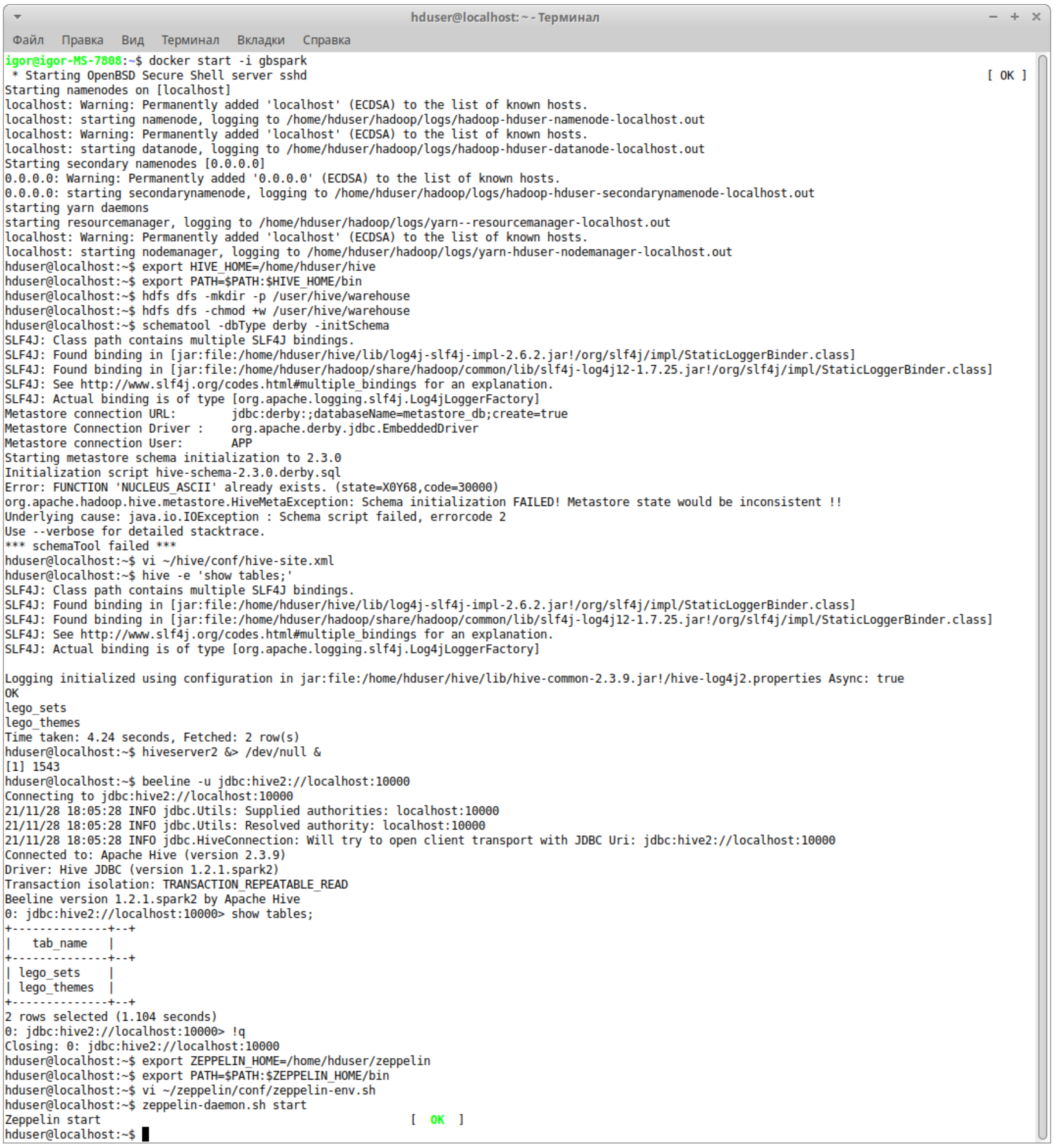
ПРАКТИЧЕСКОЕ ЗАДАНИЕ К УРОКУ 5

1. Установите Apache Zeppelin и настройте интеграцию с Apache Spark и ~~Apache Hive~~


hduser@localhost: ~ - Терминал

Файл Правка Вид Терминал Вкладки Справка

```
igor@igor-MS-7808:~$ docker start -i gbspark
 * Starting OpenBSD Secure Shell server sshd                                                                    [ OK ]
Starting namenodes on [localhost]
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: starting namenode, logging to /home/hduser/hadoop/logs/hadoop-hduser-namenode-localhost.out
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: starting datanode, logging to /home/hduser/hadoop/logs/hadoop-hduser-datanode-localhost.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /home/hduser/hadoop/logs/hadoop-hduser-secondarynamenode-localhost.out
starting yarn daemons
starting resourcemanager, logging to /home/hduser/hadoop/logs/yarn--resourcemanager-localhost.out
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
localhost: starting nodemanager, logging to /home/hduser/hadoop/logs/yarn-hduser-nodemanager-localhost.out
hduser@localhost:~$ export HIVE_HOME=/home/hduser/hive
hduser@localhost:~$ export PATH=$PATH:$HIVE_HOME/bin
hduser@localhost:~$ hdfs dfs -mkdir -p /user/hive/warehouse
hduser@localhost:~$ hdfs dfs -chmod +w /user/hive/warehouse
hduser@localhost:~$ schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :    org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:       APP
Starting metastore schema initialization to 2.3.0
Initialization script hive-schema-2.3.0.derby.sql
Error: FUNCTION 'NUCLEUS_ASCII' already exists. (state=X0Y68,code=30000)
org.apache.hadoop.hive.metastore.HiveMetaException: Schema initialization FAILED! Metastore state would be inconsistent !!
Underlying cause: java.io.IOException : Schema script failed, errorcode 2
Use --verbose for detailed stacktrace.
*** schemaTool failed ***
hduser@localhost:~$ vi ~/hive/conf/hive-site.xml
hduser@localhost:~$ hive -e 'show tables;'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hduser/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hduser/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/hduser/hive/lib/hive-common-2.3.9.jar!/hive-log4j2.properties Async: true
OK
lego_sets
lego_themes
Time taken: 4.24 seconds, Fetched: 2 row(s)
hduser@localhost:~$ hiveserver2 &> /dev/null &
[1] 1543
hduser@localhost:~$ beeline -u jdbc:hive2://localhost:10000
Connecting to jdbc:hive2://localhost:10000
21/11/28 18:05:28 INFO jdbc.Utils: Supplied authorities: localhost:10000
21/11/28 18:05:28 INFO jdbc.Utils: Resolved authority: localhost:10000
21/11/28 18:05:28 INFO jdbc.HiveConnection: Will try to open client transport with JDBC Uri: jdbc:hive2://localhost:10000
Connected to: Apache Hive (version 2.3.9)
Driver: Hive JDBC (version 1.2.1.spark2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 1.2.1.spark2 by Apache Hive
0: jdbc:hive2://localhost:10000> show tables;
+--------------+--+
|   tab_name   |
+--------------+--+
| lego_sets    |
| lego_themes  |
+--------------+--+
2 rows selected (1.104 seconds)
0: jdbc:hive2://localhost:10000> !q
Closing: 0: jdbc:hive2://localhost:10000
hduser@localhost:~$ export ZEPPELIN_HOME=/home/hduser/zeppelin
hduser@localhost:~$ export PATH=$PATH:$ZEPPELIN_HOME/bin
hduser@localhost:~$ vi ~/zeppelin/conf/zeppelin-env.sh
hduser@localhost:~$ zeppelin-daemon.sh start
Zeppelin start                                             [  OK  ]
hduser@localhost:~$
```

Переходим на http://localhost:8888:

НАСТРОЙКА ИНТЕРПРЕТАТОРА SPARK

spark.master **yarn-cluster**

2. Скачайте датасет Video Game Sales

sh
python -m pip install kaggle

sh

```
export KAGGLE_USERNAME=igortolstikov
export KAGGLE_KEY=fb99c3ad6bf931513798d91567033035
mkdir -p /home/hduser/videogame
cd /home/hduser/videogame
kaggle datasets files gregorut/videogamesales
kaggle datasets download gregorut/videogamesales
```

```
sh
cd /home/hduser/videogame
unzip videogamesales.zip
rm videogamesales.zip
ls -la
```

```
sh
hdfs dfs -put /home/hduser/videogame /user/hduser
hdfs dfs -ls /user/hduser/videogame
```

```
sh
hdfs dfs -cat /user/hduser/videogame/vgsales.csv | head
```

```
spark.sql
CREATE TABLE if not exists videogame
        using csv
        options (
                path "/user/hduser/videogame/vgsales.csv",
                header true,
                interSchema true
        );
```

```
spark.sql
SELECT * FROM videogame LIMIT 10;
```

3. Выведите самую продаваемую игру за всё время

```
spark.sql
SELECT * FROM videogame ORDER BY Rank asc LIMIT 1;
```

4. Какая платформа самая популярная в каждом регионе (NA, EU, JP)?

```
spark.sql
WITH x as (
        SELECT
                Platform,
                sum(NA_Sales) NA_Sum,
                sum(EU_Sales) EU_Sum,
                sum(JP_Sales) JP_Sum
        FROM videogame
        GROUP BY Platform
)
(
        SELECT 'NA' as region, Platform
        FROM x
        ORDER BY NA_Sum desc
```

```
	LIMIT 1
)
union
(
	SELECT 'EU' as region, Platform
	FROM x
	ORDER BY EU_Sum desc
	LIMIT 1
)
union
(
	SELECT 'JP' as region, Platform
	FROM x
	ORDER BY JP_Sum desc
	LIMIT 1
)
```

5. Какой жанр популярен больше всего в каждом регионе (NA, EU, JP)?

```
spark.sql
WITH x as (
	SELECT
		Genre,
		sum(NA_Sales) NA_Sum,
		sum(EU_Sales) EU_Sum,
		sum(JP_Sales) JP_Sum
	FROM videogame
	GROUP BY Genre
)
(
	SELECT 'NA' as region, Genre
	FROM x
	ORDER BY NA_Sum desc
	LIMIT 1
)
union
(
	SELECT 'EU' as region, Genre
	FROM x
	ORDER BY EU_Sum desc
	LIMIT 1
)
union
(
	SELECT 'JP' as region, Genre
	FROM x
	ORDER BY JP_Sum desc
	LIMIT 1
)
```

6. Выведите самый популярный жанр на каждый год

```
spark.sql
WITH x as (
	SELECT
		Genre,
		Year,
		sum(Global_Sales) Global_Sales,
		row_number() over(partition by Year ORDER BY sum(Global_Sales) desc) as
row_number
	FROM videogame
	WHERE Year != 'N/A'
	GROUP BY Year, Genre
	ORDER BY Year, Global_Sales desc
)
SELECT Year, Genre
FROM x
WHERE row_number == 1
```