Introducing Spark SQL

Working with Hive


HIVE


HIVE


hadoop


HIVE


Spark


hadoop

In:
```
from pyspark.sql import SparkSession
spark_session = SparkSession\
                    .builder\
                    .enableHiveSupport()\
                    .appName("spark sql")\
                    .master("local")\
                    .getOrCreate()
```

In:
```
from pyspark.sql import SparkSession
spark_session = SparkSession\
                    .builder\
                    .enableHiveSupport()\
                    .appName("spark sql")\
                    .master("local")\
                    .getOrCreate()
```

spark_session.sql

spark_session.sql

In:
```
spark_session.sql("""
    show databases
""").toPandas()
```

Out:

	databaseName
0	default
1	web

spark_session.sql

In:
```
spark_session.sql("""
    show databases
""").toPandas()
```

Out:

	databaseName
0	default
1	web

In:
```
spark_session.sql("""
    show tables in web
""").toPandas()
```

Out:

	database	tableName	isTemporary
0	web	access_log	False

spark_session.catalog

spark_session.catalog

```
In: spark_session.catalog.listDatabases()
```

```
Out: [Database(name=u'default', description=u'Default Hive
database', locationUri=u'hdfs://mipt-master.atp-fivt.
org:8020/user/hive/warehouse'),
 Database(name=u'web', description=u'',
locationUri=u'hdfs://mipt-master.atp-fivt.org:8020/
user/hive/warehouse/web.db')]
```

spark_session.catalog

```
In: spark_session.catalog.listDatabases()
```

```
Out: [Database(name=u'default', description=u'Default Hive
database', locationUri=u'hdfs://mipt-master.atp-fivt.
org:8020/user/hive/warehouse'),
 Database(name=u'web', description=u'',
locationUri=u'hdfs://mipt-master.atp-fivt.org:8020/
user/hive/warehouse/web.db')]
```

```
In: spark_session.catalog.listTables("web")
```

```
Out: [Table(name=u'access_log', database=u'web',
description=None, tableType=u'MANAGED',
isTemporary=False)]
```

spark_session.catalog

```
In: spark_session.catalog.listDatabases()
```

```
Out: [Database(name= default, description= Default
     Hive database , locationUri=u'hdfs://mipt-master.
     atp-fivt.org:8020/user/hive/warehouse'),
      Database(name= web , description=u'',
     locationUri=u'hdfs://mipt-master.atp-fivt.org:8020/
     user/hive/warehouse/web.db')]
```

```
In: spark_session.catalog.listTables("web")
```

```
Out: [Table(name=u'access_log', database=u'web',
     description=None, tableType=u'MANAGED',
     isTemporary=False)]
```

In:
```
spark_session.sql("""
    select * from web.access_log
""").show(3)
```

```
+---------+-------------+---------------+--------------------+--------+--------------------+
|http_code|           ip|response_length|                time|     url|          user_agent|
+---------+-------------+---------------+--------------------+--------+--------------------+
|      200| 109.106.133.8|          21546|12/Dec/2015:01:31...|/id53821|Mozilla/5.0 (Maci...|
|      200|  46.31.82.254|           8777|12/Dec/2015:01:31...|/id33929|Mozilla/5.0 (Wind...|
|      200|193.124.254.46|           8731|12/Dec/2015:01:31...|/id35754|Mozilla/4.0 (comp...|
+---------+-------------+---------------+--------------------+--------+--------------------+
only showing top 3 rows
```

spark_session.sql

spark_session.sql

Transformation ? Action

spark_session.sql

Transformation ? Action

create table as select

```
In: geoip_df

Out: DataFrame[ip: string, code: string, country: string]
```

```
In: geoip_df
```

```
Out: DataFrame[ip: string, code: string, country: string]
```

```
In: geoip_df.createTempView("geoip")
```

```
In: geoip_df
```

```
Out: DataFrame[ip: string, code: string, country: string]
```

```
In: geoip_df.createTempView("geoip")
```

```
In: spark_session.catalog.listTables("web")
```

```
Out: [Table(name=u'access_log', database=u'web', description=None,
tableType=u'MANAGED', isTemporary=False),
 Table(name=u'geoip', database=None, description=None,
tableType=u'TEMPORARY', isTemporary=True)]
```

```
In: geoip_df
```

```
Out: DataFrame[ip: string, code: string, country: string]
```

```
In: geoip_df.createTempView("geoip")
```

```
In: spark_session.catalog.listTables("web")
```

```
Out: [Table(name=u'access_log', database=u'web', description=None,
tableType=u'MANAGED', isTemporary=False),
 Table(name=u'geoip', database=None, description=None,
tableType=u'TEMPORARY', isTemporary=True)]
```

In:
```
spark_session.sql("""
    create table web.geoip as
    select * from geoip
""")
```

Out:
```
DataFrame[]
```

In:
```
spark_session.catalog.listTables("web")
```

Out:
```
[Table(name=u'access_log', database=u'web', description=None, tableType=u'MANAGED',
isTemporary=False),
 Table(name=u'geoip', database=u'web', description=None, tableType=u'MANAGED',
isTemporary=False),
 Table(name=u'geoip', database=None, description=None, tableType=u'TEMPORARY',
isTemporary=True)]
```

In:
```
spark_session.sql("""
    select * from web.geoip
""").show(3)
```

Out:
```
+---------------+----+------------------+
|             ip|code|           country|
+---------------+----+------------------+
|194.120.126.123|  NL|       Netherlands|
| 94.126.119.173|  FR|            France|
|  193.46.74.166|  RU|Russian Federation|
+---------------+----+------------------+
only showing top 3 rows
```

What have we learned:

- How to connect to hive
- How to check connections and browse databases
- How to get any table from hive into spark dataframe
- How to store any table to hive