Spark SQL: Takeaways

Syntax

- Registering an RDD as a temporary table:

```
from pyspark.sql import SQLContext
sqlCtx = SQLContext(sc)
df = sqlCtx.read.json("census_2010.json")
df.registerTempTable('census2010')
```

- Returning a list of tables:

```
tables = sqlCtx.tableNames()
```

- Querying a table in Spark:

```
sqlCtx.sql('select age from census2010').show()
```

- Calculating summary statistics for a DataFrame:

```
query = 'select males,females from census2010'
   sqlCtx.sql(query).describe().show()
```

Concepts

- Spark maintains a virtual database within a SQLContext object. This makes it possible to use Spark's SQL interface to query and interact with the data.
- Spark uses a type of SQL that is identical to SQLite to query a table.
- Spark SQL allows you to run join queries across data from multiple file types.
- Spark SQL supports the functions and operators from SQLite. Supported functions operators are as follows:
- COUNT()
- AVG()
- SUM()
- AND
- OR

Resources

- Spark SQL
- Purpose of Spark SQL