Data, BigData. Cloud.
Anaytics & Virtualizaton
HadoopExam.com

QuickTechie.com

SPARKSQL, DATAFRAME AND HIVECONTEXT

By www.HadoopExam.com

Note: These instructions should be used with the HadoopExam Apache Spark: Professional Trainings. Where it is executed and you can do hands on with trainer.

Spark SQL

⇒ Spark SQL can process data from.

- HDFS
- Cassandra
- HBase
- RDBMS

⇒ When you read data stored in HDFS using SparkSQL, you need to give/define some structure/schema.

⇒ So that, it make sense out of this data.

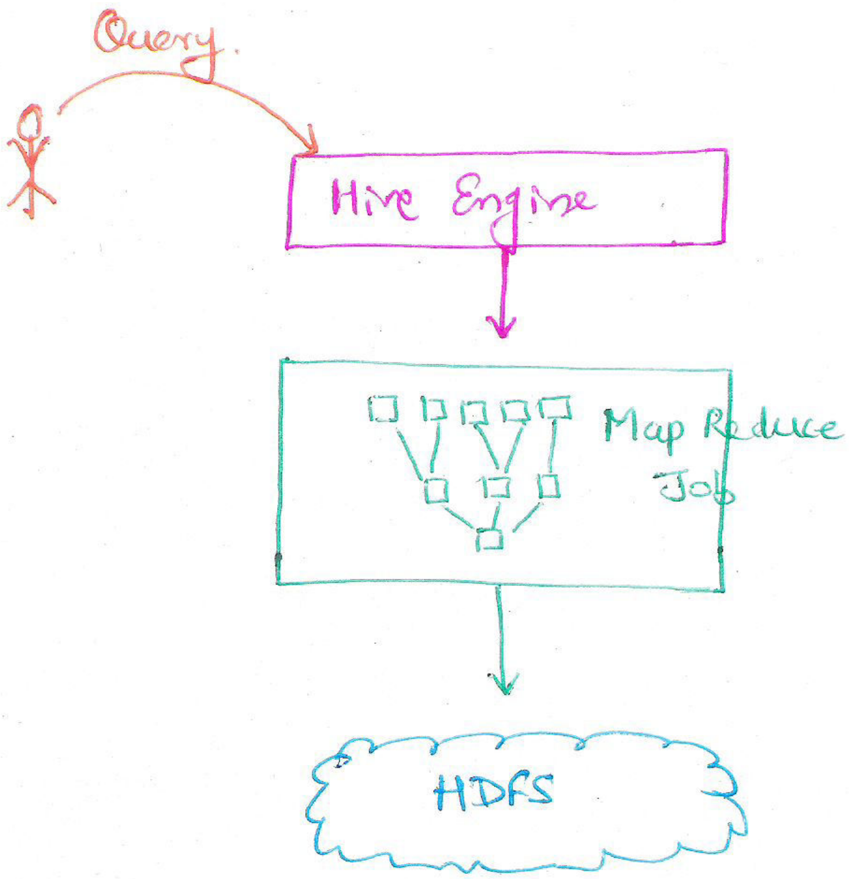
* History of Spark SQL (Not to get confused with existing all the components)

⇒ Spark 1.0 : There was a project called Shark

⇒ Shark was an attempt to make hive run on Spark.

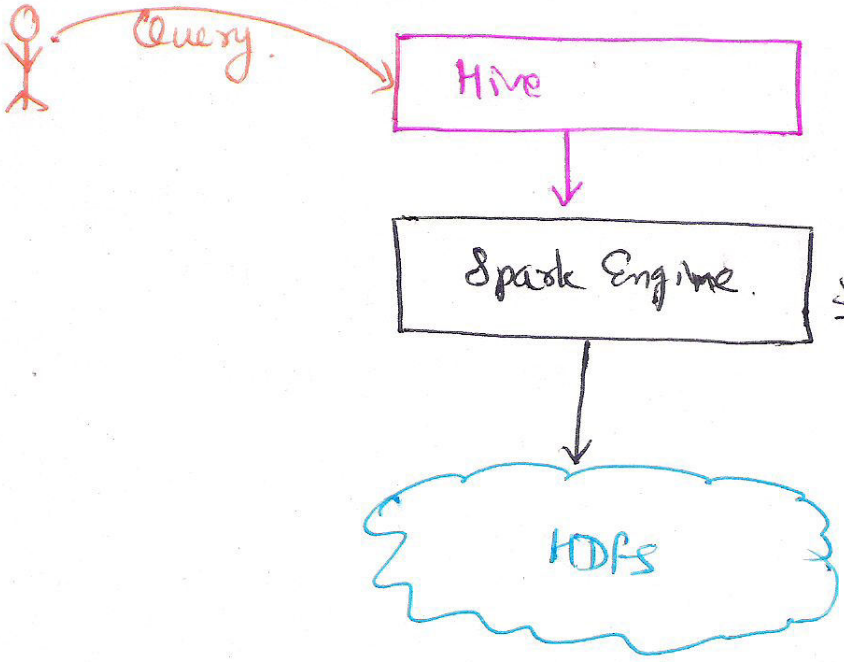
Hive: (Go through Hadoop Training for more detail)

⇒ Apache Hive is a Relational operation, which convert the SQL queries to MapReduce Job.



Hive Engine to Read data stored in HDFS.

=> Shark replaced the MapReduce, with Spark Engine. and retaining most of codebase.

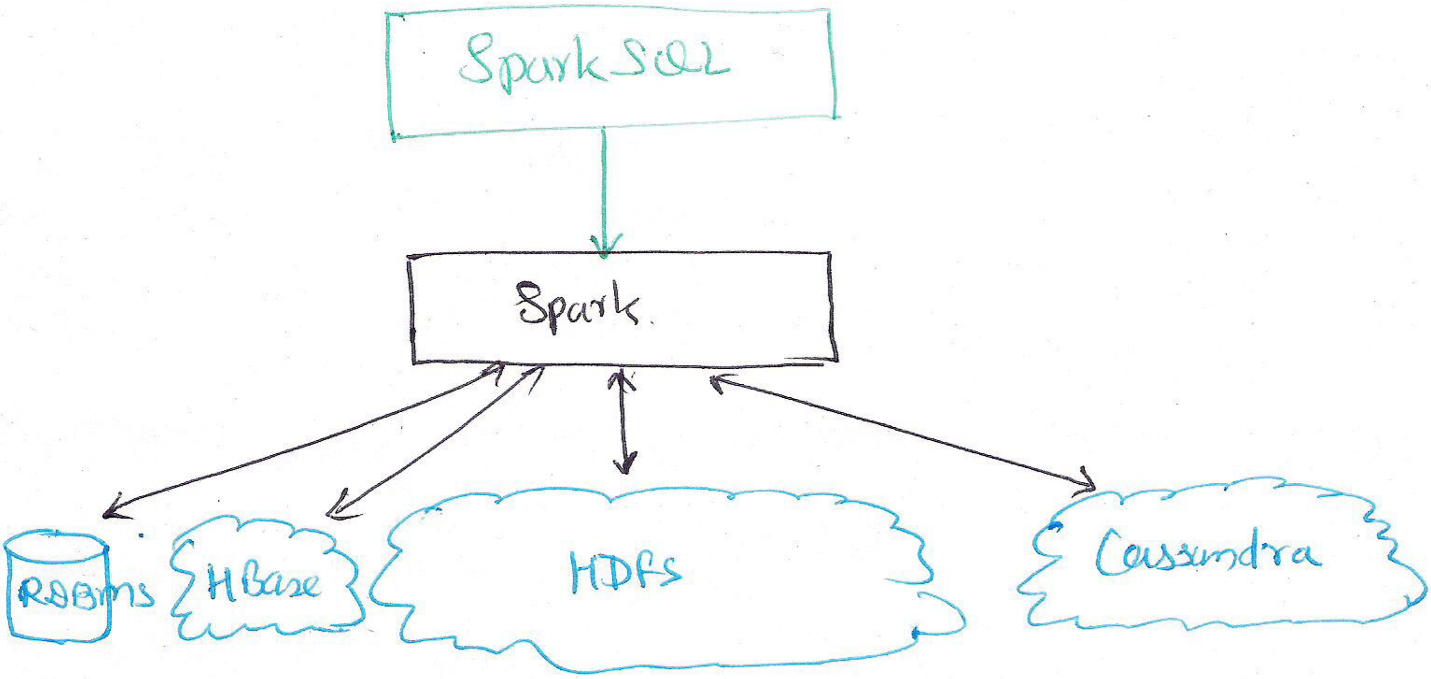


=> However, above architecture/replacement seems good for initial days.

⇒ Spark developers hit a roadblocks and could not optimize it any further.

⇒ Finally they decided to write SQL Engine from scratch.

⇒ So, now that is called : SparkSQL.



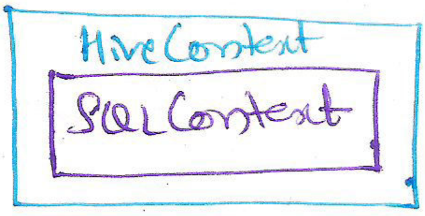
⇒ So now, there is nothing like Shark anymore, if you come across spark Shark, then it is outdated.

SparkSQL :

① Took care of all performance challenges.

② Also provides compatibility, with a new wrapper called HiveContext.

⇒ HiveContext was created on top of SQLContext



⇒ SparkSQL supports accessing data using

→ Standard SQL queries.

→ HiveQL :- Hive query language.

⇒ SparkSQL : Helps to create and run spark program faster.

→ It lets developers ~~less~~ write less code.

→ program to read less data.

⇒ Catalyst optimizer :- do many optimization for task' performance.

⇒ Dataframe :- SparkSQL uses a programming abstraction called Dataframes.

→ Dataframe is a distributed collection of data organized in named columns.

→ Dataframe is equivalent to a database table, but provides much finer level of optimization.

=> Dataframe API also ensures that spark ~~performance~~ performance is consistent across different language bindings (e.g. Python, Java, R, Scala)

RDD v/s Dataframes

=> RDD :- is an collection of objects with no idea about the format of underlying data.

[As we have seen in previous session, Employee RDD and cities RDD, does not have included column information, Developers must be aware about the structure in RDD]

=> Dataframe :- Have schema associated with them

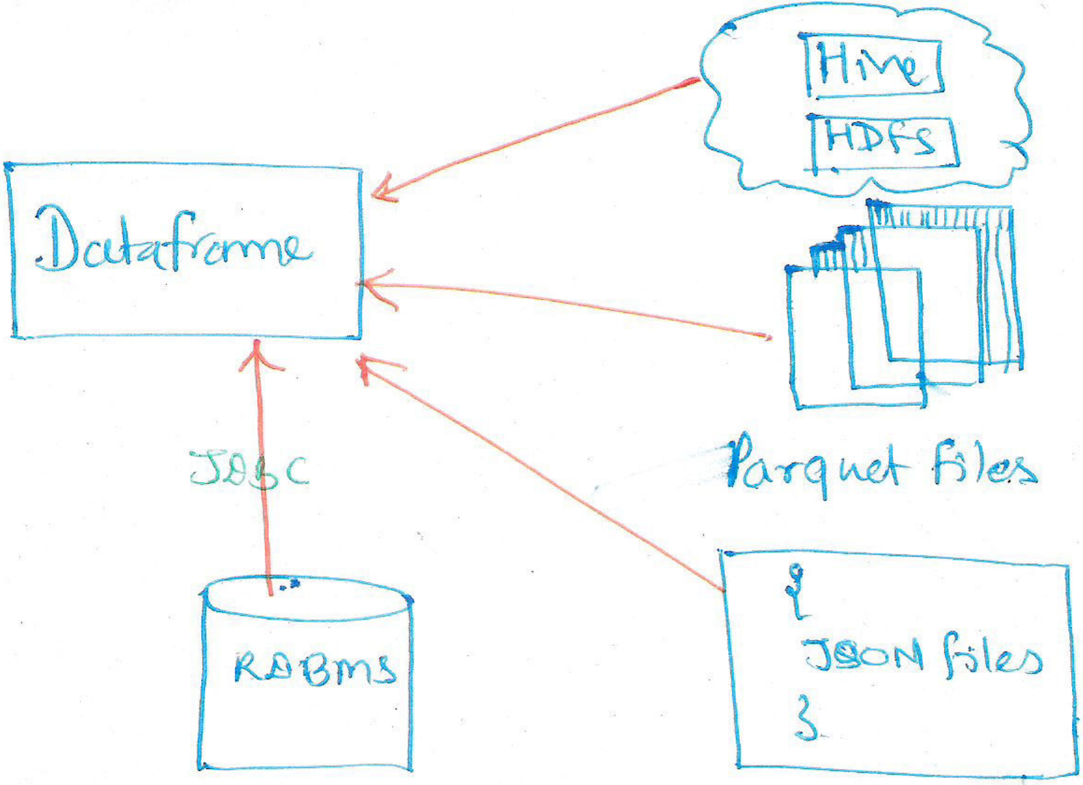
$$\boxed{\text{Dataframe} = \text{RDD} + \text{Schema}}$$

=> Schema RDD :- Upto Spark 1.2, there was a component that is called schema RDD.

SchemaRDD → Replaced by → Data frame

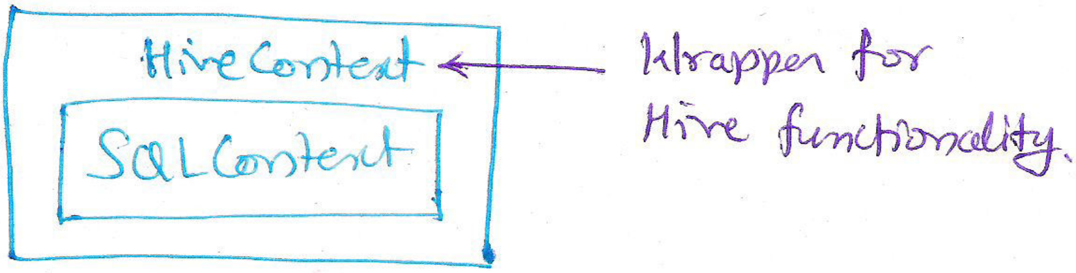
⇒ Dataframe is much Richer than SchemaRDD.

⇒ Dataframe also transparently load data from various sources.



⇒ Dataframe can be viewed as RDDs of row objects, allowing developers to call procedural API such as map.

⇒ Entry point for SparkSQL is SQLContext.



⇒ HiveContext is more tested than SQLContext.

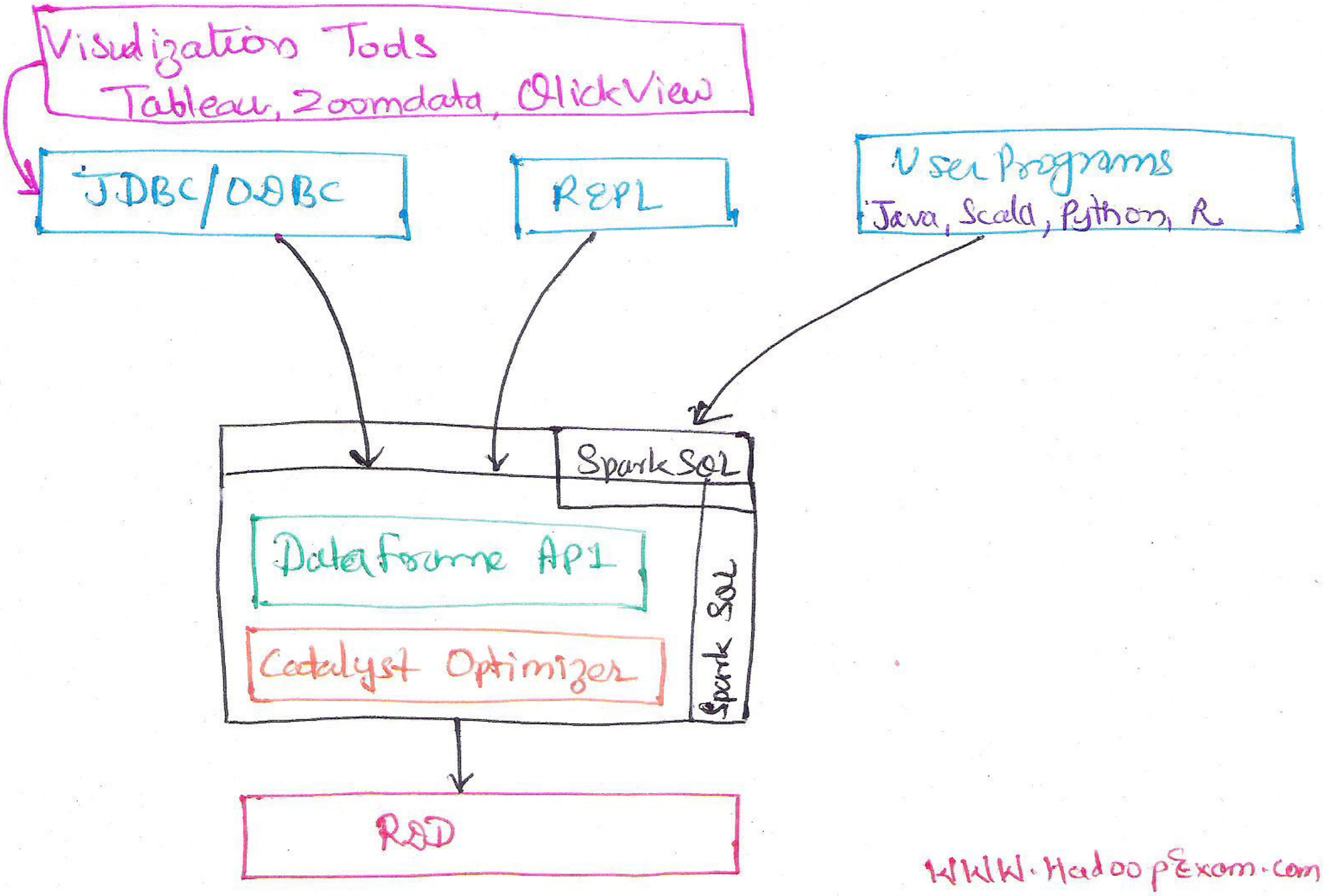
⇒ It is suggested, you always use HiveContext whether you are using Hive or not.

⇒ There are two ways to associate schema to RDDs, to create Dataframes

① Easier way is to leverage scala case classes.

② Programatically assign | specify schema for advanced needs.

→ It uses Java Reflection to deduce schema from case classes.



Catalyst * Optimizer : SparkSQL uses Catalyst Optimizer for query optimization with following goals.

1. Make adding new optimization techniques easy.
2. Enable external developers to extend the optimizer

⇒ SparkSQL uses catalyst's transformation framework in 4 phases.

1. Analyze a logical plan to resolve refrenes.
2. logical plan optimization.
3. Physical planning.
4. Code generation to compile the parts of the query to java bytecode.

[Every step is internal to SparkSQL]

We will be using

Does not require Hive setup.

1. Hive Context ⇒ (Entry point)
2. sqlContext
3. Dataframe API = (RDD + Schema)
   (Previously it was SchemaRDD)

① If you dont have an existing Hive Installation SparkSQL will create its own Hive metastore, in program works directory. (metastore-db)

② If you attempt to create tables using HiveQL's Create table statement (Not Create External table) they will be placed in the /user/hive/warehouse directory. on your filesystem.

=> If you want to work with Hive you have to use HiveContext.

=> Spark 1.5 is now ~~sp~~ support for Window function. and ability to acces Hive UDFs.

=> Window function can be used to solve quite complex problems, without going back and forth between RDDs and Dataframes.

=> HiveContext is required to start Thrift Server.

=> The biggest problem with HiveContext is that it comes with large dependencies.