Spark SQL - Intro and HIVE

» Apache HIVE is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale. Hive metastore (HMS) provides a central repository of metdata that can easily be analyzed to make informed, data driven decisions.

→ HIVE is built on apache hadoop and supports storage on S3, adls, gs etc. through hdfs

→ It allows users to read, write & manage petabytes of data using SQL

Features of Spark SQL

① HIVE Compatibility :- Can run unmodified HIVE queries on existing warehouses.
Spark SQL reuses the Hive frontend and metastore, giving you full compatibility with existing HIVE data, queries and UDFs

② Standard Connectivity :- Connect through JDBC or ODBC
Spark SQL includes a server mode with industry standard JDBC and ODBC connectivity.
Tableau, Zeomdata, QLIK → sensory, servers
↳ Ready-made Java based servers.

③ Scalability :- Use the same engine for both interactive and long queries.
→ Spark SQL takes advantage of the RDD model to support mid-query fault tolerance, letting it scale to large jobs too.
UDF :- Extensive feature of Spark SQL.
→ Plug in your own processing code and invoke it from a HIVE query.
Plain (UDF) :- I/P → single row, O/P → single row.
UDAF (User defined Aggregate Function) :- I/P → Multiple Rows, O/P → Single row. eg :- Count, max
UDTF (User-defined Table Generating Function) :- I/P → Single row, O/P → Multiple row.

classmate
Date
Page

Spark RDDs

RDD is a fundamental data structure of spark.
It is an immutable distributed collection of objects that can be stored in memory or disk across a cluster.
Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.
Parallel functional transformations (map, filter...)
Automatically rebuilt on failure
They can contain any type of Python, Java, or Scala objects including user-defined classes.

RDD is a fault tolerant collection of elements that can be operated on in parallel.
There are two ways to create RDDs:-
parallelizing an existing collection in the driver prog.
referencing a dataset in an external storage system such as shared file system, HDFS, HBase, or any data source offering a Hadoop Input format.

Spark makes use of RDD to achieve faster and efficient MapReduce operations.

classmate
Date
Page

Data Set and Data Frame.

- A distributed collection of data, which is organised into named columns.
- Conceptually, it is equivalent to relational tables with good organization techniques.
- A DataFrame can be constructed from an array of diff. sources such as Hive tables, Structured data files, external databases, or existing RDDs.
- This API was designed for modern Big data and data science app. taking inspiration from dataframe in R programming and pandas in python.

Dataframe

- Data is organized into named columns, like a table in a relational database.

Dataset :- a distributed collection of data.

- A new interface added in Spark 1.6.
- Static-typing and runtime type-safety.

Features of DataFrame.

- Ability to process the data in the size of kilobyte to petabytes on a single-node cluster to large cluster.
- Supports different data formats (Avro, csv, elastic search, and cassandra) and storage systems (HDFS, HIVE, tables, mysql, etc).
- State of art optimization and code generation through the spark SQL Catalyst optimizer (tree transformation framework).
- Can be easily integrated with all Big data tools and frameworks via Spark-Core.
- Provides API for python, Java, Scala, and R programming.

Spark SQL, the whole story

- Create and Run Spark Programs faster
1. Write less code
2. Read less data
3. Let the optimizer do the hard work.

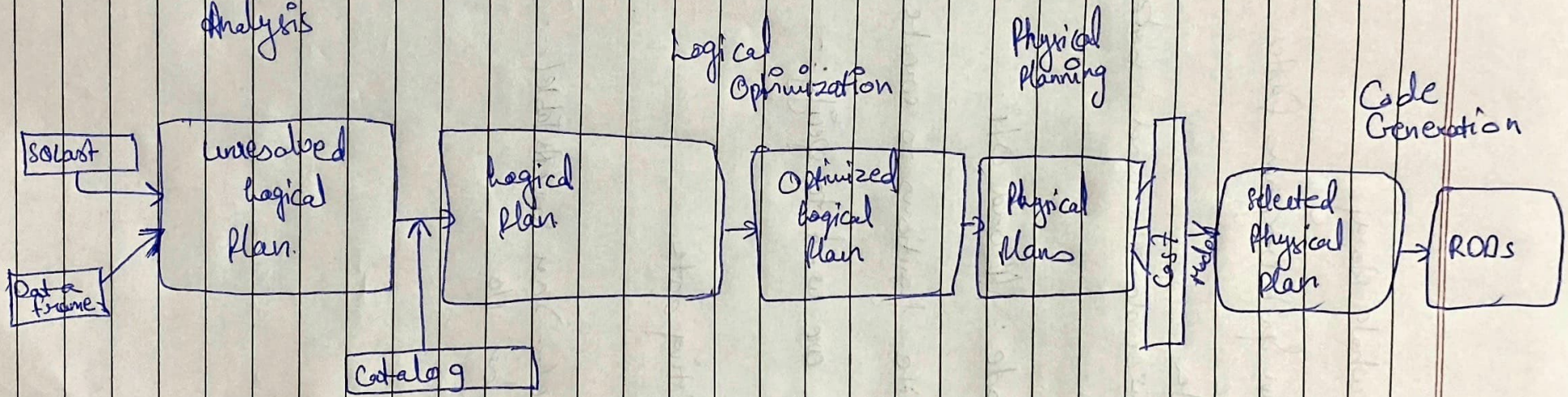
Write less Code: I/P and O/P.

- Read & write functions to create new builders for I/O
- Builder methods are used to specify:-
(i) format
(ii) partitioning
(iii) handling of existing data
(iv) and more.

- load(...), save(...) or saveAsTable(...) functions to create new builders for I/O.

1

Plan Optimization & Execution



DataFrames and SQL share the same optimization/execution pipeline.

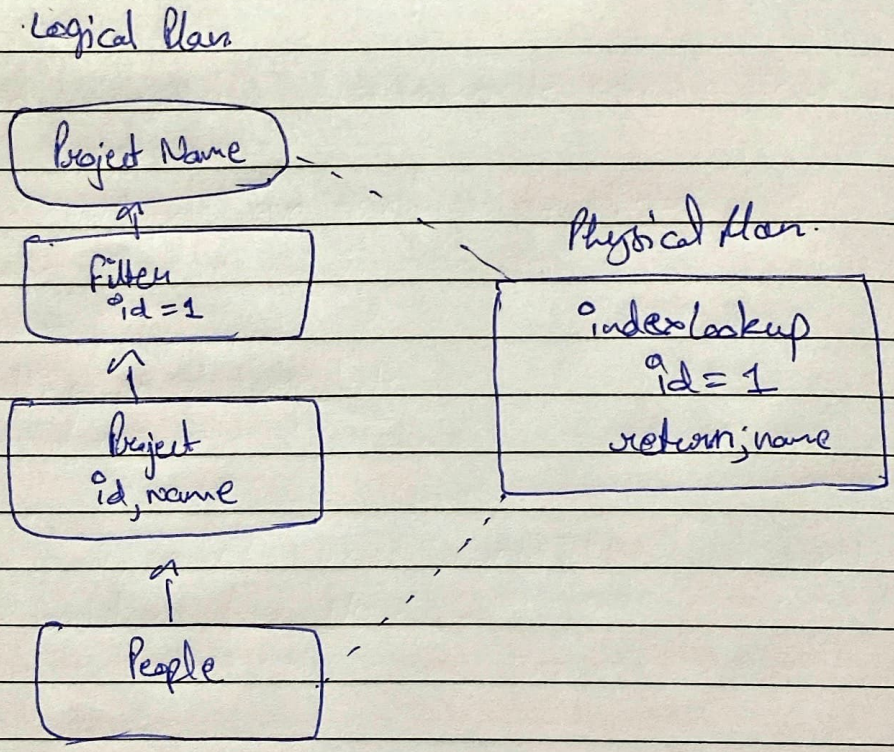
63

Optimized Execution

- Writing imperative code to optimize all possible patterns is hard.

⇒ Instead write simple rules:-

- each rule makes one change
- run many rules together to fixed point.



⇒ Use spark.sql() method and CREATE TABLE statement to create a table in Hive from Spark temporary view.

spark.sql("CREATE DATABASE IF NOT EXISTS ct")