
HIVE

What is **Apache Hive?**

Apache Hive is an open source data warehouse system built on top of Hadoop used for querying and analyzing large datasets stored in Hadoop files.
Initially, you have to write complex Map-Reduce jobs, but now with the help of the Hive, you just need to submit merely SQL queries. Hive is mainly targeted towards users who are comfortable with SQL. Hive use language called HiveQL (HQL), which is similar to SQL. HiveQL automatically translates SQL-like queries into MapReduce jobs.
Hive abstracts the complexity of Hadoop. The main thing to notice is that there is no need to learn java for Hive.
The Hive generally runs on your workstation and converts your SQL query into a series of jobs for execution on a Hadoop cluster. Apache Hive organizes data into tables. This provides a means for attaching the structure to data stored in HDFS.

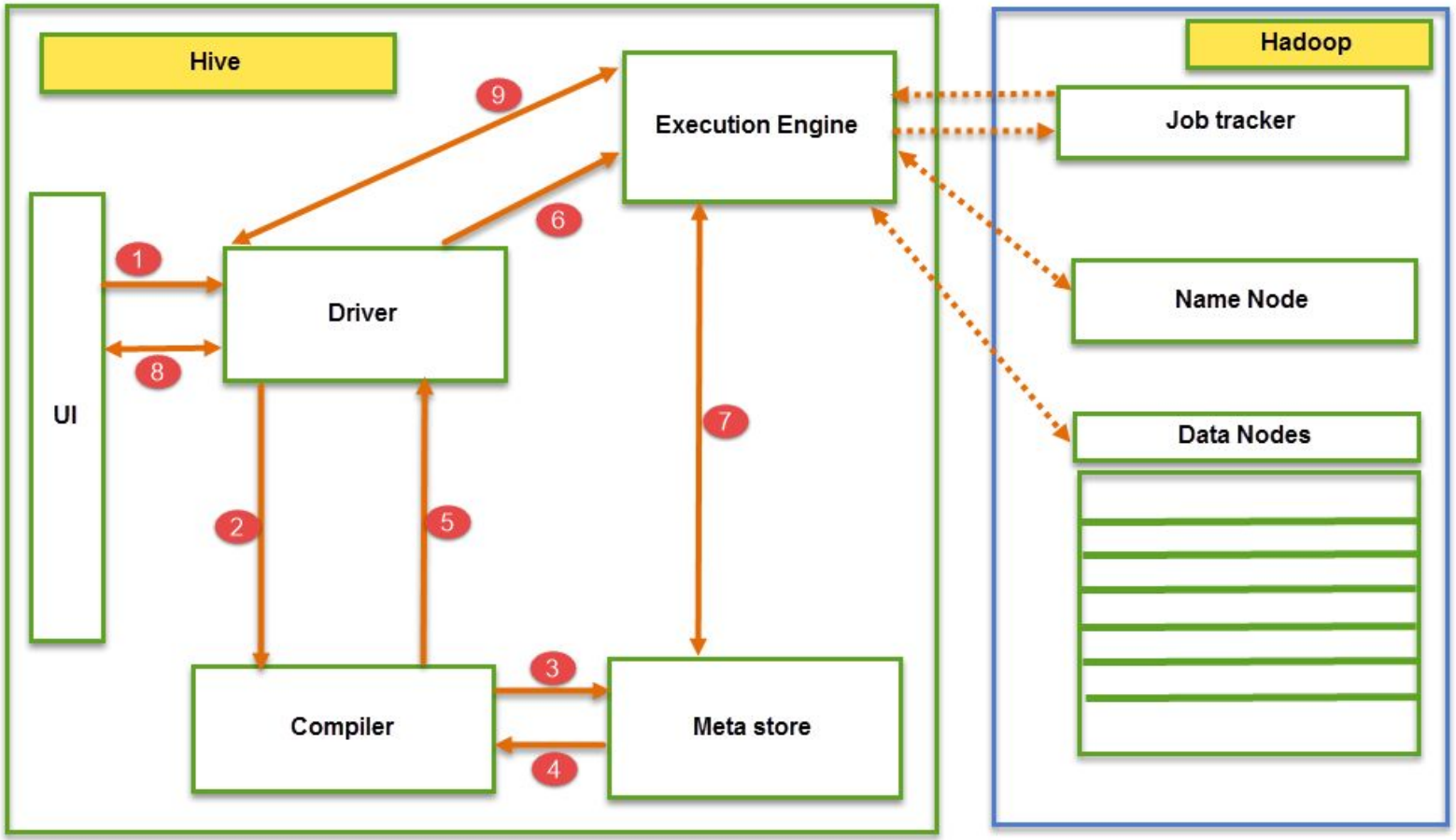
Why **Apache Hive?**

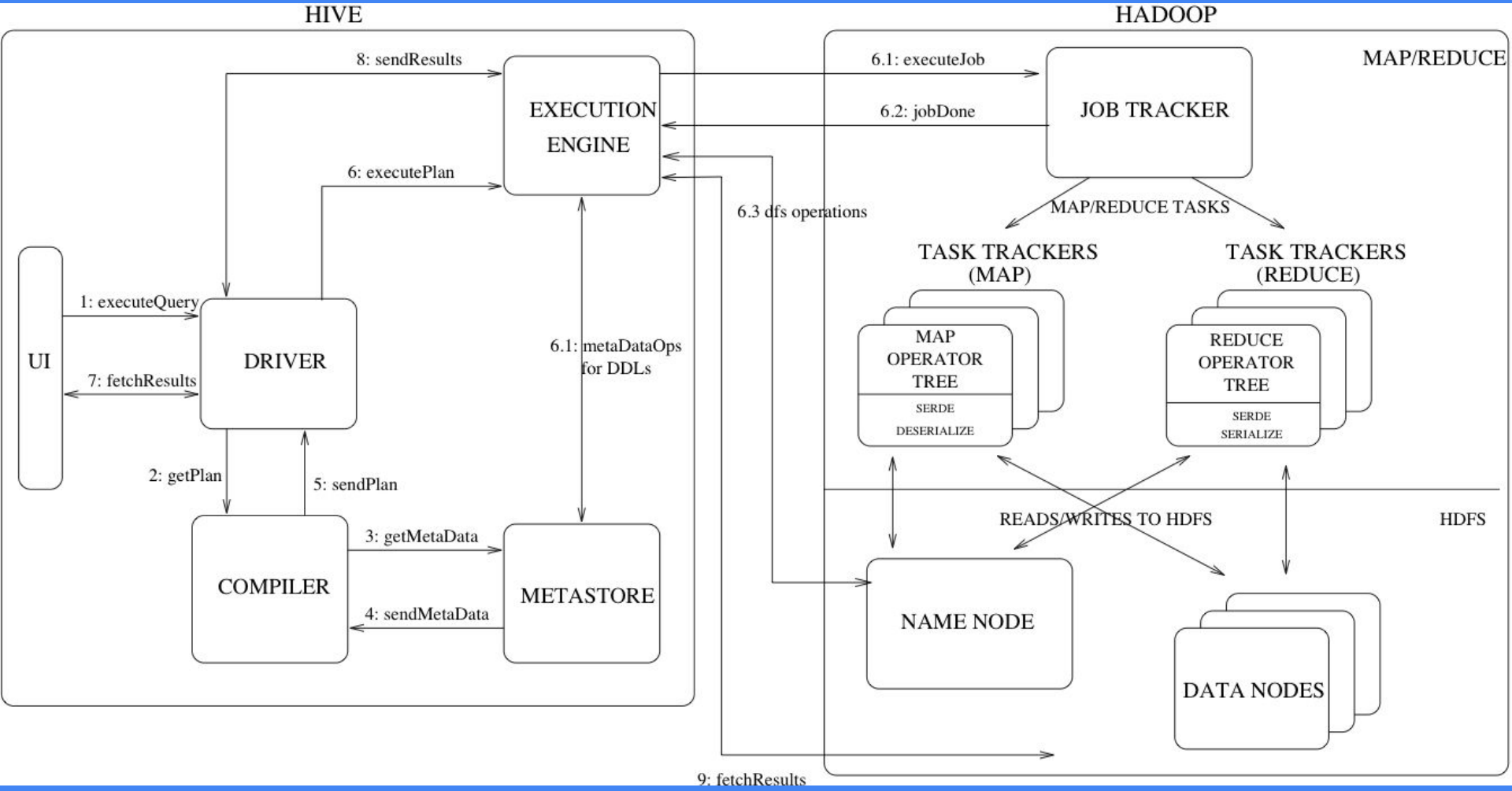
Facebook had faced a lot of challenges before the implementation of **Apache Hive**. Challenges like the size of the data being generated increased or exploded, making it very difficult to handle them. The traditional RDBMS could not handle the pressure. As a result, Facebook was looking out for better options. To overcome this problem, Facebook initially tried using MapReduce. But it has difficulty in programming and mandatory knowledge in SQL, making it an impractical solution. Hence, Apache Hive allowed them to overcome the challenges they were facing.

With **Apache Hive**, they are now able to perform the following:

- Schema flexibility and evolution
- Tables can be portioned and bucketed
- Apache Hive tables are defined directly in the HDFS
- JDBC/ODBC drivers are available

Hive Architecture


Hive
Execution Engine
Hadoop
Job tracker
Name Node
Data Nodes
UI
Driver
Compiler
Meta store
1
2
3
4
5
6
7
8
9


HIVE
HADOOP
MAP/REDUCE
8: sendResults
6.1: executeJob
6.2: jobDone
EXECUTION ENGINE
JOB TRACKER
6: executePlan
6.3 dfs operations
MAP/REDUCE TASKS
TASK TRACKERS (MAP)
TASK TRACKERS (REDUCE)
UI
1: executeQuery
7: fetchResults
DRIVER
6.1: metaDataOps for DDLs
MAP OPERATOR TREE
SERDE
DESERIALIZE
REDUCE OPERATOR TREE
SERDE
SERIALIZE
2: getPlan
5: sendPlan
READS/WRITES TO HDFS
HDFS
3: getMetaData
COMPILER
METASTORE
4: sendMetaData
NAME NODE
DATA NODES
9: fetchResults

Hive Metastore

It stores metadata for each of the tables like their schema and location.
Hive also includes the partition metadata.
This helps the driver to track the progress of various data sets distributed over the cluster.
It stores the data in a traditional RDBMS format.
Hive metadata helps the driver to keep a track of the data and it is highly crucial.
Backup server regularly replicates the data which it can retrieve in case of data loss.

Hive Driver

It acts like a controller which receives the HiveQL statements.
The driver starts the execution of the statement by creating sessions.
It monitors the life cycle and progress of the execution.
Driver stores the necessary metadata generated during the execution of a HiveQL statement.
It also acts as a collection point of data or query result obtained after the Reduce operation.

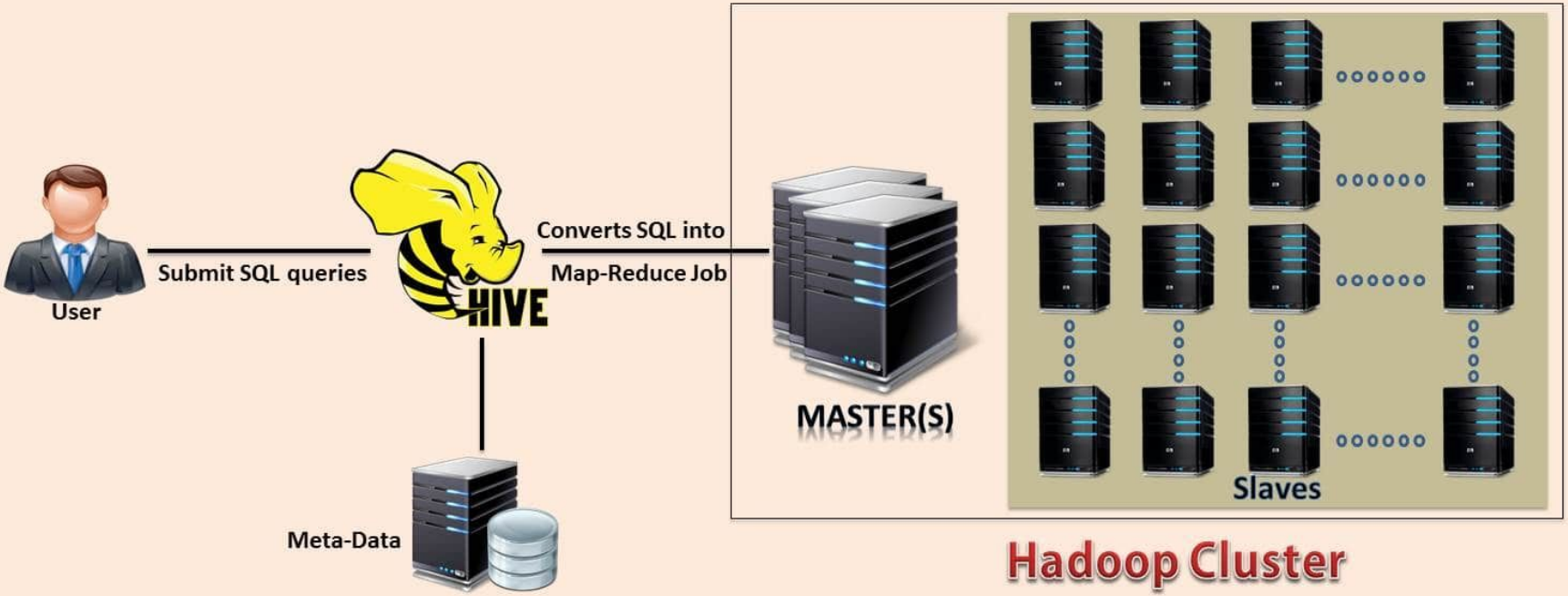
Hive Compiler

It performs the compilation of the HiveQL query.
This converts the query to an execution plan.
The plan contains the tasks. It also contains steps needed to be performed by the MapReduce to get the output as translated by the query.
The compiler in Hive converts the query to an Abstract Syntax Tree (AST). First, check for compatibility and compile-time errors, then converts the AST to a Directed Acyclic Graph (DAG).

Hive Execution Engine

Execution Engine used to communicate with Hadoop daemons such as Name node, Data nodes, and job tracker to execute the Hive query on top of Hadoop file system. It executes the execution plan created by the compiler.

The differents execution engine in Hive are: **mr** (Map Reduce, default), **tez** (Tez execution, for Hadoop 2 only), or **spark** (Spark execution, for Hive 1.1. 0 onward).


User
Submit SQL queries
HIVE
Converts SQL into
Map-Reduce Job
MASTER(S)
Slaves
Meta-Data
Hadoop Cluster

Hive Shell

The shell is the primary way with the help of which we interact with the Hive; we can issue our commands or queries in HiveQL inside the Hive shell.

Hive Shell is almost similar to MySQL Shell.

It is the command line interface for Hive.

In Hive Shell users can run HQL queries.

HiveQL is also case-insensitive (except for string comparisons) same as SQL.

```
hive> show tables;
OK
emp_details
emp_details_partitioned
hcataglo_table
hcatalog_table
olympics
sample_07
spark_olympic
userlog
Time taken: 0.017 seconds, Fetched: 8 row(s)
hive>
    >
    >
    > select * from spark_olympic;
OK
Time taken: 0.084 seconds
hive>
    > select * from spark_olympic limit 5;
OK
Michael Phelps  23      United States   2008    8/24/2008       Swimming        8       0       0       8
Michael Phelps  19      United States   2004    8/29/2004       Swimming        6       0       2       8
Michael Phelps  27      United States   2012    8/12/2012       Swimming        4       2       0       6
Natalie Coughlin        25      United States   2008    8/24/2008       Swimming        1       2       3       6
Aleksey Nemov   24      Russia  2000    10/1/2000       Gymnastics      2       1       3       6
Time taken: 0.082 seconds, Fetched: 5 row(s)
hive>
```

Hive Features

- Hive provides data summarization, query, and analysis in much easier manner.
- Hive supports external tables which make it possible to process data without actually storing in HDFS.
- Apache Hive fits the low-level interface requirement of Hadoop perfectly.
- It also supports partitioning of data at the level of tables to improve performance.
- Hive has a rule based optimizer for optimizing logical plans.
- It is scalable, familiar, and extensible.
- Using HiveQL doesn’t require any knowledge of programming language, Knowledge of basic SQL query is enough.
- We can easily process structured data in Hadoop using Hive.
- Querying in Hive is very simple as it is similar to SQL.

Hive Limitations

- Apache does not offer real-time queries and row level updates.
- Hive also provides acceptable latency for interactive data browsing.
- It is not good for online transaction processing.
- Latency for Apache Hive queries is generally very high.

Hive Optimization

Change the execution engine (set hive.execution.engine = tez or spark) for better execution plan. In traditional query, it does the load before filtering, but in tez, filter is done first

Partitioning (Static (Load data in tables) and dynamic(insert data into tables) hive.partition) Ex Partitioning on date column

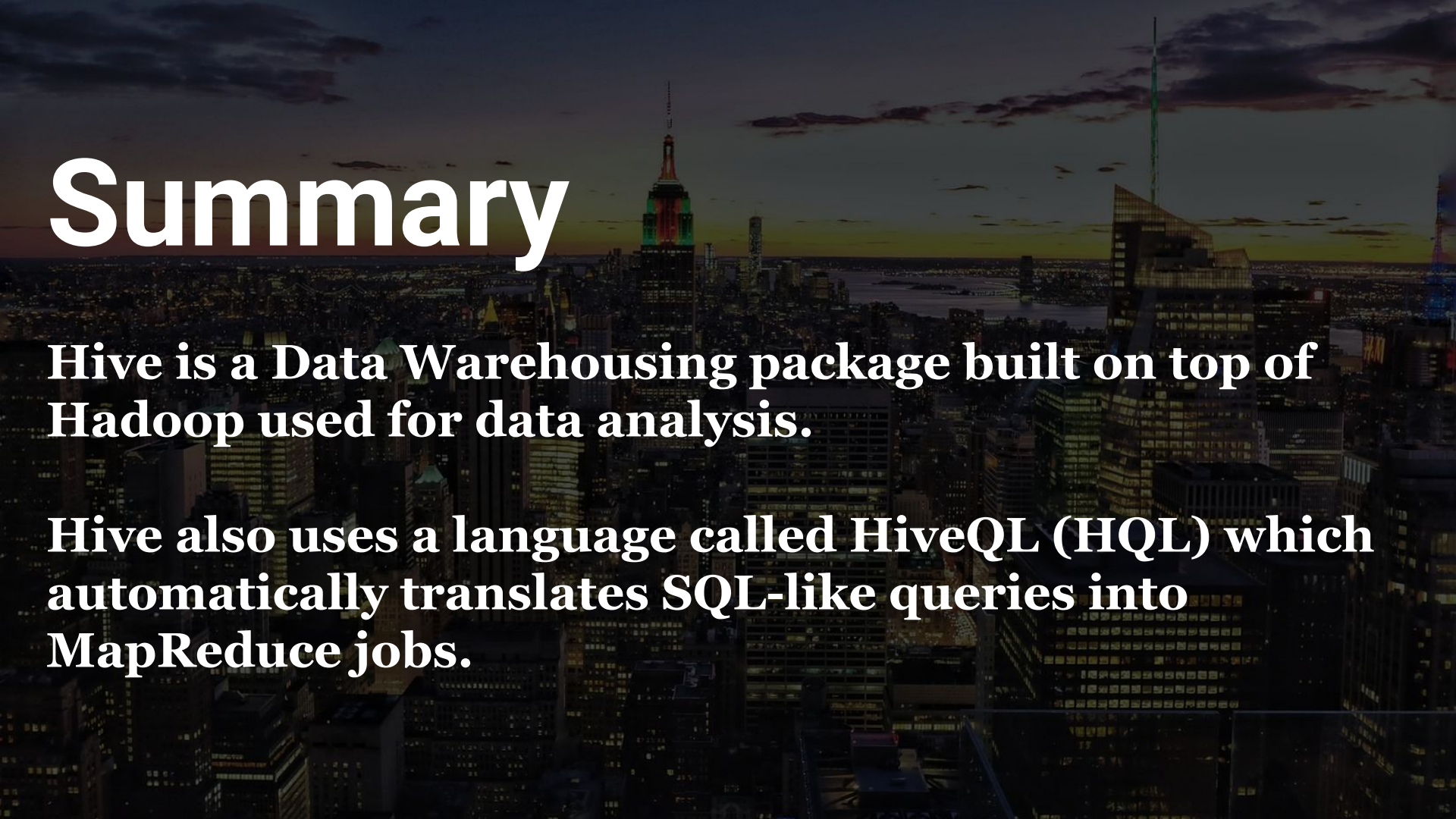
Bucketing - Create ranges for id for instance

Use joins (map side(only map, no reduce), sort merge bucketing())

Change file format to ORC

Sort by (uses multiple reducers) instead of order by (uses only one reducer)

Cluster by is combination of distributed by and sort by



Summary

Hive is a Data Warehousing package built on top of Hadoop used for data analysis.

Hive also uses a language called HiveQL (HQL) which automatically translates SQL-like queries into MapReduce jobs.