WEEK 04: APACHE HIVE BASICS-

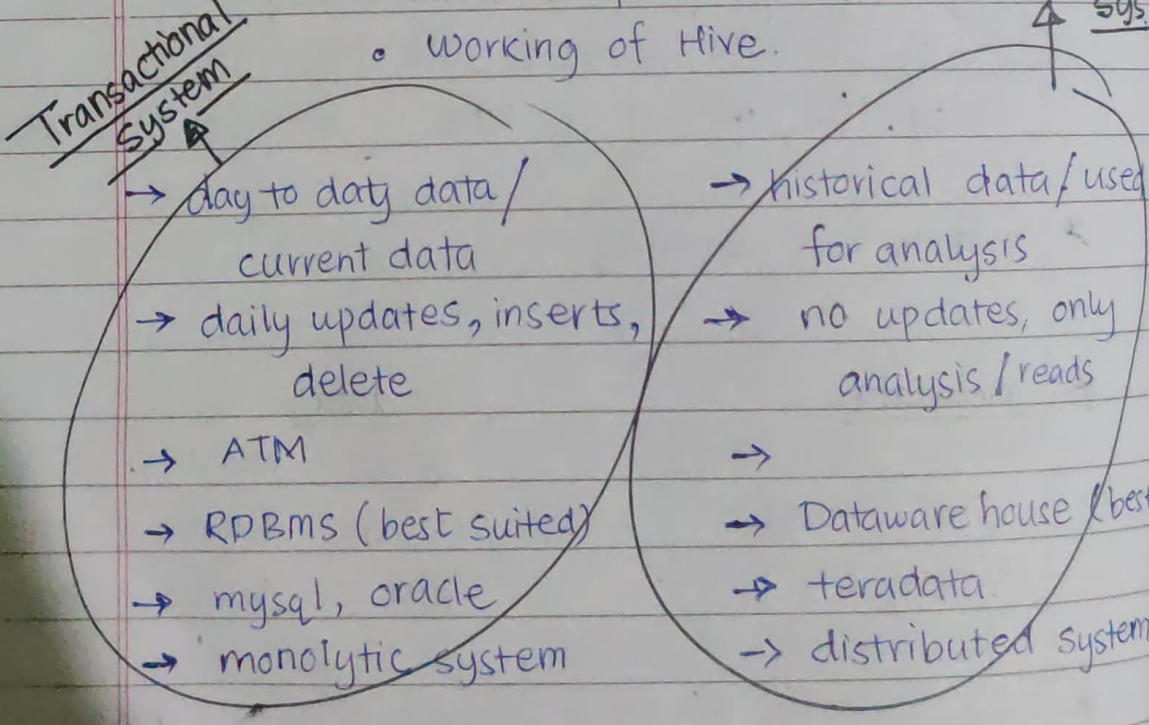
A. INTRODUCTION
B. HIVE BASICS
C. HIVE PRACTICAL
D.

HIVE:
data warehouse tool to process structured, hystor. data. (on top of HADOOP)

Tutorial 01
41 min.

A. INTRODUCTION

- Transactional & Analytical sys.
- HIVE + HQL
- Hive table, metadata
- why hive over traditional database
- Transactional & Analytical Processing.
- Data warehousing
- Hive Architecture
- Hive on top of Hadoop
- working of Hive.



Hive is open source data warehouse
(meant to solve analytical problems)

HIVE converts HQL to mapreduce job and will submit on the hadoop cluster.

ulitity : mapreduce works on java, insted of writing long java codes we can write hive querries in single lines and hive will do all further processing.

"Abstracting complexities from user"

* **Hive table**

2 parts of table.

1. Actual data (stored in HDFS) (raw format).
2. Meta data (schema / format) (stored in database) (mysql).

* Why meta data is stored in database?

1). In HDFS updates & edits are difficult.
2). fast retrival of data in database (low latancy)

* both are kept separately.

* metadata	data		
emp_name (string)	shivam	banglore	100000
city (string)	rahul	delhi	100000
salary (int)	nitish	delhi	80000

* hive metadata is stored in mysql inside database named "metastore".

Tutorial 02
Hive theory 1

- Transactional & Analytical processing
- data warehouse concepts.
- HQL, metastore.
- Hive vs RDBMS
- HQL vs SQL.

Transactional Processing	Analytical Processing
- analyze individual entry	- analyze large batch
- recent and changing data	- historical data.
- updation of data	- only reads data
- real time access.	- long jobs.
- single data source	- multiple data source

example: delivery of any product by amazon. (RDBMS) mysql

revenue generated, extraction of reports/ results. (warehouse) teradata

Data Warehouse

- Long running batch jobs
- optimized for read operations
- holds data from multiple sources
- holds data for long period.
- hystorical data.

Vertica, Teradata, Oracle, IBM.

Hive metastore

- stores metadata of all tables.
- maps files to table
- holds schema for each table.

HIVE	v/s.	RDBMS
• large datasets		• small datasets
• parallel computation		• serial com.
• high latency		• low
• read operation		• write/update/read
• not ACID compliant		• ACID compliant
• HiveQL		• SQL

- cluster of machines
- low latency means. takes less time.

HQL	v/s	SQL
minimal index support		indexes allowed.
many built in functions		less b.i.f.
only equi join		all joins. allowed.
restricted subqueries		many sub.

Tutorial 03
35 min
Hive theory 2.

- data is by default stored in warehouse directory
  /user/hive/warehouse
- if creating a database
  /user/hive/warehouse/trendytech.db
- files.
  /user/hive/warehouse/trendytech.db/files.

* metadata is stored in metastore. (rdbms (mysql))

Complex Data Types.
(ARRAY, MAP, STRUCT.)

1). **ARRAY**
- homogenous collections.

2) **MAP**
- data in form of key-values.
- unordered

3) **STRUCT**
- class
- diff data types allowed.

Built in functions.

1. UDF — user defined function
2. UDAF — u.d. aggregate f.
3. UDTF — u.d. table generating f.

UDF	UDAF	UDTF
• works on a single row	• works on multiple row	• single
• outputs a single row	• outputs a single row	• multiple
• trim(), concat(), length(), round(), floor()	• count(*), sum(), avg()	• explode(), posexplode()

* (flattens the data)
* (provides lateral view)

Set Operations on HIVE

- union (only operation of HIVE)
- minus (not supported in HIVE)
- Intersect (— " —)

union & union all

* How to use union instead of minus & intersect?

THE WHERE CLAUSE

1). IN / NOT IN
2). EXISTS / NOT EXISTS.

(a) Select id from customers
where id in (1111, 3333, 5555);

(b) Select id from customers where exists (
select cust_id from orders where
orders.cust_id == cust.id);

Tutorial 04
hivetheory 3

VIEWS IN HIVE

view = virtual table.

" it is better to share few colums rather than all columns "

Benifits :- 1) security.
2). can change according to need.
3). create diff. logical path.

Normalization

* normalization is not prefered.
* denormalization is prefered
  ↳ single table.