TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN

Thành viên:

ĐẶNG VĂN SANG	**18110352**
NGÔ ĐỒNG THIỆN	**18110370**
VÕ THÀNH ĐÔ	**18110270**

Đề tài: # TÌM HIỂU APACHE HIVE VÀ XÂY DỰNG DATA WAREHOUSE ĐƠN GIẢN

Giảng viên: thầy Huỳnh Xuân Phụng

BÁO CÁO CUỐI KÌ MÔN ĐIỆN TOÁN ĐÁM MÂY

Năm - 2020

MỤC LỤC

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA CNTT

******

ĐỀ CƯƠNG BÁO CÁO

Họ và tên sinh viên thực hiện 1: Đặng Văn Sang MSSV 1: 18110352

Họ và tên sinh viên thực hiện 2: Võ Thành Đô MSSV 2:18110270

Họ và tên sinh viên thực hiện 3: Ngô Đồng Thiện MSSV 3:18110370

Môn: Điện toán đám mây

Tên đề tài: Tìm hiểu apache hive và xây dựng data warehouse đơn giản

Nội dung thực hiện:

Lý thuyết:

- Giải thích các khái niệm cốt lõi của mô hình điện toán đám mây: cách sử dụng, đặc điểm, ưu điểm và thử thách khi sử dụng mô hình này
- Minh họa các khái niệm cơ bản về dịch vụ đám mây và mô tả cách sử dụng chúng trong các hệ thống đám mây
- Xác định các nguyên tắc cơ bản trong việc quản lý tài nguyên trong điện toán đám mây.
- Áp dụng các khái niệm cơ bản về data center để đánh giá tính được mất khi cân bằng giữa các yếu tố gồm: khả năng vận hành, hiệu quả và chi phí.
- Sử dụng các dịch vụ đám mây trong thiết kế và thực hiện các giải pháp cho bài toán.
- Đánh giá và lựa chọn giải pháp cụ thể về điện toán đám mây cho các bài toán trong thực tế
- Phân tích các mô hình lập trình đám mây khác nhau và áp dụng chúng để giải quyết các vấn đề trên đám mây.

Thực hành:

- Làm việc trong nhóm, cùng nghiên cứu và trao đổi giải quyết vấn đề trên cơ sở lập trình hướng đối tượng.
- Trình bày trước đám đông sử dụng phương tiện trình chiếu

KẾ HOẠCH THỰC HIỆN

STT	Thời gian	Công việc	Phân công
1	19/10/2020 đến 25/10/2020	Tìm hiểu khái niệm apache, apache hadoop, apache hive, warehouse,…	Thiện, Đô, Sang
2	26/10/2020 đến 01/11/2020	Tìm hiểu ứng dụng của apache hive trong các dự án thực tế	Thiện, Đô, Sang
3	02/11/2020 đến 08/11/2020	Cập nhật nội dung thực hiện trên trello	Thiện, Đô, Sang
4	09/11/2020 đến 16/11/2020	Tìm kiếm nguồn tài liệu phục vụ xây dựng đề tài	Thiện, Sang
5	17/11/2020 đến 24/11/2020	Tiến hành cài đặt các công cụ cần thiết: ubuntu, java, apache hadoop, apache hive,… Hỗ trợ fix các lỗi trong quá trình cài đặt	Đô, Thiện, Sang

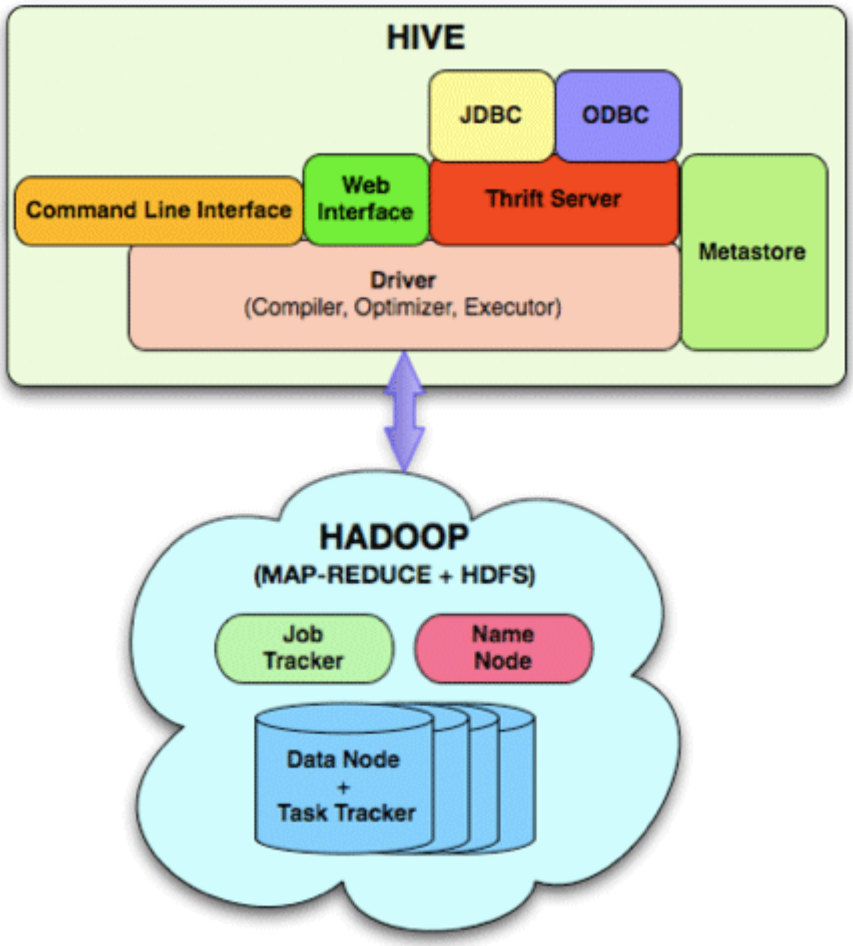
6		Thực hành làm quen với các câu lệnh trên database, trên table đơn giản	Sang, Đô
7	25/11/2020 đến 02/12/2020	Tiến hành xây dựng data warehouse đơn giản	Sang
8	03/12/2020 đến 10/12/2020	Viết báo cáo và slide trình bày. Cập nhật trên trello.	Đô
9	11/12/2020 đến 18/12/2020	Trình bày demo data warehouse đơn giản	Thiện, Sang
10	02/01/2021	Báo cáo và kết thúc môn học	Thiện, Sang, Đô

GIỚI THIỆU CƠ BẢN VỀ APACHE HIVE

1. GIỚI THIỆU

1.1. Apache Hive là gì

- Apache Hive (Hive là một công cụ SQL trên Hadoop) : Với một giao diện giống như SQL, Hive cho phép vượt qua bình phương dữ liệu từ HDFS. Phiên bản Hive của ngôn ngữ SQL được gọi là HiveQL.
- Hive là một kho dữ liệu (data warehouse) xử lý các dữ liệu dạng cấu trúc trên nền tảng hadoop. Sử dụng hive để tổng hợp, tạo truy vấn và phân tích dữ liệu một cách dễ dàng mà không cần phải hiểu nhiều về MapReduce.



1.2. So sánh Apache Hive với Hbase và PIG

1.2.1. So sánh Hive và Hbase

- Hive nên được sử dụng để truy vấn phân tích dữ liệu được thu thập trong một khoảng thời gian, để tính toán xu hướng hoặc nhật ký trang web.
- Không nên sử dụng Hive để truy vấn lại vì có thể mất một lúc trước khi bất kỳ kết quả nào được trả về. HBase hoàn hảo để truy vấn dữ liệu lớn theo thời gian thực
- Hive và HBase là hai công nghệ khác nhau dựa trên Hadoop - hive là một công cụ giống SQL chạy MapReduce và HBase là NoSQL trên Hadoop.
- Giống như Google để tìm kiếm và Facebook cho mạng xã hội, Facebook dùng hive dùng để phân tích dữ liệu trong khi Google dùng HBase để truy vấn thời gian thực. dữ liệu thậm chí có thể được đọc và ghi từ Hive sang HBase và ngược lại.

1.2.2. So sánh Hive và PIG

- Hive được sử dụng chủ yếu bởi nhà phân tích dữ liệu trong khi Pig được sử dụng chung bởi các nhà nghiên cứu và lập trình.
- Hive được sử dụng cho dữ liệu có cấu trúc trong khi Pig được sử dụng cho dữ liệu bán cấu trúc.
- Hive chủ yếu được sử dụng để tạo báo cáo trong khi Pig chủ yếu được sử dụng để lập trình.
- Hive chia các phân vùng để có thể xử lý tập hợp con của dữ liệu theo ngày hoặc theo thứ tự bảng chữ cái trong khi Pig không có bất kỳ khái niệm nào về phân vùng mặc dù có thể người ta có thể đạt được điều này thông qua các bộ lọc.

2. KIẾN TRÚC HIVE

2.1. Khái niệm Big Data

2.1.1. Định nghĩa:

Thuật ngữ "big data" được sử dụng để nói đến tập dữ liệu lớn trong đó hàng ngày nó gia tăng về cả khối lượng, tốc độ và đa dạng về kiểu dữ liệu.

2.1.2. Danh mục Big Data:

- Cấu trúc
- Không cấu trúc
- Bán cấu trúc

2.1.3. Ví dụ về Dữ liệu lớn:

- Sàn giao dịch New York tạo ra khoảng 1TB dữ liệu giao dịch mới mỗi ngày.
- Phương tiện truyền thông xã hội: Thống kê cho thấy hơn 500 terabyte dữ liệu được đưa vào cơ sở dữ liệu của trang web truyền thông xã hội Facebook mỗi ngày.
- Dữ liệu chủ yếu được tạo theo:
    - Tải lên hình ảnh và video
    - Trao đổi tin nhắn
    - Bình luận
- Máy bay phản lực / Cổng du lịch: Một động cơ phản lực ingle tạo ra hơn 10 terabyte (TB) dữ liệu trong 30 phút bay mỗi ngày. Việc tạo ra dữ liệu lên tới nhiều petabyte(PB).

2.2. Khái niệm Apache Hadoop

- Hadoop là một khung công tác nguồn mở được quản lý bởi Quỹ phần mềm Apache. Nguồn mở ngụ ý rằng nó có sẵn miễn phí và mã nguồn của nó có thể được thay đổi theo yêu cầu của người dùng. Apache Hadoop được thiết kế để lưu trữ và xử lý dữ liệu lớn một cách hiệu quả. Hadoop được sử dụng để lưu trữ dữ liệu, xử lý, phân tích, truy cập, quản trị, vận hành và bảo mật.
- Việc quản lý và xử lý đống dữ liệu này tạo ra một thách thức vô cùng lớn.Và Apache đã tạo ra một framework để quản lý và xử lý các thách thức mà big data mang lại, đó là Hadoop.
- Hadoop có thể hiểu là một framework mã nguồn mở sử dụng để lưu trữ và xử lý dữ liệu lớn. Nó bao gồm hai thành phần chính là: MapReduct và HDFS (Hadoop Distributed File System)
- Làm thế nào để làm việc với hadoop? Tất nhiên là apache cũng cung cấp cho chúng ta các công cụ để có thể làm việc được với hadoop một cách dễ dàng nhất. Sqoop, Pig, Hive là các công cụ đó.
    - Sqoop: Dùng để chuyển đổi dữ liệu qua lại giữa RDBMS(dữ liệu quan hệ) với HDFS.

- Pig: nền tảng là ngôn ngữ thủ tục được sử dụng để phát triển một kịch bản cho các hoạt động MapReduce.
- Hive: Nền tảng là SQL script để làm hoạt động MapReduce.

- Khi nào thì sử dụng Sqoop, pig, Hive? Câu trả lời là chúng ta sẽ lựa chọn chúng dựa trên dữ liệu phân tích. Với những dữ có cấu trúc rõ ràng thì Hive là lựa chọn tốt. Với những dữ liệu có cấu trúc và bán cấu trúc thì Pig sẽ dễ dàng tếp cận trong việc tạo kịch bản cho MapReduce. Còn với những dữ liệu đa dạng(có cấu trúc, bán cấu trúc, phi cấu trúc) thì các tiếp cận là tạo trương trình Java MapReduce truyền thống.

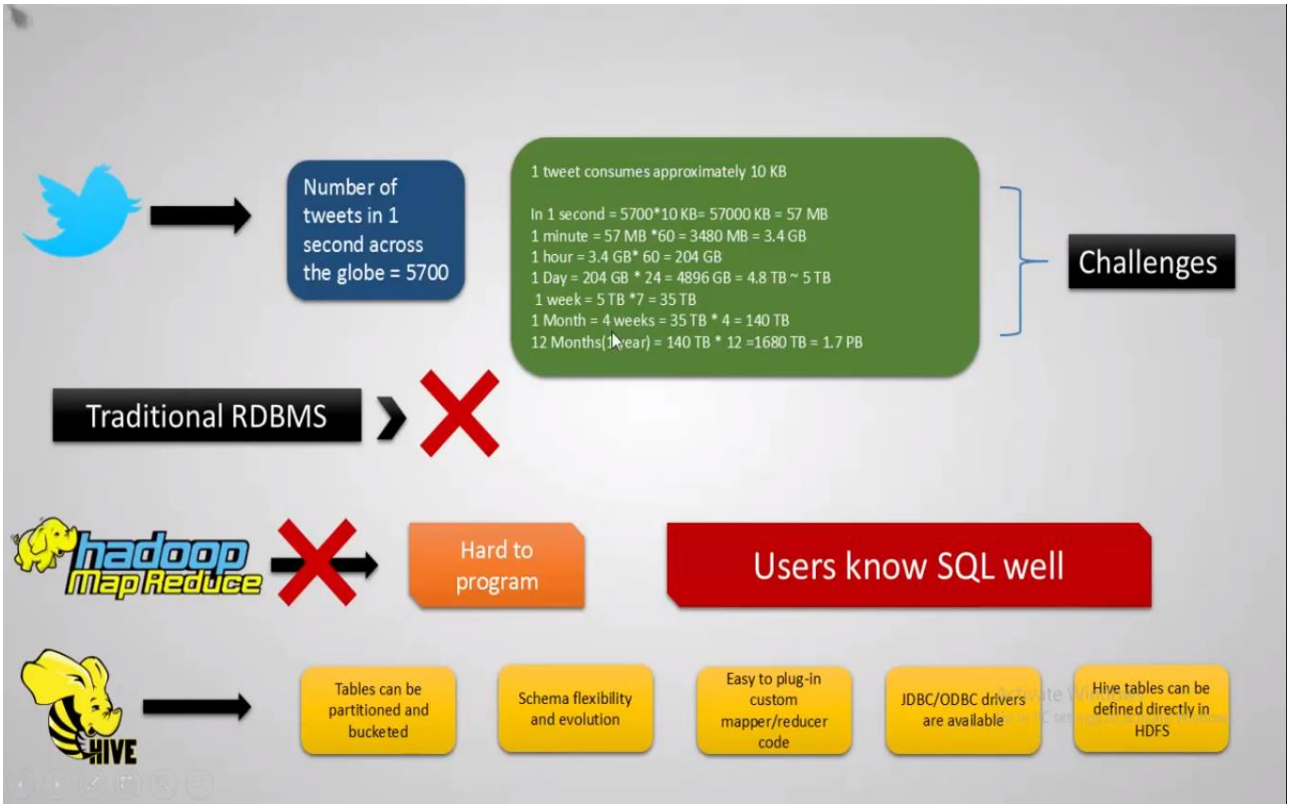
2.3. Khái niệm HDFS

- Hệ thống HDFS là nơi được sử dụng để lưu trữ và xử lý dữ liệu
- HDFS (Hệ thống tệp phân tán Hadoop) : HDFS có công việc quan trọng nhất để thực hiện trong khung Hadoop. Nó phân phối dữ liệu và lưu trữ nó trên mỗi nút có trong một cụm, đồng thời. Quá trình này làm giảm tổng thời gian lưu trữ dữ liệu vào đĩa.

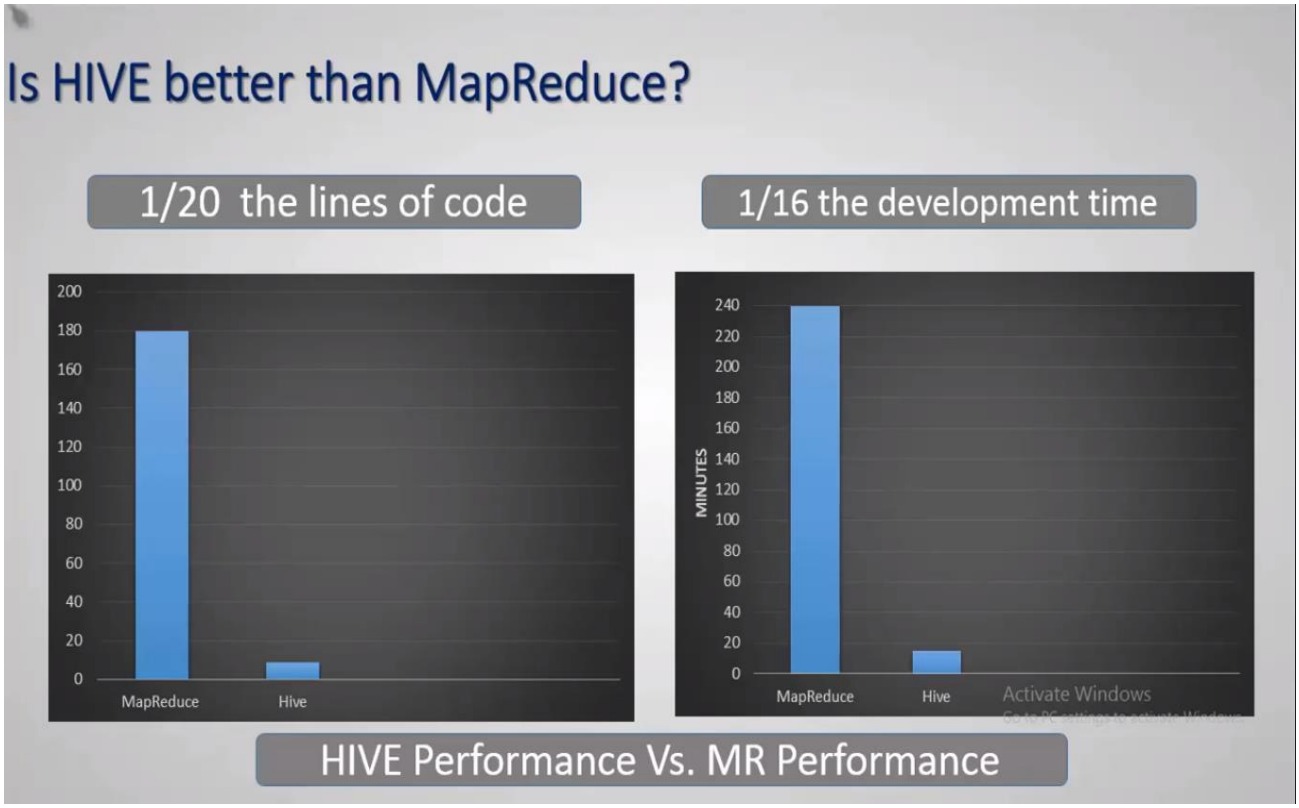
2.4. Hadoop MapReduce là gì?

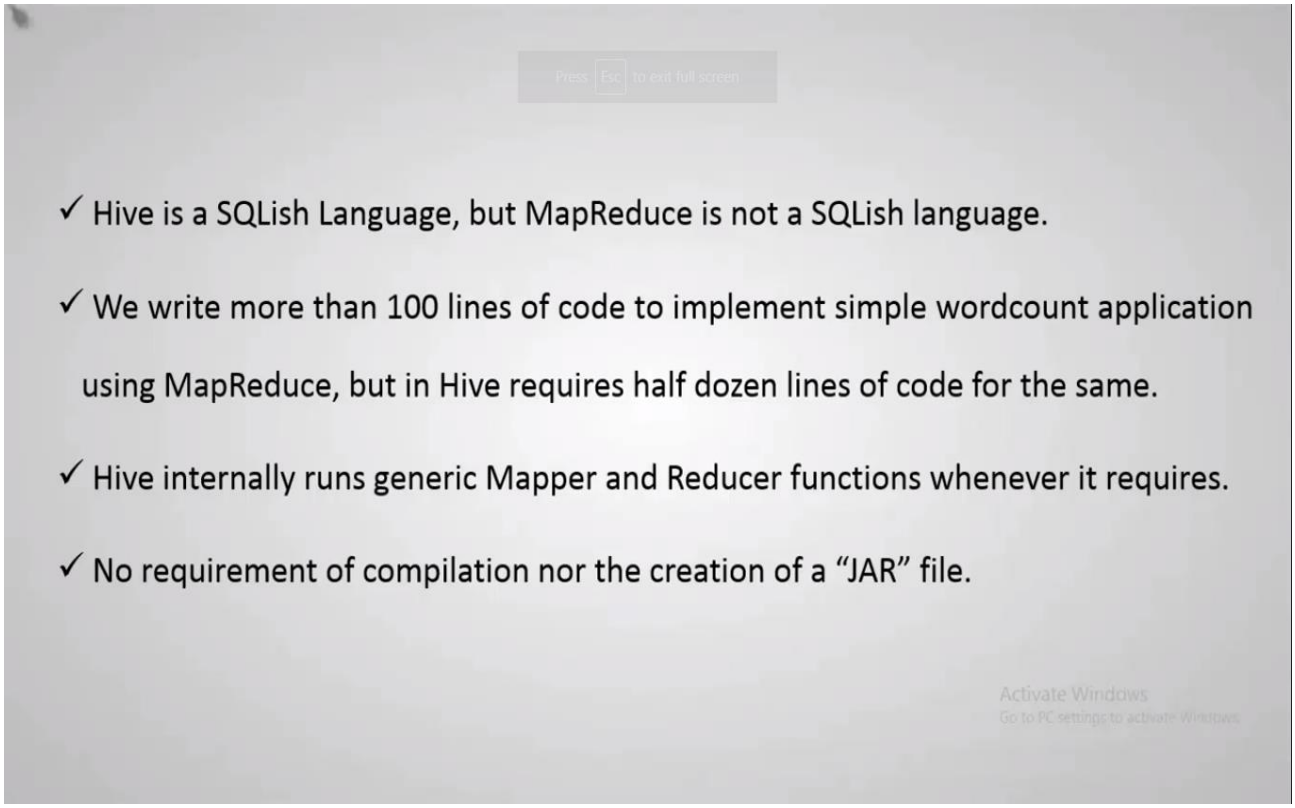
- Là một mô hình lập trình song song, nó xử lý dữ liệu có cấu trúc, bán cấu trúc, và không có cấu trúc
- MapReduce (Đọc / ghi dữ liệu lớn vào / từ Hadoop bằng MR) : Hadoop MapReduce là một phần quan trọng khác của hệ thống xử lý khối lượng dữ liệu khổng lồ được lưu trữ trong một cụm. Nó cho phép xử lý song song tất cả dữ liệu được lưu trữ bởi HDFS. Hơn nữa, nó giải quyết vấn đề chi phí xử lý cao thông qua khả năng mở rộng lớn trong một cụm.

2.5. Trường hợp nào cần dùng Hive?



2.6. So sánh Hive và MapReduce



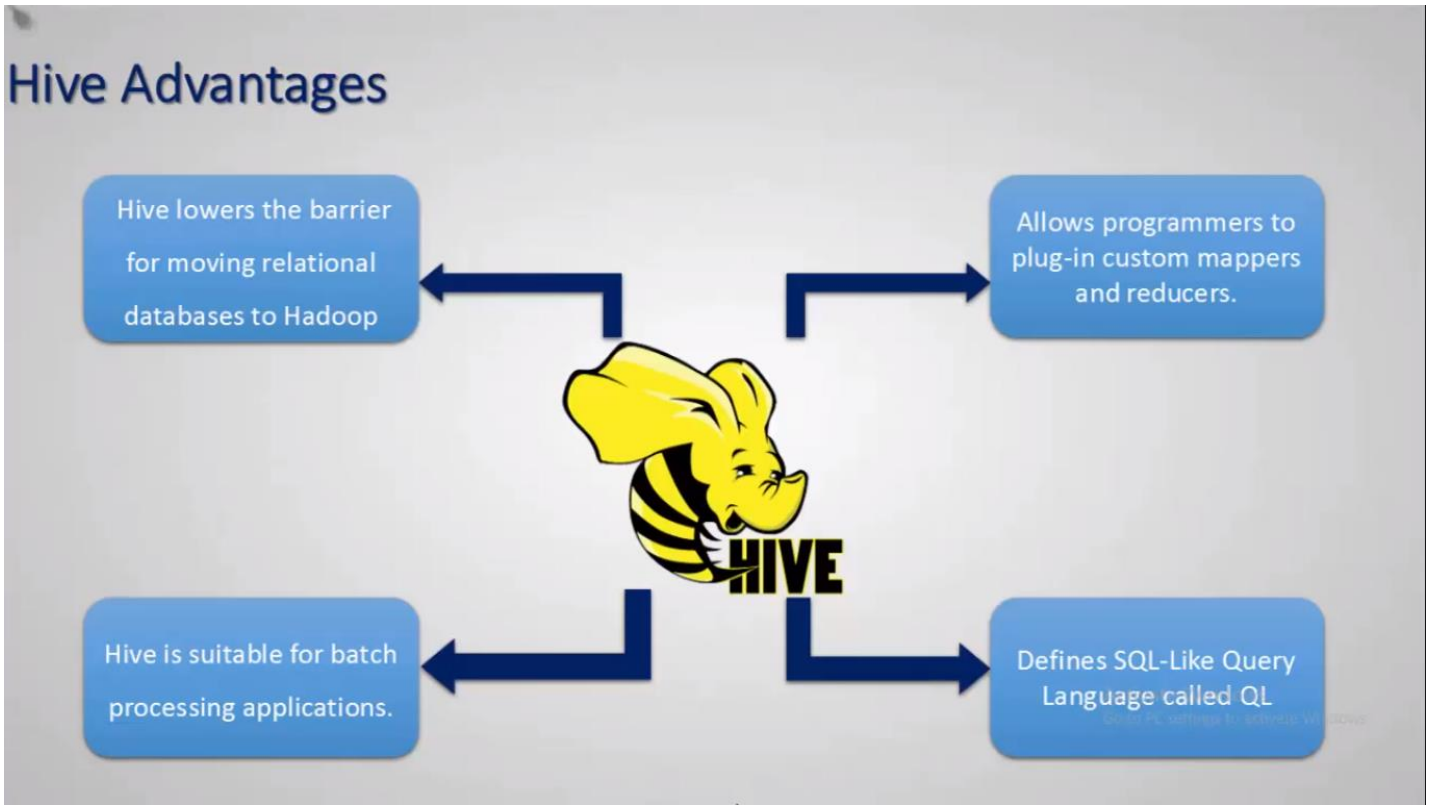


2.7. HiveQL là gì? Lợi ích của HiveQL

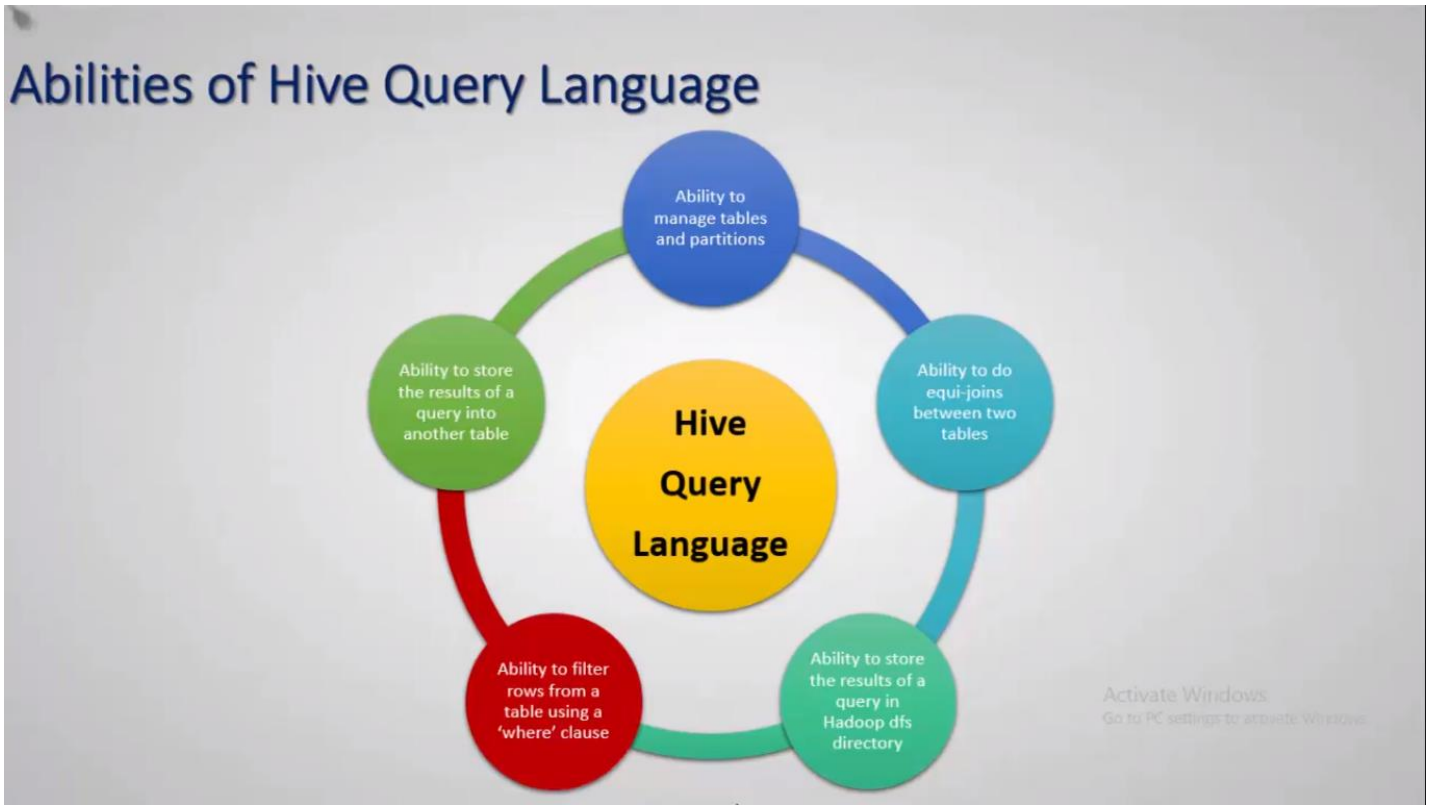
2.7.1. Khái niệm HiveQL

- Các truy vấn Hive được viết bằng HiveQL, là một ngôn ngữ truy vấn tương tự như SQL.
- Hive cho phép bạn chiếu cấu trúc trên phần lớn dữ liệu phi cấu trúc. Sau khi bạn xác định cấu trúc, bạn có thể sử dụng HiveQL để truy vấn dữ liệu mà không cần biết về Java hoặc MapReduce.

2.7.2. Lợi ích của Hive



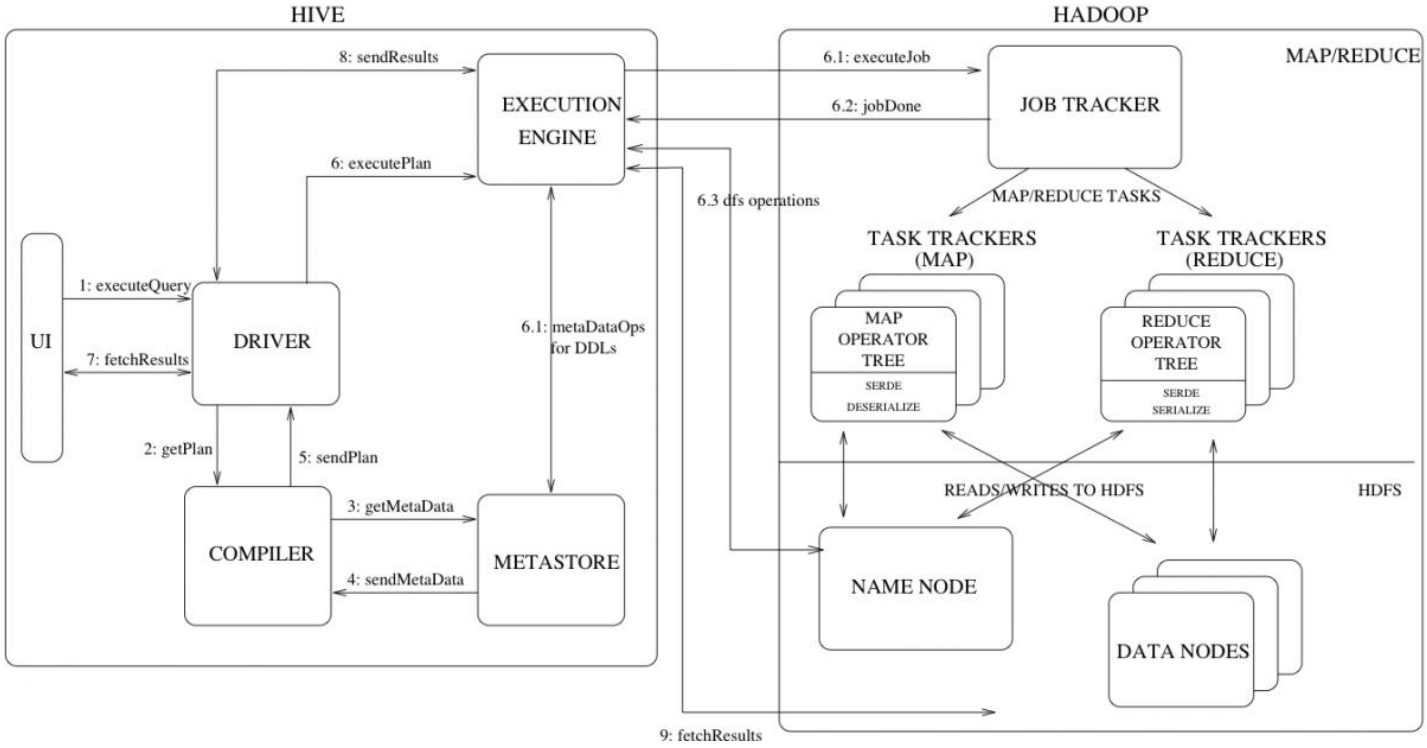
2.7.3. Khả năng của Hive



2.8. Hive Architecture

- Kiến trúc của Apache Hive

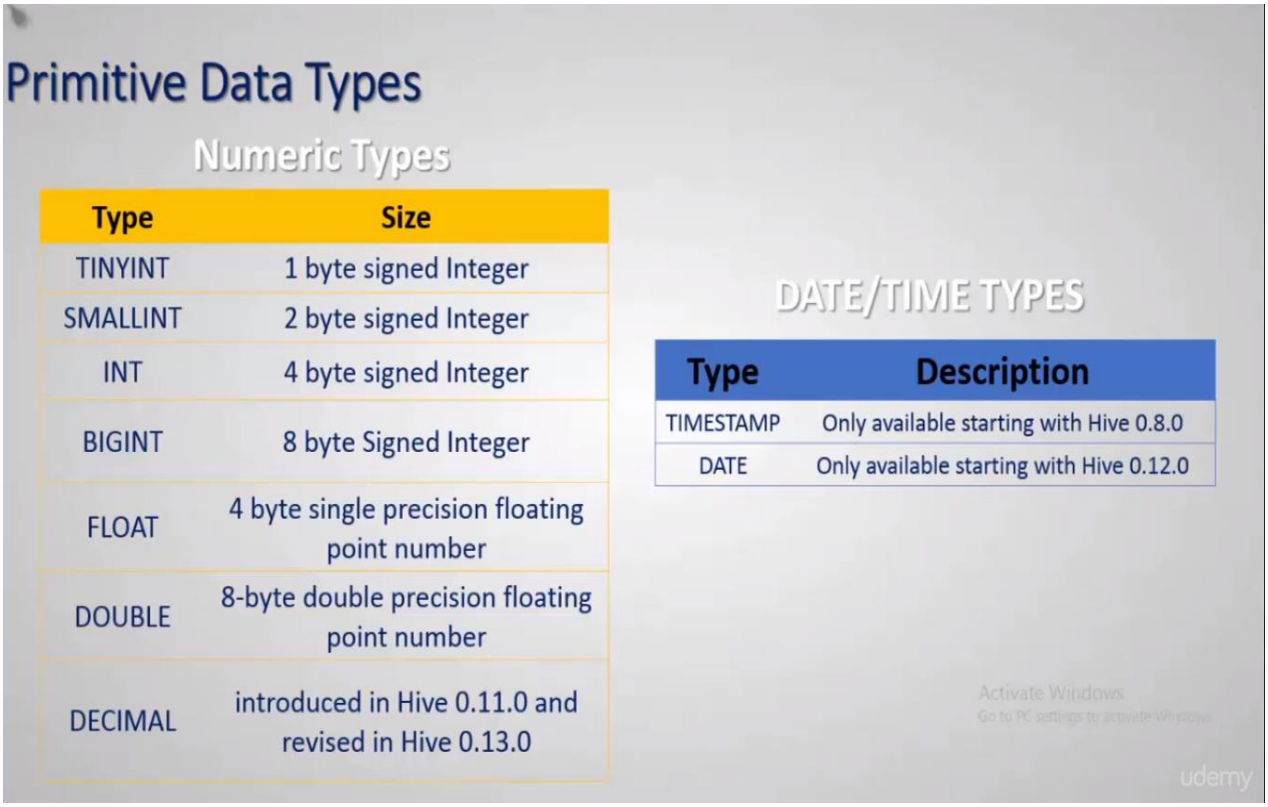
- Interface: Hive cung cấp một giao diện web để tương tác với hdfs. Tương tác command line.
- Meta store: Lưu thông tin cơ bản về cấu trúc dữ liệu. Các thông tin gồm ID của database (schema), ID của table, ID của index, định dạng của table, ...
- Hive QL: Tương tự như SQL, dùng để truy vấn dữ liệu dựa trên thông tin metastore cung cấp. Đây là phương pháp thay thế cho việc phải viết trương trình mapreduce truyền thống.
- Execution Engine: Chuyển hóa các lệnh Hive QL thành MapReduce
- HDFS: lưu trữ dữ liệu.

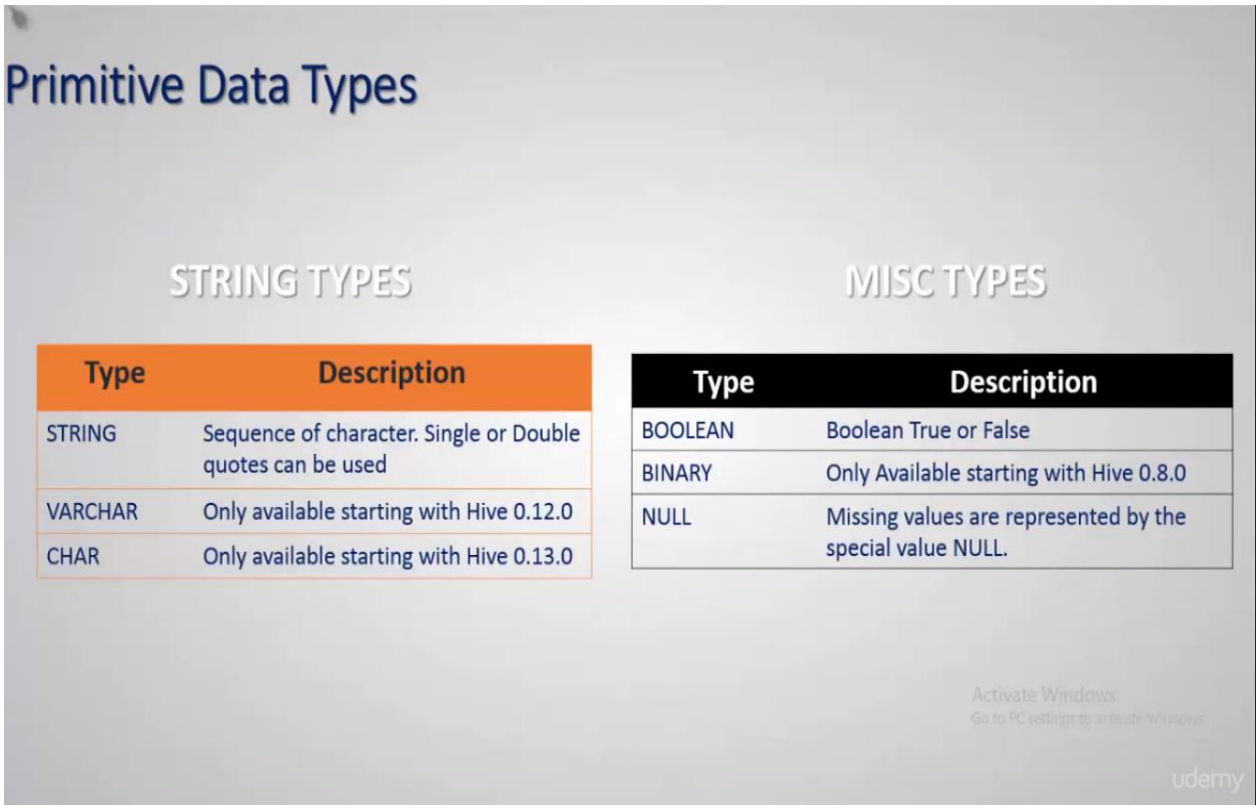


3. DATA DEFINITION LANGUAGE – DDL

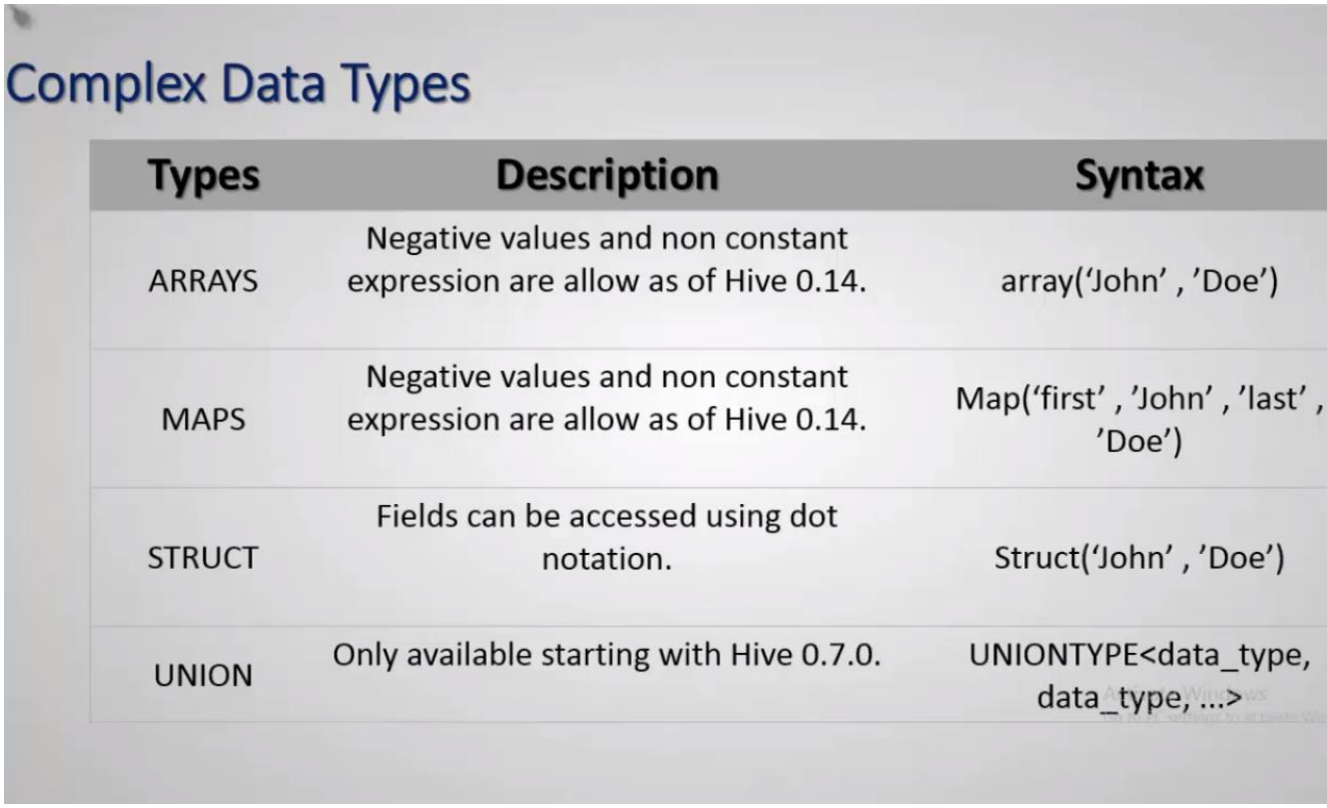
3.1. Kiểu dữ liệu trong Hive

3.1.1. Primitive Data Types



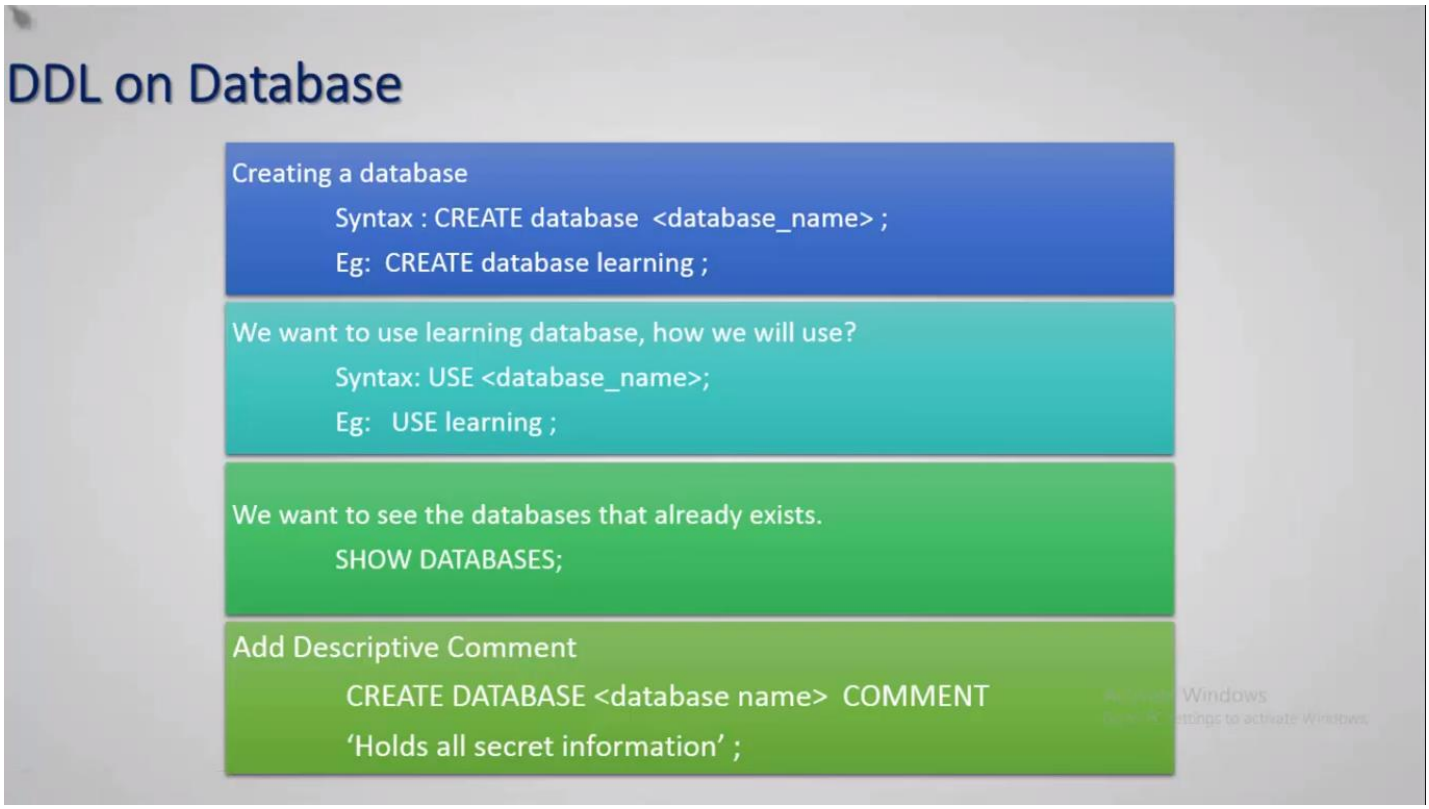


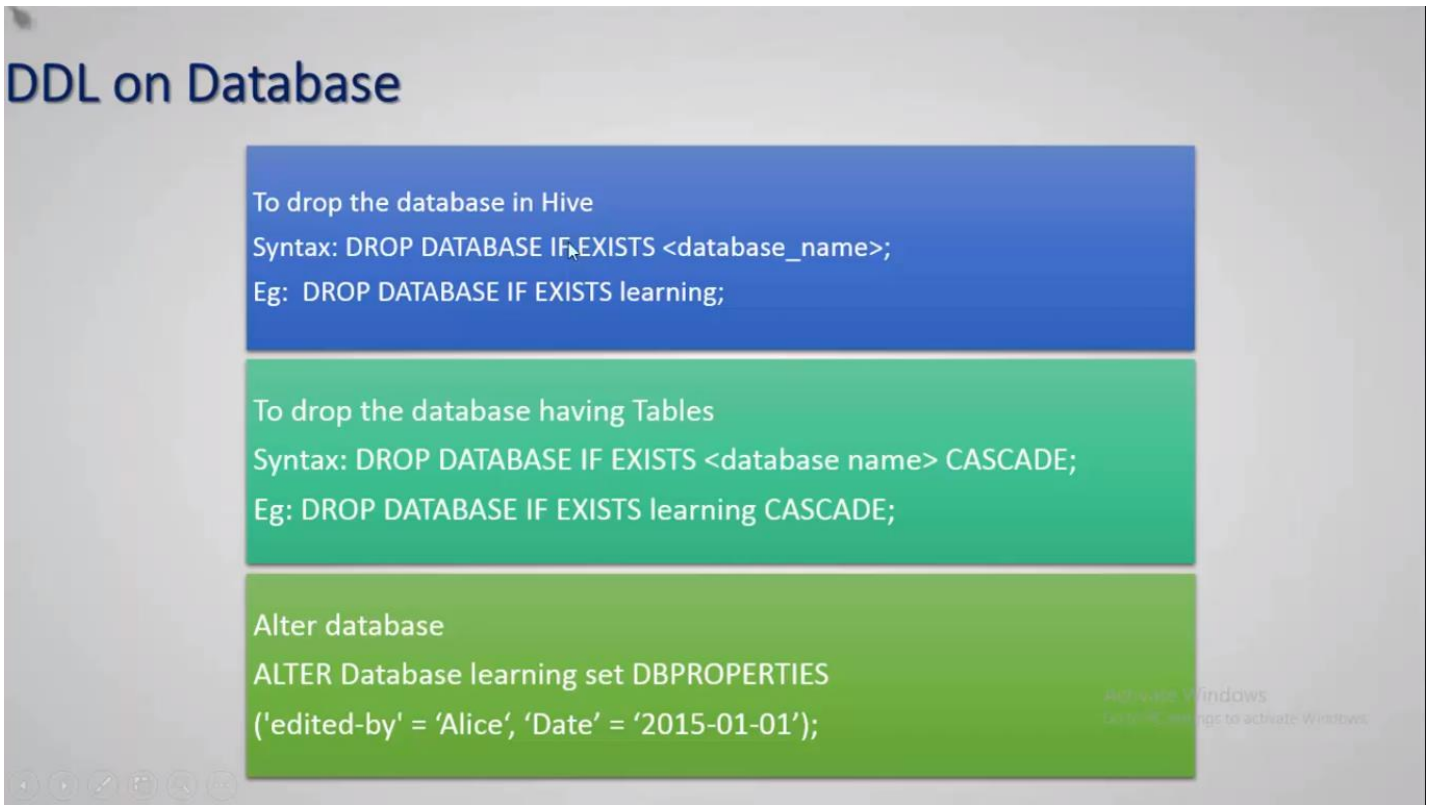
3.1.2. Complex Data Types



3.2. Ngôn ngữ truy vấn trong Hive – HiveQL

3.3. DDL trên database





3.4. DDL trên tables

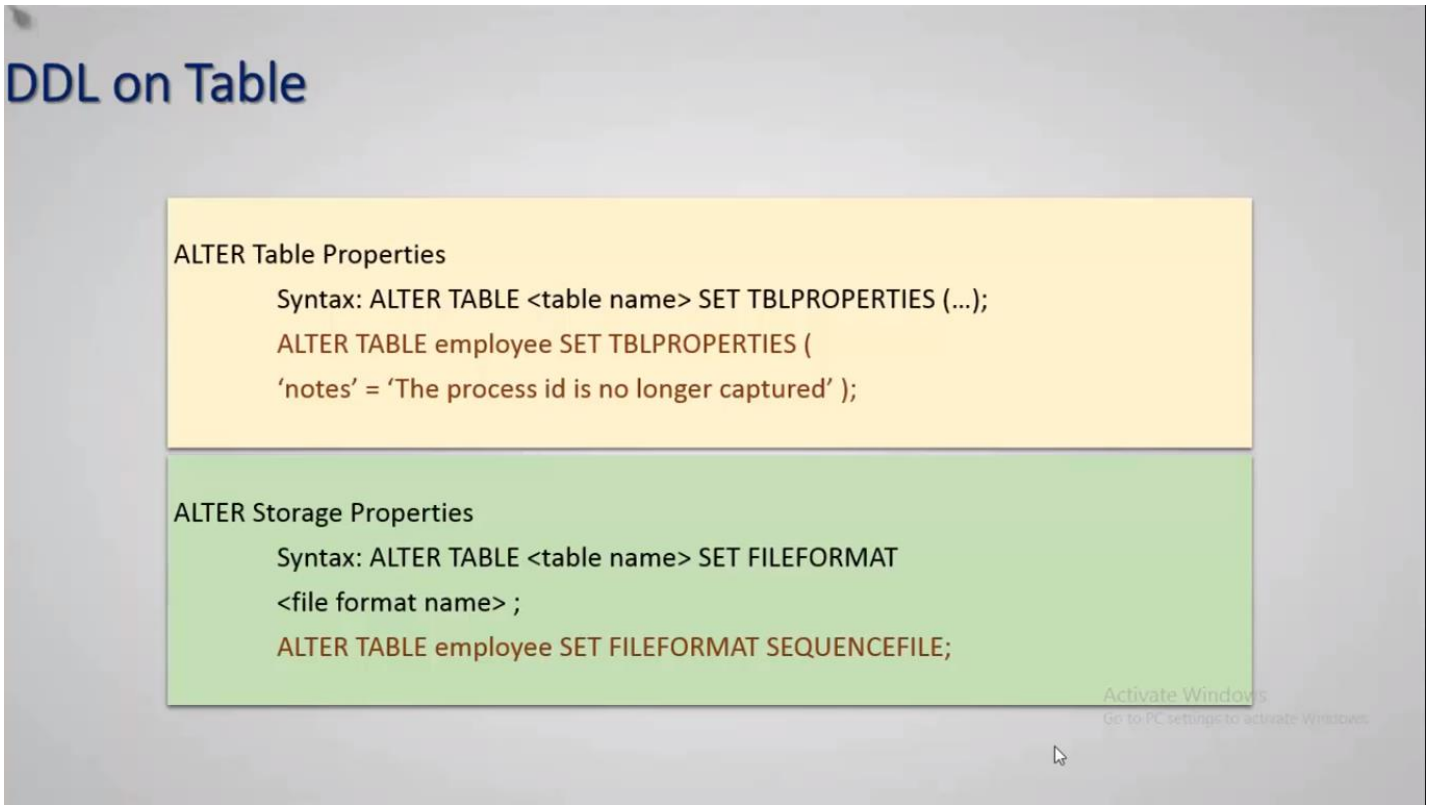
DDL on Tables

Creating a table in Apache Hive

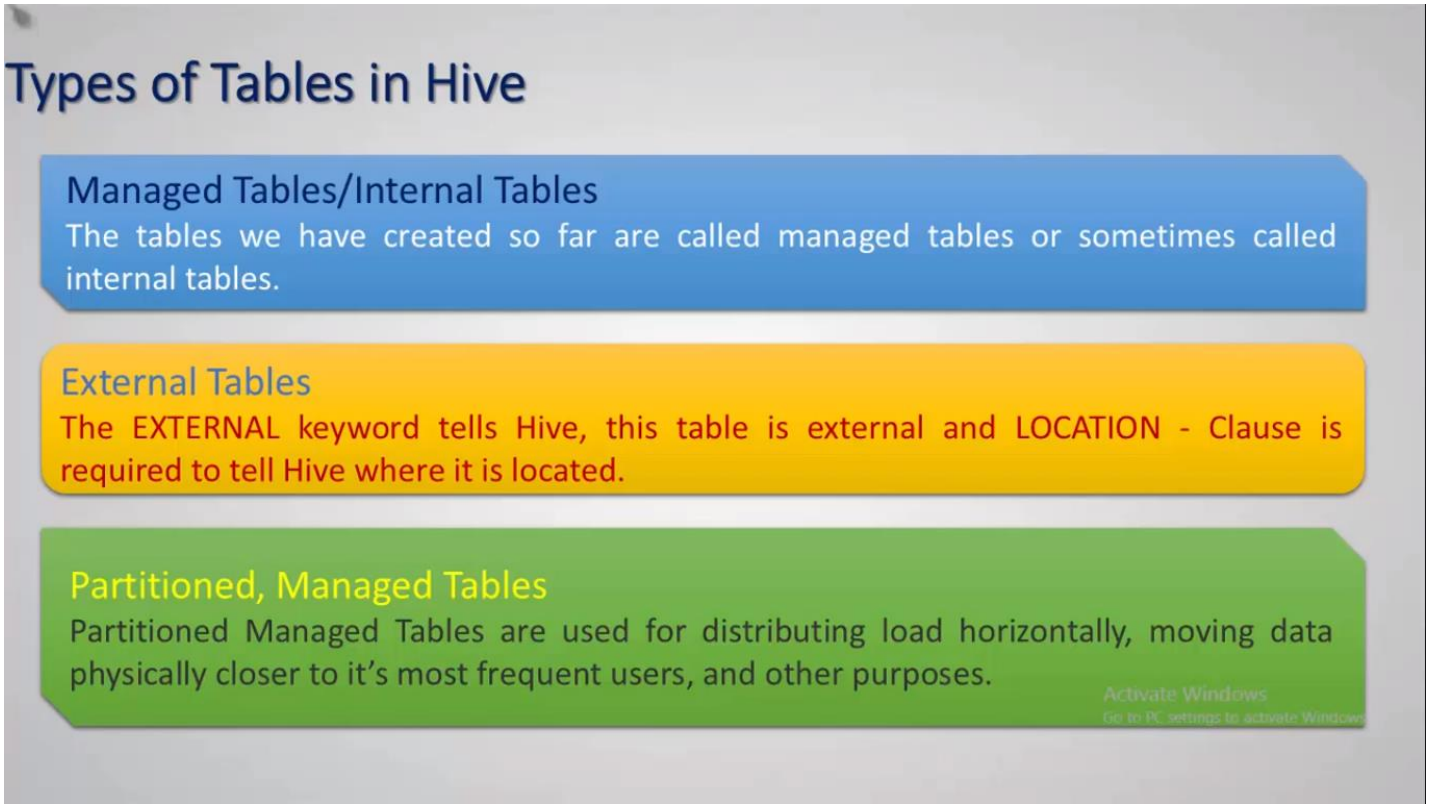
```
CREATE TABLE IF NOT EXISTS mydb.employee (
Name            STRING  COMMENT 'Employee name' ,
Salary          FLOAT   COMMENT 'Employee salary' ,
Address         STRING  COMMENT 'Employee address' )

ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;
```


DDL on Tables

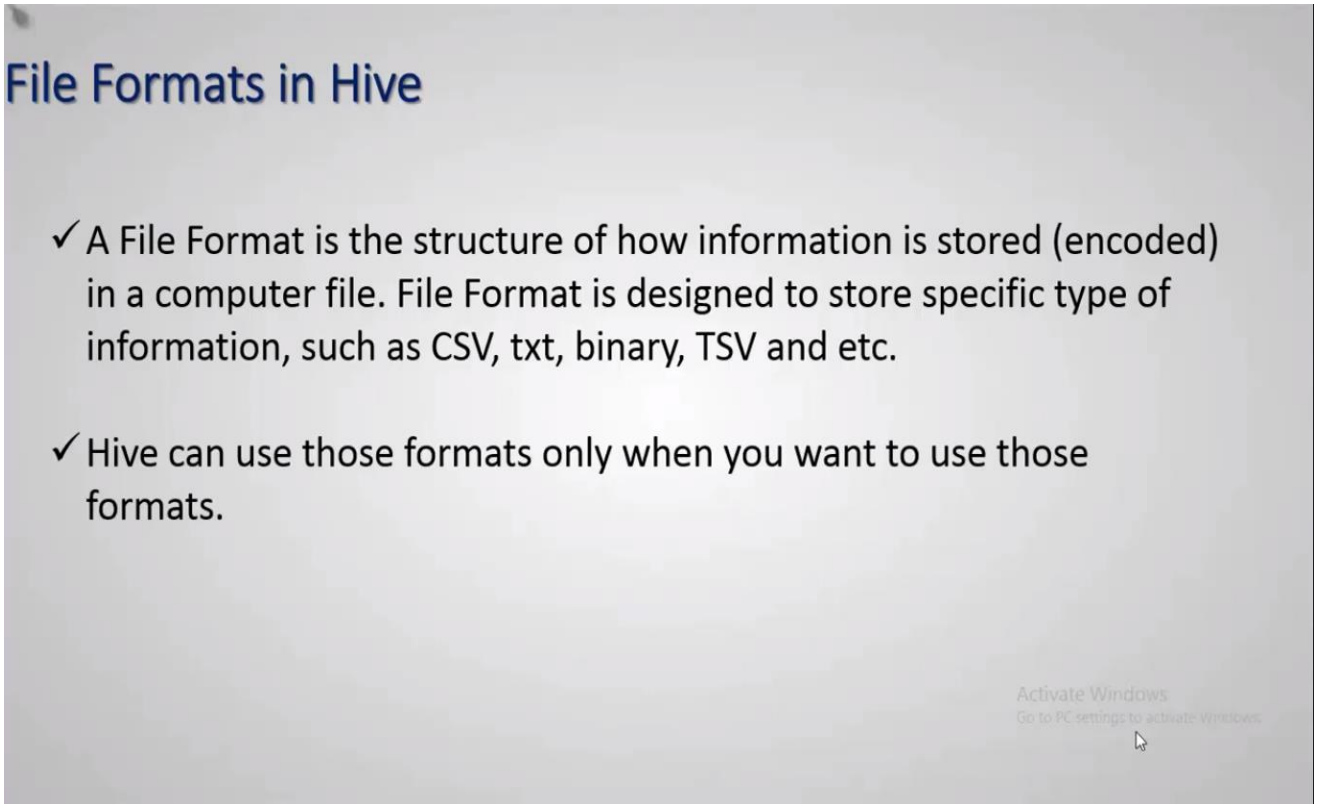
Copy Schema of an existing table	• CREATE TABLE IF NOT EXISTS mydb.new_employee • LIKE mydb.employee;
LISTS the tables which exists in the database	• SHOW TABLES; • SHOW TABLES IN mydb ; • SHOW TABLES 'empl.*' ;
See the Schema for a particular column	• Describe <table name>.<column name> ; • Ex : DESCRIBE employee.salary;
To see more verbose and readable output of description of a table	• Describe FORMATTED <table name> ; • Ex : Describe FORMATTED employee ;



3.5. Các tables khác nhau trong Hive



3.6. Định dạng File Format trong Hive



3.7. Lệnh DROP, ALTER trên TABLE

DROP TABLE:
+ Cú pháp: DROP TABLE IF EXISTS <table_name>;
+ Ví dụ: DROP TABLE IF EXIST tablevidu;

ALTER TABLE:
+ Cú pháp: ALTER TABLE <table_name> RENAME TO <new table name>;
+ Ví dụ: ALTER TABLE tencu RENAME TO tenmoi;

3.8. Thêm, xóa Partition vào bảng

Với những bảng đã tồn tại trong database, ta có thể thực hiện việc thêm Partition hoặc loại bỏ Partition

Thêm Partition:
+ Cú pháp: ALTER TABLE <table_name> ADD IF NOT EXIST PARTITION
+Ví dụ: ALTER TABLE tablevidu ADD IF NOT EXIST PARTITION (nam = 2020, thang = 1, ngay =1);

Xóa Partition
+ Cú pháp: ALTER TABLE <table_name> DROP IF NOT EXIST PARTITION(..);
+Ví dụ: ALTER TABLE tablevidu DROP IF NOT EXIST PARTITION (nam = 2020, thang = 1, ngay =1);

3.9. Lệnh cập nhật, thêm, xóa hoặc thay thế cột trong bảng

Cập nhật tên cột trong bảng:

+ Cú pháp: ALTER TABLE <table_name> CHANGE COLUMN <column_name> <new_column_name><data_type>;

+Ví dụ: ALTER TABLE tablevidu CHANGE COLUMN ten tenmoi STRING;

Thêm cột trong bảng:

+ Cú pháp: ALTER TABLE <table_name> ADD COLUMN <column_name data_type> ;

+Ví dụ: ALTER TABLE tablevidu ADD COLUMN (diachi STRING, quequan STRING);

Xóa hoặc thay thế cột trong bảng:

+ Cú pháp: ALTER TABLE <table_name> REPLACE COLUMN <new_column_name…>;

+Ví dụ: ALTER TABLE tablevidu REPLACE COLUMN (ten STRING COMMENT 'ten cu', tenNguoiDung STRING COMMENT 'ten moi');

+Ví dụ: ALTER TABLE tablevidu ADD COLUMN (diachi STRING, quequan STRING);

Xóa hoặc thay thế cột trong bảng:

+ Cú pháp: ALTER TABLE <table_name> REPLACE COLUMN <new_column_name…>;

+Ví dụ: ALTER TABLE tablevidu REPLACE COLUMN (ten STRING COMMENT 'ten cu', tenNguoiDung STRING COMMENT 'ten moi');

3.10. Lệnh cho phép, không cho phép thực thi dữ liệu có partition trên một bảng

Ví dụ ta có bảng sau:

ALTER TABLE tablevidu TOUCH PARTITION(nam = 2020, thang = 1, ngay = 6)

Khi thực thi lệnh:

ALTER TABLE tablevidu PARTITION(nam = 2020, thang = 1, ngay = 6) ENABLE OFFLINE;

Có tác dụng ngăn dữ liệu partition đó bị đọc.

Khi thực thi lệnh:

ALTER TABLE tablevidu PARTITION(nam = 2020, thang = 1, ngay = 6) ENABLE NO DROP

Có tác dụng ngăn việc xóa dữ liệu thuộc partition đó

3.11. Truyền dữ liệu vào bảng trong Apache Hive

3.12. Tác dụng:

+ Thêm dữ liệu mới vào database.
+ Phục hồi lại dữ liệu.
+ Xóa dữ liệu trong database.
+ Chỉnh sửa dữ liệu tồn tại trong database.
+ Cú pháp: LOAD DATA LOCAL INPATH 'input-file' OVERWRITE INTO TABLE

‘table_name’;
Lưu ý: lệnh OVERWRITE chỉ có sẵn từ bản Hive 0.9.0 trở lên.

3.13. Thêm dữ liệu từ một bảng sang bảng khác

+ Ví dụ: INSERT OVERWRITE TABLE ten_them SELECT id, ten, tuoi FROM ten LIMIT 2;

Thêm dữ liệu có partition: INSERT OVERWRITE TABLE ten_them PARTITION (quequan, trangthai) .., t.quequan, t.trangthai FROM ten t;

Tạo bảng mới lấy dữ liệu từ một bảng khác có điều kiện:

Ví dụ: CREATE TABLE ten_dacbiet AS SELECT ten, tuoi, quequan, trangthai FROM ten WHERE ten==’Hong’;

4. TRUY VẤN TRONG APACHE HIVE

Có Lệnh truy vấn là các câu lệnh dùng để thu thập, hiển thị dữ liệu từ database theo yêu cầu của người dung.

Việc truy vấn dữ liệu có thể là trích xuất dữ liệu hoặc một truy vấn dựa trên hành động điều kiện cụ thể.

Lệnh ACTION có thể yêu cầu các tác vụ thêm từ việc trích xuất ví dụ như thêm, cập nhật, xóa.

Lệnh SELECT đơn giản là hiển thị dữ liệu được chọn.

4.1. Lệnh SELECT

Hiển thị dữ liệu từ một bảng, ví dụ:

SELECT ten, quequan FROM ten;

Hoặc

SELECT t.ten, t.quan FROM ten t;

4.2. Toán tử trong HIVE

Toán tử số học

Phép tính	Mô tả
A + B	Công A và B
A - B	Trừ A và B
A * B	Nhân A với A
A / B	Chia A cho B
A % B	Chia lấy phần dư
A & B	Tính bitwise AND của A và B
A | B	Tính bitwise OR của A và B
A ^ B	Tính bitwise XOR của A và B
~A	Tính bitwise NOTcủa A

- Toán tử quan hệ

Phép tính	Mô tả
A = B	A bằng B
A <>, A != B	A khác B

A < B	A bé hơn B
A > B	A lớn hơn B
A >= B	A lớn hơn hoặc bằng B
A <= B	A bé hơn hoặc bằng B
A IS NULL	A rỗng
A IS NOT NULL	A không rỗng
A LIKE B	A giống với B (kiểu string)

4.3. Hàm trong HIVE

Hàm toán:

Kiểu giá trị trả về	Tên hàm
BIGINT	round(d)
BIGINT	floor(d)
BIGINT	ceil(d)
DOUBLE	pow(d,p)
DOUBLE	sqrt(d)
DOUBLE	abs(d)
DOUBLE	pi()

Hàm gộp:

Kiểu giá trị trả về	Tên hàm
BIGINT	Count()
DOUBLE	Sum(d)
DOUBLE	Avg()
DOUBLE	Min()
DOUBLE	Max()