HIVE – Distributing SQL Queries with Hadoop

Not HIVE?

High latency – not appropriate for OLTP – Online, low latency

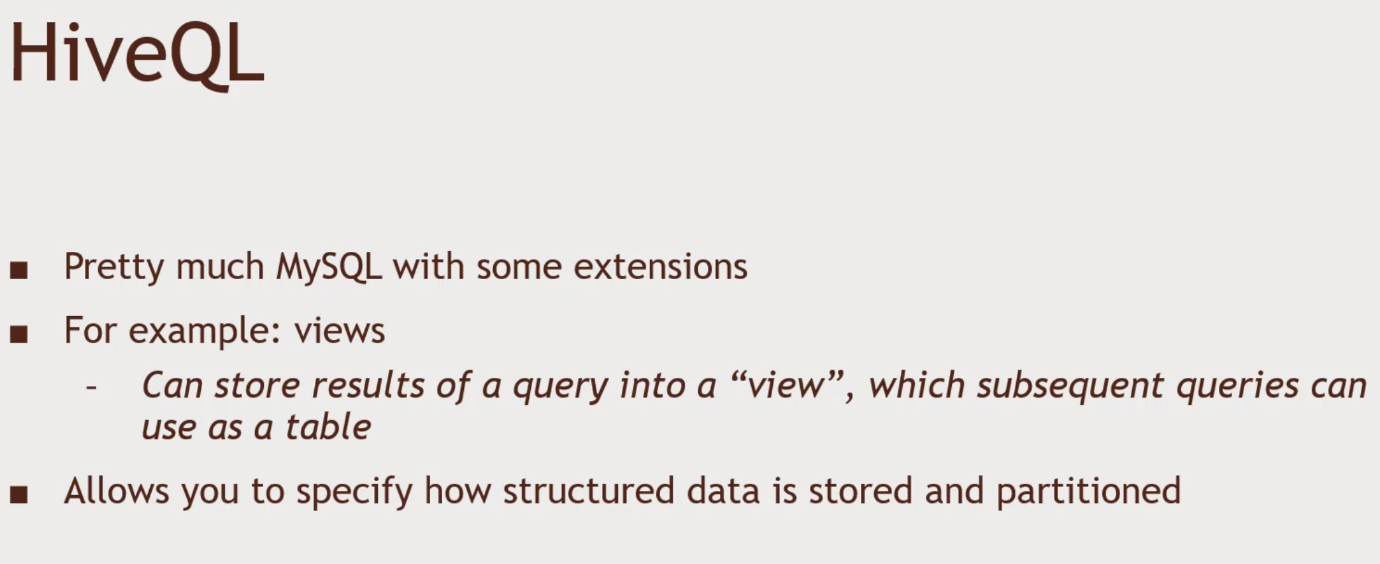
Stores data de-normalized

SQL is limited in what it can do

* Pig, Spark allows more complex stuffs

No transactions

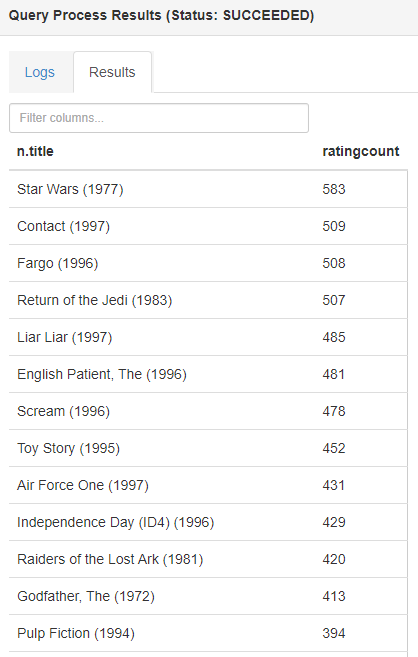
No record-level updates, inserts, deletes.



Hive View See Query Editor

DROP TABLE ratings;

DROP TABLE movie\_names;

Create a view and print out the names.

CREATE VIEW topMovieIDs AS

SELECT movieID, count(movieID) as ratingCount

FROM ratings

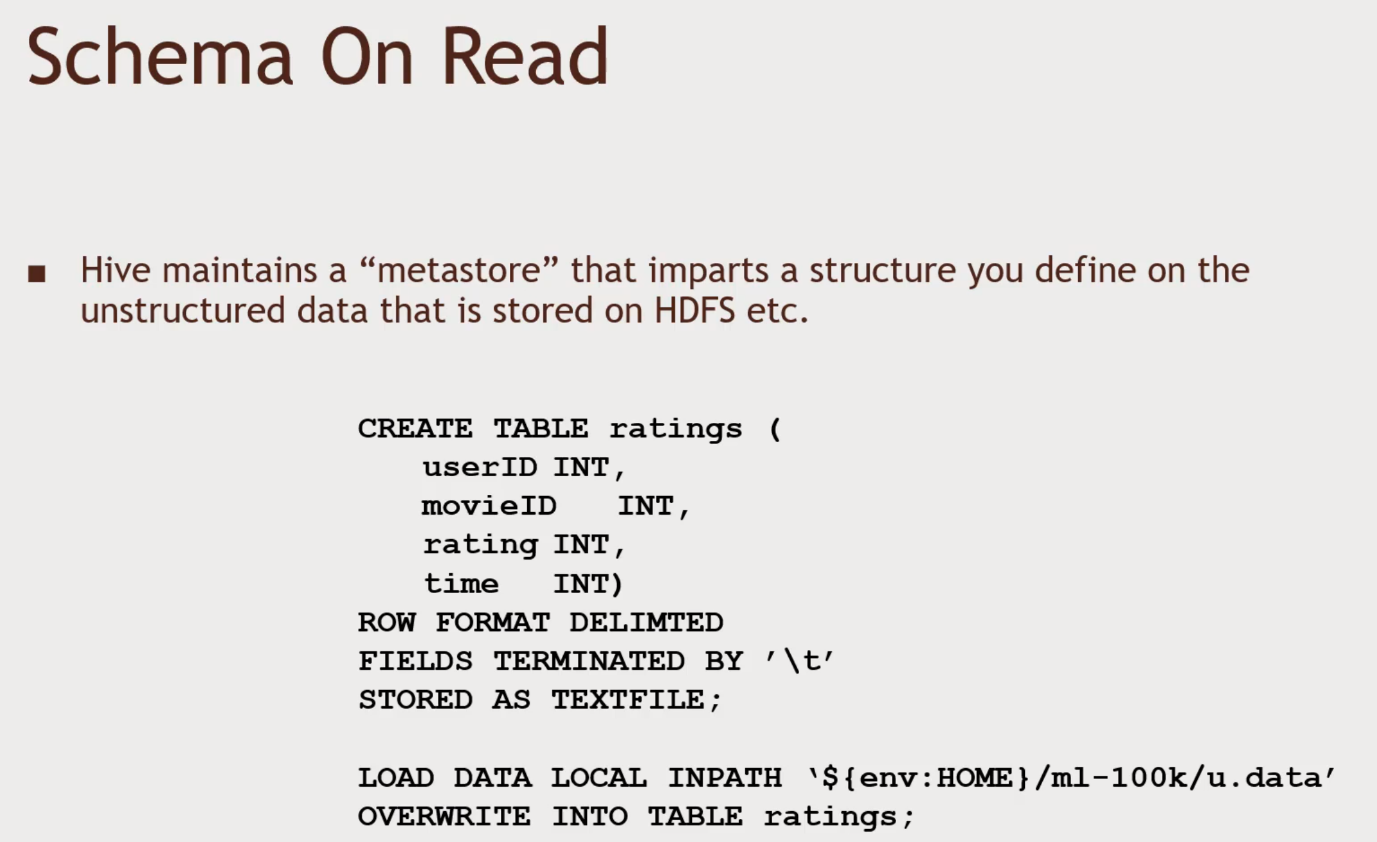
GROUP BY movieID

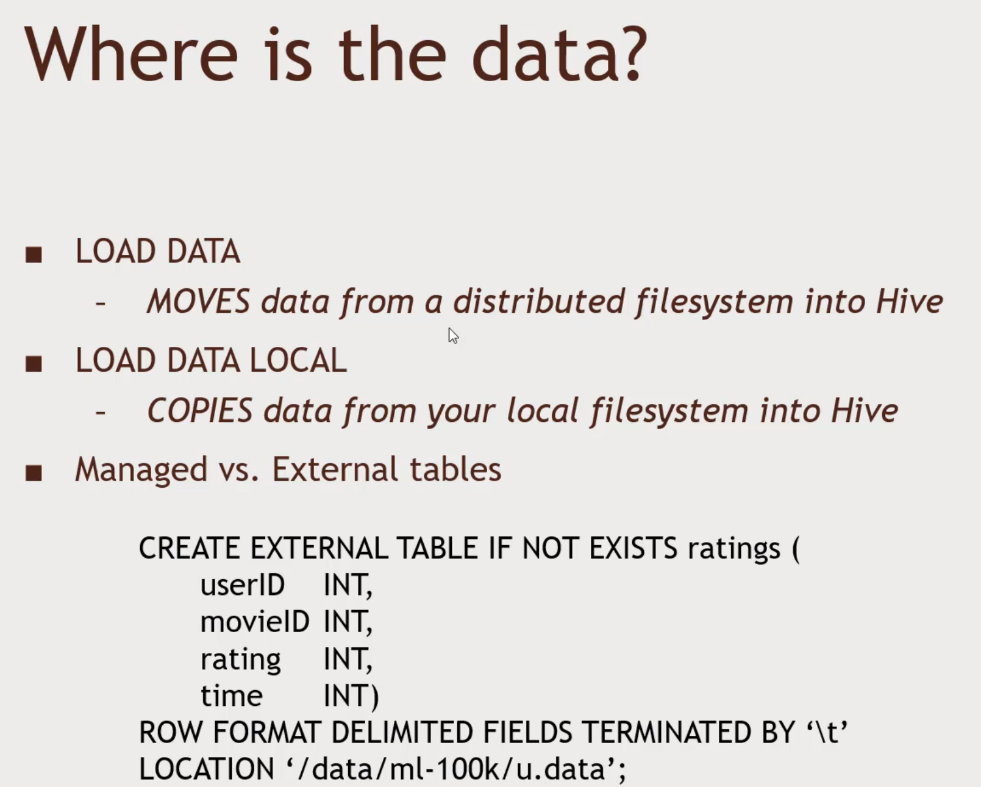
ORDER BY ratingCount DESC;

SELECT n.title, ratingCount

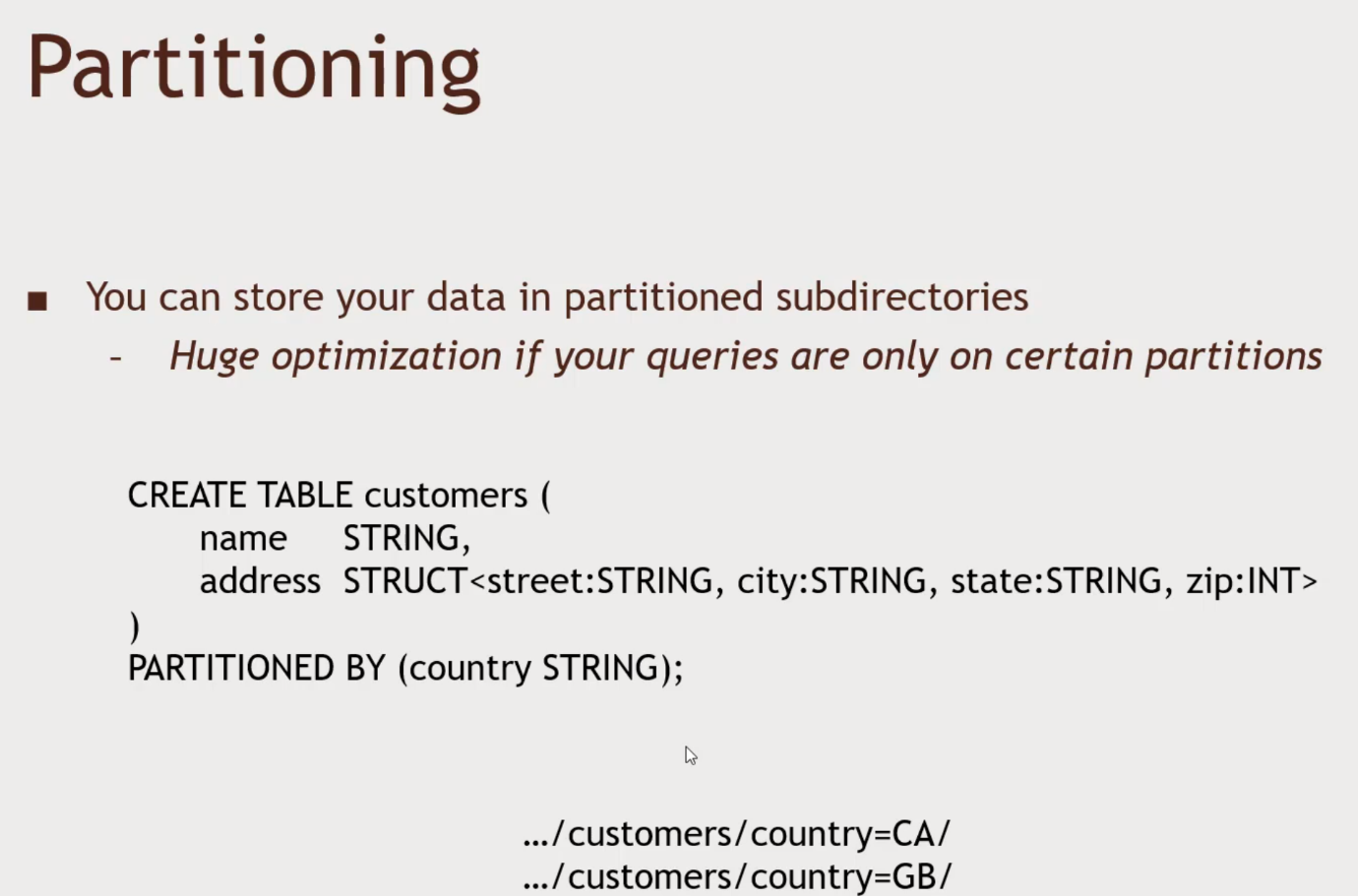
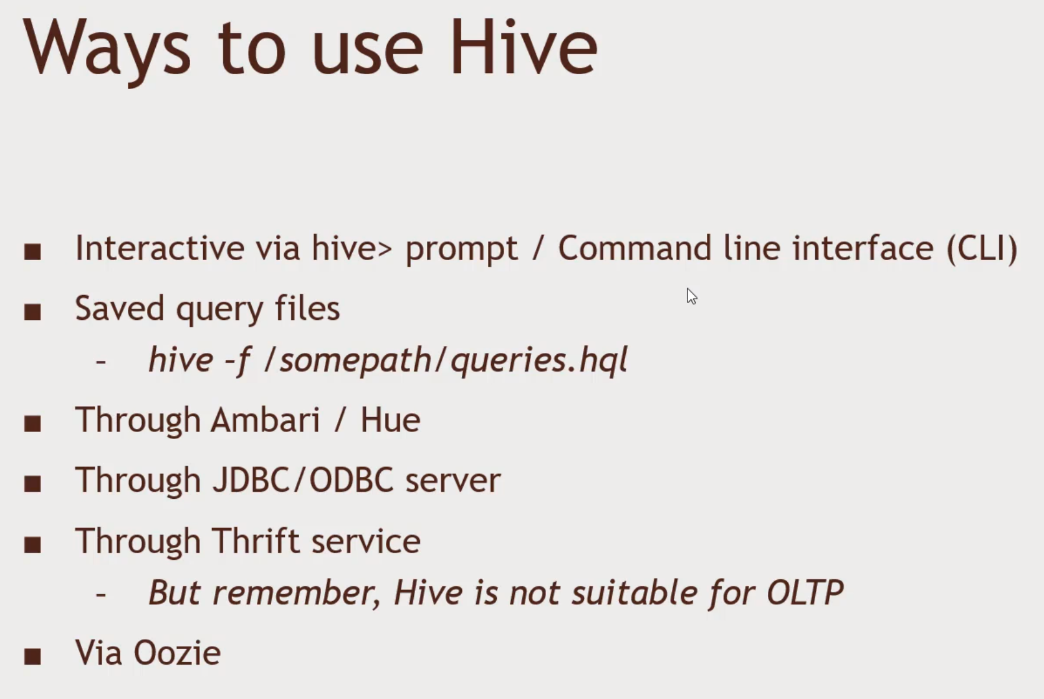
FROM topMovieIDs t JOIN names n ON t.movieID = n.movieID

DROP VIEW topmovieids;

HOW HIVE WORKS

The above method is the hardway, instead of using AMBARI.

External Table means Hive didn’t take ownership. Hive isn’t messing with the Original Data

Not restricted to the usual data type. STRUCT. Structured data in the HIVE databases.

HIVE Challenge, Find the Movie with the Highest Average Rating.

Hint: AVG() can be used on aggregated data, Consider movies with more than 10 ratings.

SQL Query:

CREATE VIEW topMovieIDs AS

SELECT movieID, AVG(rating) as avgRating, COUNT(movieID) as ratingCount

FROM ratings

GROUP BY movieID

ORDER BY avgRating DESC;

SELECT n.title, ratingCount

FROM topMovieIDs t JOIN names n ON t.movieID = n.movieID

WHERE ratingCount > 10;