COSC 416
NoSQL Databases

Apache Hive

Dr. Ramon Lawrence
University of British Columbia Okanagan
ramon.lawrence@ubc.ca

Apache Hive

Apache Hive is a query system for Hadoop designed to simplify complex, ad hoc queries.

Hive defines a language called *HiveQL* similar to SQL that is translated into a sequence of Map-Reduce programs.

- DDL - CREATE and DROP statements
- DML – SELECT, INSERT, UPDATE, DELETE

Speeds up writing Map-Reduce programs and improves performance rather than users writing code themselves.

Pig Latin was like relational algebra and HiveQL is like SQL.

HiveQL Basics

1) Terminate commands with a semi-colon.

2) Available data types: `TINYINT, SMALLINT, INT, BIGINT, BOOLEAN, FLOAT, DOUBLE, STRING, BINARY, TIMESTAMP, DECIMAL`.

3) Supports complex types including `ARRAY, MAP, STRUCT, UNIONTYPE`.

4) String literals can use either single quotes (') or double quotes ("). Hive uses C-style escaping within strings.

HiveQL CREATE TABLE

CREATE TABLE defines a schema over an existing directory of files or imports data file into Hive database. Syntax:

```
CREATE [EXTERNAL] TABLE table_name
  [(col_name data_type [COMMENT col_comment], ...)]
  [ [ROW FORMAT row_format] [STORED AS file_format] ]
  [LOCATION hdfs_path]
```

Example:

```
CREATE EXTERNAL TABLE Games (gid INT, gname STRING,
pubname STRING, releaseDate STRING, rating DOUBLE)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION '/user/rlawrenc/416/lab4/games';
```

- Defines a relation over the input files in the given directory.
- If do not use external table, data file is moved into hive folder on HDFS.
- Specifies a tab as a field specifier.

HiveQL ALTER TABLE

HiveQL supports **ALTER TABLE** command including:

- Rename table:

```
ALTER TABLE tableName RENAME TO newTableName
```

- Change column name, type, and position:

```
ALTER TABLE tableName CHANGE col_old_name col_new_name
  column_type [COMMENT comment] [FIRST|AFTER colName]
```

- Add or replace (remove) column:

```
ALTER TABLE tableName ADD|REPLACE
COLUMNS (col_name data_type [COMMENT col_comment], ...)
```

HiveQL DROP TABLE

DROP TABLE deletes the table definition and data. The data files are not deleted if the table was EXTERNAL.

```
DROP TABLE tableName
```

HiveQL Other DDL Commands

Other DDL commands available in Hive:

```
CREATE/DROP DATABASE dbName

CREATE/DROP VIEW viewName AS SELECT ...

CREATE/DROP FUNCTION funcName AS className

CREATE/DROP INDEX idxName ON TABLE tblName (colName)

SHOW DATABASES/TABLES/COLUMNS
```

HiveQL – INSERT, UPDATE, DELETE

INSERT:

```
INSERT INTO tableName
SELECT ...
```

LOAD:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE]
          INTO TABLE tablename
```

UPDATE: (no support)

DELETE: (no support)

HiveQL SELECT Statement

SELECT statement syntax:

```
SELECT [DISTINCT] <attrList>
FROM     <tableExpr>
[WHERE    <condition>]
[GROUP BY <attrList>]
[HAVING   <condition>]
[ORDER BY <attr> [ASC|DESC], … ]
[LIMIT    <number>]
```

HiveQL SELECT Statement Supported Syntax

Renaming and aliasing of columns and tables with optional AS:

```
SELECT gname AS gameName FROM Games G
```

Pattern match using LIKE:

```
SELECT gname FROM Games WHERE gname LIKE '%er'
```

Match any value in a set using IN:

```
SELECT gname FROM Games WHERE gid IN (1,2,3)
```

Determine if column IS NULL:

```
SELECT gname FROM Games WHERE pubName IS NULL
```

HiveQL Outer Joins

Supports joining tables in the `FROM` clause including outer joins.

Types: `FULL OUTER JOIN`, `LEFT OUTER JOIN`, `RIGHT OUTER JOIN`, `CROSS JOIN`, `LEFT SEMI JOIN`

- The keyword "`outer`" can be omitted for outer joins.

Note that joins in the WHERE clause are not supported.

HiveQL GROUP BY

GROUP BY is supported on any number of columns or expressions.

Standard aggregate functions of MIN, MAX, COUNT, SUM, AVG are all supported.

The **HAVING** clause is applied ***AFTER*** the `GROUP BY` clause and aggregate functions are calculated to filter the group result.

Subqueries

HiveQL only support subqueries in the FROM clause.

```
SELECT gid, gname
FROM   Games INNER JOIN (SELECT gid FROM PlayerGames) PG
       ON Games.gid = PG.gid
```

No subqueries are allowed in `SELECT`, `WHERE`, or `HAVING`.

Cannot use `EXISTS`, `IN`, `> ANY`, `> ALL`.

Conclusion

HiveQL is a powerful SQL-like interface for generating MapReduce programs on Hadoop.

Support both DDL (`CREATE`, `ALTER`, `DROP`) and DML (`SELECT`, `INSERT`, `UPDATE`, `DELETE`).

Objectives

Be able to write HiveQL SELECT queries given a schema and an English question.