



Impala SQL Support

Yue Chen

<http://linkedin.com/in/yuechen2>

<http://dataera.wordpress.com>

Cloudera Impala - SQL Supports

- Because Impala **uses the same metadata store** as Hive to record information about table structure and properties, Impala can access tables defined through the native Impala CREATE TABLE command, or tables created using the Hive data definition language (DDL).
- Impala supports data manipulation (DML) statements similar to the DML component of HiveQL.
- Impala provides many **built-in functions** with the same names and parameter types as their HiveQL equivalents

Cloudera Impala - SQL Supports

- Data Definition Languages (DDL)
- Data Manipulation Languages (DML)
- Impala Specified Languages (ISL)

Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

CREATE DATABASE Statement

usage:

```
create database first LOCATION hdfs_path;
```

```
use first;
```

CREATE TABLE Statement

usage:

```
CREATE EXTERNAL TABLE tab1 ( id INT, col_1 BOOLEAN, col_2 DOUBLE, col_3 TIMESTAMP  
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION  
'/user/cloudera/sample_data/tab1';
```

```
CREATE TABLE tab3 ( id INT, col_1 BOOLEAN, col_2 DOUBLE, month INT, day INT ) ROW  
FORMAT DELIMITED FIELDS TERMINATED BY ', ' [STORED AS file_format] ; file_format:  
PARQUET | TEXTFILE | AVRO | SEQUENCEFILE | RCFILE
```

Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

CREATE TABLE Statement

usage:

```
CREATE EXTERNAL TABLE tab1 ( id INT, col_1 BOOLEAN, col_2 DOUBLE, col_3  
TIMESTAMP ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION  
' /user/cloudera/sample_data/tab1' ;
```

```
CREATE TABLE tab3 ( id INT, col_1 BOOLEAN, col_2 DOUBLE, month INT,  
day INT ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ', ' '[STORED AS  
file_format] ;
```

file_format: PARQUET | TEXTFILE | AVRO | SEQUENCEFILE | RCFILE

Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

CREATE VIEW Statement

usage:

```
create view v1 as select * from t1; create view v2 as  
select c1, c3, c7 from t1;
```

```
create view v3 as select c1, cast(c3 as string) c3,  
concat(c4,c5) c5, trim(c6) c6, "Constant" c8 from t1;
```

```
create view v4 as select t1.c1, t2.c2 from t1 join t2 on  
t1.id = t2.id; create view some_db.v5 as select * from  
some_other_db.t1;
```

Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

ALTER TABLE Statement

usage:

```
create database d1; create database d2; create database d3;
```

```
use d1;
```

```
create table mobile (x int);
```

```
use d2; -- Move table from another database to the current one.
```

```
alter table d1.mobile rename to mobile; use d1;
```

```
alter table d2.mobile rename to d3.mobile; -- Move table from one  
database to another.
```

```
create table p1 (s string) partitioned by (month int, day int);
```

```
alter table p1 partition (month=1, day=1) set location
```

```
‘/usr/external_data/new_years_day’
```

Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

ALTER VIEW Statement

usage:

```
create table t1 (x int, y int, s string);
```

```
create table t2 like t1;
```

```
create view v1 as select * from t1;
```

```
alter view v1 as select * from t2;
```


Cloudera Impala - SQL Supports (DDL)

- Data Definition Language

Compute Stats Statement

usage:

```
show table stats t1;
```

```
show column stats t1;
```

```
compute stats t1;
```

Cloudera Impala - SQL Supports (DML)

- **DML Statements**

Impala supports data manipulation (DML) statements similar to the DML component of HiveQL.

Cloudera Impala - SQL Supports (DML)

- DML Statements
 - ✓ INSERT Statement
 - ✓ LOAD DATA Statement

Cloudera Impala - SQL Supports (DML)

- **DML Statements**

- ✓ INSERT Statement

Impala supports inserting into tables and partitions that you create with the Impala CREATE TABLE statement, or pre-defined tables and partitions created through Hive.

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- `INSERT INTO` to append data to a table.
- `INSERT OVERWRITE` to replace the data in a table.
- Copy data from another table using `SELECT` query.
- An optional `with` clause before the `INSERT` keyword, to define a subquery referenced in the `SELECT` portion.
- Create one or more new rows using constant expressions through `VALUES` clause.
- Specify the names or order of columns to be inserted, different than the columns of the table being queried by the `INSERT` statement.

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- Eg: `insert into table text_table select * from default.tab1`
- Eg: `insert overwrite table parquet_table select * from default.tab1`
- Eg: `insert into val_test_1 values (100, 99.9/10, 'abc', true, now());`
- Eg: `insert overwrite val_test_2 values (1, 'a'), (2, 'b'), (-1, 'xyzzzy');`
- Eg: `CREATE TABLE rc_table (id INT, col_1 BOOLEAN, col_2 DOUBLE, col_3
TIMESTAMP) STORED AS RCFILE;`

`insert into table rc_table select * from default.tab1;` Remote error
Backend 0:RC_FILE not implemented.

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- Eg: `insert into table text_table select * from default.tab1`
- Eg: `insert overwrite table parquet_table select * from default.tab1`
- Eg: `insert into val_test_1 values (100, 99.9/10, 'abc', true, now());`
- Eg: `insert overwrite val_test_2 values (1, 'a'), (2, 'b'), (-1, 'xyzzzy');`
- Eg: `CREATE TABLE rc_table (id INT, col_1 BOOLEAN, col_2 DOUBLE, col_3 TIMESTAMP) STORED AS RCFILE;`

`insert into table rc_table select * from default.tab1;` Remote error
Backend 0:RC_FILE not implemented.

Note: The above examples show the type of "not implemented" error that you see when attempting to insert data into a table with a file format that Impala currently does not write to

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- An optional hint clause immediately before the SELECT keyword, to fine-tune the behavior when doing an INSERT ... SELECT operation into partitioned Parquet tables. The hint keywords are [SHUFFLE] and [NOSHUFFLE], including the square brackets. Inserting into partitioned Parquet tables can be a resource-intensive operation because it potentially involves many files being written to HDFS simultaneously.

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- `select customer.address, state_lookup.state_name from customer join [broadcast] state_lookup on customer.state_id = state_lookup.state_id;`
- This query joins a large customer table with a small lookup table of less than 100 rows. The right-hand table can be broadcast efficiently to all nodes involved in the join. Thus, you would use the [broadcast] hint to force a broadcast join strategy.

Cloudera Impala - SQL Supports (DML)

- **DML Insert Statements**

- `select weather.wind_velocity, geospatial.altitude from weather join [shuffle] geospatial on weather.lat = geospatial.lat and weather.long = geospatial.long;`
- This query joins two large tables of unpredictable size. You might benchmark the query with both kinds of hints and find that it is more efficient to transmit portions of each table to other nodes for processing. Thus, you would use the `[shuffle]` hint to force a partitioned join strategy.

Cloudera Impala - SQL Supports (DML)

- **Loading Statements**

The LOAD DATA statement streamlines the ETL process for an internal Impala table by moving a data file or all the data files in a directory from an HDFS location into the Impala data directory for that table.

Cloudera Impala - SQL Supports (DML)

- Loading Statements

usage:

```
create table t1 (s string);  
load data inpath '/user/cloudera/thousand_strings.txt'  
into table t1;  
load data inpath '/user/cloudera/ten_strings.txt'  
overwrite into table t1;
```

Cloudera Impala - SQL Supports (DML)

- Loading Statements Notes:

Cloudera Impala - SQL Supports (DML)

- **Loading Statements Notes:**
- The loaded data files are moved, not copied, into the Impala data directory.

Cloudera Impala - SQL Supports (DML)

- **Loading Statements Notes:**
- The loaded data files are moved, not copied, into the Impala data directory.
- You can specify the HDFS path of a single file to be moved, or the HDFS path of a directory to move all the files inside that directory. You cannot specify any sort of wildcard to take only some of the files from a directory. When loading a directory full of data files, keep all the data files at the top level, with no nested directories underneath.

Cloudera Impala - SQL Supports (DML)

- **Loading Statements Notes:**
- The loaded data files are moved, not copied, into the Impala data directory.
- You can specify the HDFS path of a single file to be moved, or the HDFS path of a directory to move all the files inside that directory. You cannot specify any sort of wildcard to take only some of the files from a directory. When loading a directory full of data files, keep all the data files at the top level, with no nested directories underneath.
- Currently, the Impala LOAD DATA statement only imports files from HDFS, not from the local file system.

Cloudera Impala - SQL Supports (ISL)

- **Usage:**

`REFRESH table_name`

- **Notes:**

- Use the REFRESH statement to load the latest metastore metadata and block location data for a particular table in these scenarios:
- After loading new data files into the HDFS data directory for the table. (Once you have set up an ETL pipeline to bring data into Impala on a regular basis, this is typically the most frequent reason why metadata needs to be refreshed.)
- After issuing ALTER TABLE, INSERT, LOAD DATA, or other table-modifying SQL statement in Hive

Cloudera Impala - SQL Supports (ISL)

- **Usage:**

```
INVALIDATE METADATA [table_name]
```

- **Notes:**

- A metadata update for an impalad instance is required if:
 - A metadata change occurs.
 - and the change is made from another impalad instance in your cluster, or through Hive.
 - and the change is made to a database to which clients such as the Impala shell or ODBC directly connect.

Cloudera Impala - SQL Supports (ISL)

- **Usage:**

Show Statements:

SHOW DATABASES [[LIKE] '*pattern*']

SHOW SCHEMAS [[LIKE] '*pattern*'] - an alias for SHOW DATABASES

SHOW TABLES [IN *database_name*] [[LIKE] '*pattern*']

SHOW [AGGREGATE] FUNCTIONS [IN *database_name*] [[LIKE] '*pattern*']

SHOW CREATE TABLE [*database_name*].*table_name*

SHOW TABLE STATS [*database_name*.]*table_name*

SHOW COLUMN STATS [*database_name*.]*table_name*

SHOW PARTITIONS [*database_name*.]*table_name*

Cloudera Impala - SQL Supports (DML)

Impala does not support the following SQL features

- Non-scalar data types such as maps, arrays, structs.
- Extensibility mechanisms such as TRANSFORM, custom file formats.
- XML and JSON functions.
- Certain aggregate functions from HiveQL:
variance, var_pop, var_samp, stddev_pop, stddev_samp, covar_pop, covar_samp, corr, percentile, percentile_approx, histogram_numeric, collect_set.
- Sampling.
- Lateral views.
- Multiple DISTINCT clauses per query.

Cloudera Impala - SQL Supports (DML)

- Impala does not currently support these HiveQL statements:
- ANALYZE TABLE (the Impala equivalent is COMPUTE STATS)
- DESCRIBE COLUMN
- DESCRIBE DATABASE
- EXPORT TABLE
- IMPORT TABLE
- SHOW PARTITIONS
- SHOW TABLE EXTENDED
- SHOW INDEXES
- SHOW COLUMNS

References

- Cloudera Impala official documentation and slides
- http://www.cloudera.com/content/cloudera-content/cloudera-docs/Impala/latest/Installing-and-Using-Impala/ciiu_langref_sql.html