LanguageManual Select

GROUP BY; SORT/ORDER/CLUSTER/DISTRIBUTE BY; JOIN (Hive Joins, Join Optimization, Outer Join Behavior); UNION; TABLESAMPLE; Subqueries; Virtual Columns; Operators and UDFs; LATERAL VIEW; Windowing, OVER, and Analytics; Common Table Expressions

Select Syntax

```
[WITH CommonTableExpression (, CommonTableExpression)*]    (Note: Only available starting with Hive 0.13.0)
SELECT [ALL | DISTINCT] select_expr, select_expr, ...
  FROM table_reference
  [WHERE where_condition]
  [GROUP BY col_list]
  [ORDER BY col_list]
  [CLUSTER BY col_list
    | [DISTRIBUTE BY col_list] [SORT BY col_list]
  ]
 [LIMIT [offset,] rows]
```

- A SELECT statement can be part of a union query or a subquery of another query.
- `table_reference` indicates the input to the query. It can be a regular table, a view, a join construct or a subquery.
- Table names and column names are case insensitive.
  - In Hive 0.12 and earlier, only alphanumeric and underscore characters are allowed in table and column names.
  - In Hive 0.13 and later, column names can contain any Unicode character (see HIVE-6013). Any column name that is specified within backticks (`) is treated literally. Within a backtick string, use double backticks (``) to represent a backtick character.
  - To revert to pre-0.13.0 behavior and restrict column names to alphanumeric and underscore characters, set the configuration property `hive.support.quoted.identifiers` to `none`. In this configuration, backticked names are interpreted as regular expressions. For details, see Supporting Quoted Identifiers in Column Names (attached to HIVE-6013). Also see REGEX Column Specification below.
- Simple query. For example, the following query retrieves all columns and all rows from table t1.

```
SELECT * FROM t1
```

Note

As of Hive 0.13.0, FROM is optional (for example, `SELECT 1+1`).

- To get the current database (as of Hive 0.13.0), use the current_database() function:

```
SELECT current_database()
```

- To specify a database, either qualify the table names with database names ("`db_name.table_name`" starting in Hive 0.7) or issue the USE statement before the query statement (starting in Hive 0.6).

  "`db_name.table_name`" allows a query to access tables in different databases.

  USE sets the database for all subsequent HiveQL statements. Reissue it with the keyword "`default`" to reset to the default database.

```
USE database_name;
SELECT query_specifications;
USE default;
```

WHERE Clause

The WHERE condition is a boolean expression. For example, the following query returns only those sales records which have an amount greater than 10 from the US region. Hive supports a number of operators and UDFs in the WHERE clause:

```
SELECT * FROM sales WHERE amount > 10 AND region = "US"
```

As of Hive 0.13 some types of subqueries are supported in the WHERE clause.

ALL and DISTINCT Clauses

The ALL and DISTINCT options specify whether duplicate rows should be returned. If none of these options are given, the default is ALL (all matching rows are returned). DISTINCT specifies removal of duplicate rows from the result set. Note, Hive supports SELECT DISTINCT * starting in release 1.1.0 (HIVE-9194).

```
hive> SELECT col1, col2 FROM t1
    1 3
    1 3
    1 4
    2 5
hive> SELECT DISTINCT col1, col2 FROM t1
    1 3
    1 4
    2 5
hive> SELECT DISTINCT col1 FROM t1
    1
    2
```

ALL and DISTINCT can also be used in a UNION clause – see Union Syntax for more information.

Partition Based Queries

In general, a SELECT query scans the entire table (other than for sampling). If a table created using the PARTITIONED BY clause, a query can do **partition pruning** and scan only a fraction of the table relevant to the partitions specified by the query. Hive currently does partition pruning if the partition predicates are specified in the WHERE clause or the ON clause in a JOIN. For example, if table page_views is partitioned on column date, the following query retrieves rows for just days between 2008-03-01 and 2008-03-31.

```
    SELECT page_views.*
    FROM page_views
    WHERE page_views.date >= '2008-03-01' AND page_views.date <= '2008-03-31'
```

If a table page_views is joined with another table dim_users, you can specify a range of partitions in the ON clause as follows:

```
    SELECT page_views.*
    FROM page_views JOIN dim_users
      ON (page_views.user_id = dim_users.id AND page_views.date >= '2008-03-01' AND page_views.date <= '2008-03-
31')
```

- See also Partition Filter Syntax.
- See also Group By.
- See also Sort By / Cluster By / Distribute By / Order By.

HAVING Clause

Hive added support for the HAVING clause in version 0.7.0. In older versions of Hive it is possible to achieve the same effect by using a subquery, e.g:

```
SELECT col1 FROM t1 GROUP BY col1 HAVING SUM(col2) > 10
```

can also be expressed as

```
SELECT col1 FROM (SELECT col1, SUM(col2) AS col2sum FROM t1 GROUP BY col1) t2 WHERE t2.col2sum > 10
```

LIMIT Clause

The LIMIT clause can be used to constrain the number of rows returned by the SELECT statement.

LIMIT takes one or two numeric arguments, which must both be non-negative integer constants.

The first argument specifies the offset of the first row to return (as of Hive 2.0.0) and the second specifies the maximum number of rows to return.

When a single argument is given, it stands for the maximum number of rows and the offset defaults to 0.

The following query returns 5 arbitrary customers

```
SELECT * FROM customers LIMIT 5
```

The following query returns the first 5 customers to be created

```
SELECT * FROM customers ORDER BY create_date LIMIT 5
```

The following query returns the 3rd to the 7th customers to be created

```
SELECT * FROM customers ORDER BY create_date LIMIT 2,5
```

REGEX Column Specification

A SELECT statement can take regex-based column specification in Hive releases prior to 0.13.0, or in 0.13.0 and later releases if the configuration property `hive.support.quoted.identifiers` is set to `none`.

- We use Java regex syntax. Try http://www.fileformat.info/tool/regex.htm for testing purposes.
- The following query selects all columns except ds and hr.

```
SELECT `(ds|hr)?+.+` FROM sales
```

More Select Syntax

See the following documents for additional syntax and features of SELECT statements:

- GROUP BY
- SORT/ORDER/CLUSTER/DISTRIBUTE BY
- JOIN
    - Hive Joins
    - Join Optimization
    - Outer Join Behavior
- UNION
- TABLESAMPLE
- Subqueries
- Virtual Columns
- Operators and UDFs
- LATERAL VIEW
- Windowing, OVER, and Analytics
- Common Table Expressions