The ORDER BY Clause in Views

The stored query for a view can be any query—it can use any of the allowed clauses of a SELECT statement. However, using the ORDER BY clause in a view's stored query is not recommended. Sorting (arranging) result rows in order is an action best performed in the query on the view, not in the view's stored query.

To understand why this is, recall that Hive and Impala are designed to distribute query processing work across large clusters of computers. Some tasks (like filtering rows) can easily be performed in parallel on many rows distributed across these many computers. But the task of efficiently *sorting* many rows typically requires consolidating all the rows on just one or a few computers. This makes sorting rows a slow and inefficient operation; sorting is typically the bottleneck of any query that uses an ORDER BY clause. Furthermore, preserving the sort order through later query operations forces those later operations also to be slow and inefficient.

So if it is necessary to sort a result set, the sort operation should be performed *last,* after the other operations such as filtering. Because the queries *on* a view will often perform further operations, including filtering, the query stored *in* a view should *not* perform sorting; doing this would cause major inefficiencies.

Impala and newer versions of Hive (version 3.0.0 and higher, which is newer than the version on the course VM) prevent these inefficiencies from occurring by ignoring the ORDER BY clause when it is used in a view's stored query. Impala will issue a warning to inform you of this when you query a view that uses ORDER BY in its stored query. However, some applications do not display this warning. Impala Shell displays it prominently, but Hue's Impala query editor does not; you need to click Show Logs to see it in Hue. Some other applications do not display the warning at all.

Newer versions of Hive will silently ignore the ORDER BY clause in a view's stored query and will not issue any warning. Older versions of Hive (like the one on the VM) will respect the ORDER BY clause in a view's stored query and will incur the associated inefficiencies.

The exception to all of this is when the ORDER BY clause is used together with the LIMIT clause in a view’s stored query. If a view’s stored query uses ORDER BY and LIMIT *n*, then the sorting operation is much less likely to be a bottleneck, because Hive and Impala can efficiently identify the top *n* or bottom *n* rows (if *n* is fairly small—and it typically is).

So if a view’s stored query uses ORDER BY together with LIMIT, then Impala and newer versions of Hive will *not* ignore the ORDER BY clause.

Try It!

1. Using Hive (either in Hue or using Beeline on the command line), make the fly database your active database.

2. Do a quick SELECT * FROM planes LIMIT 20; to see what the first 20 rows of the planes table looks like.

3. Create a view of the planes table with all the columns, ordering by year in descending order but without a LIMIT clause (and omitting any rows where year is NULL). You can name it whatever you like. Here's the syntax, just to remind you:

```
CREATE VIEW viewname AS
    SELECT * FROM planes
        WHERE year IS NOT NULL
        ORDER BY year DESC;
```

4. Query the view just using:

```
SELECT * FROM viewname LIMIT 20;
```

Notice that the result set is indeed sorted by year (all the results should be from 2018). Also note how long it took to finish (this will matter in the next step). The time is reported in the top right of the query window, next to the active database.

5. Query the view again, but this time sort your query by tailnum:

    SELECT * FROM *viewname* ORDER BY tailnum LIMIT 20;

    Notice that the results are not all from 2018, and the query took probably almost twice as long, because it had to sort twice!

6. Now try it in Impala. First, go to Impala in Hue or using Impala Shell on the command line, and make fly the active database. Then execute:

    INVALIDATE METADATA *viewname*;

    so Impala will see the view you created in Hive.

7. Query the view just using:

    SELECT * FROM *viewname* LIMIT 20;

    a. When you did this in Hive, you got only planes from 2018; what are the results this time?

    b. Can you see the warning message indicating that the ORDER BY clause in the view has no effect on the query result? If you are using Hue, click the Show Logs button on the upper right; the warning should be visible at the bottom of the logs.

8. You can drop the view if you like.