Materialized and Non-Materialized Views

Views in Hive and Impala are typically *non-materialized.* This means that a view does not store or persist any data. It stores only a query.

When you query a view, Hive or Impala generates the result set on the fly by running the view's stored query on the underlying tables, then running your query on the result of the stored query.* It does this each time you run the query. For a complicated and computationally expensive query, this can take some time—that's the major disadvantage of non-materialized views. The major advantage is that whenever the data is changed in any of the underlying tables used in the stored query, the view will use the new data.

A *materialized* view, on the other hand, would store the data, so that the SQL engine does not need to run a query on the underlying tables every time the view is queried. This can save time. However, if the data is changed in any of the underlying tables, a materialized view will *not* use the new data; you would need to rebuild the view first.

At this time, Impala does not support materialized views, though this is something the developers are considering. (See [IMPALA-3446] for updates on this.) Hive has recently added materialized views with Hive 3.0.0. (See [HIVE-10459] and the Hive Materialized views page.) However, the VM provided for this course uses an older version of Hive, so you cannot create materialized views on the VM.

* This is essentially what happens, but in practice Hive and Impala might optimize this process by combining the operations in your query with the operations in the stored query into one single set of operations so that the result can be generated as efficiently as possible.