Session 8

Assignment 3

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 8 – Assignment 1 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 8 – Assignment 3** |
|  |  |
| **Start Date:** | 23/09/2017 |
|  |  |
|  |  |



# 

# Contents

[Contents 2](#_Toc493955620)

[Change History 3](#_Toc493955621)

[1. Problem Statement 4](#_Toc493955622)

[2. Transactions in Hive 5](#_Toc493955623)

[2.1. Preparing the environment 5](#_Toc493955624)

[2.2. Creating a Table That Supports Hive Transactions 6](#_Toc493955625)

[2.3. Inserting Data into a Hive Table 6](#_Toc493955626)

[2.4. Updating the Data in Hive Table 7](#_Toc493955627)

[2.5. Deleting a Row from Hive Table 7](#_Toc493955628)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 23/09/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

**Link** https://acadgild.com/blog/transactions-in-hive/

Refer the above given link for transactions in Hive and implement the operations given in the blog using your own sample data set and send us the screenshot.

# Transactions in Hive

Transactions in Hive are introduced in Hive 0.13, but they only partially fulfill the ACID properties like atomicity, consistency, durability, at the partition level. Here, Isolation can be provided by turning on one of the locking mechanisms available with zookeeper or in memory.

But in Hive 0.14, new API’s have been added to completely fulfill the ACID properties while performing any transaction.

Transactions are provided at the row-level in Hive 0.14. The different row-level transactions available in Hive 0.14 are as follows:

1. Insert
2. Delete
3. Update

There are numerous limitations with the present transactions available in Hive 0.14. ORC is the file format supported by Hive transaction. It is now essential to have ORC file format for performing transactions in Hive. The table needs to be bucketed in order to support transactions.

## Preparing the environment

**Row-level Transactions Available in Hive 0.14**

Let’s perform some row-level transactions available in Hive 0.14. Before creating a Hive table that supports transactions, the transaction features present in Hive needs to be turned on, as by default they are turned off.

The below properties needs to be set appropriately in **hive shell** , order-wise to work with transactions in Hive:

*hive> show databases;*

*OK*

*default*

*Time taken: 4.344 seconds, Fetched: 1 row(s)*

**This exercise will be completed using the default database.**

*hive> set hive.support.concurrency = true;*

*hive> set hive.enforce.bucketing = true;*

*hive> set hive.exec.dynamic.partition.mode = nonstrict;*

*hive> set hive.txn.manager = org.apache.hadoop.hive.ql.lockmgr.DbTxnManager;*

*hive> set hive.compactor.initiator.on = true;*

*hive> set hive.compactor.worker.threads = 8;*

If the above properties are not set properly, the ‘Insert’ operation will work but ‘Update’ and ‘Delete’ will not work and you will receive the following error:

**FAILED: SemanticException [Error 10294]: Attempt to do update or delete usingtransaction manager thatdoes not support these operations.**

## Creating a Table That Supports Hive Transactions

The command below will create a table with name ‘*college’* and the columns present in the table are ‘*clg\_id, clg\_name, clg\_loc’. W*e are *bucketing* the table by ‘*clg\_id’* and the table format is ‘*orc’,* also we are enabling the transactions in the table by specifying it inside the *TBLPROPERTIES* as ‘*transactional’=’true’*

*hive> CREATE TABLE college(clg\_id int,clg\_name string,clg\_loc string) clustered by (clg\_id) into 5 buckets stored as orc TBLPROPERTIES('transactional'='true');*

*OK*

*Time taken: 0.446 seconds*

We have successfully created a table with name ‘*college’* which supports row-level transactions of Hive.

The create table can be checked using the command ***show tables.***

## Inserting Data into a Hive Table

The command below is used to insert row wise data into the Hive table. Here, each row is separated by ‘*( )’ brackets.*

hive> INSERT INTO table college values(1,'dunc','Big Data'),(2,'Mark','Linux'),(3,'Mike','Windows'),(4,'Ben','SCCM'),(5,'Rosario','Exchange'),(6,'Tracey','Manager'),(7,'Chris','Ping');

*Loading data to table default.college*

*OK*

**To display the data**

*hive> select \* from college;*

*OK*

*5 Rosario Exchange*

*1 dunc Big Data*

*6 Tracey Manager*

*2 Mark Linux*

*7 Chris Ping*

*3 Mike Windows*

*4 Ben SCCM*

*Time taken: 0.329 seconds, Fetched: 7 row(s)*

The same data is inserted again using the above command the result is

*hive> select \* from college;*

*OK*

*5 Rosario Exchange*

*5 Rosario Exchange*

*1 dunc Big Data*

*6 Tracey Manager*

*1 dunc Big Data*

*6 Tracey Manager*

*2 Mark Linux*

*7 Chris Ping*

*2 Mark Linux*

*7 Chris Ping*

*3 Mike Windows*

*3 Mike Windows*

*4 Ben SCCM*

*4 Ben SCCM*

*Time taken: 0.176 seconds, Fetched: 14 row(s)*

## Updating the Data in Hive Table

The command below is used to update a row in Hive table.

*Hive> UPDATE college set clg\_id = 8 where clg\_id = 7;*

In this table, we have bucketed the **‘clg\_id’** column and performing the Update operation on the same column, so we have go the error

***FAILED: SemanticException[Error 10302]: Updating values of bucketing columns is not supported. Column clg\_id***

Performed on a non-bucketed column

*hive> UPDATE college set clg\_name = 'INSERTED' where clg\_id = 6;*

**The results**

*OK*

*5 Rosario Exchange*

*5 Rosario Exchange*

*1 dunc Big Data*

*6 INSERTED Manager*

*1 dunc Big Data*

*6 INSERTED Manager*

*2 Mark Linux*

*7 Chris Ping*

*2 Mark Linux*

*7 Chris Ping*

*3 Mike Windows*

*3 Mike Windows*

*4 Ben SCCM*

*4 Ben SCCM*

*Time taken: 0.148 seconds, Fetched: 14 row(s)*

## Deleting a Row from Hive Table

The above below will delete a single row in the Hive table.

*hive> delete from college where clg\_id=6;*

**The results**

*hive> select \* from college;*

*OK*

*5 Rosario Exchange*

*5 Rosario Exchange*

*1 dunc Big Data*

*1 dunc Big Data*

*2 Mark Linux*

*7 Chris Ping*

*2 Mark Linux*

*7 Chris Ping*

*3 Mike Windows*

*3 Mike Windows*

*4 Ben SCCM*

*4 Ben SCCM*

*Time taken: 0.172 seconds, Fetched: 12 row(s)*

We can see that there is no row with ****clg\_id =6****. This means that we have successfully deleted the row from the Hive table.

This is how the transactions or row-wise operations are performed in Hive.