

A **transaction** is a very small unit of a program and it may contain several low level tasks. A **transaction** in a database system must maintain Atomicity, Consistency, Isolation, and Durability – commonly known as ACID properties – in order to ensure accuracy, completeness, and data integrity. Now since HIVE is a DATAWARE HOUSE tool so to enhance its capabilities 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.

Now ACID stands for Atomicity, Consistency, Isolation, and Durability.

Atomicity means, a transaction should complete successfully or else it should fail completely i.e. it should not be left partially. Consistency ensures that any transaction will bring the database from one valid state to another state. Isolation states that every transaction should be independent of each other i.e. one transaction should not affect another. And Durability states that if a transaction is completed, it should be preserved in the database even if the machine state is lost or a system failure might occur.

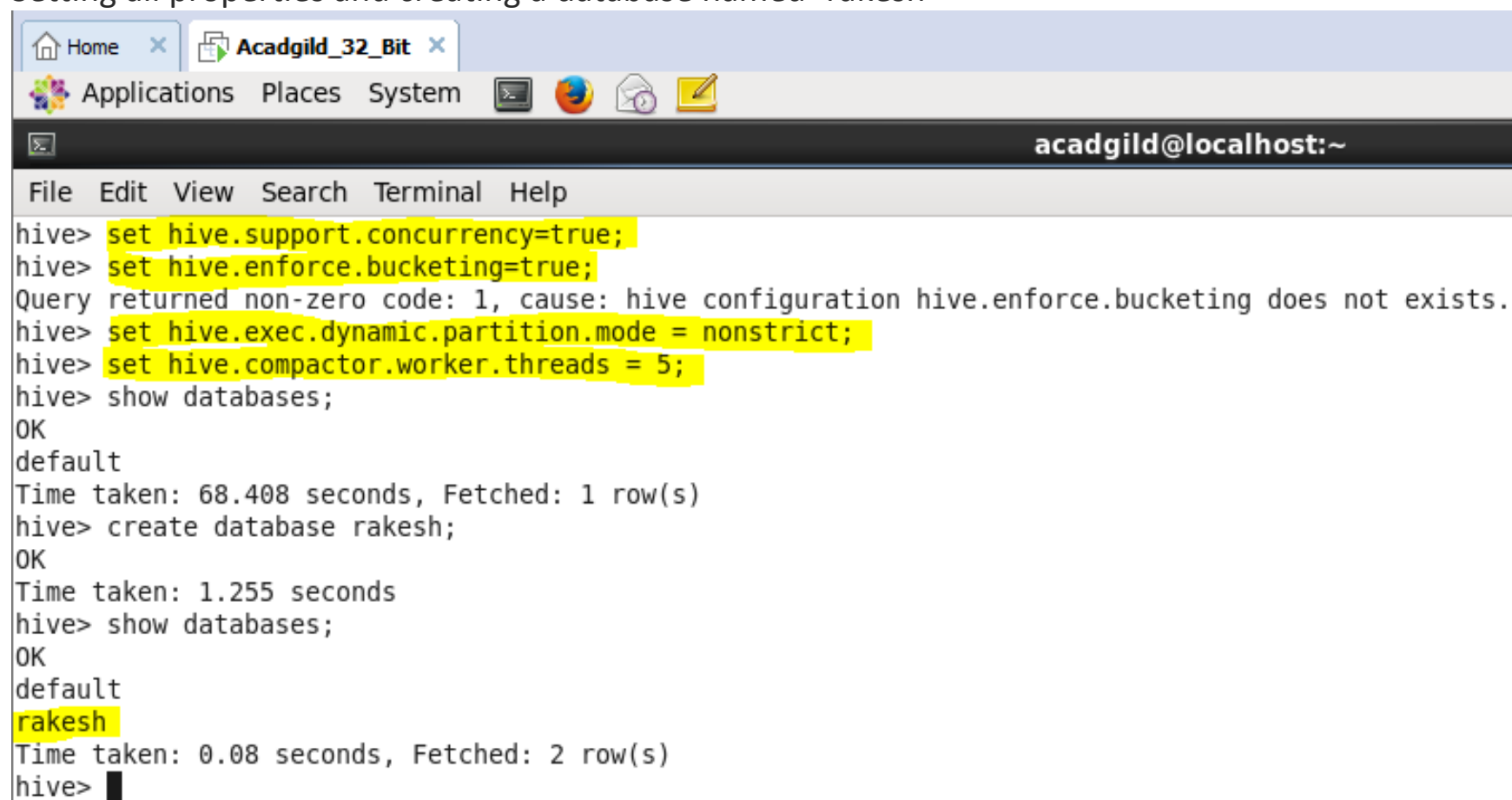
These ACID properties are essential for a transaction and every transaction should ensure that these properties are met.

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

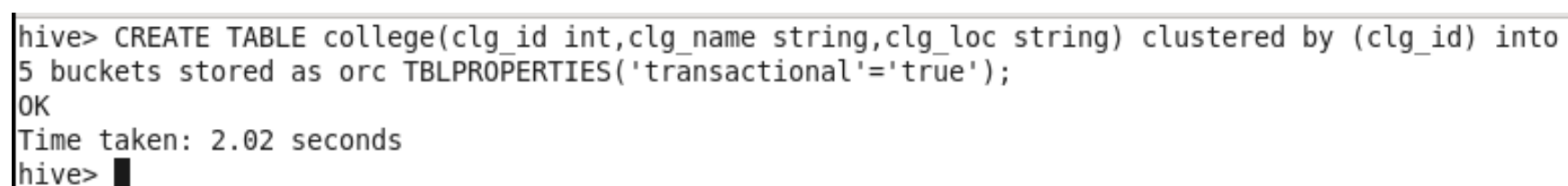
Now in order to work we will perform any transaction below properties need to be set in hive shell-

Setting all properties and creating a database named- rakesh

A screenshot of a terminal window titled 'Acadgild_32_Bit'. The terminal shows a series of Hive commands and their outputs. The commands are: 'set hive.support.concurrency=true;', 'set hive.enforce.bucketing=true;', 'set hive.exec.dynamic.partition.mode = nonstrict;', 'set hive.compactor.worker.threads = 5;', 'show databases;', 'create database rakesh;', and 'show databases;'. The outputs show the current settings, a warning about the bucketing property, the list of databases (default), and the newly created database 'rakesh'.

```
hive> set hive.support.concurrency=true;
hive> set hive.enforce.bucketing=true;
Query returned non-zero code: 1, cause: hive configuration hive.enforce.bucketing does not exists.
hive> set hive.exec.dynamic.partition.mode = nonstrict;
hive> set hive.compactor.worker.threads = 5;
hive> show databases;
OK
default
Time taken: 68.408 seconds, Fetched: 1 row(s)
hive> create database rakesh;
OK
Time taken: 1.255 seconds
hive> show databases;
OK
default
rakesh
Time taken: 0.08 seconds, Fetched: 2 row(s)
hive>
```

We will create a table named “college” first which supports hive transactions-

A screenshot of a terminal window showing the creation of a Hive table. The command is: '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');'. The output shows 'OK' and the time taken: 2.02 seconds.

```
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: 2.02 seconds
hive>
```

Now we will insert some data inside the table “college”

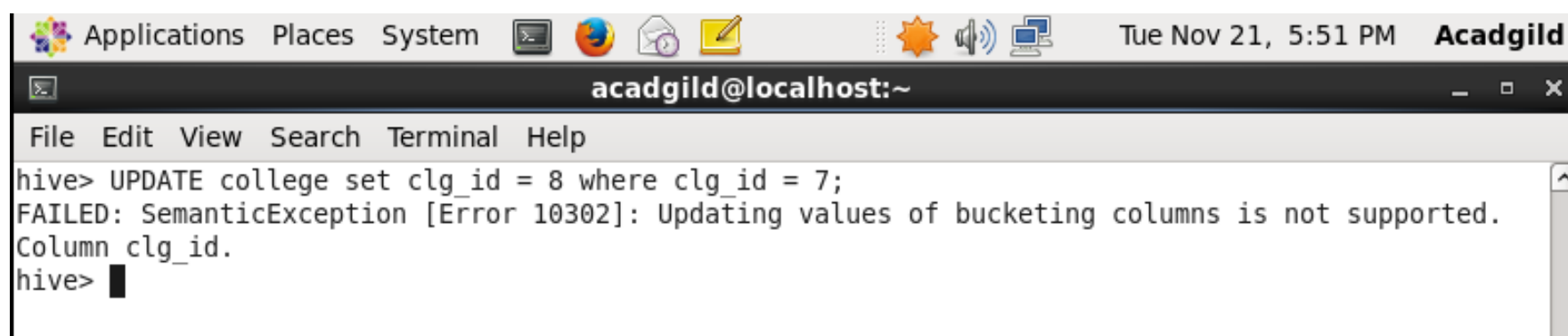
```
hive> INSERT INTO table college values(1,'nec','nlr'),(2,'vit','vlr'),(3,'srm','chen'),(4,'lpu','del'),(5,'stanford','uk'),(6,'JNTUA','atp'),(7,'cambridge','us');
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20171121173636_6fd87f52-fa56-4da5-8c5c-d8e9f6a56e0f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
```

```
hive> select * from college;
OK
5      stanford      uk
6      JNTUA      atp
1      nec      nlr
7      cambridge      us
2      vit      vlr
3      srm      chen
4      lpu      del
Time taken: 0.289 seconds, Fetched: 7 row(s)
hive>
```

From the above image, we can see that we have received an error message. This means that the Update command is not supported on the columns that are bucketed.

In this table, we have bucketed the ‘clg_id’ column and performing the Update operation on the same column, so we have got the error

FAILED: SemanticException[Error 10302]: Updating values of bucketing columns is not supported. Column clg_id

A screenshot of a terminal window titled 'acadgild@localhost:~'. The window shows a Hive command 'UPDATE college set clg_id = 8 where clg_id = 7;' followed by an error message: 'FAILED: SemanticException [Error 10302]: Updating values of bucketing columns is not supported. Column clg_id.' The terminal also shows the standard Hive warning about Hive-on-MR being deprecated and the query ID. The window's title bar includes system icons and the date 'Tue Nov 21, 5:51 PM'.

Now we will perform update operation on non-bucketed column-

```
hive> UPDATE college set clg_name = 'IIT' where clg_id = 6;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20171121174340_f062d6f8-cafc-4eba-a924-434e722d49b7
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
```

The updated data can be checked using the command select * from college.

```
hive> UPDATE college set clg_id = 8 where clg_id = 7;
FAILED: SemanticException [Error 10302]: Updating values of bucketing columns is not supported.
Column clg_id.
hive> select * from college;
OK
5      stanford      uk
6      IIT      atp
1      nec      nlr
7      cambridge      us
2      vit      vlr
3      srm      chen
4      lpu      del
Time taken: 0.214 seconds, Fetched: 7 row(s)
hive> █
```

We can also delete from the table “college”-

```
hive> delete from college where clg_id=5;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20171121175440_b2e30965-3bf3-496a-9d2e-899df41cf29d
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 5
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
```

Row with college_id is deleted from table-

	File	Edit	View	Search	Terminal	Help
--	------	------	------	--------	----------	------

```
hive> select * from college;
OK
6      IIT      atp
1      nec      nlr
7      cambridge      us
2      vit      vlr
3      srm      chen
4      lpu      del
Time taken: 0.223 seconds, Fetched: 6 row(s)
hive> █
```