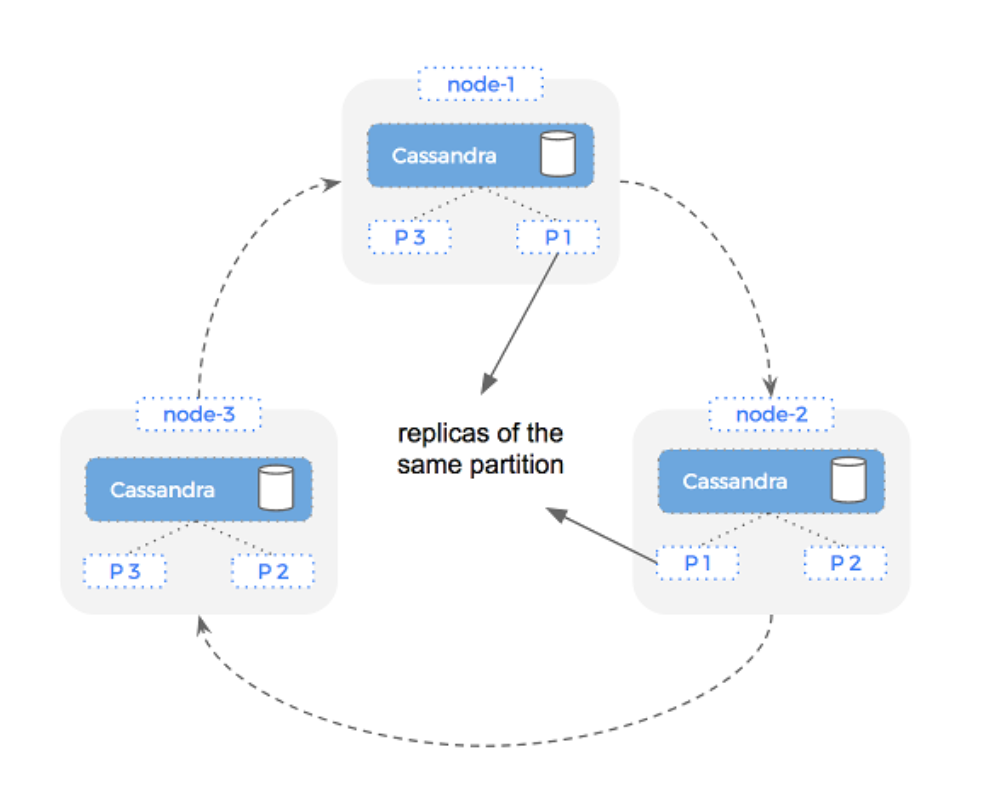
CSE 512- Distributed Database Systems

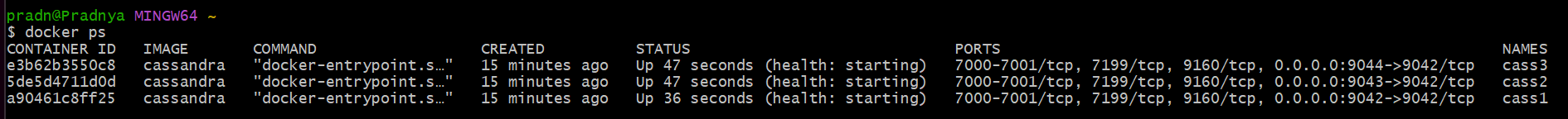
Group Project: Cassandrian

Part 5: Distributed NoSQL Database Systems Implementation

Based on the sample database found on Kaggle, the most suitable NoSQL database to cater to the requirements was **Cassandra**, because of its column-based database.

We created a Cassandra cluster of three nodes using docker, namely cass1, cass2, cass3. Cass1 & cass2 are seed nodes, i.e., cass1 & cass2 will the point of contact for the cluster and also contain all the information about the cluster. The distributed nature of Cassandra cluster ensures high availability.







The replication factor for the “commentary\_keyspace” is set to 2, which would create 2 replicas of each partition of the keyspace across the cluster.

**Data Schema & Data Model –**

Table – commentary

Commentary table stores commentary information for each ball in a match.

CREATE TABLE IF NOT EXISTS commentary (  
 Over\_No TEXT,  
 Over\_Score TEXT,  
 Short\_comm TEXT,  
 Commentary TEXT,  
 Bold\_Comm TEXT,  
 Innings\_ID TEXT,  
 Ball\_ID TEXT,  
 Match\_ID TEXT,  
 PRIMARY KEY (Innings\_ID, Ball\_ID)  
 );

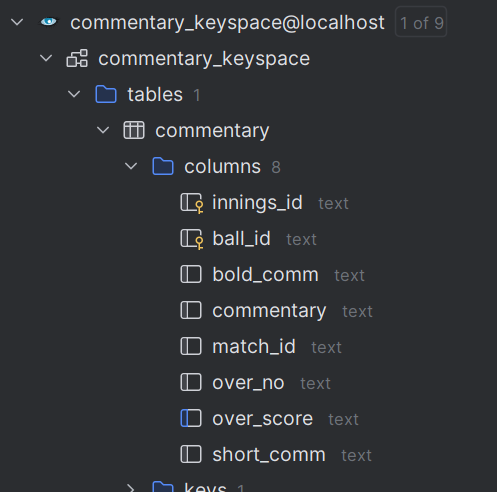
* Over\_No: Over number (1.1, 3.5, etc)
* Over\_score: Number of runs scored for that ball along with indication of wicket, byes, etc
* Short\_comm: Short Commentary
* Commentary: Long version of commentary with all details
* Bold\_comm: Comments made for entertainment
* Innings\_ID: Identifies the innings of the match. It is a combination of match\_ID & inning number eg. 1181768-1, 1181768-2
* Ball\_ID: Identifies each ball within an innings
* Match\_ID: References the match in which the commentary occurs

Innings\_ID & Ball\_ID together make the composite primary key.

**CRUD operations –**

1. Create Commentary table:

Create\_table() method creates commentary table using the “create table” query



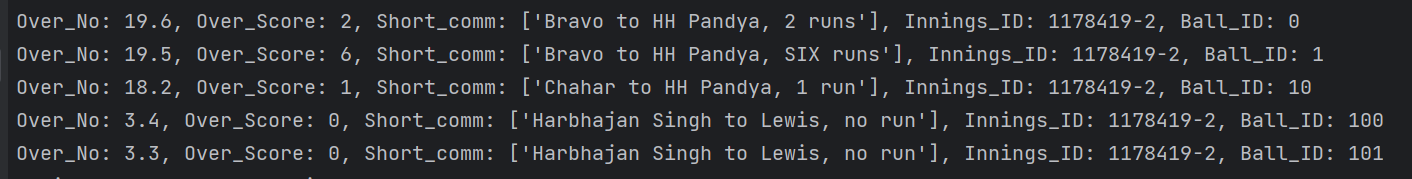
1. Insert the sample data:

Insert\_csv\_file\_data() method inserts the data from sample csv file – ipl2019\_final.csv into commentary table.



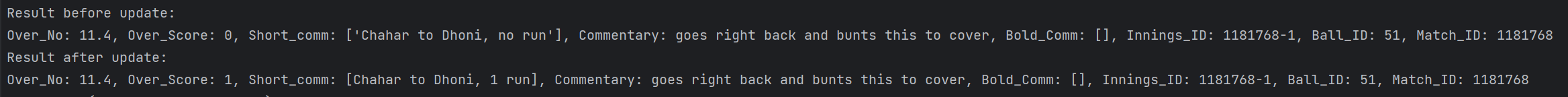
1. Sample read query:

Read\_query() method selects over\_no, over\_score, short\_comm, innings\_ID, ball\_ID of 5 entries.



1. Update query:

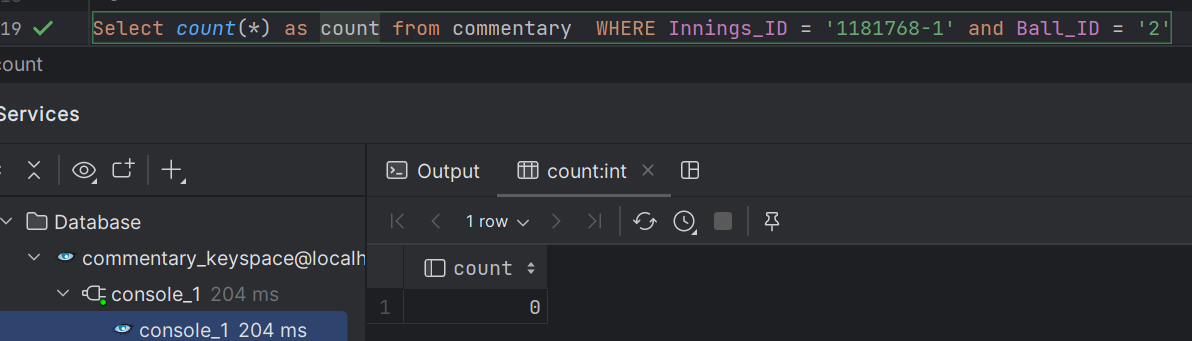
Update\_query() updates over\_score & short\_comm value of the entry with innings\_ID ‘1181768-1’ and ball\_ID ‘51’



1. Delete query:

Delete\_query() method deletes entry with innings\_ID ‘1181768-1’ and ball\_ID ‘2’





**Sample Queries to show data retrieval operations:**

1. Get sixes by innings:

Get\_sixes\_by\_innings() method get the total number of sixes in every inning. As the where clause of this query contains a non-primary column over\_score, a index needs to be created on that column, so as to not affect the computation capability of Cassandra. Create\_index() is the method that creates index on over\_score column.

