# Context Based Cassandra Query Language

Shivendra Kumar Pandey School of Computer and Systems Sciences Jawaharlal Nehru University New Delhi, India shivaiert@gmail.com

data storage and distributed computing and therefore, do no

Sudhakar

Indian Computer Emergency Response Team

Ministry of Electronics & Information Technology

New Delhi, India

sudhak82 scs@inu.ac.in

Abstract— NoSQL databases are distributed, non-relational databases designed for large-scale data storage and for parallel data processing across a large number of commodity servers. Cassandra is a NoSQL database that stores data in non-related tabular forms. Cassandra works on "query at a time" and "query at a table" concept. In our daily life the queries of a user are related to each other. If queries by a user are related, that is, the current query is related to the previous query, then there is no support in Cassandra to state this. Cassandra runs each query on the entire table because Cassandra has neither memory to remember the result of a previous query, nor supports VIEW or JOIN on tables (or keyspace). Cassandra uses Big Table (of Google) for data storage which has thousands of columns and millions of rows, so it is not efficient to run the query on such a huge table, after knowing that the result of the previous query is sufficient to answer our current query. To solve such problems of Cassandra database, we implemented a new query language named as "Context Based Cassandra Query Language". CBCQL is internally mapped to Cassandra Query Language (CQL) so it has the same power as Cassandra, but provides additional functionality of querying on result of previous query. In this paper, CBCQL is explained with examples.

Keywords—Cassandra; database; NoSQL; CQL; big data; query language; VIEW; JOIN; CBCOL.

# I. INTRODUCTION

With the development of technology and internet users, there is a need for a system that can manage data efficiently and provide high performance [1]. Relational databases are facing many challenges, especially in scaling, concurrency and in providing write throughput [2]. To solve these problems, a new type of non-relational database management system was developed. This system is known as NoSQL. NoSQL databases are highly scalable, non-relational databases and provide high read/write throughput. They support Big Data and can run on a cheap commodity server. Big Data is a heterogeneous mixture of structured, semi structured and unstructured data [3]. NoSQL is an abbreviation of "Not only SQL" [4] [5]. These databases are very popular in companies for their cost, performance, and scalability.

In our world, we see data as a large heterogeneous collection of structured and unstructured data [6]. The main idea behind NoSQL databases is that they can store and retrieve structured (any relational database that has some schema), semi structured (XML or CSV file) and unstructured data (pdf, doc, email) efficiently [7]. They support distributed

data storage and distributed computing and therefore, do not have a single point of failure [8].

In this work we have used Cassandra database. Cassandra is a distributed, column oriented, NoSQL database with high scalability, high availability and provides high performance with no single point of failure. Cassandra is the best choice for the companies that need reliability, high availability and very fast performance. Cassandra has very write throughput and good read throughput with flexible schema. Cassandra uses Big Table's data model of Google for data storage and the data distribution concept of Amazon Dynamo [9].

### A. Motivation

CQL is used to access Cassandra database. CQL is essentially 'query-at-a-time' language. That is, each query is executed, the result is given to the user and has no bearing on the next query. We believe, that users tend to ask a series of related queries which is dictated by a thought process. Consider an example. Suppose a user wants to buy a flat or home and he wants to select the best suitable one. He/she will execute a CQL query for finding the flats. A typical table in Cassandra has thousands of columns and millions of rows. As a consequence, the result will also contain millions of rows. In such a situation, it will be very difficult for the customer to select a flat or a home as per his need in such a huge table. Since Cassandra does not have memory to remember the result of the previous query, so every time user has to query the whole table. Consider the example given below:

- 1) Select price, baths, beds, city, area, parking\_lot, placeid, rpayment, sq\_\_ft, type;
- 2) Select price, baths, beds, city, area, parking\_lot, placeid, rpayment, sq\_\_ft, type WHERE type = 'Residential';
- 3) Select price, baths, beds, city, area, parking\_lot, placeid, rpayment, sq\_\_ft WHERE type = 'Residential' and beds = 3;
- 4) Select price, baths, beds, city, area, parking\_lot, placeid, rpayment, sq\_\_ft WHERE type = 'Residential' and beds = 3 and baths > 1;
- 5) Select city, price, area, parking\_lot, placeid, sq\_\_ft WHERE type = 'Residential' and beds = 3 and baths > 1 and parking lot='yes';
- 6) Select city ,price, area, parking\_lot, placeid, sq\_ft WHERE type = 'Residential' and beds = 3 and baths > 1 and parking lot='yes' and city='SACRAMENTO';

- 7) Select area, price, placeid, sq\_ft WHERE type = 'Residential ' and beds = 3 and baths > 1 and parking\_lot='yes' and city='SACRAMENTO' and area='open';
- 8) Select area, price, placeid, sq\_ft WHERE type = 'Residential' and beds = 3 and baths > 1 and parking\_lot='yes' and city='SACRAMENTO' and area='open' and price < 10000;

The above sequence of queries reflects the thought process of the user. In the first query user selects the columns that are important to him. In subsequent queries, he specifies some conditions to get the best suitable deal. It is clear from the queries that there is no need to search the entire database every time. We have to just reduce the number of rows by putting conditions on columns. But Cassandra does not support this functionality. We have tried to overcome this limitation by proposing CBCQL. Our proposed prototype has the capability of creating, saving, recalling and deleting the context.

## II. RELATED WORK

A lot of work has been done in the field of non-relational database. The term NoSQL was first used in 1998 for a relational database that omitted the use of SQL [10]. But now a day, the term NoSQL is used to differentiate non-relational databases from relational databases. NoSQL databases are providing high performance, but they have a lot of security issues. Till now we have three (or four by Tudorica [11]) main types of NoSQL databases [4] [10]. Most of the NoSQL databases are non-relational, query-at-a-time and query-at-atable. Hence the concept of context can be used to improve performance, and make them user friendly. The notion of context as defined here has been proposed earlier in [12], [13] and [14]. In [12], context is defined for a network query language and in [13], a component based query language includes the definition of a context. Stream based query language incorporated context in [14]. However, none of these uses Context within the framework of CQL. We have used the concept of context in our work to facilitate analysis of data where the result of previous query is sufficient to answer the present query of the user. The context in our proposed work is designed in such a way that it fetches result from the context and updates the context with the result. We are providing the facility of saving a context and recalling it, so backtracking will be also easy for the user while querying. To reduce the typing mistakes, we made the CBCQL case insensitive.

# III. CONTEXT BASED CASSANDRA QUERY LANGUAGE

In CBCQL, a query is similar to a CQL query, but the FROM clause is not present. The data to be picked up is available in the context and thus, the FROM clause is done away with. Every query is executed in the current context. It, in turn, updates the context which forms the context for the subsequent query. In addition, constructs to save and restore

context are also defined. This additional functionality allows the user to go back in the sequence of queries and follow a different path for querying.

#### A. Context

A context consists of the table of interest and the data corresponding to it. Since Cassandra does not support JOIN operations on tables, it is not desirable for us to have more than one table in a context. Initially, the context is null. It contains no table or data. After creating a context, the first command of the user, is to add a table. The table and the data in the table, now define the context for the first query.

There are six types of queries in CBCQL. To reduce the possibility of errors, we made the syntaxes case insensitive (except the syntax "WHERE" used in select query). They are as follows:

- Create Context: (Create Context<context name>;)
- Add Table: (Add Table;)
- Select: (Select<column\_name1>,<column\_name2>,<column\_n ame3>;)
- Save Context: (Save Context as<context name>;)
- Recall Context: (Recall Context<context\_name>;)
- Delete Context: (Delete context<context name>;)

# B. State Diagram of CBCQL System

The state diagram of CBCQL system is given below. When we create a context, it will go to state p(0,0). Initially context is empty. When we add a table, it will go to state q(r1,c1), where r1 is the number of rows in the table and c1 is the number of columns at state q. When we use Select statement or Recall statement, it will go to state r(r2,c2). This is because, in both the cases, the context is modified. If we delete our current context, the context will go to state p(0,0) which has no records. Again, we have to add a table if we want to query further in a context. The state diagram of CBCQL system is shown below.

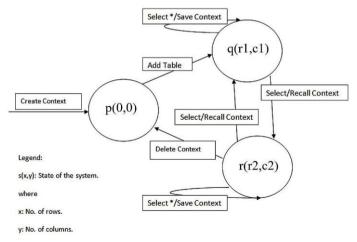


Fig. 1. State diagram of CBCQL system.

# C. An Example of Querying in CBCQL

Let's see a sequence of queries, on a database for buying a home online in a context based environment.

- 1) Select placeid, price, baths, beds, city, WHERE type='Residential';
- 2) Select placeid, price, baths, beds WHERE city='SACRAMENTO':
  - 3) Select placeid, price, baths WHERE beds='3';
  - 4) Select placeid, price WHERE baths>2;
  - 5) Select placeid WHERE piece<10000;

## D. Mapping CBCQL to CQL

When we execute a CBCQL query, it map to CQL query internally and then run on Cassandra database. The result of query map back to the GUI of CBCQL. The mapping from CBCQL to CQL is shown in the Table 1 below:

TABLE I. MAPPING OF CBCQL TO CQL

	T	
S.No	CBCQL Query	CQL Query
1.	Create Context <ccontext_name>;</ccontext_name>	No mapping.
2.	Add Table <table_name>;</table_name>	Select*from< Keyspace_name.ccontext_name> ;
3.	Select <column_name1>,&lt; column_name2&gt;,<column _name3&gt;&lt;;</column </column_name1>	Select <column_name1>,<colum n_name2&gt;from <keyspace_name.ccontext_name &gt;;</keyspace_name.ccontext_name </colum </column_name1>
4.	Select <column_name1>,&lt; column_name2&gt;WHE RE <condition>;</condition></column_name1>	Select <column_name1>,<colum n_name2&gt;from <keyspace_name.ccontext_name &gt; WHERE <condition>;</condition></keyspace_name.ccontext_name </colum </column_name1>
5.	Select <column_name1>,&lt; column_name2&gt;WHE RE <condition1> and <condition2>;</condition2></condition1></column_name1>	Select <column_name1>,<column_name2>from <reyspace_name.ccontext_name>WHERE<condition1>and <condition2>;</condition2></condition1></reyspace_name.ccontext_name></column_name2></column_name1>
6.	Save context as <scontext_name>;</scontext_name>	No direct mapping <sup>1</sup>
7.	Recall Context <scontext_name>;</scontext_name>	Select* from <keyspsce_name.scontext_n ame&gt;;</keyspsce_name.scontext_n 
8.	Delete context <dcontext_name>;</dcontext_name>	Drop table <keyspace_name.dcontext_n ame&gt;;</keyspace_name.dcontext_n 

## E. Architecture of CBCQL System

The architecture of CBCQL system is shown in Fig. 2. The front end provides CBCQL GUI for interaction. A user's

query is expressed using the GUI. This query is passed to the CBCQL system. Within the system, it is actually received by the CBCOL query engine. CBCOL query engine stores some information on metadata store and accesses information from metadata store and passes the CBCQL query with this information to guery mapper. Query mapper maps the CBCQL query to COL query and passes to Cassandra database. There may be one, more than one, or no COL query for a single CBCOL query. The mapped COL query runs on Cassandra database. The result of the query is passed to the CBCQL engine through query mapper. The CBCQL query engine stores some information from result to metadata store and sends the result to the GUI. Now the user will see the result from GUI and query on it. The system is designed in such a way that the query of the user will run only in its Context. All these processes are hidden from the user. The user will only query through the GUI and will see his result in the table of the GUI.

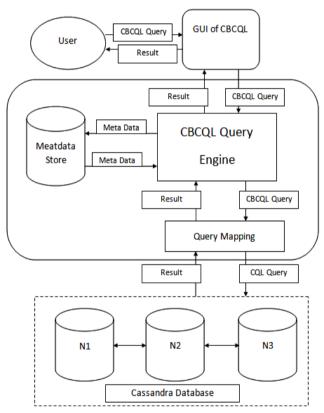


Fig.2. Architecture of CBCQL

# F. Data Set For Experiment

We have taken the dataset from [15] and did some modifications as per our need. In our dataset, there are fifteen attributes, and one thousand five hundred three rows, for describing the homes for sale. These attributes are price, baths, beds, city, area, other\_services, parking\_lot, placeID, Rpayment, sq\_ft, state, street, type, url, and zip. A user can find a desired home by putting conditions on these attributes.

<sup>&</sup>lt;sup>1</sup> Two queries are invoked for Save Context query. The first is Create table and second is Insert into table.

# G. Query Execution And Results

We created a GUI by using Java Swing. In the first part of the GUI, there is a text area, for writing query followed by a dynamic table that is created for showing the results of the queries. Next, in the third part we have shown the query execution time and the messages. The number of rows, we get from the execution of a query is printed in the textArea field of our GUI. In the last part, there are three buttons. Execute button is for executing the query. The Clear button clears the text area that is used for the query. The Exit button closes the GUI. Queries are as follows:

1) Create Context abc;

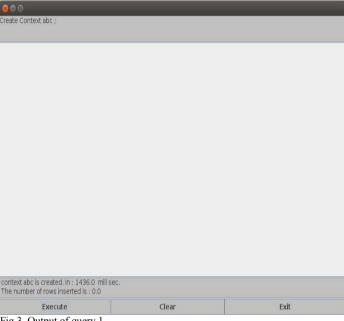


Fig.3. Output of query 1.

2) Add Table Data3:

2	3	SACRA	1				rpaym	sq_ft	state	street	type		zip
0			closed	none	fee	132774	cash	1430	SLP	2328	Resid	?	95838
	4	SACRA	closed	none	yes	135048	VISA	3615	Morelos	4186	Resid	?	95742
2	2	SACRA	closed	none	yes	133008	cash	918	San L	1016	Resid	?	95838
1	3	SACRA	closed	none	fee	132982	cash	966	tamau	523 E	Resid	?	95838
2	4	SACRA	closed	none	fee	132983	cash	1448	San L	3503	Resid	?	95820
1	3	SACRA	closed	none	public	133007	cash	1082	SLP	540 M	Resid	?	95838
3	4	SACRA	closed	none	yes	135063	cash	2056	tamau	9507	Resid	?	95758
3	3	SACRA	closed	none	yes	135040	Maste	1337	mexico	671 S	Resid	?	95815
2	3	ROSEV	closed	none	yes	135058	cash	2800	S.L.P.	4727	Resid	?	95835
3	5	SACRA	closed	none	ves	135058	cash	2800	S.L.P.	4727	Resid	?	95835
3	118	SACRA	closed	none	yes	135058	cash	2800	S.L.P.	4727	Resid	?	95835
2	3	SACRA	closed	none	yes	135068	VISA	1326	Tama	4010	Condo	?	95608
2	4	SACRA	closed	none	public	133033	Americ	1624	tamau	3935	Resid	?	95650
2	0	LINCOLN	closed	none	yes	134991	Maste	2030	SLP	5342	Resid	?	95835
1	2	SACRA	closed	none	yes	135090	cash	1120	SLP	2622	Resid	?	95833
1	3	SACRA	closed	none	yes	135090	cash	1120	SLP	2622	Resid	?	95833
1	4	SACRA	closed	none	yes	135090	cash	1120	SLP	2622	Resid	?	95833
1	46	SACRA	closed	none	yes	135090	cash	1120	SLP	2622	Resid	?	95833
2	0	SACRA	closed	none	ves	135099	Maste	1248	slp	7825	Resid	?	95828
2	5	CARMI	closed	Internet	yes	135098	cash	1039	SLP	5201 L.	Condo	?	95758
2	2	CARMI	closed	none	yes	135032	Maste	1211	SLP	648 S	Resid	?	95838
2	3	CARMI	closed	none	yes	135032	Maste	1211	SLP	648 S	Resid	?	95838
2	3	SACRA	closed	none	none	135032	Maste	1211	SLP	648 S	Resid	?	95838
2	187	SACRA	closed	none	none	135032	Maste	1211	SLP	648 S	Resid	?	95838
	2	SACRA	closed	none	yes	135090	cash	1120	SLP	2622	Resid	?	95833
1	-					135105		1022	SLP	4533 L.			95842
	2 1 3 3 3 2 2 3 3 3 2 2 2 2 1 1 1 1 1 1	2 4 4 3 3 3 4 4 3 3 5 5 3 118 2 2 3 3 2 2 4 4 0 1 1 2 1 1 4 4 1 4 4 6 2 0 0 2 2 5 2 2 2 2 2 3 3 2 2 3 3 2 2 3 3	2 4 SACRA 3 SACRA 4 SACRA 1 SACRA 1 SACRA 1 SACRA 1 SACRA 2 CARMI	2 4 SACRA closed 3 SACRA closed 3 4 SACRA closed 3 3 SACRA closed 3 3 SACRA closed 3 SACRA closed 3 SACRA closed 4 SACRA closed 5 SACRA closed 6 SACRA closed 1 SACRA closed 2 CARMII closed 2 SACRA closed 3 SACRA closed 4 SACRA closed 5 SACRA closed 6 SACRA closed 7 SACRA closed 8 SACRA closed 9 SACRA closed 1 SACRA closed 1 SACRA closed 1 SACRA closed 1 SACRA closed 2 SACRA closed 2 SACRA closed 3 SACRA closed	2 4 SACRA., closed none 3 4 SACRA., closed none 3 4 SACRA., closed none 3 3 SACRA., closed none 3 3 SACRA., closed none 3 SACRA., closed none 3 SACRA., closed none 3 SACRA., closed none 4 SACRA., closed none 5 SACRA., closed none 6 SACRA., closed none 1 SACRA., closed none 2 SACRA., closed none 1 SACRA., closed none 1 SACRA., closed none 2 SACRA., closed none 1 SACRA., closed none 1 SACRA., closed none 1 SACRA., closed none 2 SACRA., closed none 3 SACRA., closed none 1 SACRA., closed none 2 SACRA., closed none 3 SACRA., closed none	3	2 4 SACRA closed none fee 1,32993 3 SALRA closed none yes 1,35063 3 4 SACRA closed none yes 1,35063 3 3 SACRA closed none yes 1,35063 3 5 SACRA closed none yes 1,35063 3 118 SACRA closed none yes 1,35058 3 118 SACRA closed none yes 1,35058 2 2 3 SACRA closed none yes 1,35058 2 1 SACRA closed none yes 1,35068 2 1 SACRA closed none yes 1,35068 1 SACRA closed none yes 1,35061 1 2 SACRA closed none yes 1,35091 1 2 SACRA closed none yes 1,35091 1 2 SACRA closed none yes 1,35091 1 4 SACRA closed none yes 1,35090 1 4 SACRA closed none yes 1,35090 2 0 SACRA closed none yes 1,35090 3 SACRA closed none yes 1,35092 2 3 CARMII closed none yes 1,35092 2 3 SACRA closed none yes 1,35092 2 3 SACRA closed none yes 1,35032	2         4         SACFA closed none fee         13,983 cash           3         3         SACFA closed none public 133007 cash           3         4         SACFA closed none yes 155063 cash           3         3         SACFA closed none yes 155060 Maste           2         3         SACFA closed none yes 135058 cash           3         118         SACFA closed none yes 155058 cash           2         3         SACFA closed none yes 135068 MSA           2         4         SACFA closed none yes 135098 MSA           1         2         SACFA closed none yes 135090 cash           1         2         SACFA closed none yes 135090 cash           1         3         SACFA closed none yes 135090 cash           1         4         SACFA closed none yes 135090 cash           2         0         SACFA closed none yes 135090 xash           2         0         SACFA closed none yes 135090 xash           2         0         SACFA closed none yes 135093 xash           2	3	2         4         SACRA., closed once         fee         120283 cash         1448         San L           3         3         SACRA., closed once         public         130007 cash         1082 SLP           3         3         SACRA., closed once         yes         135063 cash         2056 tamau           3         3         SACRA., closed once         yes         135068 cash         2800 SLP.           3         5         SACRA., closed once         yes         135058 cash         2800 SLP.           3         118         SACRA., closed once         yes         135068 wish         2800 SLP.           2         4         SACRA., closed once         yes         135068 wish         2360 SLP.           2         4         SACRA., closed once         yes         135090 wish         1262 tamau           1         2         SACRA., closed once         yes         135090 wish         120 SLP           1         4         SACRA., closed once         yes         135090 wish         120 SLP           1         4         SACRA., closed once         yes         135090 wish         120 SLP           1         4         SACRA., closed once         yes         135090 wish	2         4         SACRAclosed none fee         132993 (ash 1448 san L. 3503           3         3         SACRAclosed none public 133007 (ash 1208 SLP 340 M           3         4         SACRAclosed none yes 135003 (ash 2056 tamau9507           3         3         SACRAclosed none yes 135008 (ash 2800 SL.P. 4727           3         5         SACRAclosed none yes 135058 (ash 2800 SL.P. 4727           3         118         SACRAclosed none yes 135058 (ash 2800 SL.P. 4727           2         3         SACRAclosed none yes 135058 (ash 2800 SL.P. 4727           2         4         SACRAclosed none yes 135068 (ash 2800 SL.P. 4727           1         2         3         SACRAclosed none yes 135068 (ash 2800 SL.P. 4727           1         2         3         SACRAclosed none yes 135098 (ash 1200 SL.P. 4727           1         2         SACRAclosed none yes 135090 (ash 1120 SLP 2622           1         2         SACRAclosed none yes 135090 (ash 1120 SLP 2622           1         4         SACRAclosed none yes 135090 (ash 1120 SLP 2622           1         4         SACRAclosed none yes 135090 (ash 1120 SLP 2622           2         5         CARMIclosed none yes 135090 (ash 120 Sh 120 SLP 2622           2         <	3	2         4         SACRAclosed none fee         139983 cash         1448         San L         3503         Resid         ?           3         3         SACRA closed none public         133007         cash         1082         SLP         \$40 M         Resid         ?           3         4         SACRA closed none yes         135063 cash         2056         tamau         5077         Resid         ?           2         3         SACRA closed none yes         135058 cash         2000         SL.P.         4727         Resid         ?           3         118         SACRA closed none yes         135058 cash         2000         SL.P.         4727         Resid         ?           2         4         SACRA closed none yes         135058 cash         1200         SL.P.         4727         Resid         ?           1         2         SACRA closed none yes         135098 cash         1226         tamau         3935         Resid         ?           1         2         SACRA closed none yes         135098 cash         1226         tamau         3935         Resid         ?           1         2         SACRA closed none yes </td

Fig. 4. Output of query 2.

This query has added the table in the context. Table Data3 is our data set. In our dataset, we have taken fifteen attributes to describe a home. The client will put conditions on these attributes to get the most suitable deal for him.

3) Select price, baths, beds, city, area, parking lot, placeid . rpayment . sa ft . type :

price	baths	beds	city	area	parking lot	placeid	rpayment	sq ft	type
05878	2	3	SACRAMEN	closed	fee	132774	cash	1430	Residential
68500	0	4	SACRAMEN	closed	yes	135048	VISA	3615	Residential
1000	2	2	SACRAMEN	closed	yes	133008	cash	918	Residential
14000	1	3	SACRAMEN	closed	fee	132982	cash	966	Residential
14000	2	4	SACRAMEN	closed	fee	132983	cash	1448	Residential
5000	1	3	SACRAMEN	closed	public	133007	cash	1082	Residential
77980	3	4	SACRAMEN	closed	ves	135063	cash	2056	Residential
52000	3	3	SACRAMEN	closed	ves	135040	MasterCar	1337	Residential
04037	2	3	ROSEVILLE	closed	ves	135058	cash	2800	Residential
04037	3	5	SACRAMEN	closed	ves	135058	cash	2800	Residential
04037	3	118	SACRAMEN	closed	ves	135058	cash	2800	Residential
51000	2	3	SACRAMEN	closed	ves	135068	VISA	1326	Condo
39000	2	4	SACRAMEN	closed	public	133033	American	1624	Residential
37760	2	0	LINCOLN	closed	ves	134991	MasterCar	2030	Residential
64000	1	2	SACRAMEN	closed	yes	135090	cash	1120	Residential
64000	1	3	SACRAMEN	closed	yes	135090	cash	1120	Residential
64000	1	4	SACRAMEN	closed	yes	135090	cash	1120	Residential
64000	1	46	SACRAMEN	closed	ves	135090	cash	1120	Residential
64000	2	0	SACRAMEN	closed	ves	135099	MasterCar	1248	Residential
64000	2	5	CARMICHAEL	closed	yes	135098	cash	1039	Condo
35000	2	2	CARMICHAEL	closed	yes	135032	MasterCar	1211	Residential
35000	2	3	CARMICHAEL	closed	yes	135032	MasterCar	1211	Residential
35000	2	3	SACRAMEN	closed	none	135032	MasterCar	1211	Residential
35000	2	187	SACRAMEN	closed	none	135032	MasterCar	1211	Residential
88325	1	2	SACRAMEN	closed	yes	135090	cash	1120	Residential
08750	2	2	SACRAMEN	closed	yes	135105	cash	1022	Residential
	ECUTED SUG per of rows in Execut	serted is : 1	In: 7540.0 m 503.0		lear			Exit	

Fig.5. Output of query 3.

In this query, the client selects some relevant attributes to him and leaves the remaining.

4) Select \* WHERE type='Residential';

3	SACRAMEN	closed	fee	132774	A CONTRACTOR OF THE PARTY OF TH	-	
4	010011151			132//4	cash	1430	Residentia
	SACRAMEN	closed	ves	135048	VISA	3615	Residentia
2	SACRAMEN	closed	ves	133008	cash	918	Residentia
3	SACRAMEN	closed	fee	132982	cash	966	Residentia
4	SACRAMEN	closed	fee	132983	cash	1448	Residentia
3	SACRAMEN	closed	public	133007	cash	1082	Residentia
4	SACRAMEN	closed	ves	135063	cash	2056	Residentia
3	SACRAMEN	closed	ves	135040	MasterCar	1337	Residentia
3	ROSEVILLE	closed	yes	135058	cash	2800	Residentia
5	SACRAMEN	closed	ves	135058	cash	2800	Residentia
118	SACRAMEN	closed	ves	135058	cash	2800	Residentia
4	SACRAMEN	closed	public	133033	American	1624	Residentia
0	LINCOLN	closed	ves	134991	MasterCar	2030	Residentia
2	SACRAMEN	closed	ves	135090	cash	1120	Residentia
3	SACRAMEN	closed	yes	135090	cash	1120	Residentia
4	SACRAMEN	closed	yes	135090	cash	1120	Residentia
46	SACRAMEN	closed	yes	135090	cash	1120	Residentia
0	SACRAMEN	closed	yes	135099	MasterCar	1248	Residentia
2	CARMICHAEL	closed	yes	135032	MasterCar	1211	Residentia
3	CARMICHAEL	closed	yes	135032	MasterCar	1211	Residentia
3	SACRAMEN	closed	none	135032	MasterCar	1211	Residentia
187	SACRAMEN	closed	none	135032	MasterCar	1211	Residentia
2	SACRAMEN	closed	yes	135090	cash	1120	Residentia
2	SACRAMEN	closed	yes	135105	cash	1022	Residentia
	010011451	alacad.	ves	135105	cash	1022	Residentia
15	SACRAMEN	ciosea	yes	122102	Casii	1022	Iveginelitia
	3 4 4 3 3 5 5 118 4 4 0 0 2 2 3 3 4 4 4 6 0 0 2 2 3 3 3 3 3 3 187	3 SACRAMEN 4 SACRAMEN 3 SACRAMEN 5 SACRAMEN 118 SACRAMEN 0 UNICOLV 2 SACRAMEN 4 SACRAMEN 0 SACRAMEN 2 SACRAMEN 4 SACRAMEN 2 CARMEN 14 SACRAMEN 2 CARMEN 3 SACRAMEN 4 SACRAMEN 5 SACRAMEN 6 SACRAMEN 1 SACRAMEN 2 CARMICHAEL 3 SACRAMEN 2 CARMICHAEL 3 SACRAMEN 4 SACRAMEN 5 SACRAMEN 5 SACRAMEN	3 SACRAMEN closed 4 SACRAMEN closed 3 SACRAMEN closed 5 SACRAMEN closed 5 SACRAMEN closed 118 SACRAMEN closed 4 SACRAMEN closed 0 UINCOLV closed 2 SACRAMEN closed 3 SACRAMEN closed 4 SACRAMEN closed 5 SACRAMEN closed 6 SACRAMEN closed 7 SACRAMEN closed 8 SACRAMEN closed 9 SACRAMEN closed 10 SACRAMEN closed 1187 SACRAMEN closed 1187 SACRAMEN closed	3 SACRAMEN closed public 4 SACRAMEN closed yes 3 SACRAMEN closed yes 5 SACRAMEN closed yes 5 SACRAMEN closed yes 5 SACRAMEN closed yes 6 SACRAMEN closed yes 4 SACRAMEN closed public 0 UNCOIN closed public 2 SACRAMEN closed yes 3 SACRAMEN closed yes 4 SACRAMEN closed yes 6 SACRAMEN closed yes 7 SACRAMEN closed yes 8 CARMICHAEL closed yes 9 CARMICHAEL closed yes 1 CARMICHAEL closed yes 1 SACRAMEN closed none	3 SACRAMEN closed public 133007 4 SACRAMEN closed yes 135063 3 SACRAMEN closed yes 135040 3 ROSEVILLE closed yes 135058 5 SACRAMEN closed yes 135058 118 SACRAMEN closed yes 135058 4 SACRAMEN closed yes 135058 2 SACRAMEN closed yes 135090 3 LINCOLN closed yes 134991 2 SACRAMEN closed yes 135090 3 SACRAMEN closed yes 135090 4 SACRAMEN closed yes 135090 4 SACRAMEN closed yes 135090 4 SACRAMEN closed yes 135090 2 CARMICHAEL closed yes 135090 3 SACRAMEN closed yes 135090 4 SACRAMEN closed yes 135090 3 SACRAMEN closed yes 135090 3 SACRAMEN closed yes 135032 3 SACRAMEN closed yes 135032 3 SACRAMEN closed yes 135032 3 SACRAMEN closed none 135032 3 SACRAMEN closed none 135032	3   SACRAMEN   closed   public   133007   cash	3 SACRAMEN (closed public 133007 cash 1082 4 SACRAMEN (closed yes 135063 cash 2006 5 SACRAMEN (closed yes 135064 MasterCar 1337 6 SACRAMEN (closed yes 135058 cash 2800 5 SACRAMEN (closed yes 135058 cash 2800 118 SACRAMEN (closed yes 135058 cash 2800 4 SACRAMEN (closed yes 135058 cash 2800 6 UNIOCUN (closed public 133033 American 1624 0 UNIOCUN (closed yes 135090 cash 120 2 SACRAMEN (closed yes 135090 cash 1120 3 SACRAMEN (closed yes 135090 cash 1120 4 SACRAMEN (closed yes 135090 cash 1120 4 SACRAMEN (closed yes 135090 cash 1120 5 SACRAMEN (closed yes 135090 cash 1120 6 SACRAMEN (closed yes 135090 cash 1120 7 SACRAMEN (closed yes 135090 cash 1120 8 SACRAMEN (closed yes 135092 MasterCar 1211 8 SACRAMEN (closed none 135032 MasterCar 1211 187 SACRAMEN (closed none 135032 MasterCar 1211

Fig. 6. Output of query 4.

This guery selects all records for home of type residential. The result of this query has same fields as of query 3 as shown in Fig. 6.

5) Select price, baths, beds, city, area, parking\_lot, placeid, rpayment, sq ft WHERE beds = 3 and baths > 1;

price	baths	beds	city	area	parking lot	placeid	rpayment	sq ft	
05878	2	3	SACRAMENTO	closed	fee	132774		1430	
52000	3	3	SACRAMENTO	closed	ves	135040	MasterCard	1337	
04037	2	3	ROSEVILLE	closed	ves	135058	cash	2800	
35000	2	3	CARMICHAEL	closed	ves	135032	MasterCard	1211	
35000	2	3	SACRAMENTO	closed	none	135032	MasterCard	1211	
08750	3	3	SACRAMENTO	closed	ves	135105	cash	1022	
79000	2	3	SACRAMENTO	closed	public	135021	cash	1265	
36250	2	3	GOLD RIVER	closed	ves	135001	VISA	1291	
38750	2	3	ELVERTA	closed	public	135098	bank debit	1116	
38750	3	3	FAIR OAKS	closed	public	135098	bank debit		
40000	2	3	ELK GROVE	open	ves	135031	MasterCard		
40000	2	3	GALT	closed	none	135031	MasterCard	1266	
40000	2	3	SACRAMENTO	closed	yes	135031	VISA	1080	
74313	2	3	SACRAMENTO	closed	ves	135086	MasterCard	1714	
47308	2	3	NORTH HIGH	closed	yes	135097	MasterCard	1082	
47308	2	3	SACRAMENTO	closed	ves	135097	MasterCard	1082	
00000	4	3	SACRAMENTO	closed	yes	135046	MasterCard	1511	
06716	2	3	RANCHO COR	. closed	yes	135035	MasterCard	1080	
60014	2	3	ROSEVILLE	closed	yes	135066	cash	1018	
23000	3	3	CARMICHAEL	closed	yes	135102	VISA	1240	
23225	3	3	SACRAMENTO	open	yes	135034	American Ex	1050	
36500	2	3	ROSEVILLE	closed	yes	135098	MasterCard	1380	
36500	2	3	SACRAMENTO	closed	yes	135098	MasterCard	1380	
34555	2	3	NORTH HIGH	closed	yes	135098	VISA	1152	
34555	3	3	ROSEVILLE	closed	yes	135098	VISA	1152	
35750	2	3	SACRAMENTO	closed	yes	135055	cash	2354	

Fig.7. Output of query 5.

This query selects the record for a home with three bedrooms and at least two bathrooms.

6) Save context as pqr;

price	baths	beds	city	area	parking lot	placeid	rpayment	sq_ft	
205878	2	3	SACRAMENTO	closed	fee	132774	cash	1430	
52000	3	3	SACRAMENTO	closed	ves	135040	MasterCard	1337	
804037	2	3	ROSEVILLE	closed	ves	135058	cash	2800	
35000	2	3	CARMICHAEL	closed	ves	135032	MasterCard	1211	
35000	2	3	SACRAMENTO	closed	none	135032	MasterCard	1211	
08750	3	3	SACRAMENTO	closed	ves	135105	cash	1022	
79000	2	3	SACRAMENTO	closed	public	135021	cash	1265	
36250	2	3	GOLD RIVER	closed	yes	135001	VISA	1291	
38750	2	3	ELVERTA	closed	public	135098	bank debit	1116	
38750	3	3	FAIR OAKS	closed	public	135098	bank debit	1116	
40000	2	3	ELK GROVE	open	yes	135031	MasterCard		
40000	2	3	GALT	closed	none	135031	MasterCard	1266	
40000	2	3	SACRAMENTO	closed	yes	135031	VISA	1080	
74313	2	3	SACRAMENTO	closed	ves	135086	MasterCard	1714	
47308	2	3	NORTH HIGH	closed	yes	135097	MasterCard	1082	
47308	2	3	SACRAMENTO	closed	yes	135097	MasterCard	1082	
00000	4	3	SACRAMENTO	closed	yes	135046	MasterCard	1511	
06716	2	3	RANCHO COR	closed	yes	135035	MasterCard	1080	
260014	2	3	ROSEVILLE	closed	yes	135066	cash	1018	
23000	3	3	CARMICHAEL	closed	yes	135102	VISA	1240	
23225	3	3	SACRAMENTO	open	yes	135034	American_Ex	1050	
36500	2	3	ROSEVILLE	closed	yes	135098	MasterCard	1380	
36500	2	3	SACRAMENTO	closed	yes	135098	MasterCard	1380	
34555	2	3	NORTH HIGH	closed	yes	135098	VISA	1152	
34555	3	3	ROSEVILLE	closed	yes	135098	VISA	1152	
35750	2	3	SACRAMENTO	closed	yes	135055	cash	2354	

Fig. 8. Output of query 6.

Context is being saved in this query. The essential requirement of the client was a home for residential purpose with three bedrooms and at least two bathrooms. The result of this query is fulfilling all these conditions. Since, in context based querying, we cannot backtrack so it's better to save context and when we need, recall the context.

7) Select city, price, area, parking\_lot, placeid, sq\_ft WHERE parking lot='ves' and city='SACRAMENTO';

city	price	area	parking lot	placeid	sq_ft
ACRAMENTO	452000	closed	yes	135040	1337
ACRAMENTO	108750	closed	ves	135105	1022
ACRAMENTO	140000	closed	yes	135031	1080
ACRAMENTO	174313	closed	yes	135086	1714
ACRAMENTO	147308	closed	yes	135097	1082
ACRAMENTO	600000	closed	yes	135046	1511
ACRAMENTO	123225	open	yes	135034	1050
ACRAMENTO	136500	closed	yes	135098	1380
ACRAMENTO	335750	closed	yes	135055	2354
ACRAMENTO	242638	closed	yes	135071	2163
ACRAMENTO	241000	closed	yes	135071	1269
ACRAMENTO	198000	closed	yes	135078	1266
ACRAMENTO	154000	open	yes	135092	1207
ACRAMENTO	280908	closed	yes	135064	1284
ACRAMENTO	122000	closed	yes	135103	1118
ACRAMENTO	174250	closed	yes	135086	1463
ACRAMENTO	460000	closed	yes	135048	2687
ACRAMENTO	65000	closed	yes	133029	796
ACRAMENTO	65000	closed	yes	133030	932
ACRAMENTO	62050	closed	ves	133030	623
ACRAMENTO	90000	closed	yes	135040	1337
ACRAMENTO	427500	closed	yes	135042	800
ACRAMENTO	236073	closed	yes	135073	1277
ACRAMENTO	582000	closed	yes	133035	2222
ACRAMENTO	183200	closed	yes	135080	1603
ACRAMENTO	260000	open	yes	135067	1541
The number of r	ED SUCCESSFULLY In rows inserted is : 107 execute	7.0	Clear		Exit

Fig. 9. Output of query 7.

The result of this query retrieves the detail of home with three bedrooms and at least two bathrooms for a resident in the Sacramento city.

8) Select area , price , placeid , sq\_ft WHERE area='open' and price < 10000;

	-4	placeid	4	
area	price 4897	135008	sq_ft 1953	
open open	4897	135013	1393	
open	4897	135027	1104	_
open	4897	135028	1320	

Fig. 10. Output of query 8.

This query displays the detail of all homes, fulfilling all the above conditions and in less than 10000, with an open area.

9) Recall Context pqr;

price	baths	beds	city	area	parking lot	placeid	rpayment	sq_ft	
205878	2	3	SACRAMENTO	closed	fee	132774	cash	1430	
52000	3	3	SACRAMENTO	closed	ves	135040	MasterCard	1337	
04037	2	3	ROSEVILLE	closed	ves	135058	cash	2800	
35000	2	3	CARMICHAEL	closed	yes	135032	MasterCard	1211	
35000	2	3	SACRAMENTO	closed	none	135032	MasterCard	1211	
108750	3	3	SACRAMENTO	closed	ves	135105	cash	1022	
79000	2	3	SACRAMENTO	closed	public	135021	cash	1265	
236250	2	3	GOLD RIVER	closed	yes	135001	VISA	1291	
138750	2	3	ELVERTA	closed	public	135098	bank debit	1116	
138750	3	3	FAIR OAKS	closed	public	135098	bank debit	1116	
40000	2	3	ELK GROVE	open	yes	135031	MasterCard	1266	
140000	2	3	GALT	closed	none	135031	MasterCard	1266	
40000	2	3	SACRAMENTO	closed	yes	135031	VISA	1080	
174313	2	3	SACRAMENTO	closed	yes	135086	MasterCard	1714	
147308	2	3	NORTH HIGH	closed	yes	135097	MasterCard	1082	
147308	2	3	SACRAMENTO	closed	yes	135097	MasterCard	1082	
000000	4	3	SACRAMENTO	closed	yes	135046	MasterCard	1511	
106716	2	3	RANCHO COR	. closed	yes	135035	MasterCard	1080	
260014	2	3	ROSEVILLE	closed	yes	135066	cash	1018	
123000	3	3	CARMICHAEL	closed	yes	135102	VISA	1240	
23225	3	3	SACRAMENTO	open	yes	135034	American_Ex	1050	
136500	2	3	ROSEVILLE	closed	yes	135098	MasterCard	1380	
136500	2	3	SACRAMENTO	closed	yes	135098	MasterCard	1380	
134555	2	3	NORTH HIGH	closed	yes	135098	VISA	1152	
134555	3	3	ROSEVILLE	closed	yes	135098	VISA	1152	
335750	2	3	SACRAMENTO	closed	yes	135055	cash	2354	

Fig. 11. Output of query 9.

If the client does not find a better deal, he can recall the context and search with some other conditions as in another city or different price. Now the subsequent query will run on the recalled context.

10) Delete Context pqr;

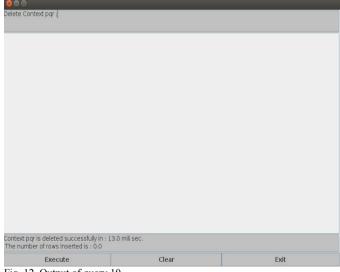


Fig. 12. Output of query 10.

This query deleted the context pqr. If pqr was our current context, then we cannot query further in the context before adding a table in the context, or recalling a context.

Here an example was demonstrated to show the working of our CBCQL system. We have shown the queries and screenshot of the results. The sequence of queries reflect the thought process of the user. In some queries there was only one condition to narrow down the result and in some other cases more than one. For example, we combined multiple

conditions at some places because they are logically related and are considered together in our daily life, as number of bedrooms and bathrooms.

#### IV. CONCLUSION & FUTURE WORK

In this paper, we have proposed a new query language named as Context Based Cassandra Query Language. The purpose of this language is to provide a mechanism by which a user can ask a sequence of related queries. As a result, an easy way of querying with simpler queries and dictated by the thought process was provided. The user has to specify only SELECT and WHERE clause in the context. The context is designed in such a way that it fetches result from the context and updates the context with the result. Once a condition was expressed in a query within a context, there was no need to repeat the condition in the subsequent queries. We provided the facility of saving a context and recalling it, so backtracking is also easy for the user while querying. CBCQL has the same power as Cassandra with additional functionality because it is built over and above Cassandra. For using CBCQL we have provided a GUI which is very easy to use and simple to understand. CBCOL has a very simple and case insensitive syntax, so the possibility of errors is reduced. Cassandra is a new database and there is a major difference in terms of power and functionality in every new version of Cassandra. Even with low support of Java for Cassandra database the system was fully implemented with the desired results. In this paper. we implemented CBCQL for native data types of CQL. In future CBCQL can be implemented for collection data types and string data types (custom data types) of CQL.

### ACKNOWLEDGMENT

The authors would like to thank CSIR for providing financial assistance. The authors would like express their sincere gratitude to all the staffs of SC & SS, JNU for their support behind this work.

# REFERENCES

- [1] S.K.Gajendran, "A survey on nosql databases", Technical report, University of Illinois, 2012.
- [2] D. Zhang, "Inconsistencies in big data", In Cognitive Informatics & Cognitive Computing (ICCI\* CC) 12<sup>th</sup> IEEE International Conference on IEEE, pp. 61-67,2013.
- [3] P.S. Duggal & S.Paul, "Big Data Analysis: Challenges and Solutions", In International Conference on Cloud, Big Data and Trust, pp. 13-15,2013.
- [4] A.B.M. Moniruzzaman & S.A. Hossain, "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison", arXiv preprint arXiv:1307.0191,2013.
- [5] R.Cattell, "Scalable SQL and NoSQL data stores", ACM Sigmod Record, 39(4), pp.12-27,2011.
- [6] R. Agrawal, A. Ailamaki, P. A. Bernstein, E.A. Brewer, M.J. Carey, S. Chaudhuri, & G. Weikum, "The Claremont report on database research". ACM Sigmod Record, 37(3), pp. 9-19,2008.
- [7] C. Nance, T.Losser, R. Iype, & G.Harmon, "Nosql vs rdbms-why there is room for both", In Proceedings of the Southern Association for Information Systems Conference, pp.111-116,2013.
- [8] A.K. Zaki, "NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and Its Security Challenges", International Journal of Research in Engineering and Technology (IJRET), 3(15), pp. 403-409,2014.

- [9] G. Wang, & J. Tang (2012), "The nosql principles and basic application of cassandra model", In proceedings of Computer Science & Service System (CSSS) on IEEE, pp.1332-1335,2012.
- [10] C. Strauch, U.L.S. Sites & W.Kriha, "NoSQL databases". Lecture Notes, Stuttgart Media University, 2011.
- [11] B.G. Tudorica, & C.Bucur, "A comparison between several NoSQL databases with comments and notes", In Roedunet International Conference (RoEduNet), 10th IEEE, pp.1-5,2011.
- [12] N. Parimala, N. Prakash, B.L.N. Rao, & N. Bolloju, "A Query Facility to a network DBMS", The Computer Journal, 32(1), 55-62, 1989
- [13] N. Parimala, "Explicit operation specification for component databases", The Computer Journal, 45(2), pp. 202-212, 2002.

- [14] N. Parimala & S. Bhawna, "Continuous multiple olap queries for data streams", International Journal of Cooperative Information Systems, 21(02), pp.141-164,2012.
- [15] Sacramento\_Homes\_for\_Sale, "Retrieved from Sacramento\_Homes\_for\_Sale" December, 2014, [Online].

  Available: http://samplecsvs.s3.amazonaws.com/Sacramento\_Homes\_for\_Sale.csv.