Hungarian Association of Agricultural Informatics
European Federation for Information Technology in Agriculture, Food and the Environment

Journal of Agricultural Informatics. 2015 Vol. 6, No. 3
journal.magisz.org

Data storage and management related to soil carbon cycle by a NoSQL engine on a SQL platform - Joker Tao

Bence Mátyás[1], György Mátyás[2], Judit Horváth[3], János Kátai[4]

I N F O
Received 16 June 2015
Accepted 17 Sept 2015
Available on-line 12 Oct 2015
Responsible Editor: M. Herdon

Keywords:
GHG, NoSQL, Big Data, Joker Tao. Soil carbon cycle

A B S T R A C T

Carbon dioxide (CO_2) is the most important greenhouse gas contributing 60 % toward global warming, respectively (IPCC, 2007). The correlation management in researches which focus on greenhouse gases (GHG) emission and soil properties require huge amount of data. These data are inducted from measuring devices directly or from systems which are used for experimental data storage and management. Generally, these data come from different database systems which use different data storage structure.
The main objective of the present study was to discuss a non-relational database model called Joker Tao (JT) which provides Big Data management for soil data and their GHG production potential related to carbon cycle in a relational database environment (Oracle APEX).

1. Introduction

There are huge number of studies available in the technical literature which focus on GHG emission under various soil types, cropping, irrigation and fertilizer management (Cabre et al., 1994; Glatzel et al., 2004; Kong et al., 2013; Singla and Inubushi, 2014; Singla et al., 2014; Matyas et al., 2014). Big Data management is needed for storing and managing these scientific data because the number of the data lines are continuously increasing in databases. Of the many different data-models, the relational model - also known as Relational Database Management System (RDBMS) has been dominating since the 80's, with implementations such as Oracle databases (I1), MySQL (I2) and Microsoft SQL Servers (I3). Organizations that collect large amounts of unstructured data are increasingly turning to non-relational databases, or NoSQL databases, as it is now often called (Leavitt, 2010). NoSQL databases focus on the analytical processing of large scale datasets, offering increased scalability over commodity hardware (Konstantinou et al., 2011). It can be both useful and time saving if such large amount of data (more than 1 million data lines) with their attributes and meta-data could be indexed and managed in one physical data table but in structured concept. Database projects consist of three essential tasks: data collection, data preparation, and data modelling (Pyle, 1999; Westerman, 2001). Integrated mechanisms of such models can ensure that the framework stores inputted data into one non virtual (physical) data table. JT ensures that every inputted data can be indexed so there is no need to use sequencing searches which slow down the queries. One of the most important benefits of the mentioned model is the user can expand the database with new attributes (new data type or unit) during and after the data entry processes (Mátyás et al. 2014).

[1] Bence Mátyás
University of Debrecen, Centre for Agricultural and Applied Economic Sciences, Debrecen, Hungary
matyasbence@agr.unideb.hu
[2] György Mátyás
Statik Kontroll Ltd., Miskolc, Hungary
matyas.gyorgy.istvan@gmail.com
[3] Judit Horváth
University of Debrecen, Centre for Agricultural and Applied Economic Sciences, Debrecen, Hungary
judithorvath89@gmail.com
[4] János Kátai
University of Debrecen, Centre for Agricultural and Applied Economic Sciences, Debrecen, Hungary
katai@agr.unideb.hu

2. Material and Methods

2.1. Data sites used in the study

We continue the 30 years old long-term experiment of Látókép in our experiments. The aforementioned long-term fertilization experiment was setup in 1983, and our samples were taken from spring 2014 to spring 2015. The examinations of soil respiration processes and factors that influence soil respiration are required in optimal management. In our research, we are interested in knowing how the growing levels of fertilization influence the microbial processes under non-irrigated and irrigated conditions in maize mono, bi, and triculture (Table 1).

Table 1. Labels of fertilizer doses in different cultures

	Dose	Name	Monoculture	Biculture	Triculture
Non irrigated	Control	-	1	11	21
	N_{60} P_{45} K_{45}	small	2	12	22
	N_{120} P_{90} K_{90}	Small-medium	3	13	23
	N_{180} P_{135} K_{135}	Medium-high	4	14	24
	N_{240} P_{180} K_{180}	High	5	15	25
Irrigated	Control	-	6	16	26
	N_{60} P_{45} K_{45}	small	7	17	27
	N_{120} P_{90} K_{90}	Small-medium	8	18	28
	N_{180} P_{135} K_{135}	Medium-high	9	19	29
	N_{240} P_{180} K_{180}	High	10	20	30

2.2. JT database

The JT data modelling provides universal data storage and data management, so different data are able to be stored and managed under one platform regardless of which environment or database-structure they come from. Contents of JT are: -data model, -database model, -JT logic, -GUI. JT ensures interoperability between noSQL and SQL environment. The users can use the usual virtual data-tables while integrated mechanisms of the system stores the inputted data into one physical data-table. Different from the MongoDB, in JT no applications to manage data which are stored in unstructured data-storage concept need to be created. The associative attributes of the system are manifested in management of data connections and correlations from different environments.

2.3. Data model

Contrary to the relational data models, data lines with same ID values identify one entity in JT. Also in the non-relational models, the inputted data are not stored in unstructured concept as they are in JT, eliminating the need to create several application for data management. Thanks to the code table below, each data (entitiy, attribute, data connection, formula) are able to be stored and managed under one physical data table. There is no sequential search because every data lines are indexed (Table 2).

Table 2. Physical data table structure of JT

ID	Attribute	Value
1	1	Name
1	3	4
2	1	Size
2	3	5
3	1	Data type
3	3	7
3	8	4;5;6;7
4	1	alpha
5	1	num
6	1	date
7	1	code
8	1	list

Generally, 3 fields are used in the one physical data structure for entity identification, but occasionally 4 fields can be useful. We stored the Index field (which is used for list storage) with ID, Attribute and Value using our code table.

Similarly to most NoSQL models, the speed of JT system lies in vertical data expanding, and sequential search is not used which slows queries. One of the most important benefit in JT is every data could be entity and attribute at the same time in the data entry processes, and later the actually situation determines how the system interpret the stored data thanks to storing the four dimension of knowledge.

2.4. Database model

JT memorizes the different data, attributes and contacts by uniform storage layout. Each attribute is equal (key indexed) in this model because we do not know in the data entry processes what the aim of the query will be. The system operates the data by the same operation (proceeding) which leaves the datatype out of consideration. Every single A and B search path is further engraved (saved). The path number increases. Path number = path number + 1. Practised knowledge if a path number is a large amount. The often used searches are prioritised (Mátyás et al., 2015).

Firstly, JT selects all data lines from the database and writes the number of these data lines. Following this all data lines are inserted to one data table. The sequence of the fields is set (in this case for 6 fields). Secondly, a file reader and a .txt file are created.
Finally, JT reads every data from the .txt file and puts them to the one data table.

Identifier of Group_ID, Uniq_ID, Field_ID, Array_Index, Seek_Value, Field_Value were created to separate attributes of different entities. The Data.txt file was created to complete the insertion process. The method reads input data from virtual data tables and inserts them into one physical data table. Identifier of Group_ID, Uniq_ID, Array_Index, and Field_Value were used for entity identification as below in Java code:

```
public static String getEntityName() throws Exception
    {
    Connection conn = broker.getConnection();
    PreparedStatementpstmt = conn.prepareStatement("select * from joker");
    ResultSetrs = pstmt.executeQuery();
    inti = 0;
    while (rs.next()) {
    i++;
    }
    System.out.println("number of records: " + i);
```

```
        broker.freeConnection(conn);
        return "";
        }
    public static void insertJokerRow(Integer GROUP_ID, Integer UNIQ_ID, Integer FIELD_ID, Integer ARRAY_INDEX, String SEEK_VALUE, String FIELD_VALUE) throws Exception {
    if (GROUP_ID == null) pstmt.setNull(1, 2); else pstmt.setInt(1, GROUP_ID.intValue());
    if (UNIQ_ID == null) pstmt.setNull(2, 2); else pstmt.setInt(2, UNIQ_ID.intValue());
    if (FIELD_ID == null) pstmt.setNull(3, 2); else pstmt.setInt(3, FIELD_ID.intValue());
    if (ARRAY_INDEX == null) pstmt.setNull(4, 2); else pstmt.setInt(4, ARRAY_INDEX.intValue());
    if (SEEK_VALUE == null) pstmt.setNull(5, 12); else pstmt.setString(5, SEEK_VALUE);
    if (FIELD_VALUE == null) pstmt.setNull(6, 12); else pstmt.setString(6, FIELD_VALUE);
    pstmt.execute();
     }
    public static void readFile() throws Exception
     {
       File f = new File("data.txt");
    BufferedReaderbr = new BufferedReader(new FileReader(f));
    while (br.ready()) {
        String line = br.readLine();
    int GROUP_ID = Integer.parseInt(line.substring(0, 10));
    int UNIQ_ID = Integer.parseInt(line.substring(11, 21));
    int ARRAY_INDEX = Integer.parseInt(line.substring(22, 32));
        String FIELD_VALUE = line.length() > 32 ? line.substring(33, line.length()) : "";
    insertJokerRow(Integer.valueOf(GROUP_ID), Integer.valueOf(UNIQ_ID), null, Integer.valueOf(ARRAY_INDEX), null, FIELD_VALUE);
       }
    br.close();
    }
```

The developed model operates under multiple conditions, real (data from physical data table) or virtual (data from virtual data tables) objects. The structure of data storing was determined vertically.

2.5. JT logic

Thanks to the programmed four dimension of knowledge in the code table, the stored data in the system can be interpreted as entity and attribute at the same time. The four dimension of knowledge prevail on the physical data storage layer. The four dimension of knowledge are: -genum proximum of entity, -differencia specifica of entity, -genum proxiumum of attribute, -differencia specifica of attribute. The genum proximum is the group (set) where the data belongs. The differencia specifica is the distinguishing feature that allows the individual identification of data within a certain group (not only ID, due to the usage of path numbers other attributes included).

Similarly to the other variables, the four dimension of knowledge are stored in our code table. This makes it possible for flexible data interpretation and associative search. Integrated mechanisms ensure

that the inputted data from virtual data tables can be put into JT's physical data table automatically. So users do not need to learn JT coding.

3. Results and Discussion

SQL is generally preferred when the aim is a structured format in data storage whereas NoSQL is best when the main objective is big data management.

With Joker Tao we present a database technology which comprises of a NoSQL engine on an SQL platform and serves data from different data storage concepts with no conversion necessary.

At the end of each data in a database, there is a value related to given attributes of the given entity. The above fundament is surrounded by a data-driven shell and JT manipulates this shell as follows:

- queries its value, multiplicity, and existence.
- expands its physical data lines which results in attribute, entity, entity occurrence, definition of the relationship between entities, and relationship occurrence in logic level.

The update of the queried and created data lines results varies depending on the options used to prepare the shell.

3.1. Database

A database was created in a relational environment (Oracle APEX framework) which is based on NoSQL database structure. In this database the following attributes were stored regarding the treatments: *Doses* (control, small, small-medium, medium-high, high), *Culture* (maize mono, bi, tri), *Irrigation* (irrigated, non-irrigated treatment). The basic physical and chemical parameters were: *Silt and clay fraction, Hygroscopicity, Arany-type of plasticity index, Moisture content, Hydrolytic acidity, Organic-C, Nitrate-N, AL-soluble P, AL-soluble K*. The main attributes of carbon cycle were: *Humus, Organic C, Number of bacteria (10^6 cfu g^{-1}), Number of fungus (10^3 cfu g^{-1}), Soil respiration (mg CO2 * $100g^{-1}$ * 10 day^{-1}), MBC extraction, MBC x Organic C^{-1}* (Table 4)

3.2. Query tests

A database was created in Oracle APEX environment based on JT model which ensures faster queries than the usual- relational database models. The faster queries prevailed on every attribute could be indexed in JT (as it shown below) which is not usual in relational environment where generally 2-3 attributes are indexed (Table 3).

```
CREATE TABLE  "JOKER TAO"
   (      "T ID" NUMBER,
          "T ATTRIBUTE" NUMBER,
          "T_VALUE" VARCHAR2(100),
          "O_ID" NUMBER,
           CONSTRAINT "JOKER TAO PK" PRIMARY KEY ("O ID") ENABLE
   ) ;

CREATE INDEX  "JOKER_TAO_IDX1" ON  "JOKER_TAO" ("T_ID")
/
  ALTER INDEX  "JOKER_TAO_IDX1"  UNUSABLE;

CREATE INDEX  "JOKER TAO IDX2" ON  "JOKER TAO" ("T ATTRIBUTE")
  ;

CREATE INDEX  "JOKER_TAO_IDX3" ON  "JOKER_TAO" ("T_VALUE")
  ;

CREATE OR REPLACE TRIGGER  "BI JOKER TAO"
  before insert on "JOKER_TAO"
  for each row
begin
  if :NEW."O_ID" is null then
    select "JOKER TAO SEQ".nextval into :NEW."O ID" from sys.dual;
  end if;
end;
```

```
/
ALTER TRIGGER  "BI_JOKER_TAO" ENABLE;
```

With the developed model, all data tables in a database can be converted to one physical data table. In this case there is no need to use new memory units, so the queries are faster than in the relational data models.

Table 3. Comparison query times

Positive results (datalines)	Relational model (sec)	JT model (sec)
1	<0,01	<0,01
10	0,01	<0,01
100	0,01	<0,01
1000	0,09	0,04
10000	0,87	0,4
18185	1,57	0,89

The query tests based on the Oracle APEX server processing time (despite of the client computer capacity). The results show that in case of 1000 data lines or more (positive results in the query) significantly faster queries could be reached with JT model usage.

3.3. Experimental results

We examined the following soil physical and chemical parameters (Table 4).

Table 4. Some physical and chemical properties of the soil

Parameter	Value
Silt and clay fraction	36,4
Hygroscopicity	2,24
Arany-type of plasticity index	38
Moisture content	19-20
Hydrolytic acidity	6,71
Organic-C	5,63
Nitrate-N	5,94
AL-soluble P	1,4
AL-soluble	7,4
pH H_2O	48,6
pHMKCl	222

In our experiments, we examined the soil microbial processes related the soil carbon cycle (Table 5).

Table 5. The effects of irrigation and fertilization on the triculture treatments

Dose	Humus	Organic C	Number of bacteria (106 cfu g-1)	Number of fungus (103 cfu g-1)	Soil respiration (mg CO_2 * 100g-1 * 10 day-1)	MBC extraction	MBCx Organic C	CO_2 x MBC-1 (qCO2)
21	2,8	1,7	5,9	37,0	14,4	193,0	1,1	0,7
22	3,1	1,7	*4,8	*15,5	*17,1	*210,8	1,2	0,8
23	3	1,9	5,7	*46,0	14,7	*217,8	1,3	0,7
24	*3,1	1,8	*4,0	37,0	15,0	*183,9	1,0	0,8
25	2,9	1,7	*4,2	*25,0	14,3	201,3	1,2	0,7
SzD5%	0,2		1,0	9,0	2,6	15,4		
26	3,0	1,7	4,5	28,0	13,2	67,9	0,4	1,9
27	3,0	1,8	5,1	23,5	*16,3	96,0	0,5	1,7
28	3,0	1,7	4,2	27,0	*16,3	*118,7	0,7	1,4
29	3,1	1,9	*3,5	*45,0	15,4	*357,4	2,0	0,4
30	*3,3	1,9	*5,4	29,0	*16,3	*137,7	0,7	1,2
SzD5%	0,3		0,6	6,3	3,0	43,8		

Our results and those statistics suggest that the bi and triculture influenced higher microbial activity which was reflected by the microbial biomass carbon (MBC), number of fungus and soil respiration.

Conclusion

The interoperability between SQL and NoSQL often garners a lot of attention in the world of database and many solutions have been proposed to solve this issue. In the present study, we present a solution which puts forth a whole new approach that eliminates the need for conversion and file compatibility problems by combining the different data storage concepts into a physical data storage level.

The data storage of long term experiments need flexible database concept, because of the continuous refinement during the researching processes. Data models that ensure faster queries in relational environment could be useful for the scientific projects which expand their datasets continuously. In this model the user does not need to export the scientific results for statistical analysing software, because the statistical formulas can be stored in the same physical data table which is used for the measurement data storage.

The data models which ensure Big Data management in relational environment can be useful for correlation management and continuously data entry processes in long term experiments. These databases are good bases for material cycle models, because new attributes can be determined during and after data entry.

The experimental data from different devices which use different data storage structures can be stored and managed under one platform with no conversion necessary.

Acknowledgement

The corresponding author is thankful to György Mátyás, the idea owner of JT. JT is a Hungarian product which was developed in 2008 (R.number: INNO-1-2008-0015 MFB- 00897/2008) thanks to an EU application.. The authors are also thankful to the Call Tec Consulting Ltd. (the first one with highest Oracle certification in Hungary) to validated JT firstly.

References

(I1)Oracle Databases from web: http://www.oracle.com/us/products/database/overview/index.html

(I2)MySQL Databases from web: http://www.mysql.com/

(I3)Microsoft SQL Server Databases from web: http://www.microsoft.com/en-us/sqlserver/default.aspx

Cabrera, M. L., Chiang, S. C., Merka, W. C., Pancorbo, O. C., & Thompson, S. (1994). Nitrous oxide and carbon dioxide emissions from pelletized and nonpelletized poultry litter incorporated into soil. Plant and Soil, 163, 189-195.,doi: 10.1007/bf00007967

Glatzel, S., Basiliko, N., & Moore, T. (2004). Carbon dioxide and methane production potentials of peats from natural, harvested and restored sites, Eastern Quebec, Canada. Wetlands, 24, 261-267., doi: 10.1672/0277-5212(2004)024[0261:cdampp]2.0.co;2

IPCC (2007). Changes in atmospheric constituents and in radiative forcing. In Climate change, the physical science basis, contribution of working group I to the fourth assessment report of the intergovernmental panel on climate change, Cambridge University Press, Cambridge, UK.

Kong, Y., Nagano, H., Katai, J., Vago, I., Olah, A. Z., Yashima, M., & Inubushi, K. (2013). CO2, N2O and CH4 production/consumption potentials of soils under different land-use types in central Japan and eastern Hungary. Soil Science and Plant Nutrition, 59, 455-462.; doi: 10.1080/00380768.2013.775005

Konstantinou, I., Angelou, E., Boumpouka, C., Tsoumakos, D., & Koziris, N. (2011, October). On the elasticity of nosql databases over cloud management platforms. In Proceedings of the 20th ACM international conference on Information and knowledge management (pp. 2385-2388). ACM.

Leavitt, N. (2010). Will NoSQL databases live up to their promise?. Computer,43(2), 12-14.; doi: 10.1109/mc.2010.58

Mátyás, B., Mátyás, Gy., Szendrei, M., Ankit, S., Yuhua, K., Kátai, J., Zsuposné, O. Á., Kazuyuki, I. (2014) Datamanipulation of correlation between soil properties and GHG emisiion. 9th International Soil Science Congress, The Soul of Soil and Civilization, Book of Proceeding, 583-588. p.

Mátyás, B., Tállai, M., Kátai, J., Horváth, J. (2015) Effects of fertilization on soil microbial parameters, Acta Agraria,

Pyle, D. (1999). Data preparation for data mining. San Francisco, CA: Morgan Kaufmann, Inc..

Singla, A., & Inubushi, K. (2014). Effect of biochar on CH4 and N2O emission from soils vegetated with paddy. Paddy and Water Environment, 12, 239-243., doi:: 10.1007/s10333-013-0357-3

Singla, A., Sakata, R., Hanazawa, S., & Inubushi, K. (2014). Methane production/oxidation potential and methanogenic archaeal diversity in two paddy soils of Japan. International Journal of Ecology and Environmental Sciences, 40, 49-55.

Westerman, P. (2001). Data warehousing: using the Wal-Mart model. San Diego, CA: Academic Press.