🏆 *reviewed paper*

Coping with Dynamic, Unstructured Data Sets – NoSQL a Buzzword or a Savior?

Johannes Scholz

(Dr. Johannes Scholz, TU Vienna, Gusshausstr. 27-29, 1040 Vienna, Austria, scholz@geoinfo.tuwien.ac.at)

1 ABSTRACT

Database research mentions a number of topics that challenge current data storage concepts. Among those topics are: managing and creating collections of unstructured data, horizontal scaling, mobile and cloud computing. As relational databases have limitations due to their underlying relational model, this article shows that upcoming NoSQL databases have characteristics meeting those challenges. Hence, they are widely used in Web 2.0 applications that manage great, unstructured data collections with dynamic content.

An example from land management dealing with the cadastral document collection underpins the argument that NoSQL databases should be considered as data storage in the spatial domain. As documents are regarded as unstructured datasets with a certain dynamic behaviour, they share certain semantics but differ in syntax. This is justified by non existing schema for datasets and the different temporal contexts of documents.

2 INTRODUCTION

Contemporary relational database management systems (RDBMSs) rely on the relational model, published by Codd (1970). The relational model describes the way how data are stored within databases, based on n-ary relations on given sets. Formally, any relation is a subset of the cartesian product of sets (domains), where reasoning is done using a two-valued predicate logic. Data stored according to the relational model are viewed using the table metaphor, where each row represents an n-tuple of the relation itself. Each column of the table is labelled with the name of the corresponding set (domain). Operations on data are performed utilizing relational algebra, that allows to express a rich set of possible data transactions.

RDBMSs inherit the ACID – *a*tomicity, *c*onsistency, *i*solation and *d*urability – properties, which are defined by Gray (1983), ensuring that transactions in any relational database are performed in a reliable manner. Prior to that, Codd (1970, p. 387) mentions consistency as a „*serious practical problem as more and more different types of data are integrated into common data banks*“. In recent RDBMSs consistency is the central issue that limits the performance and horizontal scalability. The latter is of significant importance for Web 2.0 applications with a growing number of users and/or data volumes, which require additional network nodes – i.e. servers.

In 2009, the term NoSQL database has emerged (Evans 2009), which serves as an umbrella term for a number of different database concepts (Edlich, Friedland et al. 2010). Common characteristics of NoSQL databases are a non-relational data model and the absence of ACID properties – especially consistency. These database systems are designed for Web 2.0 applications that require the handling of great data volumes with replication. Currently NoSQL databses are employed as data storage approach for e.g. Amazon, Google, Yahoo or Facebook.

Based on these fundamentals this article reviews the trends and challenges in current database research. Additionally, NoSQL databases are presented that meet a number of those challenges. The research question in this article concerns the applicability of nonrelational databases – in particular Document databases – for the cadastral document collection which contains information for spatial planning. The cadastral document collection is used because it shares a number of commonalities with Web 2.0 data collections, which are unstructured and dynamic in nature. Due to the fact, that NoSQL databases are designed for unstructured, dynamic datasets and furthermore not used in spatial planning so far, this paper elaborates on Document databases for this application area.

The paper is organized as follows. The first part highlights challenges for current relational databases followed by a general introduction in NoSQL databases. The research question is addressed in chapter five, where the nature of the cadastral document collection is analyzed followed by an evaluation of NoSQL databases for the document collection.

3 CHALLENGES FOR RELATIONAL DATABASES

The relational model is roughly 30 years old, given the first publication (Codd 1970). Over the years this approach has proven extremely successful and is regarded as the main database concept, which is underpinned by numerous (commercial) implementations following this concept (e.g. Oracle, PostgreSQL, MySQL). The demands of the upcoming Internet, especially Web 2.0, altered the requirements of databases in a drastic way, which is mentioned in the Claremont report on database research (Agrawal, Ailamaki et al. 2008). In the following paragraphs some of the challenges and research opportunities are discussed.

The report mentions a change in the core database engines, as the requirements are shifted due to the emerging Web 2.0. Traditional relational databases support frequent data read/write operations with low data transfer or voluminous batch processes which require only few write transactions. In addition, RDBMS have poor performance in data-intensive tasks like text indexing and delivering media content (Agrawal, Ailamaki et al. 2008). For RDBMS following the ACID principles consistency is of major importance. Thus, horizontal scaling over a vast number of nodes is not possible, which would require consistent data over a great number of network nodes. As cloud data services and cloud computing are emerging topics that utilize a distributed architecture, databases should support horizontal scaling. Hence, the management of data storage organized in cloud environments is of particular importance, and requires limited human intervention, high-variance workloads as well as a number of shared infrastructures. The latter three factors challenge RDBMSs as they require alterations to support such architectures. In order to overcome the limitations of traditional RDMBS Agrawal, Ailamaki et al. (2008) list important research topics, which including:

- design of continuous self-tuning methods to perform query optimization and physical data layout
- design of systems utilizing non-relational data models
- emphasis on performance and scaling issues rather than consistency and availability

Due to emerging Web 2.0 applications the amount of structured and unstructured data is growing. The management of data can be supported by database technology. To cope with the volume of data available, semantics can be employed to interpret the meaning of data and for search issues. Context is a significant aspect of semantics which can be retrieved from unstructured and heterogeneous data using e.g. text analysis. This cannot be achieved by RDBMSs, as they hardly support unstructured data. In addition, Web 2.0 data collections cannot be „squeezed“ into a predefined schema without causing friction. Thus, the Clarmont report argues for schema free data storage capable of inferring schemas from the data in a dynamic fashion.

Mobile applications become apparently popular through the availabilty of mobile internet and growing market penetration of smartphones. Through a number of Web 2.0 applications available as mobile applications, users deliver and receive data based on their context (e.g. position, personal preferences, social environment) in real-time. Thus, data created by users are heterogeneous and voluminous in nature, and raise the issue of privacy – what is visible to whom? This requires intelligent approaches to handle heterogeneous data streams, and efficient filtering techniques, besides of developing appropriate storage and retrieval methods that consider horizontal scaling.

4 NOSQL DATABASES

A coherent defintion of NoSQL databases does not exist, due to a lack of organizations in this particular field. The first attempt to define the term NoSQL databases is published in Edlich, Friedland et al. (2010). There databases that follow some of the following principles are regarded as NoSQL databases:

- non-relational data model
- tailored towards distributed and horizontal scalability
- open source
- schema free or at least weak schema restrictions
- support for a simple replication approach
- simple application programming interface
- other consistency approaches are used – *eventual consistency* and *basically available, soft state eventually consistent* (BASE) but not ACID

4.1 Global characteristics and theoretical concepts

As NoSQL databases follow the principles listed above, theoretical concepts are necessary that enable the implementation of the NoSQL functionalities. A number of fundamental concepts are mentioned in literature: map/reduce, eventual consistency, consistent hashing, multi version concurrency control and vector clocks.

The fundamental concept of NoSQL databases is the underlying Map/Reduce approach (Dean and Ghemawat 2004) which is a framework that supports the handling of large data volumes over distributed network nodes. To handle very large data volumes and operations on them, any problem is divided into subproblems that are distributed to other network nodes – the *map* step. Subsequently, these network nodes are allowed to do the same distribution process. In the end the subproblems are solved by certain network nodes and the results are passed on to the master node that interprets the results – *reduce* step.

Eventual consistency (Vogels 2008) is a concept that is a direct consequence of Brewers *c*onsistency, *a*vailability and *p*artition tolerance (CAP) theorem (Lynch and Gilbert 2002). The CAP theorem argues that the concepts of consistency, availability and partition tolerance cannot be provided by any computer system in a simultaneous manner. Furthermore, any distributed database can only fulfill two principles. Hence, distributed databases for the Web 2.0 should emphasize availability and partition tolerance over consistency. Thus, a consistent state is eventually reached in NoSQL systems, but not immediately after a transaction. This consistency approach is denoted as BASE (Pritchett 2008).

Consistent hashing is a principle that supports storage and retrieval mechanisms in distributed data storages, where the number of network nodes is subject to change (Karger, Lehman et al. 1997). This change is a result of network errors or hardware failure and/or the process of adding new servers to the system during the lifecycle of an application. These changes in the physical system architecture do not result in an extensive data migration, due to the consistent hashing principle (Decandia, Hastorun et al. 2008). This principle uses an address space with connected ends – a ring – where the hash values of network nodes are assigned to. Thus, if changes are made to the network layout (removal or addition of a server), only the network nodes in the vicinity of the hash value of the altered network node are affected (Edlich, Friedland et al. 2010). Hence, this principle reduces data migration operations in dynamic distributed data storage systems.

Multiversion concurrency control (MVCC) is an approach to support concurrent transactions in a database. As opposed to classical transaction management using locks on the datasets, MVCC allows parallel read and write operations. MVCC realizes this by creating new versions of the dataset as a write process occurs. The new dataset is tagged with a unique version number and the number of preceding version of the dataset – which is altered by the current write process. Thus, it allows that previous versions of a dataset can be queried and version conflicts can be resolved either by the system or the user. The MVCC approach does not make any statement about the identification of dataset versions – i.e. version numbering.

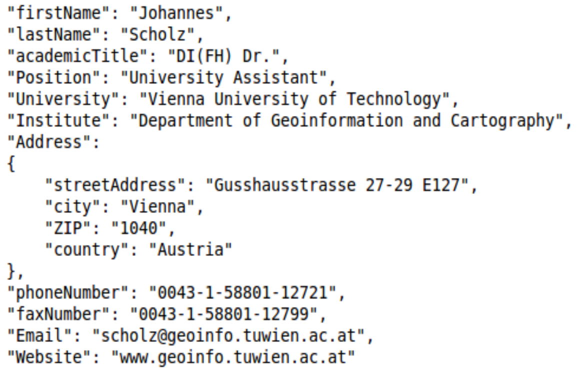
For distributed systems vector clocks (Fidge 1988, Mattern 1988, Lamport 1978) are useful to generate an ordering of events in a system. In NoSQL systems data are dispersed over a number of network nodes in a redundant way. As network nodes may update datasets in an uncoordinated manner, any network node must be able to determine the current version of a certain dataset. Hence, each network node – i.e. database – is able to detect different versions and decide on the ordering of versions autonomously.

4.2 Types of NoSQL Databases

A classification of NoSQL systems is complex, as a great number of different database projects identify themselves as NoSQL systems. Edlich, Friedland et al. (2010) publish a categorization that lists four types: Wide Column Stores, Document Stores, Key/Value/Tuple Stores, Graph Databases. These categories will be briefly described in the next paragraphs.

Based on the original idea, Wide Column Stores are defined as databases where attributes are stored in a column oriented way, based on their data type (Khoshafian, Copeland, et al. 1987, Abadi 2007). Due to the fact that contemporary Wide Column Store projects like Google's BigTable are described as “sparse, distributed multi-dimensional sorted map” (Chang, Dean et al. 2006) they do not entirely follow the original idea of Wide Column Stores. Nevertheless, they are designed to support the storage of voluminous data in distributed environments. Multi-dimensionality in BigTable is achieved by storing multiple versions of a dataset (Chang, Dean et al. 2006), which are identified by attached timestamps. The Wide Column approach has proven successful, as it is employed as data storage for e.g. Amazon or Google. Nevertheless, Edlich, Friedland, et al. (2010) mention that these projects do not support complex queries like the join operation.

Document Stores are databases that are able to handle data collections like JSON or RDF (see Fig. 1) in conjunction with a unique identifier. Important for document stores is the fact that they are schema free, which is important for Web 2.0 applications. Thus, the structure of a dataset does not have to be defined beforehand – the application on top of the document based storage cares about a schema. That is important when dealing with documents with varying syntax. An example that is highlighted in Anderson, Lenhart, et al. (2010) analyzes business cards. Business cards contain similar information, but differ in syntax. As one person might have a fax number on his/her business card, another person does not mention this. As humans we are able to deal with this fact that documents (i.e. real world objects) differ in syntax, but share certain semantics. In a relational database this fact has to be modeled accordingly, whereas this is not necessary in a document based storage. There data can be aggregated by the database and/or the application after they have been entered into the database. Of importance is the possibility to query documents stored in the database using a published application programming interface – so called Views in CouchDB.



```
"firstName": "Johannes",
"lastName": "Scholz",
"academicTitle": "DI(FH) Dr.",
"Position": "University Assistant",
"University": "Vienna University of Technology",
"Institute": "Department of Geoinformation and Cartography",
"Address":
{
    "streetAddress": "Gusshausstrasse 27-29 E127",
    "city": "Vienna",
    "ZIP": "1040",
    "country": "Austria"
},
"phoneNumber": "0043-1-58801-12721",
"faxNumber": "0043-1-58801-12799",
"Email": "scholz@geoinfo.tuwien.ac.at",
"Website": "www.geoinfo.tuwien.ac.at"
```

Fig. 1: The author's business card in a JSON representation.

Key/Value/Tuple Stores are databases that utilize a storage of keys and associated values without any relations. To support scaling issues the key/values tuples are stored in a distributed database system using a hash table. The advantages of Key/Value Stores lie in the ability to cope with voluminous data in a distributed environment especially when scaling out.

Graph databases are databases following graph theoretical approaches. Thus, they have nodes, their embedded properties and edges with embedded properties (Angles and Gutierrez 2008). Properties are key/value relationships that have a certain predefined schema. The elements of a graph are identified by a unique number. The nodes in graph databases represent entities and edges the relations between them. Additionally, edges have an orientation, which allows the modeling of complex relationships between entities. Query techniques allow the retrieval of the entities and their relationships based on graph theoretical algorithms like Breadth First Search, Depth First Search, determination of Hamilton cycles. Thus, the algorithms are capable of analyzing transitive relationships, which would be hardly possible in relational database systems. Examples of such transitive relationships are ancestor searches: Who is the great-grandfather of person X, or which ancestors were born after 1800? These are variable-length joins which are natively supported by a graph data structure.

5 NOSQL FOR SPATIAL DOMAIN?

In the preceding sections the paper states that NoSQL databases are designed for large data volumes and Web 2.0 applications. In the context of Web 2.0 the spatial dimension is of particular interest, as countless applications like Flickr or Panoramio enable the user to georeference their content. First attempts towards storing spatial data in Document databases – mongoDB and CouchDB – have been undertaken. Furthermore, the specification of geoJSON gives evidence that the spatial capabilities of document based storages will be extended. In addition, Graph databases support spatial data natively, as coordinates can be assigned nodes and topology is handled in the systems as well.

The next two sections focus on the usage of document databases – as one representative of NoSQL databases – in the spatial domain. In detail, selected documents of the cadastral document collection are analyzed concerning their semantics and syntax. Subsequently, the advantages of document databases for the document collection are analyzed, with special consideration of spatial planning issues.

5.1 Cadastral document collection

This paper elaborates on the cadastral document collection and in particular on the purchase of land contracts which are part of the document collection (Republik Österreich 2010; Kodek 2007; Feil, Marent et al. 2005). Due to the fact that these contracts are the basis for e.g. property subdivisions and changes in the cadastre (ownership, land register identification number) they are an important source of information for planning issues. The section highlights the issue that the syntax of documents changes while preserving a certain semantic due to the temporal context and/or not fixed schema of data. This is shown on the basis of historic and recent purchase of land contracts. Current purchase of land contracts should contain the following data (Österreichische Notariatskammer 2011):

- vendor
- purchaser
- property: description of the property (e.g. non built-up, built-up, farmed land, condominium, commercial/industrial object);
- legal situation: e.g. servitudes, natural hazard zone, spatial planning zone, building permissions;
- infrastructure: description of e.g. roads, connection to sewerage system, electricity
- purchase price
- payment modalities
- defects liability: e.g. pollution legacy
- entry in other registers: e.g. water register, fishing register

In Table 1 the listed contract items are analyzed concerning their relevance for spatial planning and if these items can be "squeezed" into a fixed database schema. Here the items vendor and purchaser are identified as being more related to a fixed schema, due to existing "standard" elements – e.g. first name, surname, address, city, zip. Items purchase price and payment modalities are regarded as having more relation to a fixed schema, analogue to vendor or purchaser. Items property, legal situation, infrastructure, defects liability and other registers are described in an unstructured way and are not part of a contract per se and/or vary in content. The contract items which can hardly described using a predefined schema can be seen as unstructured data, as they have no common schema.

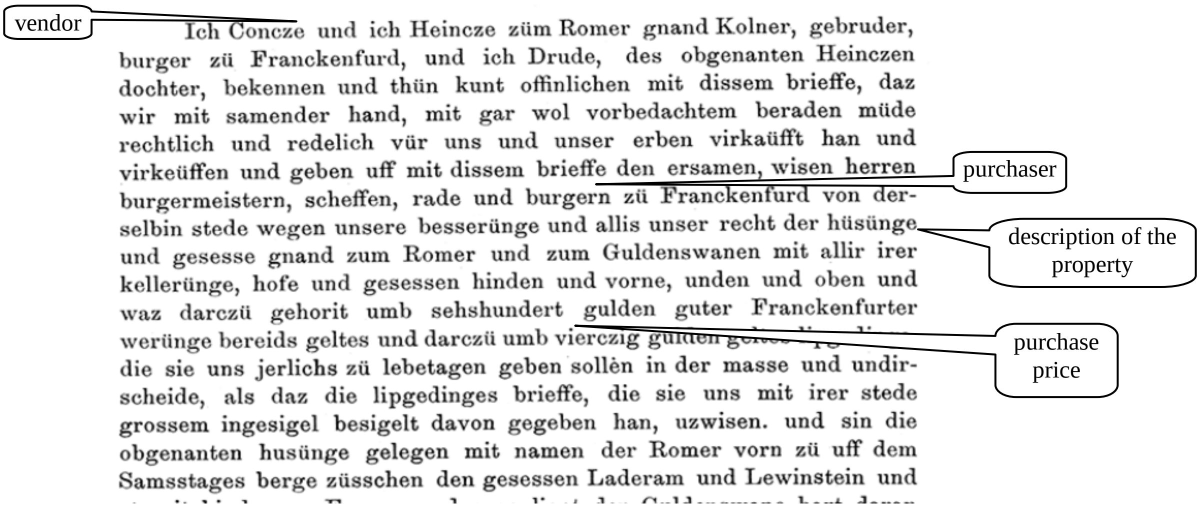
contract item	relevant for spatial planning	fixed schema	no fixed schema
vendor		✔	
purchaser		✔	
Property	✔		✔
legal situation	✔		✔
infrastructure	✔		✔
purchase price		✔	
payment modalities		✔	
defects liability	✔		✔
other registers			✔

Table 1: Items of a purchase of land contract, their relevance for spatial planning and classification if the item can be handled with a fixed schema.

Additionally, historical documents in the document collection show the genesis of parcels. Documents having a certain age have a different syntax than recent documents, as e.g. natural hazard zoning started in

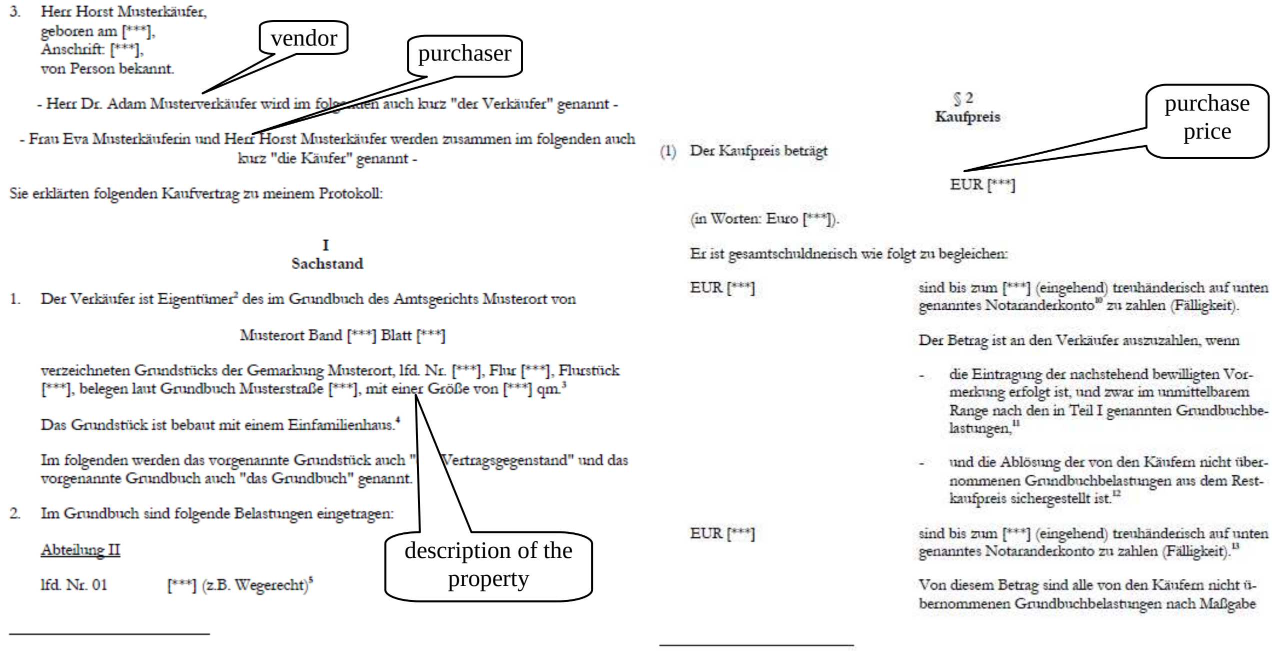
1975 as an effect of the Austrian Forest Act. Due to the fact that the Austrian Conference on Spatial Planning was founded in 1971, there are hardly any contracts before the 1970ies that contain information on spatial planning zones – in fact they are mostly not found in contemporary contracts either. Looking at historical and contemporary purchase of land contracts the syntax change is obvious. In Figure 2 an excerpt of a historic contract is presented, which documents the purchase of the Römer building in Frankfurt/Main in the year 1405 – the town hall of this city. In Figure 2 the items vendor, purchaser, description of the property and purchase price are identified and marked accordingly. The same items are highlighted in Figure 3 – a contemporary purchase of land specimen. It is observable that the meaning – i.e. the semantics – of the contract did not change very much, as similar items of a contract can be found in the historic and the contemporary contract. Obviously, the syntax did change, as language, the monetary system and the way how ownership and property are identified – i.e. land register identification number, parcels – have changed. Hence, as the document collection contains documents with different temporal context, there is a shift in syntax to a great and in semantics to a lesser extent.

These arguments mentioned in the previous paragraphs underpin the fact, that the cadastral document collection contains unstructured data. Even when looking at one particular class of documents – purchase of land contracts – a change of syntax is observable, while semantics remains fixed. This can be explained with the temporal context of documents and the absence of a fixed schema for specific items of a document.



Ich Concze und ich Heincze züm Romer gnand Kolner, gebruder, burger zü Franckenfurd, und ich Drude, des obgenanten Heinczen dochter, bekennen und thün kunt offinlichen mit dissem brieffe, daz wir mit samender hand, mit gar wol vorbedachtem beraden müde rechtlich und redelich vür uns und unser erben virkaüfft han und virkeüffen und geben uff mit dissem brieffe den ersamen, wisen herren burgermeistern, scheffen, rade und burgern zü Franckenfurd von derselbin stede wegen unsere besserünge und allis unser recht der hüsünge und gesesse gnand zum Romer und zum Guldenswanen mit allir irer kellerünge, hofe und gesessen hinden und vorne, unden und oben und waz darczü gehorit umb sehshundert gulden guter Franckenfurter werünge bereids geltes und darczü umb vierczig gulden [illegible] die sie uns jerlichs zü lebetagen geben sollèn in der masse und undirscheide, als daz die lipgedinges brieffe, die sie uns mit irer stede grossem ingesigel besigelt davon gegeben han, uzwisen. und sin die obgenanten husünge gelegen mit namen der Romer vorn zü uff dem Samsstages berge züsschen den gesessen Laderam und Lewinstein und

Fig. 2: Excerpt of the purchase of land contract of the Römer building (Frankfurt/Main) from 1405, with selected items of the contract



3. Herr Horst Musterkäufer,
geboren am [***],
Anschrift: [***],
von Person bekannt.

- Herr Dr. Adam Musterverkäufer wird im folgenden auch kurz "der Verkäufer" genannt -

- Frau Eva Musterkäuferin und Herr Horst Musterkäufer werden zusammen im folgenden auch kurz "die Käufer" genannt -

Sie erklärten folgenden Kaufvertrag zu meinem Protokoll:

I
Sachstand

1. Der Verkäufer ist Eigentümer[2] des im Grundbuch des Amtsgerichts Musterort von

Musterort Band [***] Blatt [***]

verzeichneten Grundstücks der Gemarkung Musterort, lfd. Nr. [***], Flur [***], Flurstück [***], belegen laut Grundbuch Musterstraße [***], mit einer Größe von [***] qm.[3]

Das Grundstück ist bebaut mit einem Einfamilienhaus.[4]

Im folgenden werden das vorgenannte Grundstück auch "Vertragsgegenstand" und das vorgenannte Grundbuch auch "das Grundbuch" genannt.

2. Im Grundbuch sind folgende Belastungen eingetragen:

<u>Abteilung II</u>

lfd. Nr. 01 [***] (z.B. Wegerecht)[5]

§ 2
Kaufpreis

(1) Der Kaufpreis beträgt

EUR [***]

(in Worten: Euro [***]).

Er ist gesamtschuldnerisch wie folgt zu begleichen:

EUR [***] sind bis zum [***] (eingehend) treuhänderisch auf unten genanntes Notaranderkonto[10] zu zahlen (Fälligkeit).

Der Betrag ist an den Verkäufer auszuzahlen, wenn

- die Eintragung der nachstehend bewilligten Vormerkung erfolgt ist, und zwar im unmittelbarem Range nach den in Teil I genannten Grundbuchbelastungen,[11]
- und die Ablösung der von den Käufern nicht übernommenen Grundbuchbelastungen aus dem Restkaufpreis sichergestellt ist.[12]

EUR [***] sind bis zum [***] (eingehend) treuhänderisch auf unten genanntes Notaranderkonto zu zahlen (Fälligkeit).[13]

Von diesem Betrag sind alle von den Käufern nicht übernommenen Grundbuchbelastungen nach Maßgabe

Fig. 3: Excerpt of a contemporary purchase of land contract example.

As the cadastre is a “living” environment, the content is subject to constant change. Typical changes are the result of purchase, subdivision, merge, creation or deletion processes in the cadatstre. Through these processes new documents are created that are stored in the document collection. Hence, a number of documents and contracts are attached to each parcel in the course of its lifetime. For the year 2009 a number of 3.1 Million entries exist in the Austrian land register with approximately 10.4 Million parcels (Österreichisches Justizministerium 2010). In 2009 about 684000 new entries are created in the Austrian land register, and the system managed and answered 12 million queries. These are low numbers compared to global Web 2.0 platforms like Facebook or Twitter. Nevertheless, the system shows a certain dynamic behavior, with a number of update and read processes. In addition, the document collection has to cope with voluminous data as whole documents have to be handled properly.

5.2 NoSQL document database for document collection?

In section 5.1 the nature of the cadastral document collection is analyzed which results in the finding that the document collection shows characteristics of an unstructured data set. Furthermore, the documents share a certain semantic but differ in syntactics. Finally, the cadastral document collection is a dynamic database, with a number of queries and updates to be processed.

Given the nature of the cadastral document collection, a Document Store meets the characteristics and requirements to a high degree. Table 2 highlights similarities of Document Stores and the document collection, which support the argument of considering NoSQL databases for the document collection. In Table 2 the purpose and the nature of the document collection and the characteristics of Document Stores are listed, which coincides as both deal with unstructured data that have to be stored and retrieved accordingly. NoSQL databases support dynamics in data management as multiple versions can be identified and queried, which is – at least to a certain degree – important for the cadastral document collection. Eventual concurrent processes are handled by MVCC in a NoSQL database. Such processes may happen, as a number of land registry courts access and alter the cadastral document collection. Data security is of significant importance for the cadastral document collection and can be realized with a distributed data storage utilizing a number of network nodes exploiting the functionalities of NoSQL databases (Map/Reduce, Consistent Hashing, Vector Clocks) – which opposes classical backup strategies.

	Document collection	Document Database (NoSQL)
purpose	storage and retrieval of documents	storage, retrieval and query mechanisms for documents
nature of data (content)	predominantly unstructured	structured and unstructured (schema free)
dynamic behavior	yes (at least to a certain extent)	supports dynamic processes
multiple versions	may exist	multipe versions are supported (MVCC)
concurrernt processes	may exist	handled by MVCC
data security	important	supports distributed data storage

Table 2: Similarities of cadastral document collection and NoSQL Document Storage.

NoSQL based Document Stores enable several functionalities which are of interest for spatial planning issues. The possibility to search in documents enables spatial planners to analyze and relate a number of documents with the help of database technology, which helps to uncover the information treasure that hides in the document collection. Thus, documents can be evaluated and analyzed concerning several parameter, like property price levels. Additionally, this makes information accessible, which is only present in the contracts and is not considered otherwise, like (legal) restrictions defined in the contract, but not mentioned as servitude in the land register.

Data mining on the document collection can be enhanced by incorporating the spatial dimension and spatial relations. As NoSQL databases start to support spatial indexing and spatial data storage, this functionality will be available soon, which will foster the analytic capabilities of the non-relational databases. Hence, complex (spatial) queries addressing spatial planning questions can be answered by mining unstructured data too. Currently the following NoSQL projects support spatial data (at least to a minor degree):

- CouchDB with GeoCouch (Mische 2010, 2011)
- mongoDB (10gen Inc. 2011)
- Neo4j using Neo4j spatial (Neo Technology 2011)

Hence, NoSQL databases should be considered as an alternative for specific tasks. Nevertheless, standard RDBMS will remain in usage as NoSQL systems should not be used for each and every purpose. Edlich, Friedland et al. (2010, pp. 271-283) mention that a requirements analysis is necessary to decide on the basic database approach – relational or non-relational. This is necessary as NoSQL systems are able to handle certain requirements well, e.g. horizontal scaling (cloud computing), schema free data, replication issues. On the other hand, if consistency, security or complex query processes are required, classical RDBMS are preferable.

6 CONCLUSION

This paper elaborates on non-relational databases, which are subsumed under the umbrella term NoSQL databases. Relational databases are facing a number of challenges due to Web 2.0 applications, distributed architectures and a growing amount of unstructured data. As NoSQL databases address these issues, as they are designed to meet the requirements of Web 2.0. This article analyzes the possibilities of NoSQL databases for spatial planning issues and the spatial domain in general, based on the cadastral document collection.

The nature of elements of the cadastral document collection is closely related to unstructured data. Contained documents share similar semantics but differ syntactically which can be explained with the temporal context and the fact that certain items of documents cannot be described with predefined schemata – which is shown based on purchase of land contracts. Additionally, the document collection has a certain dynamics, as a number of update processes take place.

The capabilities of Document stores – a type of NoSQL databases – meet the requirements and characteristics of the cadastral document collection. Hence, they should be considered as an alternative to relational databases. Of importance for spatial planning processes may also be data concerning the history of a piece of land and (legal) constraints not mentioned in the land register. The ability of Document Stores to query documents addresses this issue. Furthermore, spatial data handling and indexing is implemented in NoSQL databases, which increases their analytical capabilities. As the spatial enablement of NoSQL databases is – except for graph databases – on a “basic” level, there is space for improvement. Nevertheless, they offer new possibilities for analyzing unstructured spatial data present in distributed environments.

Concluding, NoSQL databases have advantages for applications that share characteristics with Web 2.0. Nevertheless, traditional relational databases are designed to support consistent transactions, high security, and complex queries and thus will remain popular. Through availability of a variety of database approaches developers have the freedom to choose the concept that satisfies the application requirements best.

7 REFERENCES

10GEN INC.: mongoDB Manual: Geospatial Indexing, Url: http://www.mongodb.org/display/DOCS/Geospatial+Indexing, visited: 24-02-2011, 2011.

ABADI, D.: Column-Stores For Wide and Sparse Data. In: Proceedings of the 3rd Biennial Conference on Innovative Data Systems Research (CIDR), Url: http://db.csail.mit.edu/projects/cstore/abadicidr07.pdf, visited 17-02-2011, 2007.

AGRAWAL, R., AILMAKI, A., BERNSTEIN, P. A., BREWER, E. A., CAREY, M. J., CHAUDHURI, S., DOAN, A., FLORESCU, D., FRANKLIN, M. J., GARCIA-MOLINA, H., GEHRKE, J., GRUENWALD, L., HAAS, L. M., HALVEY, A. Y., HELLERSTEIN, J. M., IOANNIDIS, Y. E., KORTH, H. F., KOSSMANN, D., MADDEN, S., MAGOULAS, R., OOI, B. C., O'REILLY, T., RAMAKRISHNAN, R., SARAWAGI, S., STONEBAKER, M., SZALAY, A. S., WEIKUM, G.: The Claremont report on database research. ACM SIGMOD Record, Volume 37, Issue 3, 9-19, 2008.

ANDERSON, J. C., LEHNARDT, J., SLATER. N: CouchDB: The Definitive Guide. O'Reilly Media, Inc., 2010.

ANGLES, R., GUTIERREZ, C.: Survey of graph database models. ACM Computing Surveys, Vol. 40, Issue 1, pp. 1:1-1:39, 2008.

CHANG, F., DEAN, J., GHEMAWAT, S., HSIEH, W. C., WALLACH, D. A., BURROWS, M., CHANDRA, T., FIKES, A., GRUBER, R. E.: Bigtable: A distributed storage system for structured data. In: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation OSDI'06, pp. 205–218, 2006.

CODD, E.F.: A Relational Model of Data for Large Shared Data Banks. In: Communications of the ACM, Vol. 13, Issue 6, pp. 377–387, 1970.

DEAN, J. and GHEMAWAT, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI'04, 6th Symposium on Operating Systems Design and Implementation, Url: http://labs.google.com/papers/mapreduce-osdi04.pdf, visited: 17-02-2011, 2004.

DECANDIA, G., HASTORUN, D., JAMPANI, M., KAKULAPATI, G., LAKSHMAN, A., PILCHIN, A., SIVASUBRAMANIAN, S., VOSSHALL, P., VOGELS, W.: Dynamo: Amazon's highly available key-value store. In: SOSP'07 - Proceedings of 21st ACM SIGOPS Symposium on Operating Systems, pp. 205-220, 2007.

EDLICH, S., FRIEDLAND, A., HAMPE, J., BRAUER, B.: NoSQL: Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken. Hanser Fachbuchverlag, München, 2010.

EVANS, E.: NOSQL 2009. Url: http://blog.sym-link.com/2009/05/12/nosql_2009.html, visited: 16-02-2011, 2009.

FEIL, E.: Grundbuchsgesetz. Linde Verlag, Vienna 1998.

FEIL, E., MARENT, K.-H., PREISL, G.: Grundbuchsrecht. Linde Verlag, Vienna, 2005.

FIDGE, C. J.: Timestamps in Message-Passing Systems That Preserve the Partial Ordering. In: K. Raymond (Ed.). Proceedings of the 11th Australian Computer Science Conference (ACSC'88), pp. 56–66, 1988.

GRAY, J.: The Transaction Concept: Virtues and Limitations. In: Proceedings of the 7th International Conference on Very Large Databases, pp. 144–154, Cannes, France, 1981.

HAERDER, T., REUTER, A.: Principles of Transaction-Oriented Database Recovery. In: ACM Computing Surveys, Vol. 15, Issue 4, pp. 287–317, 1983.

KARGER, D., LEHMAN, E., LEIGHTON, T., PANIGRAHY, R., LEVINE, M., AND LEWIN, D.: Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on theory of Computing. STOC '97. ACM Press, pp. 654-663, New York, 1997.

KHOSHAFIAN, S., COPELAND, G., JAGODIS, T., BORAL H., VALDURIEZ, P.: A query processing strategy for the decomposed storage model. In: ICDE, pp. 636–643,1987.

KODEK, G. E.: Kommentar zum Grundbuchsgesetz. Manz, Vienna, 2007.

LAMPORT, L.: Time, Clocks, and the ordering of events in a distributed system. Communications of the ACM, Volume 21, Issue 7, pp. 558-565, 1978.

LYNCH, N., GILBERT, S.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, Volume 33, Issue 2, pp. 51-59, 2002.

MATTERN, F.: Virtual Time and Global States of Distributed Systems. In: Cosnard, M. (Ed.), Proceedings of the Workshop on Parallel and Distributed Algorithms, pp. 215–226, France, 1988.

MISCHE, V.: Welcome to the world of GeoCouch. Url: https://github.com/vmx/couchdb/, visited: 24-02-2011, 2011.

MISCHE, V.: GeoCouch: A spatial index for CouchDB. Presentation at FOSS5G 2010, Url: http://2010.foss4g.org/presentations/3048.pdf, visited: 24-02-2011, 2010.

NEO TECHNOLOGY INC.: Neo4j spatial wiki. Url: http://wiki.neo4j.org/content/Neo4j_Spatial, visited: 24-02-2011, 2011.

ÖSTERREICHISCHE NOTARIATSKAMMER: Checkliste Kaufvertrag/Liegenschaft, Url: http://www.notar.at/notar/de/home/infoservice/checklisten/kaufvertrag__liegenschaft/, visited: 17-02-2011, 2011.

ÖSTERREICHISCHES JUSITZMINISTERIM: IT-Anwendungen in der österreichischen Justiz. Url: http://www.justiz.gv.at/internet/file/8ab4ac8322985dd501229ce3fb1900b4.de.0/folder_justiz-online_0310_de.pdf, visited: 23-02-2011, 2010.

PRITCHETT, D.: BASE: an ACID alternative. ACM Queue, Vol. 6, Issue 3, pp. 48-55, 2008.

RAYNAL, M.: About Logical Clocks for Distributed Systems. ACM SIGOPS Operating Systems Review, Vol. 26, Issue 1, pp. 41-48, 1992.

REPUBLIK ÖSTERREICH: Grundbuchsgesetz 1955, BGBl. Nr. 39/1955, in der Fassung BGBl. I Nr. 58/2010, 2010

VOGELS, W.: Eventually Consistent. Queue, Vol 6, Issue 6, pp. 14-19, 2008.