TALLINN UNIVERSITY OF TECHNOLOGY

Department of Computer Engineering

Analysis of programming languages (IAG0450)

A Comparison of SQL and NoSQL Databases

Eyyup Direk IASM 156396

Sergii spivakov IASM157382

Alex Semojonov IASM153752

Tallinn 2016

CONTENTS

Tallinn 2016

INTRODUCTION

Our lives are flooded with all kind of information.Thanks to databases we interact with information easily and seamlessly on a daily basis.When we look at the database definitions more or less we would see something like these

"A database is an organized collection of information treated as a unit. The purpose of a database is to collect, store, and retrieve related information for use by database applications. "[1]

"A database is a collection of information that is organized so that it can easily be accessed, managed, and updated."[2]

Databases were used to organize information long before the computer age.it would be a good idea to take a closer look to Databases history.

DATABASES HISTORY

if we go back to first ages of human civilization ,just after the invention of writing, we could easily see people were using the clay tablets or carvings on walls to be able to record data.If we come little bit closer to near past we would encounter first stages of modern times databases ,which examples are ship manifests,card catalogs and product inventories,libraries,governmental records and statistics . These things gave a necessary birth of databases.

Computers allowed us to automate our databases.Early computer databases were constructed with flat file model a single consecutive list of records but when it comes to search, this was an inefficient way.We needed to search and maintain large volumes of records,which has to be faster ,reliable and safe.In 1960s Ibm has used a hierarchical model for their information management systems. This was constituted by tree structural systems which every node has pointer to its child nodes it has been used successfully by nasa for the lunar lander. A more flexible database model was developed by Charles bachman, which was using the same tree structure but every node might have more than one parent nodes yet when the database get complicated it was hard to manage all the pointers between nodes.

In 1970s Ted Codd has developed relational database model which was proposing to organize the data into simple tables with related information. There were no pointers to maintain, relations were maintained by having matching data fields in each table.It made it a lot easier to access, merge and change data. Several companies used it as a base for commercial products. In 1975 IBM produced an experimental relational databases named system R. it used structured query language developed by Don Chamberlain and Raymond boyce to search and modify data.

In 1977 ,Oracle was introduced as a first commercially available relational database compatible with sql. Since then many companies and individuals have contributed the evolution of relational databases and structured query languages.

Parallel to sql and relational databases, there has been another way of implementing databases but it wasn't as popular as relational –databases until big data era. Today ,we are all connected any social networks and services with our smart phones ,tablets and near future with internet of things. Introducing of this services into our life bring another necessity about the way of recording data which gained to no-sql popularity. The idea of No-Sql is on the contrary of the sql ,it doesn't require strict rules and it is very flexible and scalable and besides dominant web and social service companies like google, facebook, yahoo, it also targets the startup-companies, enterprise companies and open-source developers which requires to be very flexible in aspect of database models.

RELATIONAL DATABASES(SQL)

As it is mentioned in history part ,relational databases exists since 1970s and E. F. Codd defined a relational model based on mathematical set theory.

A relational database is a database that conforms to the relational model. The relational model has the following major aspects:

- Structures

Well-defined objects store or access the data of a database.

- Operations

  Clearly defined actions enable applications to manipulate the data and structures of a database.

- Integrity rules

  Integrity rules govern operations on the data and structures of a database.

A relational database stores data in a set of simple relations. A relation is a set of tuples. A tuple is an unordered set of attribute values.

A table is a two-dimensional representation of a relation in the form of rows (tuples) and columns (attributes). Each row in a table has the same set of columns. A relational database is a database that stores data in relations (tables). For example, a relational database could store information about company employees in an employee table, a department table, and a salary table.

Relational Database Management System (RDBMS)

Basicly,Rdbms is a product or system which presents data as collection of rows and columns .Besides that it requires using SQL as a query language which allows user to retrieve ,delete,manipulate, create and all kind of transactions to implement. An RDBMS distinguishes between the following types of operations:

Logical operations

In this case, an application specifies what content is required. For example, an application requests an employee name or adds an employee record to a table.

Physical operations

In this case, the RDBMS determines how things should be done and carries out the operation. For example, after an application queries a table, the database may use an index to find the requested rows, read the data into memory, and perform many other steps before returning a result to the user. The RDBMS stores and retrieves data so that physical operations are transparent to database applications.

The most common RDBMSs are Oracle,MySql,PostgreSql,MsSql.All this database management systems support relational model as represented by SQL language.

SQL is a set-based declarative language that provides an interface to an RDBMS such as Oracle Database. In contrast to procedural languages such as C, which describe how things should be done, SQL is nonprocedural and describes what should be done.[3]

SQL is the ANSI standard language for relational databases. All operations on the data in an Oracle,MySql,PostgreSql,MsSql databases are performed using SQL statements. For example, you use SQL to create tables and query and modify data in tables.

A SQL statement can be thought of as a very simple, but powerful, computer program or instruction. Users specify the result that they want (for example, the names of employees), not how to derive it.

Examples of SQL statements;

```
Select item_name,item_value ,order_date from orders;

INSERT INTO suppliers

(supplier_id, supplier_name) VALUES(5000, 'Apple');
```

SQL statements have 2 main groups. One is called DDL(data definition language) and the other is DML(data manipulation language) which includes normal select statements.

DDL

Main ddl statements are CREATE, ALTER, or DROP which really defines or totally deletes an object or type or tables structure.

DML

Main dml statements are SELECT,INSERT,DELETE,UPDATE ,ALTER which query or change the contents.

The database must ensure and have the integrity which most of the time case must be so that multiple users can work concurrently without corrupting one another's data.In the heart of the integrity and consistency ,there is a term which is called transaction which is a logical, atomic unit of work that contains one or more SQL statements. An RDBMS must be able to group SQL statements so that they are either all committed, which means they are applied to the database, or all rolled back, which means they are undone.

To be able to maintain this integrity the transactions have 4 important characteristics that is simply called ACID consists of the first characters of atomicitiy,consistency,isolation and durability properties.

Atomicity

Transactions are all-or-nothing. Either all operations go through, or

none do.

Consistency

Transactions lead database from one consistent state to another.

Isolation

Transactions cannot see intermediate (not committed) results of each

other.

Durability

DBMS must ensure that after committing a transaction all its

changes are saved (they can't get lost, for instance, because of power

failure).

NON-RELATIONAL DATABASES

It is a new trend in database world tough, in reality it is not a totally new thing. Such databases have existed since the late 1960s, but did not obtain the "NoSQL" moniker until a surge of popularity in the early twenty-first century,the term of “NoSQL” was in fact first used by Carlo Strozzi in 1998 as the name of file-based database he was developing. Ironically it’s relational database just one without a SQL interface . As such it is not actually a part of the whole NoSQL movement we see today. The term re-surfaced in 2009 when Eric Evans used it to name the current surge in non-relational databases

In contrast to relational database, NoSQL databases do not rely on table,column,row -like structures and use more flexible data models. NoSQL can mean “not SQL” or “not only SQL.”

As RDBMS have increasingly failed to meet the performance, scalability, and flexibility needs that next-generation, data-intensive applications require, NoSQL databases have been adopted by mainstream enterprises. NoSQL is particularly useful for storing unstructured data, which is growing far more rapidly than structured data and does not fit the relational schemas of RDBMS. Common types of unstructured data include: user and session data; chat, messaging, and log data; time series data such as IoT and device data; and large objects such as video and images.

NoSQL TYPES

NoSql databases differs from each other by data model or another way to say how they store the data.

Key-Value databases

Key-value stores are the simplest NoSQL data stores to use from an API perspective. The client can either get the value for the key, put a value for a key, or delete a key from the data store. Stored data can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key, without caring or knowing what's inside; it's the responsibility of the application to understand what was stored. Since key-value stores always use primary-key access, they generally have great performance and can be easily scaled.[4]

Main popular key-value databases are *Riak*, *Redis , Berkeley DB, upscaledb* (especially suited for embedded use), *Amazon DynamoDB* (not open-source), *Project Voldemort and Couchbase*.

Document databases

Documents are the main concept in document databases. The database stores and retrieves documents, which can be XML, JSON, BSON, and so on. These documents are self-describing, hierarchical tree data structures which can consist of maps, collections, and scalar values. The documents stored are similar to each other but do not have to be exactly the same.

```
# Person Collection (Person is the root entity)
[
{
    "ID": 0,
    "first_name": "Steven",
    "last_name": "Edouard",
    "accounts": [
        {
            "id": 0,
            "account_type": "Investment",
            "account_balance": "80000.00",
            "currency": "USD"
        },
        {
            "id": 1,
            "account_type": "Savings",
            "account_balance": "70400.00",
            "currency": "USD"
        },
        {
            "id": 2,
            "account_type": "Checking",
            "account_balance": "80000.00",
            "currency": "USD"
        }
    ]
}
{
    "ID": 1,
    "first_name": "Sam",
    "last_name": "Brightwood",
    "accounts":[

        {
            "id": 3,
            "account_type": "Checking",
            "account_balance": "4500.00",
            "currency": "YEN"
        }
        {
            "id": 4,
            "account_type": "Investment",
            "account_balance": "4500.00",
            "currency": "YEN"
        },
        {

            "id": 5,
            "account_type": "Savings",
            "account_balance": "4500.00",
            "currency": "YEN"
        }
    ]
}

]
```

Figure 1: *here is a sample what data structure may look like in a document based databases*

They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key.

MongoDB, CouchDB , Terrastore, OrientDB, RavenDB are the well-known prominent document based databases.

Wide Column Store / Column Families

Column-family databases store data in column families as rows that have many columns associated with a row key .it may sound similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together.

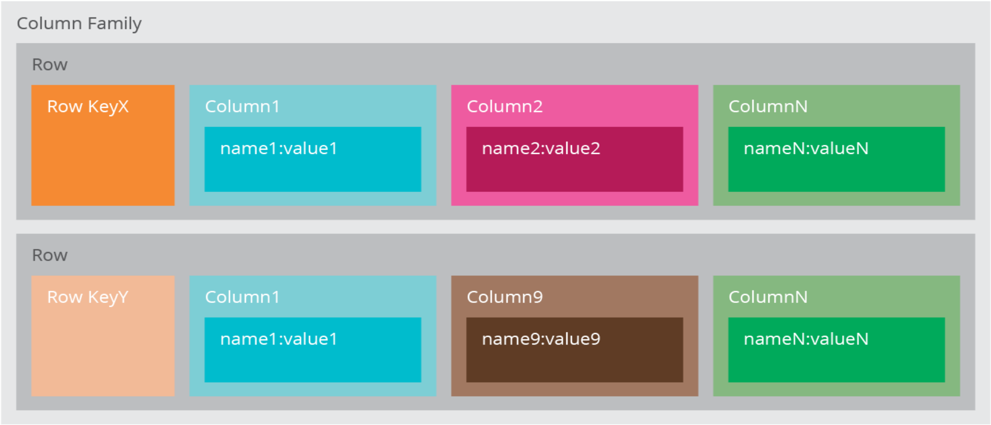


Figure 2: *Here is a sample structure what data structure may look like in a wide column databases*

Cassandra is one of the popular column-family databases; there are others, such as HBase, Hypertable, and Amazon DynamoDB.

Graph stores

A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

In Graph stores ,Relations are known as edges that can have properties. Edges have directional significance; nodes are organized by relationships which allow you to find interesting patterns between the nodes. The organization of the graph lets the data to be stored once and then interpreted in different ways based on relationships. [5]

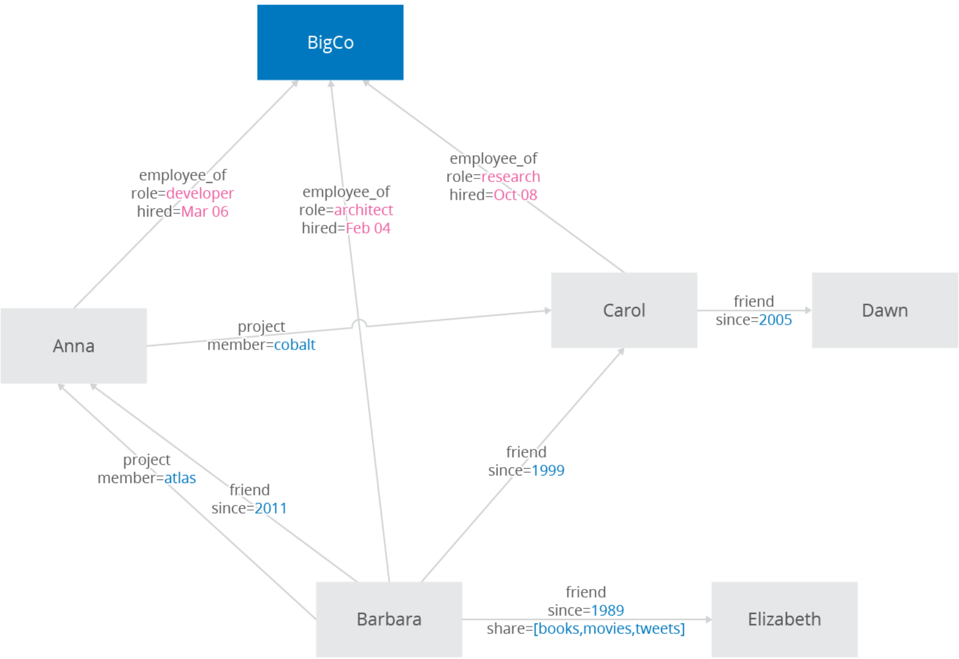


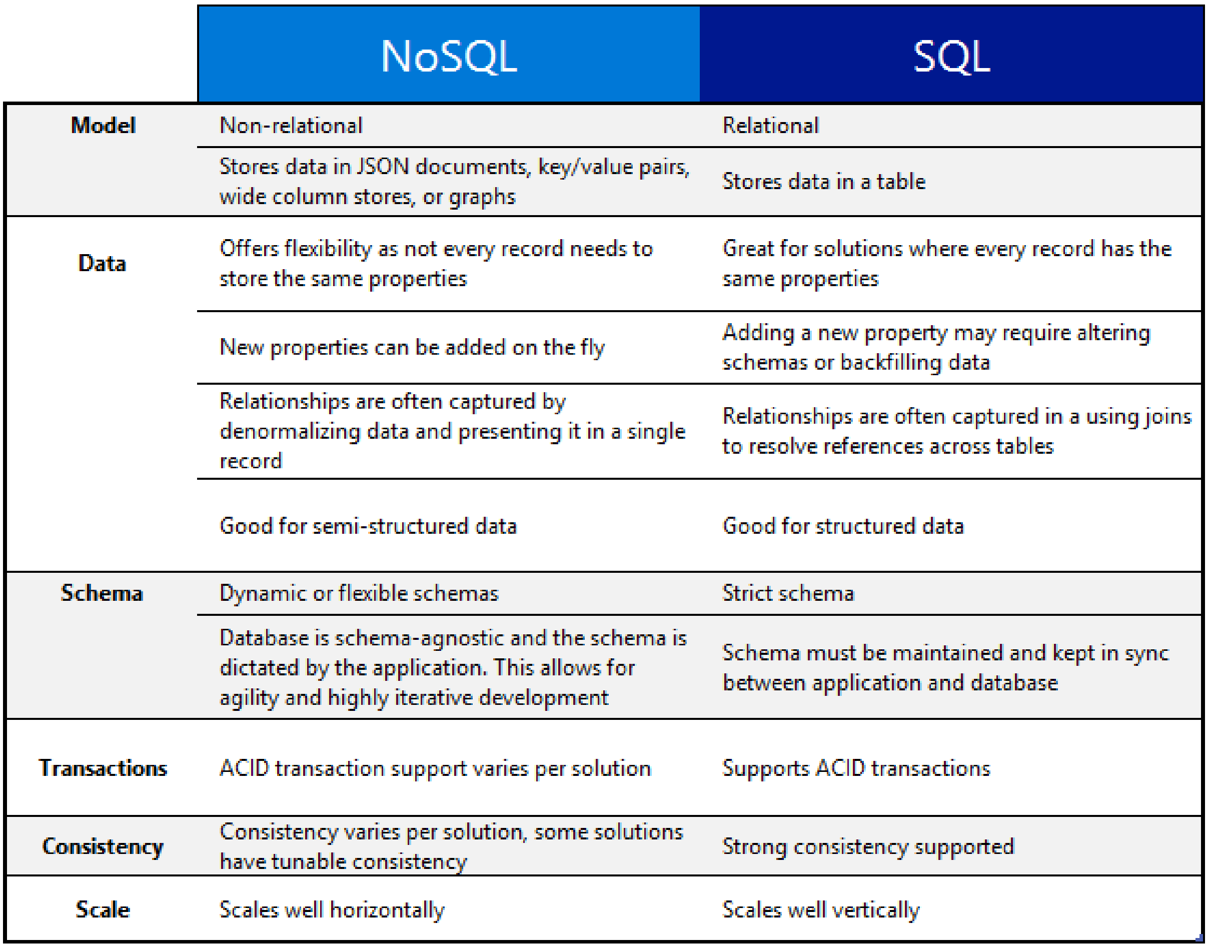
Figure 3: *Here is a sample what data structure may look like in a graph store*

There are many graph databases available, such as Neo4J, Infinite Graph, OrientDB, or FlockDB.

COMPARISON OF SQL AND NOSQL

As it is mentioned above ,The popularity of NoSql databases increases since 2009 yet it is not overthrowing relational databases.

Specially companies or governmental institutions that has strict structured data they rather prefer relational databases whereas social web sites or search engines like Facebook,yahoo,Google prefer NoSql databases.Also NoSql databases may have advantages for startup companies which have not yet strictly defined strucured data. The following table shows main differences between two databases.



	NoSQL	SQL
Model	Non-relational	Relational
	Stores data in JSON documents, key/value pairs, wide column stores, or graphs	Stores data in a table
Data	Offers flexibility as not every record needs to store the same properties	Great for solutions where every record has the same properties
	New properties can be added on the fly	Adding a new property may require altering schemas or backfilling data
	Relationships are often captured by denormalizing data and presenting it in a single record	Relationships are often captured in a using joins to resolve references across tables
	Good for semi-structured data	Good for structured data
Schema	Dynamic or flexible schemas	Strict schema
	Database is schema-agnostic and the schema is dictated by the application. This allows for agility and highly iterative development	Schema must be maintained and kept in sync between application and database
Transactions	ACID transaction support varies per solution	Supports ACID transactions
Consistency	Consistency varies per solution, some solutions have tunable consistency	Strong consistency supported
Scale	Scales well horizontally	Scales well vertically

Table 1 :Main differences between NoSql and Sql[6]

SUMMARY

Altough NoSql owes its popularity to Sqls flexibility and scalibility shortage it doesn’t mean it either totally dominates the market or it can totally outrun sql databases.Even if NoSql has many advantages over Sql databases ,each databases has their own unique structure and strong sides.On the other hand from some common aspects, we can absolutely compare Sql with NoSql databases.

REFERENCES

[1] http://docs.oracle.com/cd/E25178_01/server.1111/e25789/intro.htm

[2] http://searchsqlserver.techtarget.com/definition/database

[3]https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001

[4] http://basho.com/resources/nosql-databases/

[5] https://www.thoughtworks.com/insights/blog/nosql-databases-overview

[6] https://docs.microsoft.com/nl-nl/azure/documentdb/documentdb-nosql-vs-sql#nosql-vs-sql-comparison

Tallinn 2016