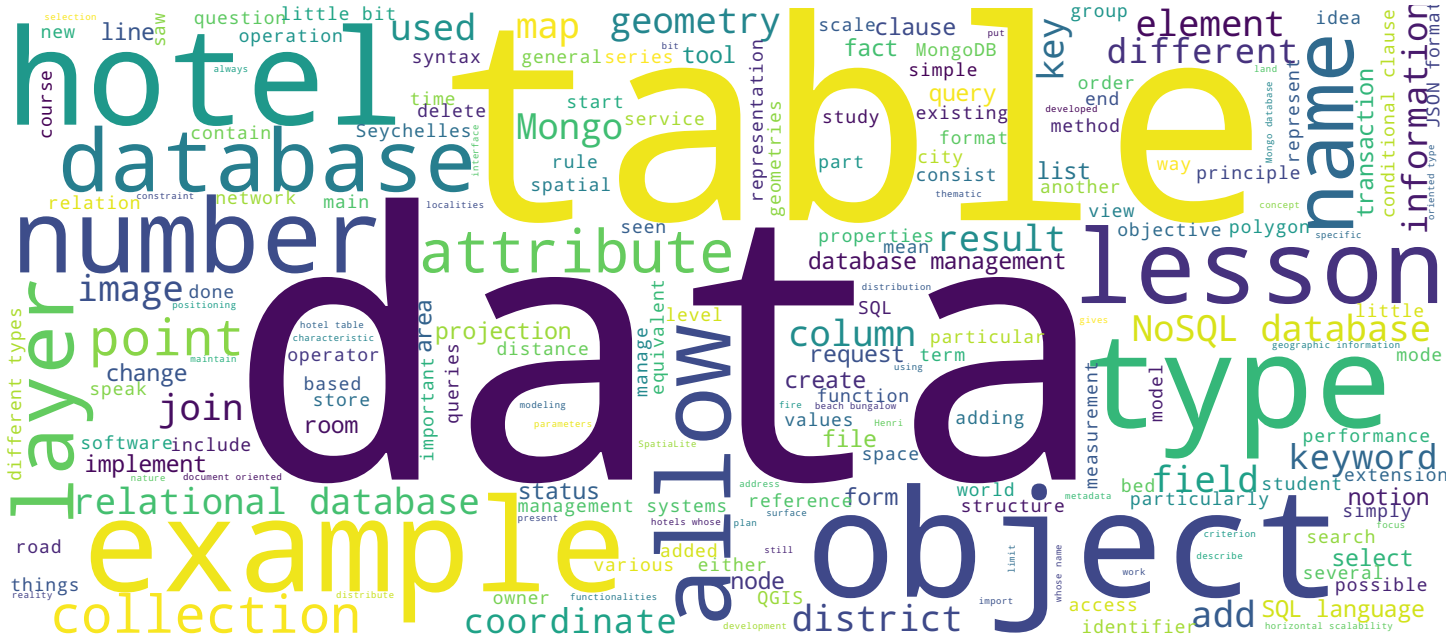
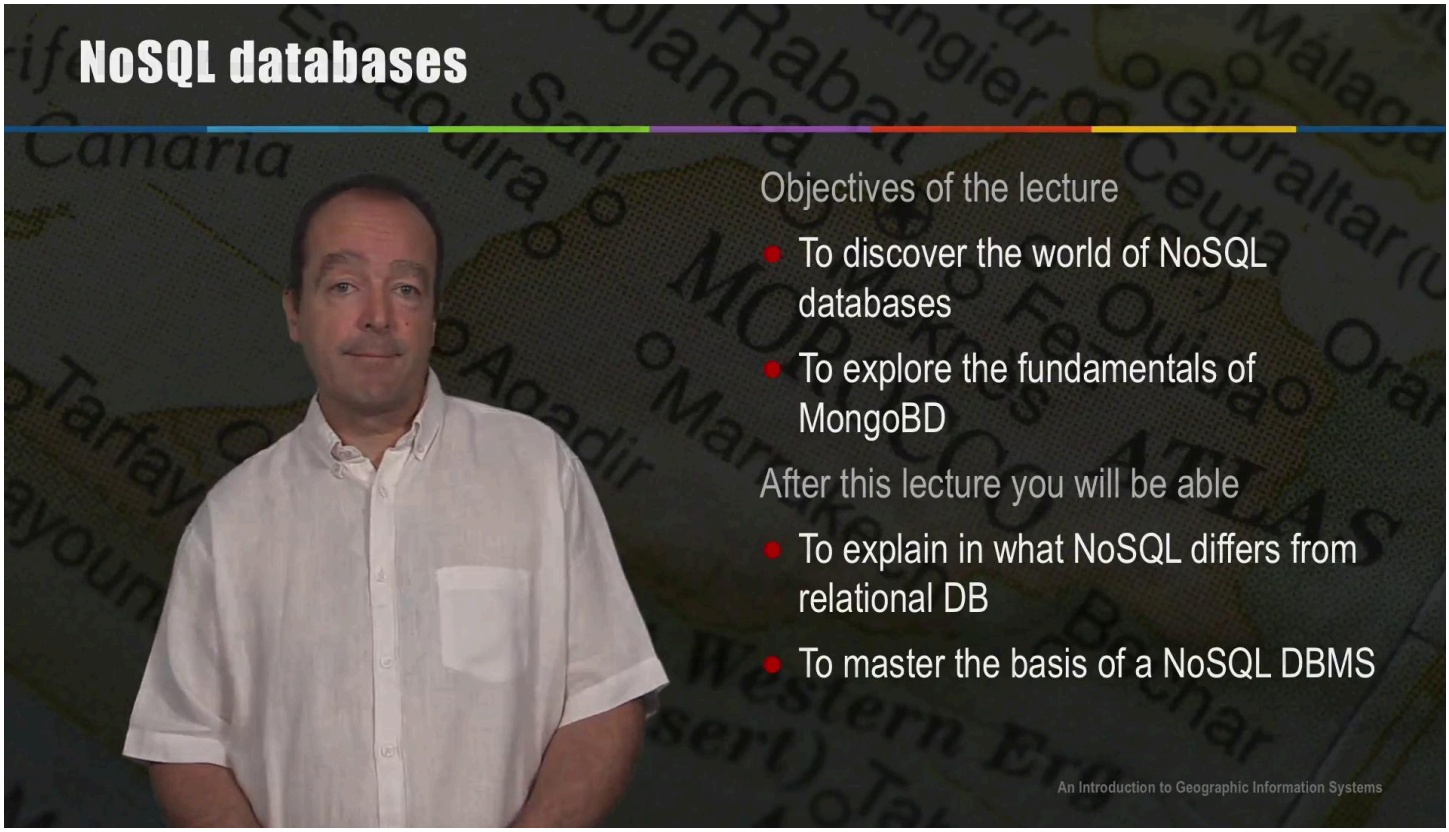
NoSQL databases

An Introduction to Geographic Information Systems

Stéphane Joost, Marc Soutter, Fernand Kouamé, Amadou Sall



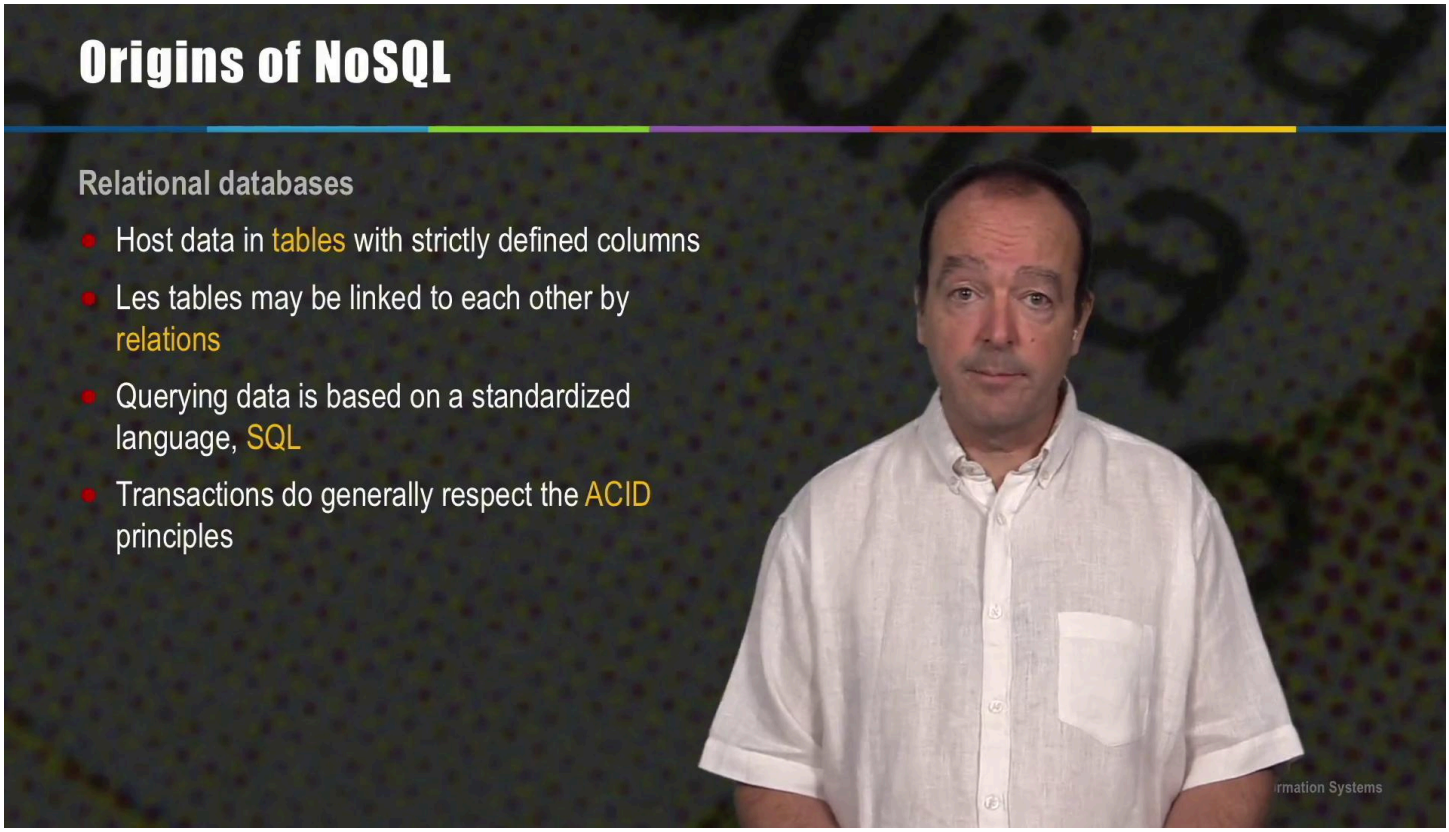


This lesson will focus on NoSQL databases which are a vast family grouping a large number of different types of database and which are distinguished from conventional relational databases that we have seen so far by the fact that they initially did not implement the SQL language. So the logic of information storage is a little different and the rules of use also. We will address all these questions during the lesson. The objective of this lesson is to complement a little bit what we have learned on relational databases by opening the field to exploring NoSQL databases and perhaps more particularly to one of them which is the Mongo database. At the end of this lesson you should be able to explain how the NoSQL world differs from the world of relational databases and you should be able to use one of these databases a little bit.

Notes

Summary

0m 22s

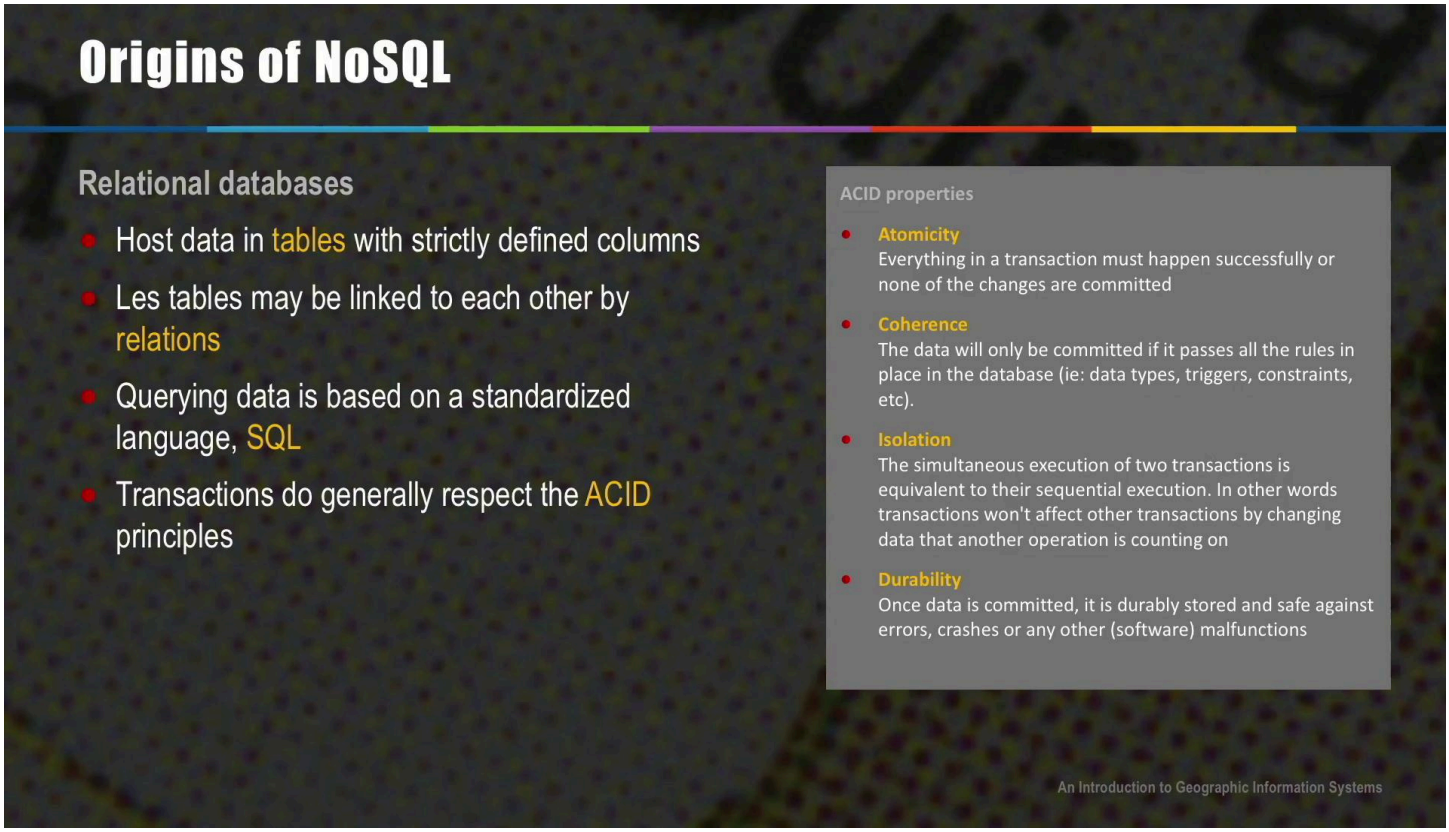


In this lesson we will first discuss the origins of NoSQL then we will review the different types of NoSQL databases and database management systems associated with them before focusing more particularly on MongoDB which is a type of NoSQL database. And finally, we will see how the CRUD functions, so Create, Read, Update, Delete, are implemented in a system like MongoDB. We saw in the lesson on relational databases that this type of database stores data in tables whose columns have well defined types, that these tables can be linked together by relations, that the querying of the data is based on a standardized language, the SQL language, and finally that the transactions carried out in these tables generally meet the principles of ACID namely atomicity, coherence, isolation and durability.

Notes

Summary

1m 27s

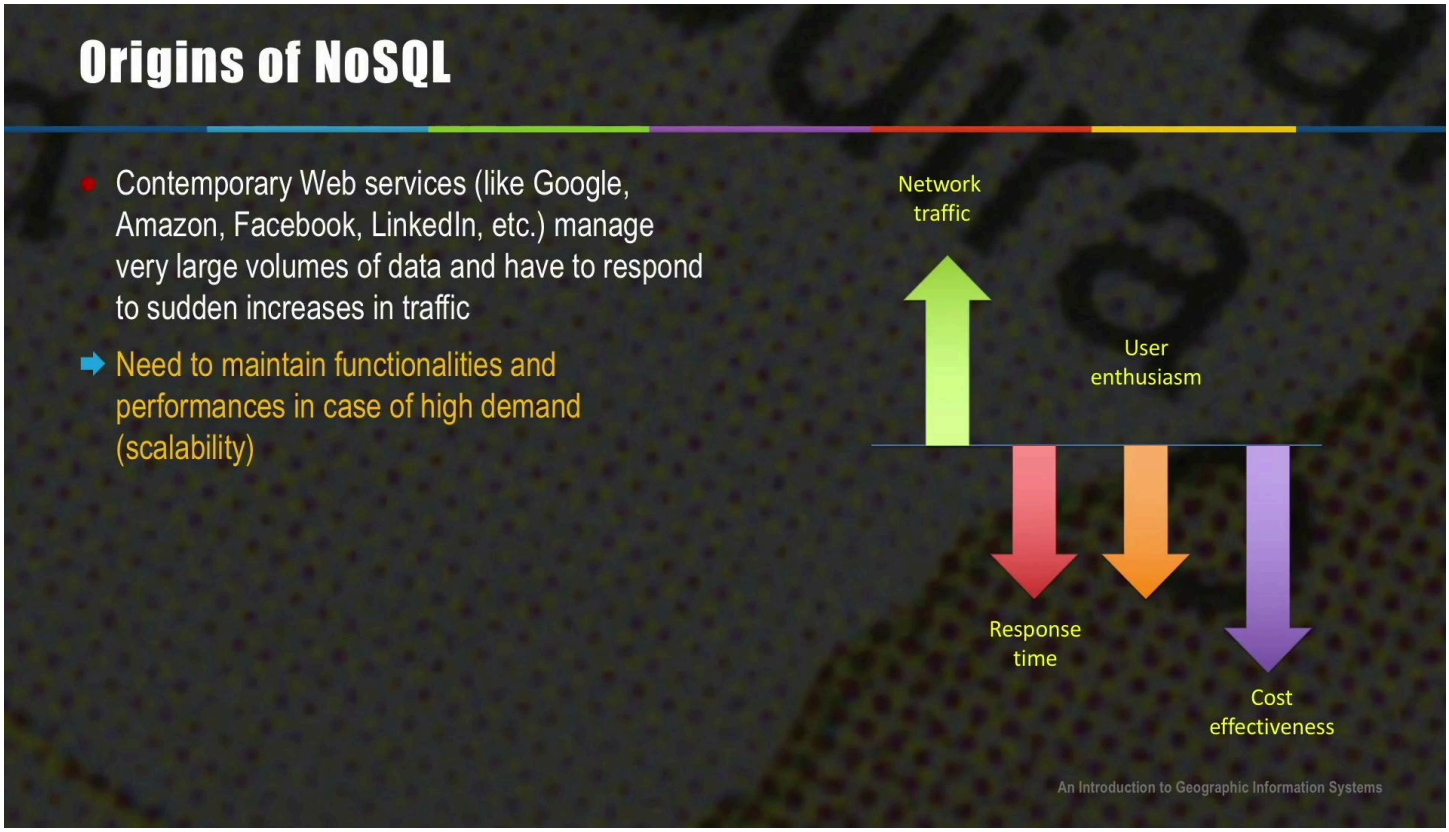


Atomicity means that all the changes of a transaction must be successfully carried out otherwise all of these changes must be invalidated. Coherence, that a transaction is validated only if all the rules in force are respected on the data type, on the constraints, the primaries, the null values, possible or not, etc. Isolation means that the simultaneous execution of two transactions is equivalent to their sequential execution so one transaction does not affect the other transaction and durability means that after validation the data are durably saved, regardless of errors, crashes or other malfunctions which may occur.

Notes

Summary

2m 41s



The second web generation saw the emerging of services that manage very large volumes of data like Google, Amazon, Facebook, LinkedIn, etc. and that have to deal with sudden influx of punctual traffic. Influx of traffic generally means a decrease in response time, a decrease in user enthusiasm and loss of profitability, hence the need to maintain the functionalities and performances in case of high demand, a characteristic called scalability or upgradability.

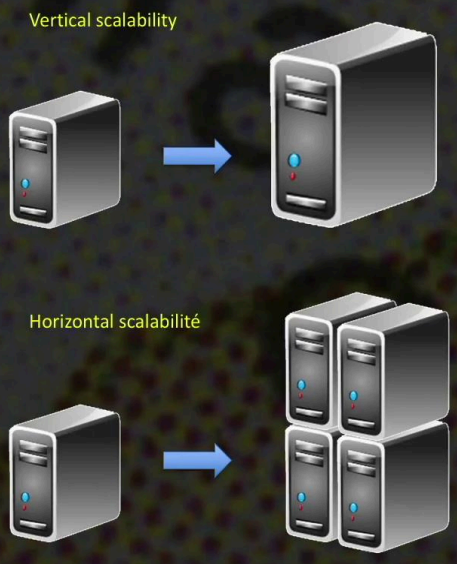
Notes

Summary

3m 24s

Origins of NoSQL

- Contemporary Web services (like Google, Amazon, Facebook, LinkedIn, etc.) manage very large volumes of data and have to respond to sudden increases in traffic
- Need to maintain functionalities and performances in case of high demand (scalability)
  - By an increase in power (vertical) … or better by a multiplication of nodes (horizontal) and distribution of services on the numerous nodes of the network
  - By an improved adjustment to the type of managed data



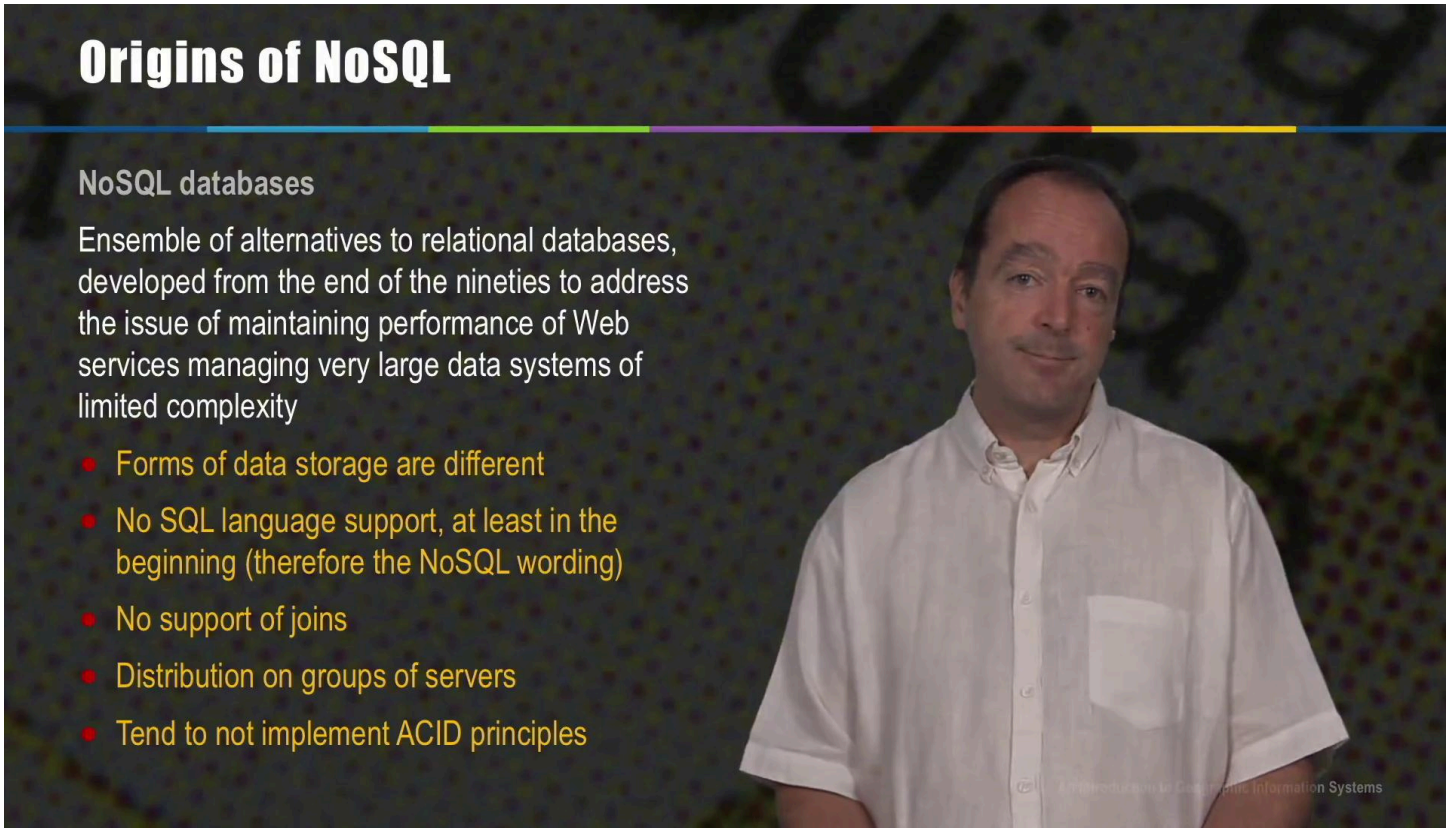
An Introduction to Geographic Information Systems

This maintenance of functionalities and performances can be obtained by increasing power and we speak then of vertical scalability or by reduction of the nodes and distribution of services on the nodes of the network and we then speak of horizontal scalability. A better adjustment of softwares to managed data types also allows to accelerate the data processing and to maintain the performance.

Notes

Summary

3m 59s

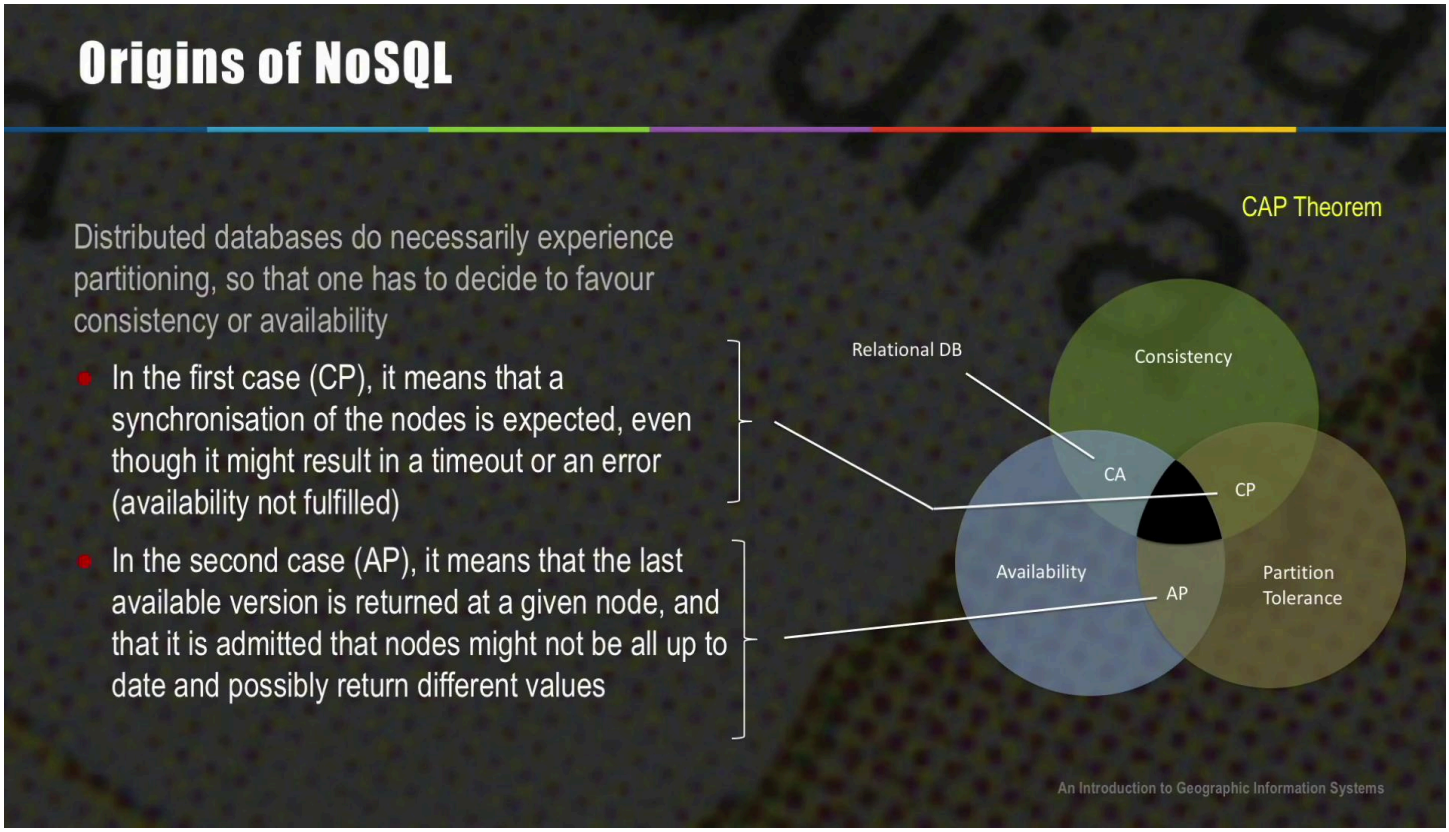


We therefore see that the NoSQL databases group together a set of alternatives to relational databases that were developed in the late 90's to meet the need to maintain the performances of some big players of the web which manage very large databases of low to moderate complexity. The main characteristics of NoSQL databases are first of all in the data storage form which differs quite a bit from what we are used to in the world of relational databases in not providing, at least at the origin, SQL language implementation, in not bringing support to the joins, so we cannot make joins in NoSQL databases, to work as a distributed system, that is to say to distribute the data on several servers and finally a tendency not to implement the ACID principles.

Notes

Summary

4m 25s

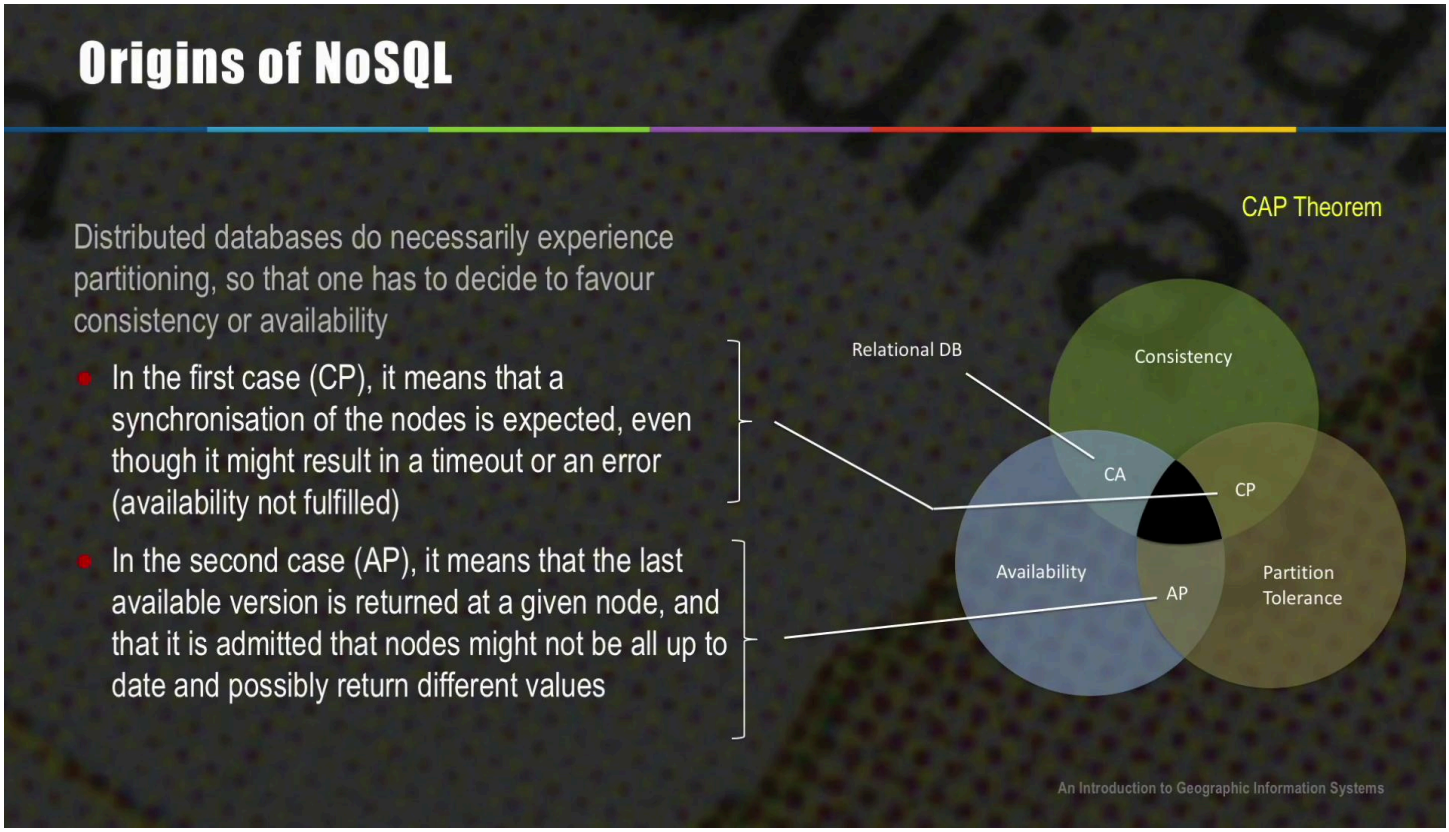


The principle of horizontal scalability, so to want to distribute the service or data on the many machines constituting a network, so on a set of nodes, leads directly to this theorem called "CAP theorem" which stipulates that only 2 of the 3 criteria of availability coherence and tolerance to partitioning can be satisfied simultaneously. Coherence means that all the nodes of a system see exactly the same data at the same time. Availability means that every time a query is sent to one of the system nodes we are sure to receive an answer either in terms of success or failure of the request but we receives a response. The notion of partitioning tolerance which expresses the idea that in a distributed system it frequently happens that the different nodes can be isolated from each other either because a node no longer works it was the victim of a crash, a power failure, communication breakdowns, no more communication, etc. In the case of NoSQL systems where we attempt to distribute the load and the information on the many nodes of a network, partitioning is a reality so that in fact we have to choose to favor the coherence or the availability.

Notes

Summary

5m 37s

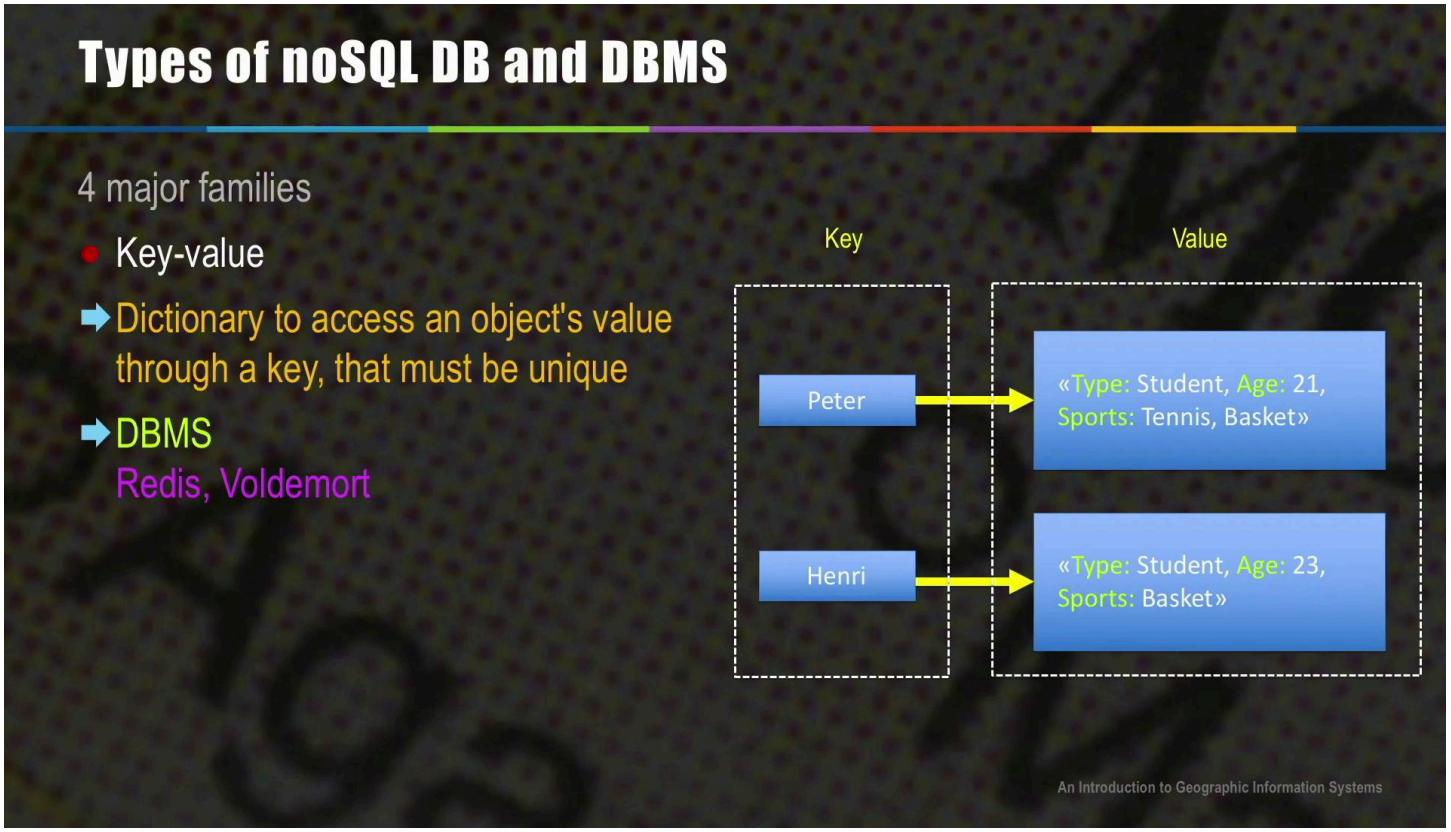


In the first case the systems called "CP, so rather coherence partitioning, this means that we expects a synchronization of the nodes before ensuring the answer with the risk of having timeout or errors from time to time, error messages, and so the availability criterion is not respected. Or we go to AP solutions where we favor the availability. In this case, the node that we question will send the latest available version of a data at the level of a node which means that we accept that all nodes may not always be up-to-date and sometimes send different values. After all the question is to know what is the tolerable update time. We see that the relational databases which by their very nature, by the existence of relations between the tables, do not lend themselves well to a distribution of data on several different machines, are usually found in the CA register, so coherence-availability, but do not lend themselves at all to a horizontal scalability distributed on a network.

Notes

Summary

7m 08s

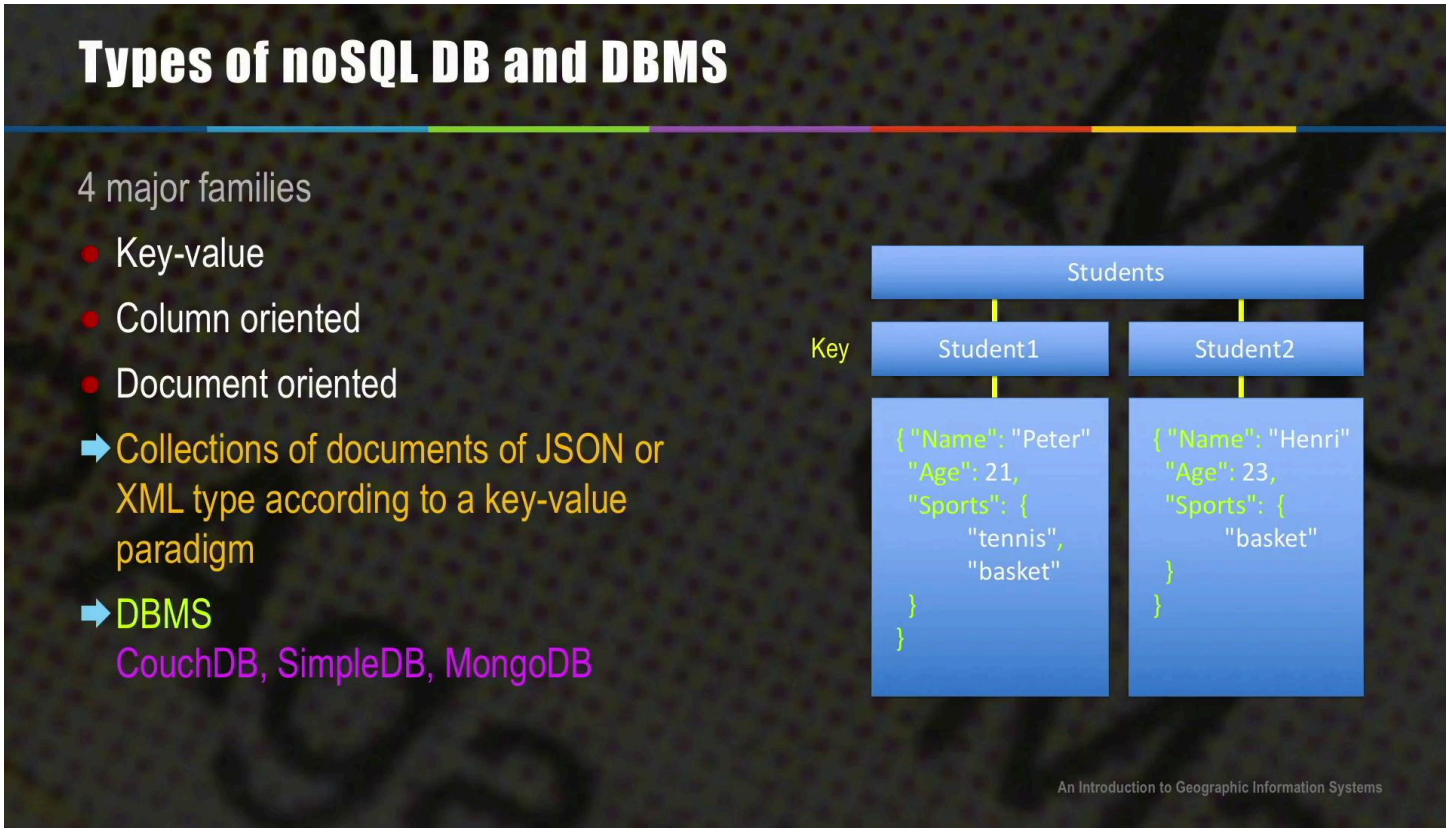


But if we look a little closer at the world of NoSQL databases we can see that we can distinguish four large families of NoSQL databases which are distinguished by the way they store information. The first of these families is constituted by the key-value type databases which are based on dictionaries allowing to access the value of an object by means of a key which must be unique. As an example, we can see here that the key can be represented by a first name, Pierre or Henri, which gives access to a series of values the type of activity, age, type of sport, etc. The main database management systems that implement this key-value type are Redis and Voldemort which was developed by LinkedIn. to manage its databases.

Notes

Summary

8m 39s

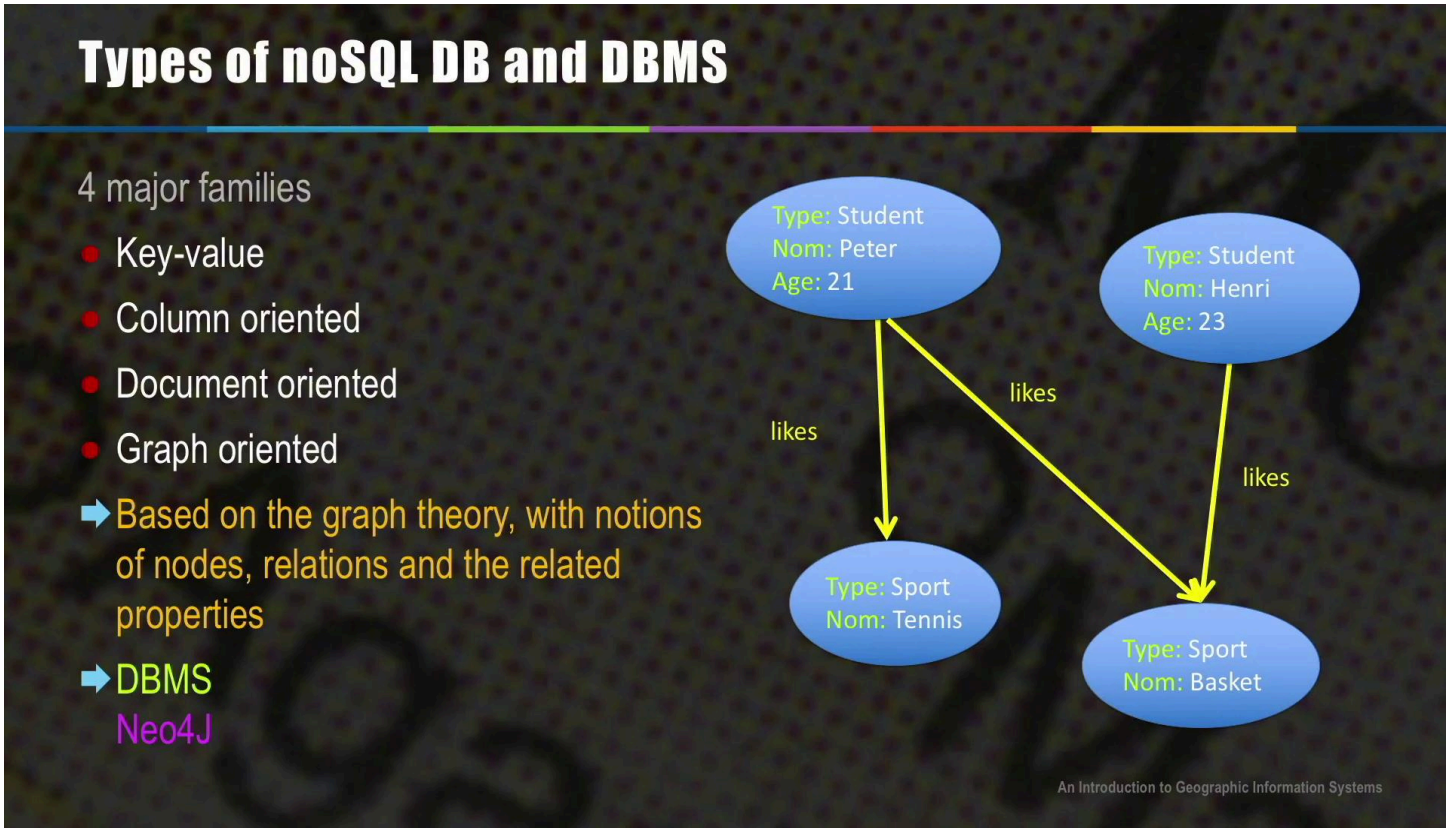


The second type of NoSQL database is the column-oriented type which is a bit like the relational model since the data can be represented by tables. Simply the storage is organized by columns on a key-value basis where for each column the key is associated with the value of this key in this column. The advantage of this system is to allow a very dynamic management of a very high number of columns. The main database management systems which implement this model are Cassandra, Hbase, BigTable and Vertica. The third type of NoSQL database called "document-oriented" also rests on a key-value paradigm by associating documents which are of type JSON or XML to the key. So here we have the same example of Pierre and Henri with student 1 and student 2 which are the keys and behind a JSON syntax which gives the attributes of the object. Database management systems that implement this model are mainly CouchDB, SimpleDB and MongoDB.

Notes

Summary

9m 30s

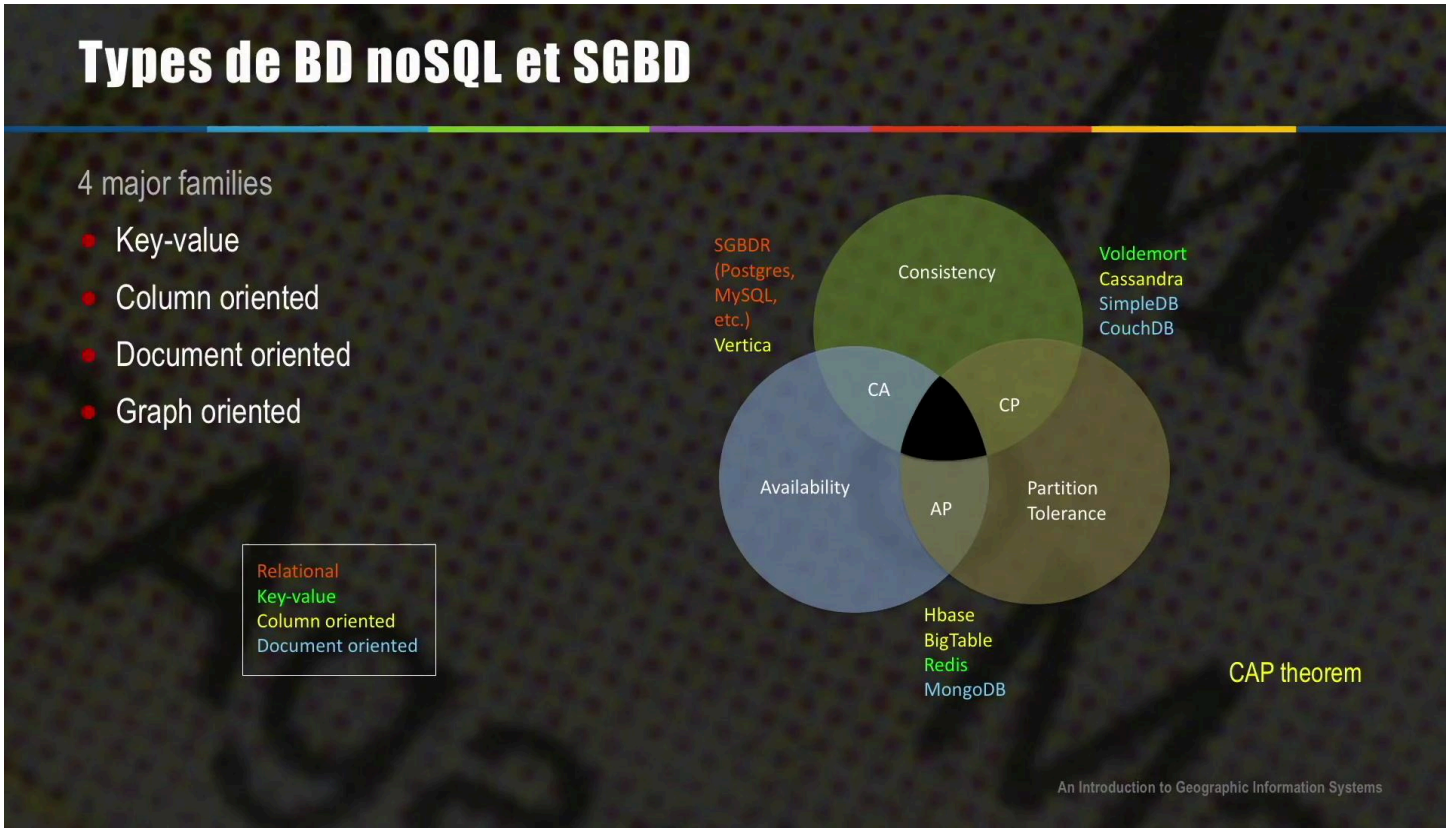


And finally, the NoSQL databases of "graph-oriented" type which are based on the graph theory, on the notions of node, relation and property to store the information, concerning here the students Pierre and Henri. The main and almost only database management system, the only system somewhat completed, is Neo4J.

Notes

Summary

10m 52s



In this figure which takes up the ideas of the CAP theorem we represented the different types of database management systems depending on their characteristics on their positioning in this diagram by representing in red-orange the relational database management systems like Postgres, MySQL, etc. The systems in green of the key-value type, like Voldemort and Redis. The column-oriented systems in yellow, with Cassandra on the CP side, HBase and BigTable on the AP side. And finally in blue the document-oriented DBMS systems where we see that MongoDB is a system that is on the side of availability while its alter egos, SimpleBD and CouchDB, are more on the side of coherence.

Notes

Summary

11m 18s

MongoBD

mongoDB.

NoSQL database
Free, document-oriented

- Tables vs collections

	nom character varying(254)	id integer	chambres double precision	lits double precision	statut character varying(254)
1	Beach Bungalows	1	4	8	Small Hotel
2	Grand' Anse Beach Villa	2	9	18	Small Hotel
3	Le Lagon Baron Hotel	4	10	20	Small Hotel
4	Les Cabanes Des Pecheurs	5	5	10	Small Hotel
5	Britannia	6	12	24	Small Hotel
6	Villa Flamboyant	7	6	12	Small Hotel
7	Black Parrot	8	12	24	Small Hotel
8	Laurier	9	6	12	Small Hotel
9	Le Duc De Praslin	10	15	30	Small Hotel
10	Cafe Des Arts	11	4	8	Small Hotel
11	Village Du Paradis	12	10	20	Small Hotel
12	La Cuvette	13	8	16	Small Hotel

MongoDB is a free NoSQL database of document-oriented type. In this database the notion of a table we had in relational databases is replaced by the notion of collection.

Notes

Summary

12m 19s

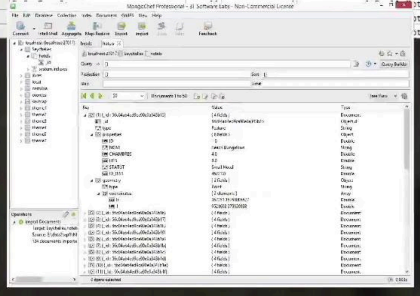
MongoBD

mongoDB.

NoSQL database
Free, document-oriented

- Tables vs collections

	nom character varying(254)	id integer	chambres double precision	lits double precision	statut character varying(254)
1	Beach Bungalows	1	4	8	Small Hotel
2	Grand' Anse Beach Villa	2	9	18	Small Hotel
3	Le Lagon Baron Hotel	4	10	20	Small Hotel
4	Les Cabanes Des Pecheurs	5	5	10	Small Hotel
5	Britannia	6	12	24	Small Hotel
6	Villa Flamboyant	7	6	12	Small Hotel
7	Black Parrot	8	12	24	Small Hotel
8	Laurier	9	6	12	Small Hotel
9	Le Duc De Praslin	10	15	30	Small Hotel
10					
11					
12					

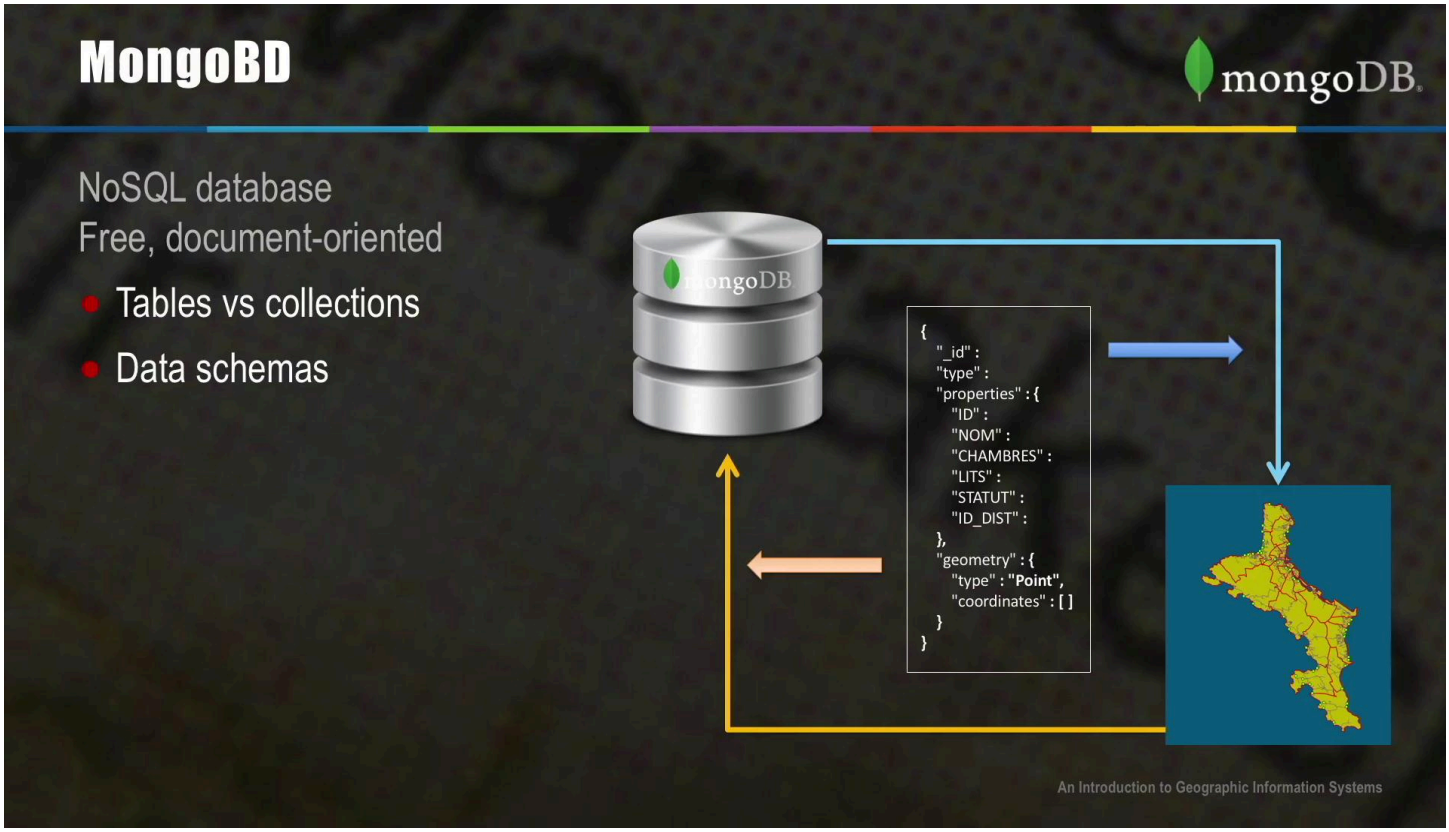


Here we have the example of the Postgres table of the Seychelles' hotels. If we look at how this data is handled in Mongo we have here a user interface which allows to see this collection of objects, so the hotels, 124 items in the collection and then for each object a tree structure identifying the object type, the collection of its properties, the geometry and in the geometry the coordinates. This tree representation can be transformed into a representation in JSON format where we find this idea of collection in the syntax specific to the JSON format. It can also be represented in a tabular view simply with successive stages that we can browse to access the coordinates or properties of objects.

Notes

Summary

12m 39s

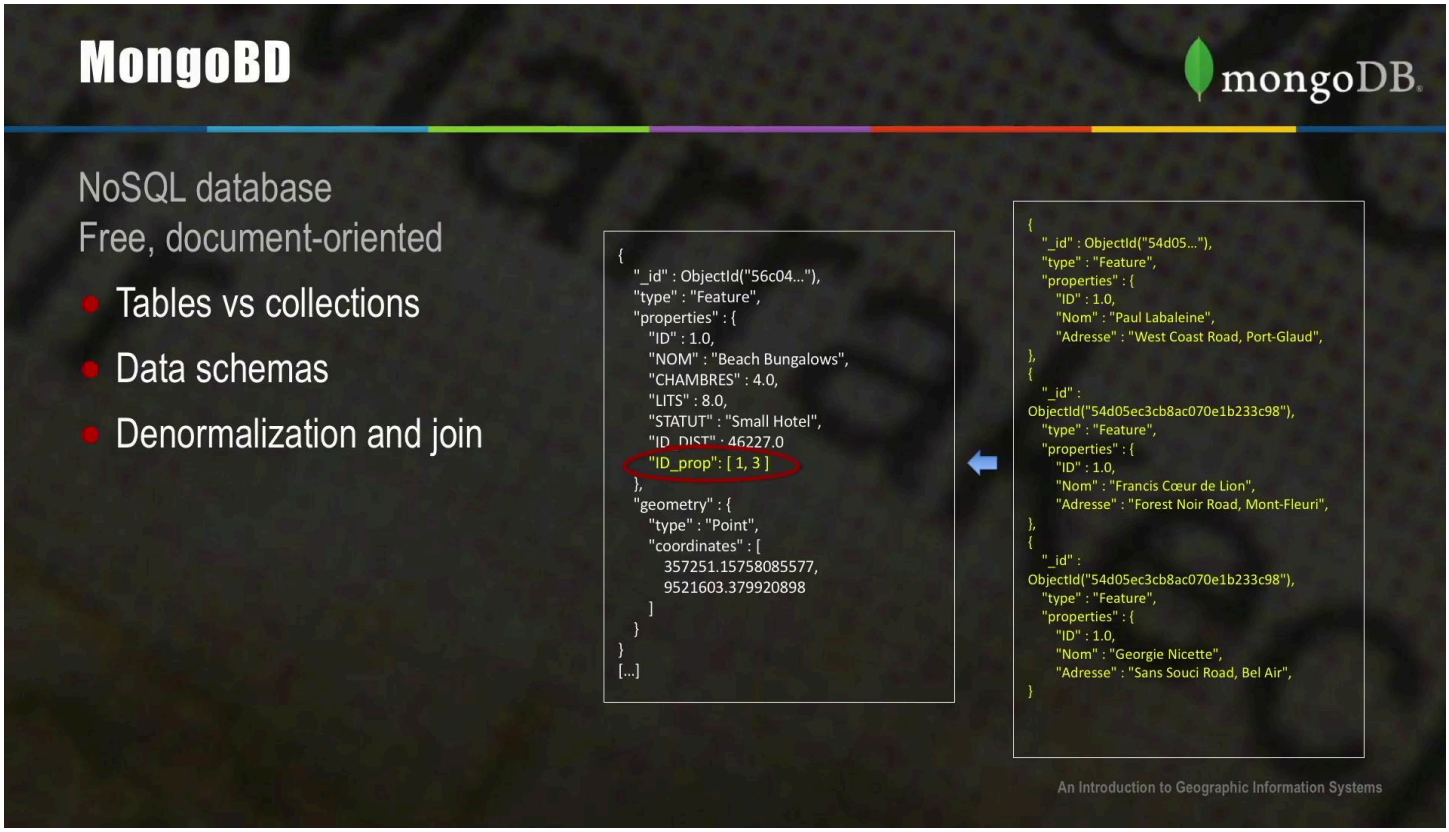


The second important element is the concept of a data diagram which, in a relational database, is very important. It is unavoidable. In a system like Mongo we are free to store the data absolutely as we want to put everything in a collection. But if we want to be able to use the data stored in a Mongo database to represent objects in a map we have to make sure these objects follow a certain structure to find the coordinates of the point and the properties of the object. The best way to find this structure is to store objects that have this structure at the beginning. In other words, the Mongo database is not restricting from the point of view of the schematization of the data but neglecting the schematization makes life much more complicated.

Notes

Summary

13m 42s

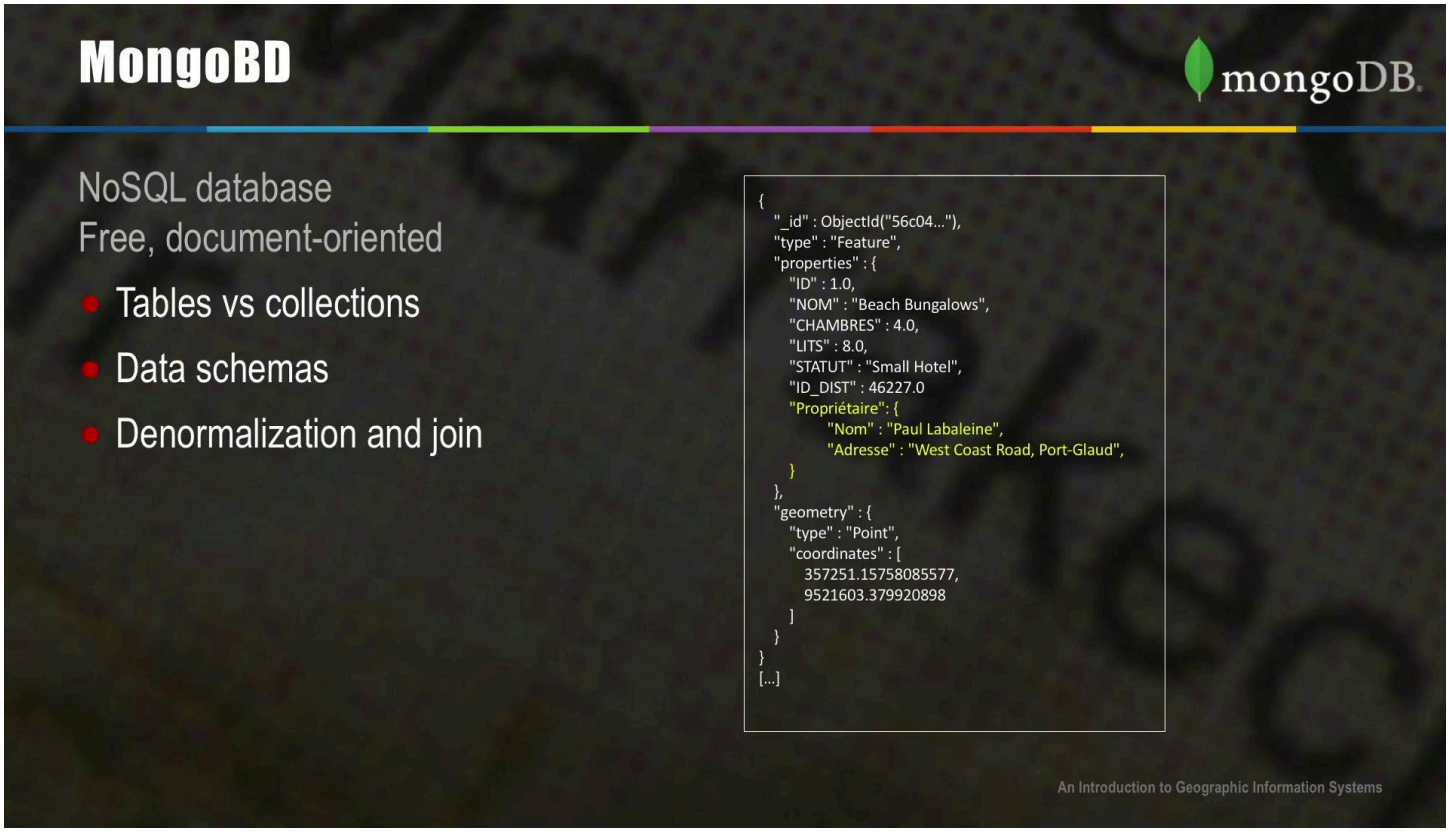


De-standardization and join. In a relational database system when we want to add a piece of information as in the case of the hotels of the Seychelles, if we want to add the information relating to the owners of these hotelsm as several hotels can have the same owner, to avoid data redundancy we would create a separate indirect table which lists the owners and we would add in the original table a column with the owner ID. A join would thus be established between these two tables. This process of data separation is what we call in the world of relational databases, the standardization. Let us note also that one of the weaknesses of the relational world, which has already been mentioned when this field was presented, the difficulty we would have here to store information in the case where a hotel has several owners. In the case of Mongo we could of course proceed in a similar manner by adding in the JSON document an owner ID field and then by having a list of owners with their addresses still in JSON format. Let's also note that the JSON syntax allows easily to represent the case of several owners.

Notes

Summary

14m 35s

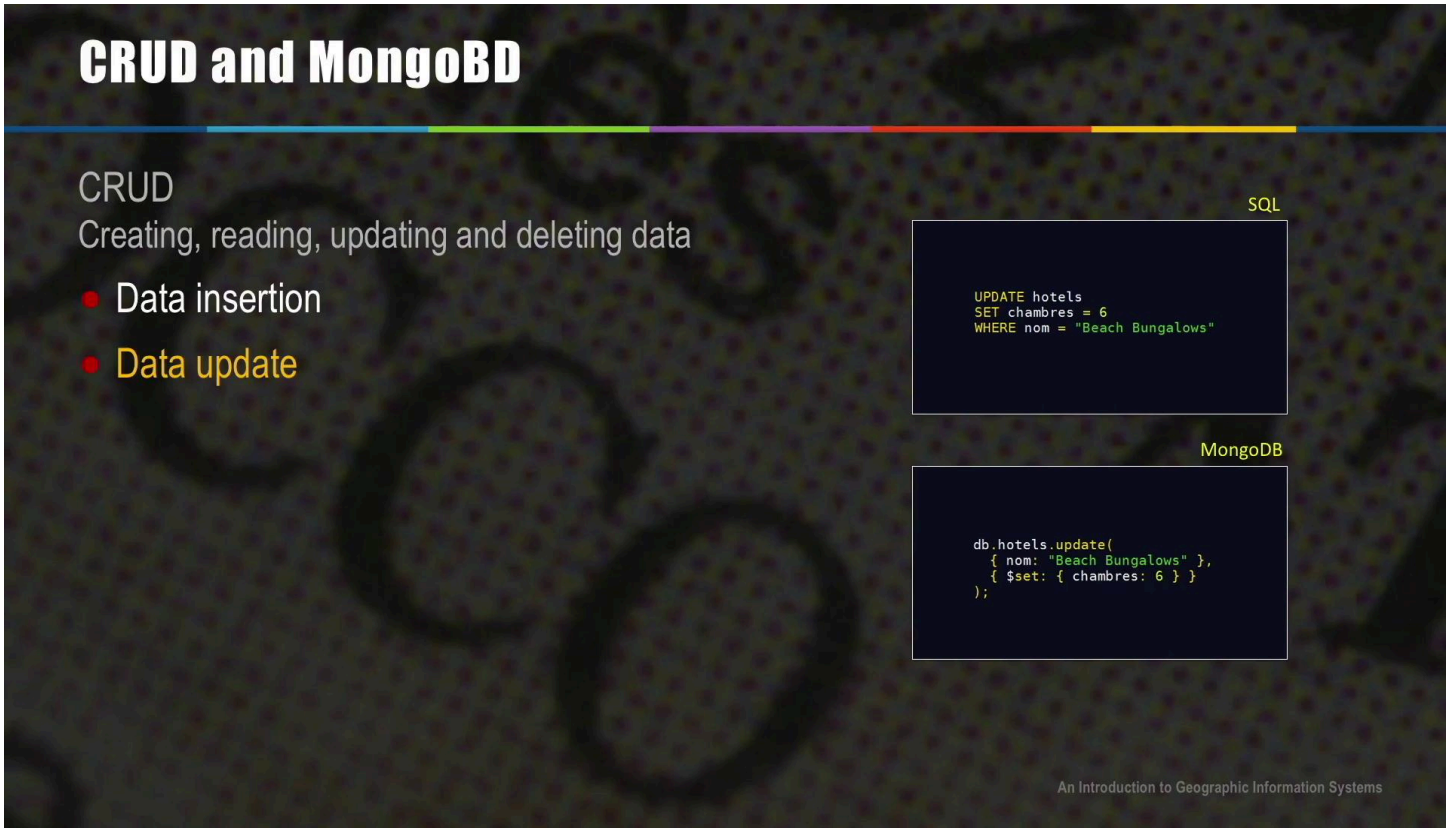


In reality, in the NoSQL world as we do not implement the joins we prefer to list explicitly the elements of information related to the owner, even if it means living with a certain redundancy. It is for this reason that NoSQL systems lend themselves particularly well to manage large volumes of data but data with a low complexity with few tabular relations. In the case of complex data systems with tables that are linked together, the relational database system remains more interesting in general.

Notes

Summary

15m 57s

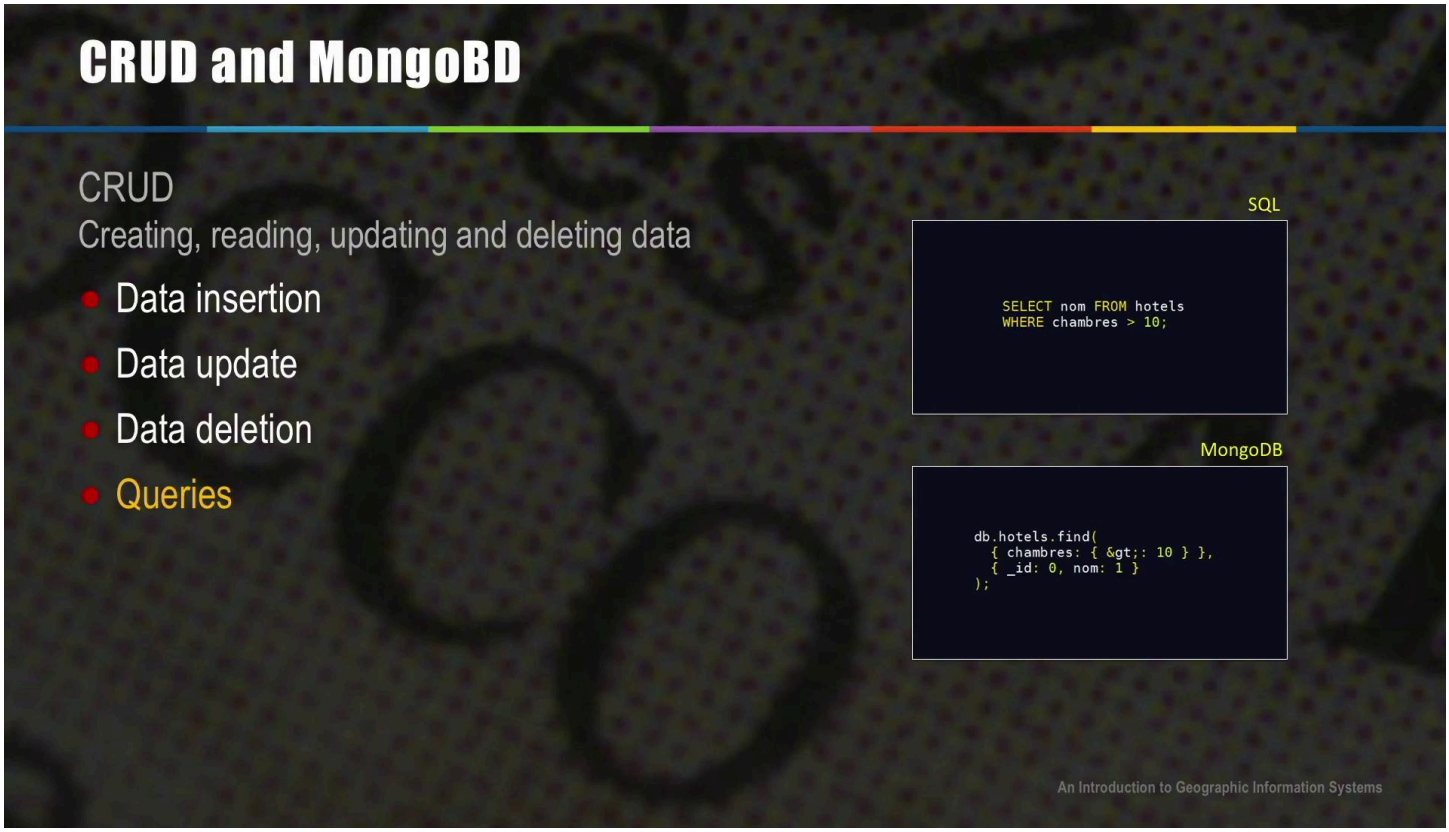


To conclude we will see some examples of implementation of basic functions of databases. So the functions of creation, reading updating and data deletion which we group under the CRUD acronym. We start with the insertion of data, we will systematically compare the SQL syntax with the syntax found in Mongo. In the SQL, the introduction of a new data set in the hotel table involves the INSERT INTO keywords, the list of columns and then the list of values corresponding to these columns. In the case of Mongo we have something a bit equivalent with just the DB keywords which refer to the current database, hotels is the targeted collection and we insert in this collection a new object which has three attributes the name, the number of rooms, the number of beds and the status with the values of these 4 attributes. Updating the data. In the case of SQL, the UPDATE keyword in the hotel table and then the SET keyword, the variable that we want to change, so the room variable that goes to 6 and the conditional clause to identify the hotel on which the number of rooms has increased to 6.

Notes

Summary

16m 43s

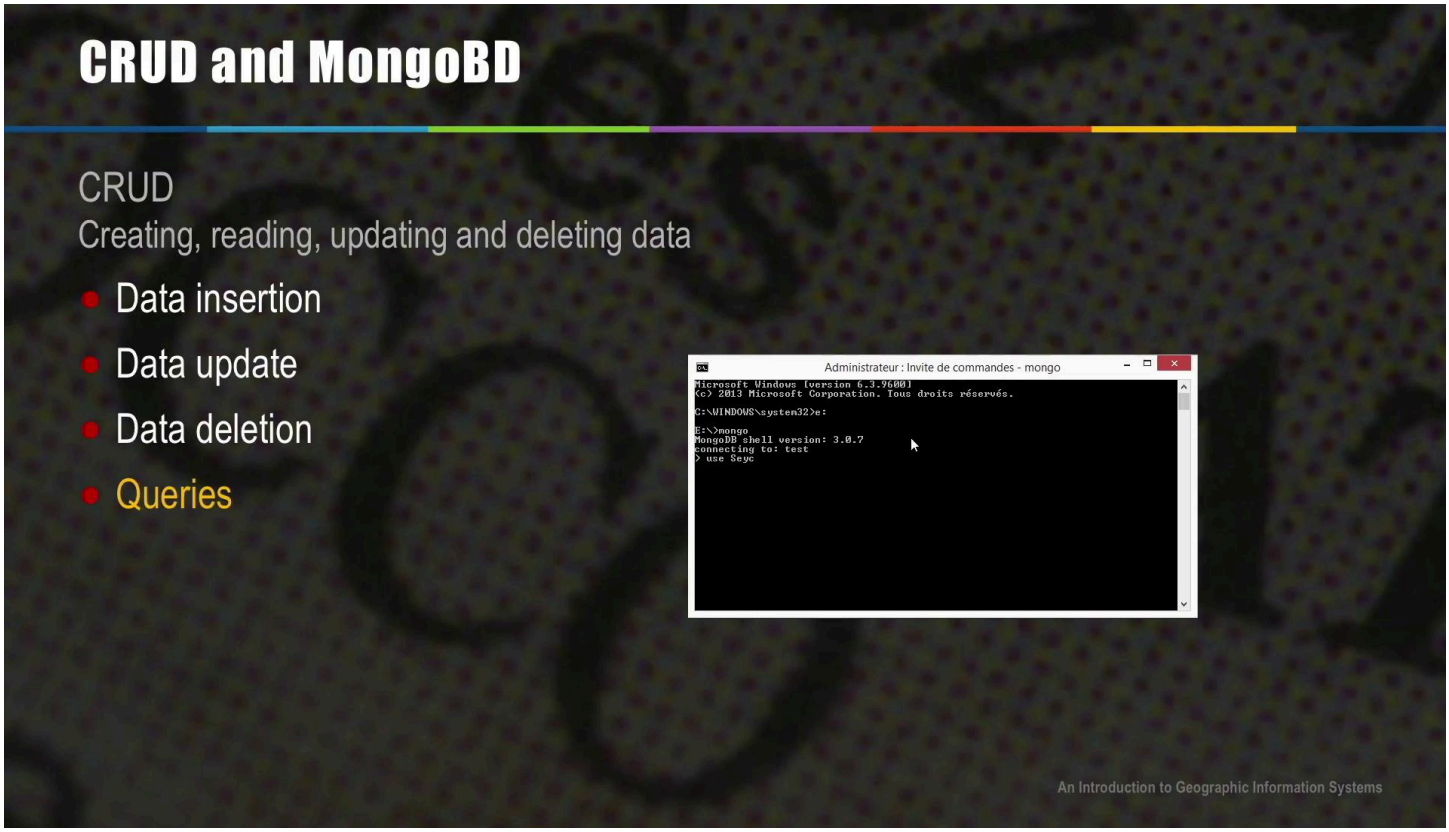


The equivalent of the Mongo side with the UPDATE keyword also which applies to the collection of hotels and then simply the reference to the object which will change so we will search for objects whose name corresponds to "beach bungalows" and we will apply a set on the room variable which takes the value 6. So something quite similar to the SQL language. For the data deletion, the DELETE FROM of the SQL language followed by the name of the table and then the conditional clause which is found almost identically in Mongo with the REMOVE function applied to the hotel collection, the document is deleted which has "beach bungalows" as attribute-name. Finally, some examples of more general requests starting with selection request of hotels whose number of rooms is more than 10. In SQL, the classic SELECT FROM with a conditional clause. In Mongo we find the FIND keyword which allows to search for objects. The search criterion in the first line, the room field and then the operator "larger than" characterized by the ampersand preceding the GT keyword and then the criterion 10 and in the second line the things we want to send like the id of the object is sent by default, we must put a zero if we don't want to have the identifier and then a 1 to send the name.

Notes

Summary

18m 09s



So it's 0, 1, true, false. A second example of a similar basic query, so counting the number of objects sent by the selection request of hotels with more than 10 rooms with something quite equivalent in Mongo, a "count" function applied to the collection of hotels an on the fact that the rooms... the room number is equal to 10. Last query example a little more elaborated where we seek to list the number of hotels there is for each status. In the SQL a classic query with a grouping by status and we count in the number of candidates in each status. In the case of MongoDB we use the keyword the aggregation function on the collection of hotels with the status as a rule of grouping and the sum as a rule of calculation. These few examples show that in a NoSQL software such as Mongo that does not implement the SQL language we still have a query language which allows to question the data in an effective way. A language that is also implemented in the graphic interface. But before moving to the graphic interface let's look at how things appear on the command line which is the initial and principal mode of use of Mongo.

Notes

Summary

19m 50s

CRUD and MongoBD

CRUD

Creating, reading, updating and deleting data

- Data insertion
- Data update
- Data deletion
- Queries

An Introduction to Geographic Information Systems

With the Mongo command we access the shell of Mongo, USE SEYCHELLES to say that we want to use the Seychelles database which will now be called DB and DB hotels, so as a reference the collection of hotels and we search for the first object of the collection which is the Beach Bungalow hotel, displayed in JSON format.

Notes

Summary

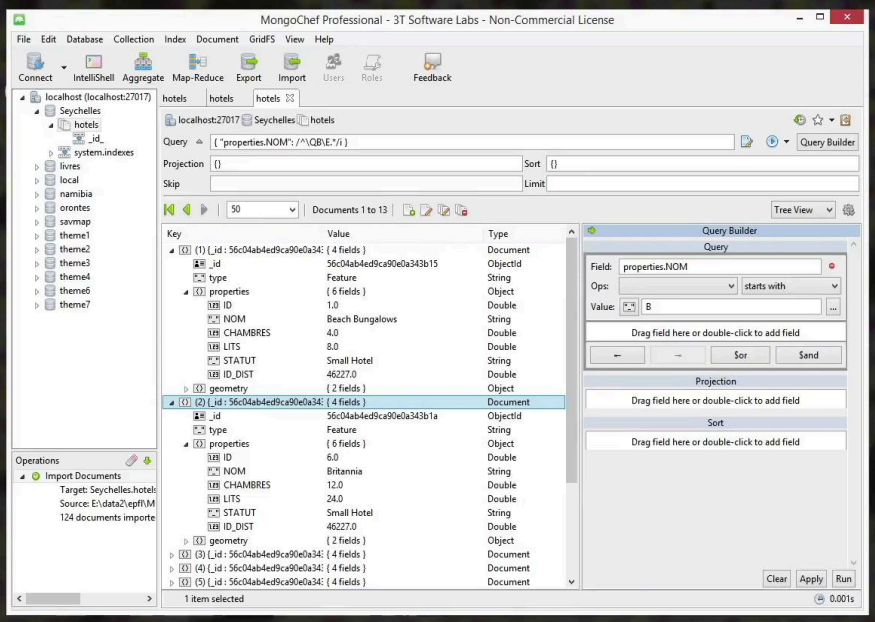
21m 28s

CRUD and MongoBD

CRUD

Creating, reading, updating and

- Data insertion
- Data update
- Data deletion
- Queries



In the case of a more elaborated graphical user interface we have a small query creation tool, we can for example drag the name field and search for all the hotels whose name begins with B and we find a whole series of them.

Notes

Summary

21m 50s