Databases

Robert Kiliszewski

November 2017

Abstract

In this review I will be talking about databases, their beginning, their development, languages, their current state and what they will become in the nearest future.

1 Introduction

[1]Databases have been around for quiet some time. Between 1910's and 1960's the way data was stored was using punch cards. The way punch cards worked was there was a large amount of cards with holes in them to represent data. In order to read the data they were put through a machine that read binary 1's and 0's, there was a small light on the machine when the card passed through, if there was a hole in the card at the reader in the machine the light shined meaning there was data in that section which was a yes. That's how people stored large amounts of data for example in a census.

This was not an efficient way of storing data since the population was growing and people needed to store large amounts of data in the most efficient way so IBM has began development of a digital database called System r in 1974.

2 System R (1970's)

[5]In 1974 IBM began their development of System R which was the first implementation of [2]SQL(Structured Query Language). Their purpose was to develop a relational database with quick transaction times and storing data digitally which would make it more convenient than having thousands or even millions of punch cards, everything would be stored digitally and could be accessed anywhere without having to transport physical items. As of their development, SQL has become the standard query language for most databases that are used today and thousands of people use it but there are also companies who developed different languages in order to expand the variety of what you can do.

3 Desktop (1980's)

[4]In the 1980's Desktop computing started emerging and so did desktop databases. As computers were moving towards more powerful machines they were able to make their databases user friendly and very easy for the user of databases to manipulate the data with dBASE. dBASE was lightweight and easy to use and understand. The creator of dBASE said: "dBASE was different from programs like BASIC, C, FORTRAN, and COBOL in that a lot of the dirty work had already been done. The data manipulation is done by dBASE instead of by the user, so the user can concentrate on what he is doing, rather than having to mess with the dirty details of opening, reading, and closing files, and managing space allocation.".

4 Object Oriented (1990's)

As object oriented programming began to see daylight in the 1990's, programmers and designers began seeing data in a database as objects rather than just some random words that are being stored. They saw peoples names, age, address and phone numbers as objects that are bound to that specific person who posses that information. With the birth of object oriented databases people were able to make connections and relations between pieces of data and were able to link 2 different databases with different data.

5 NoSQL (2000's)

[9]As data became larger and larger over the years and data became more complex with more information to store, people started to think of different ways of storing data. In the early 21'st century, NoSQL (No Structured Query Language) started becoming a language of it's own. Simply meaning No SQL meaning not using SQL, NoSQL became wildly popular with unstructured databases and real time data storage.

Big companies use many databases both SQL and NoSQL in order to store their data. NoSQL is a very good data storing system for real time data storage as well as storing large amounts of unstructured data as opposed to SQL that stores structured data in tables with rows and columns.

6 SQL vs NoSQL

Why use SQL over NoSQL and vice versa ? There are many key differences between the two and depending on what database you are going to use then the choice is up to you entirely.

Main difference between the two is that SQL is a Relational database and NoSQL is a distributed database, this means that all the data in the relational database can be connected and made as a relation whereas in the distributed database there are no relations between entities for e.g. A company will have a relational database where employees have a connection in some other table than their main information like name and age table on the other hand you can think of a relational database as users created on a website, they don't have any connection between each other.

How do you increase their capacity? In order to increase the capacity of a SQL(Vertically scalable) database you need to increase the size of the hardware such as a hard drive and for a NoSQL(Horizontally scalable) database if you want to increase its size then you need to increase the size of the server that you are storing the data on which is more costly but its capable of handling more data.

Finally, the way data is stored in the both of them is different too, in a SQL database, data is stored in tables where everything data has its own block of value and is able to be connected to another piece of data, whereas in the NoSQL database all their data is stored in document so each piece of data for example a user on a website has their own file in the server with all their details on it.

7 Creation

In order to create a small database you need a basic knowledge of the language you are going to use and the basic structure of the database. But for a big company to create a database that is going to have millions of entities, it isn't so easy.

It can be very costly to setup a database on a large scale. The company will need to train their engineers on how they want their database to be, they will need to invest in the hardware and software, they will need to plan ahead of the deployment in order for them to not run into trouble in the long run, for example running out of physical space for their hardware, in order for that to not happen they need to carefully plan out their architecture. As well as training and planning, in order to set up a big scale database you must have a team that maintains the database on a daily basis, you must have a team for emergencies such as failure or server downtime.

Adding all these costs together, it's very easy to say that creating and maintaining a database is not a cheap task to pull off.

8 Storage

[7]So how can data be stored ? There are multiple ways of storing data but generally data is sorted by name, address, id etc.

Relational Databases are the most common databases in usage. In a relational database data is stored in relations. Every data set is organized with columns where an id, name etc exists in the table of data. In a relational database, there are primary and foreign keys. A primary key is a value in a data set that identifies that column of data which has to be unique and no other data set can have the same id. On the other hand a foreign key is a field in one table that uniquely identifies a row of data in another table or the same table. In other words the foreign key of one table refers to a primary key in another table. There are also graphical databases such as Neo4j where the data stored can be displayed in a graphical manner using diagrams and bubbles as data representation.

But if we keep creating entities then what happens to the entities that are not used anymore ? Data redundancy is a huge problem when it comes to databases. Data redundancy is data that is repeated and/or not needed anymore in the database but still exists as an entity. There are many algorithms to prevent data redundancy but sometimes data slips by that algorithm and isn't noticed but is piling up and taking up space, also data that isn't needed anymore can be forgotten about and not removed when it's unnecessary causing slowness in the database and takes about large chunks of space. This matter needs to be controlled and taken care of in every day to day used database.

9 Security

[8]Are data bases secure and do they protect important data from unwanted access? There is a broad range of security protocols in order to protect data from people who shouldn't be seeing certain or all entries in a database. All of these methods that are used have to be tested and they need to be 100 percent functional before the deployment of any database.

9.1 Physical

[3]Probably the most overlooked type of security, not only does physical protection of a database protect its content, but it also protects the database from natural disasters, flooding, fire, theft, vandalism and many more. Not only does a database have to be protected from unwanted access but it also has to be protected from damage as any sort of damage to the database can be fatal and cause many problems if the data lost is of a multi-million company. It has three main components: Access control, Surveillance and Testing. The company or individual who is trying to protect their database must control who has access to a database. Many ways of protecting it is passwords, bio metrics locking mechanism, if the person trying to access a database isn't authorized then do not let them get into it.

Next is Surveillance. Important databases should be monitored 24/7 in order to prevent theft or vandalism from being done to the hardware where the database is stored, as well as a database should have a notification or alarm system for anyone who tries to breach its content.

Last but not least Testing. A database must be recoverable if anything fatal happens to it. It must be tested if it's recoverable and any defense mechanisms or any back ups must be tested if they are functioning to their maximum potential in order to be a well working database.

9.2 Technical

Technical database protection is inside the software itself. The security behind it is to prevent any unauthorized personnel from accessing the data in a database that does not know the password for it or doesn't have a registered bio metric security with the database, whether it's a fingerprint or hand scanner or even face recognition.

Data encryption can be a second layer to data protection if for any reason the first layer(Authorization) is breached. Data can be encrypted so that only staff who maintain the database are able to decrypt the data inside the database for example Credit Card details or Identification details.

10 Current State

What is the current state of our database technology? We currently have many different database providers, some of them include MongoDB, Oracle, SQL, MariaDB and many, many more. We have local databases which can be accessed only on our local machines or we can have cloud services for databases on the cloud which can be accessed from any machine provided we have access to it. We have graphical databases that display data in such way that data can be seen as nodes (As done in Neo4j) that have relations and connections to other nodes. There are endless possibilities on how we can store data and what we can store which we are yet to invent.

11 Future

[7]We are constantly moving and developing new things as each day goes by. With new technology arising each day we are open to new opportunities when it comes to the digital world, which is the same when it comes to databases. Databases are constantly being improved to have better security, to have a better and easier way of storing data as well as being the most efficient at storing data. Currently the most popular of databases are cloud stored ones. They allow users to create databases outside of their local environment and let others see it at ease but that is not cheap to do so, services such as Azure provide cloud storage for free for the first year and if you wish to use their services then you have to pay.

What databases are prone to move to in the future are time-based databases as being predicted by Neo4j. In the current state of databases, they are not capable of keeping up with live streamed data or rapid changing data for example the stock market, so programmers all over the world are trying to develop a database which will be a time-based database. This will allow data to be changed with the time and update its data at the same time as it is changed. This would be a huge step forward in data processing and storing for large companies and the stock market.

12 Conclusion

Databases play a huge part in our current lives. They have went from a small idea created by IBM to being used daily world wide by millions of people and constantly changing and improving to be more secure and so that our information isn't given to unwanted hands and companies. They have been around since punch cards and now have moved towards digital storage and we keep moving forward with our technology to make data storing more efficient and user friendly.

If not for databases we wouldn't be where we are today, we wouldn't have Facebook or YouTube as we would not be able to store huge amounts of data as we are today.

Even if you think you aren't using database systems then you are wrong, you as an internet user are contributing to the development and change of data every day even if its just doing an online survey or creating an account which in the long run is helping to make our technology move towards the better.

References

[1] Stephen Fortune "A Brief History of Databases", 27/02/2014 Avant.org

[2] Jim Melton, Alan R. Simon "Understanding the New SQL: A Complete Guide" Morgan Kaufmann Publishers, San Francisco, California.

[3] Margaret Rouse, Michael Cobb, "Physical Security", http://searchsecurity.techtarget.com/

[4] Shiann-Tsong Sheu, Dept. of Electr. Eng., Tamkang Univ., Tamsui, Taiwan, Published in: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.

[5] Donald D. Chamberlin, Morton M. Astrahan "A History and Evaluation of System R" IBM Research Laboratory San Jose, California.

[6] "Fast Subsequence Matching in Time-Series Databases" Christos Faloutsos, m.Ranganathany, Yannis Manolopoulosz, Department of Computer Science and Institute for Systems Research (ISR) University of Maryland at College Park

[7] Aileen Agricola, "Predictions for the Future of Database Technology", February 13, 2017, Extract from Graph Technology – A Data Standby By For Every Fortune 500 Company, written by Emil Eifrem and published in Computer Business Review.

[8] Teresa F. Lunt, Eduardo B. Fernandez, " Newsletter ACM SIGMOD Record - Directions for future database research and development Homepage archive Volume 19 Issue 4, Dec. 1990 Pages 90 - 97 "

[9] "NoSQL Databases" - Christof Strauch, Hochschule der Medien, Stuttgart (Stuttgart Media University)