ACID compliance and history of database

ACID and very brief History of Database - https://youtu.be/yhigMyo9kt0

From Amazon S3 - for download (same as youtube videos)

ACID and very brief History of Database

Early computers were based on coding and doing complex calculations. This lasted up until the 1960s when the first database management system was created.

Charles W. Bachman in 1960 created the Integrated Database System. This was the “first” DBMS. IBM saw this as a huge step in computers and created its own system called IMS. The IBM system would be the core of a new accounting and tracking system for the Saturn V moon missions.

Becham’s group would build their system into a business language called COBOL. In 2000 it was still estimated that COBOL represented 60% of all lines of software in commercial operation. Many banks and unitizes still rely on COBOL as a key processing system today.

Becham’s work would lead to a standardization task force, The Database Task Group and it’s CODASYL approach for searching and processing data. The system was based on a network of relationships between the data and following the network-relations to do updates. Better and easier approaches would eventually displace this Network-Database approach.

Edgar Codd from IBM research was frustrated with the way that CODASYL represented and searched data. His research evolved into a paper A Relational Model of Data for Large Shared Data Banks published in 1970. The Codd paper is one of the most important papers of all time in computer science. It details the difficulty of network database and shows a system for consistency. The system is called ACID compliance. It basically requires the ability to produce consisted behavior with multiple concurrent users of a system.

ACID stands for:

- Atomicity: A transaction must be completed in its entirety or not at all.
- Consistency: A transaction must transform a database from one consistent state to another consistent state.
- Isolation: Each transaction must occur independently of other transactions occurring at the same time.
- Durability: Once a transaction completes, it is committed, it must not be lost. It must be saved for the future.

ACID is very hard. An example is what you need to get it on a Linux or Unix system.

ACID is required for most financial transactions.

Also the Codd paper outlines a system for accessing the data. This system evolved into a standard called Structured Query Language or SQL. The good thing about SQL is it applied to lots of different systems and has a strong underpinning of set theory. This language is a “declarative” language - where you specify what you want and the computer writes the program to get that data. This is different from an imperative program where you say do A, then B, then C. Over the decades the process of “writing” the program has improved and usually today the DBMS system will do a good job at writing your program for you.

Throe out the 70’s, 80’s and early 90’s different systems used this relational model and ACID compliance - it took years for them to really work out the details to get ACID compliance.

One of the developers at IBM left the company, bought a company in California, and re-named ti to Oracle. Larry Elision then took the developers and built a SQL compliant database under the company name.

Now remember that old and out of favor system for accessing data. The Network Database where you follow the data relations manually - the one that was around in 1970. That has resurrected its head as NoSQL and Network databases. There are some uses cases that make sense - but they are limited - usually very large systems like Facebook where performance and isolation of the network search has a huge impact. Google when to it’s code base and studied all of the defects and how NoSQL was being used - the conclusion was that the lack of ACID compliance and the burden of applications implementing concurrency controls was sufficiently large and error prone that they would internally ban the use of all *ALL* NoSQL databases in favor of developing there own distributed SQL based system. The system uses the PostgreSQL language on top of a set of automatically synchronized chunks of data. Google calls this, F1: A Distributed SQL Satabase That Scales and referees to it as the world’s first world-scale database. It is running on millions of computers right now. It uses atomic clocks to get millisecond synchronization around the world.

There is an open-source version of F1 based on PostgreSQL - it is called Cockroach Database.