Source : http://avant.org/project/history-of-databases/

Avant.org / Project / A Brief History of Databases

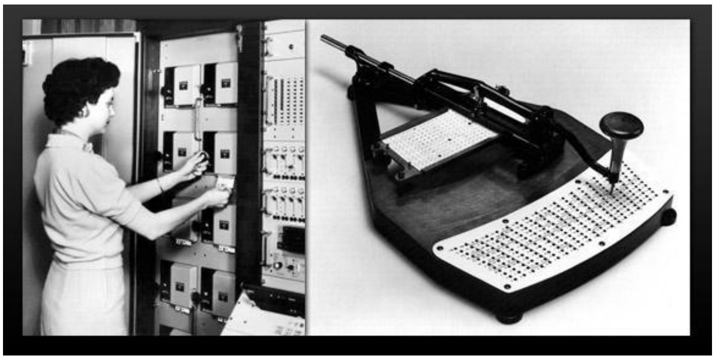
Avant.org / Project / A Brief History of Databases

27 February 2014

Databases are mundane, the epitome of the everyday in digital society. Despite the enthusiasm and curiosity that such a ubiquitous and important item merits, arguably the only people to discuss them are those with curiosity enough to thumb through the dry and technical literature that chronicles the database's ascension.[1]

Which is a shame, because the use of databases actually illuminates so much about how we come to terms with the world around us. The history of databases is a tale of experts at different times attempting to make sense of complexity. As a result, the first information explosions of the early computer era left an enduring impact on how we think about structuring information. The practices, frameworks, and uses of databases, so pioneering at the time, have since become intrinsic to how organizations manage data. If we are facing another data deluge (for there have been many), it's different in kind to the ones that preceded it. The speed of today's data production is precipitated not from a sudden appearance of entirely new technologies but because the demand and accessibility has steadily risen through the strata of society as databases become more and more ubiquitous and essential to aspects of our daily lives. And turns out we're not drowning in data; we instead appear to have made a sort of unspoken peace with it, just as the Venetians and Dutch before us. We've built edifices to house the data and, witnessing that this did little to stem the flow, have subsequently created our enterprises atop and around them. Surveying the history of databases illuminates a lot about how we come to terms with the world around us, and how organizations have come to terms with us.

Unit Records & Punch Card Databases



Punched card reader (L) and writer ® | Image from A Brief History of Communication Technology.

The history of data processing is punctuated with many high water marks of data abundance. Each successive wave has been incrementally greater in volume, but all are united by the trope that data production exceeds what tabulators (whether machine or human) can handle. The growing amount of data gathered by the 1880 US Census (which took human tabulators 8 of the 10 years before the next census to compute) saw Herman Hollerith kickstart the data processing industry. He devised "Hollerith cards" (his personal brand of punchcard) and the keypunch, sorter, and tabulator unit record machines. The latter three machines were built for the sole purpose of crunching numbers, with the data represented by holes on the punch cards. Hollerith's Tabulating Machine Company was later merged with three other companies into International Business Machines (IBM), an enterprise that casts a long shadow over this history of databases.

Punch Card Proliferation, Paper Data Reels & Data Drums

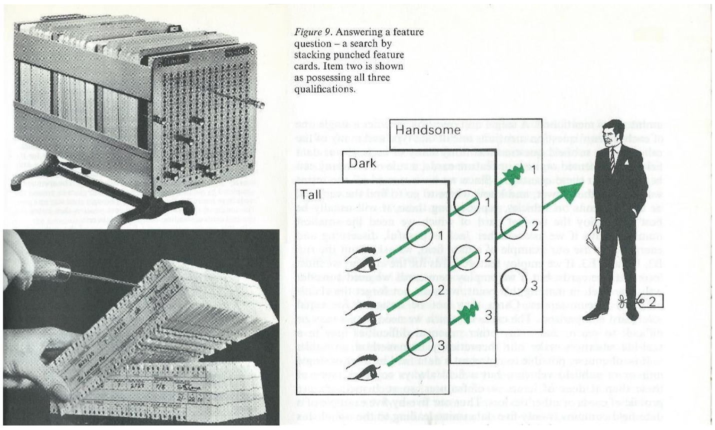


Figure 9. Answering a feature question – a search by stacking punched feature cards. Item two is shown as possessing all three qualifications.
Handsome
Dark
Tall
1
2
3
1
2
3
1
2
3
2

Among other initiatives, Thomas J. Watson Sr. insisted on well-groomed, dark-suited salesmen when leading IBM from 1914 onwards. | Image composite from J.L. Jolley, Data Study, 1968.

The revolution of data organization that punch cards instigated soon translated to domains other than governance, with companies eager to gain a competitive edge turning to this revolutionary means of restructuring their administration and services. From 1910 to the mid-1960s, punch cards and tabulating mechanisms were the prerequisite components of any office environment. All the while IBM continued to corner the market on large-scale, custom-built tabulating solutions for enterprise. Storage media diversified: In addition to punch cards, businesses began to incorporate reels of punched tape (which had long been used in textiles and player pianos) and later magnetic tape (just like audio cassette tapes, but with 1s and 0s in lieu of waveforms). These developments shared a common feature—the manner in which the data was recorded was instrumental in determining how it could then be accessed. Or in contemporary computer science parlance: Information retrieval was wholly dependent on how data is materially organized. Images (above) indicate clever mechanical means of quickly retrieving punch card information. In contrast, data tape required that one spool through to a particular location in order to retrieve a desired record.

Image courtesy Marcin Wichary.

File Systems

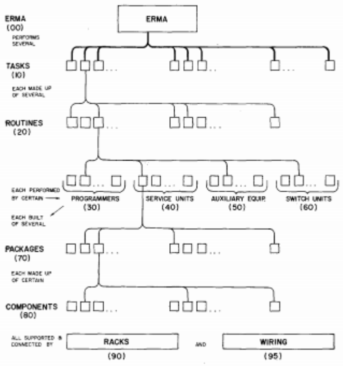


Fig. 2. The ERMA Mark I hierarchical structure

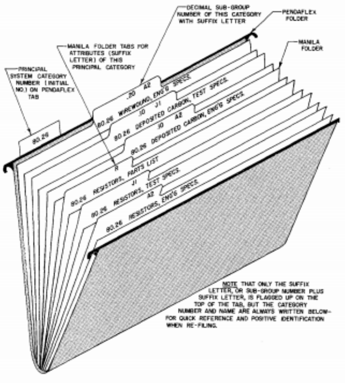
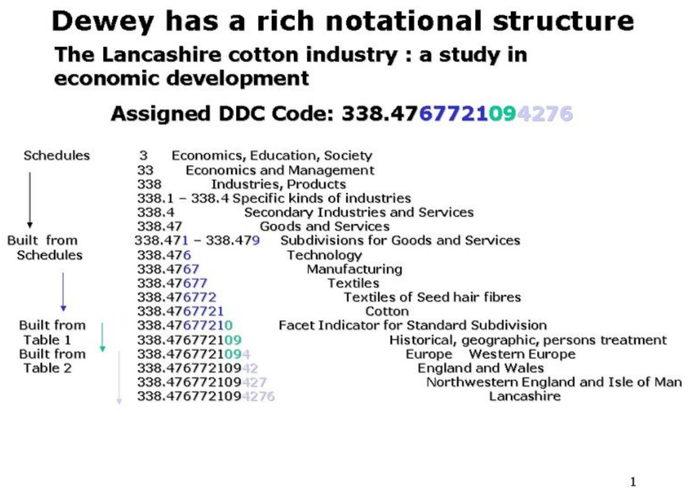

DECIMAL SUB-GROUP NUMBER OF THIS CATEGORY WITH SUFFIX LETTER
PENDAFLEX FOLDER
MANILA FOLDER
MANILA FOLDER TABS FOR ATTRIBUTES (SUFFIX LETTER) OF THIS PRINCIPAL CATEGORY
PRINCIPAL SYSTEM CATEGORY NUMBER (INITIAL NO.) ON PENDAFLEX TAB
80.26
.20 A2
80.26 WIREWOUND, ENG'G SPECS.
.10 J1
80.26 DEPOSITED CARBON, TEST SPECS.
.10 A2
80.26 DEPOSITED CARBON, ENG'G SPECS.
R
80.26 RESISTORS, PARTS LIST
J1
80.26 RESISTORS, TEST SPECS.
A2
80.26 RESISTORS, ENG'G SPECS.
NOTE THAT ONLY THE SUFFIX LETTER, OR SUB-GROUP NUMBER PLUS SUFFIX LETTER, IS FLAGGED UP ON THE TOP OF THE TAB, BUT THE CATEGORY NUMBER AND NAME ARE ALWAYS WRITTEN BELOW—FOR QUICK REFERENCE AND POSITIVE IDENTIFICATION WHEN RE-FILING.

Fig. 3. Records storage and arrangement

Image from a paper presented by G. A. Barnard, III & L. Fein at the December 1958 eastern joint computer conference.

The file system was conceived as an overarching organizational paradigm that closely resembled that of a filing cabinet. Records were treated as discrete objects which could be placed in folders (or directories). These folders could themselves be placed in other folders, creating a hierarchy that terminated in a single directory which contained all records and child folders. One such early filesystem, the Electronic Recording Machine Accounting (ERMA) Mark 1, was developed to keep track of banking records and adopted an organizational schema similar to Library Classification systems. In this way every record (or book) was categorized broadly by topic, each of which were enumerated; those topics could then be further partitioned, at which point the subdivision was indicated by appending a secondary values.



Dewey has a rich notational structure

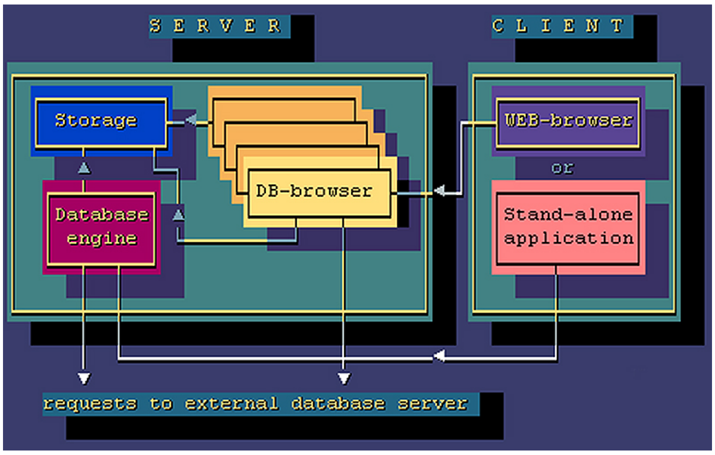
The Lancashire cotton industry : a study in economic development

Assigned DDC Code: 338.4767721094276

Schedules	3	Economics, Education, Society
	33	Economics and Management
	338	Industries, Products
	338.1 – 338.4	Specific kinds of industries
	338.4	Secondary Industries and Services
	338.47	Goods and Services
Built from Schedules	338.471 – 338.479	Subdivisions for Goods and Services
	338.476	Technology
	338.4767	Manufacturing
	338.47677	Textiles
	338.476772	Textiles of Seed hair fibres
	338.4767721	Cotton
Built from Table 1	338.47677210	Facet Indicator for Standard Subdivision
	338.476772109	Historical, geographic, persons treatment
Built from Table 2	338.4767721094	Europe Western Europe
	338.47677210942	England and Wales
	338.476772109427	Northwestern England and Isle of Man
	338.4767721094276	Lancashire

Hierarchical Dewey decimal system notation | Image from Dead Media Archive.

Database Management Systems

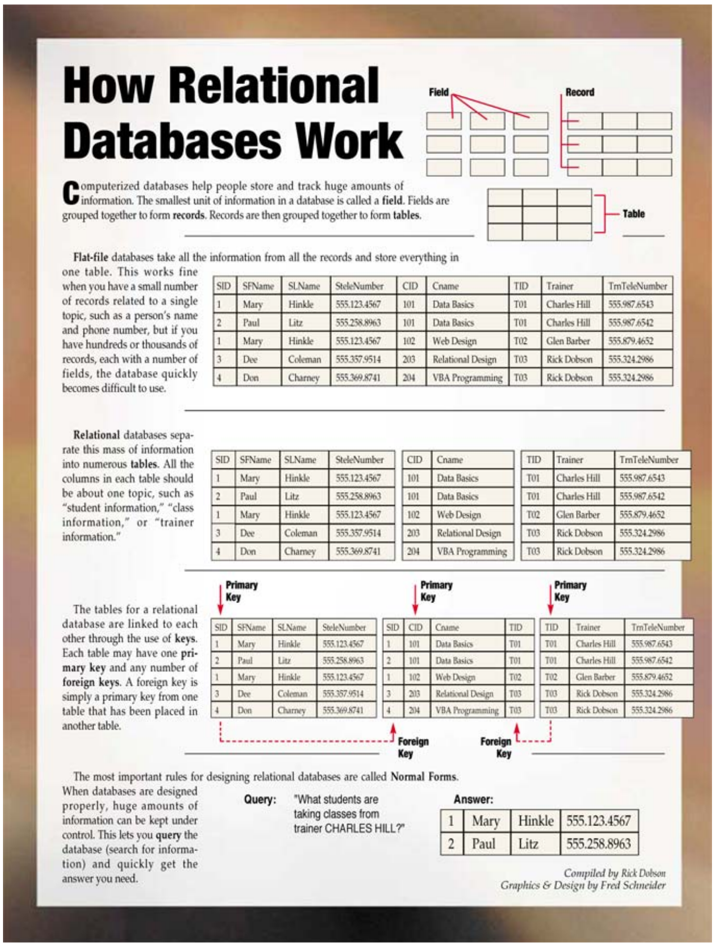


DBMS server client interaction | Image from DBMS ConteXt.

In the 1960s, as vendors began marketing computerized logistics technologies for manufacturing and wider laboratory use, we saw the advent of database management systems (DBMS). DBMSs, or the modern database, allowed users to marshal vast quantities of data. The arduous task of organizing records on the storage medium for optimal access was now handled by a subsystem called the database management system.

Two paradigms emerged, a Hierarchical model typified by IBM's Information Management System and the 'Network' model as epitomized in Charles Bachman's Integrated Data Store. Their respective uses indicate what serious business databases were becoming. The former was developed in conjunction with the Apollo program to catalog the materials needed for the Saturn V moon rocket, and the latter the Apollo Space vehicle as well as General Electric's manufacturing operations. The business impetus behind the field of data processing had begun to show in evidence: CODASYL (Conference on Data Systems Languages), the largest computing conference of the decade, rather than being composed of academic entities was composed of business enterprises like General Motors and US Steel.

Relational Databases


Field
Record
Table
Primary Key
Primary Key
Primary Key
Foreign Key
Foreign Key

How Relational Databases Work

Computerized databases help people store and track huge amounts of information. The smallest unit of information in a database is called a **field**. Fields are grouped together to form **records**. Records are then grouped together to form **tables**.

Flat-file databases take all the information from all the records and store everything in one table. This works fine when you have a small number of records related to a single topic, such as a person's name and phone number, but if you have hundreds or thousands of records, each with a number of fields, the database quickly becomes difficult to use.

SID	SFName	SLName	SteleNumber	CID	Cname	TID	Trainer	TrnTeleNumber
1	Mary	Hinkle	555.123.4567	101	Data Basics	T01	Charles Hill	555.987.6543
2	Paul	Litz	555.258.8963	101	Data Basics	T01	Charles Hill	555.987.6542
1	Mary	Hinkle	555.123.4567	102	Web Design	T02	Glen Barber	555.879.4652
3	Dee	Coleman	555.357.9514	203	Relational Design	T03	Rick Dobson	555.324.2986
4	Don	Charney	555.369.8741	204	VBA Programming	T03	Rick Dobson	555.324.2986

Relational databases separate this mass of information into numerous **tables**. All the columns in each table should be about one topic, such as "student information," "class information," or "trainer information."

SID	SFName	SLName	SteleNumber
1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963
1	Mary	Hinkle	555.123.4567
3	Dee	Coleman	555.357.9514
4	Don	Charney	555.369.8741

CID	Cname
101	Data Basics
101	Data Basics
102	Web Design
203	Relational Design
204	VBA Programming

TID	Trainer	TrnTeleNumber
T01	Charles Hill	555.987.6543
T01	Charles Hill	555.987.6542
T02	Glen Barber	555.879.4652
T03	Rick Dobson	555.324.2986
T03	Rick Dobson	555.324.2986

The tables for a relational database are linked to each other through the use of **keys**. Each table may have one **primary key** and any number of **foreign keys**. A foreign key is simply a primary key from one table that has been placed in another table.

SID	SFName	SLName	SteleNumber
1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963
1	Mary	Hinkle	555.123.4567
3	Dee	Coleman	555.357.9514
4	Don	Charney	555.369.8741

SID	CID	Cname	TID
1	101	Data Basics	T01
2	101	Data Basics	T01
1	102	Web Design	T02
3	203	Relational Design	T03
4	204	VBA Programming	T03

TID	Trainer	TrnTeleNumber
T01	Charles Hill	555.987.6543
T01	Charles Hill	555.987.6542
T02	Glen Barber	555.879.4652
T03	Rick Dobson	555.324.2986
T03	Rick Dobson	555.324.2986

The most important rules for designing relational databases are called **Normal Forms**. When databases are designed properly, huge amounts of information can be kept under control. This lets you **query** the database (search for information) and quickly get the answer you need.

Query: "What students are taking classes from trainer CHARLES HILL?"

Answer:

1	Mary	Hinkle	555.123.4567
2	Paul	Litz	555.258.8963

Compiled by Rick Dobson
Graphics & Design by Fred Schneider

How Relational Databases Work | Image from IBM100.

Edgar Codd, then working at IBM's San Jose Research Laboratory in 1973, opened his soon-to-be revolutionary relational database model with the following declaration:

"Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation)."

He was directly addressing the problems identified with the navigational paradigm: Any user had to navigate a significant amount of complexity to get at the data they were seeking. His model, articulated in “A Relational Model of Data for Large Shared Data Banks,” was a conceptual revolution: rather than conceiving of data as a simple means of organization, the database could now be used as a tool for querying data to find relations hidden within.

Relational databases separated data from applications accessing that data, enabling manipulation of information through the use of a query language, whereby selection of specific data could be performed efficiently through construction of statements containing logical operators.

IBM developed a prototype relational database model as early as 1974 called System R, which would later become the widely used Structured Query Language (SQL) database upon its release in 1981. However, Oracle (as “Relational Software, Inc.”) were first to commercialize the technology in 1979, at which point the relational database became the dominant form of bulk storage of our digital economy.

NoSQL

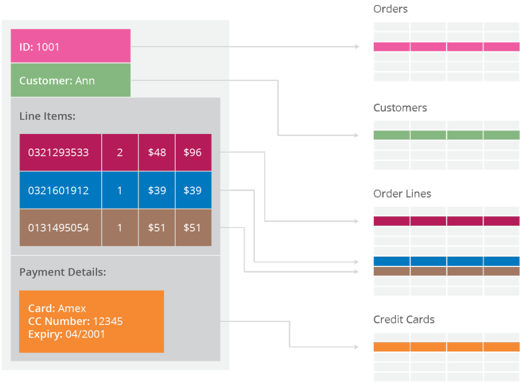


Image courtesy Thoughtworks.

The typifying feature of NoSQL databases is essentially the rejection of the ‘relational structuring of data’ inherent to RDBMS. The recent impetus behind enterprises turning to NoSQL, commonly referred to as not only SQL, has been the latest explosion in transaction volume which must be recorded as so much commerce is conducted online. This in parallel with the boon of cheap online storage has popularized NoSQL. It makes a better friend of the ad-hoc changes and dynamism demanded by a growing enterprise than the relational database does. Creating a relational database involves research and consideration of what data conceivably needs to be tracked in order to construct a relational schema. However, if you’re an agile App in startup mode, the NoSQL format allows you to voraciously hoard any and all points of data (even ones you hadn’t imagined at the outset of setting up your database)—after all, you never know when it may be useful down the line.

The jury is undecided over whether NoSQL will supplant the relational model. The skepticism surrounding its candidacy illuminates a novel moment in this data history. One question begged of Big Data has been: Is anybody actually handling data big enough to merit a change to NoSQL architectures? This may be the first point in the history of databases that a data reservoir has found the world wanting in terms of incoming volumes of data.

Cover image of IBM RAMAC 305 (1956) courtesy *IBM*.

. I of course jest – plenty of accessible histories on the matter exist for even the moderately curious.↑

Notes

Some have mentioned the absence of graph databases in this brief history. An original draft included information on graph systems but was ultimately removed for concision.

In addition, an addendum on the unconventional ZigZag database can be found on Stephen’s personal webpage.

Syllabi including this text

Data and Databases, Columbia University

Introduction to Software Engineering, Arizona State Univesity

Reading Data, Clemson Univesity

Electronic Research and the Rhetoric of Information, Washington State University

Digital Humanities Summer Workshop, Northwestern University