Physical Data Models

Physical Data Modeling

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Kevin C.C. Chang, Professor

Computer Science @ Illinois

Learning Objectives

By the end of this video, you will be able to:

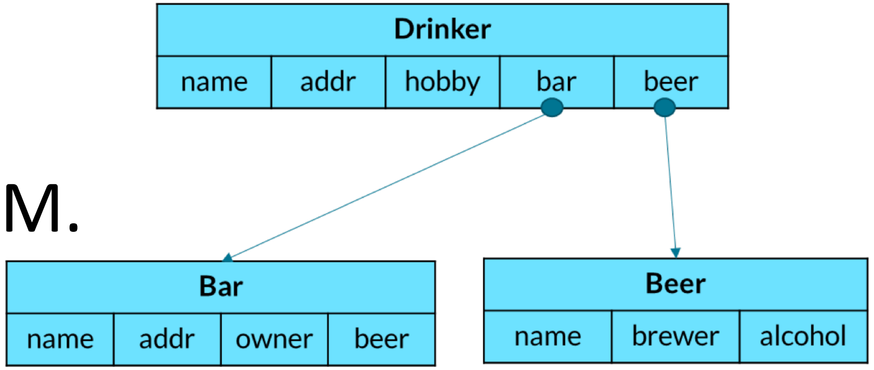
- Identify major physical data models.
- Explain the evolution of physical data models.

Physical Data Models

- Pre-relational
  - (1960s) Hierarchical Model.
  - (1960s) Network Model.
- **(1970s) Relational Model -- which is our focus!**
- Post-relational
  - (1980s) Object-Oriented.
  - (1980s) Object-Relational Model.
  - (1990s) Document Model.
  - (1990s) Key-Value Model.
  - (2000s) Graph Model.

Hierarchical Model

- First database model. Created in 1960s at IBM.
- Records connected in hierarchies.
- Data accessed by traversing the hierarchies.



Example hierarchical model

- Example: **IBM IMS**, 1968 ~ Present!
  - IBM Information Management System. Still active used!
  - *"IMS is used by many of the top Fortune 1000 companies worldwide. Collectively these companies process more than **50 billion transactions per day** in IMS."* IBM Information Management System (IMS), 2017.
  - Twitter account: @IBM_IMS, @IMS_at_50.

IMS DB and TM

@IMS_at_50

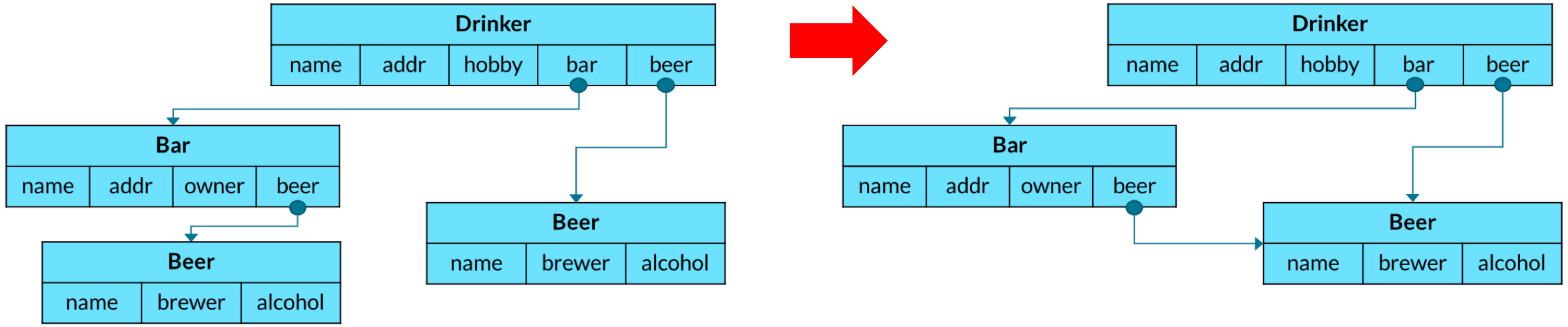
Your favorite enterprise hierarchical database and transaction manager turns 50 this year! Let the celebrations begin!

Silicon Valley, CA

IMS DB and TM Twitter account. Retrieved from https://twitter.com/IMS_at_50

Network Model

- Inspired by hierarchical model, addressing many restrictions.
- Extending tree-like hierarchies to more general networks.
- Data accessed by navigation the networks, from any node.
- Invented by Charles Backman, winning 1973 Turing Award.
- Standardized by CODASYL (Conference on Data Systems Languages Consortium) 1969.



Drinker
name addr hobby bar beer

Bar
name addr owner beer

Beer
name brewer alcohol

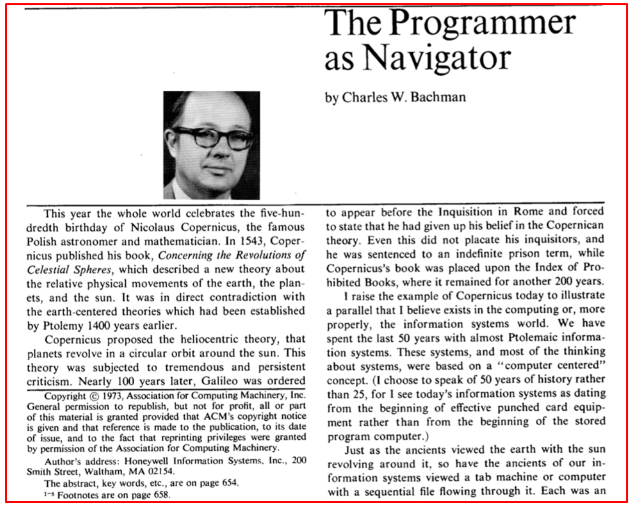
Beer
name brewer alcohol

Drinker
name addr hobby bar beer

Bar
name addr owner beer

Beer
name brewer alcohol

From hierarchical model to network model



The Programmer as Navigator

by Charles W. Bachman

This year the whole world celebrates the five-hundredth birthday of Nicolaus Copernicus, the famous Polish astronomer and mathematician. In 1543, Copernicus published his book, *Concerning the Revolutions of Celestial Spheres*, which described a new theory about the relative physical movements of the earth, the planets, and the sun. It was in direct contradiction with the earth-centered theories which had been established by Ptolemy 1400 years earlier.

Copernicus proposed the heliocentric theory, that planets revolve in a circular orbit around the sun. This theory was subjected to tremendous and persistent criticism. Nearly 100 years later, Galileo was ordered to appear before the Inquisition in Rome and forced to state that he had given up his belief in the Copernican theory. Even this did not placate his inquisitors, and he was sentenced to an indefinite prison term, while Copernicus's book was placed upon the Index of Prohibited Books, where it remained for another 200 years.

I raise the example of Copernicus today to illustrate a parallel that I believe exists in the computing or, more properly, the information systems world. We have spent the last 50 years with almost Ptolemaic information systems. These systems, and most of the thinking about systems, were based on a "computer centered" concept. (I choose to speak of 50 years of history rather than 25, for I see today's information systems as dating from the beginning of effective punched card equipment rather than from the beginning of the stored program computer.)

Just as the ancients viewed the earth with the sun revolving around it, so have the ancients of our information systems viewed a tab machine or computer with a sequential file flowing through it. Each was an

Author's address: Honeywell Information Systems, Inc., 200 Smith Street, Waltham, MA 02154.

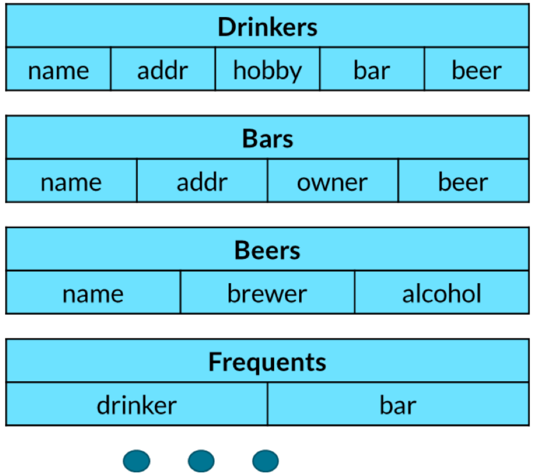
The abstract, key words, etc., are on page 654.

[1-5] Footnotes are on page 658.

The programmer as navigator (Bachman, 1973)

Relational Model

- By Edgar Codd at IBM Labs, 1969, winning 1981 Turing Award.
- Motivation: Network model encodes access paths- fragile, inefficient.
- Records stored in relations, without fixed inter-connections.
- Data accessed by computing over relations.
- Most popular: E.g., MySQL, PostgreSQL.



Drinkers				
name	addr	hobby	bar	beer

Bars			
name	addr	owner	beer

Beers		
name	brewer	alcohol

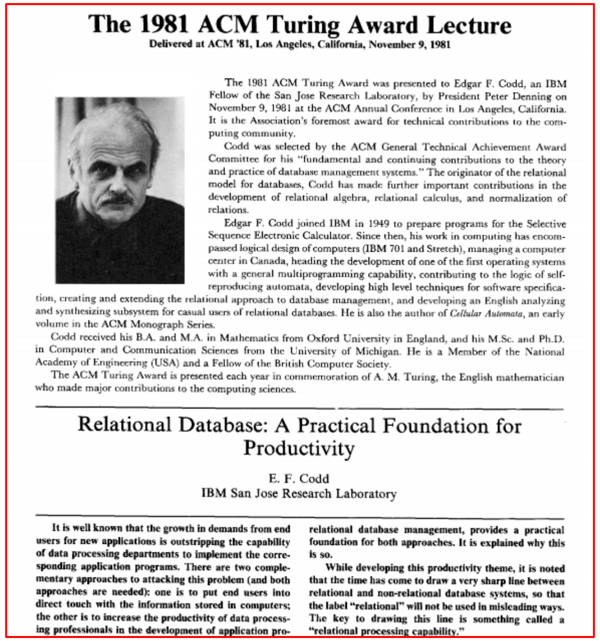
Frequents	
drinker	bar

● ● ●

Example relational model

- σ (Drinkers)
- Driners $\bowtie$ Bars

Example computation on relations



The 1981 ACM Turing Award Lecture

Delivered at ACM '81, Los Angeles, California, November 9, 1981

The 1981 ACM Turing Award was presented to Edgar F. Codd, an IBM Fellow of the San Jose Research Laboratory, by President Peter Denning on November 9, 1981 at the ACM Annual Conference in Los Angeles, California. It is the Association's foremost award for technical contributions to the computing community.

Codd was selected by the ACM General Technical Achievement Award Committee for his "fundamental and continuing contributions to the theory and practice of database management systems." The originator of the relational model for databases, Codd has made further important contributions in the development of relational algebra, relational calculus, and normalization of relations.

Edgar F. Codd joined IBM in 1949 to prepare programs for the Selective Sequence Electronic Calculator. Since then, his work in computing has encompassed logical design of computers (IBM 701 and Stretch), managing a computer center in Canada, heading the development of one of the first operating systems with a general multiprogramming capability, contributing to the logic of self-reproducing automata, developing high level techniques for software specification, creating and extending the relational approach to database management, and developing an English analyzing and synthesizing subsystem for casual users of relational databases. He is also the author of *Cellular Automata*, an early volume in the ACM Monograph Series.

Codd received his B.A. and M.A. in Mathematics from Oxford University in England, and his M.Sc. and Ph.D. in Computer and Communication Sciences from the University of Michigan. He is a Member of the National Academy of Engineering (USA) and a Fellow of the British Computer Society.

The ACM Turing Award is presented each year in commemoration of A. M. Turing, the English mathematician who made major contributions to the computing sciences.

Relational Database: A Practical Foundation for Productivity

E. F. Codd
IBM San Jose Research Laboratory

It is well known that the growth in demands from end users for new applications is outstripping the capability of data processing departments to implement the corresponding application programs. There are two complementary approaches to attacking this problem (and both approaches are needed): one is to put end users into direct touch with the information stored in computers; the other is to increase the productivity of data processing professionals in the development of application pro-

relational database management, provides a practical foundation for both approaches. It is explained why this is so.

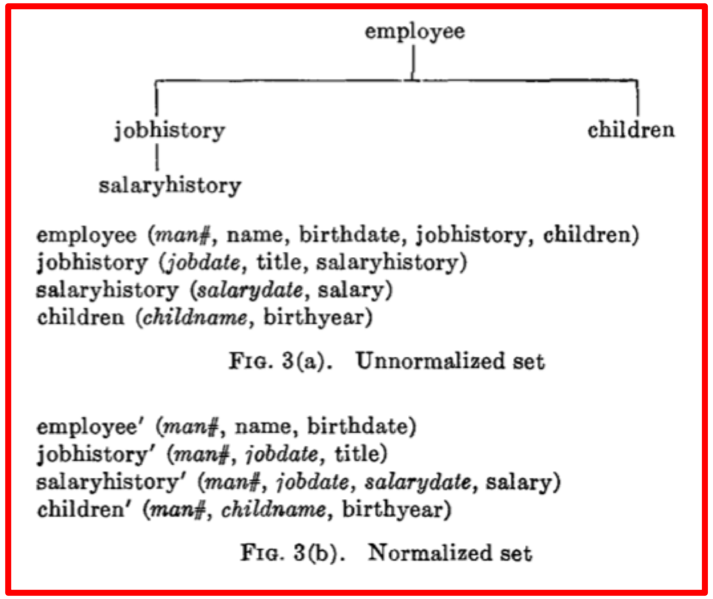
While developing this productivity theme, it is noted that the time has come to draw a very sharp line between relational and non-relational database systems, so that the label "relational" will not be used in misleading ways. The key to drawing this line is something called a "relational processing capability."

Codd's 1982 ACM Turing Award Lecture (Codd, 1982)

Post-Relational Models: Two Driving Forces

- **Meeting programming paradigms**
  - Driven by the ”impedance mismatch” with object-oriented programming.
  - (1980s) Object-Oriented.
  - (1980s) Object-Relational Model.

- **Dealing with data in various new settings**
  - Driven by applications beyond enterprise data management.
  - (1990s) Document Model.
  - (1990s) Key-Value Model.
  - (2000s) Graph Model.


employee
jobhistory
children
salaryhistory

employee (*man#*, name, birthdate, jobhistory, children)
jobhistory (*jobdate*, title, salaryhistory)
salaryhistory (*salarydate*, salary)
children (*childname*, birthyear)

Fig. 3(a). Unnormalized set

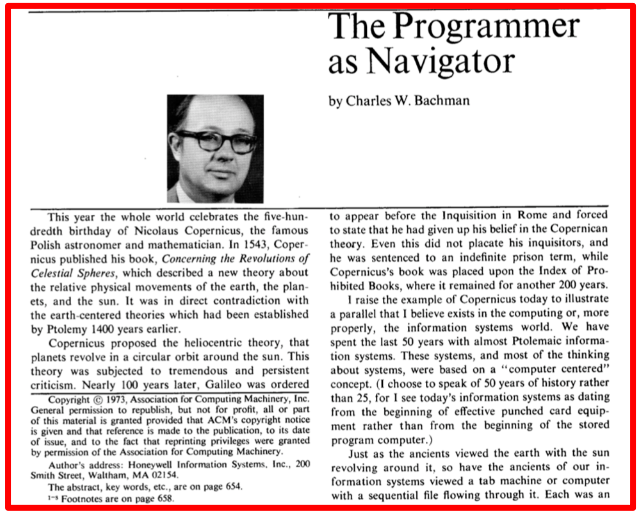
employee′ (*man#*, name, birthdate)
jobhistory′ (*man#*, *jobdate*, title)
salaryhistory′ (*man#*, *jobdate*, *salarydate*, salary)
children′ (*man#*, *childname*, birthyear)

Fig. 3(b). Normalized set

Example enterprise data management scenario from Codd’s 1970 relational model paper (Codd 1972)

Food for Thought

While Bachman touted the triumph of programmers for their ***ability and freedom to navigate****, Codd was concerned of the* ***burden and fragility of the same.***
What do you think?



The Programmer as Navigator

by Charles W. Bachman

This year the whole world celebrates the five-hundredth birthday of Nicolaus Copernicus, the famous Polish astronomer and mathematician. In 1543, Copernicus published his book, *Concerning the Revolutions of Celestial Spheres*, which described a new theory about the relative physical movements of the earth, the planets, and the sun. It was in direct contradiction with the earth-centered theories which had been established by Ptolemy 1400 years earlier.

Copernicus proposed the heliocentric theory, that planets revolve in a circular orbit around the sun. This theory was subjected to tremendous and persistent criticism. Nearly 100 years later, Galileo was ordered to appear before the Inquisition in Rome and forced to state that he had given up his belief in the Copernican theory. Even this did not placate his inquisitors, and he was sentenced to an indefinite prison term, while Copernicus's book was placed upon the Index of Prohibited Books, where it remained for another 200 years.

I raise the example of Copernicus today to illustrate a parallel that I believe exists in the computing or, more properly, the information systems world. We have spent the last 50 years with almost Ptolemaic information systems. These systems, and most of the thinking about systems, were based on a "computer centered" concept. (I choose to speak of 50 years of history rather than 25, for I see today's information systems as dating from the beginning of effective punched card equipment rather than from the beginning of the stored program computer.)

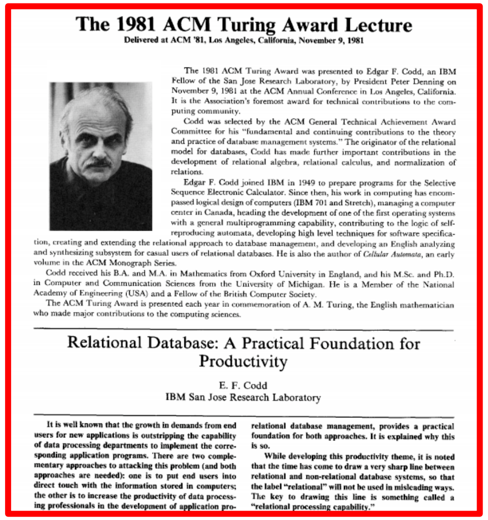
Just as the ancients viewed the earth with the sun revolving around it, so have the ancients of our information systems viewed a tab machine or computer with a sequential file flowing through it. Each was an

Author's address: Honeywell Information Systems, Inc., 200 Smith Street, Waltham, MA 02154.

The abstract, key words, etc., are on page 654.

1-5 Footnotes are on page 658.

vs.



The 1981 ACM Turing Award Lecture

Delivered at ACM '81, Los Angeles, California, November 9, 1981

The 1981 ACM Turing Award was presented to Edgar F. Codd, an IBM Fellow of the San Jose Research Laboratory, by President Peter Denning on November 9, 1981 at the ACM Annual Conference in Los Angeles, California. It is the Association's foremost award for technical contributions to the computing community.

Codd was selected by the ACM General Technical Achievement Award Committee for his "fundamental and continuing contributions to the theory and practice of database management systems." The originator of the relational model for databases, Codd has made further important contributions in the development of relational algebra, relational calculus, and normalization of relations.

Edgar F. Codd joined IBM in 1949 to prepare programs for the Selective Sequence Electronic Calculator. Since then, his work in computing has encompassed logical design of computers (IBM 701 and Stretch), managing a computer center in Canada, heading the development of one of the first operating systems with a general multiprogramming capability, contributing to the logic of self-reproducing automata, developing high level techniques for software specification, creating and extending the relational approach to database management, and developing an English analyzing and synthesizing subsystem for casual users of relational databases. He is also the author of *Cellular Automata*, an early volume in the ACM Monograph Series.

Codd received his B.A. and M.A. in Mathematics from Oxford University in England, and his M.Sc. and Ph.D. in Computer and Communication Sciences from the University of Michigan. He is a Member of the National Academy of Engineering (USA) and a Fellow of the British Computer Society.

The ACM Turing Award is presented each year in commemoration of A. M. Turing, the English mathematician who made major contributions to the computing sciences.

Relational Database: A Practical Foundation for Productivity

E. F. Codd
IBM San Jose Research Laboratory

It is well known that the growth in demands from end users for new applications is outstripping the capability of data processing departments to implement the corresponding application programs. There are two complementary approaches to attacking this problem (and both approaches are needed): one is to put end users into direct touch with the information stored in computers; the other is to increase the productivity of data processing professionals in the development of application pro- relational database management, provides a practical foundation for both approaches. It is explained why this is so.

While developing this productivity theme, it is noted that the time has come to draw a very sharp line between relational and non-relational database systems, so that the label "relational" will not be used in misleading ways. The key to drawing this line is something called a "relational processing capability."

References

- IBM Information Management System (IMS), 2017. Retrieved from https://www-01.ibm.com/software/data/ims/index.html.
- Bachman, 1973. Charles W. Bachman, *The Programmer as Navigator.* ACM Turing Award lecture, Communications of the ACM, Volume 16, Issue 11, 1973, pp. 653–658.
- Codd, 1982 . E. F. Codd: Relational Database: A Practical Foundation for Productivity. Commun. ACM 25(2): 109-117 (1982).
- Codd 1972. E. F. Codd: A Relational Model of Data for Large Shared Data Banks. Commun. ACM 13(6): 377-387 (1970).