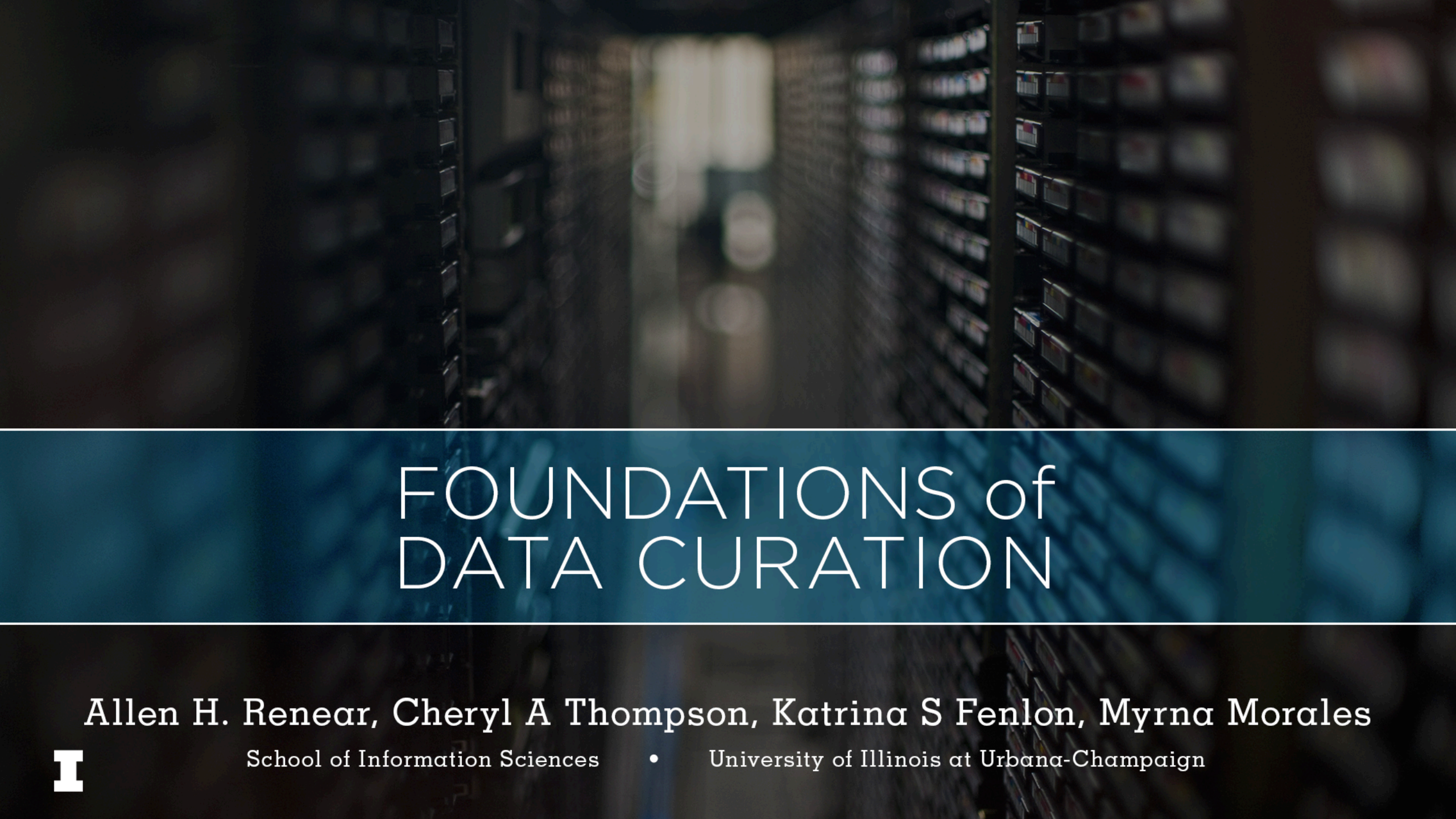
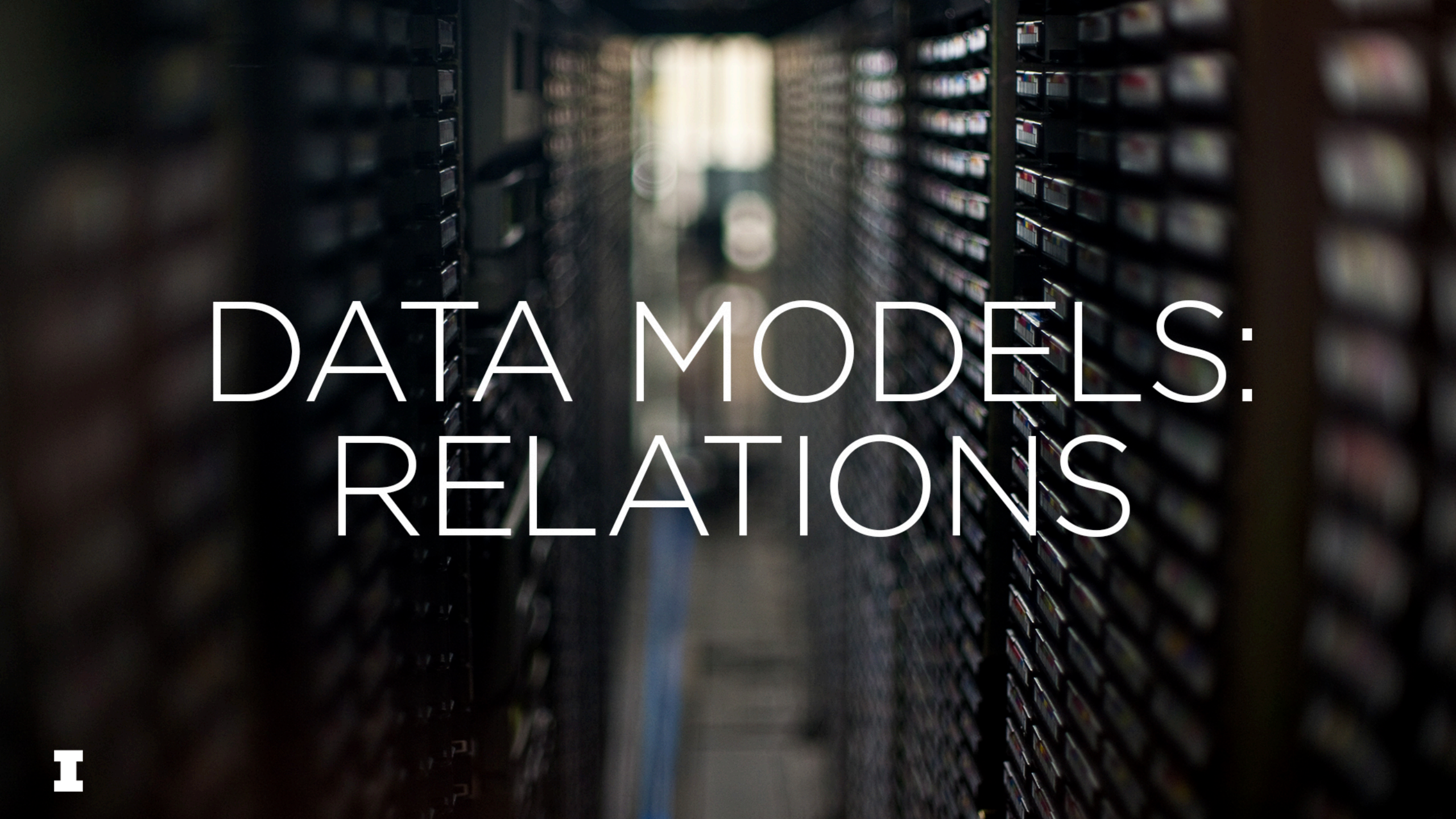


FOUNDATIONS of DATA CURATION

Allen H. Renear, Cheryl A Thompson, Katrina S Fenlon, Myrna Morales

School of Information Sciences • University of Illinois at Urbana-Champaign



DATA MODELS: RELATIONS

3

THE RELATIONAL MODEL

The Relational Model

- The problem, the solution
- Relations
    - The relational model
    - How the relational model addresses the problem
    - Relations from a mathematical POV
- The two fundamental principles of data organization
    - Abstraction
    - Indirection

The problem, again

We have just described two fundamental problems facing data management:

Programs and users interact with data directly via its storage structure
(And those storage structures vary wildly).

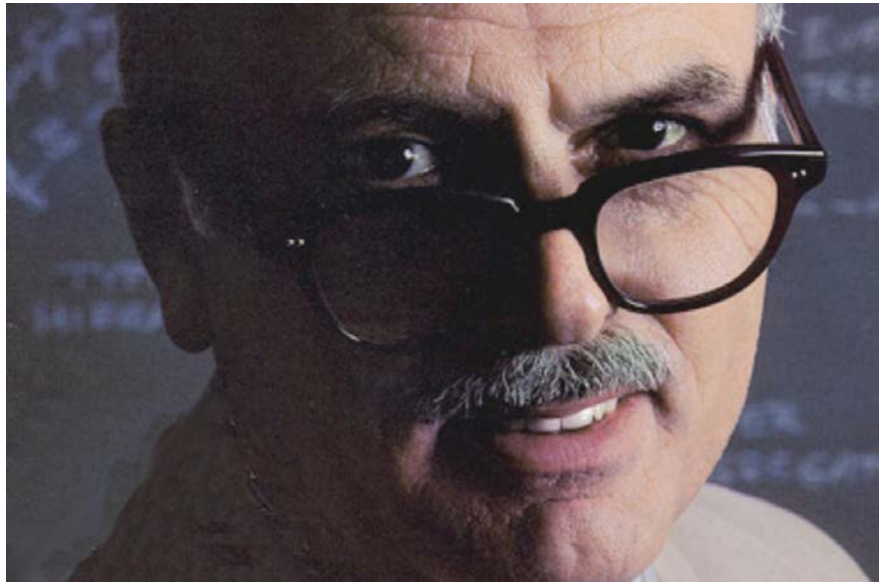
The intrinsic nature of the information is not reflected in the management systems.
These systems to not explicitly reflect the attributes, relationships, etc. that are the genuine components of the information being stored and managed.

The Solution

In 1970 E. F. Codd proposed a simple solution.

Conceptualize data as relations (tables)
and then map those relations
to whatever storage methods are being used

It changed the world.



E. F Codd

EF Codd in "A Relational Model of Data for Large Shared Data Banks" (1970).
Perhaps the most cited paper in computer science.

Relations (aka Tables)

Works	Work	Author	Title	Date	Last_Name	First_Name
	W58425	P42425	Moby Dick	1851	Melville	Herman
	W85246	P24246	The Scarlett Letter	1860	Hawthorne	Nathaniel
	W55427	P24246	Fanshawe	1828	Hawthorne	Nathaniel

Why it works

The relational model is a simple single high-level abstraction for conceptualizing information;

It is indifferent to the details of physical storage and processing

All interactions with the data are in relational (tabular) terms, such as attributes, values, tuples. Those interactions are translated into instructions expressed in terms of storage data instructions.

Operations on data are based on formally defined, well understood operations from logic and set theory.

How it works

Many of the problems described earlier are solved or mitigated by this approach

- Programmers and other users need know nothing about storage methods
- Programmers can learn a common language and approach to data management
- Documentation will have a common structure and organization
- Established mathematical methods may be used (such as set theory and logic)
- Data from different sources can be integrated more easily
- Data will be easier to check for validity and quality
- Data independence is supported:
  - storage methods can be changed without impacting programs
  - new data constructs can be added without impacting programs

Simple and relentless

Conceptualize all information in terms of relations (rows, columns, values)

Whenever you say anything , *say it in relations*
Whenever you do anything, *do it with relations*
Whenever you talk, *talk in relations*
Whenever you think, *think in relations*

Rows, columns, and values. Nothing else. ***Ever!***

Upload your data in rows and columns, query your data in rows and columns, receive your query results in rows and columns, only buy software that works on rows and columns, eat, drink and sleep rows and columns

Pay no attention to how the information is stored— that's not your problem!

relations, relations, relations, relations

The end

Internal/External

“Whatever you do in the privacy of your own CPU is your business,
but the interface you present in public must be: *relations*.”
[adapted from Michael Sperberg-McQueen]

Relations, from a mathematical POV

- A relation is a set of n-tuples:

{

<	W58425	P42425	Moby Dick	1851	Melville	Herman >
<	W85246	P24246	The Scarlet Letter	1860	Hawthorne	Nathaniel >
<	W55427	P24246	Fanshawe	1828	Hawthorne	Nathaniel >

}

Relations, from a mathematical POV

More formally:

An ***n-ary relation*** on sets A_1, A_2, …,A_n is any subset ***R*** of A_1 x A_2 x … x A_n.

Where each set A_1, A_2, …,A_n is the set of possible values for an attribute

And A_1 x A_2 x … x A_n. is the cartesian product of those sets and so is the set of all sets of n-tuples (relations) for those attribute values. And **R** will thus be one of those sets of n-tuples.

The sets A_i are the ***domains*** of the relation, and *n* is the ***degree*** of the relation.

[From Rosen. . .]

Two key principles: abstraction and indirection

Abstraction

Our data model is an *abstraction*; it abstracts away from the transient and varying details of storage and processing and focuses on the essential features of the data itself.

Indirection

Relational data management systems do not *directly* interact with the stored data representations, instead they interact *indirectly* with the stored data representations, via the relational representation that is mapped to the actual storage representation.

FOUNDATIONS OF DATA CURATION (IS531)

Allen H. Renear, Cheryl A Thompson, Katrina S Fenlon, Myrna Morales
School of Information Sciences
University of Illinois at Urbana-Champaign

Includes material adapted from work by Carole Palmer, Melissa Cragin, David Dubin, Karen Wickett, Bertram Ludaescher, Ruth Duerr and Simone Sacchi.

Comments and corrections to: renear@illinois.edu.