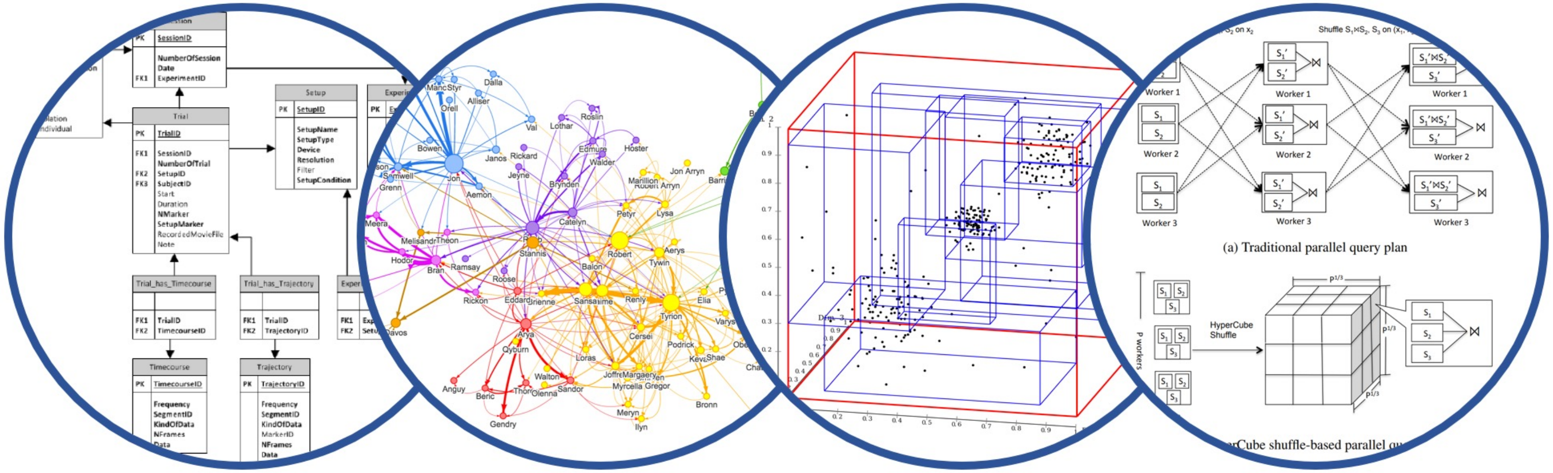


CSE 414: Intro to Data Management

Introduction

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Outline

1. Administrivia

2. Databases, DBMS

3. The Relational Data Model

414 Staff

Instructor:

- Dan Suciu
  suciu@cs

TAs:

- Zareef Amyeen
- Eden Chmielewski
- Cindy Fu
- Arjun Jagnani
- Moe Kayali
- Aaron D Kim
- Madrona Kelly Maling
- Qirui Wang
- Emi Kamaleiokekua Yoshikawa
- Andrew Mingwei Zhang

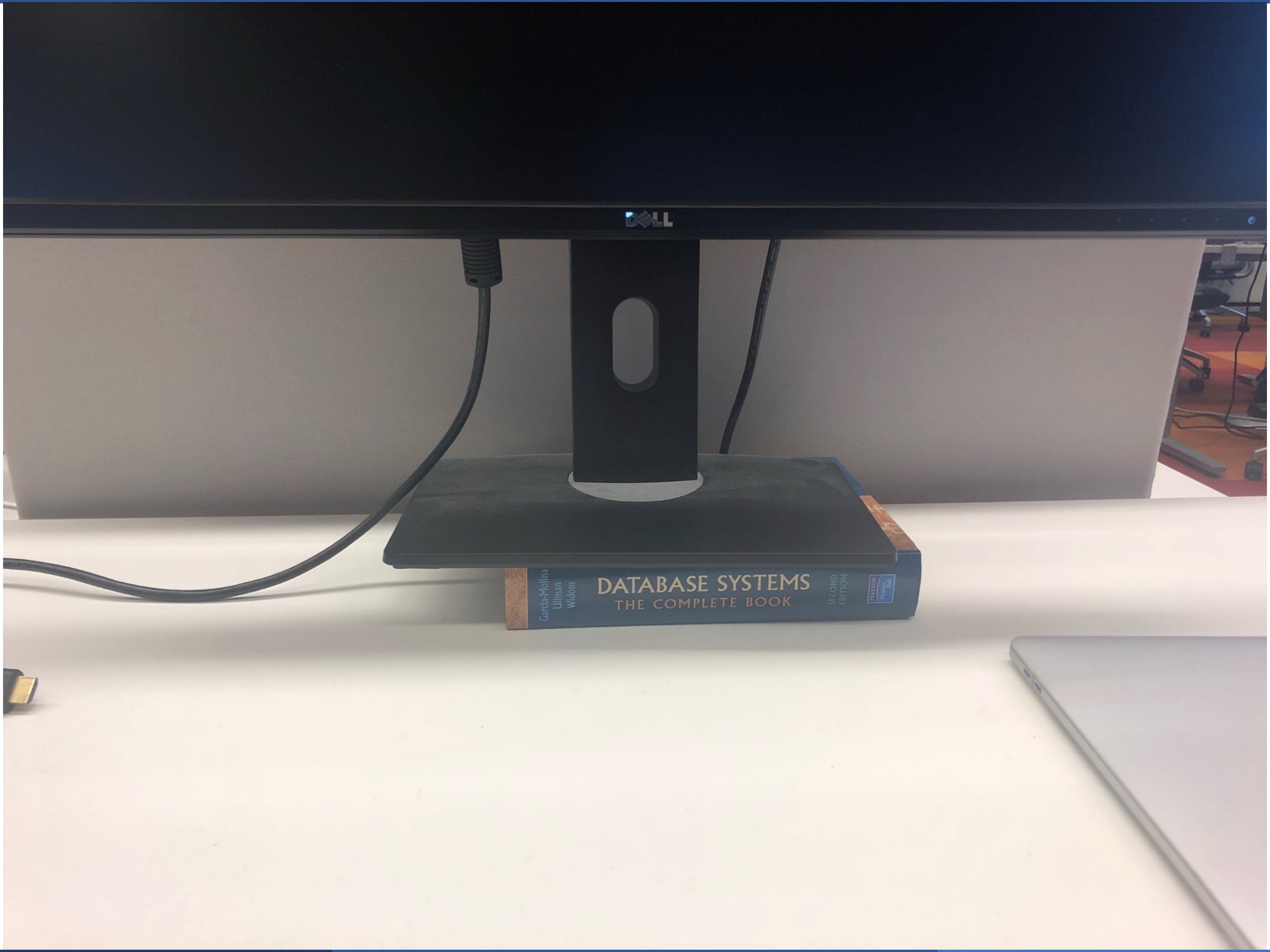
Course Format

- Lectures: in person, in this room
  - Attend. Arrive on time. Pay attention.
- Sections: in person, see locations at my.uw.edu
  - Bring your laptop
- Several homework assignments
  - First assignment published on gradescope
- Two exams:
  - Midterm: Friday, April 26, 10:30-11:20 in class
  - Final: Monday, June 3, 8:30-10:20 same room

Communication

- Website:
  - https://cs.uw.edu/414 same as
  - https://courses.cs.washington.edu/courses/cse414/24sp/
- Ed message board (link on website)
  - All course-related questions
  - Log in today, enable email notifications
- Class mailing list
  - Very low traffic, only important announcements

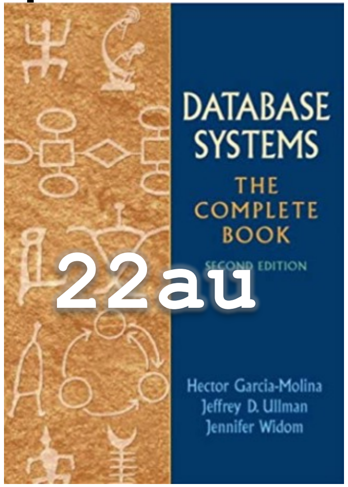
Textbook


DELL
Garcia-Molina
Ullman
Widom
DATABASE SYSTEMS
THE COMPLETE BOOK
SECOND EDITION
PEARSON Prentice Hall

Textbook

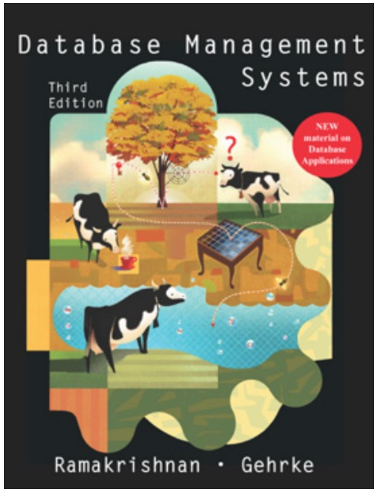
Main textbook, available at the bookstore or pdf:

- *Database Systems: The Complete Book*,
  Hector Garcia-Molina,
  Jeffrey Ullman,
  Jennifer Widom,
  second edition.



Also useful:

- *Database Management Systems*
  (3rd Edition)


Database Management Systems
Third Edition
NEW material on Database Applications
Ramakrishnan • Gehrke

Grading

- Grading:
  - Homeworks 50%, Exams 20%+30%
- Late days:
  - 6 in total, max 2/assignment in 24 hours chunks
- Collaboration:
  - Do complete homeworks individually
  - Do discuss concepts, but see previous item
  - Don’t show your work
  - Don’t post it on the Web
  - Don’t look at other peoples’ work
  - Don’t use AI tools to produce your work

Questions?

Questions?

Let’s get started!

Database

What is a database ?

Give examples of databases

Database

What is a database ?

- A collection of files storing related data

Give examples of databases

Database

What is a database ?

- A collection of files storing related data

Give examples of databases

- Accounts database
- Payroll database
- UW’s student database
- Amazon’s products database
- Airline reservation database

Database Management System

What is a DBMS ?

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Examples of DBMSs

- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Cloud: Snowflake, Redshift, BigQuery, SQL Azure
- Open source: MySQL (Sun/Oracle), PostgreSQL, DuckDB
- Open source library: **SQLite**

Database Management System

What is a DBMS ?

- *“A big program written by someone else that allows us to manage efficiently a large database and allows it to persist over long periods of time”*

Examples of DBMSs

- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Cloud: Snowflake, Redshift, BigQuery, SQL Azure
- Open source: MySQL (Sun/Oracle), PostgreSQL, DuckDB
- Open source library: **SQLite**

A DBMS needs a Data Model

Data Models

Example

Database of patients, their names, their health status…

How do we describe information?

Example

Database of patients, their names, their health status…

How do we describe information?

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy	…
345	Bob	Critical	…

Example

Database of patients, their names, their health status…

How do we describe information?

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy	…
345	Bob	Critical	…

Data Model

A **Data Model** is a mathematical formalism to describe data. It is how we can talk about data conceptually without having to think about implementation.

3 Parts of a Data Model

The 3 parts of any data model

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy?	…
345	Bob	Critical	…

3 Parts of a Data Model

The 3 parts of any data model

- Instance
  - The actual **data**

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy?	…
345	Bob	Critical	…

3 Parts of a Data Model

The 3 parts of any data model

- Instance
  - The actual **data**
- Schema
  - A **description** of what data is being stored

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy?	…
345	Bob	Critical	…

3 Parts of a Data Model

The 3 parts of any data model

- Instance
  - The actual **data**
- Schema
  - A **description** of what data is being stored
- Query Language
  - How to retrieve and manipulate data

Medical Records

PatientID	Name	Status	Notes
123	Alex	Healthy?	…
345	Bob	Critical	…

“Which patients are critical?”

```
SELECT * FROM records
WHERE status=“critical”
```

Data Models

There are lots of models out there!

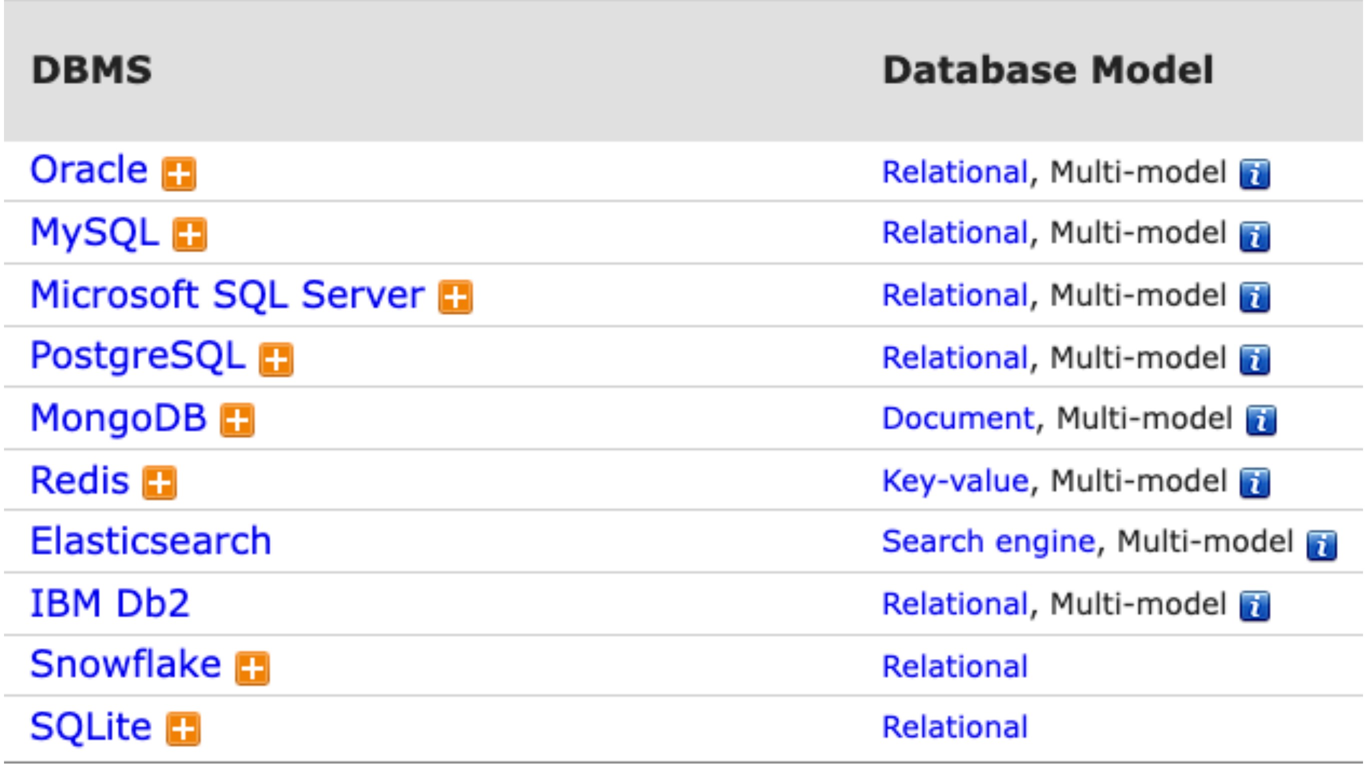
- Relational
- Semi-structured
- Key-value pairs
- Graph
- OO
- …

Data Models

There are lots of models out there!

- Relational
- Semi-structured
- Key-value pairs
- Graph
- OO
- …

https://db-engines.com/en/ranking



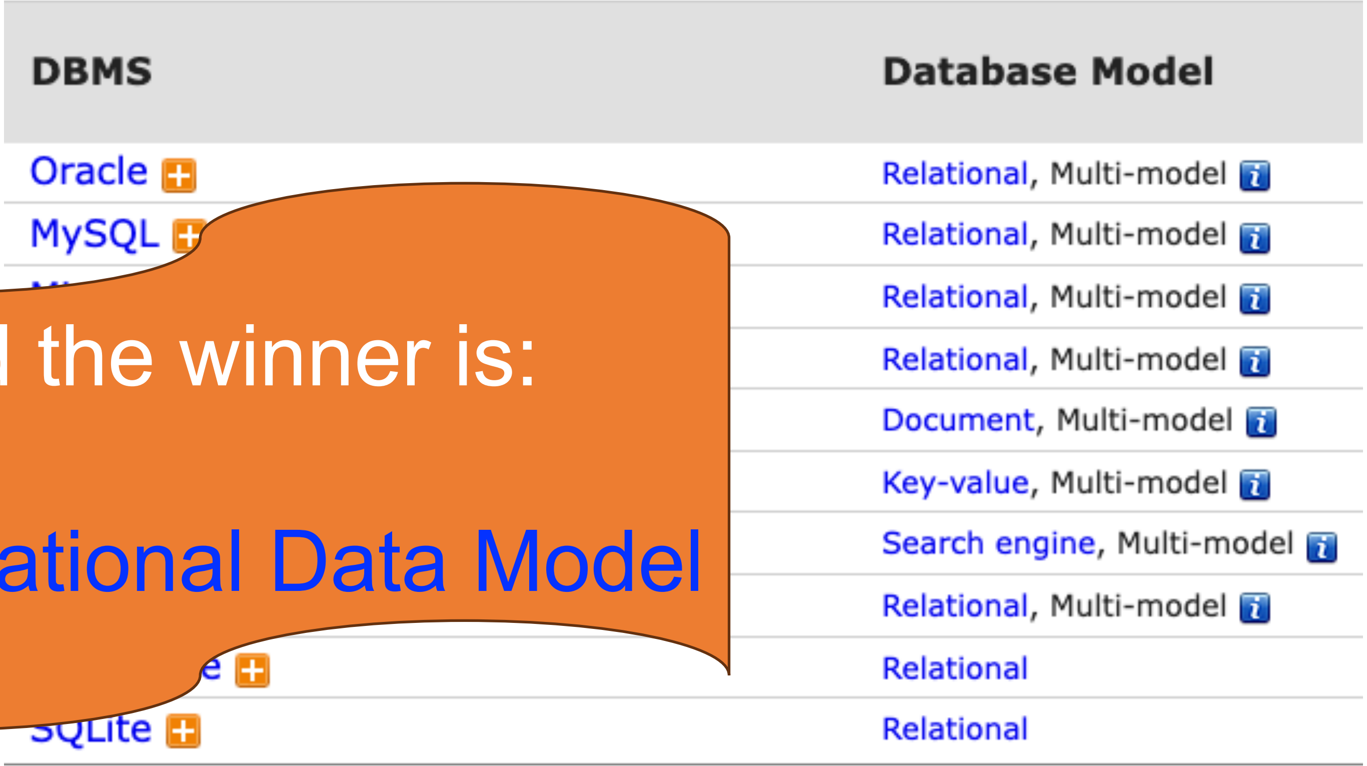
DBMS	Database Model
Oracle	Relational, Multi-model
MySQL	Relational, Multi-model
Microsoft SQL Server	Relational, Multi-model
PostgreSQL	Relational, Multi-model
MongoDB	Document, Multi-model
Redis	Key-value, Multi-model
Elasticsearch	Search engine, Multi-model
IBM Db2	Relational, Multi-model
Snowflake	Relational
SQLite	Relational

Data Models

There are lots of models out there!

- Relational
- Semi-structured
- Key-value pairs
- Graph
- OO
- …

https://db-engines.com/en/ranking



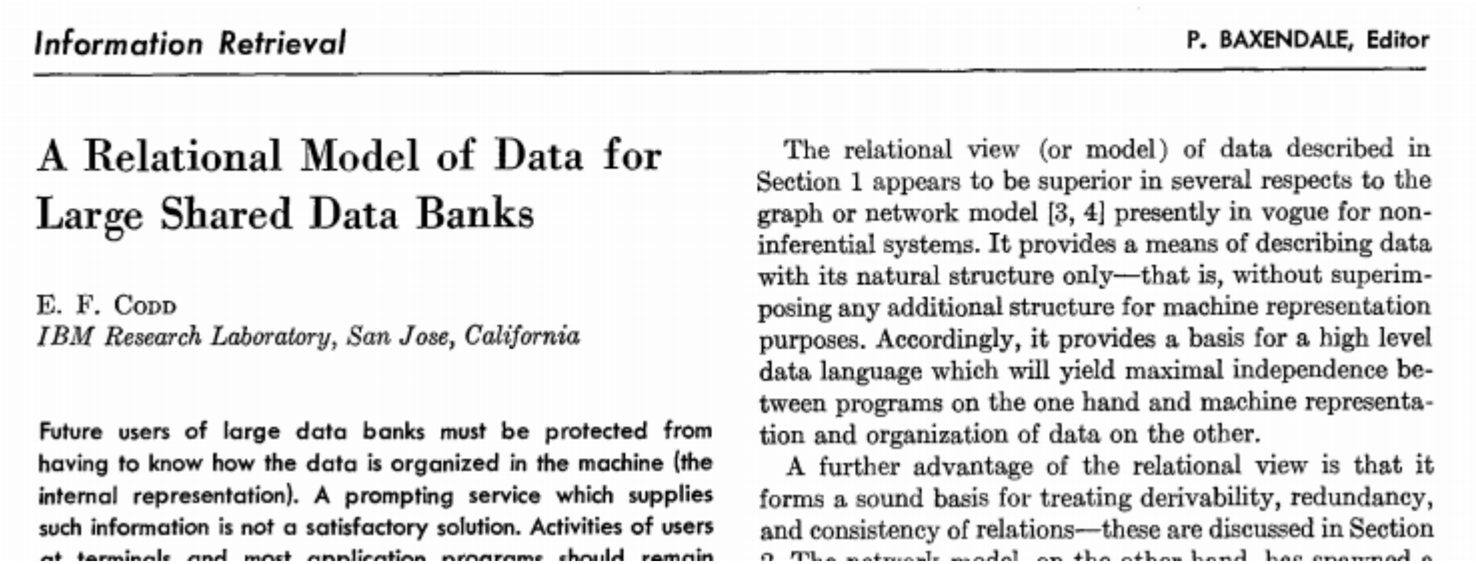
DBMS	Database Model
Oracle	Relational, Multi-model
MySQL	Relational, Multi-model
	Relational, Multi-model
	Relational, Multi-model
	Document, Multi-model
	Key-value, Multi-model
	Search engine, Multi-model
	Relational, Multi-model
	Relational
SQLite	Relational

And the winner is:

The Relational Data Model

Relational Data Model

What is the Relational Model?



Information Retrieval — P. BAXENDALE, Editor

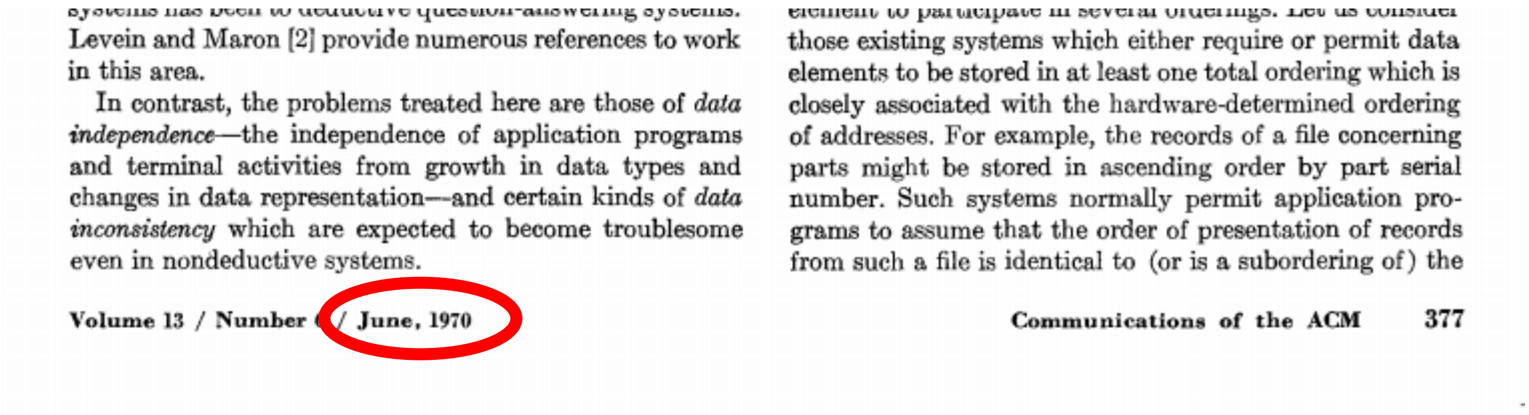
A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a



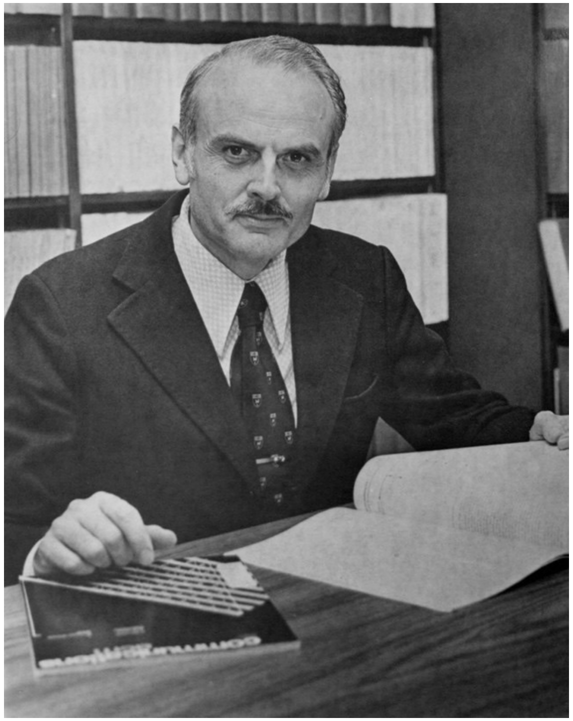
systems has been to deductive question-answering systems. Levein and Maron [2] provide numerous references to work in this area.

In contrast, the problems treated here are those of *data independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of *data inconsistency* which are expected to become troublesome even in nondeductive systems.

element to participate in several orderings. Let us consider those existing systems which either require or permit data elements to be stored in at least one total ordering which is closely associated with the hardware-determined ordering of addresses. For example, the records of a file concerning parts might be stored in ascending order by part serial number. Such systems normally permit application programs to assume that the order of presentation of records from such a file is identical to (or is a subordering of) the

Volume 13 / Number 6 / June, 1970 — Communications of the ACM 377

The Relational Model



Ted Codd

Turing Award 1981

The Relational Model

- Data is stored in simple, flat relations
- Is retrieved via a set-at-a-time query language
- No prescription for the physical representation

The Relational Model

- Data is stored in simple, flat relations

We start here

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation

Components of the Relational Model

Payroll (UserId, Name, Job, Salary)

Components of the Relational Model

Schema, describes data

Payroll (UserId, Name, Job, Salary)

Components of the Relational Model

Schema, describes data

Payroll (UserId, Name, Job, Salary)

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Components of the Relational Model

Schema, describes data

Payroll (UserId, Name, Job, Salary)

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Instance of actual data

Components of the Relational Model

Table/ Relation

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

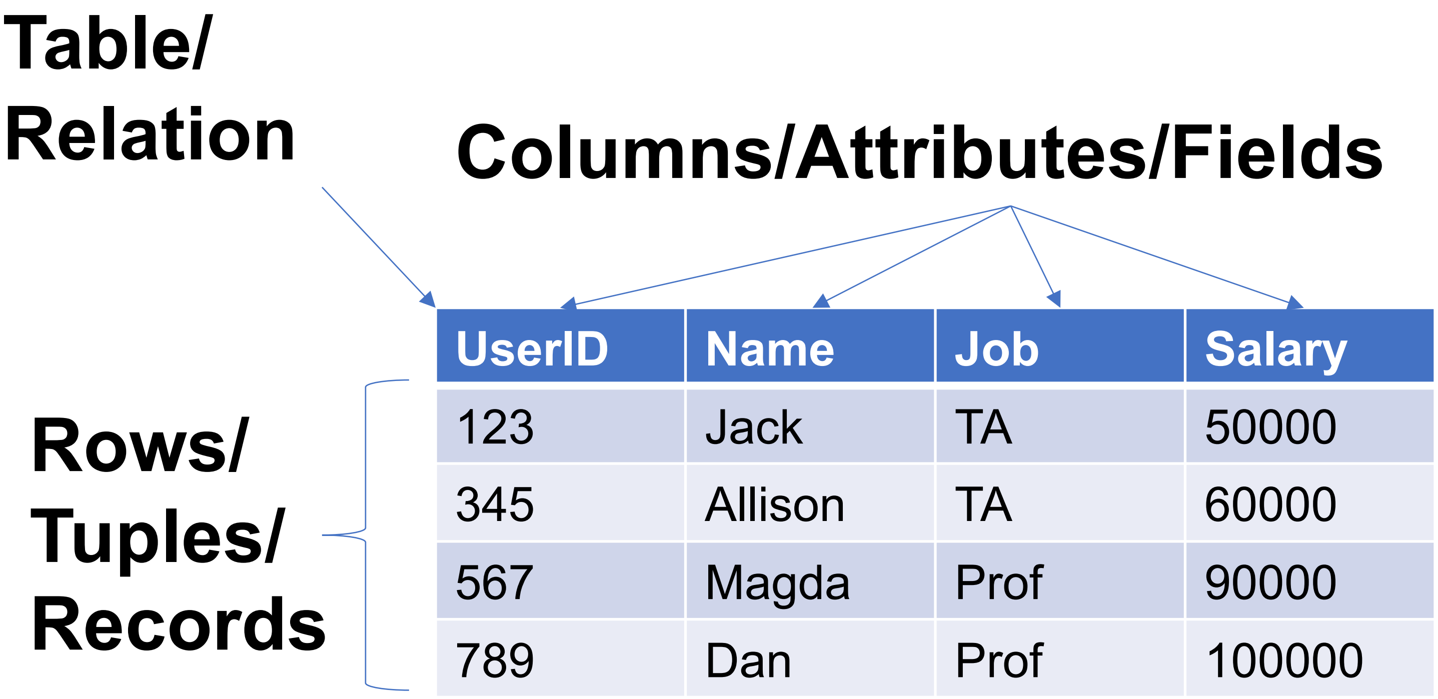
Components of the Relational Model

Table/ Relation

Rows/ Tuples/ Records

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Components of the Relational Model



Table/ Relation

Columns/Attributes/Fields

Rows/ Tuples/ Records

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Characteristics of the Relational Model

- Set semantics

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Characteristics of the Relational Model

- Set semantics
- Order doesn’t matter

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

=

UserID	Name	Job	Salary
567	Magda	Prof	90000
123	Jack	TA	50000
789	Dan	Prof	100000
345	Allison	TA	60000

Characteristics of the Relational Model

- Set semantics
- Order doesn’t matter
- Duplicates not allowed

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000
789	Dan	Prof	100000

Violates set semantics!

Characteristics of the Relational Model

- Set semantics
- Order doesn't matter
- Duplicates not allowed
- …but systems do allow them

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000
789	Dan	Prof	100000

Allowed by systems, but bad idea

Characteristics of the Relational Model

- Attributes are **typed** and **static**
  - INTEGER, FLOAT, VARCHAR(n), DATETIME, …

UserID	Name	Job	Salary
123	Jack	TA	banana
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Violates attribute type assuming INT

Characteristics of the Relational Model

- Attributes are **typed** and **static**
  - INTEGER, FLOAT, VARCHAR(n), DATETIME, …
- Tables are **flat**

No sub-tables allowed!

<table>
<tr><th>UserID</th><th>Name</th><th>Job</th><th>Salary</th></tr>
<tr><td>123</td><td>Jack</td><td><table><tr><th>JobName</th><th>HasBananas</th></tr><tr><td>TA</td><td>0</td></tr><tr><td>farmer</td><td>1</td></tr></table></td><td>0000</td></tr>
<tr><td>345</td><td>Allison</td><td>TA</td><td>60000</td></tr>
<tr><td>567</td><td>Magda</td><td>Prof</td><td>90000</td></tr>
<tr><td>789</td><td>Dan</td><td>Prof</td><td>100000</td></tr>
</table>

The Relational Model

- Data is stored in simple, flat relations

We saw this

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation

The Relational Model

- Data is stored in simple, flat relations

We saw this

- Is retrieved via a set-at-a-time query language

What doe this mean?

- No prescription for the physical representation

Characteristics of the Relational Model

But how is this data ACTUALLY stored?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

Characteristics of the Relational Model

But how is this data ACTUALLY stored?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

“123\tJack\tTA\t50000\t345\tAllison…” or maybe

“123\t345\t567\t789\tJack\tAllison…”

Characteristics of the Relational Model

But how is this data ACTUALLY stored?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

~~“123\tJack\tTA\t50000\t345\tAllison…” or maybe~~

~~“123\t345\t567\t789\tJack\tAllison…”~~

No prescription for physical storage: system decides

Characteristics of the Relational Model

But how is this data ACTUALLY stored?

Payroll

UserID	Name	Job	Salary
123	Jack	TA	50000
345	Allison	TA	60000
567	Magda	Prof	90000
789	Dan	Prof	100000

~~“123\tJack\tTA\t50000\t345\tAllison…” or maybe~~

~~“123\t345\t567\t789\tJack\tAllison…”~~

Physical Data Independence

No prescription for physical storage: system decides

The Relational Model

- Data is stored in simple, flat relations
- Is retrieved via a set-at-a-time query language
- No prescription for the physical representation

The Relational Model

We discussed this…

- Data is stored in simple, flat relations
- Is retrieved via a set-at-a-time query language
- No prescription for the physical representation

…and this

The Relational Model

- Data is stored in simple, flat relations

Next Lectures: SQL

- Is retrieved via a set-at-a-time query language

- No prescription for the physical representation