CSE 199: UB Seminar (Fall 2026) - Data management

Lecture 1: What, How and Why?

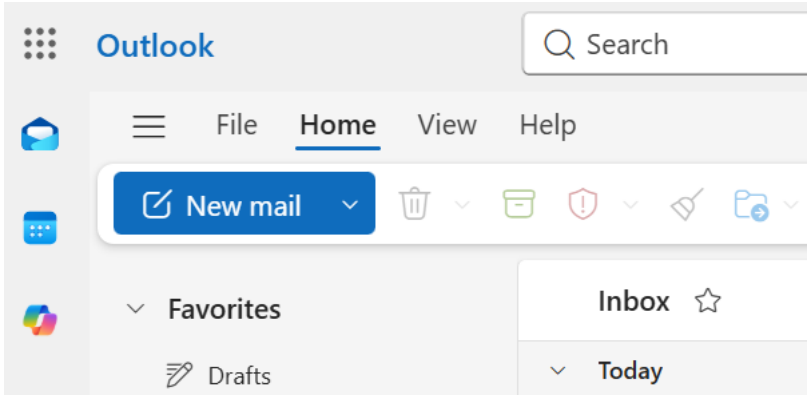
Zhuoyue Zhao
zzhao35 [at] buffalo [dot] edu

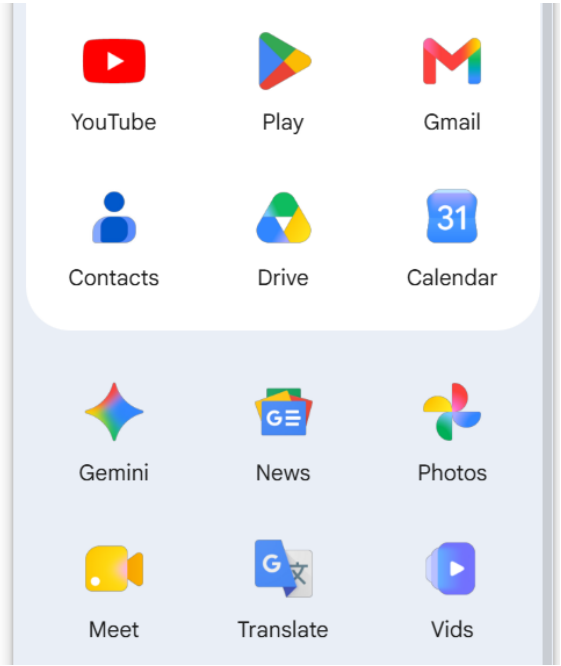
University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

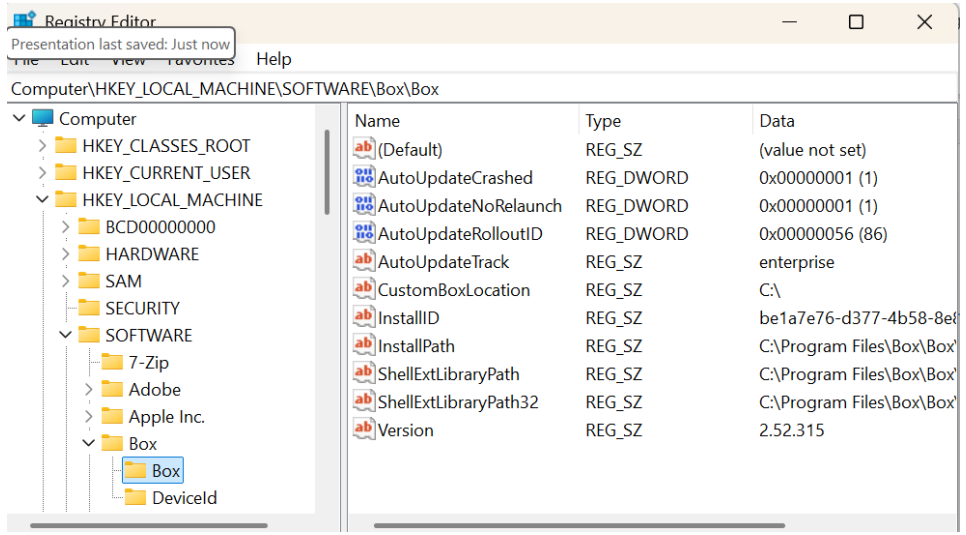
What is a Database?

- Database is
  - a collection of interrelated data
  - often organized in a certain structure for convenient and efficient access
  - behind any modern application you can find

UB : LEARNS



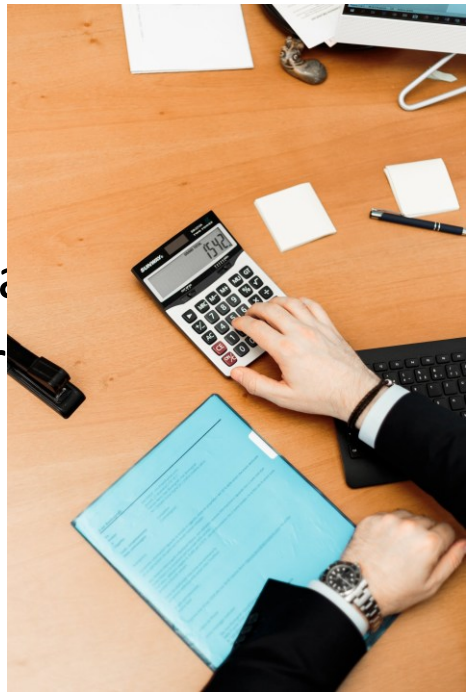




How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts

  - Step 2: do some analysis
    - How much did my spend on grocery and fast food in Febura
    - How much could I have saved if I cook by myself in Feburar
    - What about January/last quarter/last year/past five years?

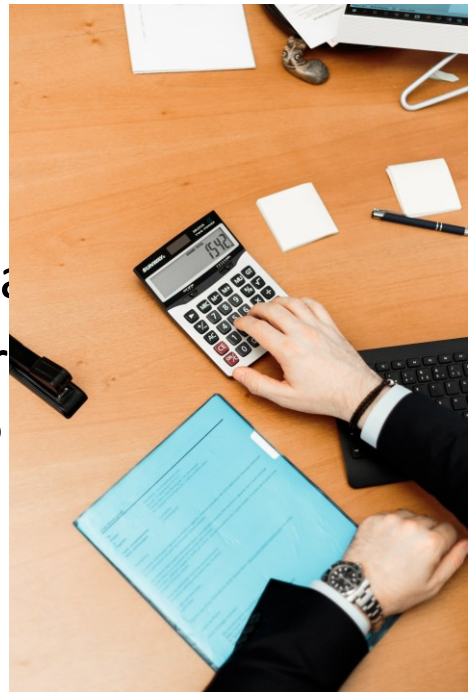


How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: write them down on a notebook

Date	Amount	Description
2/1	$20.21	Grocery
2/2	$10.54	Fast food
2/3	$39.22	Cell phone bill
…		
2/27	$33.00	Clothes

  - Step 3: do some analysis
    - How much did my spend on grocery and fast food in Febura
    - How much could I have saved if I cook by myself in Feburar
    - What about January/last quarter/last year/past five years?



How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    store them in a text file

```
Date   Amount   Description
2/1    $20.21    Grocery
2/2    $10.54   Fast food
2/3    $39.22    Cell phone bill
…
2/27   $33.00    Clothes
```

  - Step 3: do some analysis
    - How much did my spend on grocer
    - How much could I have saved if I co
    - What about January/last quarter/la

```
f = open('myspend_feb_22.txt', 'r')
grocery = 0
fast_food = 0
for line in f:
    date, amount, desc = line.split(' ')
    if desc == 'Fast food':
        fast_food += eval(amount)
    elif desc == 'Grocery':
        grocery += eval(amount)
……
```

How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    ~~store them in a text file~~
    use a spreadsheet

```
Date   Amount   Description
2/1    $20.21    Grocery
2/2    $10.54   Fast food
2/3    $39.22    Cell phone bill
…
2/27   $33.00    Clothes
```

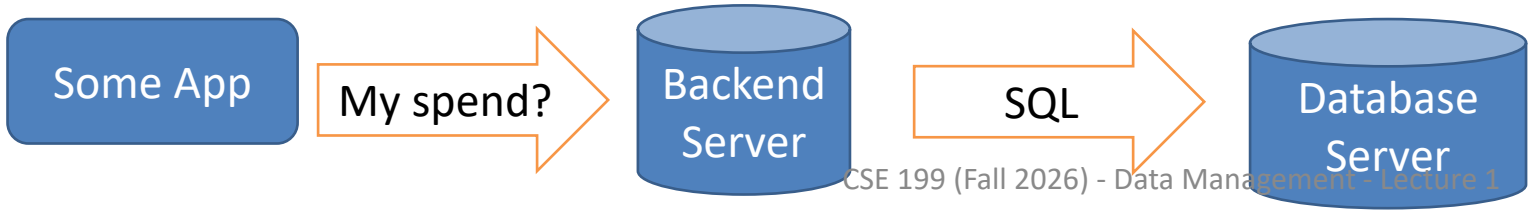
	A	B	C	D	E
1	Date	Amount	Description		
2	1-Feb	20.21	Grocery		
3	2-Feb	10.54	Fast food		
4	3-Feb	39.22	Cell phone		
5					
6					
7		Grocery	=SUMIFS(B2:B4,C2:C4,"Grocery")		

  - Step 3: do some analysis
    - How much did my spend on grocery and fast f
    - How much could I have saved if I cook by myse
    - What about January/last quarter/last year/pas

How to manage a database?

- Suppose I'd like to track my daily spending
- What I can do:
  - Step 1: collect all the receipts
  - Step 2: ~~write them down on a notebook~~
    ~~store them in a text file~~
    ~~use a spreadsheet~~
    use/build a personal finance app
  - Step 3: do some analysis
    - How much did my spend on grocery and fast food in Feburary?
    - How much could I have saved if I cook by myself in Feburary?
    - What about January/last quarter/last year/past five years?

Date	Amount	Description
2/1	$20.21	Grocery
2/2	$10.54	Fast food
2/3	$39.22	Cell phone bill
…		
2/27	$33.00	Clothes



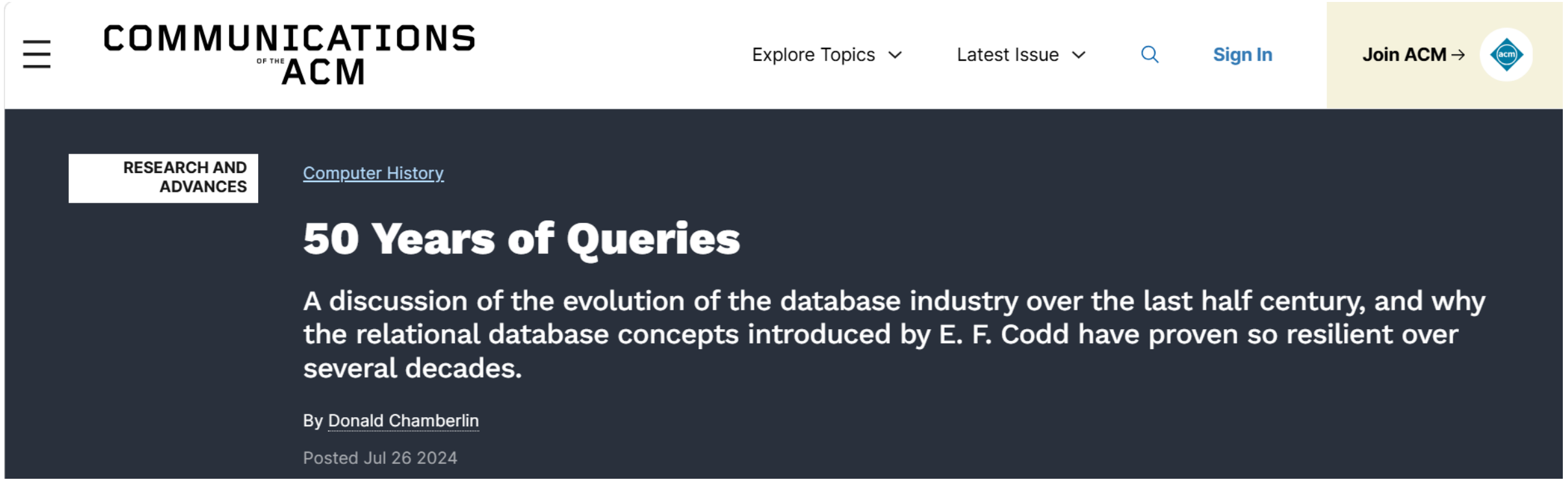

```
SELECT category, SUM(amount)
FROM spend
WHERE userid = 123456
GROUP BY category;
```

Why using a DataBase Management System?

- DataBase Management System (DBMS) is a software system for convenient and efficient data access over databases,

  which provides:
  - Data abstraction
    - Flexible data manipulation and query interfaces
    - Scalable data storage
    - Efficient query and transaction processing
  - Integrity checks
  - Concurrency control and atomicity
  - Fault tolerance
  - Security and privacy
  - …

A historical review of query processing

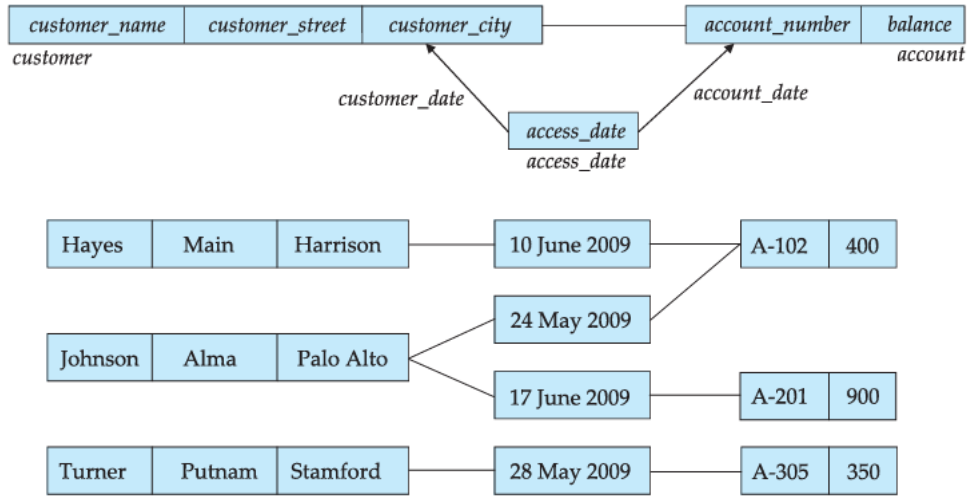


[50 Years of Queries – Communications of the ACM]

Two competing data models

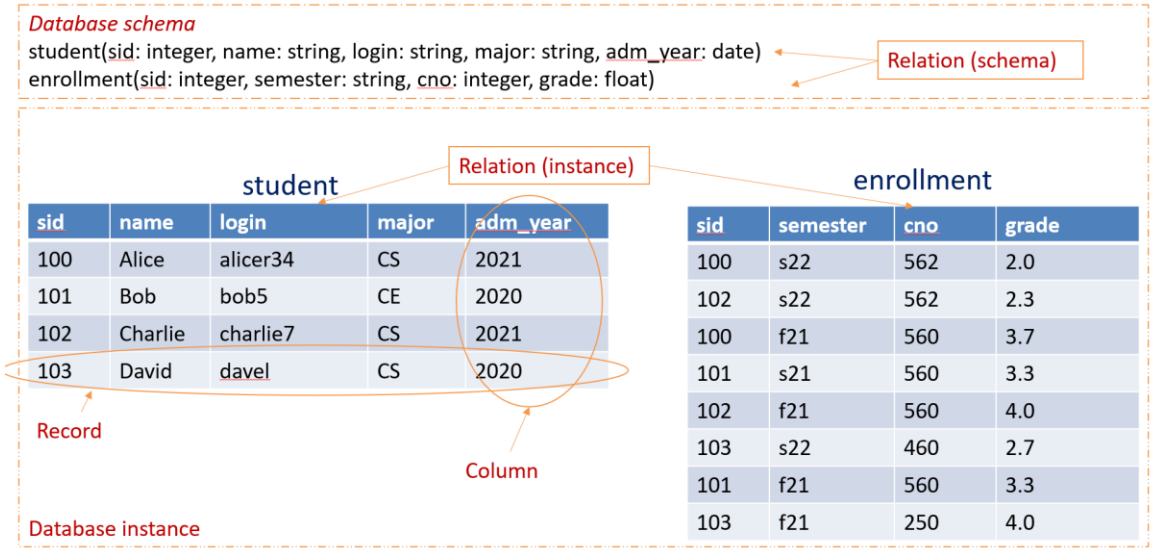
Network Model/DBTG [1]

- Links between data structures
  - Must navigate through owner-member links
  - Less flexible



Relational Model/SQL

- Tabular data
  - Can model one-to-one/one-to-many/many-to-many relationships
  - Much more flexible

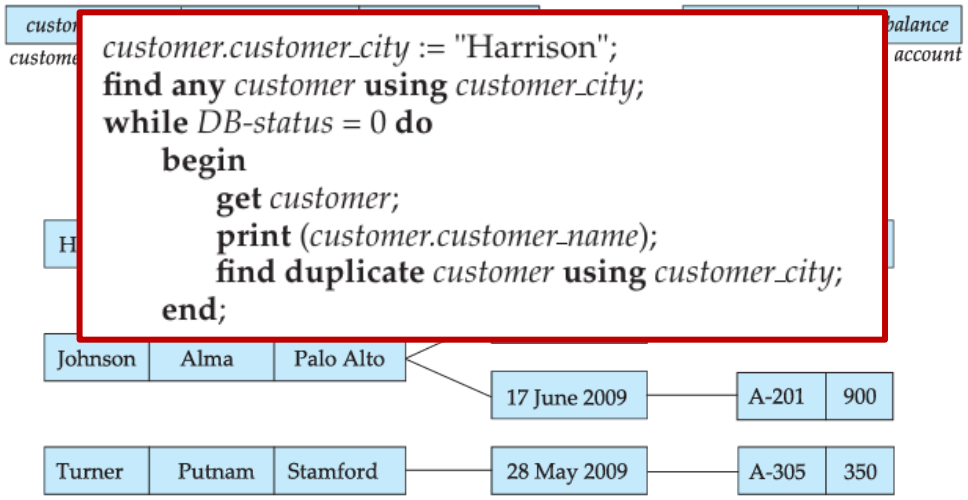


[1] Silberschatz et al., Database System Concepts, 6th edition, Appendix D (2010).

Two competing data models (cont'd)

Network Model/DBTG [1]

- Links between data structures
  - Must navigate through owner-member links
  - Less flexible
- Embedded in Programming Languages
  - Unfriendly to non-programmers



[1] Silberschatz et al., Database System Concepts, 6th edition, Appendix D (2010).

Relational Model/SQL

- Tabular data
  - Can model one-to-one/one-to-many/many-to-many relationships
  - Much more flexible
- "English"-like query language (SQL)
  - "Friendly" to non-programmers

```
SELECT S.name, E.grade
FROM student S, enrollment E
WHERE S.sid=E.sid AND E.cno=562;
```

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

Record

Column

sid	semester	cno	grade
100	s22	562	2.0
102	s22	562	2.3
100	f21	560	3.7
101	s21	560	3.3
102	f21	560	4.0
103	s22	460	2.7
101	f21	560	3.3
103	f21	250	4.0

Database instance

SQL was designed to be user friendly

The first publication of the proposed SEQUEL language was the paper that appeared in the Proceedings of SIGFIDET 1974.[8] Shortly after this paper appeared, my friend Ray Boyce died, suddenly and unexpectedly, of a brain aneurism.

Ray's death was a great loss, but the SEQUEL work continued. To test our hypothesis that SEQUEL could be understood by non-programmers, a psychologist at IBM Research named Phyllis Reisner conducted an experiment at San Jose State University in which she showed that college students with no experience in programming or data handling could gain a reasonable proficiency with SEQUEL after a few hours of instruction.[21]

As the System R project matured, the SEQUEL language continued to evolve. A 1976 paper titled SEQUEL 2[7] extended the query syntax to cover insert, delete, and update operations; view definitions; integrity assertions; and triggered actions. The language defined in that paper would be immediately recognized by database developers working today. In 1977, the SEQUEL name was shortened to SQL, an acronym for "Structured Query Language."

50 Years of Queries – Communications of the ACM

Quiz time

student

sid	name	login	major	adm_year
100	Alice	alicer34	CS	2021
101	Bob	bob5	CE	2020
102	Charlie	charlie7	CS	2021
103	David	davel	CS	2020

```
SELECT name, login, major
FROM student
WHERE adm_year = 2020;
```

- Submit your answer on tophat.

Commercialization and standardization

- Many companies joined the relational DBMS industry in the 80s/90s
  - IBM, Microsoft, Oracle, etc.
  - and many more open-source software from the mid-1990s

- SQL, formerly SEQUEL, was standardized by
  - ANSI (American National Standards Institute) in 1986
    - Updated with new features every a few years.

  - Created a DBMS market where
    companies can compete with different product over the same interfaces

Many things that relational model got right

- Flexibility
- Performance

- Standard
- Data is sticky

- Research published early and openly
- High-quality open-source implementation

50 Years of Queries – Communications of the ACM

Our plans

- **Network model & DBTG**
  - Lecture 2 & Recitation 1
- **Relational model & SQL**
  - Lecture 3, homework
- **NoSQL, NewSQL, and Natural Language Queries**
  - Lectures 4 & 5, Recitation 2