Kurs **Datenbankgrundlagen und Modellierung**

Sebastian Maneth, Universität Bremen
maneth@uni-bremen.de

Sommersemester 2023

17.4.2023
Vorlesung 1: Einführung

Kurs **Datenbankgrundlagen und Modellierung**

2V + 2Ü 03-IBGP-DBM
ECTS: 6

Vorlesung (Maneth)
Mo 10:15—11:45 HS 2010 (Großer Hörsaal)

Übung (Westenberg)
Mi 16:15—17:45 NW2 C0290 (Hörsaal 1)

Fragestunden (Haria & Maneth)
Do 10:15 — 11:45 NW1 H 2 - W0020
Do 14:15 — 15:45 GW2 B3009 (Großer Studierraum)

Kurs **Datenbankgrundlagen und Modellierung**

2V + 2Ü 03-IBGP-DBM
ECTS: 6

Vorlesung (Maneth)
Mo 10:15—11:45 HS 2010 (Großer Hörsaal)

Übung (Westenberg)
Mi 16:15—17:45 NW2 C0290 (Hörsaal 1)

Fragestunden (Haria & Maneth)
Do 10:15 — 11:45 NW1 H 2 - W0020
Do 14:15 — 15:45 GW2 B3009 (Großer Studierraum)

Scheinerwerb:
Erfolgreiche Teilnahme an **Klausur**

(d.h. es müssen keine Übungsblätter abgegeben werden)

Kurs **Datenbankgrundlagen und Modellierung**

17.4.	Vorlesung 1
24.4.	V2
1.5.	keine Vorlesung
4.5.	Fragestunden finden statt
8.5.	V3
10.5.	Ü1
11.5.	Fragestunden
15.5.	keine Vorlesung
17.5.	Ü2
22.5.	V4
24.5.	Ü3
25.5.	Fragestunden
29.5.	V5
31.5.	Ü4
1.6.	Fragestunden
5.6.	V6
7.6.	Ü5

Kurs **Datenbankgrundlagen und Modellierung**

17.4.	Vorlesung 1 — Intro
24.4.	V2 — ER, SQL
1.5.	keine Vorlesung
4.5.	Fragestunden finden statt
8.5.	V3 — SQL
10.5.	Ü1
11.5.	Fragestunden
15.5.	keine Vorlesung
17.5.	Ü2
22.5.	V4 — SQL
24.5.	Ü3
25.5.	Fragestunden
29.5.	V5 — funct. dependencies
31.5.	Ü4
1.6.	Fragestunden
5.6.	V6 — normal forms
7.6.	Ü5
8.6.	Fragestunden

Kurs **Datenbankgrundlagen und Modellierung**

17.4.	Vorlesung 1 — Intro
24.4.	V2 — ER, SQL
1.5.	keine Vorlesung
4.5.	Fragestunden finden statt
8.5.	V3 — SQL
10.5.	Ü1
11.5.	Fragestunden
15.5.	keine Vorlesung
17.5.	Ü2
22.5.	V4 — SQL
24.5.	Ü3
25.5.	Fragestunden
29.5.	V5 — funct. dependencies
31.5.	Ü4
1.6.	Fragestunden
5.6.	V6 — normal forms
7.6.	Ü5
8.6.	Fragestunden
12.6.	V7
14.6.	Ü6
15.6.	Fragestunden
19.6.	V8
21.6.	Ü7
22.6.	Fragestunden
26.6.	V9
28.6.	Ü8
29.6.	Fragestunden
3.7.	V10
5.7.	Ü9
6.7.	Fragestunden
10.7.	V11

Kurs **Datenbankgrundlagen und Modellierung**

17.4.	Vorlesung 1 — Intro
24.4.	V2 — ER, SQL
1.5.	keine Vorlesung
4.5.	Fragestunden finden statt
8.5.	V3 — SQL
10.5.	Ü1
11.5.	Fragestunden
15.5.	keine Vorlesung
17.5.	Ü2
22.5.	V4 — SQL
24.5.	Ü3
25.5.	Fragestunden
29.5.	V5 — funct. dependencies
31.5.	Ü4
1.6.	Fragestunden
5.6.	V6 — normal forms
7.6.	Ü5
8.6.	Fragestunden
12.6.	V7 — modelling Intro
14.6.	Ü6
15.6.	Fragestunden
19.6.	V8 — class diagrams
21.6.	Ü7
22.6.	Fragestunden
26.6.	V9 — state charts
28.6.	Ü8
29.6.	Fragestunden
3.7.	V10 — sequence diagrams
5.7.	Ü9
6.7.	Fragestunden
10.7.	V11 — Klausurvorbereitung

Kurs **Datenbankgrundlagen und Modellierung**

Kurs **Datenbankgrundlagen und Modellierung**

(1.) Entity-Relationship Diagrams
Creating and Modifying Tables

(2.) Database Normalforms

(3.) SQL

(4.) Statical Modeling
- — Class Diagrams
- — Object Diagrams

(5.) Dynamical Modeling
- — state charts
- — sequence diagrams

Kurs **Datenbankgrundlagen und Modellierung**

(1.) Entity-Relationship Diagrams
Creating and Modifying Tables

(2.) Database Normalforms

(3.) SQL

(alte) DBG Themen

Es werden weiterhin Wiederholungsklausuren für nur diesen Teil angeboten.

(5.) Statical Modeling
— Class Diagrams
— Object Diagrams

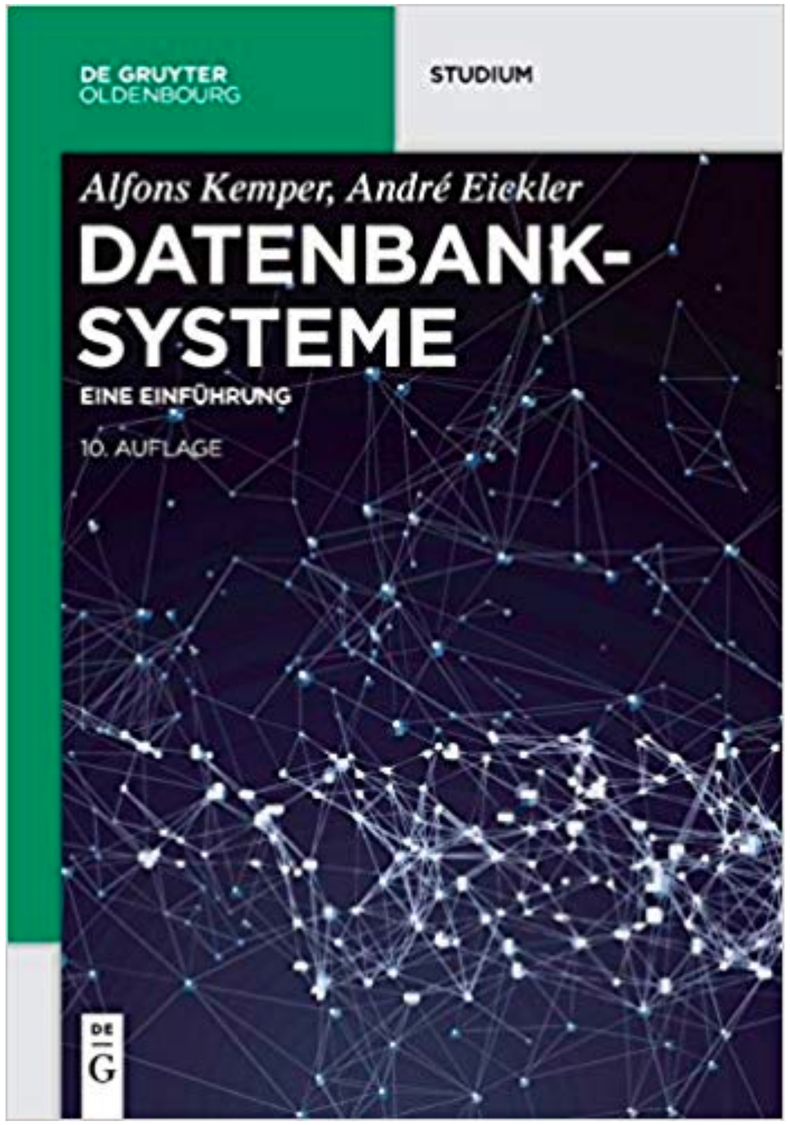
(6.) Dynamical Modeling
— state charts
— sequence diagrams

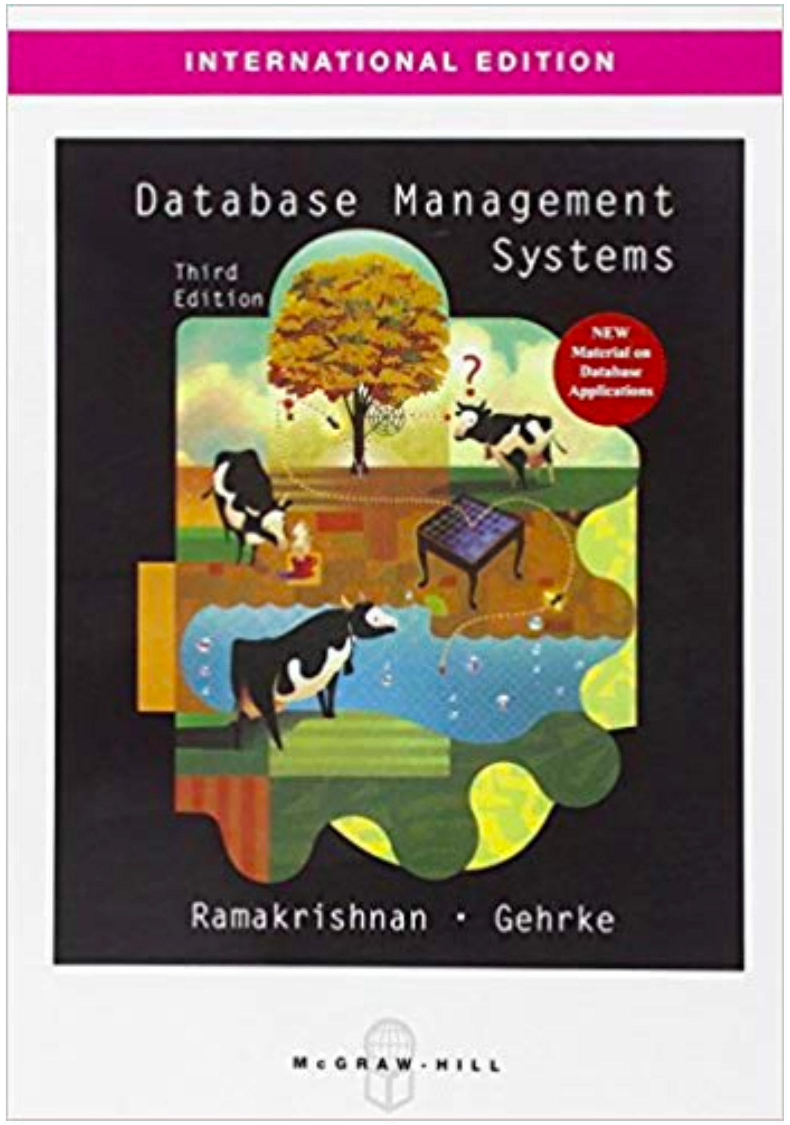
neue Modellierungsthemen

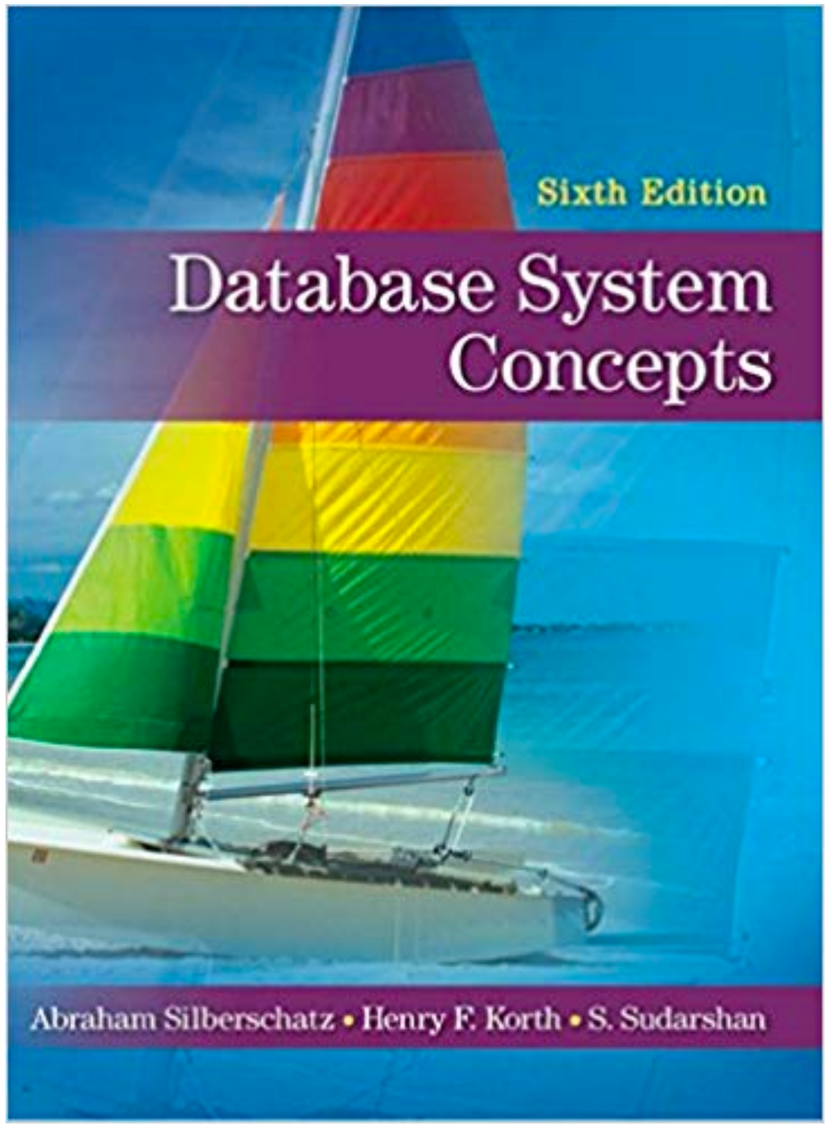
Literature

The lecture slides should be sufficient to master the contents of this course

Additionally, these **Database** textbooks may be helpful:



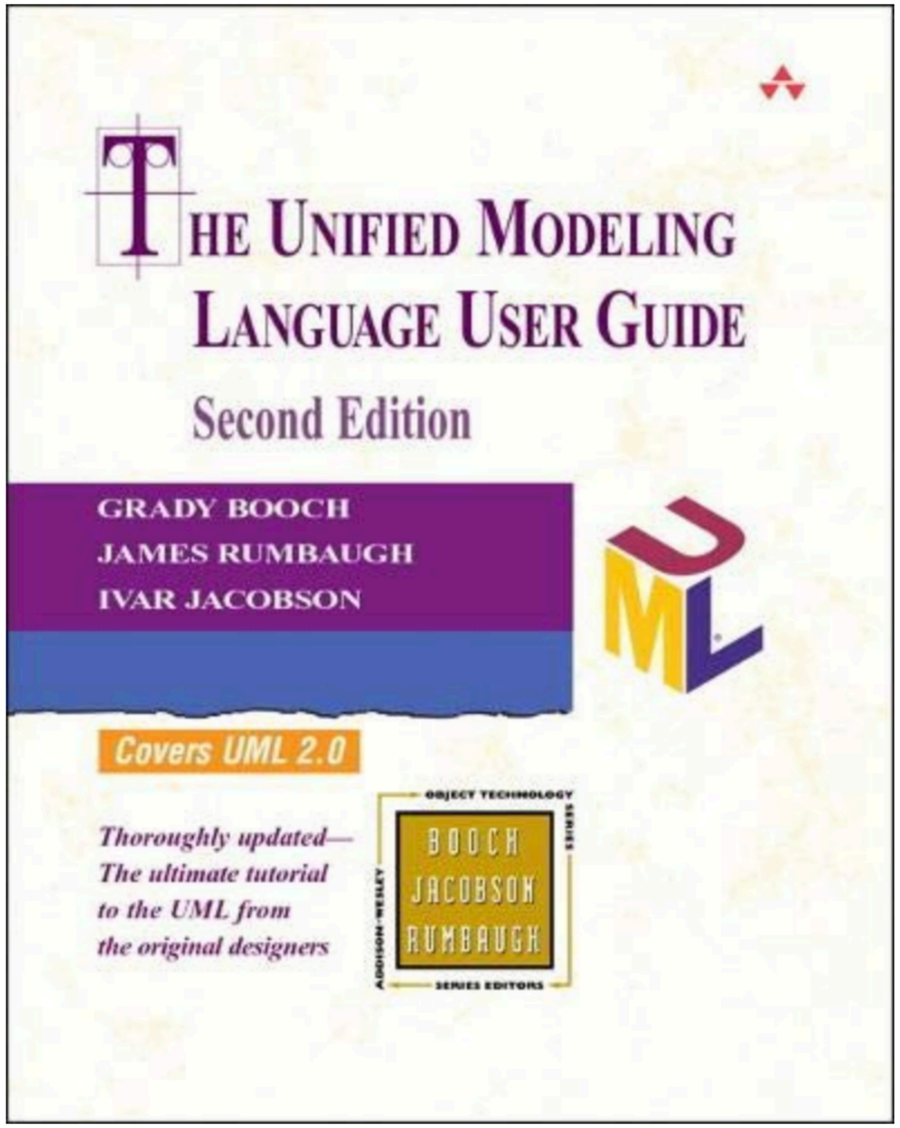


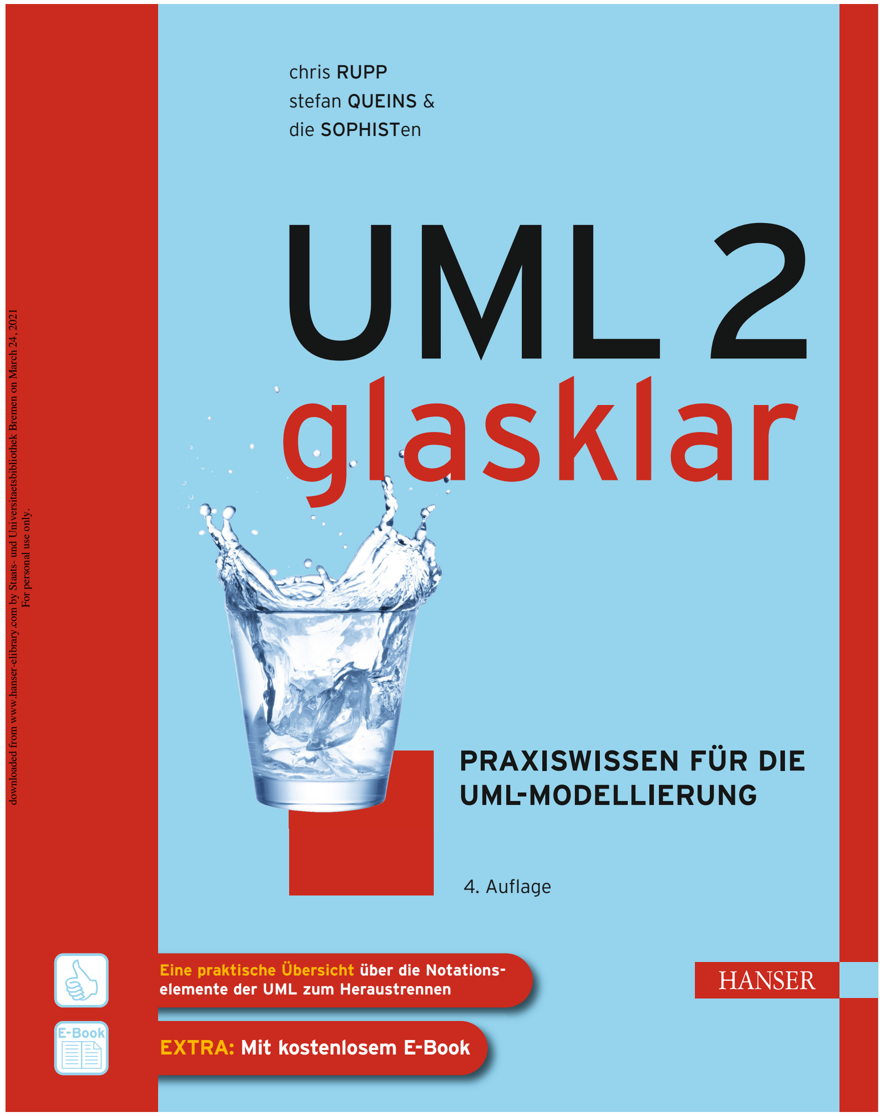


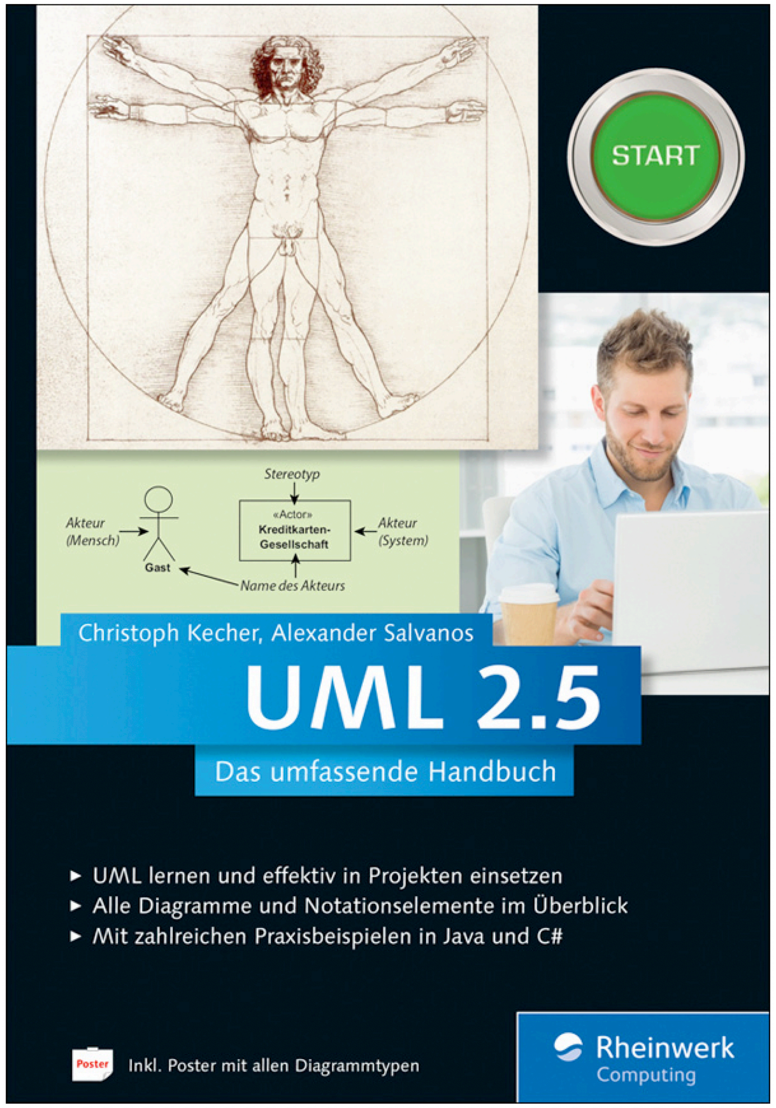
Literature

The lecture slides should be sufficient to master the contents of this course

Additionally, these **Modeling** textbooks may be helpful:




chris RUPP
stefan QUEINS &
die SOPHISTen
UML 2
glasklar
PRAXISWISSEN FÜR DIE UML-MODELLIERUNG
4. Auflage
Eine praktische Übersicht über die Notationselemente der UML zum Heraustrennen
EXTRA: Mit kostenlosem E-Book
E-Book
HANSER
downloaded from www.hanser-elibrary.com by Staats- und Universitaetsbibliothek Bremen on March 24, 2021
For personal use only.


START
Stereotyp
Akteur (Mensch)
«Actor» Kreditkarten-Gesellschaft
Akteur (System)
Gast
Name des Akteurs
Christoph Kecher, Alexander Salvanos
UML 2.5
Das umfassende Handbuch
UML lernen und effektiv in Projekten einsetzen
Alle Diagramme und Notationselemente im Überblick
Mit zahlreichen Praxisbeispielen in Java und C#
Poster
Inkl. Poster mit allen Diagrammtypen
Rheinwerk Computing

Contents of this Lecture

You will learn

— how to **model data**, esp. in a **relational way**

— how to **store data** in a **relational database management system (RDBMS)**

— how to **express queries**, using **SQL**

— how to express and **enforce integrity constraints** about the data

Contents of this Lecture

You will learn

— how to **model data**, esp. in a **relational way**

— how to **store data** in a **relational database management system (RDBMS)**

— how to **express queries**, using **SQL**

— how to express and **enforce integrity constraints** about the data

— how to **model software**, esp. in an **object oriented way**

— how to (hierarchically) **relate classes**, using **class diagrams**

— how to **model time relationships** using **state charts** and **sequence diagrams**

Today’s Lecture

Today’s Lecture

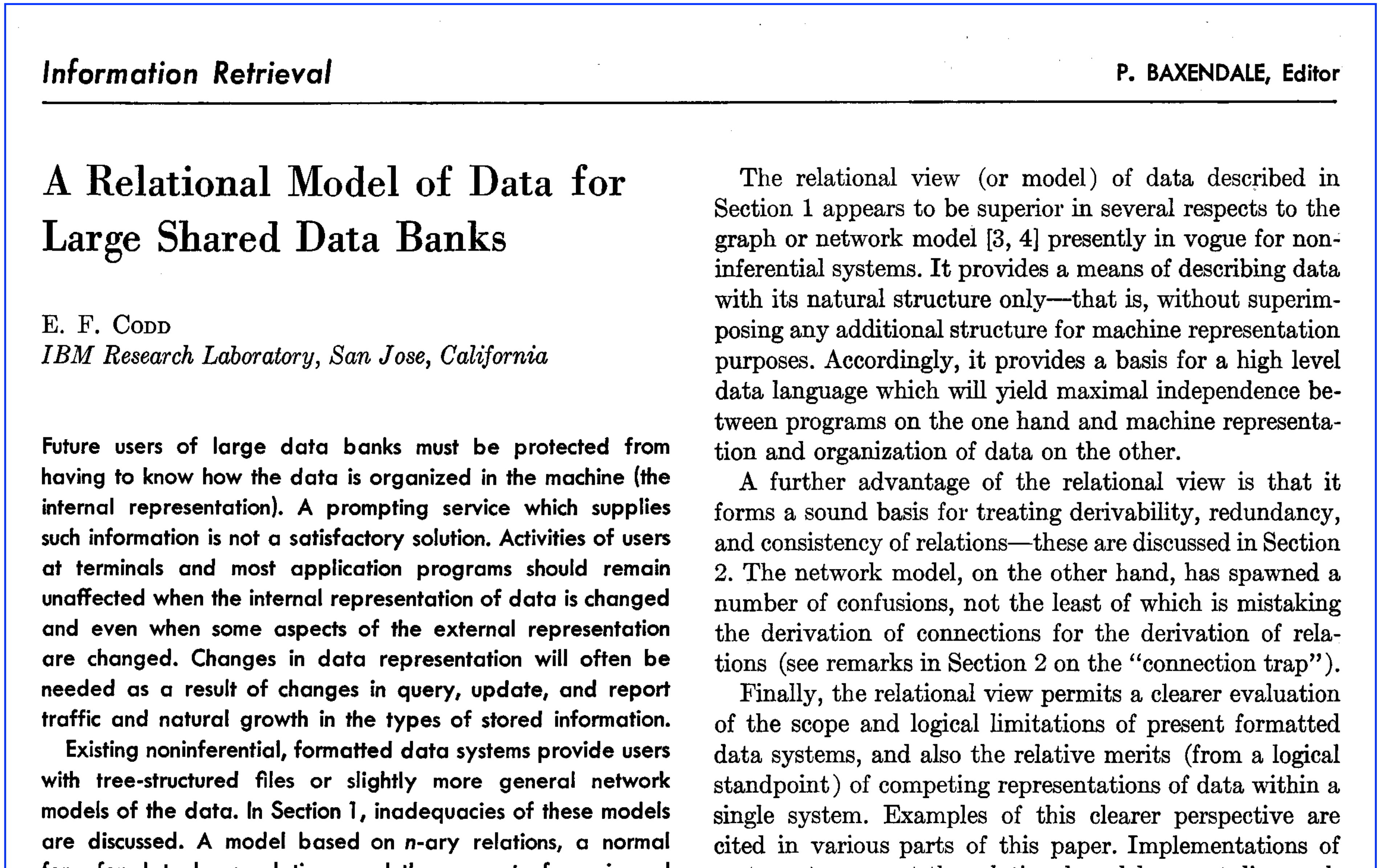
A Quick Introduction to Database Systems

What is so special about RDBMS?

Relational Database Management System (RDBMS)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in **1970**
—> **decouple form of data from physical storage!**



Information Retrieval **P. BAXENDALE, Editor**

A Relational Model of Data for Large Shared Data Banks

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

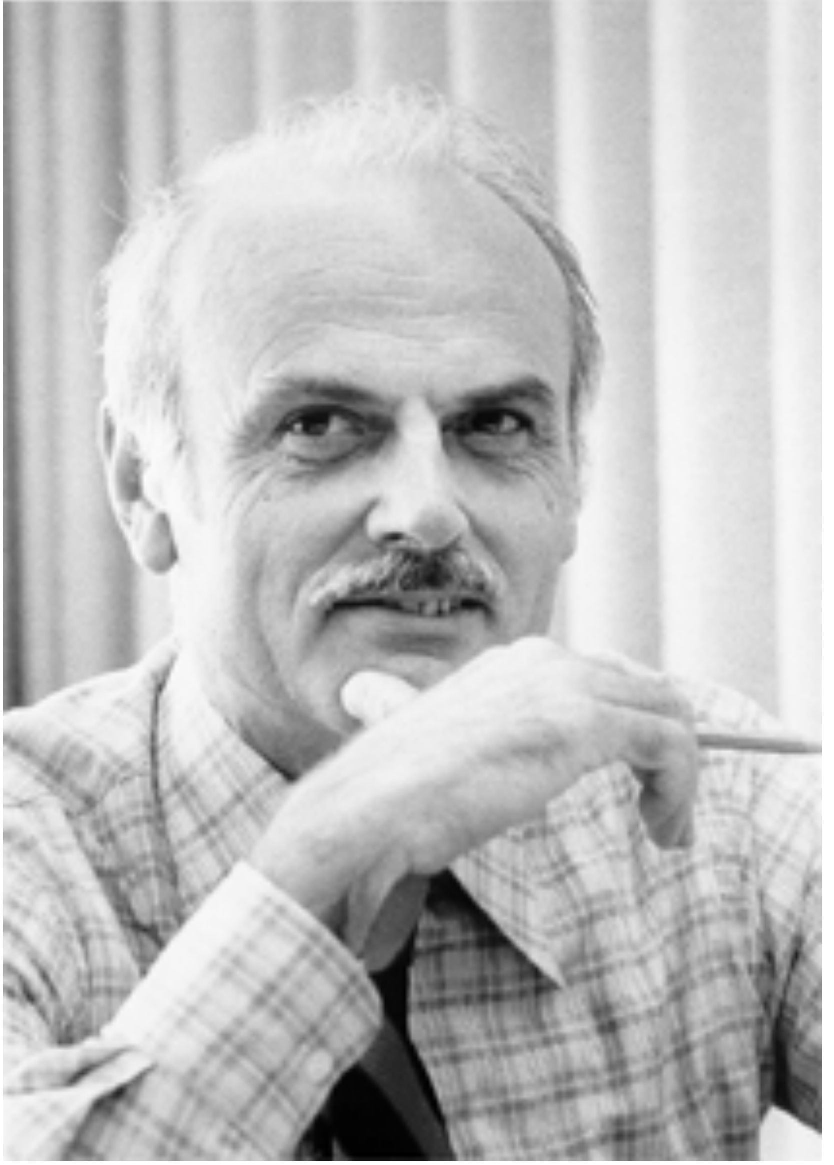
Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal

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 non-inferential 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 number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of



Edgar F. Codd
Turing Award, 1981

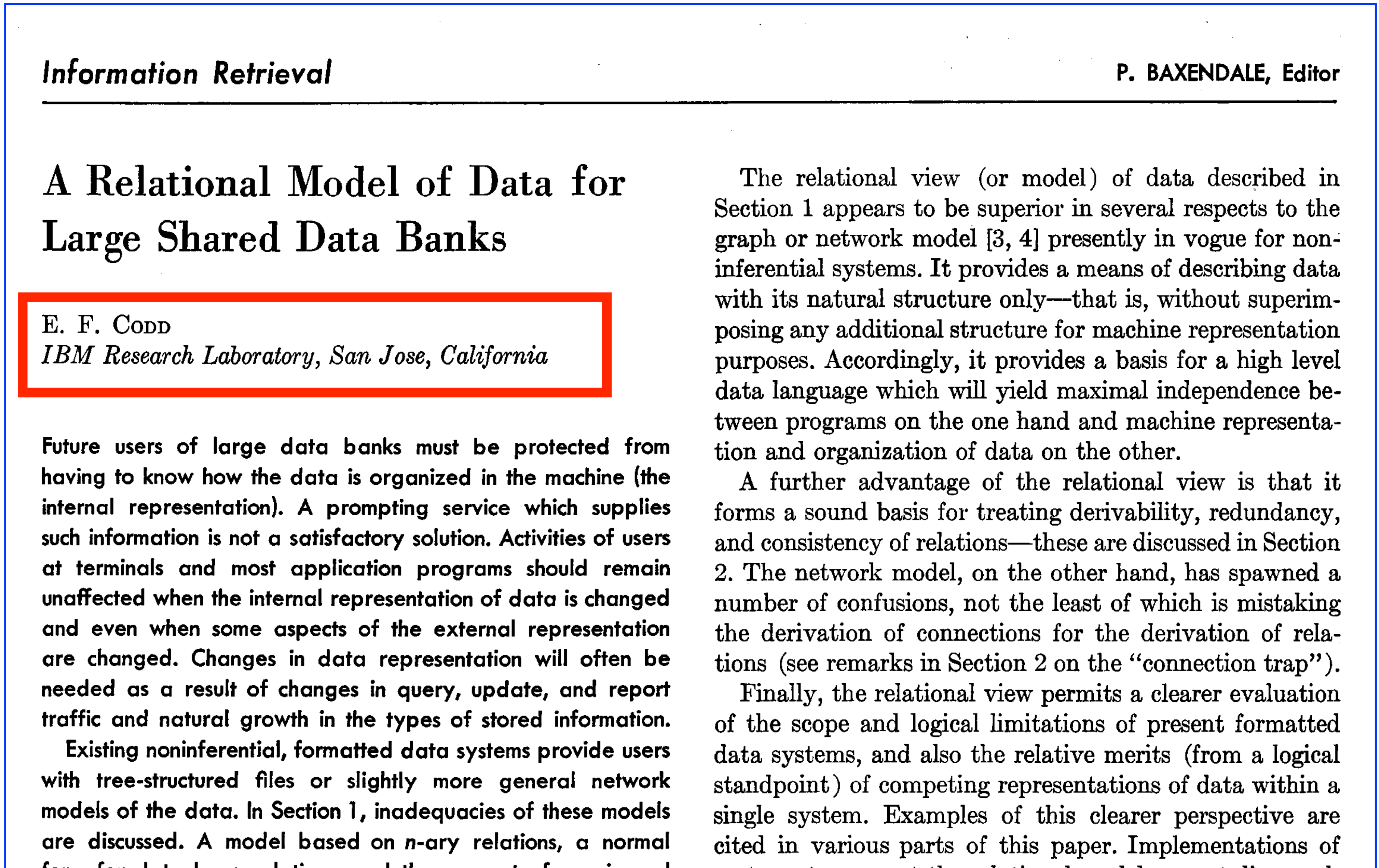
Commun. ACM 13(6):
pages 377–387 (**1970**)

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in **1970**
—> **decouple form of data from physical storage!**



Information Retrieval **P. BAXENDALE, Editor**

A Relational Model of Data for Large Shared Data Banks

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

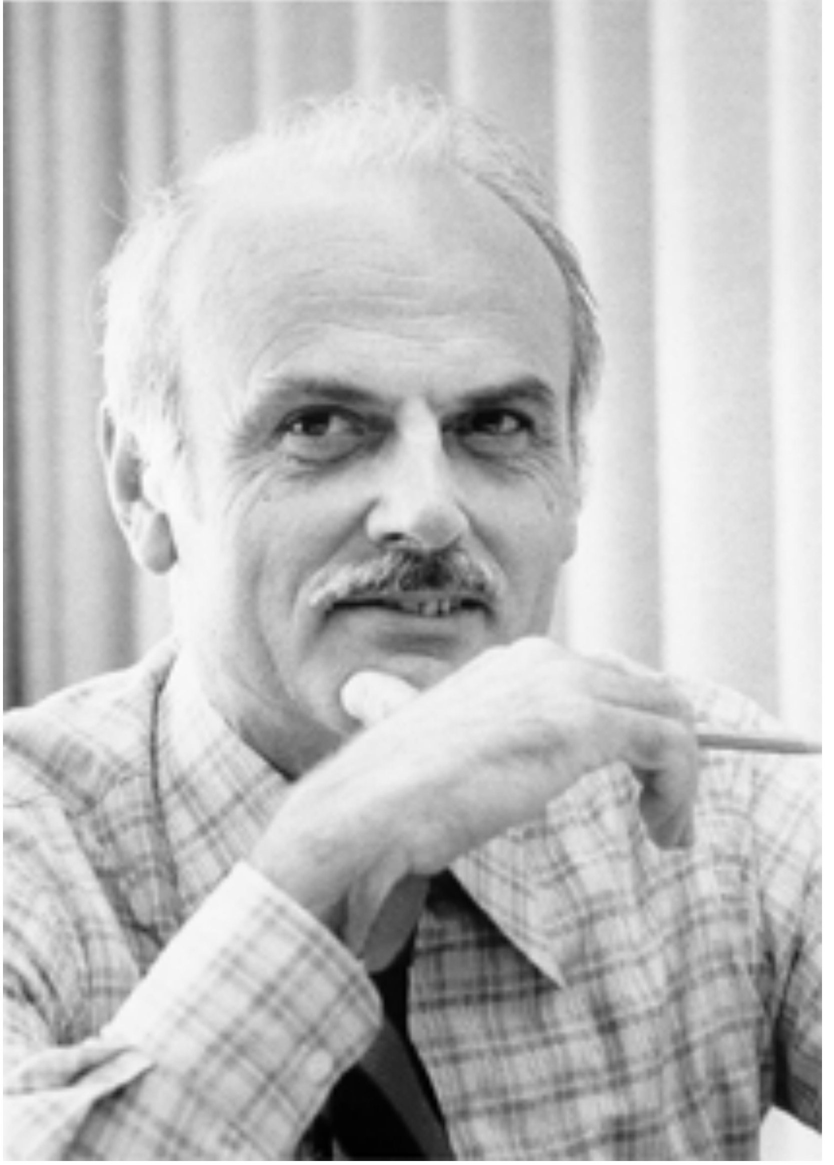
Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal

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 non-inferential 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 number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of



Edgar F. Codd
Turing Award, 1981

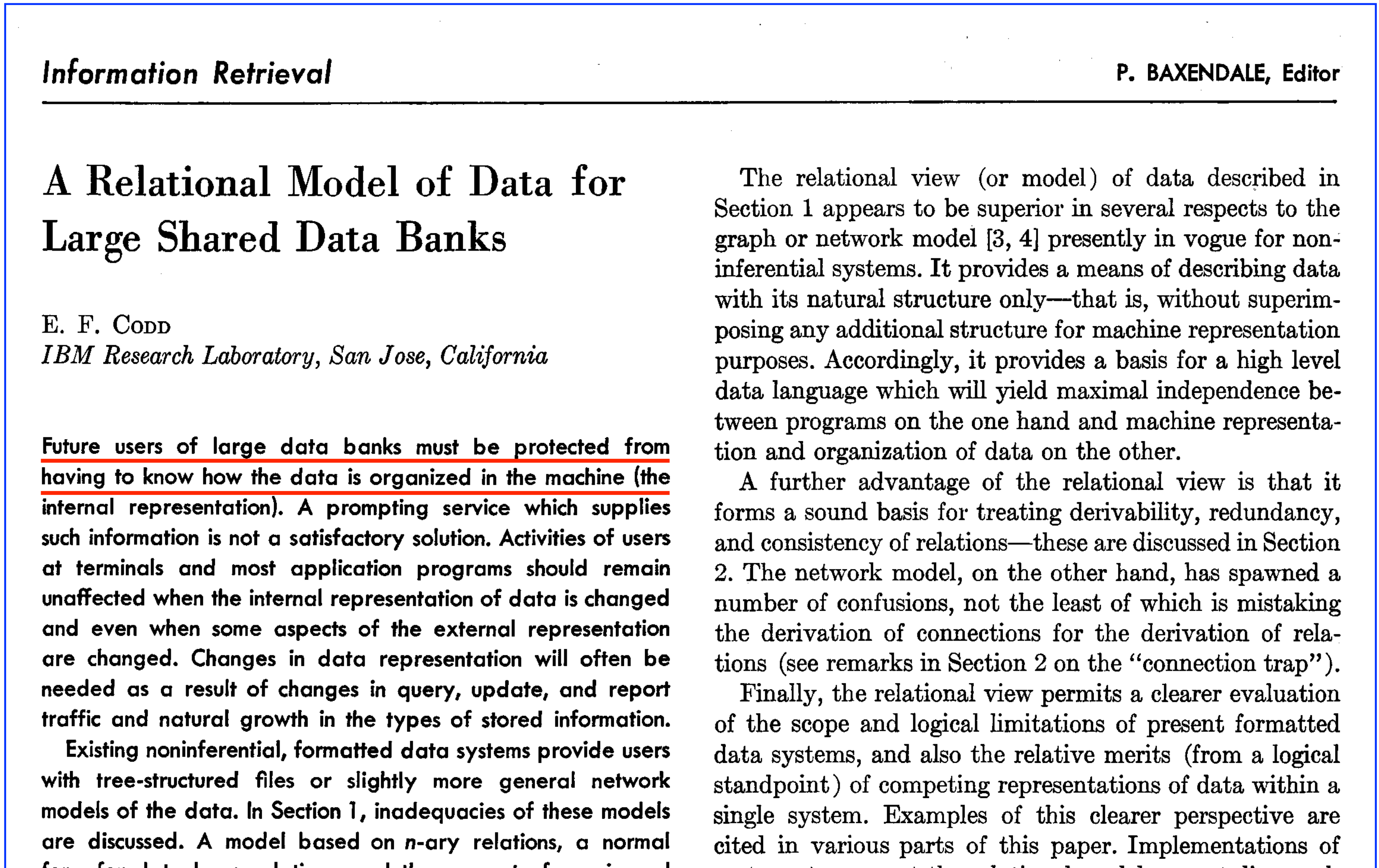
Commun. ACM 13(6): pages 377–387 (**1970**)

What is so special about RDBMS?

Relational Database Management System (RDBMS)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in **1970**
—> **decouple form of data from physical storage!**



Information Retrieval **P. BAXENDALE, Editor**

A Relational Model of Data for Large Shared Data Banks

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

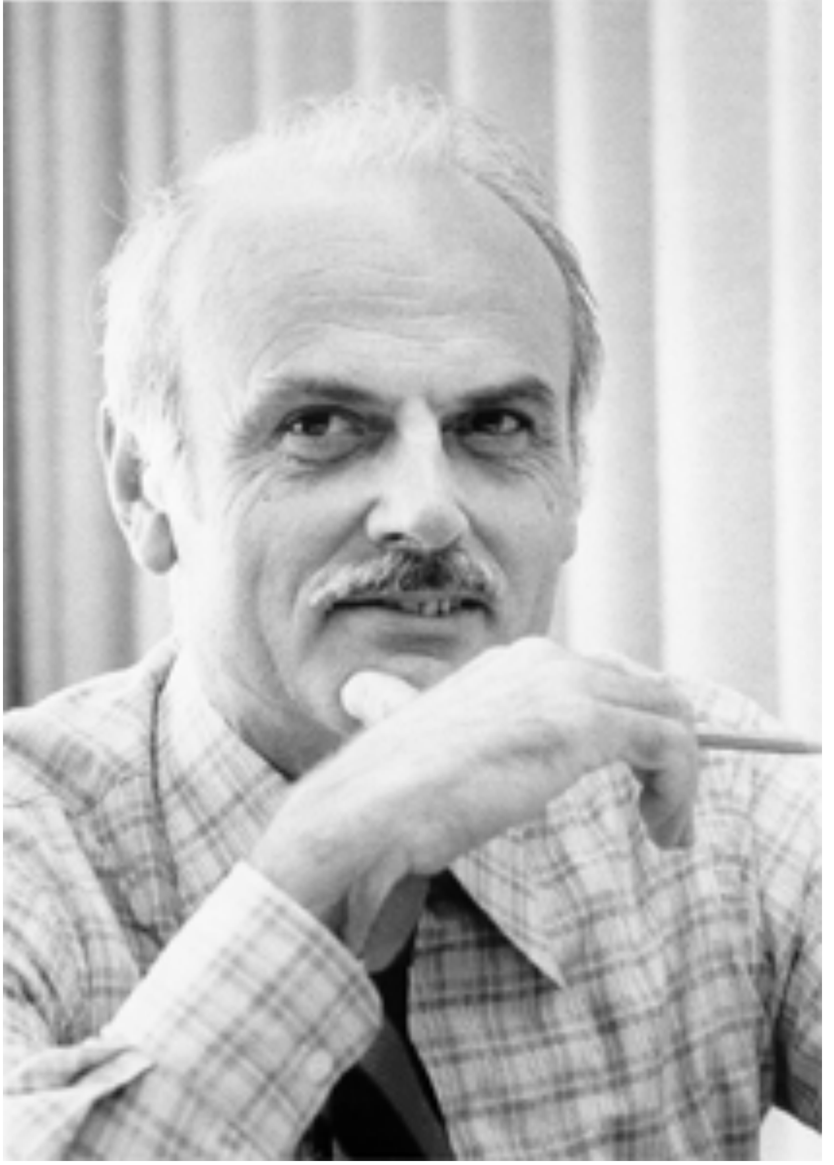
Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal

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 non-inferential 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 number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of



Edgar F. Codd
Turing Award, 1981

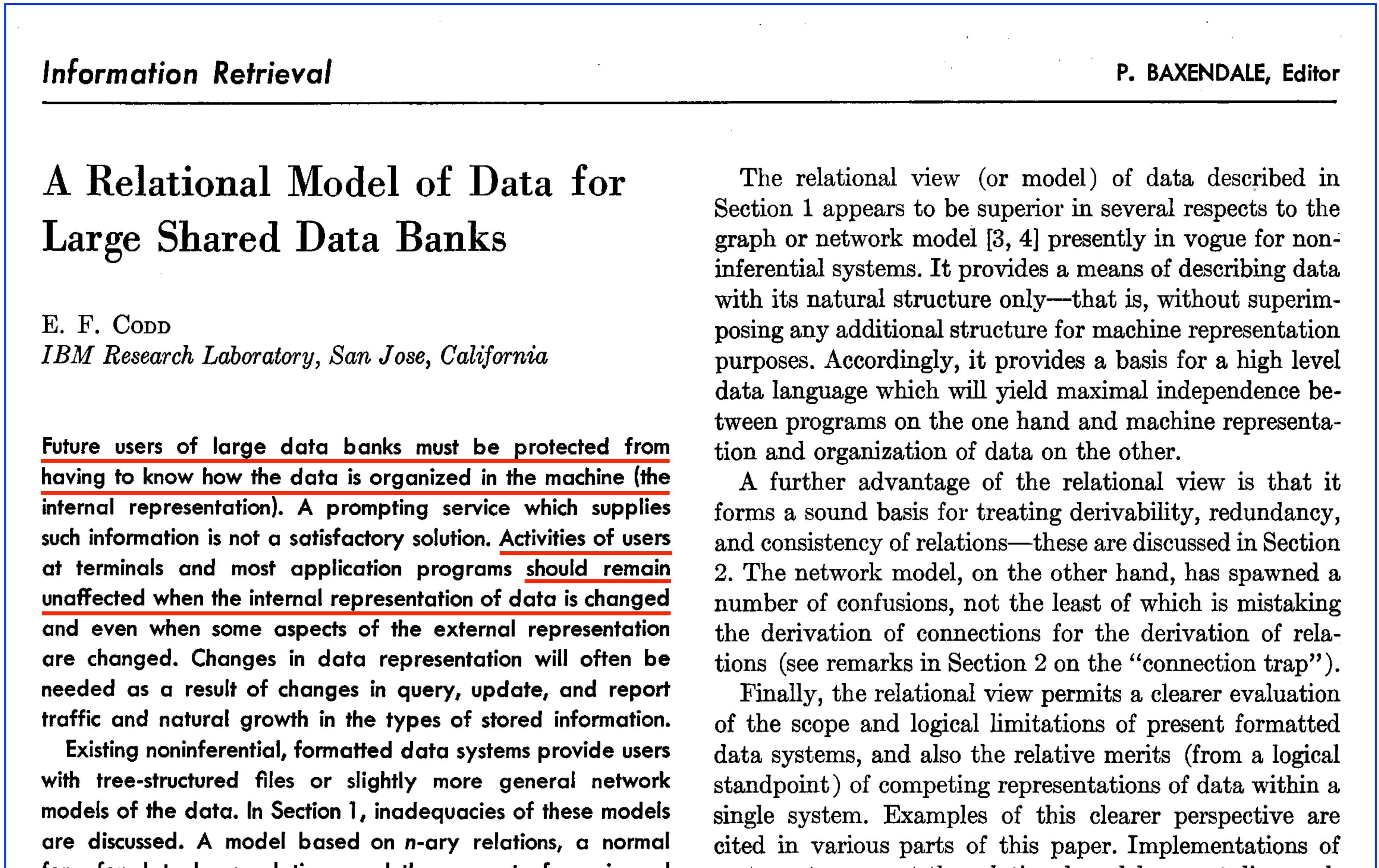
Commun. ACM 13(6): pages 377–387 (**1970**)

What is so special about RDBMS?

Relational Database Management System (RDBMS)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in **1970**
—> **decouple form of data from physical storage!**



Information Retrieval **P. BAXENDALE, Editor**

A Relational Model of Data for Large Shared Data Banks

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

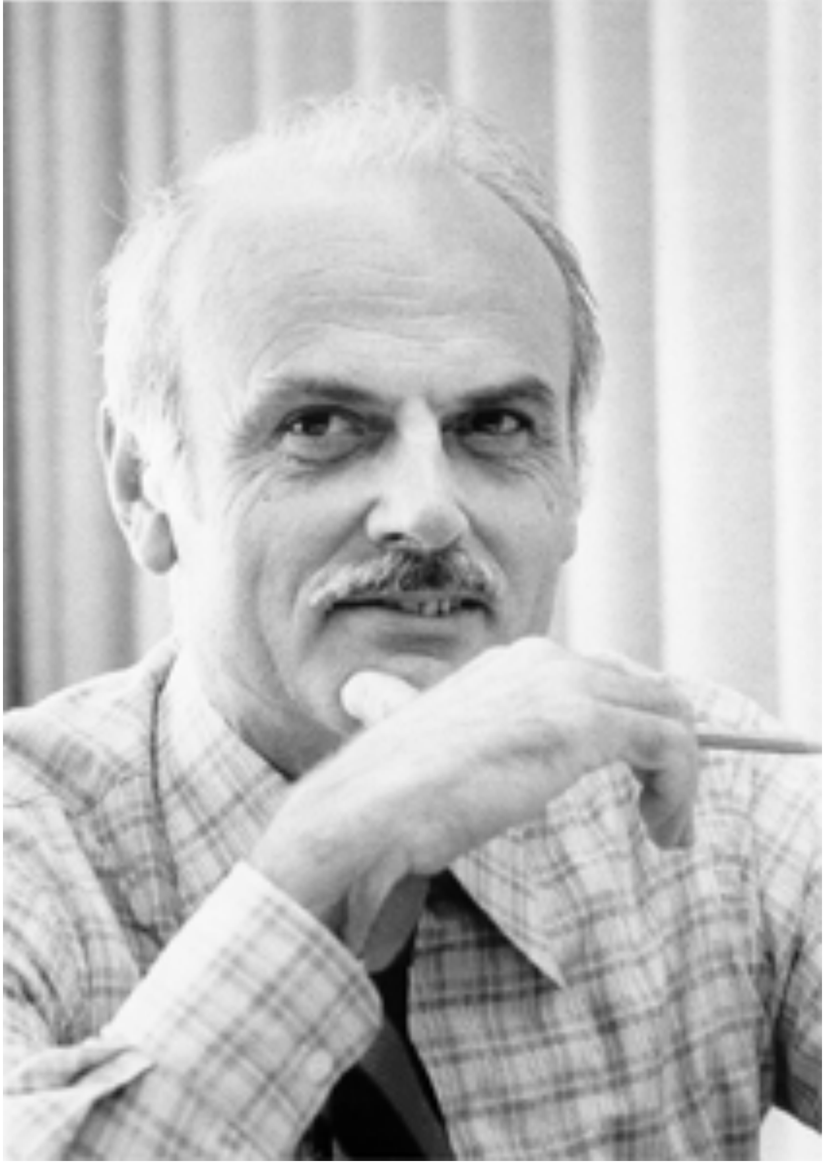
Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal

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 non-inferential 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 number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the "connection trap").

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of



Edgar F. Codd
Turing Award, 1981

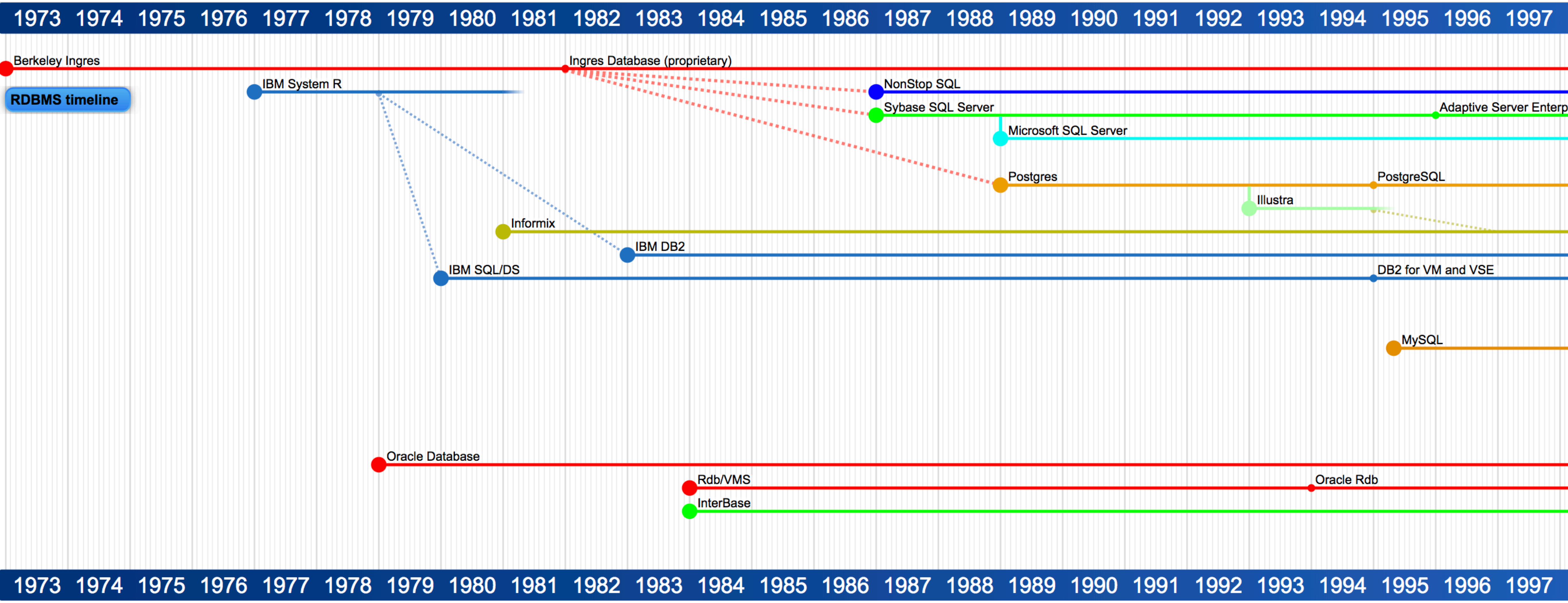
Commun. ACM 13(6): pages 377–387 (**1970**)

What is so special about RDBMS?

Relational Database Management System (RDBMS)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in 1970
—> **decouple form of data from physical storage!**



What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!

“relational database” — coined by Edgar F. Codd in 1970
—> **decouple form of data from physical storage!**

— Oracle, IBM, Microsoft: roots of their RDBMSs are from the 1970s!

— IBM Db2: *free* version for Win / Linux!
(but has size limitations)

— Other powerful *free* RDBMS available,
e.g., PostgreSQL, MySQL

Less powerful: SQLite3

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!

"relational database" — coined by Edgar F. Codd in 1970

—> **decouple form of data from physical storage!**

— Oracle, IBM, Microsoft: roots of their RDBMSs are from the 1970s!

— IBM Db2: *free* version for Win / Linux!
(but has size limitations)

— Other powerful *free* RDBMS available,
e.g., PostgreSQL, MySQL

Less powerful: SQLite3

Poll: have you used any of these?

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

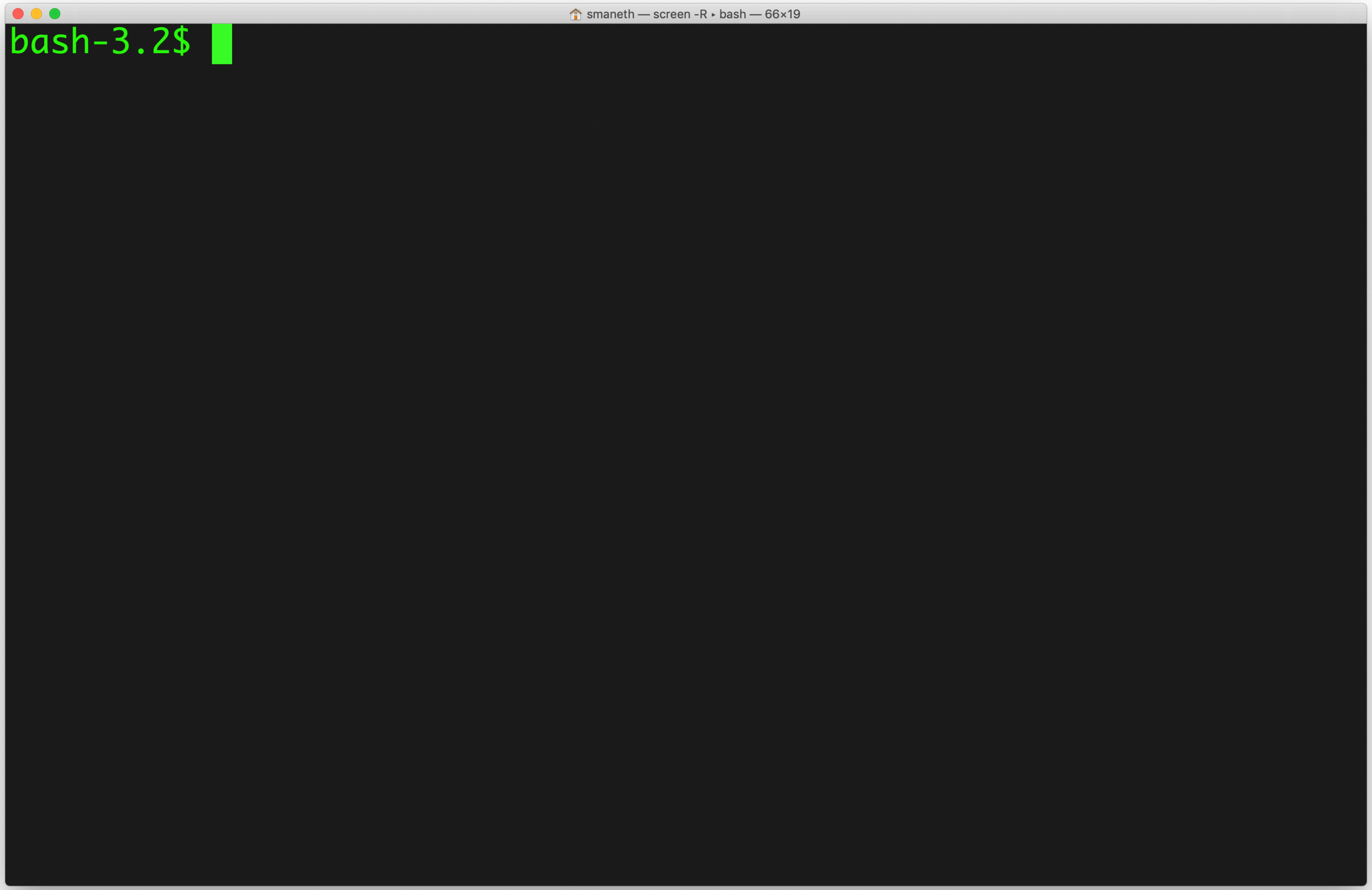
— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

— install SSH on your computer / phone
and SSH into any informatics Linux machine
(e.g., x01, x02, etc)



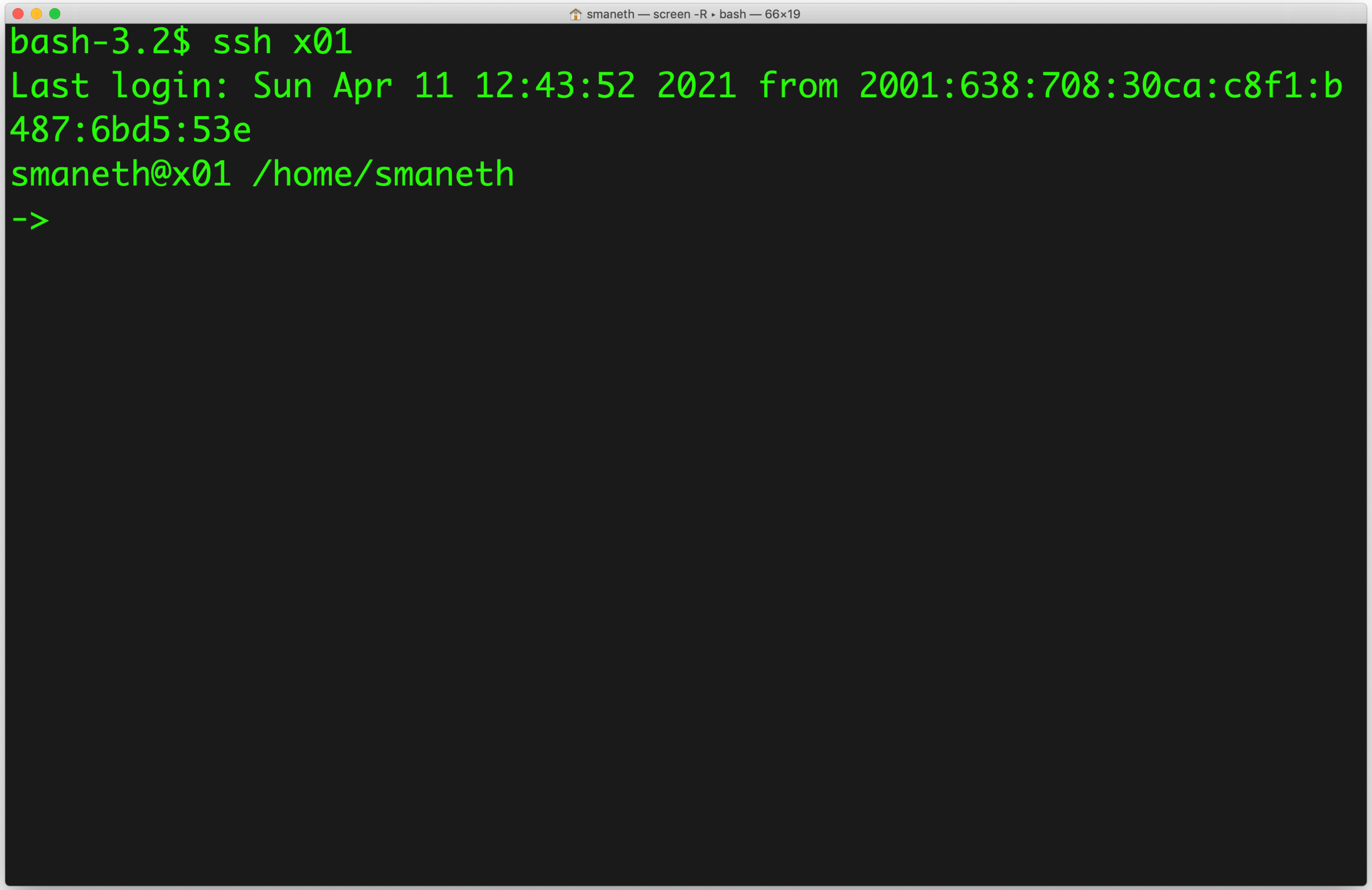
```
bash-3.2$ 
```

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

— install SSH on your computer / phone
and SSH into any informatics Linux machine
(e.g., x01, x02, etc)



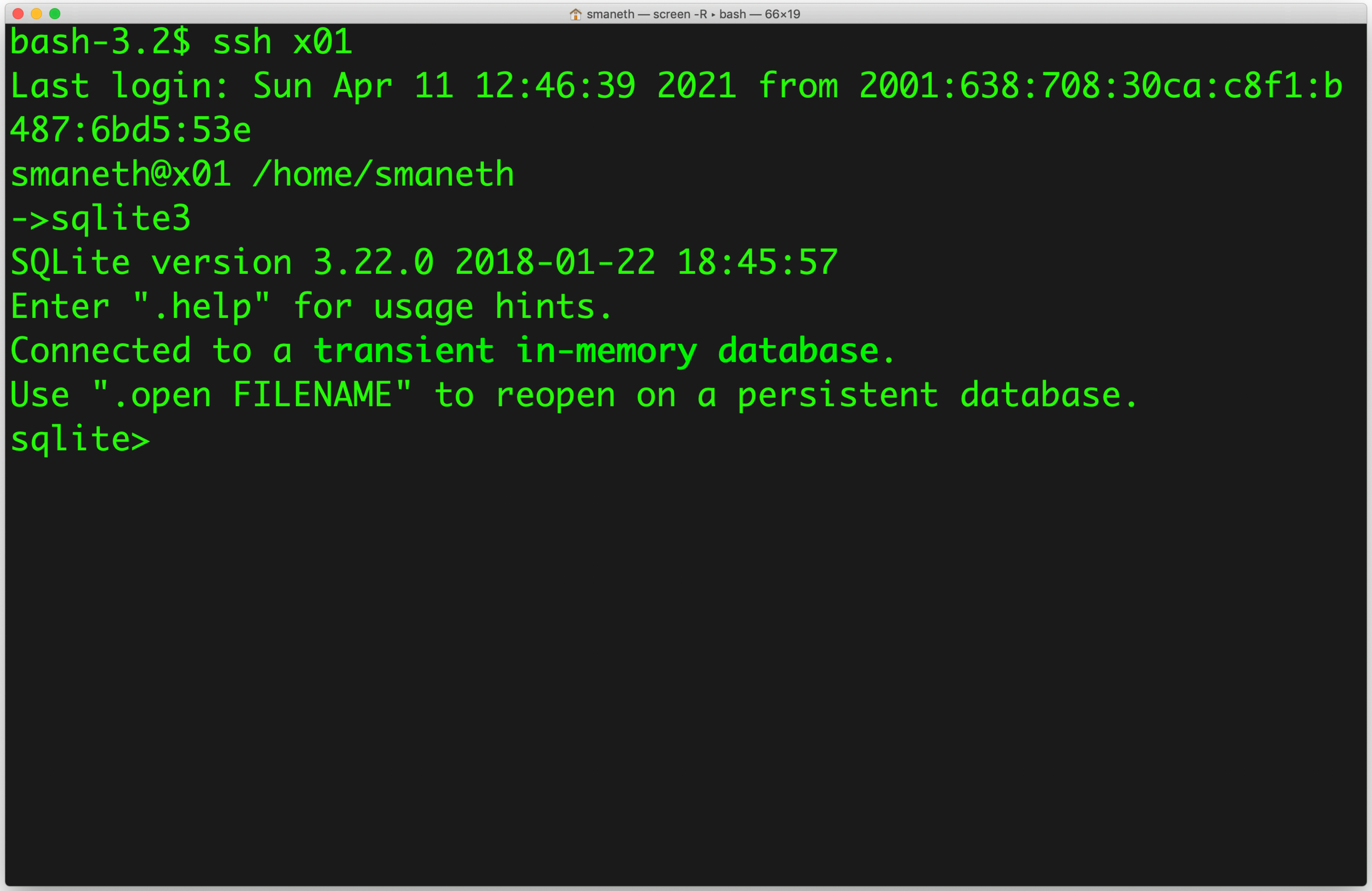
```
bash-3.2$ ssh x01
Last login: Sun Apr 11 12:43:52 2021 from 2001:638:708:30ca:c8f1:b
487:6bd5:53e
smaneth@x01 /home/smaneth
->
```

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

— install SSH on your computer / phone
and SSH into any informatics Linux machine
(e.g., x01, x02, etc)



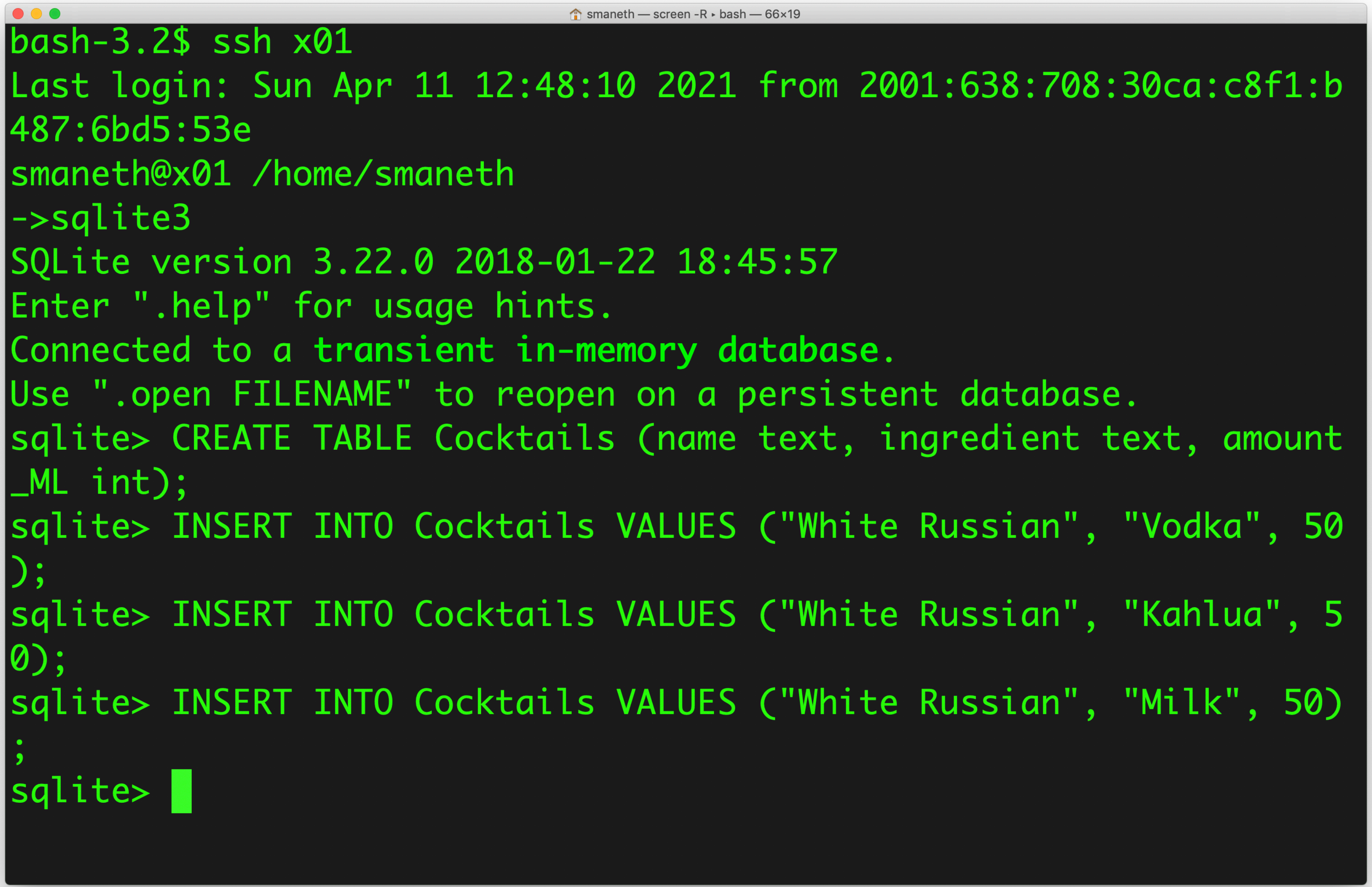
```
bash-3.2$ ssh x01
Last login: Sun Apr 11 12:46:39 2021 from 2001:638:708:30ca:c8f1:b
487:6bd5:53e
smaneth@x01 /home/smaneth
->sqlite3
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

— install SSH on your computer / phone
and SSH into any informatics Linux machine
(e.g., x01, x02, etc)



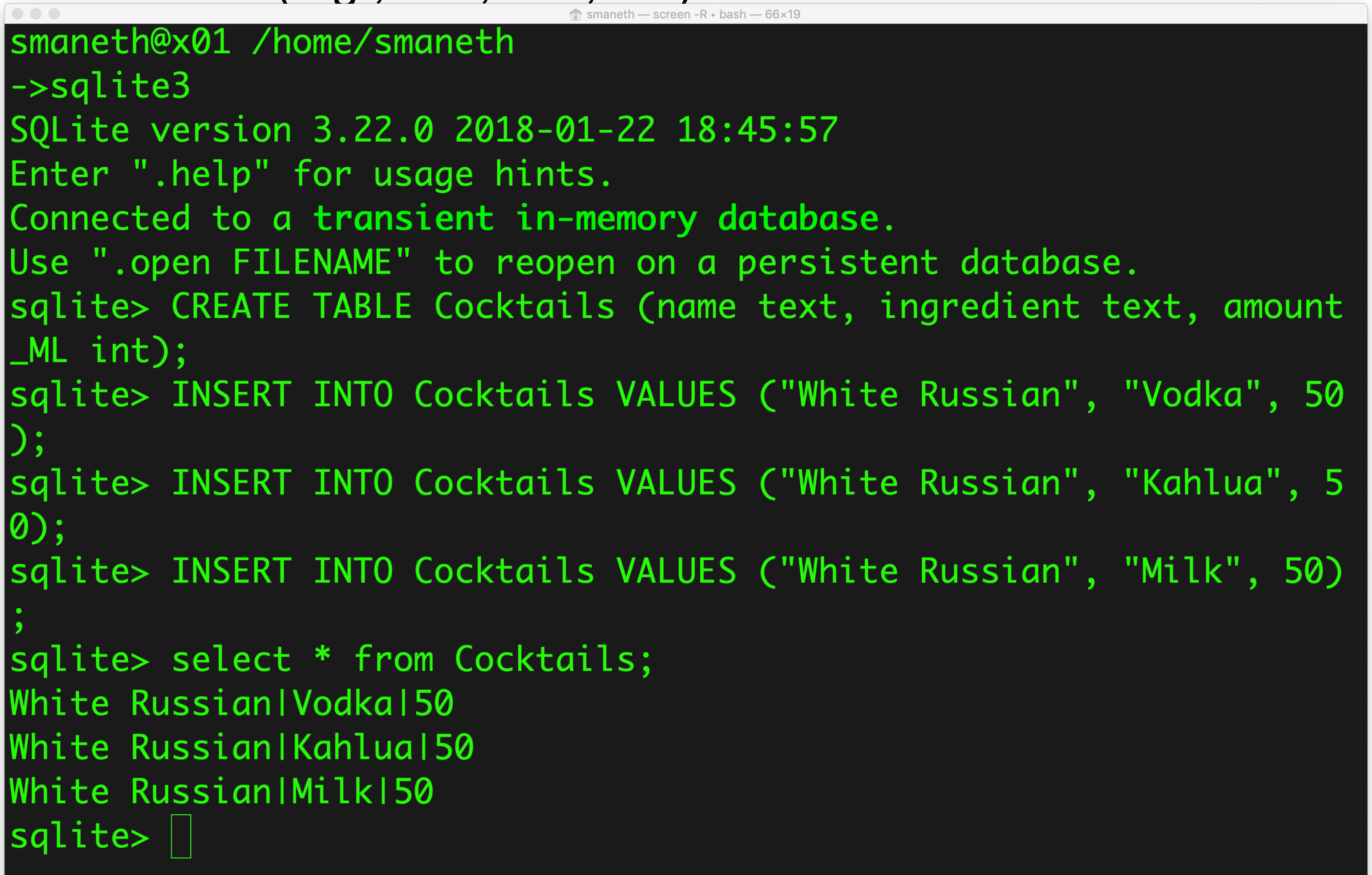
```
bash-3.2$ ssh x01
Last login: Sun Apr 11 12:48:10 2021 from 2001:638:708:30ca:c8f1:b
487:6bd5:53e
smaneth@x01 /home/smaneth
->sqlite3
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> CREATE TABLE Cocktails (name text, ingredient text, amount
_ML int);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Vodka", 50
);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Kahlua", 5
0);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Milk", 50)
;
sqlite> 
```

Note:

All of you should be able to run SQlite3
(needed for some Exercises!)

— install it on your computer
(available for Android, Linux, Mac OS, Windows, etc)

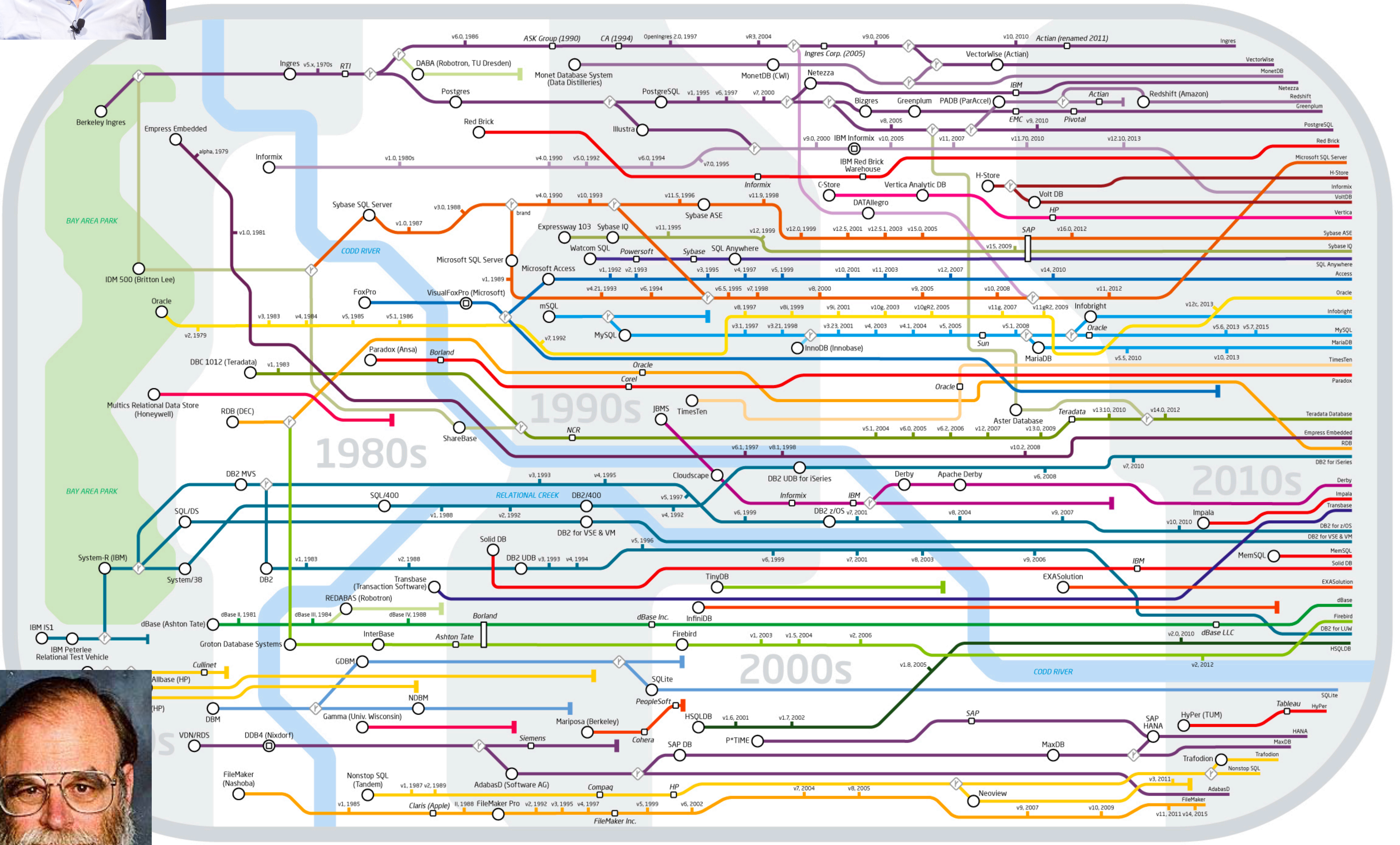
— install SSH on your computer / phone
and SSH into any informatics Linux machine
(e.g., x01, x02, etc)



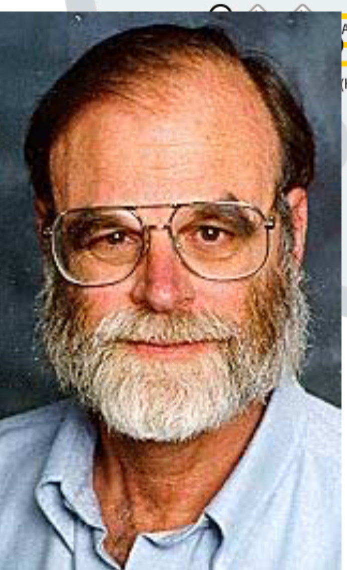
```
smaneth@x01 /home/smaneth
->sqlite3
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> CREATE TABLE Cocktails (name text, ingredient text, amount
_ML int);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Vodka", 50
);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Kahlua", 5
0);
sqlite> INSERT INTO Cocktails VALUES ("White Russian", "Milk", 50)
;
sqlite> select * from Cocktails;
White Russian|Vodka|50
White Russian|Kahlua|50
White Russian|Milk|50
sqlite> 
```

Michael Stonebreaker
Turing Award 2014

Genealogy of Relational Database Management Systems



Felix Naumann, Jana Bauckmann, Claudia Exeler, Jan-Peer Rudolph, Fabian Tschirschnitz
Contact - Hasso Plattner Institut, University of Potsdam, felix.naumann@hpi.de
Design - Alexander Sandt Grafik-Design, Hamburg
Version 6.0 - October 2018
https://hpi.de/naumann/projects/rdbms-genealogy.html

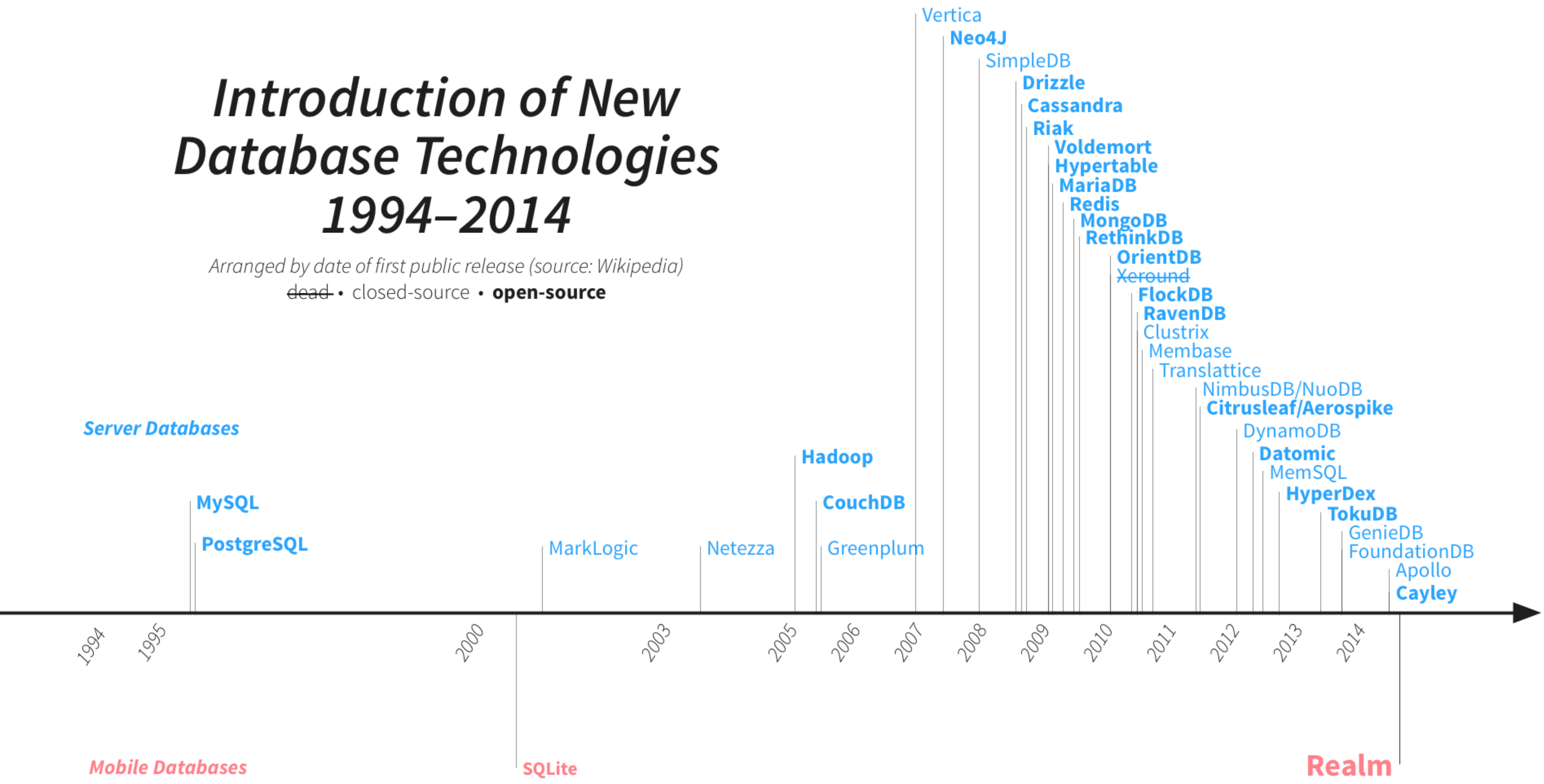


James Gray (disappeard 2007)
Turing Award 1998

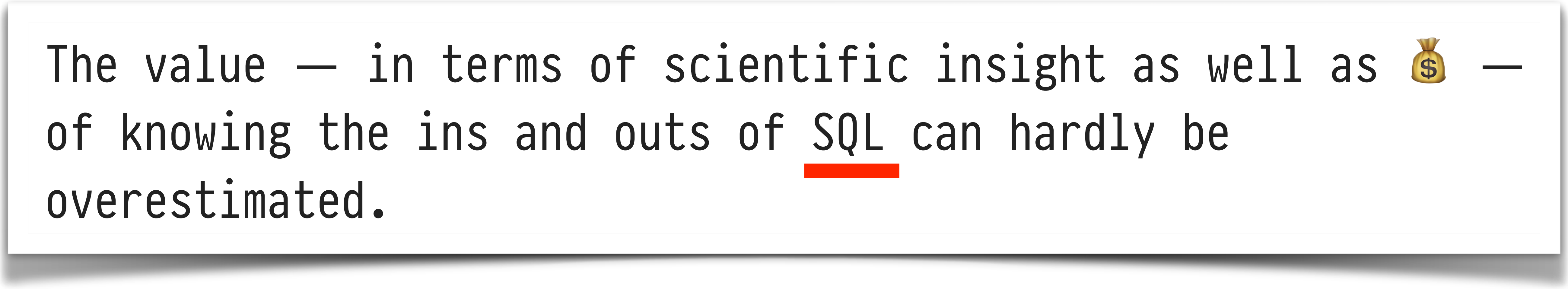
Introduction of New Database Technologies 1994–2014

Arranged by date of first public release (source: Wikipedia)

~~dead~~ • closed-source • **open-source**

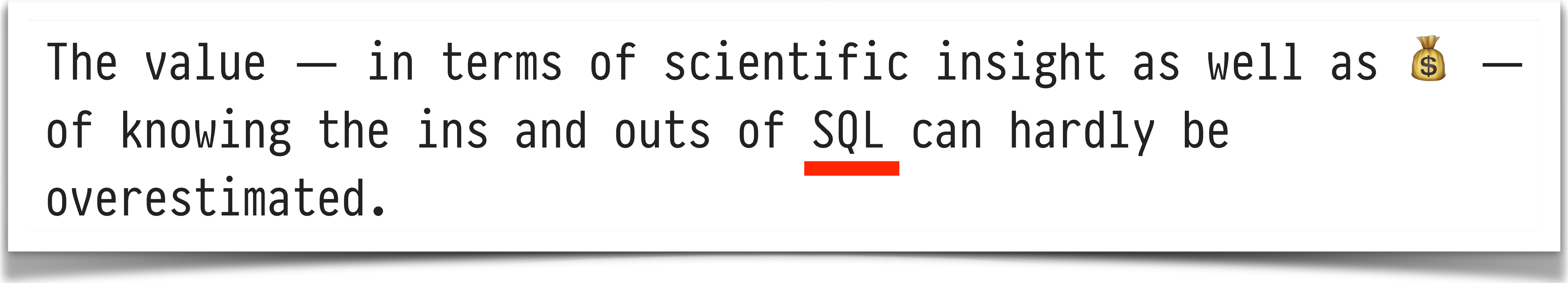


Thorsten Grust (Univ. Tübingen)
Lecture “Advanced SQL”
(from lecture 1):



The value — in terms of scientific insight as well as 💰 — of knowing the ins and outs of SQL can hardly be overestimated.

Thorsten Grust (Univ. Tübingen)
Lecture “Advanced SQL”
(from lecture 1):

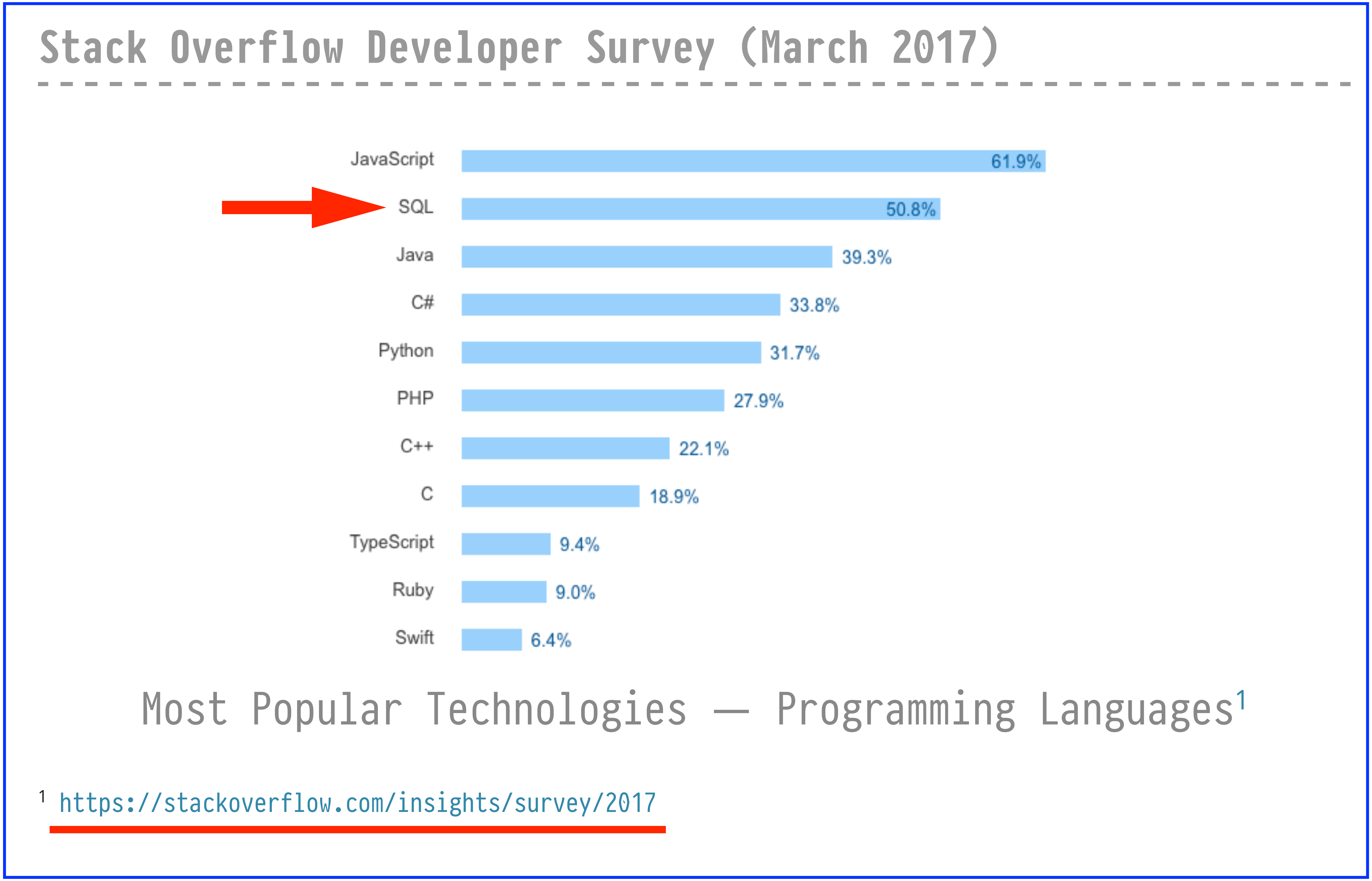


The value — in terms of scientific insight as well as 💰 — of knowing the ins and outs of SQL can hardly be overestimated.

From Wikipedea (accessed 15.10.2019):

“In 2018, Oracle was the third-largest software company by revenue.”

Thorsten Grust (Univ. Tübingen)
Lecture “Advanced SQL” (from lecture 1):



Stack Overflow Developer Survey (March 2017)

Most Popular Technologies — Programming Languages[1]

[1] https://stackoverflow.com/insights/survey/2017

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!

2. Despite all these new databases (NoSQL, graph databases, parallel, etc) **most data** today (esp. enterprise data) is stored in RDBMS

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!
2. Despite all these new databases (NoSQL, graph databases, parallel, etc) **most data** today (esp. enterprise data) is stored in RDBMS

Technically, what is special about RDBMS?

— SQL (Standard Query Language)

— Query Optimization

— Transaction Management (& crash recovery!)

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!

2. Despite all these new databases (NoSQL, graph databases, parallel, etc) **most data** today (esp. enterprise data) is stored in RDBMS

Technically, what is special about RDBMS?

— SQL (Standard Query Language) ← **Complex Queries**

— Query Optimization ← **High-Speed Execution**

— Transaction Management ← **Data Safety**

What is so special about RDBMS?

Relational Database Management System (**RDBMS**)

1. RDBMS are very mature pieces of software!
2. Despite all these new databases (NoSQL, graph databases, parallel, etc) **most data** today (esp. enterprise data) is stored in RDBMS

Technically, what is special about RDBMS?

— SQL (Standard Query Language) ← **Complex Queries**

— Query Optimization ← **High-Speed Execution**

— Transaction Management ← **Data Safety**

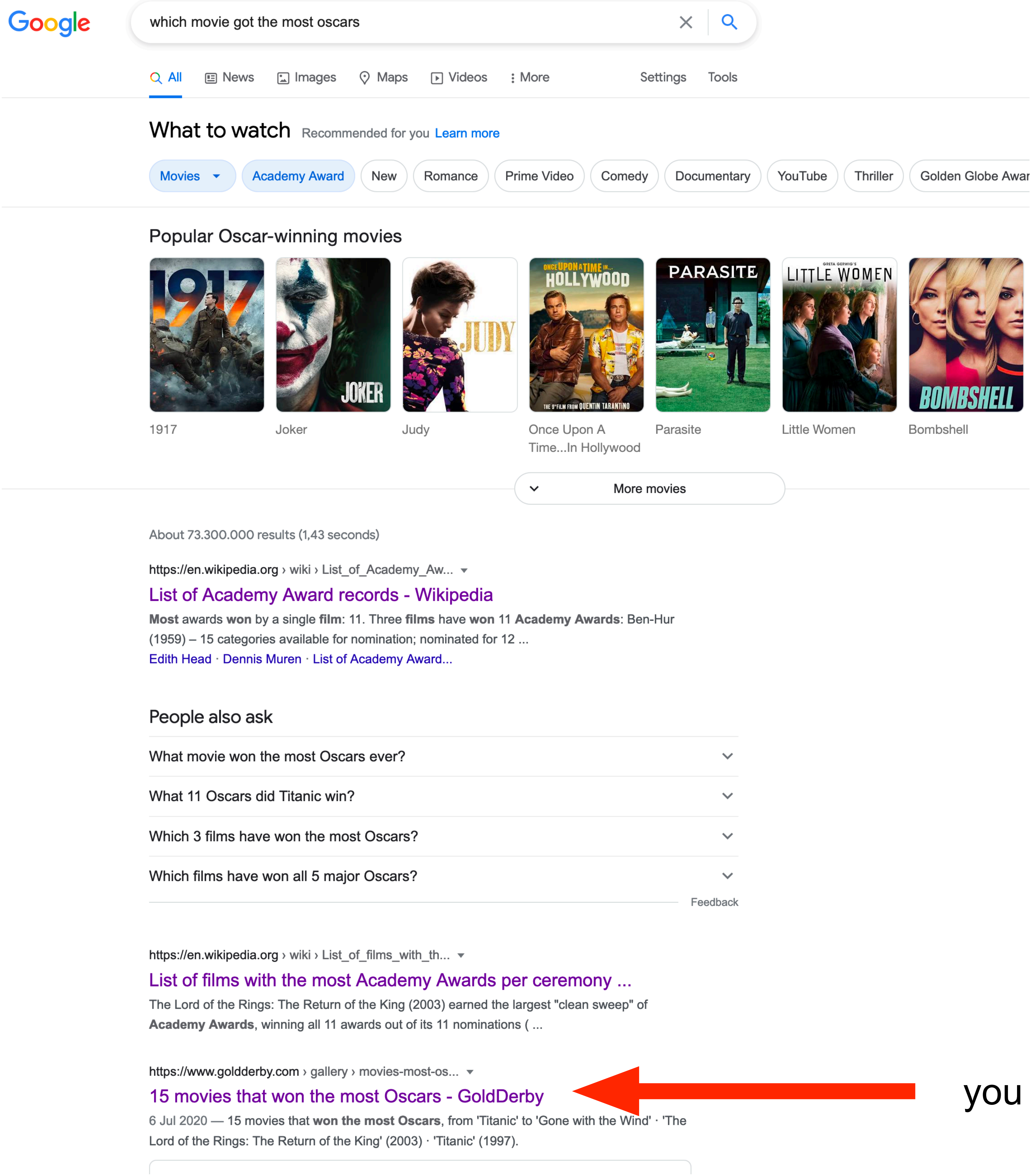
… alas, in **this course** you only learn about the first topic (SQL)

— course “**Datenbanksysteme**” teaches the other two topics.

Complex Queries
High-Speed Execution

Let’s see some examples.
Why not use Full-Text Search (a la Google) for all queries?

— which 20 movies won the most Oscars?



which movie got the most oscars

What to watch Recommended for you Learn more

Movies | Academy Award | New | Romance | Prime Video | Comedy | Documentary | YouTube | Thriller | Golden Globe Awar

Popular Oscar-winning movies

1917 | Joker | Judy | Once Upon A Time...In Hollywood | Parasite | Little Women | Bombshell

More movies

About 73.300.000 results (1,43 seconds)

https://en.wikipedia.org › wiki › List_of_Academy_Aw...

List of Academy Award records - Wikipedia

Most awards **won** by a single **film**: 11. Three **films** have **won** 11 **Academy Awards**: Ben-Hur (1959) – 15 categories available for nomination; nominated for 12 ...

Edith Head · Dennis Muren · List of Academy Award...

People also ask

What movie won the most Oscars ever?

What 11 Oscars did Titanic win?

Which 3 films have won the most Oscars?

Which films have won all 5 major Oscars?

Feedback

https://en.wikipedia.org › wiki › List_of_films_with_th...

List of films with the most Academy Awards per ceremony ...

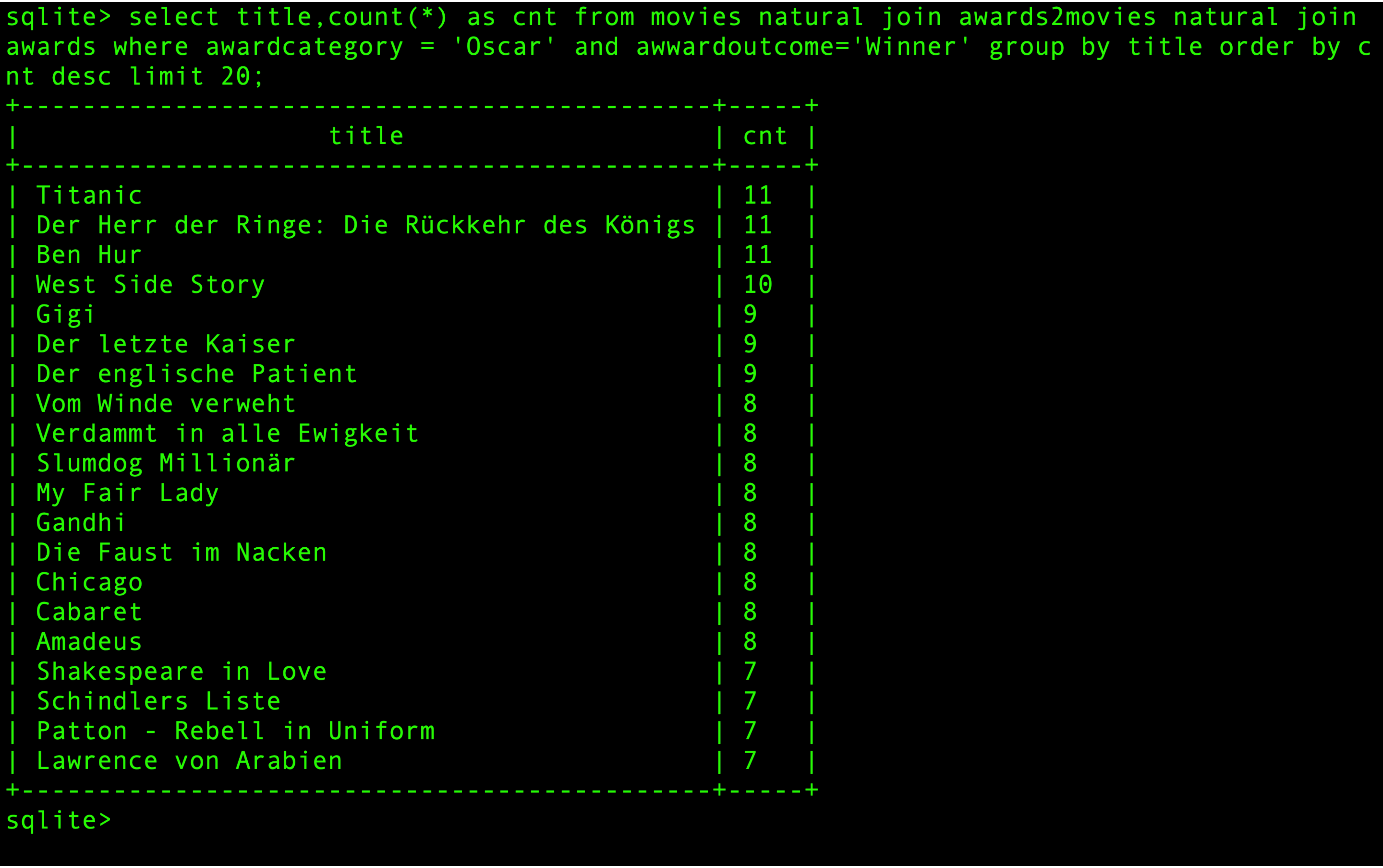
The Lord of the Rings: The Return of the King (2003) earned the largest "clean sweep" of **Academy Awards**, winning all 11 awards out of its 11 nominations (...

https://www.goldderby.com › gallery › movies-most-os...

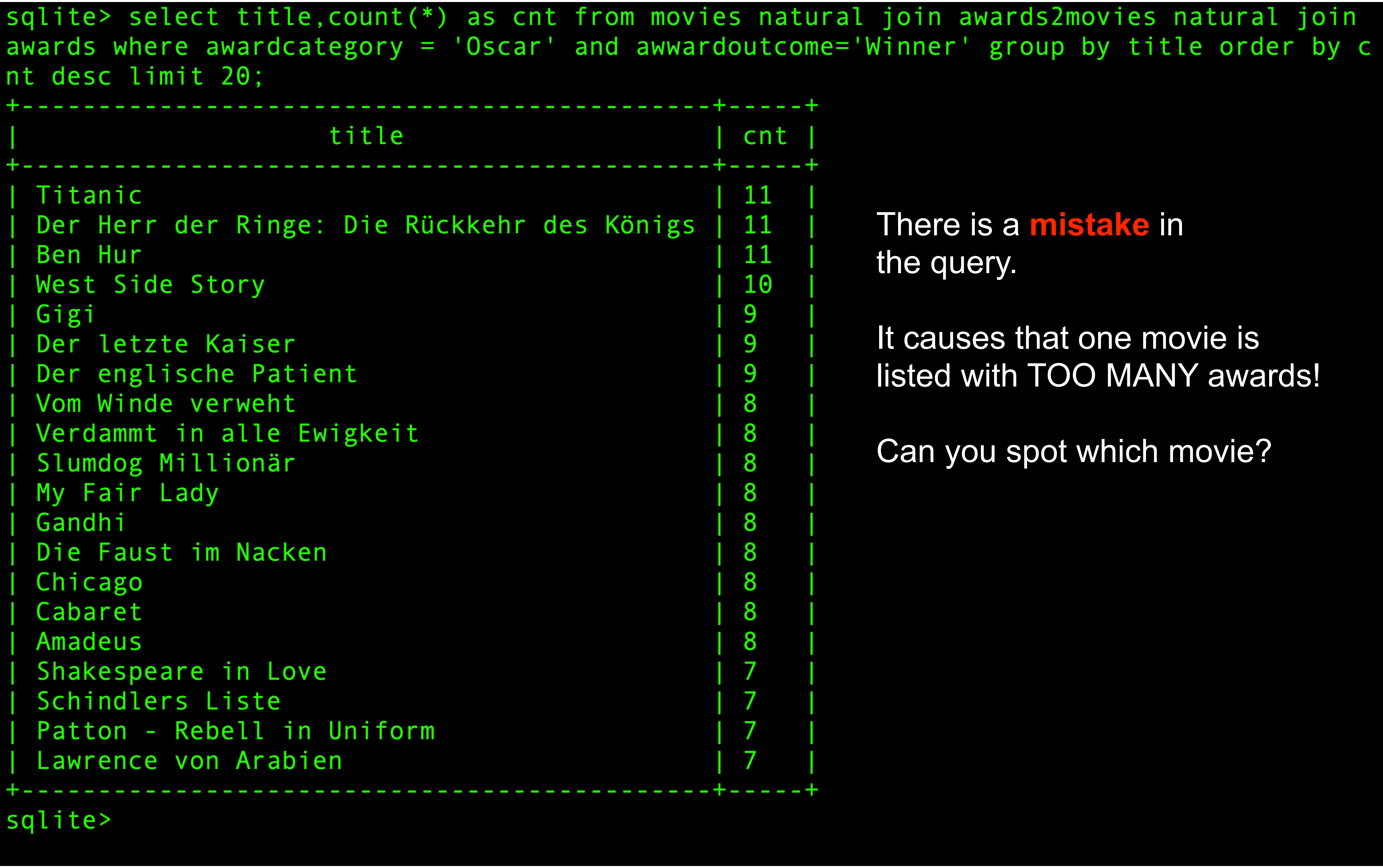
15 movies that won the most Oscars - GoldDerby

6 Jul 2020 — 15 movies that **won the most Oscars**, from 'Titanic' to 'Gone with the Wind' · 'The Lord of the Rings: The Return of the King' (2003) · 'Titanic' (1997).

you may be **lucky** :-)



```
sqlite> select title,count(*) as cnt from movies natural join awards2movies natural join
awards where awardcategory = 'Oscar' and awwardoutcome='Winner' group by title order by c
nt desc limit 20;
+---------------------------------------------+-----+
|                    title                    | cnt |
+---------------------------------------------+-----+
| Titanic                                     | 11  |
| Der Herr der Ringe: Die Rückkehr des Königs | 11  |
| Ben Hur                                     | 11  |
| West Side Story                             | 10  |
| Gigi                                        | 9   |
| Der letzte Kaiser                           | 9   |
| Der englische Patient                       | 9   |
| Vom Winde verweht                           | 8   |
| Verdammt in alle Ewigkeit                   | 8   |
| Slumdog Millionär                           | 8   |
| My Fair Lady                                | 8   |
| Gandhi                                      | 8   |
| Die Faust im Nacken                         | 8   |
| Chicago                                     | 8   |
| Cabaret                                     | 8   |
| Amadeus                                     | 8   |
| Shakespeare in Love                         | 7   |
| Schindlers Liste                            | 7   |
| Patton - Rebell in Uniform                  | 7   |
| Lawrence von Arabien                        | 7   |
+---------------------------------------------+-----+
sqlite>
```

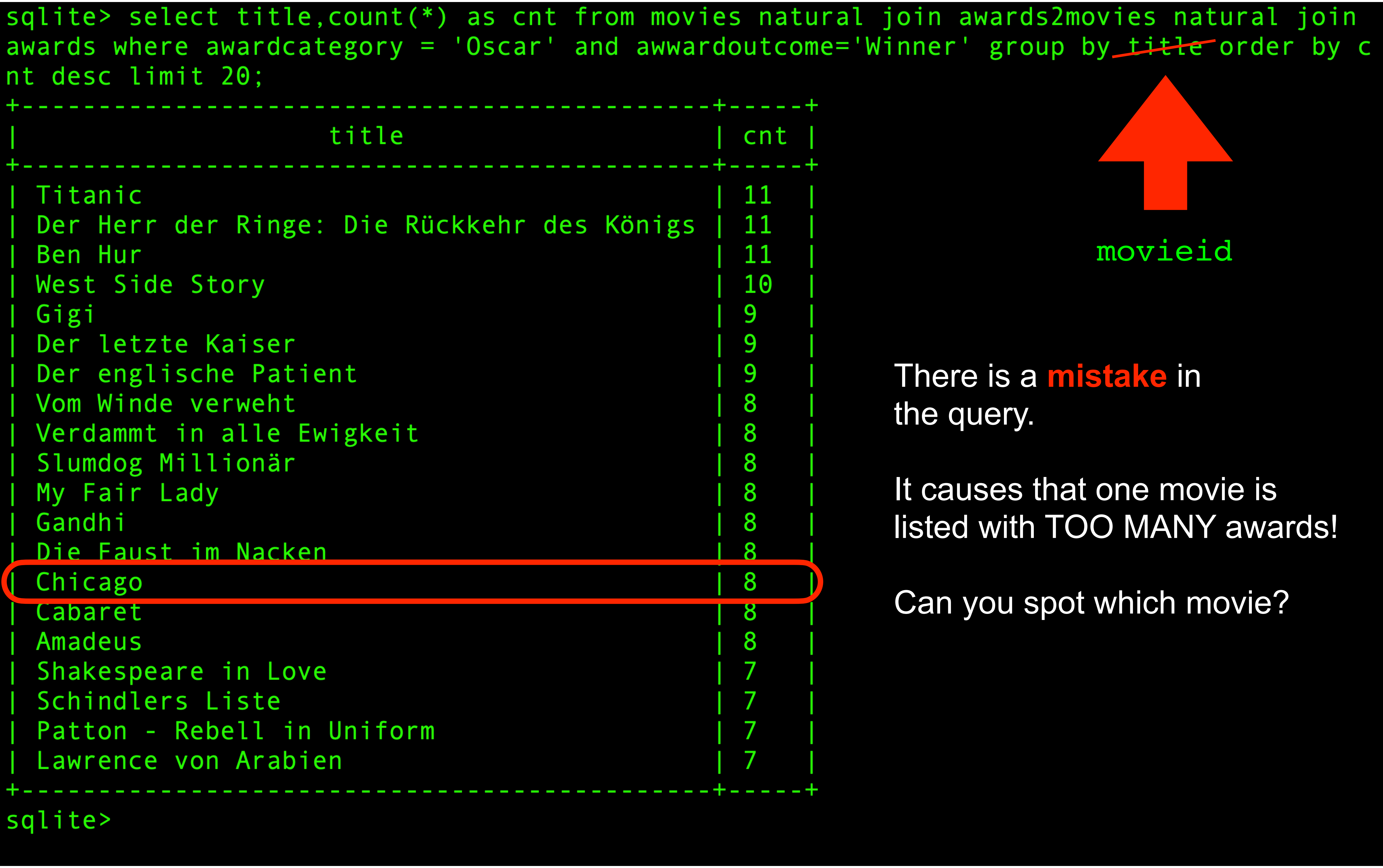


```
sqlite> select title,count(*) as cnt from movies natural join awards2movies natural join
awards where awardcategory = 'Oscar' and awwardoutcome='Winner' group by title order by c
nt desc limit 20;
+---------------------------------------------+-----+
|                    title                    | cnt |
+---------------------------------------------+-----+
| Titanic                                     | 11  |
| Der Herr der Ringe: Die Rückkehr des Königs | 11  |
| Ben Hur                                     | 11  |
| West Side Story                             | 10  |
| Gigi                                        | 9   |
| Der letzte Kaiser                           | 9   |
| Der englische Patient                       | 9   |
| Vom Winde verweht                           | 8   |
| Verdammt in alle Ewigkeit                   | 8   |
| Slumdog Millionär                           | 8   |
| My Fair Lady                                | 8   |
| Gandhi                                      | 8   |
| Die Faust im Nacken                         | 8   |
| Chicago                                     | 8   |
| Cabaret                                     | 8   |
| Amadeus                                     | 8   |
| Shakespeare in Love                         | 7   |
| Schindlers Liste                            | 7   |
| Patton - Rebell in Uniform                  | 7   |
| Lawrence von Arabien                        | 7   |
+---------------------------------------------+-----+
sqlite>
```

There is a **mistake** in the query.

It causes that one movie is listed with TOO MANY awards!

Can you spot which movie?



```
sqlite> select title,count(*) as cnt from movies natural join awards2movies natural join
awards where awardcategory = 'Oscar' and awwardoutcome='Winner' group by ~~title~~ order by c
nt desc limit 20;
+-----------------------------------------+-----+
|                  title                  | cnt |
+-----------------------------------------+-----+
| Titanic                                 | 11  |
| Der Herr der Ringe: Die Rückkehr des Königs | 11  |
| Ben Hur                                 | 11  |
| West Side Story                         | 10  |
| Gigi                                    | 9   |
| Der letzte Kaiser                       | 9   |
| Der englische Patient                   | 9   |
| Vom Winde verweht                       | 8   |
| Verdammt in alle Ewigkeit               | 8   |
| Slumdog Millionär                       | 8   |
| My Fair Lady                            | 8   |
| Gandhi                                  | 8   |
| Die Faust im Nacken                     | 8   |
| Chicago                                 | 8   |
| Cabaret                                 | 8   |
| Amadeus                                 | 8   |
| Shakespeare in Love                     | 7   |
| Schindlers Liste                        | 7   |
| Patton - Rebell in Uniform              | 7   |
| Lawrence von Arabien                    | 7   |
+-----------------------------------------+-----+
sqlite>
```

movieid

There is a **mistake** in the query.

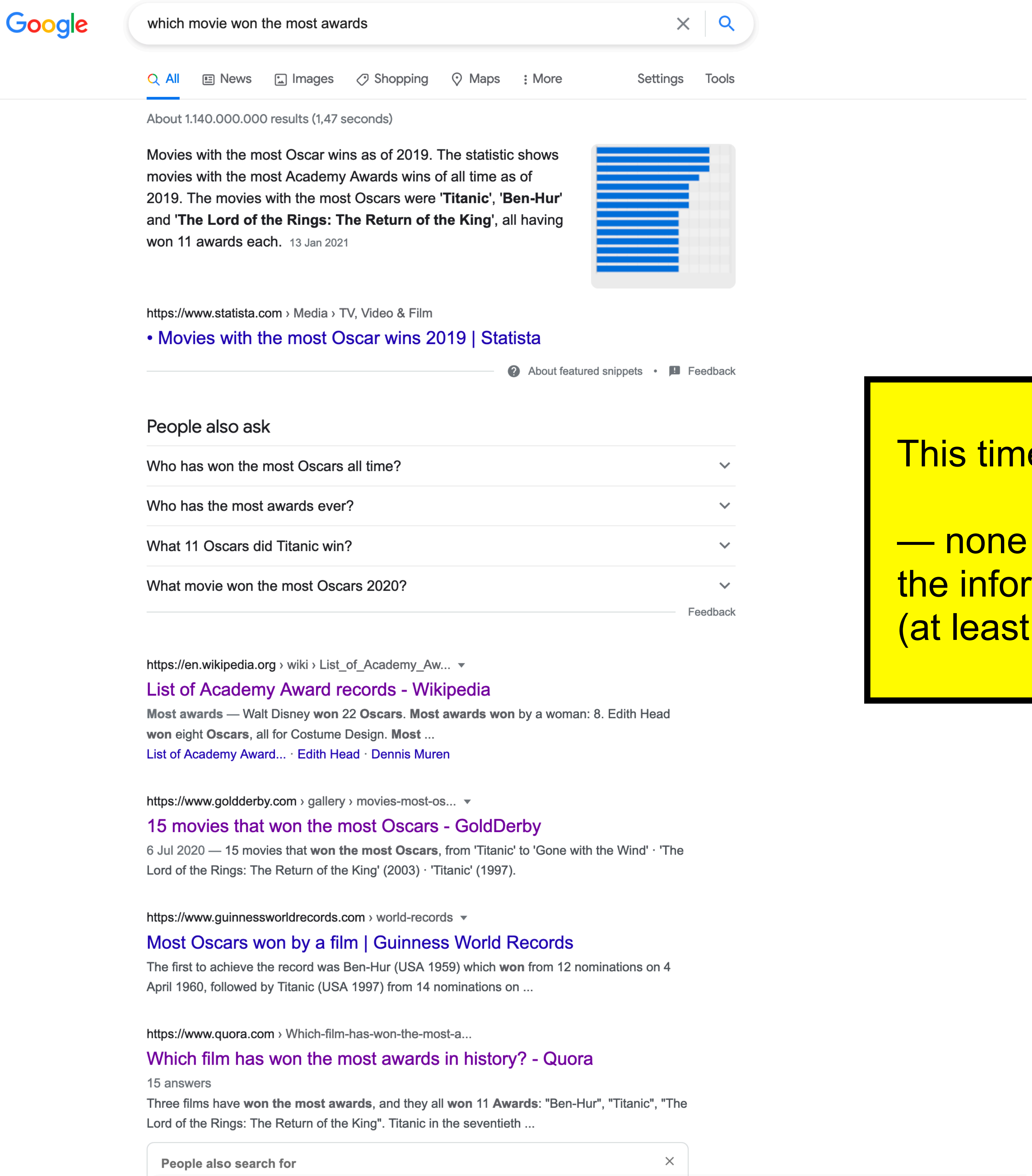
It causes that one movie is listed with TOO MANY awards!

Can you spot which movie?

Complex Queries
High-Speed Execution

Let’s see some examples.
Why not use Full-Text Search (a la Google) for all queries?

— which 20 movies won the most awards?



About 1.140.000.000 results (1,47 seconds)

Movies with the most Oscar wins as of 2019. The statistic shows movies with the most Academy Awards wins of all time as of 2019. The movies with the most Oscars were **'Titanic'**, **'Ben-Hur'** and **'The Lord of the Rings: The Return of the King'**, all having won 11 awards each. 13 Jan 2021

https://www.statista.com › Media › TV, Video & Film

• Movies with the most Oscar wins 2019 | Statista

People also ask

Who has won the most Oscars all time?

Who has the most awards ever?

What 11 Oscars did Titanic win?

What movie won the most Oscars 2020?

https://en.wikipedia.org › wiki › List_of_Academy_Aw... ▾

List of Academy Award records - Wikipedia

Most awards — Walt Disney **won** 22 **Oscars**. **Most awards won** by a woman: 8. Edith Head **won** eight **Oscars**, all for Costume Design. **Most** ...

List of Academy Award... · Edith Head · Dennis Muren

https://www.goldderby.com › gallery › movies-most-os... ▾

15 movies that won the most Oscars - GoldDerby

6 Jul 2020 — 15 movies that **won the most Oscars**, from 'Titanic' to 'Gone with the Wind' · 'The Lord of the Rings: The Return of the King' (2003) · 'Titanic' (1997).

https://www.guinnessworldrecords.com › world-records ▾

Most Oscars won by a film | Guinness World Records

The first to achieve the record was Ben-Hur (USA 1959) which **won** from 12 nominations on 4 April 1960, followed by Titanic (USA 1997) from 14 nominations on ...

https://www.quora.com › Which-film-has-won-the-most-a...

Which film has won the most awards in history? - Quora

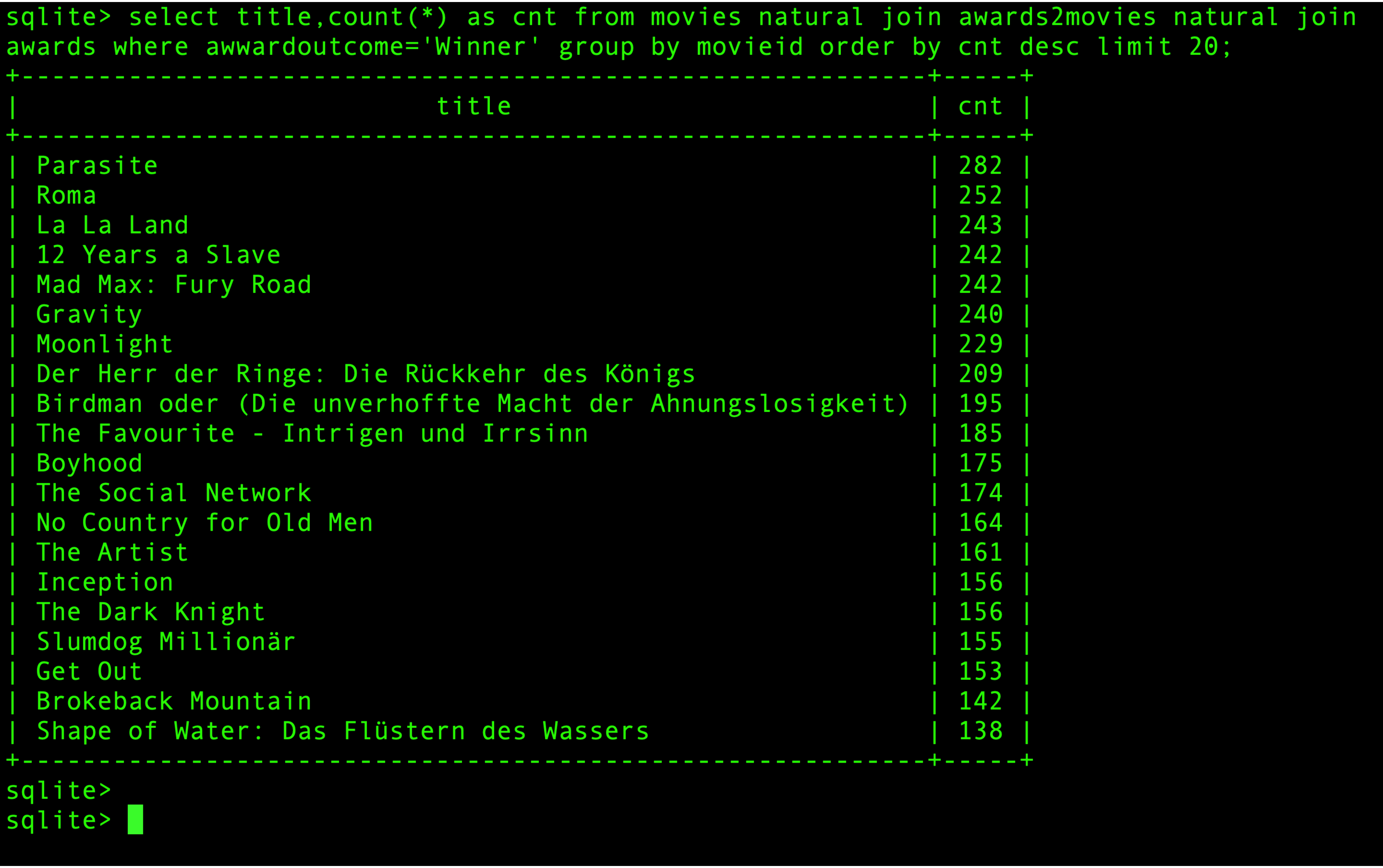
15 answers

Three films have **won the most awards**, and they all **won** 11 **Awards**: "Ben-Hur", "Titanic", "The Lord of the Rings: The Return of the King". Titanic in the seventieth ...

People also search for

This time we are **unlucky** :-(

— none of the pages contains the information
(at least not the first 50 links)



```
sqlite> select title,count(*) as cnt from movies natural join awards2movies natural join
awards where awwardoutcome='Winner' group by movieid order by cnt desc limit 20;
+-----------------------------------------------------------+-----+
|                           title                           | cnt |
+-----------------------------------------------------------+-----+
| Parasite                                                  | 282 |
| Roma                                                      | 252 |
| La La Land                                                | 243 |
| 12 Years a Slave                                          | 242 |
| Mad Max: Fury Road                                        | 242 |
| Gravity                                                   | 240 |
| Moonlight                                                 | 229 |
| Der Herr der Ringe: Die Rückkehr des Königs               | 209 |
| Birdman oder (Die unverhoffte Macht der Ahnungslosigkeit) | 195 |
| The Favourite - Intrigen und Irrsinn                      | 185 |
| Boyhood                                                   | 175 |
| The Social Network                                        | 174 |
| No Country for Old Men                                    | 164 |
| The Artist                                                | 161 |
| Inception                                                 | 156 |
| The Dark Knight                                           | 156 |
| Slumdog Millionär                                         | 155 |
| Get Out                                                   | 153 |
| Brokeback Mountain                                        | 142 |
| Shape of Water: Das Flüstern des Wassers                  | 138 |
+-----------------------------------------------------------+-----+
sqlite>
sqlite>
```

Complex Queries
High-Speed Execution

Why not use **Full-Text Search (a la Google)** for all queries?

— which 20 movies won the most Oscars?

— which 20 movies won the most awards?

— what is the average length of movies made by Werner Herzog?

etc.

Relational Database and
SQL
offer **endlessly more possibilities**!

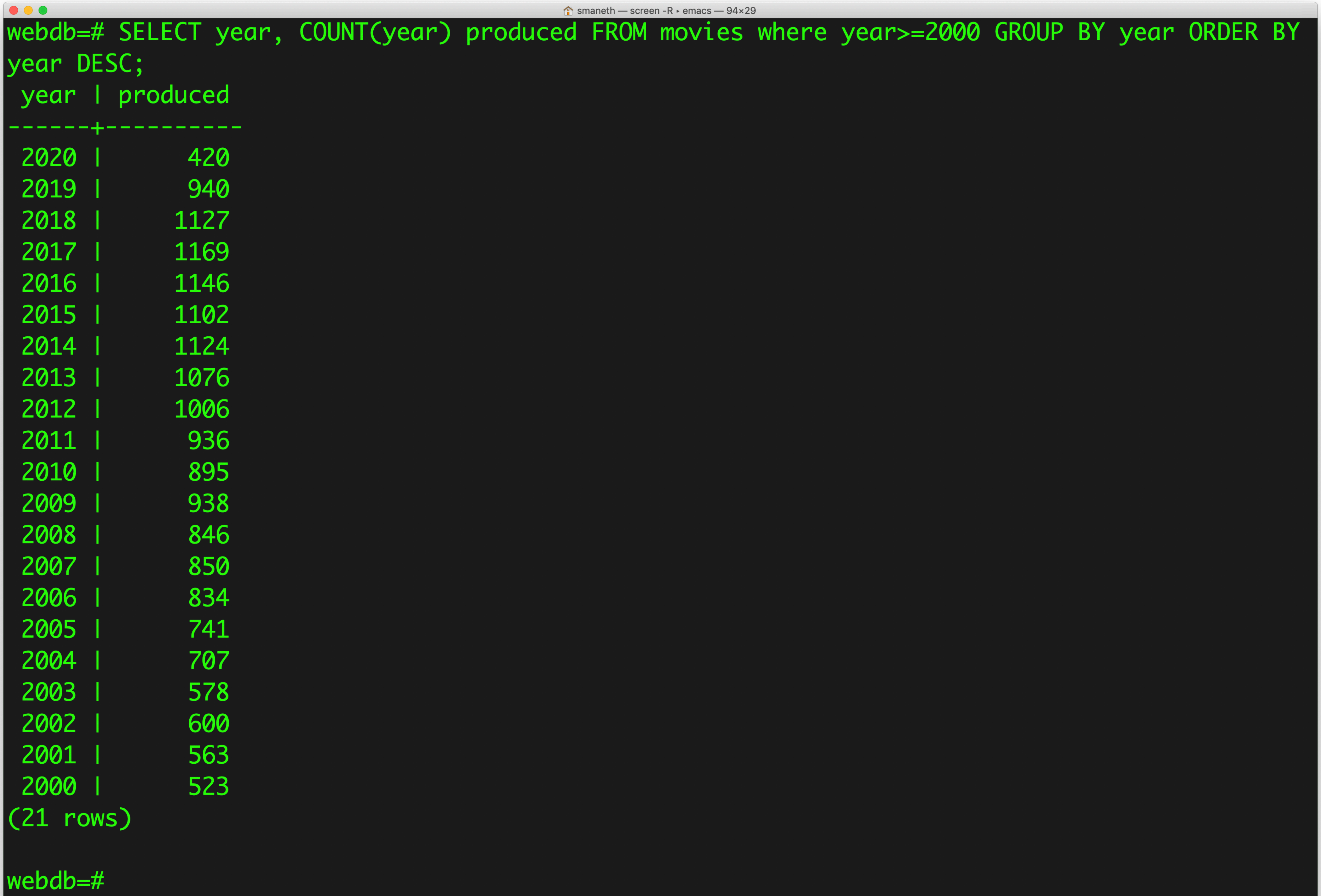
Note:

— it is easy to export query results into CSV files

— results can then be displayed graphically

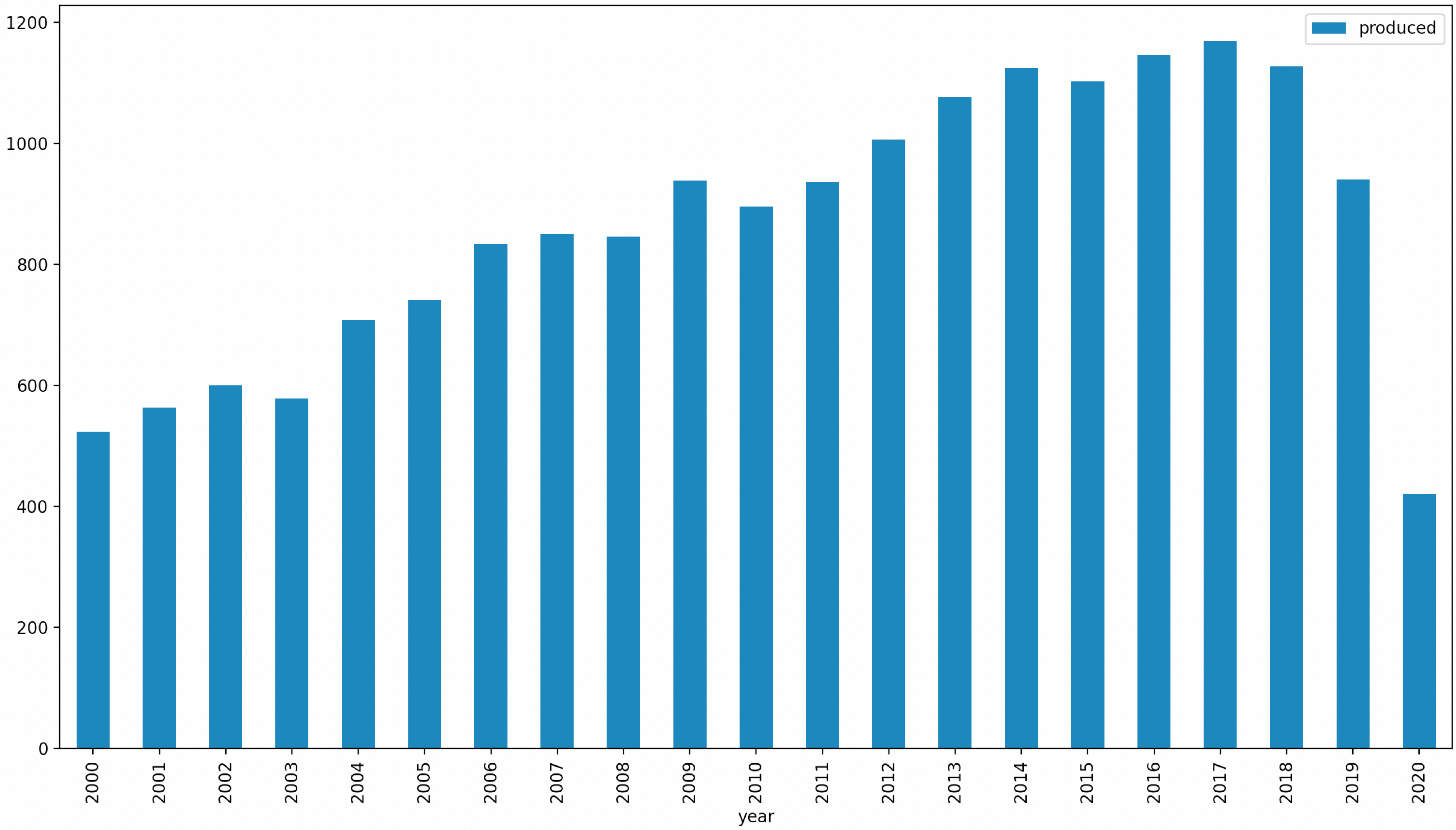
Offers convenient data analytics platform.

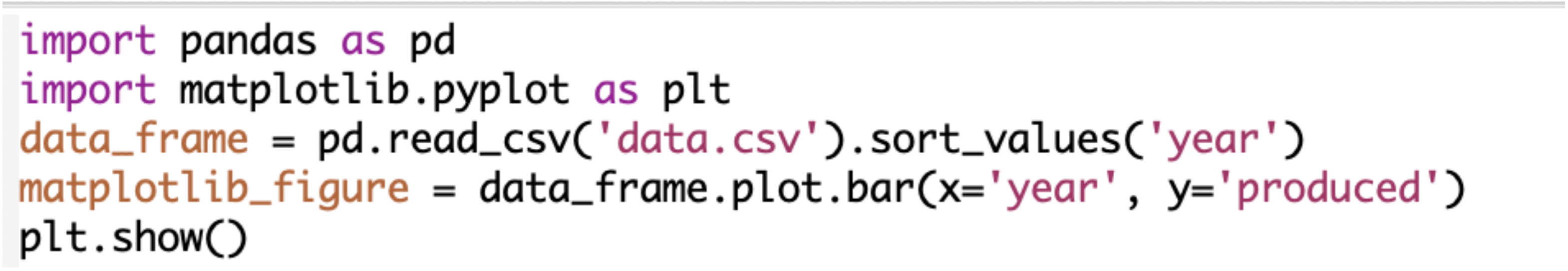
How many movies per year, from year 2000 onwards?



```
webdb=# SELECT year, COUNT(year) produced FROM movies where year>=2000 GROUP BY year ORDER BY
year DESC;
 year | produced
------+----------
 2020 |      420
 2019 |      940
 2018 |     1127
 2017 |     1169
 2016 |     1146
 2015 |     1102
 2014 |     1124
 2013 |     1076
 2012 |     1006
 2011 |      936
 2010 |      895
 2009 |      938
 2008 |      846
 2007 |      850
 2006 |      834
 2005 |      741
 2004 |      707
 2003 |      578
 2002 |      600
 2001 |      563
 2000 |      523
(21 rows)

webdb=#
```



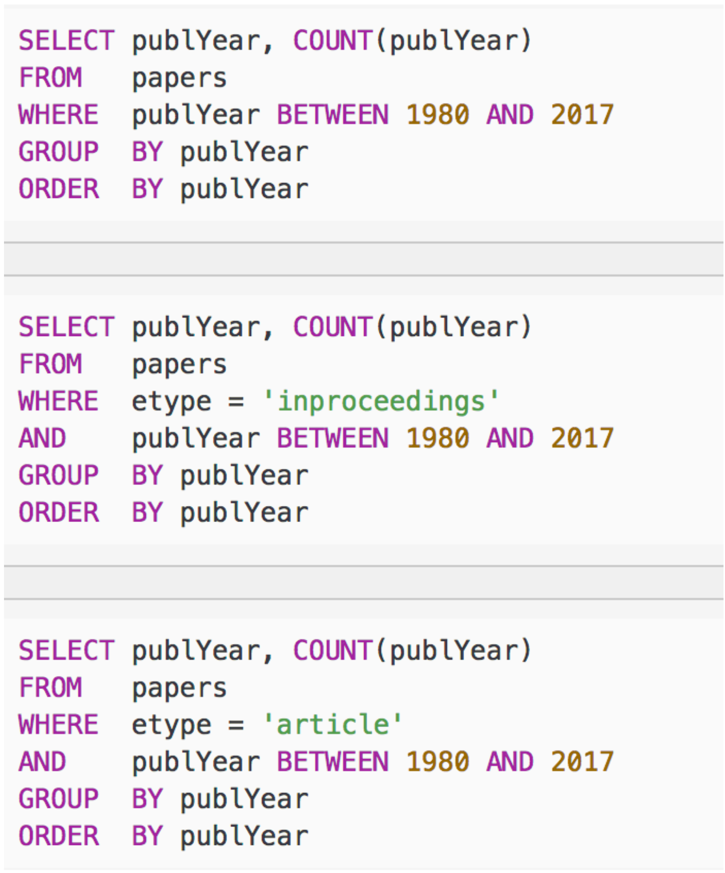


```
import pandas as pd
import matplotlib.pyplot as plt
data_frame = pd.read_csv('data.csv').sort_values('year')
matplotlib_figure = data_frame.plot.bar(x='year', y='produced')
plt.show()
```

Example Database: Publications in Computer Science

Complex Queries
High-Speed Execution

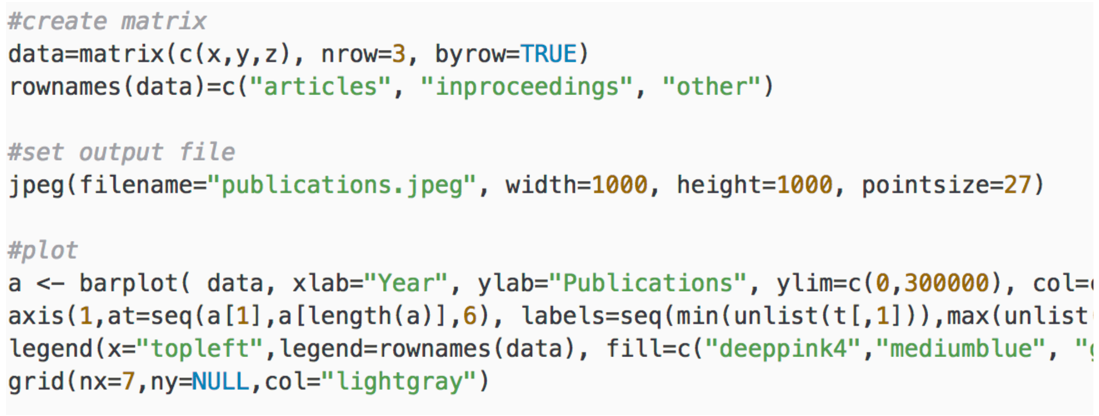
RDBMS + graphical display (e.g., gnuplot, python, R, etc)
= powerful data analytics platform



```
SELECT publYear, COUNT(publYear)
FROM   papers
WHERE  publYear BETWEEN 1980 AND 2017
GROUP  BY publYear
ORDER  BY publYear
```

```
SELECT publYear, COUNT(publYear)
FROM   papers
WHERE  etype = 'inproceedings'
AND    publYear BETWEEN 1980 AND 2017
GROUP  BY publYear
ORDER  BY publYear
```

```
SELECT publYear, COUNT(publYear)
FROM   papers
WHERE  etype = 'article'
AND    publYear BETWEEN 1980 AND 2017
GROUP  BY publYear
ORDER  BY publYear
```

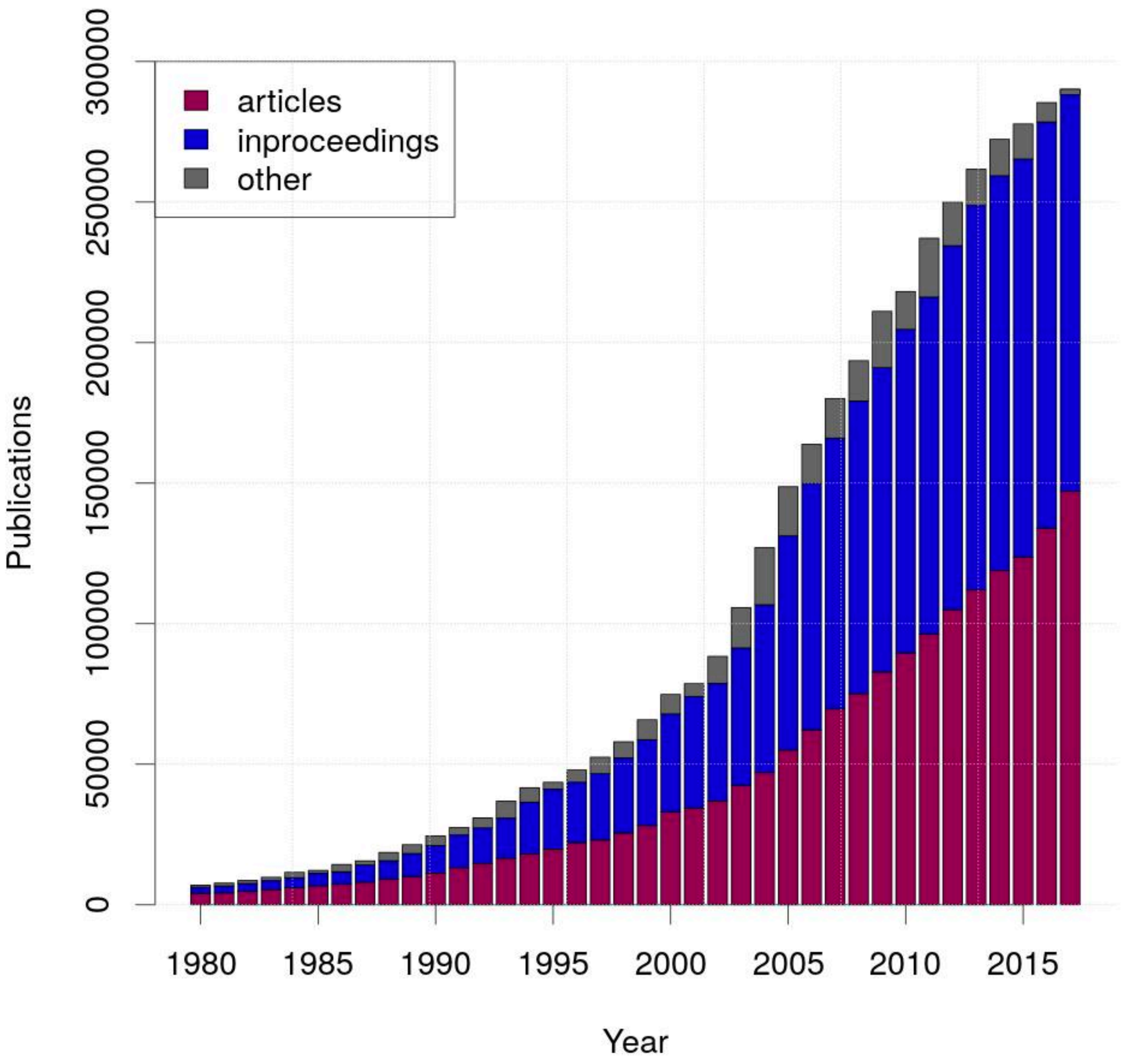


```
#create matrix
data=matrix(c(x,y,z), nrow=3, byrow=TRUE)
rownames(data)=c("articles", "inproceedings", "other")

#set output file
jpeg(filename="publications.jpeg", width=1000, height=1000, pointsize=27)

#plot
a <- barplot( data, xlab="Year", ylab="Publications", ylim=c(0,300000), col=
axis(1,at=seq(a[1],a[length(a)],6), labels=seq(min(unlist(t[,1])),max(unlist
legend(x="topleft",legend=rownames(data), fill=c("deeppink4","mediumblue", "
grid(nx=7,ny=NULL,col="lightgray")
```

#publications in computer science



Claim:

In the early years the **names of authors** always appeard **alphabetically**.

(because this is common in mathematics, and computer science
was scientifically initiated from within mathematics)

Over time this has changed and been inverted.

(because computer science became more and more an
engineering field;
in engineering it is common to order the names of authors
according to certain rules)

Claim:

In the early years the **names of authors** always appeard **alphabetically**.

(because this is common in mathematics, and computer science
was scientifically initiated from within mathematics)

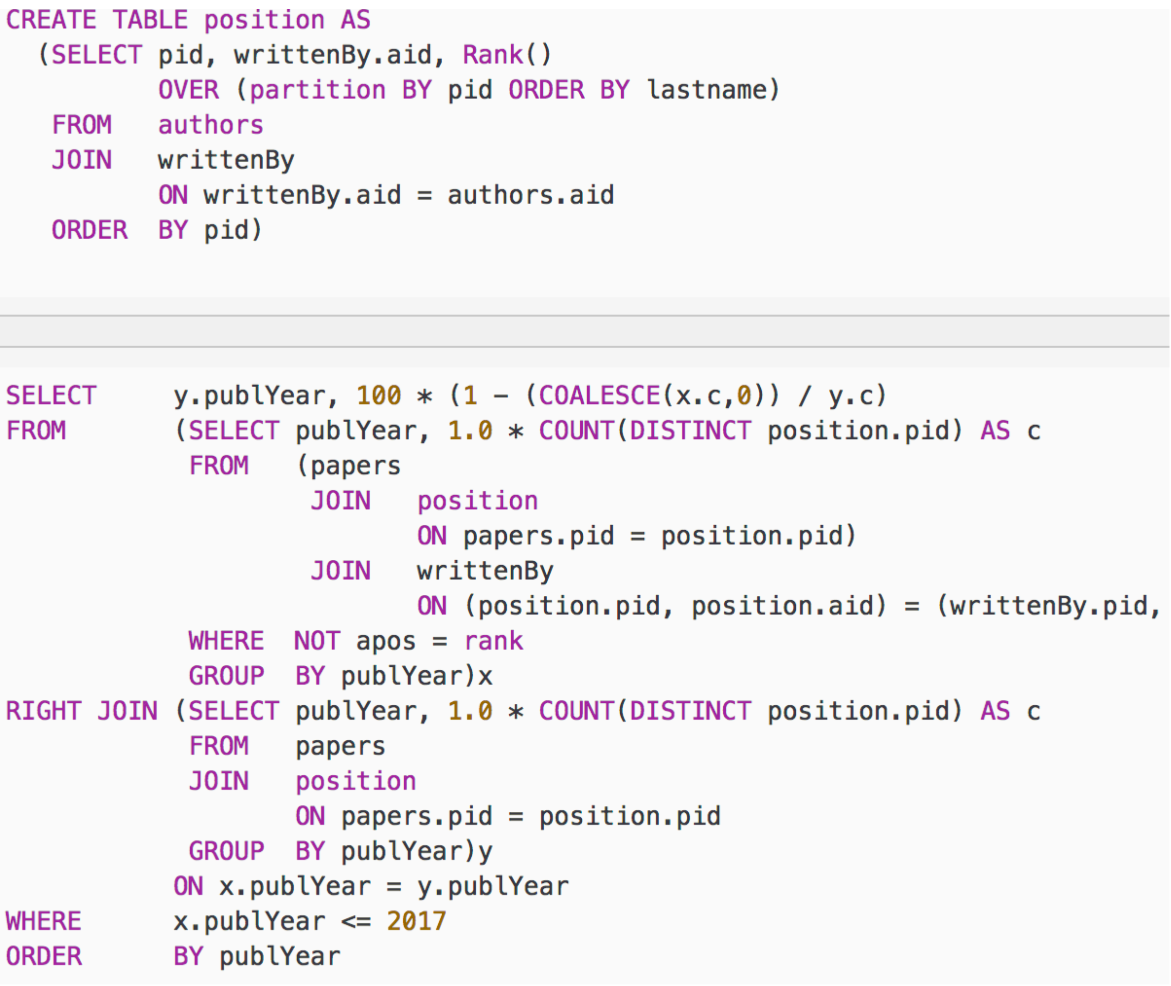
Over time this has changed and been inverted.

(because computer science became more and more an
engineering field;
in engineering it is common to order the names of authors
according to certain rules)

— is this TRUE of FALSE?

Complex Queries
High-Speed Execution

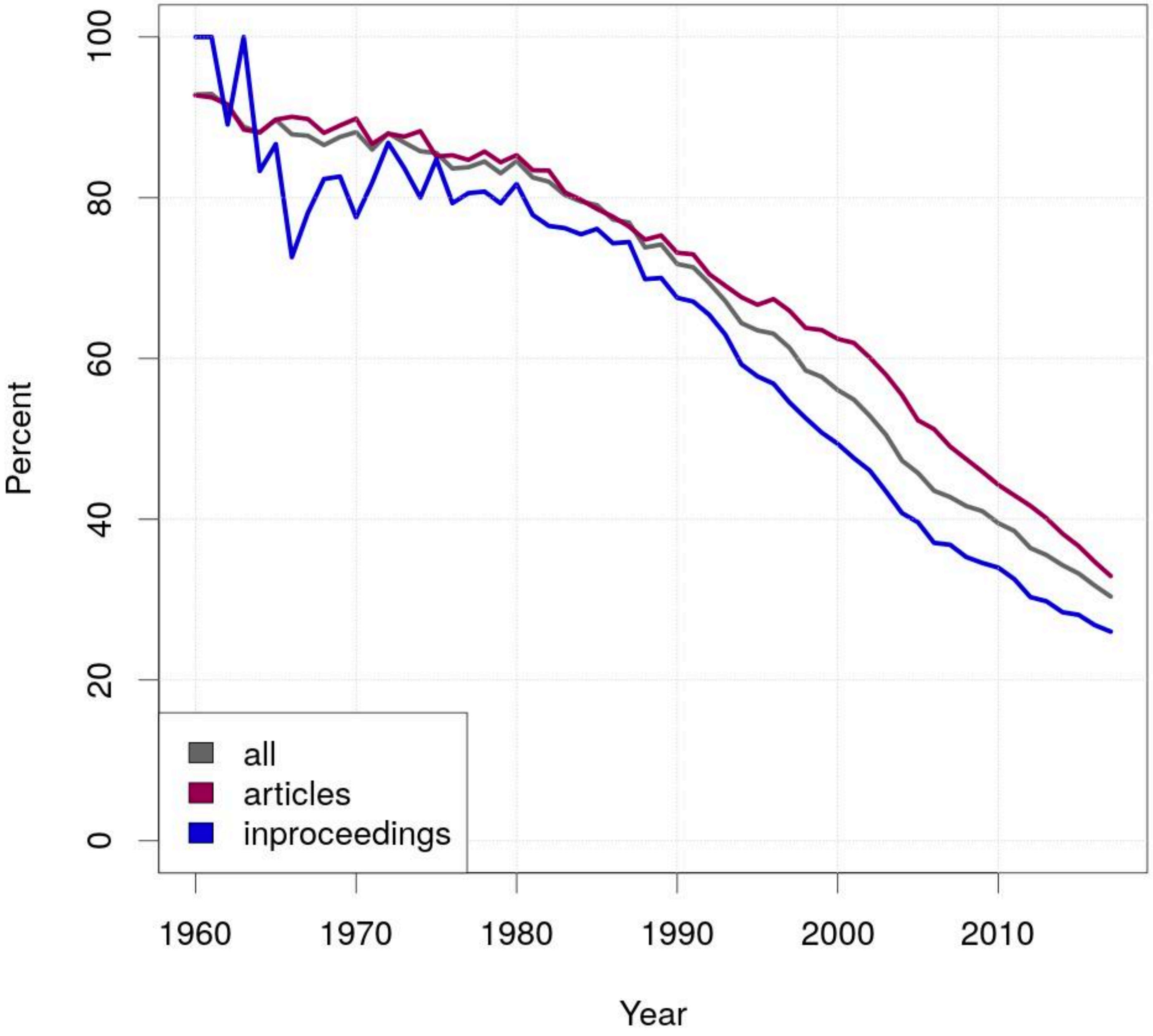
RDBMS + graphical display (e.g., gnuplot, python, R, etc)
= powerful data analytics platform



```
CREATE TABLE position AS
  (SELECT pid, writtenBy.aid, Rank()
          OVER (partition BY pid ORDER BY lastname)
   FROM   authors
   JOIN   writtenBy
          ON writtenBy.aid = authors.aid
   ORDER  BY pid)
```

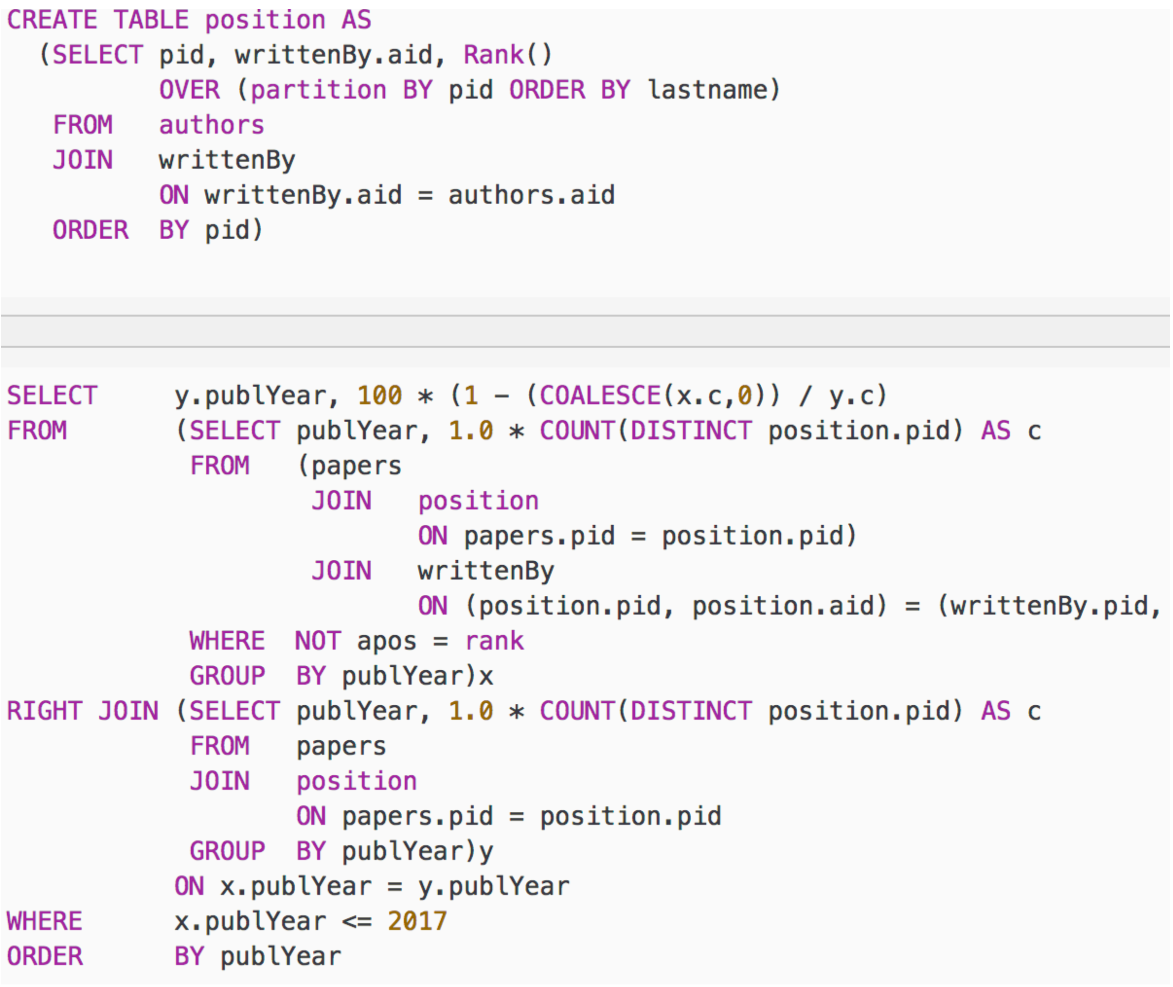
```
SELECT     y.publYear, 100 * (1 - (COALESCE(x.c,0)) / y.c)
FROM       (SELECT publYear, 1.0 * COUNT(DISTINCT position.pid) AS c
            FROM   (papers
                    JOIN   position
                           ON papers.pid = position.pid)
                    JOIN   writtenBy
                           ON (position.pid, position.aid) = (writtenBy.pid,
            WHERE  NOT apos = rank
            GROUP  BY publYear)x
RIGHT JOIN (SELECT publYear, 1.0 * COUNT(DISTINCT position.pid) AS c
            FROM   papers
            JOIN   position
                   ON papers.pid = position.pid
            GROUP  BY publYear)y
           ON x.publYear = y.publYear
WHERE      x.publYear <= 2017
ORDER      BY publYear
```

alphabetical author list (percentage)



Complex Queries
High-Speed Execution

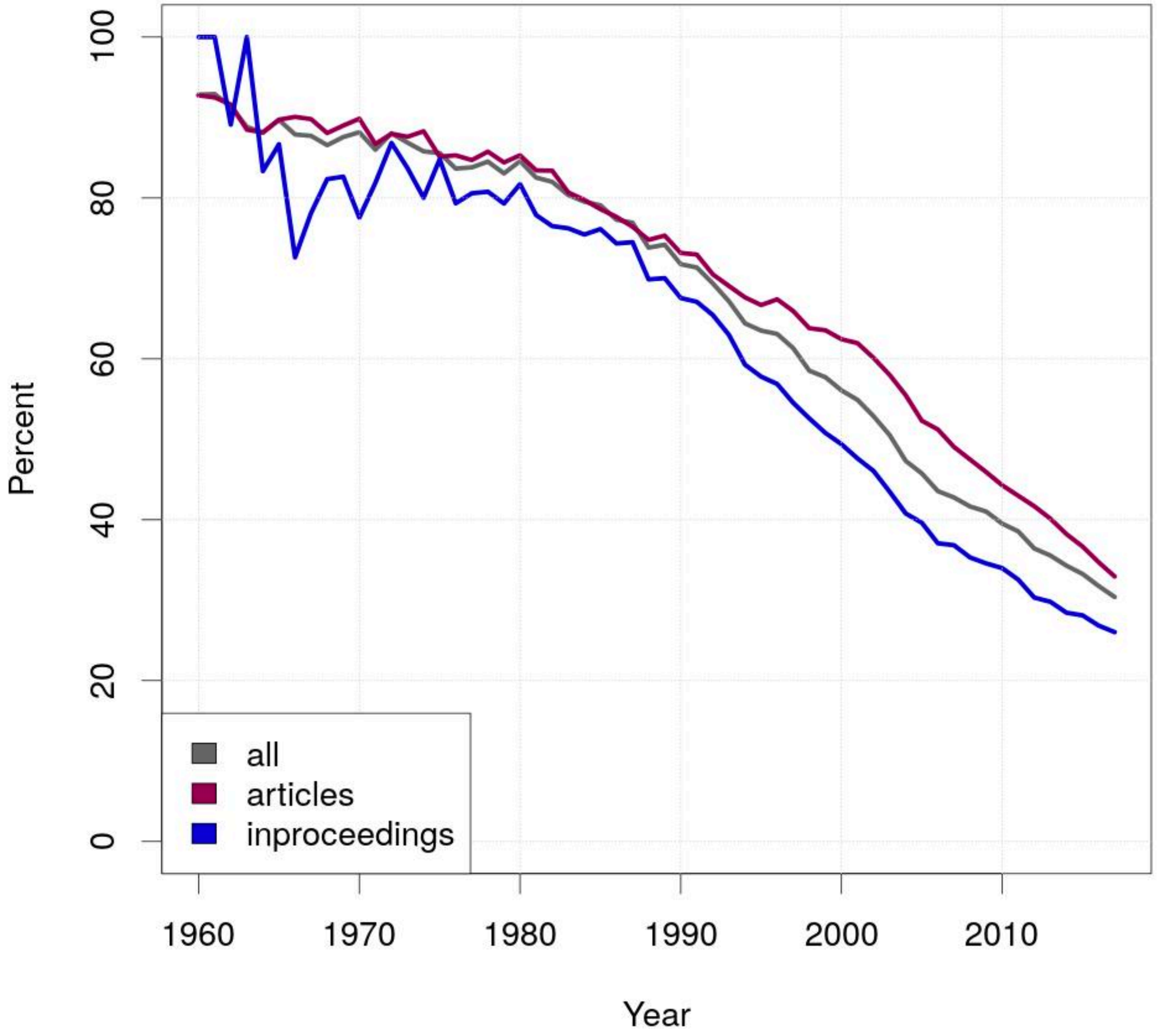
RDBMS + graphical display (e.g., gnuplot, python, R, etc)
= powerful data analytics platform



```
CREATE TABLE position AS
  (SELECT pid, writtenBy.aid, Rank()
          OVER (partition BY pid ORDER BY lastname)
   FROM   authors
   JOIN   writtenBy
          ON writtenBy.aid = authors.aid
   ORDER  BY pid)
```

```
SELECT     y.publYear, 100 * (1 - (COALESCE(x.c,0)) / y.c)
FROM       (SELECT publYear, 1.0 * COUNT(DISTINCT position.pid) AS c
            FROM   (papers
                    JOIN   position
                           ON papers.pid = position.pid)
                    JOIN   writtenBy
                           ON (position.pid, position.aid) = (writtenBy.pid,
            WHERE  NOT apos = rank
            GROUP  BY publYear)x
RIGHT JOIN (SELECT publYear, 1.0 * COUNT(DISTINCT position.pid) AS c
            FROM   papers
            JOIN   position
                   ON papers.pid = position.pid
            GROUP  BY publYear)y
           ON x.publYear = y.publYear
WHERE      x.publYear <= 2017
ORDER      BY publYear
```

alphabetical author list (percentage)



Complex Queries
High-Speed Execution

Build the “Co-Author Graph”:

```
create table CA(aid int, cid int);

insert into CA ( Q1 );
```

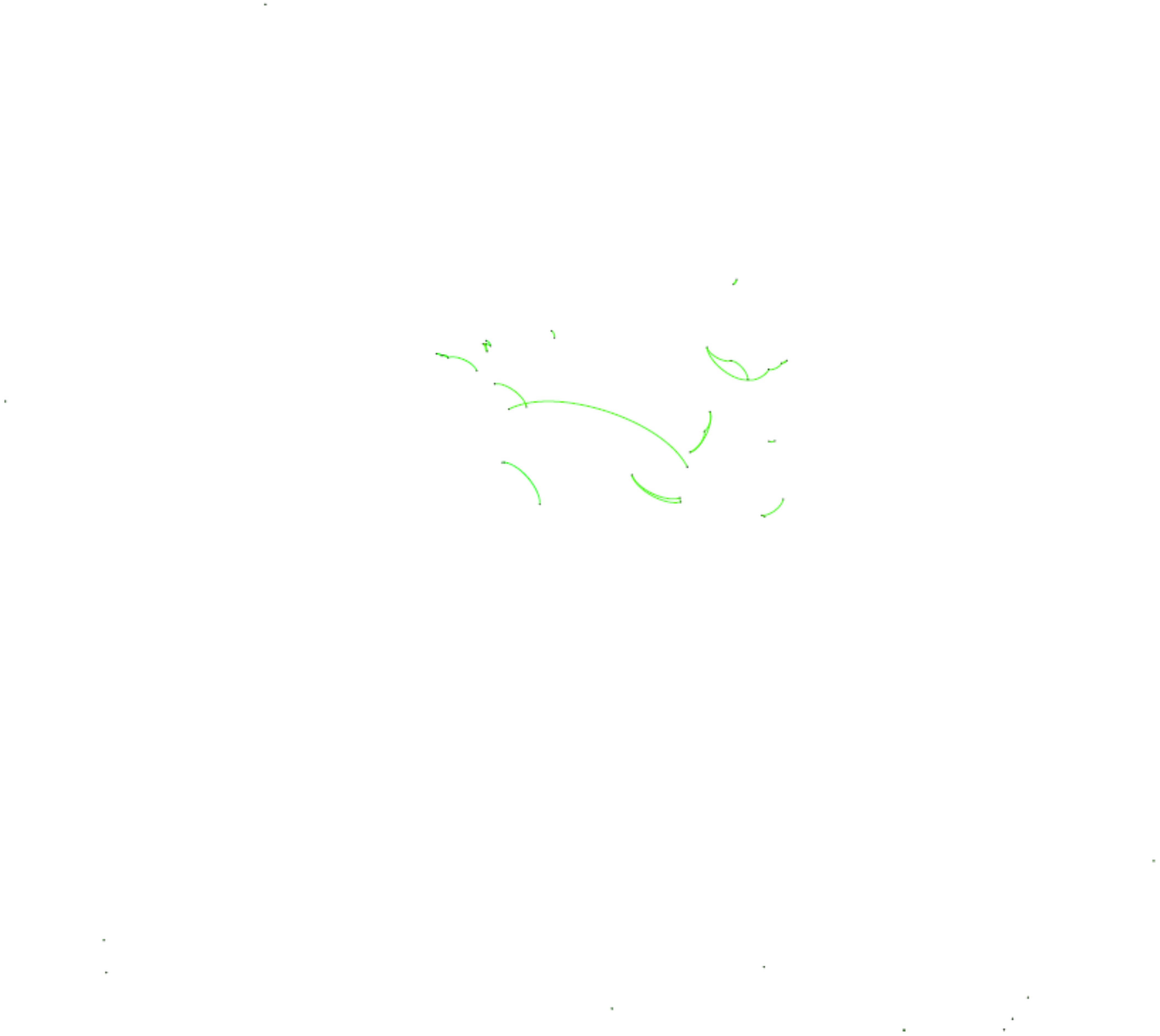
```
select distinct w.aid,ww.aid
from writtenBy w join writtenBy ww on w.pid=ww.pid
Where not(w.aid=ww.aid);
Run Time: real 605.125 user 267.468000 sys 319.000000
```

The co-author graph is a typicall
“collaboration network”

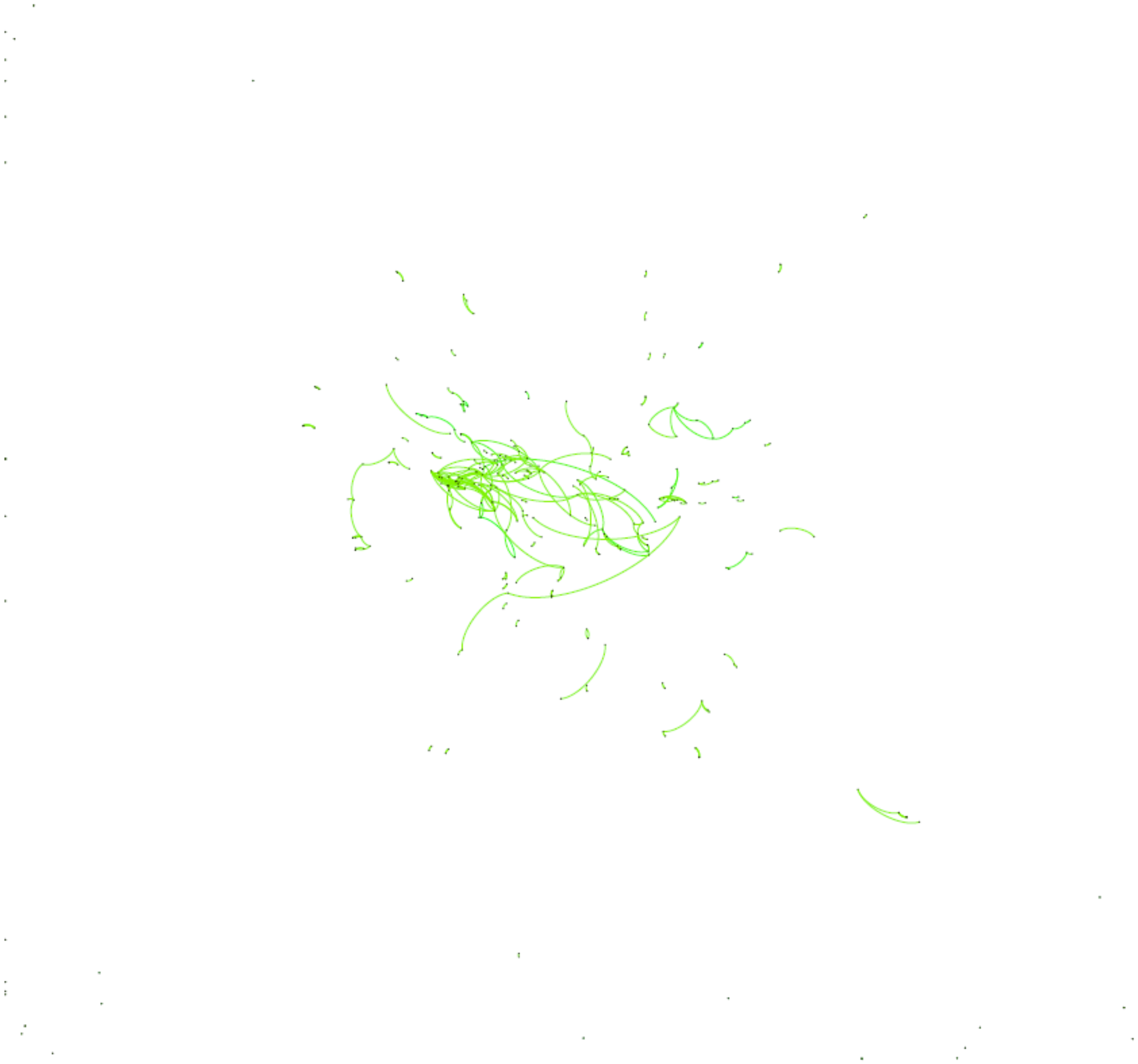
It follows certain **power laws**!

(e.g., a “popular” author with many co-authors will become more and more popular over time)

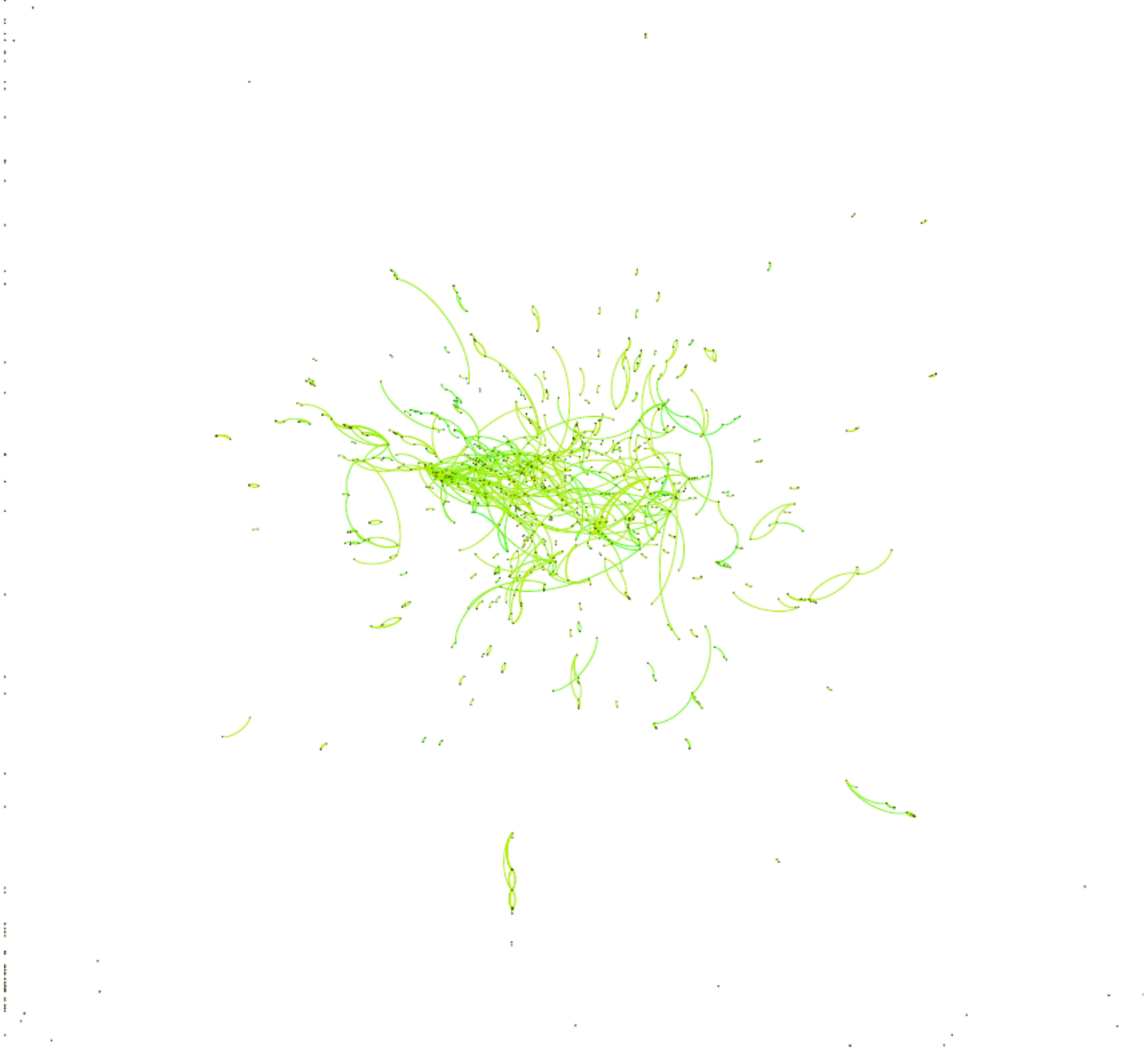
1955



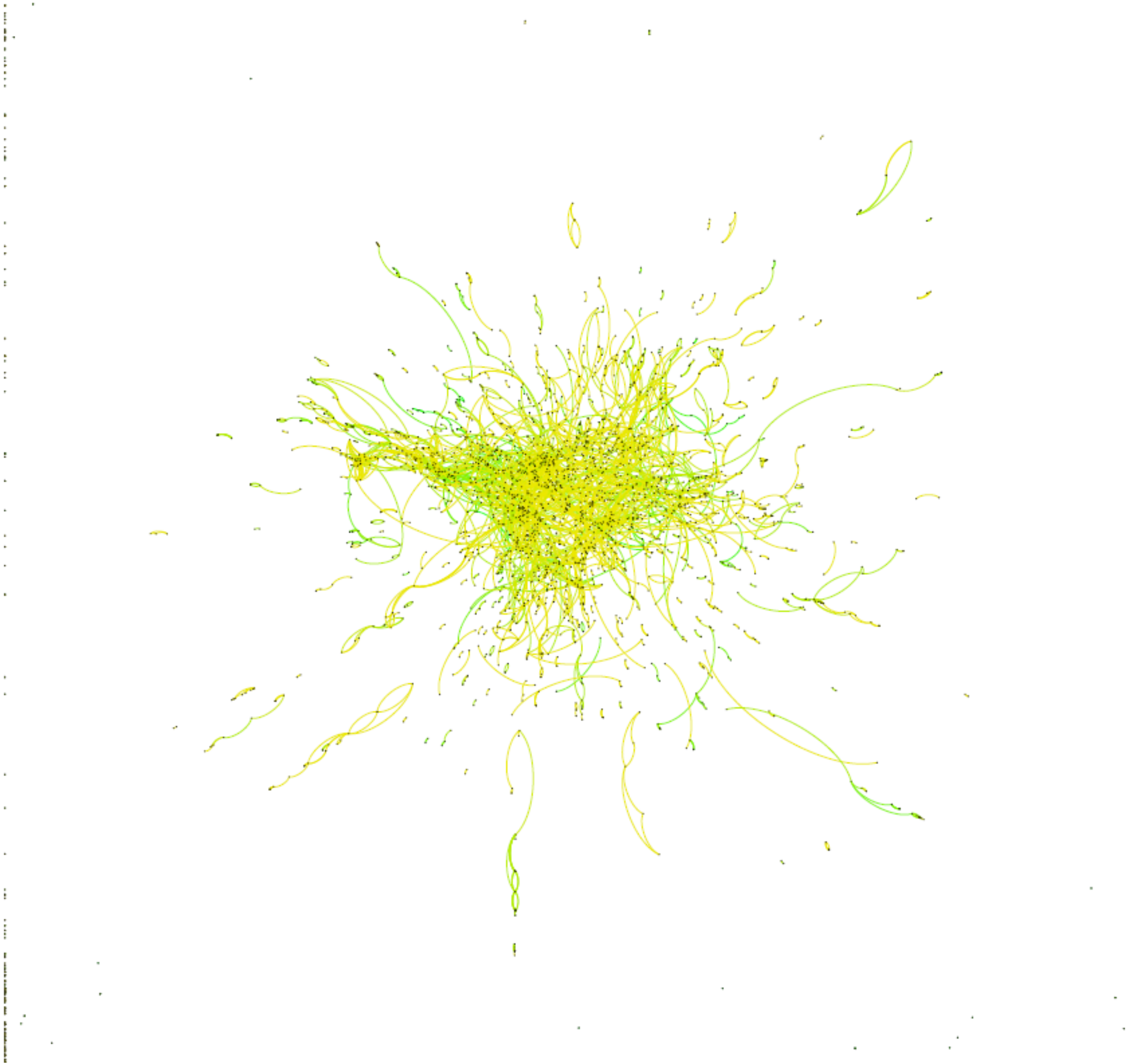
1960



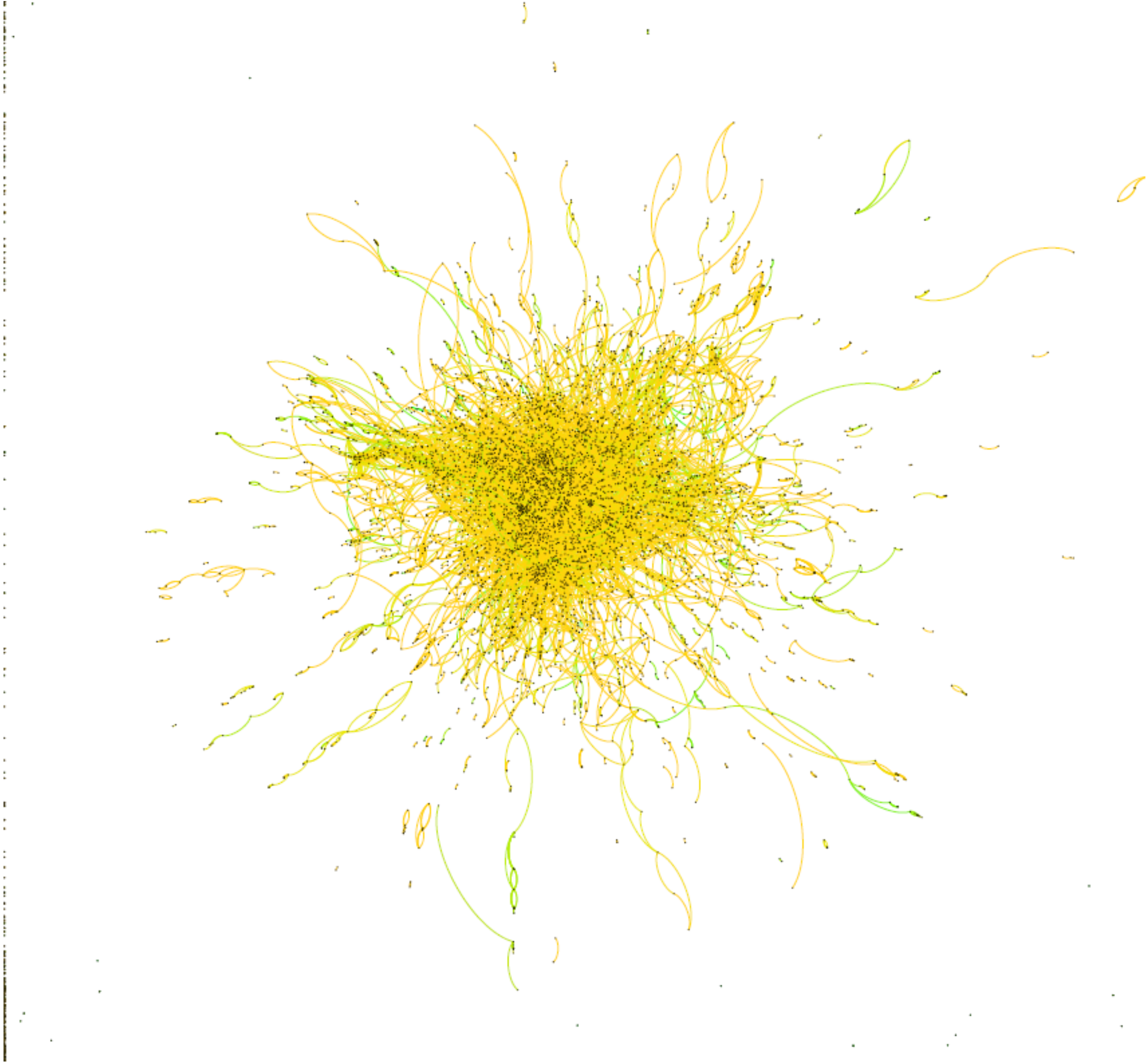
1965



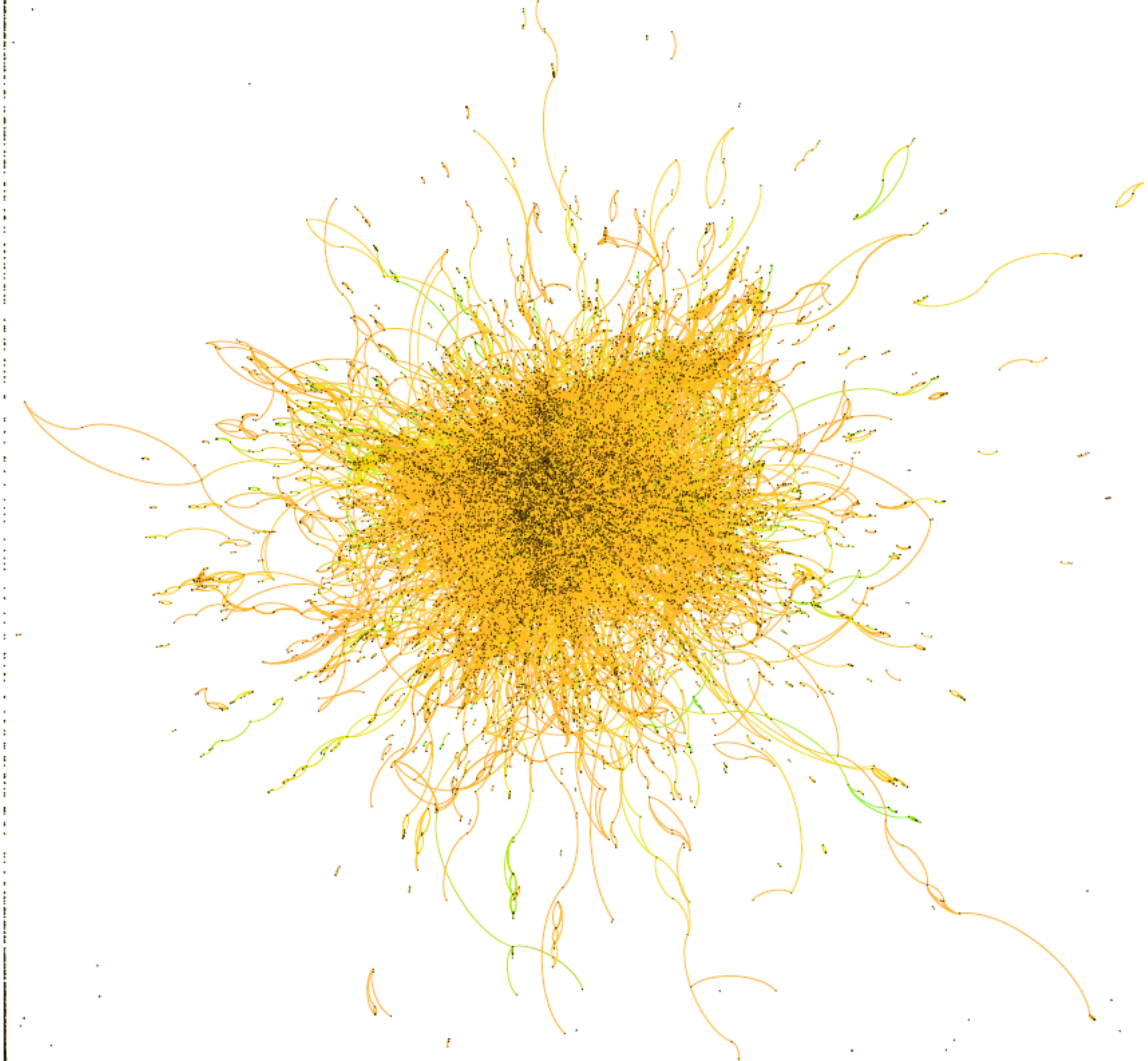
1970



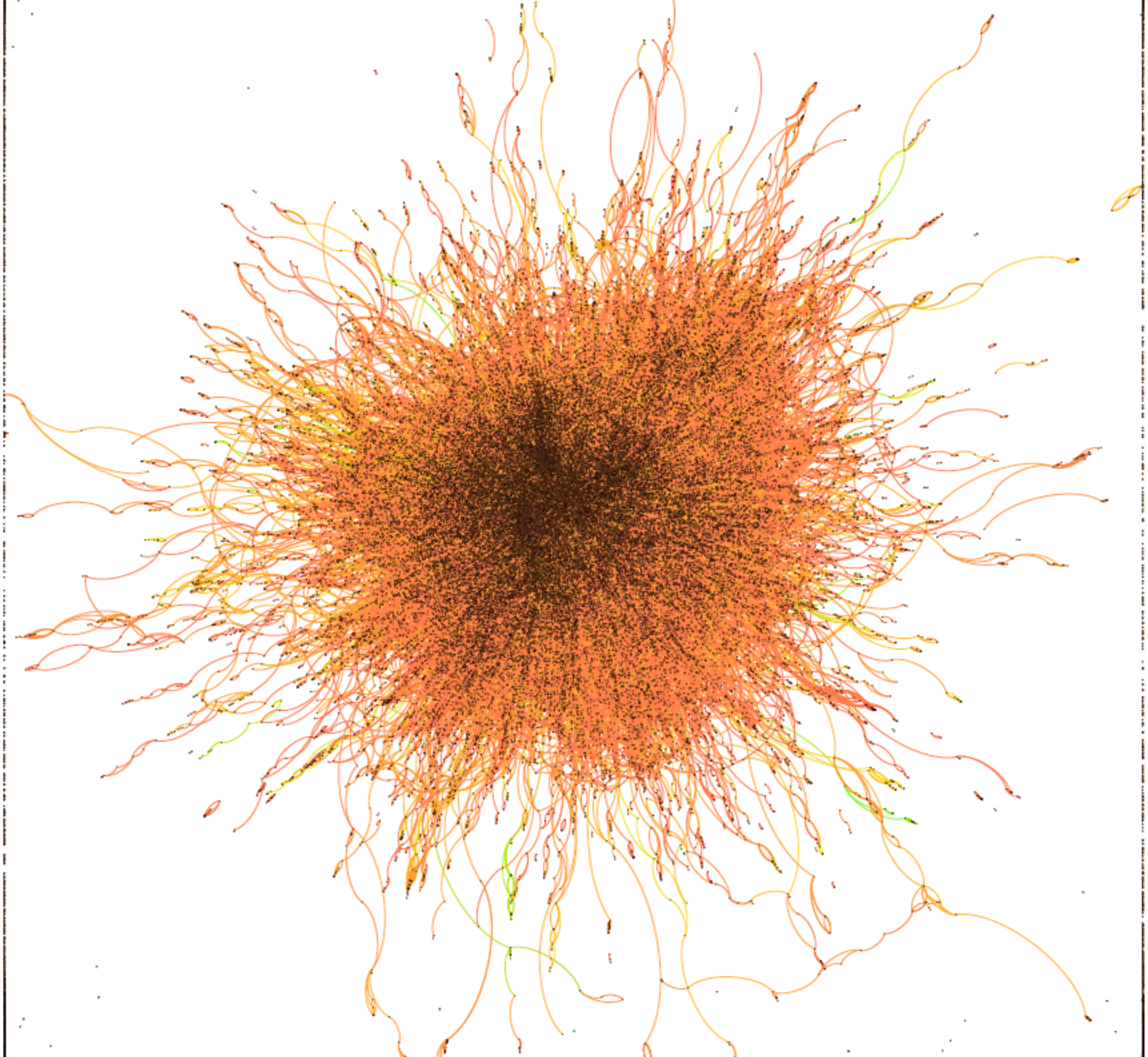
1975



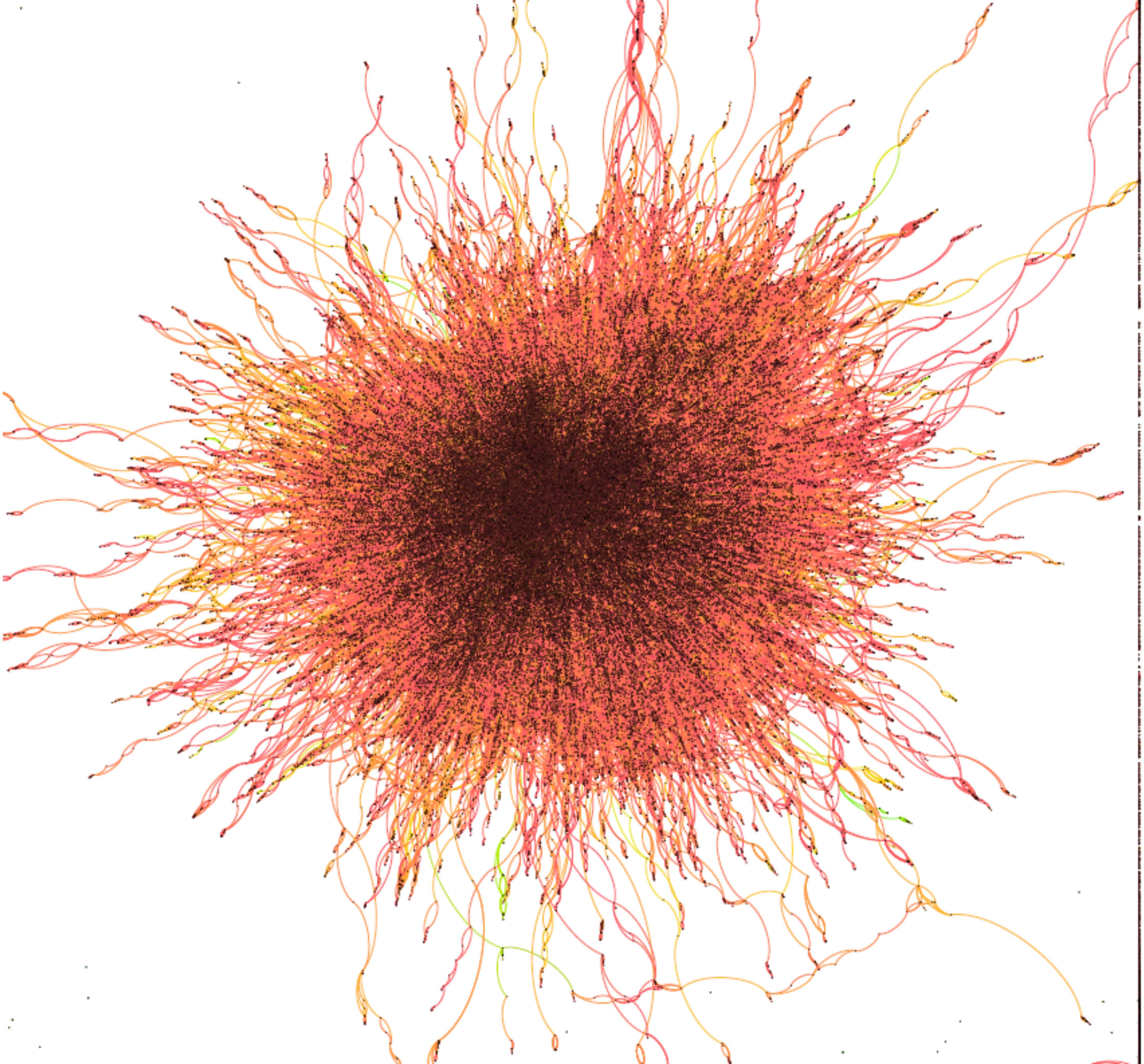
1980

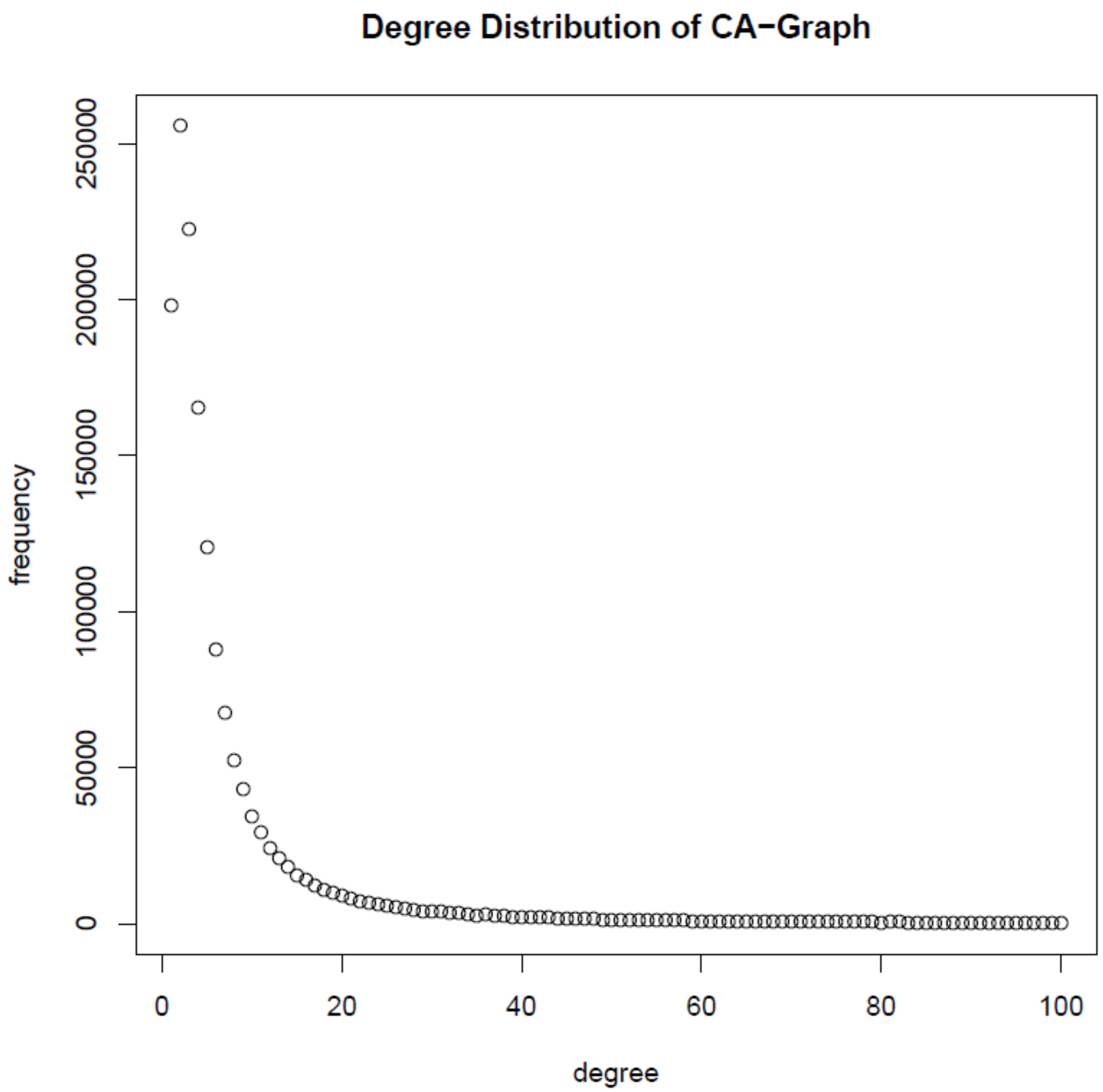


1990



1995




Degree Distribution of CA−Graph
frequency
0
50000
100000
150000
200000
250000
degree
0
20
40
60
80
100

Does it follow a “power law”?

$y = ax^{\mathbf{b}}$

E.g. Zipf-distribution (**b** = -1)

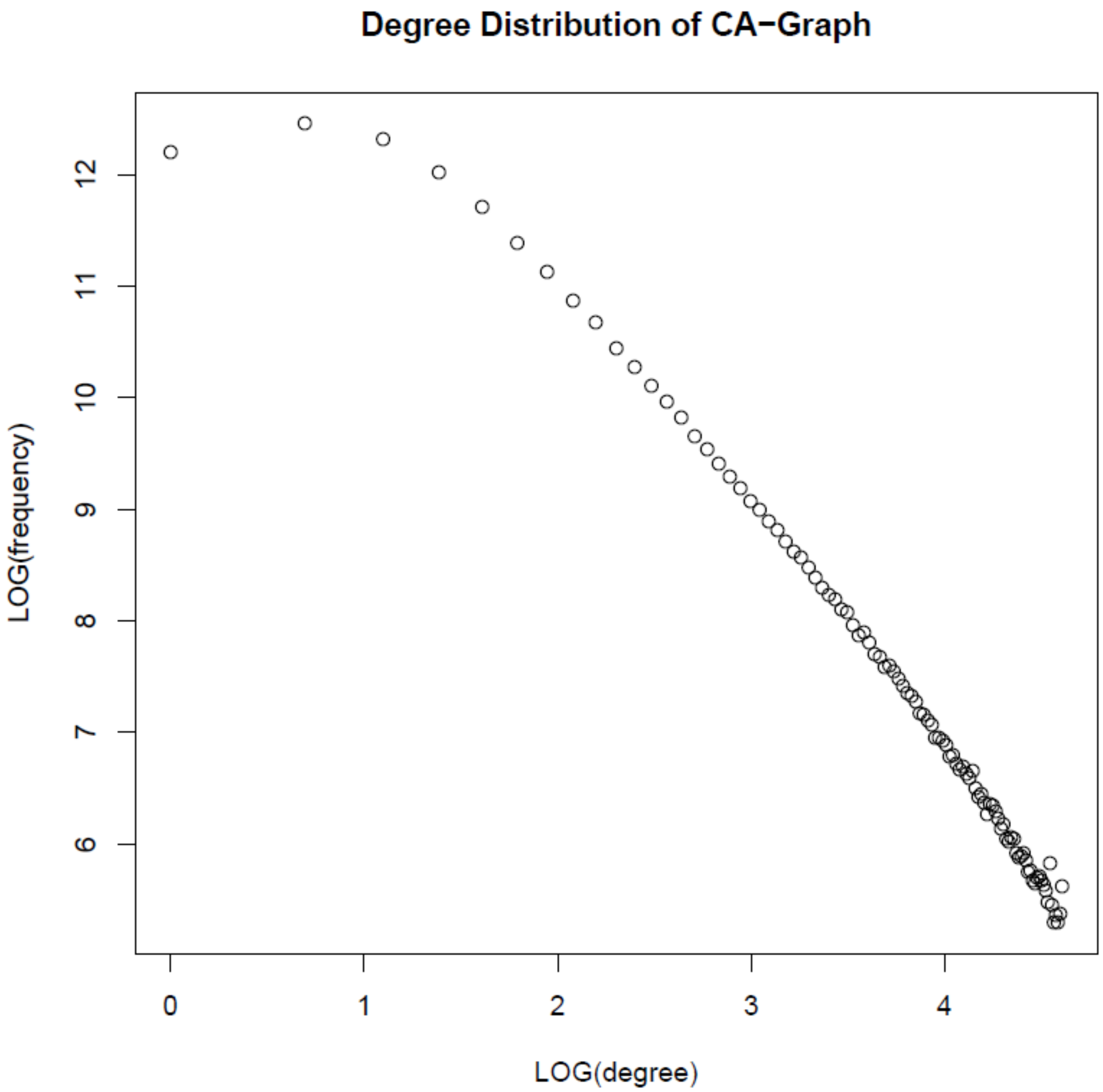
→ take **log**(.)

$\mathbf{log}(y) = \mathbf{log}(a) + \mathbf{b}\ \mathbf{log}(x)$

→ draw x and y on a log-scale ($Y := \mathbf{log}(y)$ and $X := \log(x)$)

$Y = \mathbf{log}(a) + bX$

a line



Yes!!

— this is almost a line.

So it does follow a **Power Law**!

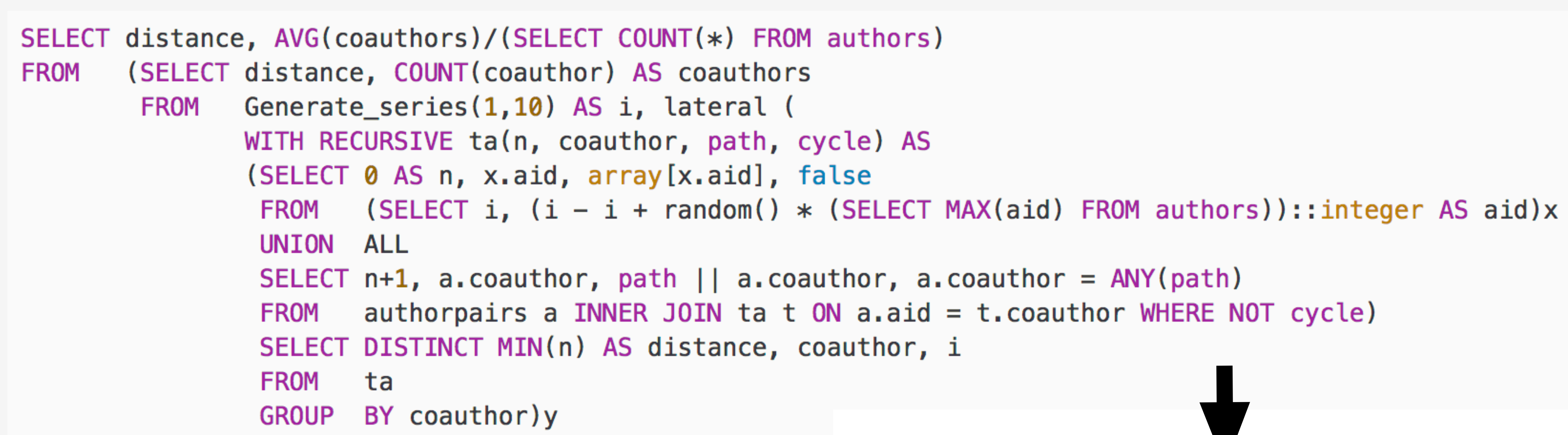
Small World: 6 degrees of separation

"I read somewhere that everybody on this planet is separated by only six other people. ***Six degrees of separation****. Between us and everybody else on this planet. The president of the United States. A gondolier in Venice. fill in the names. I find that*

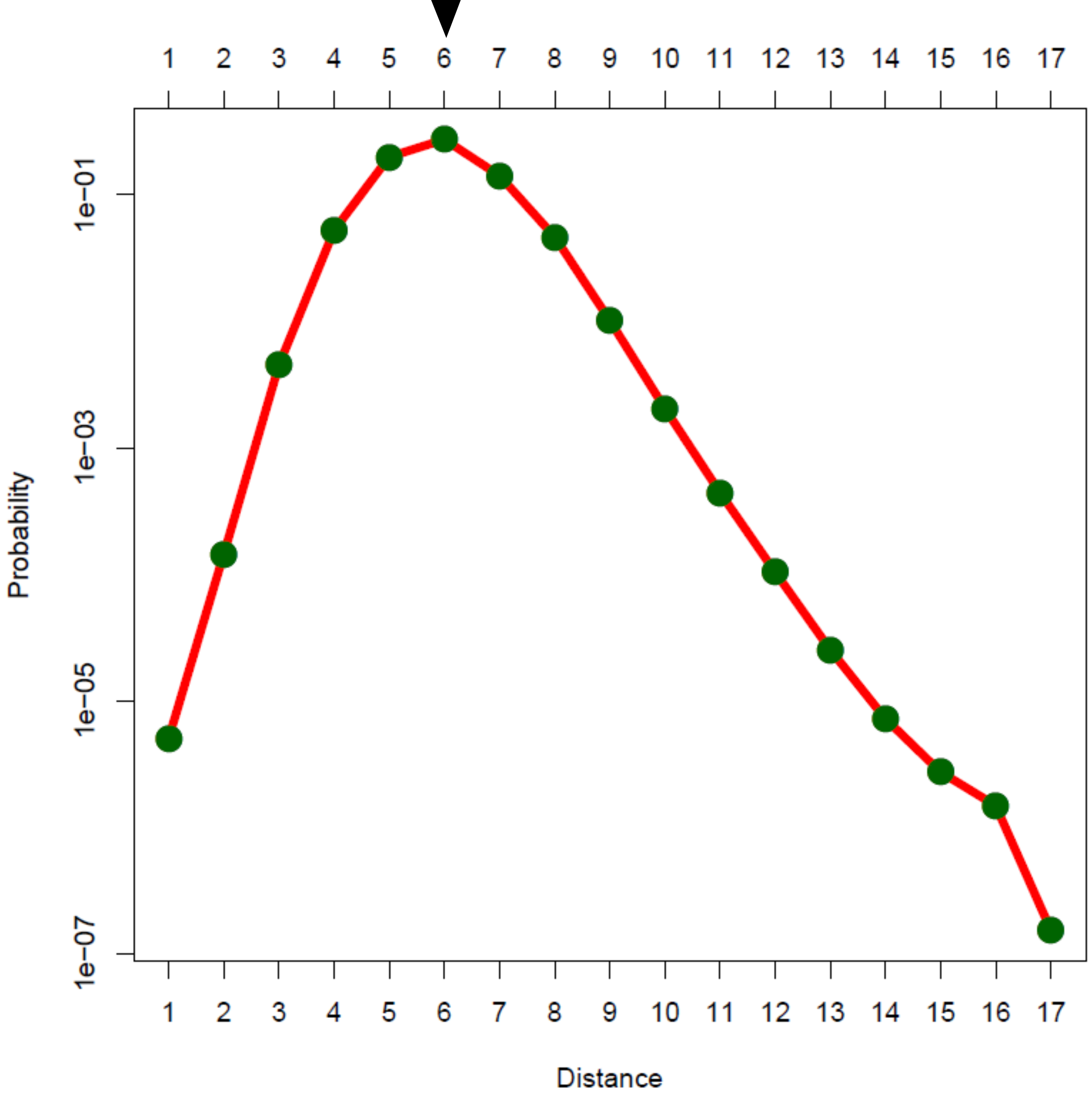
A) tremendously comforting that we're so close and

B) like Chinese water torture that we're so close. Because you have to find the right six people to make the connection. It's not just big names. It's anyone. A native in a rain forest. A Tierra del Fuegan. An Eskimo. I am bound to everyone on this planet by a trail of six people. It's a profound thought. How Paul found us. How to find the man whose son he pretends to be. Or perhaps is his son, although I doubt it. How every person is a new door, opening up into other worlds. Six degrees of separation between me and everyone else on this planet. But to find the right six people."

John Guare



```
SELECT distance, AVG(coauthors)/(SELECT COUNT(*) FROM authors)
FROM   (SELECT distance, COUNT(coauthor) AS coauthors
        FROM   Generate_series(1,10) AS i, lateral (
               WITH RECURSIVE ta(n, coauthor, path, cycle) AS
               (SELECT 0 AS n, x.aid, array[x.aid], false
                FROM   (SELECT i, (i - i + random() * (SELECT MAX(aid) FROM authors))::integer AS aid)x
                UNION  ALL
                SELECT n+1, a.coauthor, path || a.coauthor, a.coauthor = ANY(path)
                FROM   authorpairs a INNER JOIN ta t ON a.aid = t.coauthor WHERE NOT cycle)
                SELECT DISTINCT MIN(n) AS distance, coauthor, i
                FROM   ta
                GROUP  BY coauthor)y
        WHERE  NOT distance=0
        GROUP  BY distance, y.i
        ORDER  BY y.i)z
GROUP  BY distance
ORDER  BY distance
```



Kurs **Datenbankgrundlagen und Modellierung**

1. Overview of Database Systems

— the following slides are from
Jens Teubner's 2018 lecture “Information Systems” (TU Dortmund)

— you can access the slides here:

http://dbis.cs.tu-dortmund.de/cms/en/teaching/ss18/infosys/slides/overview.pdf