Database Management System

Lecture: Introduction & Course Logistics

(Winter 2022-23)

Two courses

1. Database Management System (CS204) (3-0-0-3)

2. Database Management System Lab (CS262) (0-0-3-2)

Teaching Assistants

Atul Sharma - 202171002 (PhD)

Darshna Parmar - 202271001 (PhD)

KHUSHBU VALLABHBHAI PARMAR - 202262005 (MTech)

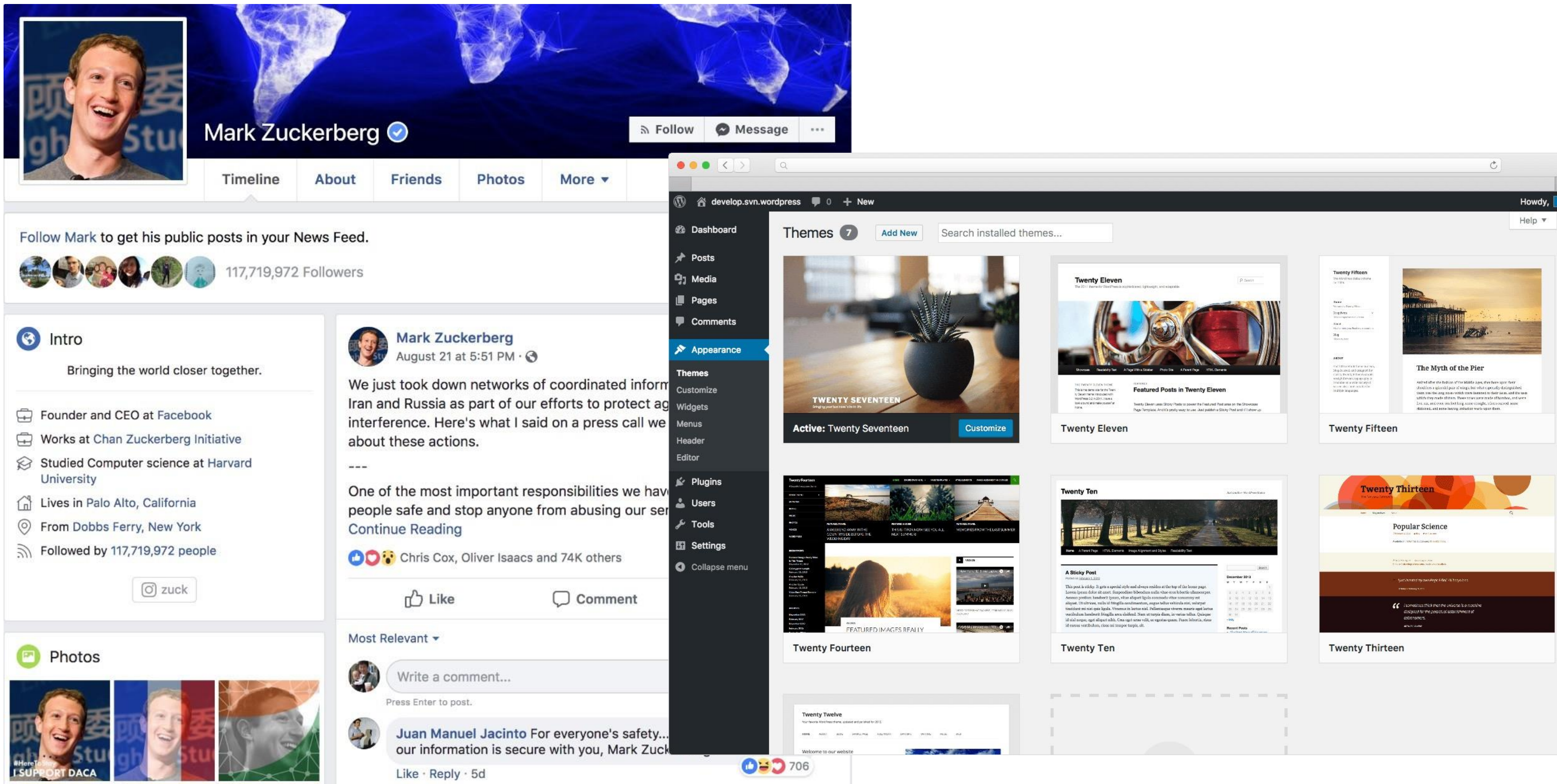
Database Management System (CS204)

- Course Evaluation
  - Mid-Term evaluation: 30%
  - End-Term evaluation: 45%
  - Assignments/Quiz (2 Nos.): 25%

Database Management System (CS262)

- Course Evaluation
    - Mid-term Viva: 20%
    - Lab Assignments: 50%
    - End-Term Project Evaluation: 30%

Warm up Qns



Facebook uses ? to store posts

WordPress uses ? to manage components of a website (pages, links, menus, etc.)

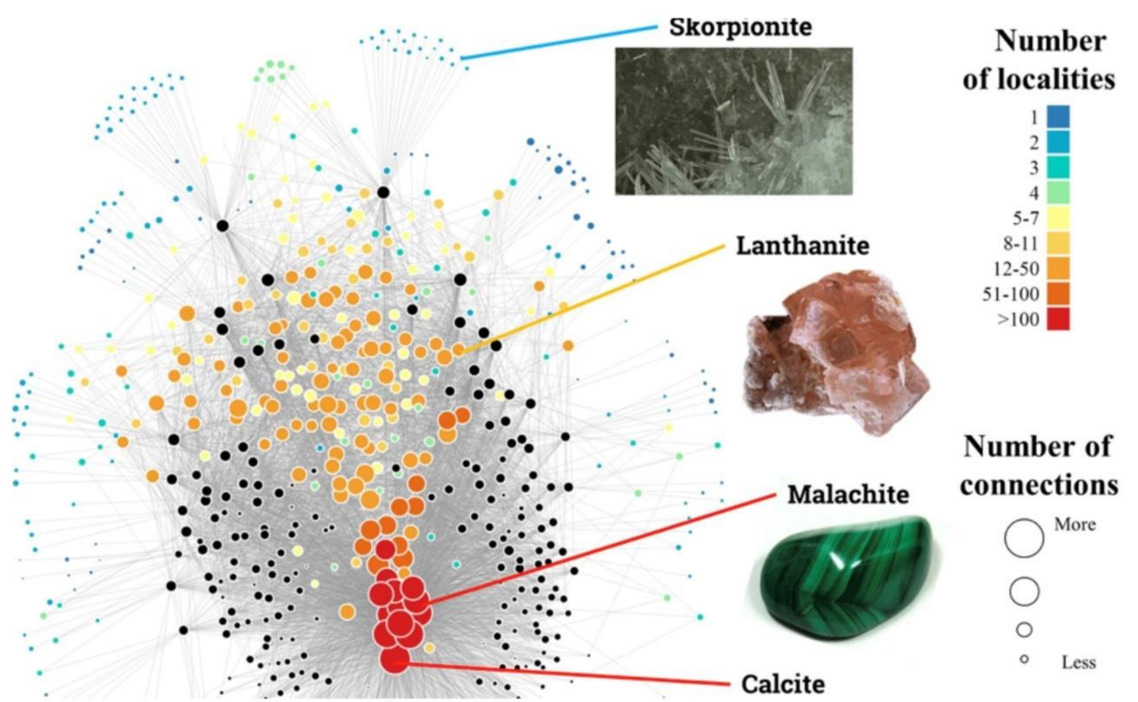
Data → gold

MINING.com NEWS MARKETS PEOPLE GROUPS JOBS COURSES MORE

How big data can help find new mineral deposits

Valentina Ruiz Leotaud | Aug. 2, 2018, 4:11 PM |

PEOPLEMINE FACEBOOK LINKEDIN TWITTER EMAIL PRINT



http://www.mining.com/big-data-can-help-find-new-mineral-deposits/

Data → fun and profit

The New York Times

When Sports Betting Is Legal, the Value of Game Data Soars

A trader working at William Hill, an international sports betting book, in Las Vegas.

Bridget Bennett for The New York Times

https://www.nytimes.com/2018/07/02/sports/sports-betting.html

Data → power

POLITICO

Cambridge Analytica whistleblower Chris Wylie speaks during a press conference at the Frontline Club on March 26, 2018 in London | Dan Kitwood/Getty Images

Cambridge Analytica helped 'cheat' Brexit vote and US election, claims whistleblower

Giving evidence to MPs, Chris Wylie claimed the company's actions during the Brexit campaign were 'a breach of the law.'

By **MARK SCOTT** | 3/27/18, 5:46 PM CET | Updated 3/29/18, 9:18 PM CET

https://www.politico.eu/article/cambridge-analytica-chris-wylie-brexit-trump-britain-data-protection-privacy-facebook/

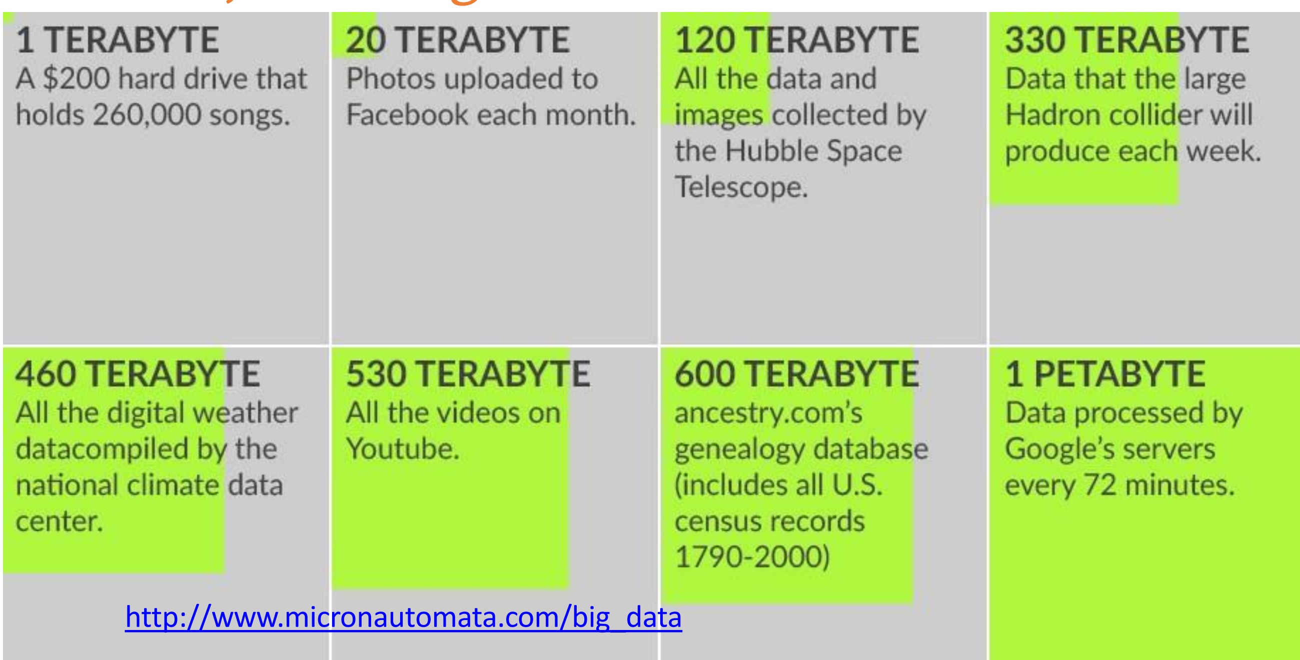
Democratizing data (and analysis)

- Democratization of data: more data—relevant to you and the society—are being collected – make it available to all
  - "Smart planet" IBM's vision
  - Government of Sunshine – Laws and regulations requiring transparency and disclosure in government and business.

- But few people know how to analyze them

Challenges

- Moore's Law:
  Processing power doubles every 18 months
- But amount of data doubles every 9 months
  - Disk sales (# of bits) doubles every 9 months
  - Parkinson's Law: *Data expands to fill the space available for storage*



http://www.micronautomata.com/big_data

Moore’s Law reversed

Time to process all data
doubles every 18 months!

- Data doubles in every 9 months, is it ok?
  - No, so we need smarter data management and processing techniques

So, what is a database system?

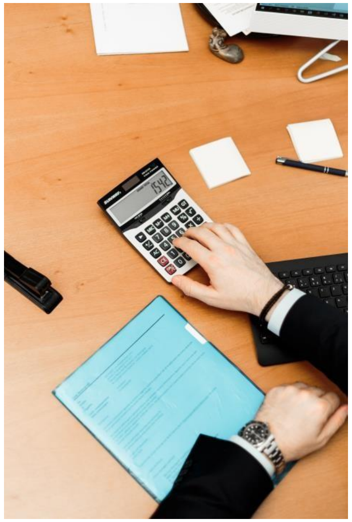
- Database:
  - a collection of interrelated data
  - often organized in a certain structure for convenient and efficient access

- Database system, DataBase Management System (DBMS): is a software system for convenient and efficient data access (creation, maintenance and use) over electronic databases.

How to manage a database?

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

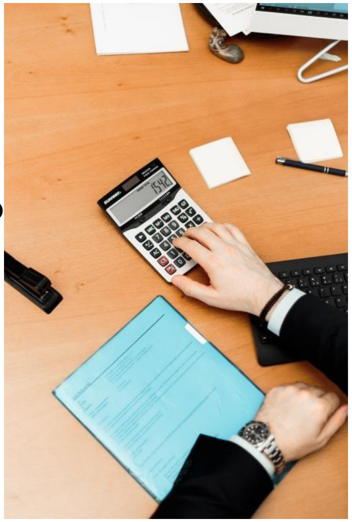
  - Step 2: do some analysis
    - How much I spend on grocery and fast food in February?
    - How much could I have saved if I cook by myself in February?
    - What about January/last quarter/last year/past five years?



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 2: do some analysis
    - How much did my spend on grocery and fast food in February?
    - How much could I have saved if I cook by myself in February?
    - 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
```

```
f = open('myspend_feb_22.txt', 'r')
grocery = 0
fast_food = 0

    date, amount, desc = line.split(' ')
    if desc == 'Fast food':
        fast_food=fast_food + amount
    elif desc == 'Grocery':
        grocery = grocery + amount
……
```

  - Step 2: do some analysis
    - How much did my spend on grocer
    - How much could I have saved if I c
    - What about January/last quarter/la

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 2: 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/past five years?

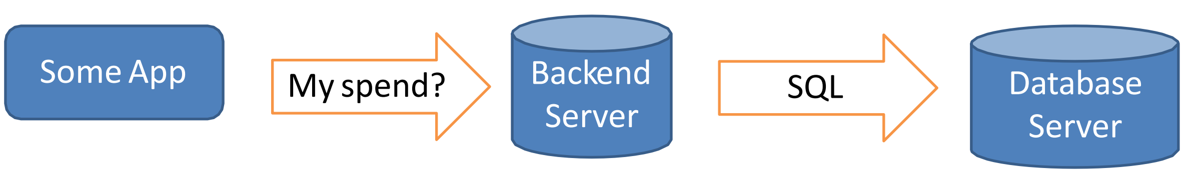
How to manage a database?

- 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~~
    ~~use a spreadsheet~~
    use some personal finance app

```
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;
```

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



Runs a DBMS

Brief History of Databases

History of Database systems

- Manual Systems: 1950s
  - Data was stored as paper records
  - Significant man-power and inefficient

- Magnetic tapes: till early 1960s
  - punch cards for input
  - Provides only sequential access – inefficient

- Late 1960s and 1970s:
  - Hard disk allows direct data access – fast
  - 1968- Data stored in files (File-based system)
    - Security issues – data redundancy

History of Database systems

- Late 1960s and 1980s: non-relational databases
  - Hierarchical DB – IBM's first Info. Mgmt. sys. (IMS)
  - Network DB – Charles Bachmann's first DBMS named IDS (Integrated Data Store) (at Honeywell)
    - leads to CODASYL- Conf. on Data Systems Language
      - Developed programming language COBOL
    - CODASYL+DBTG proposed 1st Net. model- IDMS (Information Data Mgmt. System)- popular

History of Database systems

- 1080-present: Relational Database
  - Ted Codd defines Relational Data Model
  - 2 prototypes
    - IBM Research begins System R prototype lead to DB2
    - UC Berkeley begins Ingres prototype followed by PostgreSQL
  - P.Chenn proposed ER Model for DB design – conceptual model later implemented into Relational model

History of Database systems

- 1080s: Relational Database
  - SQL becomes industry standard (ISO & ANSI)
  - Parallel and Distributed database
  - Object-Oriented databases
- 1990s:
  - Large databases/warehouses
  - Data-mining Applications
  - Emergence of Web commerce
- Early 2000s:
  - XML and XQuery standards
  - Automated database administration

New Database Applications

- Database apps. – Scientific apps. –large amount of data
  - Storage and retrieval of images
  - Storage and retrieval of videos
  - Data Mining applications
  - Weather data applications
  - Time-series applications –stores data on regular interval

- Rel DB not suited for the above apps-complex Data Structures, New data types, New query language, new indexing techniques

History of Database systems

- Later 2000s:
  - NoSQL term introduced in 1998- unstructured data – no schema – distributed DB – highly scalable
    - Ex: MongoDB, CouchDB, IBM's DB2, Neo4J, HBase, OrientDB, Redis, IBM Cloudant, RevenDB, Google BigTable (2004), Yahoo PNuts, Amazon Dynamo (2007), Apache Cassandra (2008),…

- NewSQL (2011): NoSQL + Relational DB (ACID properties)
  - Ex: CockroachDB, Spanner DB, Apache Trafodian, NeoDB, ClustrixDB

Most used DBMSs

1. **MySQL-** is one of the most commonly used SQL databases, popular choice for developers working with structured datasets, that won’t change much over time. Not always suitable for applications that require flexibility and mobility (the amount and type) of data that is being stored.

2. **PostgreSQL -** It is also a SQL database, deals with the structured databases. Unlike other SQL databases, it is compatible with JSON, so it accommodates more data types than just structured data. It ensures that the application can run smoothly as the database grows.

3. **MongoDB -** one of the most popular NoSQL databases, It is a document-based database (JSON-like documents), proficient with the JSON data-interchange format, useful for large set of distributed data (cloud apps), supports flexible and scalable data.

4. **AWS DynamoDB** a NoSQL database, known for its efficiency for the retrieval of information and data. In contrast to MongoDB, it supports both document and key-value data structures, limited support for different datatypes. Like other databases, DynamoDB is a highly scalable and complex database system for developers to manage big and dynamic data.

5. **Couchbase** – an open-source, distributed multi-model NoSQL document-based cloud database. An excellent option for apps that require a database that is scalable and changeable over time.

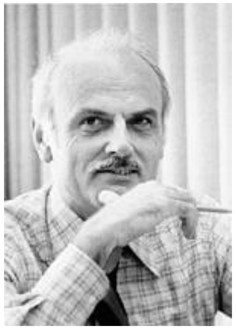
NewSQL: ClustrixDB, NuoDB, CockroachDB

How to design a DBMS?

3 Turing Award Winners!

- Charles Bachman, 1973

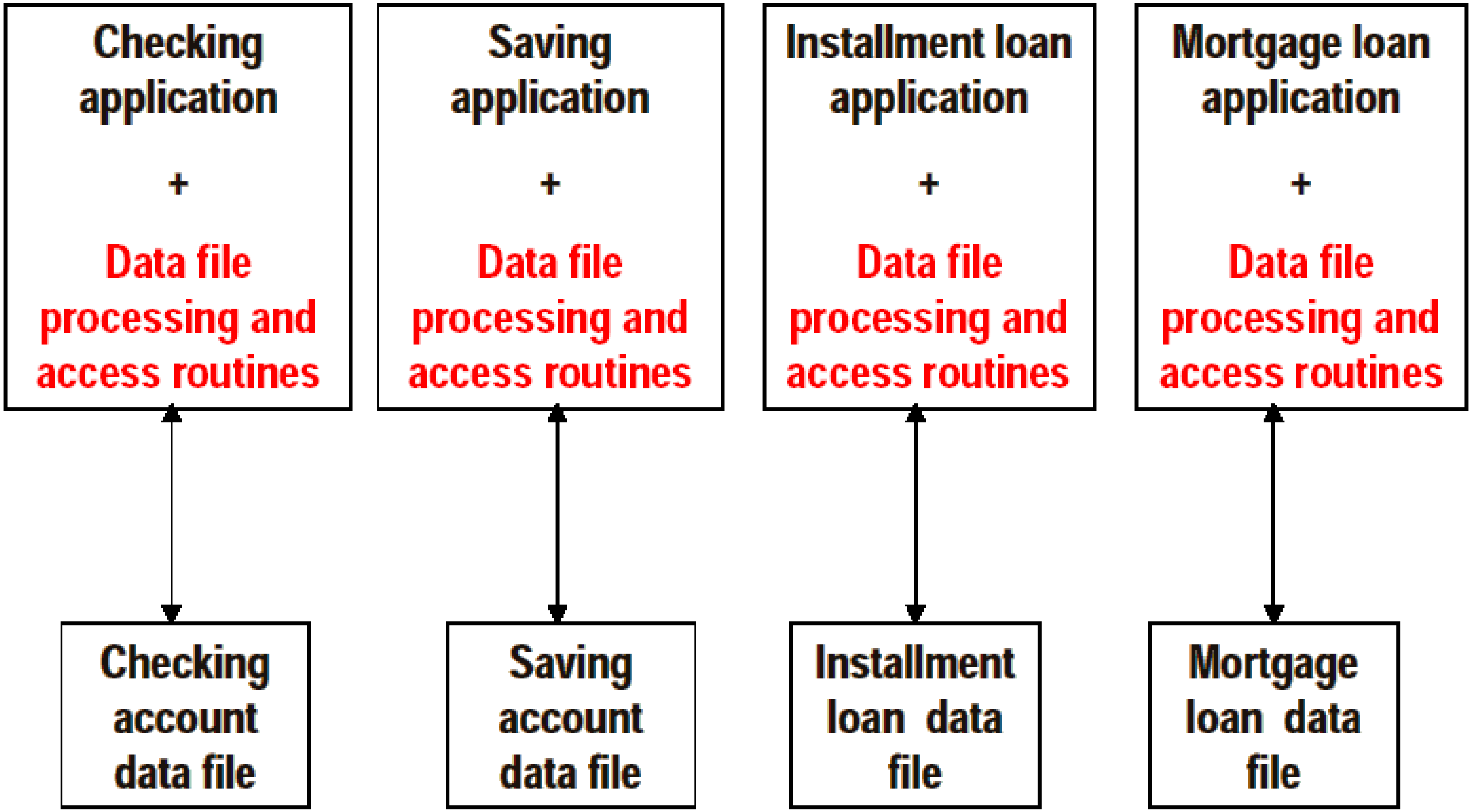
- Edgar F. Codd, 1981



- Michael Stonebraker, 2014

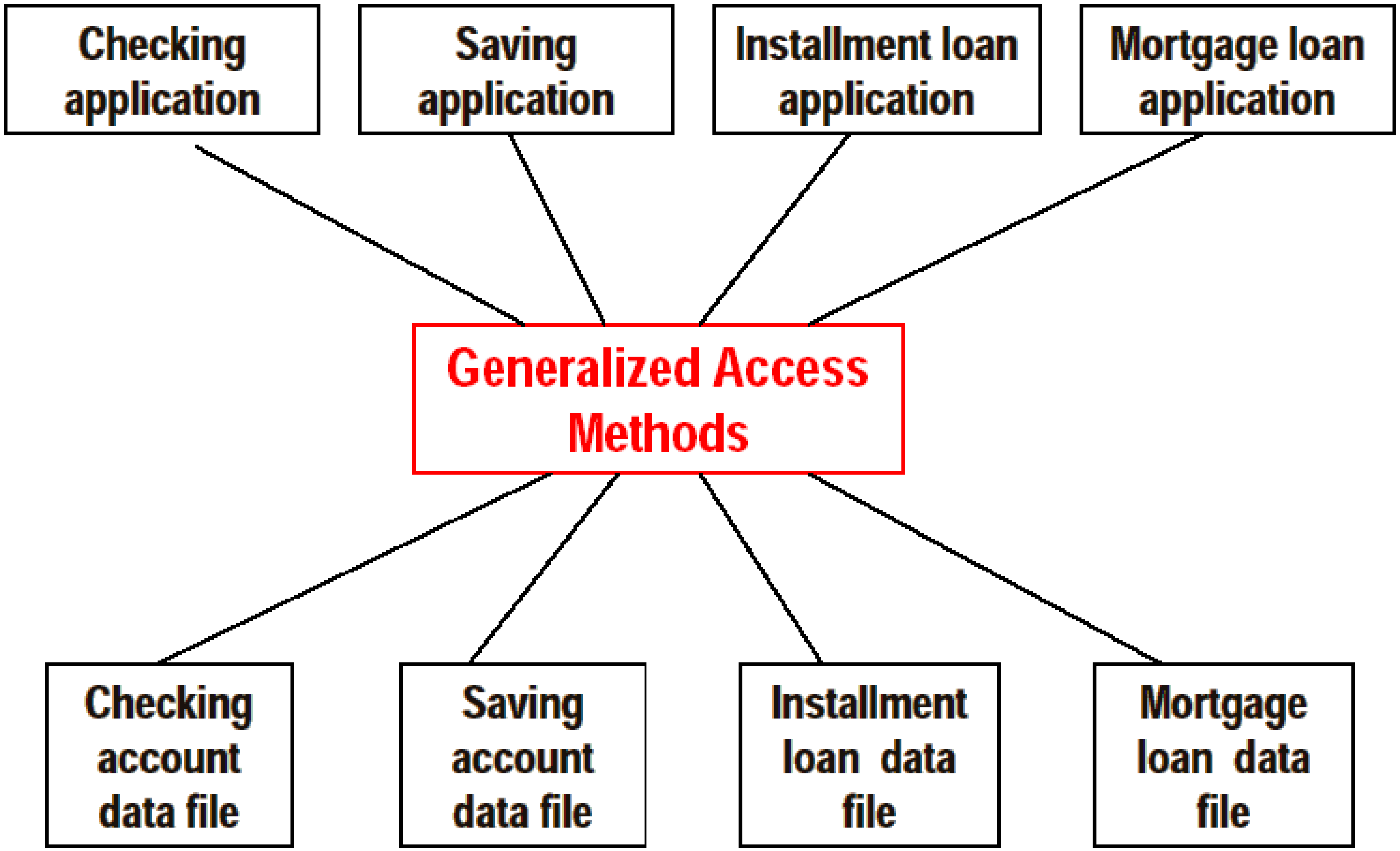
https://amturing.acm.org/award_winners/bachman_9385610.cfm
https://en.wikipedia.org/wiki/Edgar_F._Codd
https://en.wikipedia.org/wiki/Michael_Stonebraker

The birth of DBMS – 1



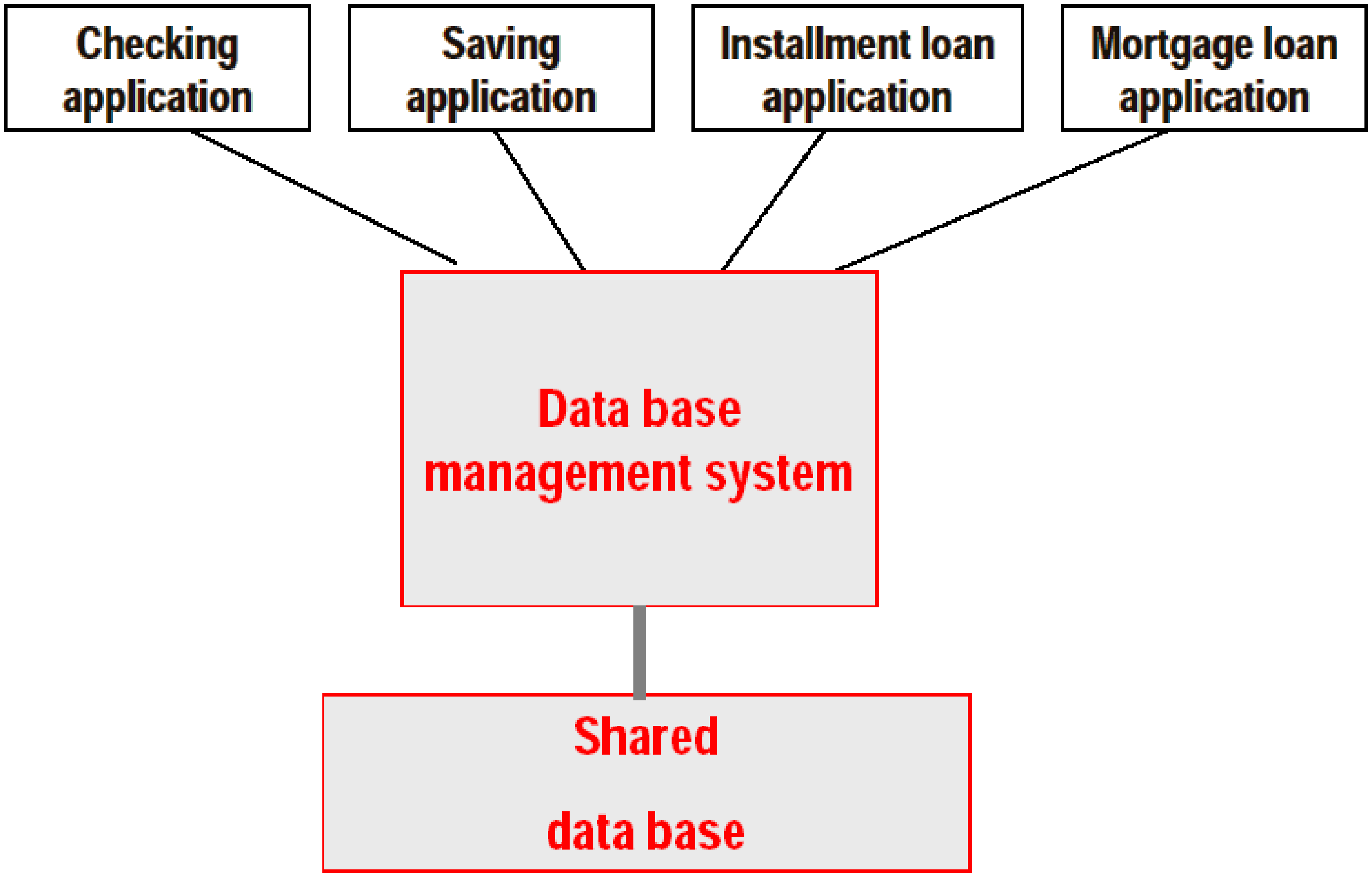
From Hans-J. Schek's *VLDB* 2000 slides

The birth of DBMS – 2



From Hans-J. Schek's *VLDB* 2000 slides

The birth of DBMS – 3



From Hans-J. Schek's *VLDB* 2000 slides

Early efforts

- "Factoring out" data management functionalities from applications and standardizing these functionalities is an important first step
  - CODASYL standard (circa 1960's)

  ☞ Bachman got a Turing award for this in 1973

- But getting the abstraction right (the API between applications and the DBMS) is still tricky

https://amturing.acm.org/award_winners/bachman_9385610.cfm

Continue with our bank example…

- Query: Who have accounts with 0 balance managed by a branch in Ahmedabad?

- Pseudo-code of a CODASYL application:

```
Use index on account(balance) to get accounts with 0 balance;
For each account record:
        Get the branch id of this account;
        Use index on branch(id) to get the branch record;
        If the branch record's location field reads "Ahmedabad":
                Output the owner field of the account record.
```

What's wrong?

With the CODASYL approach, to write correct & efficient code, programmers need to

- know how data is organized physically
- worry about data/workload characteristics

The relational revolution (1970's)

- A simple model: data is stored in relations (tables)

Account_id	name	balance	Branch_id
142	Amit	10000	2
123	Shreya	0	1
…	…	…	…

Branch_id	location
1	Delhi
2	Ahmedabad
…	…

- A declarative query language: SQL

```
SELECT Account.owner
    FROM Account, Branch
    WHERE Account.balance = 0
        AND Branch.location = ' Ahmedabad'
        AND Account.branch_id = Branch.branch_id;
```

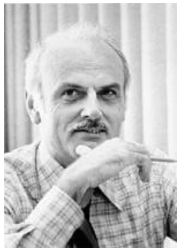
The relational revolution (1970's)

- Programmers specifies what answers a query should return, but not how the query is executed

- DBMS picks the best execution strategy based on physical structure of the data, etc.

☞ Provides physical data independence

- And a Turing Award for E. F. Codd in 1981

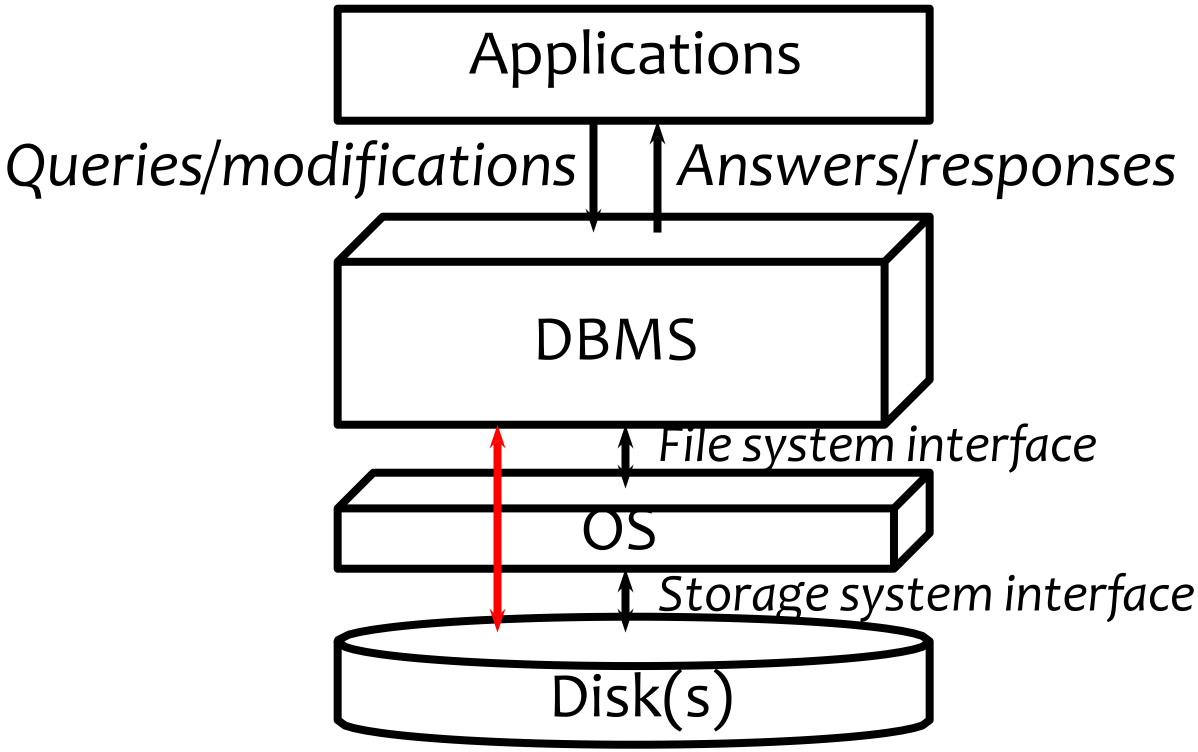


https://en.wikipedia.org/wiki/Edgar_F._Codd

Standard DBMS features

- Logical data model; declarative queries and updates → physical data independence
- Multi-user concurrent access; persistent storage of data; safety from system failures
- Performance, performance, performance

Standard DBMS architecture

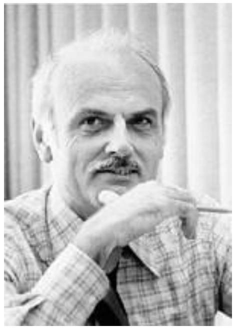


- Much of the OS may be bypassed for performance and safety

Modern DBMSs

- Charles Bachman, 1973

- Edgar F. Codd, 1981



- Michael Stonebraker, 2014

Relational DBMS (e.g. Ingres, Postgres) and modern DBMSs (e.g. C-store, H-store)

https://amturing.acm.org/award_winners/bachman_9385610.cfm
https://en.wikipedia.org/wiki/Edgar_F._Codd
https://en.wikipedia.org/wiki/Michael_Stonebraker