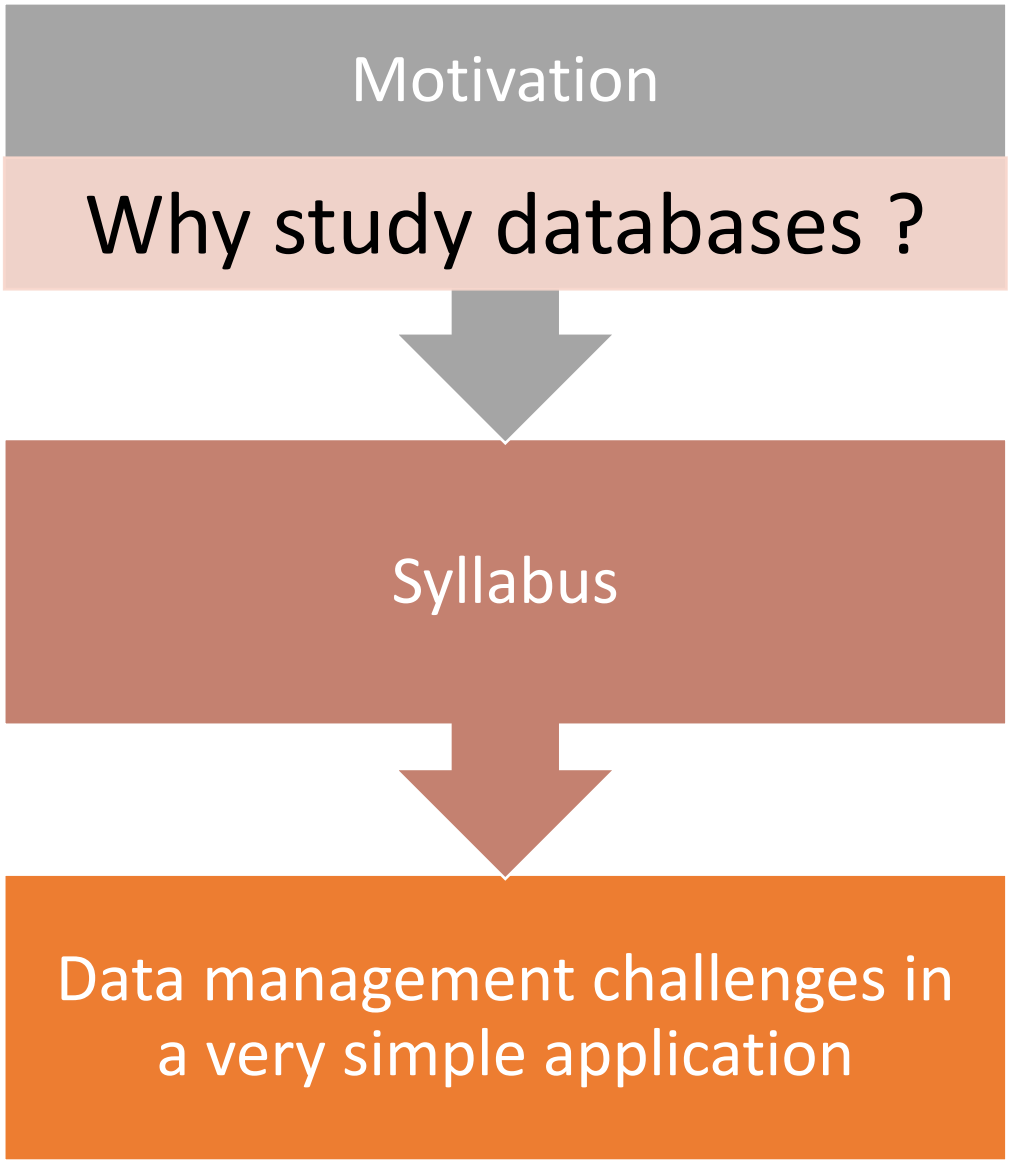
Introduction to Database Management Systems

Wen-Chih Peng (彭文志)
wcpengcs@nycu.edu.tw

Outline


Motivation
Why study databases ?
Syllabus
Data management challenges in a very simple application

We are in digital worlds

Web

Multimedia data

- Youtube, photo sharing

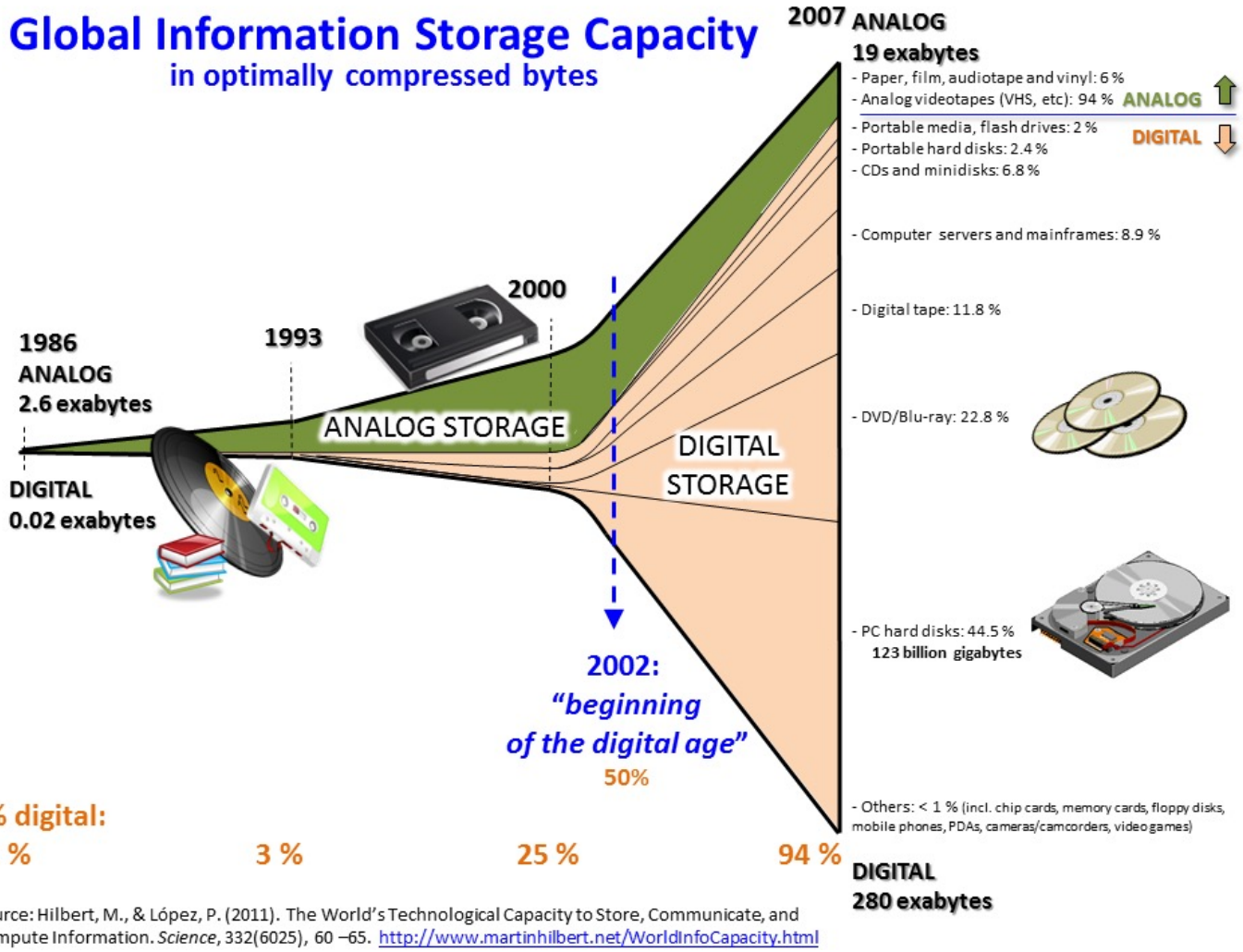
Social media services

- Facebook, Line, WeChat

IoT (Internet of Things)

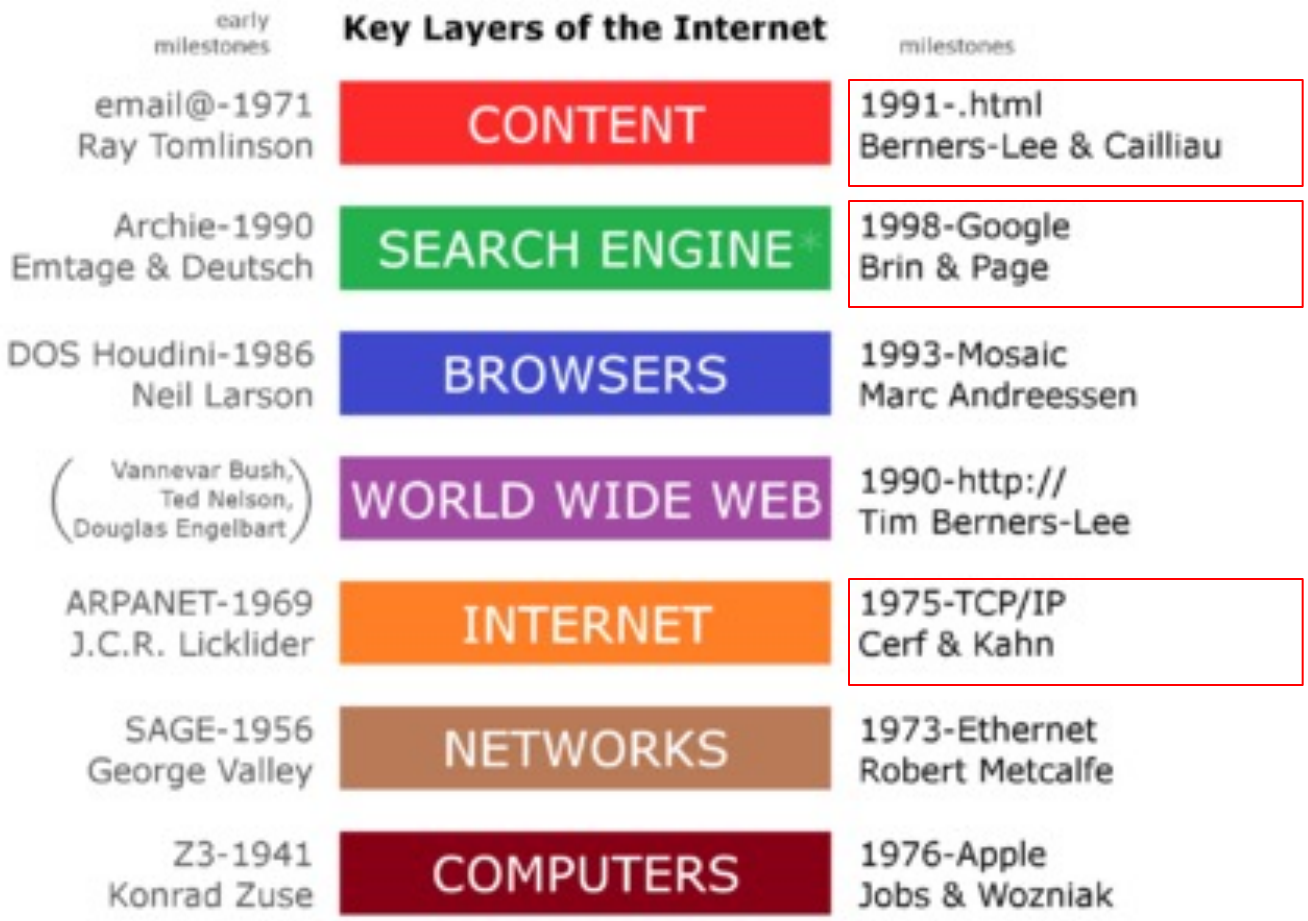
- The network of physical objects or "things”embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the production, operator and/or other connected devices (From Wikipedia)

We are data rich
Driving force – Digital Storage



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

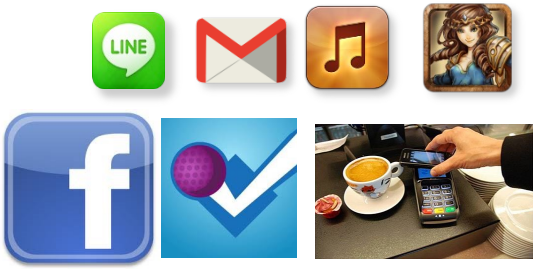
We are data rich
Driving force – Internet/Web



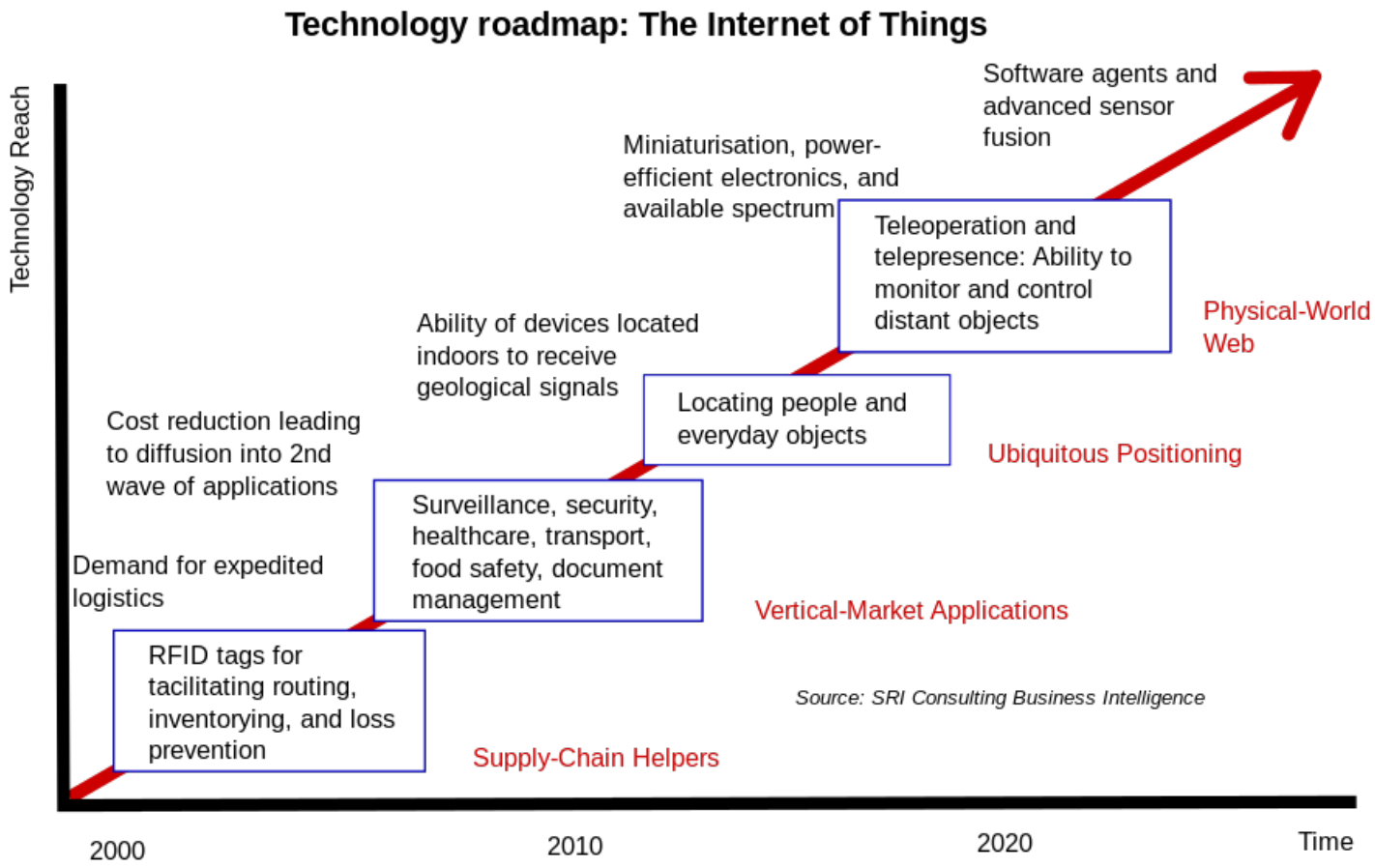
We are data rich
Driving force – Not only phones, always on-line

Time: 1983
Phones, SMS

iPhone
Time: 2007
Email, Safari



More data in the coming years
Everything is on-line



Motivation: Data Overload

- Huge amount of data in this world
- Everywhere you see…
  - Personal (emails, data on your computer)
  - Enterprise
    - Banks, supermarkets, universities, airlines
  - Scientific (biological, astronomical)

Web data: Amazon reviews

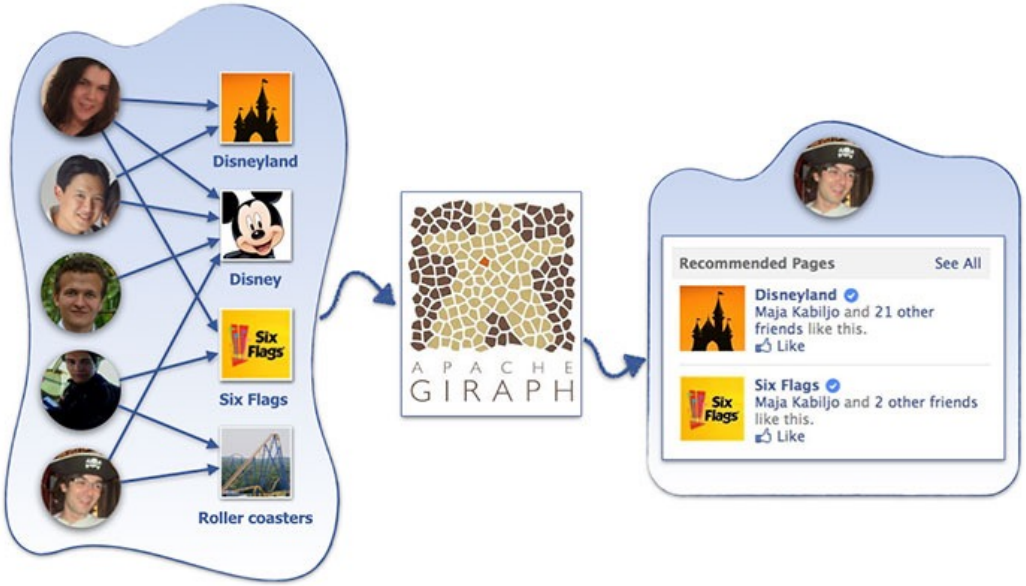
- Dataset is provided by SNAP (https://snap.stanford.edu/data/web-Amazon.html)

  - Number of reviews 34,686,770
  - Number of users 6,643,669
  - Number of products 2,441,053
  - Users with > 50 reviews 56,772
  - Median no. of words per review 82
  - Timespan Jun 1995 - Mar 2013

One simple task (look-like)

- Finding top 500 user pairs according to their purchasing behaviors (i.e., their buying products)
- Measuring purchasing behaviors by Jaccard similarity

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$



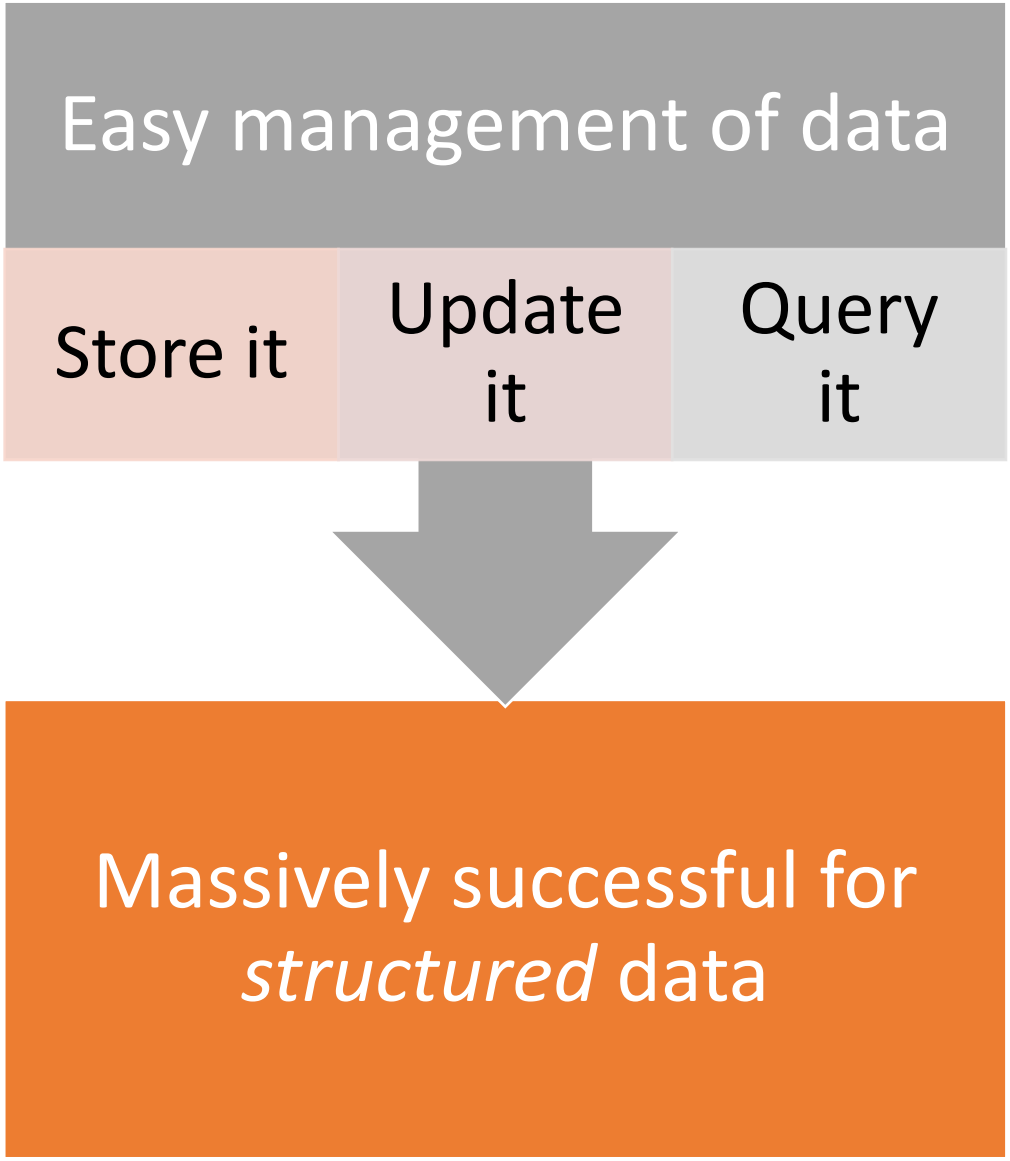
Another example

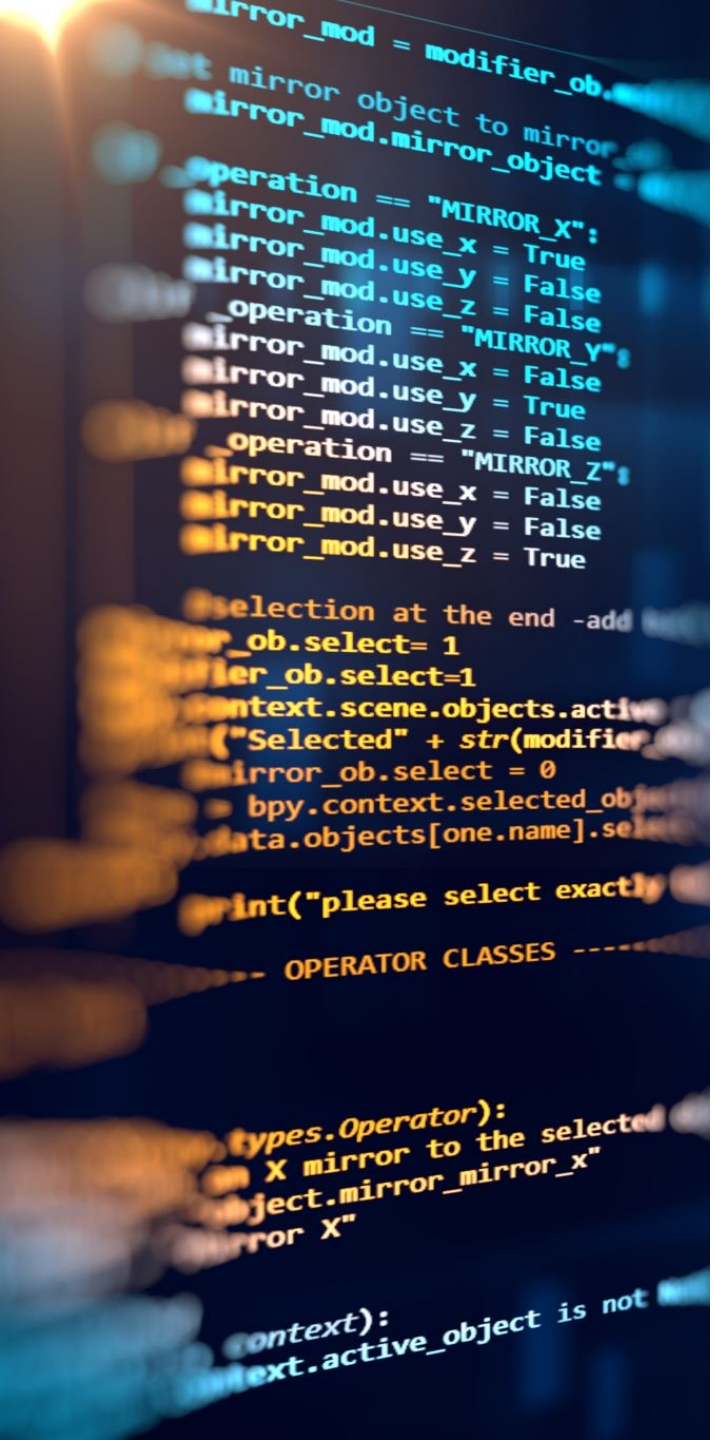
- Airline on-time performance (http://stat-computing.org/dataexpo/2009/)
- The data consists of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. This is a large dataset: there are nearly 120 million records in total, and takes up 1.6 gigabytes of space compressed and 12 gigabytes when uncompressed.

Some query tasks

- When is the best time of day/day of week/time of year to fly to minimize delays?
- Do older planes suffer more delays?
- How does the number of people flying between different locations change over time?
- How well does weather predict plane delays?
- Can you detect cascading failures as delays in one airport create delays in others? Are there critical links in the system?

DBMS to the Rescue





Materials covered

- data modeling
- database languages
  - SQL
- relational database design principles
- file system organizations
  - indexing methods
  - query optimization
- transaction processing
- recovery mechanisms
- concurrency control

Course Information

- Instructor: Wen-Chih Peng
  - Office: EC 542
  - Email: wcpengcs@nycu.edu.tw
  - TAs:
    - 柯秉志 jklzxcvbnm1225.cs11@nycu.edu.tw
    - 簡言哲 dfg15243.cs12@nycu.edu.tw
    - 羅名志 max230620089@gmail.com
  - Class Webpage:
    - New e3.nctu.edu.tw

Course Information

- Reference Textbooks:
  - Fundamentals of Database Systems
    - Seventh edition
    - Ramez Elmasri, Shamkant B. Navathe
  - Database Systems: A Practical Approach to Design, Implementation, and Management
    - 6th Edition
    - Thomas Connolly, Carolyn Begg
  - Database System Concepts
    - Sixth Edition
    - Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- ChatGPT is your virtual TA ☺

Grading (Tentative)

- Workload:
  - 1 warm up + 3 assignments (5% + 3*15%)
  - 2 in-class exam (15% + 15%)
  - 1 final project (20%)
  - Class participation (5% bonus)

CSCS10022: Introduction to Database Systems

Spring 2023 Fall

課程內容大綱	Weeks	搭配作業
Introduction to DB	1 Sep. 14	HW0 announcement and start to find you final project team
Database System Concepts & Relational Model	2 Sep. 21	
Relational Algebra	3 Sep. 28	
SQL	4 Oct. 5	HW0 deadline HW1 announcement
SQL (cont.)	5 Oct. 12	Final project match deadline
SQL (cont.) & Seq2sql	6 Oct. 19	HW1 deadline
Storage & Query Processing	7 Oct. 26	HW2 announcement
Query Processing (cont.)	8 Nov. 2	
Midterm Exam	9 Nov. 9	
Index Structures	10 Nov. 16	HW2 deadline
Transactions.	11 Nov. 23	Proposal deadline HW3 announcemen
Concurrency	12 Nov. 30	
ER Model, Relational Model	13 Dec. 7	HW3 deadline
Normal Form	14 Dec. 14	
Final Exam	15 Dec. 21	
Final Project Presentation（線上）	16 Dec. 28	
Final Project Presentation（線上）	17 Jan. 4	Final Project Deadline
彈性上課	18 Jan. 11	

加簽方式

- 填寫 Google 表單
- 只有現在在場學生可以參加加簽
- 待會下課**至教室前方找助教簽名**
- 原則上
  - 急需學分者優先
  - 高年級優先
  - 先備知識滿足者優先
- 9/22 23:00 前將加入成功加簽者
  - 加簽完成後將寄信通知

https://forms.gle/3TXLDWUPys3UPaPz7

One example

- Data management challenges in a very simple application
  - Why we can’t use a file system to do database management

Example

Simple Banking Application

Need to store information about:

- Accounts
- Customers

Need to support:

- ATM transactions
- Queries about the data

Instructive to see how a naïve solution will work

A *file-system* based solution

- Data stored in files in ASCII format
  - #-seperated files in /usr/db directory
  - /usr/db/accounts

    Account Number # Balance

    101 # 900

    102 # 700

    …
  - /usr/db/customers

    Customer Name # Customer Address # Account Number

    Johnson # 101 University Blvd # 101

    Smith # 1300 K St # 102

    Johnson # 101 University Blvd # 103

    …

A *file-system* based solution

- Write application programs to support the operations
  - In your favorite programming language
  - To support withdrawals by a customer for amount $X from account Y
    - Scan /usr/db/accounts, and look for Y in the 1st field
    - Subtract $X from the 2nd field, and rewrite the file
  - To support finding names of all customers on street Z
    - Scan /usr/db/customers, and look for (partial) matches for Z in the addess field

What's wrong with this solution ?

- Data redundancy and inconsistency
  - No control of *redundancy*

Customer Name # Customer Address # Account Number

Johnson # 101 University Blvd # 101

Smith # 1300 K St # 102

Johnson # 101 University Blvd # 103

  - Inconsistencies
    - Data in different files may not agree
    - Very critical issue

What's wrong with this solution ?

- Evolution of the database is hard
  - Delete an account
    - Will have to rewrite the entire file
  - Add a new field to the *accounts* file, *or*
    split the *customers* file in two parts:
    - Rewriting the entire file least of the worries
    - Will probably have to rewrite all the application programs

What's wrong with this solution ?

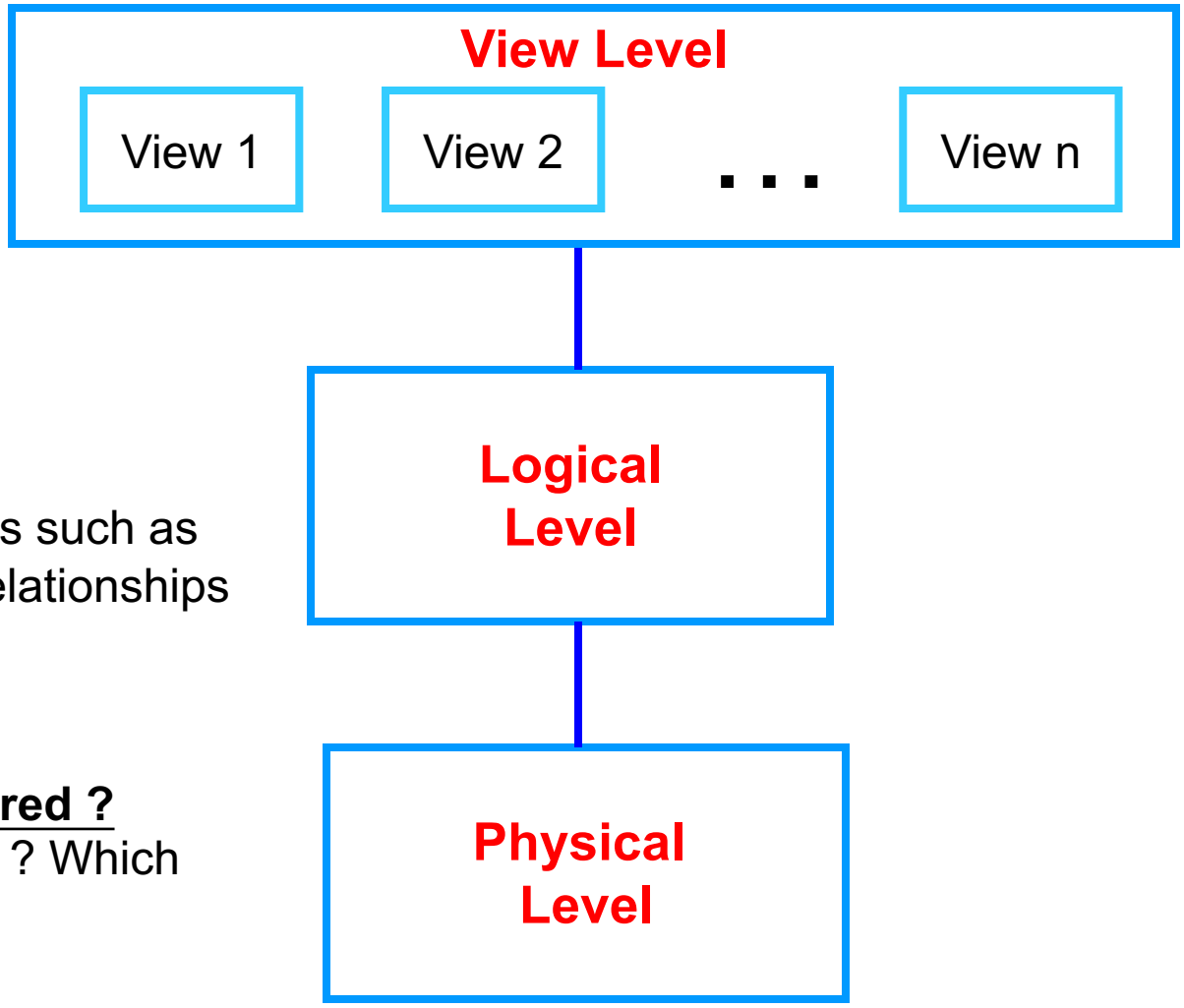
- Difficulties in Data Retrieval
  - No sophisticated tools for selective data access
    - Access only the data for customer X
    - Inefficient to scan the entire file
  - Limited reuse
    - Find customers who live in area code 301
    - Unfortunately, no application program already written
    - Write a new program every time ?

What's wrong with this solution ?

- Semantic constraints
  - Semantic integrity constraints become part of program code
    - *Balance* should not fall below 0
    - Every program that modifies the *balance* will have to enforce this constraint

Data Abstraction

What data users and application programs see ?



What data is stored ?

describe data properties such as data semantics, data relationships

How data is actually stored ?

e.g. are we using disks ? Which file system ?

Data Abstraction: Banking Example

- Logical level:
  - Provide an abstraction of *tables*
  - Two tables can be accessed:
    - *accounts*
      - Columns: account number, balance
    - *customers*
      - Columns: name, address, account number
- View level:
  - A teller (non-manager) can only see a part of the *accounts* table
    - Not containing high balance accounts

Customer-Name	ID	customer-street	customer-city

Data Abstraction: Banking Example

- Physical Level:
  - Each table is stored in a separate ASCII file
  - # separated fields

- Identical to what we had before ?
  - BUT the users are not aware of this
    - They only see the tables
    - The application programs are written over the tables abstraction
  - Can change the physical level without affecting users
  - In fact, can even change the logical level without affecting the *teller*

DBMS at a glance

- Data Models
  - Conceptual representation of the data
- Data Retrieval
  - How to ask questions of the database
  - How to answer those questions
- Data Storage
  - How/where to store data, how to access it
- Data Integrity
  - Manage crashes, concurrency
  - Manage semantic inconsistencies

Advantages of DBMS

- Data independence
- Efficient data access
- Data integrity and security
- Data administration
- Concurrent access and crash recovery

Overall:

Reduced application development time and cost

Some videos about DB

What's database management ?

3 Mins to tell you the concept of DB

Application about using DB in sport team management

Lessons learned from DB