ECE30030/ITP30010

Database Systems

Charmgil Hong

charmgil@handong.edu

Spring, 2023

Handong Global University

Agenda

- Course Overview
- Course Motivation
- Administrivia

Course Overview

- Course: ECE30030/ITP30010 Database Systems
  - ITP-section#1 (English): Mon/Thu 10:00-11:15am @OH401
  - ITP-section#2 (English): Mon/Thu 11:30am-12:45pm @OH401
  - ECE-section (Korean): Tue/Fri 2:30-3:45pm @OH401

- Instructor: Charmgil Hong (홍참길)
  - Office: NTH201
  - Email: charmgil@handong.edu
  - Office hours: TBD

- Teaching Assistants: TBD

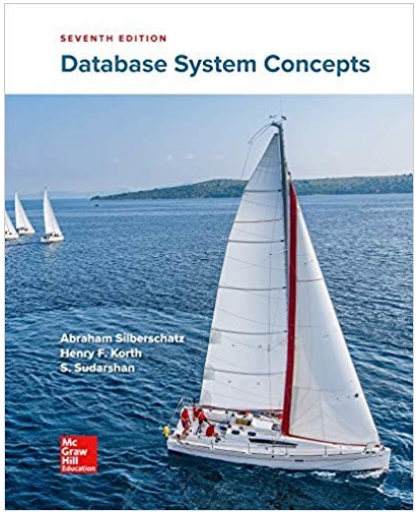
Course Overview

- Course objectives:
  - Students understand the concepts and underlying mechanisms of database management system
  - Students can represent database designs in modeling languages and analyze the designs with respect to given constraints
  - Students can articulate the relational database language (structured query language)
  - Students can extract, transform, and load data from a database and analyze it using modern algorithms

Course Overview

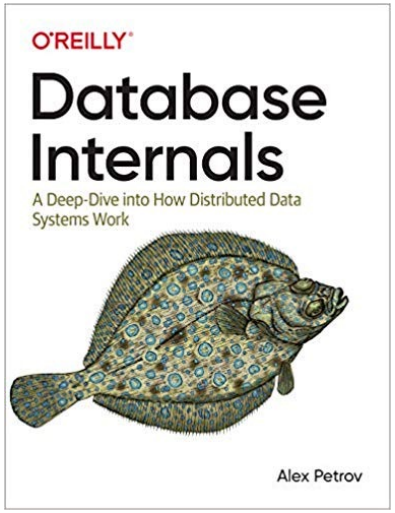
- Prerequisite: Discrete Math (이산수학)

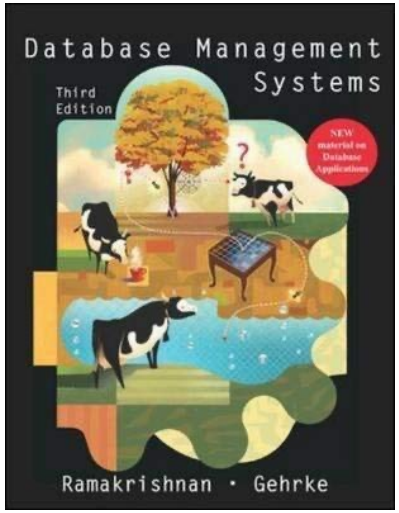
- Main text
  - Abraham Silberschatz, Henry F. Korth, S. Sudarshan. ***Database System Concepts, 7th edition.*** McGraw Hill. 2019.



Course Overview

- Supplementary reference
  - Alex Petrov. ***Database Internals: A Deep Dive into How Distributed Data Systems Work.*** O'Reilly Media. 2019.
  - Raghu Ramakrishnan, Johannes Gehrke. ***Database Management Systems, 3rd edition.*** McGraw Hill. 2003.





Course Overview

- This course is offered in English (ITP30010 only)
  - Use English in your homework, exams, and all communication in class
  - You will get no credit for your submission if it is not in English

GETTING STARTED

Motivation

- Database
  - Organized collection of inter-related data that models some aspect of the real-world
  - Databases are one of the core components of most computer applications
    - Examples (next slides)

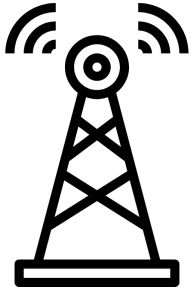
Motivation

- Database examples
  - Universities
    - Registration, grades
  - Financial market
    - Credit card transactions
    - Sales and purchase information of stocks and bonds
    - Real-time market data
  - Enterprise information
    - Sales: customer products, purchases
    - Accounting: payments, receipts, assets
    - Human resources: employee profile, salaries, taxes

* Image src: https://pngimage.net/university-png-6/; http://www.pngpix.com/download/money-us-dollars-png-transparent-image; https://stephengashler.com/icon-enterprise/

Motivation

- Database examples
  - Airlines
    - Reservations, schedules
  - Web services
    - Online commerce: customer information, product information, order tracking
    - Online advertisement
  - Telecommunication
    - Call records
    - Texts
    - Data usage
    - Monthly bills



* Image src: http://pregem.com/airline-icon-4/; https://icon-icons.com/icon/globe-www/52833; https://www.flaticon.com/free-icon/telecommunications_1124729

Motivation

- In the early data, database applications were built directly on top of file systems, which leads to:
  - **Data redundancy and inconsistency**
    - Data is stored in multiple file formats and locations
      → Resulting induplication of information
  - **Difficulty in accessing data**
    - Need to write a new program to carry out each new task
  - **Data isolation**
    - Meaning: a property that determines when and how changes made by one operation become visible to others
    - Cannot be controlled with files

Motivation

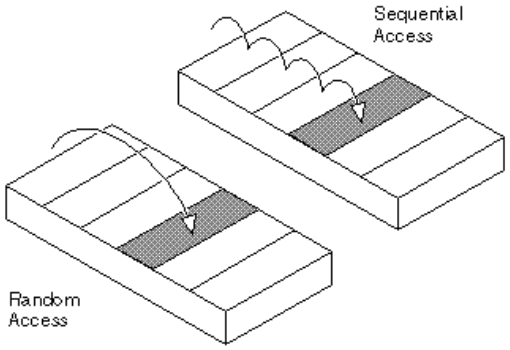
- In the early data, database applications were built directly on top of file systems, which leads to (cont'd):
  - **Integrity problems**
    - Integrity constraints (*e.g.*, account balance >= 0) become "buried" in program code, rather than being stated explicitly
  - **Concurrency problems**
    - Uncontrolled concurrent accesses can lead to inconsistencies
  - **Security problems**
    - Hard to provide a fine-grained user access control

➔ **Database systems offer solutions to all the above problems!**

Brief History of Database Systems

- ~ early 1960s:
  - Data processing using magnetic tapes for storage
    - Tapes provided only sequential access
  - Punched cards for input

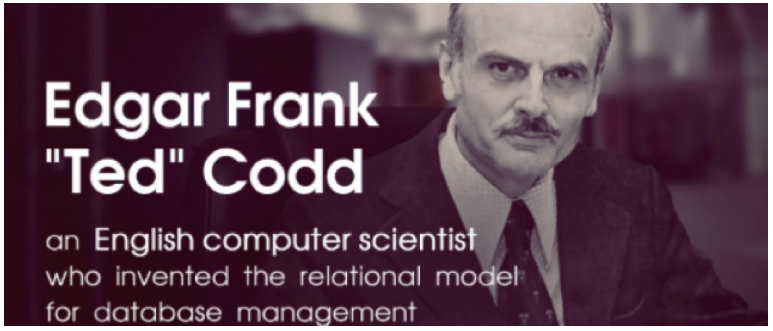
System 360 (IBM)

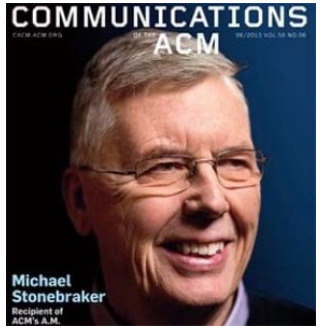


Random Access	Sequential Access
- Any data can be accessed randomly - Faster data retrieval - RAM, Hard disk, SSD, DVD, …	- Data must be accessed in order - Slower data retrieval - Tape drives

Brief History of Database Systems

- Late 1960s and 1970s:
  - Hard disks allowed direct access to data
  - Network and hierarchical data models in use
  - Ted Codd defined the relational data model
    - The work won the ACM Turing Award (1981)
    - IBM Research began System R prototype
    - UC Berkeley (Michael Stonebraker) began Ingres prototype
    - Oracle released first commercial relational database

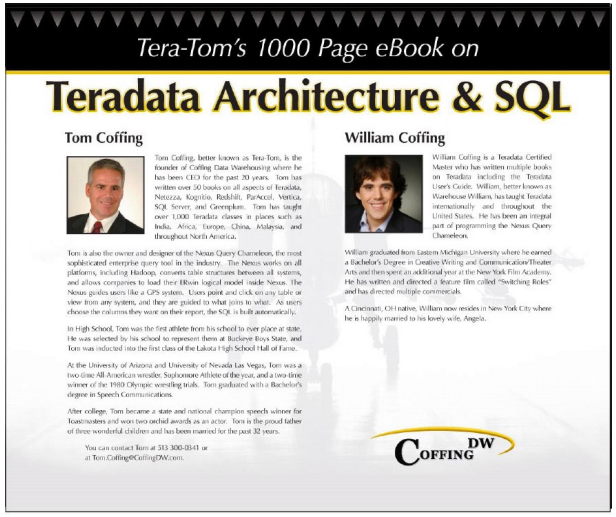




* Image src: https://www.tag-cyber.com/analysis/articles/finding-and-protecting-databases
https://www.voltdb.com/files/pre-recorded-webinar-stonebraker-says-fix-legacy-database-design-left-behind/

Brief History of Database Systems

- 1980s:
  - Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
  - Parallel and distributed database systems
    - Wisconsin, IBM, Teradata
  - Object-oriented database systems



* Image src: https://www.oreilly.com/library/view/tera-toms-1000-page/9781940540214/backcover.xhtml
https://landscape.cncf.io/selected=ibm-db2; https://www.flaticon.com/free-icon/sql_603201

Brief History of Database Systems

- 1990s:
  - Large decision support and data-mining application
  - Large multi-terabyte data warehouses
  - Emergence of Web commerce

* Image src: https://www.knkx.org/post/technology-predictions-2012; https://officechai.com/startups/amazon-first-office/

Brief History of Database Systems

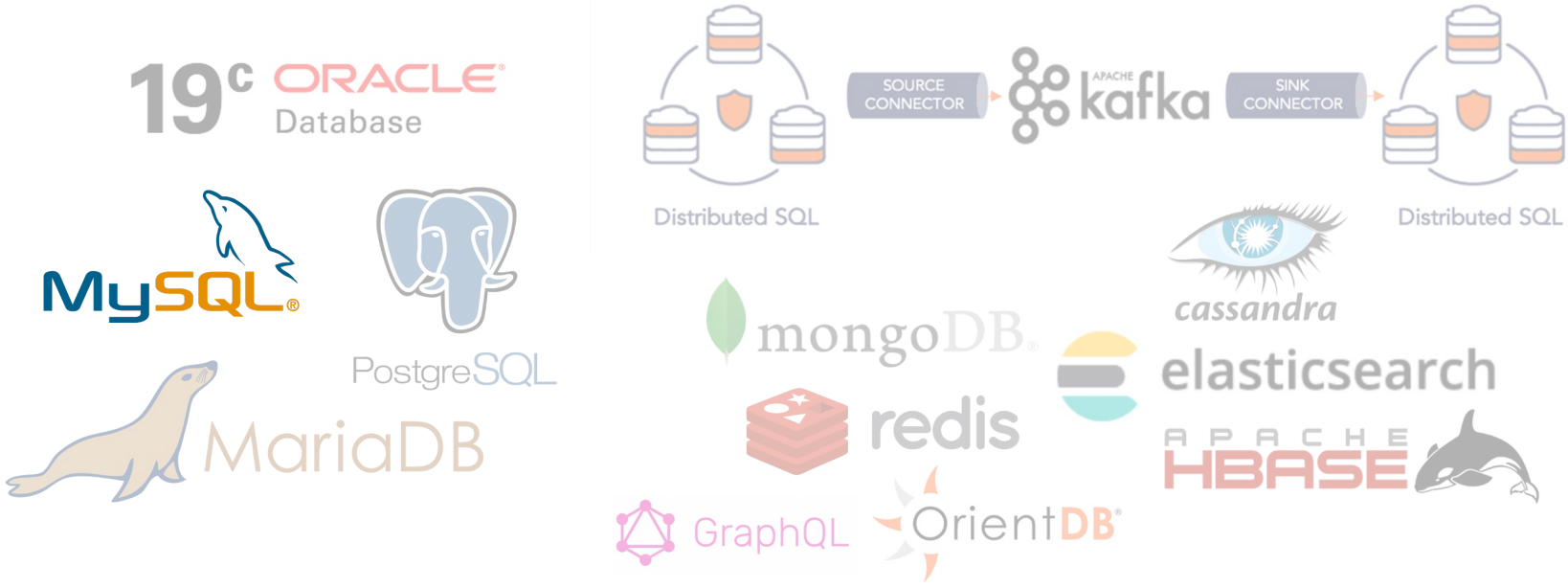
- 2000s:
  - Big data storage systems
    - Google BigTable, Yahoo PNuts, Amazon
    - NoSQL systems
  - Big data analysis: beyond SQL
    - Map reduce

Brief History of Database Systems

- 2010s:
  - SQL reloaded
    - SQL front-end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases

Brief History of Database Systems

- 2010s:
  - SQL reloaded
    - SQL front-end to Map Reduce systems
    - Massively parallel database systems
    - Multi-core main-memory databases



Schedule (tentative)

- We will study the next 4 big themes
  - Using R-DBMSs (from a developer's perspective)
  - Design and manipulation of data
  - Understanding the Internals of R-DBMSs
  - New Trends in Database Systems and Data Science

Schedule (tentative)

	Subject		Subject
1	Admin, Introduction DBMS, Relation data model	9	Keys, Functions/Procedures, Triggers
2	Relational algebra Installing a DBMS	10	Database storages
3	Structured Query Language (SQL DML)	11	Buffer management
4	Structured Query Language (SQL DDL)	12	Hashtables
5	Entity-Rleationship (ER) diagrams Normalization theory	13	Indexes
6	Advanced SQL Integrity constraints, Views	14	DB's for data analysis
7	Transactions	15	DB's for data analysis
8	Midterm	16	Final

Administrivia

- Grading
  - **Midterm: 20%**
  - **Final: 25%**
  - **Assignments: 25%**
    - Make sure to submit your work before each deadline
    - Late submissions will be accepted within 24 hours after the deadline with a penalty of -20% of the assignment grade
      - Submissions made after 24 hours from the deadline will be rejected
    - For additional extensions, reasonable excuse should be submitted before the deadline
  - **Term project: 23%**
  - **Participation: 7%**

Administrivia

- Grading
  - Assignments (25%)
    - There will be 5 to 6 homework assignments through the semester, consist of various activities
      - Short answer questions that that involve reading (textbook)
      - Problem solving
      - Programming assignment (in SQL)
  - Term project (23%)
    - Students will team up to design a database and optimize its performance
      - You are NOT required to build a DB application
      - You will receive a large chunk of data + target tasks
      - Your objective is to come up with a nice DB design and implementation that yield the best performance in class

Administrivia

- Attendance
  - Offline meetings are offered once a week
    - Pre-recorded video lectures will be provided on LMS
    - Offline meetings are held for discussion
  - Attendance does not directly go to your grade, but if you are absent for more than ¼ classes you will automatically get an “F”

Administrivia

- Policy related to COVID-19
  - Offline attendance is required for all students
  - One may ask for online attendance in case of proven illness
  - For in-class meetings
    - Open windows and ventilate before each class
    - No food nor drink is allowed in the classroom
    - Face masks are recommended

Administrivia

- Grading
  - Class participation (7%)
    - Offline meetings consist of student questions; students are highly encouraged to participate in and lead classes
      - Students are encouraged not only to ask questions but also to answer others' questions
    - You may speak in 한국어 during discussions
    - Every input you make counts towards your participation score

Administrivia

- Please review HGU CSEE Standard
  - 한국어: https://drive.google.com/file/d/0B9iQGS7v1k9ORGhXSHNyTkpvQW8/view
  - English: https://drive.google.com/file/d/0B9iQGS7v1k9Ob0oxTExmMjhPU28/view

Honor Code

- Please review HGU CSEE Standard
  - 한국어: https://drive.google.com/file/d/0B9iQGS7v1k9ORGhXSHNyTkpvQW8/view
  - English: https://drive.google.com/file/d/0B9iQGS7v1k9Ob0oxTExmMjhPU28/view

- Attendance
  - Marking her/his own attendance sheet without attending the class or marking other student's attendance sheet who is absent is regarded as cheating.
- Assignments
  - Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.
  - Showing or lending one's own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.
  - It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment.
- Team Project
  - It is cheating if a team uses the whole or part of the product that is not completed by team members except when it is clearly allowed.

Honor Code

- Please review HGU CSEE Standard
  - 한국어: https://drive.google.com/file/d/0B9iQGS7v1k9ORGhXSHNyTkpvQW8/view
  - English: https://drive.google.com/file/d/0B9iQGS7v1k9Ob0oxTExmMjhPU28/view

- 출석
  - 수업시간에 참석하지 않고 출석을 체크하는 어떠한 행위도 부정행위에 해당한다.
- 과제
  - … 인터넷 등에서 획득한 과제물 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
  - 자신의 과제물을 타인에게 보여주거나 빌려주는 것은… 부정행위에 해당한다.
  - 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
- 팀 프로젝트
  - 실험 및 실습 과정에서 (팀으로 수행하는 경우에는 다른 팀에 속한) 동료 학생이 작성한 사전보고서/코드의 일부 혹은 전부를 참조하여 실험 및 실습과정을 수행하는 것은 부정 행위에 해당한다.

Administrivia

- Any of the followings will result in **failure (F)**:
  - **Conducting any form of cheating or academic dishonesty**
  - **Not appearing more than 3/4 of all meetings**
    - **Three times of tardiness will be countered as one absence**
  - **Not taking any of the midterm and final exam**

Sounds cool?
Let's dive in!