



**KOÇ  
UNIVERSITY**

# **Database Management Systems**

## **Course Introduction**

**M. Emre Gürsoy**

Assistant Professor  
Department of Computer Engineering

[www.memregursoy.com](http://www.memregursoy.com)



# Today's Plan

---

- **Course logistics**
  - Going through the course syllabus
  - Description of course policies and grading
  - Q&A about course logistics
  
- **Motivation and sneak peek**
  - Why study databases and DBMSs?
  - How are they useful / why take this course?
  - What will you learn in this course?



# Recording Disclaimer

- **TL;DR: The lectures are recorded**



The synchronous sessions are recorded (audiovisual recordings). The students are not required to keep their cameras on during class.

The audiovisual recordings, presentations, readings and any other works offered as the course materials aim to support remote and online learning. They are only for the personal use of the students. Further use of course materials other than the personal and educational purposes as defined in this disclaimer, such as making copies, reproductions, replications, submission and sharing on different platforms including the digital ones or commercial usages are strictly prohibited and illegal.

The persons violating the above-mentioned prohibitions can be subject to the administrative, civil, and criminal sanctions under the Law on Higher Education Nr. 2547, the By-Law on Disciplinary Matters of Higher Education Students, the Law on Intellectual Property Nr. 5846, the Criminal Law Nr. 5237, the Law on Obligations Nr. 6098, and any other relevant legislation.

The academic expressions, views, and discussions in the course materials including the audio-visual recordings fall within the scope of the freedom of science and art.



# Course Description

---

- Undergraduate course on DBs and DBMSs
  - Both conceptual and practical aspects
- The course will cover:
  - Database design & data models
  - Query languages
  - Database normalization and schema refinement
  - Transaction management
  - Scheduling and concurrency
  - Indexing
  - Tentative topics: Query optimization, NoSQL



# Instructor



## M. Emre Gürsoy

Assistant Professor @ Koç University

[www.memregursoy.com](http://www.memregursoy.com)

PhD in Computer Science, 2020, Georgia Tech  
MS in Computer Science, 2015, UCLA  
BS in Computer Sci. and Engr., 2013, Sabanci Uni.  
High school, 2009, Robert College

**Research Areas:** Data privacy, security, data analytics and mining, adversarial machine learning, IoT, security&privacy in AI

- ❑ 12 journal pubs, 26 conference and workshop pubs, 3 US patents
- ❑ Frequently invited reviewer for intl. IEEE/ACM journals and conferences
- ❑ Several awards and grants (recently: EdgeSys 2020 **Best Paper Award**, TUBITAK **CAREER** Grant in 2021, etc.)



# Lectures

- **Tues + Thur @ 1:00-2:15 PM, SOS B08**
- **Current plan**
  - Online until April, then: ??
  - I am planning to lecture from SOS B08 regardless
  - I am planning to provide Zoom access regardless
    - Note: Interaction with Zoom participants may decrease if we switch to face-to-face or hybrid education
  - Lecture attendance will not be graded
    - Exception: During the semester, if < 15-20% of the class is attending lectures, we may introduce grade incentives/penalties for attendance
  - Lecture recordings will be shared via Panopto



# PS Sections

- You should be in **one** of the three PS sections:
  - PS-A (Friday 2:30-3:40 PM, CASE B24)
  - PS-B (Friday 4:00-5:10 PM, CASE B24)
  - PS-C (Friday 1:00-2:10 PM, SNA A22)
- Will be led by TAs, same material in each section
- Depending on week, **problem solving** or **practical (hands-on) experience** with DBMS software
  - For the prior: questions are typically from previous years' assignments and exams!
  - For the latter, you need your laptop
    - Installation prior to coming to PS can be necessary
    - We will make announcements in such cases



# Delivery of PS Sections

- How will PS sections be held
  - Online until April, then: ??
  - Until April, we plan to hold them online from the classroom (similar to lectures)
  - You can access via Zoom
  - PS attendance will **not** be graded
  - PS recordings will **not** be shared
    - **This is different from the lectures!!**
- **We don't have PS hours this week!**





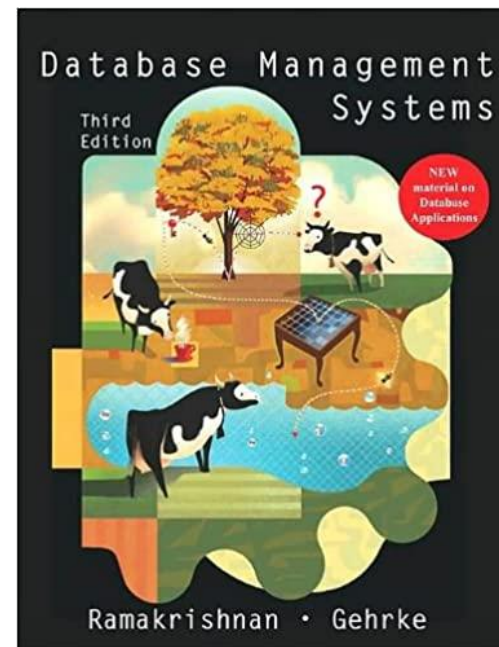
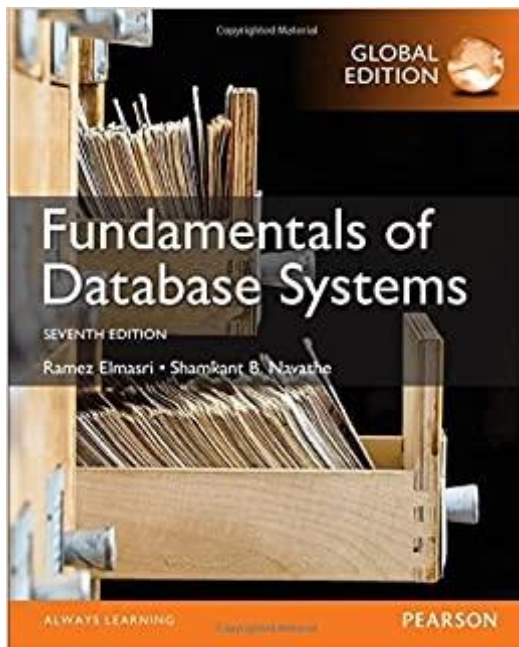
# Pre-requisites

- **COMP202: Data Structures and Algorithms** is the pre-requisite for this course
  - It will be strictly enforced
  - I will not give consent to take COMP306 if you didn't pass COMP202 already
- Graduate students:
  - No pre-requisite from KU course pool
  - But similar background is needed
    - Proficiency in math & programming are necessary!



# Course Textbook

- No mandatory textbook
- Following two books can be beneficial:
  - "Fundamentals of Database Systems" by Elmasri and Navathe
  - "Database Management Systems" by Ramakrishnan and Gehrke





# Textbook Policy

- Course lectures + slides are your **primary** resources
- Textbooks are your **secondary** resources
- We may not cover all chapters, we may not cover everything in a chapter
  - **You are responsible for material that is covered in the lectures**
  - **You should not rely on material from the books or elsewhere that is not covered in the class**
- There may be discrepancies between resources
  - Book to book, edition to edition
  - **The version that is taught in the lectures is FINAL**
  - **I will not accept any objections such as: "But Chapter N of the M'th edition of book X says ..."**



# Grading (Tentative)

Midterm Exams (2x)	48% (24% each)
Final Exam	27%
Homework Assignments	15%
Group Project	10%

## ■ Midterms and Final:

- Traditional written exams, hopefully **face-to-face (on campus)**
- Scheduled by the Registrar's Office
- Mixture of question types: True/False, multiple choice, short problems, query writing, ...
- I typically offer some **bonus points** for extra credit
- Final exam is **cumulative**



# Homework Assignments

- Tentatively 4-6 HW assignments, expected: 5
  - Total worth: 15%
- HWs for conceptual topics will be “pen and paper”
- HWs for practical topics will require implementation
  - Ex: **web programming** (PHP + HTML) or **data science** (Python) with **SQL** back-end

**HOMEWORK ASSIGNMENTS MUST BE COMPLETED INDIVIDUALLY. CHEATING OR CHEATING ATTEMPTS WILL NOT BE TOLERATED.**



# Late Homework Policy

- For late submission of HWs:
  - Up to 10 mins late: **-5% penalty**
  - 10 mins to 1 hour late: **-20% penalty**
  - > 1 hour late: **not accepted**
- Exceptions: emergencies (e.g., medical) with proper documentation (accepted by KU Health Center)
  - As much as possible, let me know **ahead of time**
- Do not ask for extensions close to a deadline (or after a deadline) – I want to be fair to all students



# Group Project

- 4-5 students per group (tentatively)
- In the second half of the semester
- Pick one application area of DBMSs and implement a DBMS-powered software project
  - Back-end **must** use DBMS, front-end is your choice
- Steps:
  1. Design your application and database
  2. Populate DB with real/realistic data
  3. Integrate complex queries & functionalities
  4. Demonstrate working prototype



# Project Ideas

- Movie tracking and rating site (mini-IMDB)
- Supermarket management app
  - Products, employees, customers, purchases, ...
- University management app (mini-KUSIS)
- Library management app
  - Books, authors, customers, borrowing/selling
- E-commerce or online retail site (mini-Amazon)
- Real estate and house rental site
  
- **Functionality is important**
- **Making good use of course concepts is important**





# Academic Honesty

- Collaboration is allowed only in the **Group Project**
- In all other graded items, **STUDENTS MUST WORK INDIVIDUALLY.**
- Violations will be immediately reported to the University Disciplinary Committee.
  - No warnings will be given to the involved student(s)
  - No tolerance to cheating

Koç University Student Code of Conduct:

<https://apdd.ku.edu.tr/en/academic-policies/student-code-of-conduct/>



# Exam Policy

---

- As of the beginning of the semester, I hope (and plan) to conduct all exams physically, in-person
- Unless there is a university-wide announcement that enforces all exams to be held online, **EVERYONE must take in-person exams with no exceptions**
  - By taking this course, you are assumed to have read and understood this rule



# Make-up Exams

- You can get a make-up exam only if you miss a midterm or final with a **valid** excuse
  - Must be accepted and approved by the University (e.g., Health Center), and sent to me officially
- A **single, joint make-up exam** is given at the end of the semester
  - Covers all topics
  - You take this make-up regardless of which exam you missed (counts in place of your missing exam)
  - **Friendly advice:** take the regular exams 😊



---

**Done with the logistics part...**

**Questions?**



# Basic Definitions

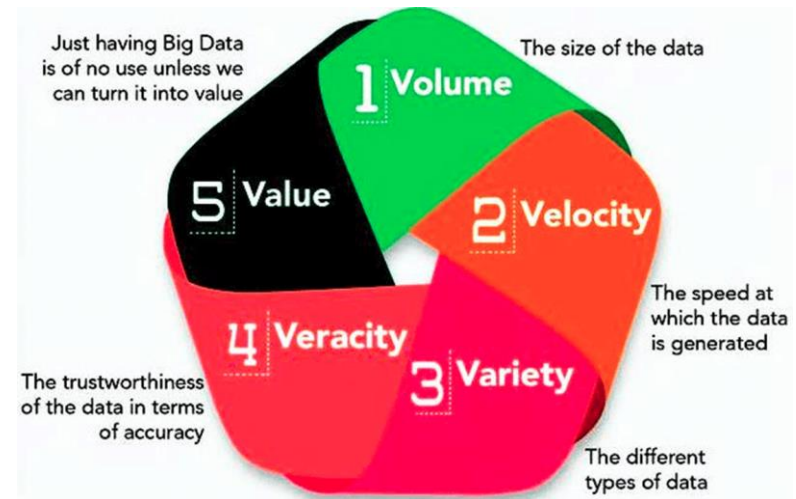
---

- **Data:** Known facts that can be recorded and have an implicit meaning.
- **Database (DB):** A collection of related data.
- **Database Management System (DBMS):** A software package or system to facilitate the creation and maintenance of a computerized database.



# Why Do We Need DBs?

- There are large volumes of data everywhere
- We need to store and process this data effectively, before we can use it or turn it into value





# Who Uses DBs?

- **DBs are pervasive in many industries**
  - **Banking**
    - Client information, accounts, cards, money transactions, ...
  - **Airlines**
    - Flights, bookings, scheduling, pilots, customers, ...
  - **Telecommunication**
    - Customer info, call archives, bills, balances, ...
  - **Sales and/or e-commerce**
    - Products, buyers, sellers, pricing, sales records, shipping records, ...
  - **Education**
  - ... and many more!



# A Simple Database

- University DB:
  - University has many courses
  - A course can be taught by different instructors in different years and semesters, and can have different sections
  - Courses have prerequisites, but different sections of the same course should have same prerequisites
  - Students enroll into courses and receive letter grades
  - Students can take a course multiple times (but not in the same semester)
  - ...

**COURSE**

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

**SECTION**

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

**GRADE\_REPORT**

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

**PREREQUISITE**

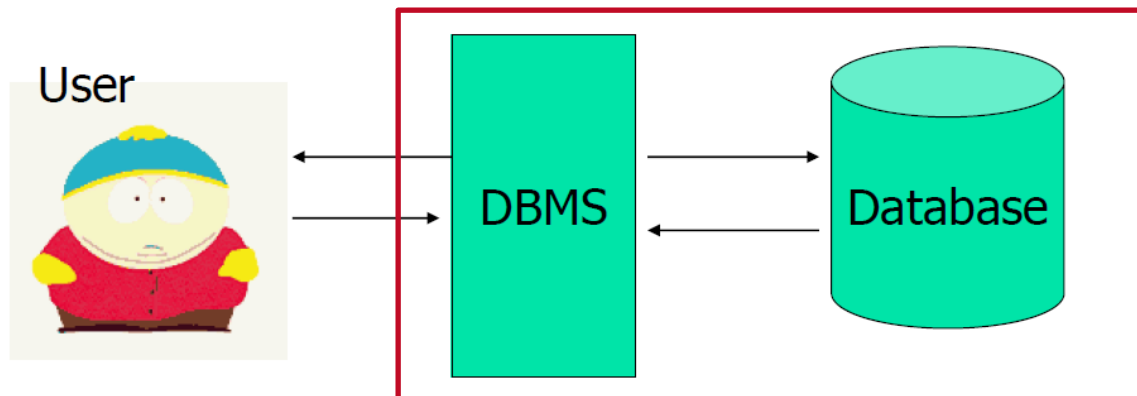
Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310





# Need for DB+DBMS Experts

- Most large-scale software projects have a database component
- Someone needs to:
  - Design the DB (what are the tables? what are the data types?)
  - Implement the DB, DBMS, and set up the related environment
  - Maintain and refine the DB (changing requirements?)
  - Write complex queries and ensure their efficiency
  - Ensure procedures exist for backups, data recovery, data integrity, parallelism, concurrency, ...





# Advantages of DBMSs

- "Can't we just store data in large txt/csv files?"
- DBMSs offer several advantages:
  - Data integrity and consistency
  - Remove duplications
  - Persistent, durable storage; backup and recovery
  - Parallelism, support for concurrent usage
  - Indexing, efficient query processing (much faster than scanning through large txt/csv files!)
  - ... and many more that we won't cover in this course (e.g.: access control mechanisms for security)
- But of course there could be cases where DBMSs may be **infeasible** (e.g.: embedded systems) or **overkill** (5 kb csv file, quick 30 LoC script)



# List of Topics

- Introduction (**today**)
- Entity-Relationship model (E-R model)
- Relational model
- E-R model to relational model conversion
- Relational algebra
- Structured Query Language (SQL)
- Functional dependencies, normal forms and normalization
- Transaction management, ACID properties, scheduling, serializability
- Concurrency control
- Indexing (hash-based and tree-based)
- **[Tentative]** Query processing and optimization
- NoSQL databases