# Introduction to Databases

*Spring 2014*

**Instructor:** *Prof.* Bongki Moon
Contact: bkmoon@snu.ac.kr (✉), 02-880-1842 (☎), 02-886-7589 (FAX), Engr 301-402 (☕)
Office hours: Tue/Thu 2:00-3:00, or by appointment.

**Teaching Assistant:** Seoung Gook Sohn (sgsohn@snu.ac.kr, 02-880-1855)

**Class schedule:** Tue/Thu 15:30-16:45, Engr. Bldg. 301, Room 203

**Overview and Topics:** This is an introductory database course at the undergraduate level that covers the basic topics of database technology. The topics include data models, formal languages and SQL, database design, application design, storage and file structures, indexing methods, and transactions.

**Prerequisites:**

- 4190.204 *Data Structures*, or an equivalent undergraduate-level data structure course.
- Sufficient programming skill enough to be able to design, code, and debug programs consisting of more than five hundred lines of code.

**Textbook and Lecture slides:** The lectures will be based on the following textbook.

- Avi Silberschatz, Henry F. Korth and S. Sudarshan, *Database System Concepts*, Sixth Edition, McGraw-Hill, ISBN 0-07-352332-1, 2010.

The lecture slides are made available publicly by the authors, and downloadable from `http://db-book.com/db6/slide-dir`. You can also download solutions to practice exercises from `http://db-book.com/db6/practice-exer-dir/`. For more information, visit the publisher's web site `http://www.mhhe.com/silberschatz`.

**Grading:** This course is offered for three credits. The grade is broken down as follows.

| | | |
|---|---|---|
| • Attendance | 5% | |
| • Term project | 35% | |
| • Exams | 60% | (Two midterms 40% + Final 20%) |

The final grades may be curved, but a weighted total of 90% and above is guaranteed to be an A, 80% and above at least a B, 70% and above at least a C, and 60% and above at least a D. No determinations will be made until the final exam is graded.

**Web-based Learning Management System:**
The SNU eTL system is an open source Learning Management System (LMS). You can access eTL at: `http://etl.snu.ac.kr` and login with your *mySNU* id and password.

The eTL server will have the course syllabus, lecture notes, and projects. It will also be used for submission of projects and assignments (if any), and posting grades.

**Term project:** Students are required to complete a term project in a small group of *three to four* people throughout the semester. The goal of the project is to understand the design of a database-driven web based information system. Students are to identify an application domain to work on, design an underlying database, define application functions, and implement the database and application.

The term project will be completed in a few stages. For each stage of the project, students are required to submit a short report and an archive of source code files if necessary. Detailed information will be provided shortly.

**Late assignment policy:** Each part of the project must be turned in electronically at the SNU eTL site (or some place similar). *No email submission* will be accepted. A late project may be turned in within 24 hours after the deadline for a 10% penalty (*i.e.*, 10% deduction of your credits). *No projects late for more than 24 hours* will be accepted unless a valid excuse (*e.g.*, documented illness or family emergency) is given to the instructor by the day prior to the due date. For fairness, this policy will be strictly enforced.

**Exam policy:** Exams will consist primarily of short-answer and problem-solving questions, but code-writing questions are also possible. Calculators are the only electronic devices that are permitted on exams. No other electronic devices such as mobile phones are allowed.

All students are expected to take the exams at the announced exam times. Make-up exams will be given only in *extreme* circumstances. The instructor will decide whether a circumstance is extreme or not. For example, being in a documented car accident on the way to the exam is likely to count as an extreme circumstance. A zero will be given for an exam missed under a sub-extreme circumstance.

**Attendance policy:** The University requires attendance to be recorded. However, we realize that absence from class is sometimes unavoidable due to medical or personal reasons. It is the responsibility of the student to report the reason for his/her absence to the instructor and provide documentation if requested. The instructor will decide whether the student's absence should be excused. Each student is accountable for all work missed because of absence.

**Academic integrity and conduct:** Academic integrity is a commitment to five fundamental values: honesty, trust, fairness, respect, and responsibility. Students are encouraged to discuss the course work with classmates; both giving and receiving advice will help you learn. However, students need understand that academic integrity must not be violated under any circumstance.

Briefly, you will not accept solutions from others, you will not give solutions to others, and you will not tamper with graded papers, code, or exams. It is OK to talk with other students about **what** the problems are but it is not OK to talk about **how** the problems are to be solved. Excessive collaboration is considered a violation.

Each student must develop and write her/his own code and homework solutions. Each student must turn in her or his own work. The one exception is test cases for programming assignments. You are allowed to share test cases and their expected results.

Students who violate academic integrity will be subject to penalties. There is no violation that is not serious. Every single violation will result in *Reduction in course grade* as well as *Total loss of credit for work involved.* A second violation committed by the same student will be reported to the Student Discipline Committee with *Failing grade in course* (*i.e.*, failure of the entire course) as a minimum penalty. Refer to Article 2 of the Code of Academic Integrity for further details on the penalties and the procedures involved.

# Topics and Tentative Schedule

| Date | Topic Description | Milestones |
|---|---|---|
| Week 01 | Course introduction,<br>Introduction to Relational Model | |
| Week 02 | Formal Relational Query Languages | |
| Week 03 | Introduction to SQL | |
| Week 04 | Intermediate SQL | |
| Week 05 | Advanced SQL | [Midterm #1] |
| Week 06 | Database Design and ER Model | |
| Week 07 | Relational Database Design | |
| Week 08 | Application Design and Development,<br>Performance Benchmarks | |
| Week 09 | Storage and File Structure | |
| Week 10 | Indexing: Basic Concepts | [Midterm #2] |
| Week 11 | Hashing: Static and Dynamic,<br>Bitmap indexes, Spatial/Temporal/Mobile | |
| Week 12 | Transactions: ACID, Isolation Levels, SQL statements | |
| Week 13 | Recovery and Atomicity,<br>Database System Architectures | |
| Week 14 | Case Studies: Oracle and PostgreSQL | |
| Week 15 | Review and foresight | [Final Exam] |