CS2102 Lecture 0
Course Admin

CS2102: Database Systems

- **Lecturers**:

  Chan Chee Yong
  Email: chancy@comp.nus.edu.sg
  Office: #03-24, COM1

  Xiao Xiaokui
  Email: xiaoxk@comp.nus.edu.sg
  Office: #03-25, COM1

- **Synchronous Online Lectures**: Friday, 12-2pm

CS2102 Teaching Assistants

- **Aaron Seah Ter Chuan** (e0266255@u.nus.edu)

- **Jeffry Hartano** (dcsjeff@nus.edu.sg)

- **Jiang Yixing** (jiang@u.nus.edu)

- **Joel Lim** (lim.joel@u.nus.edu)

- **Luo Xinjian** (xinjian.luo@u.nus.edu)

- **Ruhani Suri** (suriruhani@u.nus.edu)

- **Sean Low Jun Kai** (sean.low.jk@u.nus.edu)

- **Song Kai** (song.kai@u.nus.edu)

- **Wei Jianxin** (jianxinwei@u.nus.edu)

Prerequisites for CS2102

- **Advanced Programming** (CS1020/CS1020E/CS2020/CS2030/CS2040/CS2040C)

- **Discrete Mathematics** (CS1231/MA1100)

Learning Objectives

The aim of this module is to introduce the fundamental concepts and techniques necessary for the understanding and practice of design and implementation of database applications and of the management of data with relational database management systems. The module covers practical and theoretical aspects of design with entity-relationship model, theory of functional dependencies and normalisation by decomposition in third and Boyce-Codd normal forms. The module covers practical and theoretical aspects of programming with SQL data definition and manipulation sublanguages, and relational algebra/calculus.

Learning Objectives (cont.)

- How to design an entity-relationship data model to capture the data requirements for an application and translate the conceptual data model to a relational database schema

- How to refine a relational database schema using schema decompositions to avoid update anomalies

- How to use SQL to define relational schemas and write SQL queries on the schemas

- How to reason about the correctness of relational queries based on the concepts of formal query languages (relational algebra/calculus)

- How to apply knowledge of relational database systems to develop database appplications

CS2102 Reference Books

Information also available at LumiNUS → Module Details → Readings

- Raghu Ramakrishnan & Johannes Gehrke,
  Database Management Systems,
  McGraw-Hill, Third Edition, 2003.
  (QA76.9 Dbm.Ra 2003)

- Avi Silberschatz, Hank Korth & S. Sudarshan,
  Database Systems Concepts,
  McGraw-Hill, Seventh Edition, 2019.
  (QA76.9.D3 Sil 2020)

- Hector Garcia-Molina, Jeffrey Ullman, & Jennifer Widom,
  Database Systems: The Complete Book,
  Prentice Hall, Second Edition, 2009.
  (QA76.9 Dbm.Gar 2009)

Workload & Assessment

- Number of credits = 4
- Workload per week = 10 hours
  - 2 lecture hours
  - 1 tutorial hour
  - 4 assignment & project hours
  - 3 preparatory work hours
- Tentative Module Assessment:

Assessment Component	%
Tutorial Participation	2
Assignments	10
Team Project	28
Midterm Test	20
Final Test	40

Lectures

- Lectures will be conducted in online synchronous mode

- Lecture recordings will be made available at LumiNUS $\rightarrow$ Multimedia

- The following two lectures will be replaced with make-up lectures
    - Week 5: February 12 (Chinese New Year)
    - Week 11: April 2 (Good Friday)

Tentative Lecture Schedule

Week	Topic
1	Introduction
2	Relational Algebra
3	SQL (Part 1)
4	Entity Relationship Data Model
5	SQL (Part 2)
6	SQL (Part 3)
-	Recess Week
7	Application development with SQL (Part 1)
8	Application development with SQL (Part 2)
9	Schema Refinement (Part 1)
10	Midterm test
11	Schema Refinement (Part 2)
12	Schema Refinement (Part 3)
13	Course Review
14	Reading Week
15	Final Test

Tutorials

- Face-to-face tutorials (may be converted to online if things change)

  - Duration: 45 minutes
  - Start from Week 3

- Not all tutorial questions will be discussed in class

- Tutorial participation (2 marks)

  - Each student will be pre-assigned to prepare the answer for a question
  - The answer files must be uloaded to the appropriate LumiNUS file folder (for the tutorial group) by **Monday, 9am**
    - Each file must be named using the format **T<xx>_Q<yy>_<name>.pdf**, where xx = tutorial number, yy = question number, & name = name of student.
  - For each discussed question, one of the assigned students will be randomly picked to lead the discussion
  - Other students are expected to participate as well: ask questions, suggest alternative answers, etc.

Assignments

- Individual assignments (10 marks)
- Tentative assignment deadlines

#	Assignment Topic	Due Date
1	SQL	Week 8
2	Schema Refinement	Week 12

Project

- **Objective**: Develop a database application

    - Software: PostgreSQL database server
    - Languages: SQL & PL/pgSQL

- Project teams of 4 members each

    - Team members could belong to different tutorial groups

- Team registration will be be opened by end of Week 2

- Teams to be formed by the end of Week 3

- If you're unable to form a team or you belong to a team with fewer than 4 members, you may be assigned/reassigned to a random team

Project Deadlines

Due Date	Task
Week 3	Project Team Registration
Week 6	Submission of ER Data Model Design
Week 12	Submission of Project Report & Code
Week 13	Project Evaluation

Midterm & Final Tests

- **Midterm test**: March 26 (Friday), 12-2pm

- **Final test**: April 30 (Friday), 9-11am

- Both tests will be conducted online using **Examplify**
  - Closed-book tests
  - Allowed to have one double-sided A4-sized sheet of notes

- Might change to face-to-face tests if things change

Examplify Software

- `https://wiki.nus.edu.sg/display/DA/Student`

- **Examplify** will be used for both midterm & final tests

  - System requirements:

    `https://wiki.nus.edu.sg/display/DA/Device+Minimum+System+Requirements`

  - CIT's briefing slides:

    `https://wiki.nus.edu.sg/display/DA/CIT+Student+Briefing+Slides`

  - Installation: `https://wiki.nus.edu.sg/display/DA/Download+and+Install+Examplify`
  - Service help desk: `http://www.cit.edu.sg/da-help`
  - Emergency email: citbox25@nus.edu.sg

Linux Server Account

- You will be accessing PostgreSQL & other software from SoC's Linux server **stu.comp.nus.edu.sg**

- Access to stu server requires a SoC account

- If you don't have a SoC account,

    - Visit `https://mysoc.nus.edu.sg/~newacct` to create your SoC account
    - Visit `https://mysoc.nus.edu.sg/~myacct/services.cgi` to activate your access to the linux server

- Once your SoC account is activated, you can ssh to the stu server

- Instructions on how to access the PostgreSQL server will be announced next week

Course Policies

- Students are responsible for the following:
  - Attending lectures & tutorials
  - Checking LumiNUS/emails for course-related announcements/updates

- For clarifications on lecture material, the best way is to post your questions on LumiNUS Forum. Questions emailed to the lecturer may be posted to LumiNUS and answered there. You can also email the lecturer to arrange for consultation sessions.

Course Policies (cont.)

- **Zero-tolerance for plagiarism**
- Students will be reported to University for disciplinary action for plagiarism/cheating offence
  - Offenders will receive F grade for the module
- Resources:
  - `http://www.nus.edu.sg/celc/programmes/plagiarism.html`
  - `https://www.comp.nus.edu.sg/cug/plagiarism/`

Course Policies (cont.)

`https://www.comp.nus.edu.sg/cug/plagiarism/`

All students share the responsibility for upholding the academic standards and reputation of the University. Academic honesty is a prerequisite condition in the pursuit and acquisition of knowledge. Academic dishonesty is any misrepresentation with the intent to deceive or failure to acknowledge the source or falsification of information or inaccuracy of statements or cheating at examinations/tests or inappropriate use of resources. There are many forms of academic dishonesty and plagiarism is one of them. **Plagiarism is generally defined as the practice of taking someone else's work or ideas and passing them off as one's own** *(The New Oxford Dictionary of English). The University does not condone plagiarism.*

Course Policies (cont.)

`http://www.nus.edu.sg/registrar/administrative-policies/acceptance-record.html`

Students should adopt this rule - **You have the obligation to make clear to the assessor which is your own work, and which is the work of others.** *Otherwise, your assessor is entitled to assume that everything being presented for assessment is being presented as entirely your own work. This is a minimum standard.*

A student may not knowingly intend to plagiarise, but that should not be used as an excuse for plagiarism. **Students should seek clarification from their instructors or supervisors if they are unsure whether or not they are plagiarising the work of another person.**