

EDAF75 Database Technology

Lecture 1

Christian.Soderberg@cs.lth.se

January 20, 2025



Administration

- ▶ EDAF75: Database Technology
- ▶ Christian Söderberg (christian.soderberg@cs.lth.se)
- ▶ Omfattning: 7.5 hp \approx 6.5 hp + 1.0 hp
- ▶ Obligatoriskt: laborationer + projekt + tenta
- ▶ Registrera dig i Ladok senast 23:59 ikväll
- ▶ Kursombud: ?, ?



Administration

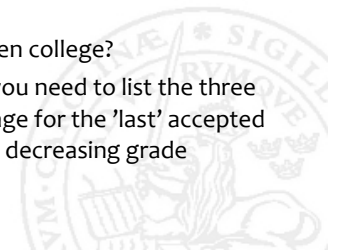
- ▶ Moodle: [edaf75-2025](#)
- ▶ Kurshemsida: Länk från Moodle – *läs reglerna!*
- ▶ Kurslitteratur: Material och länkar på kurshemsidan
- ▶ Schema och syllabus



Att fundera på

We've been asked to write code to handle students and their college applications. Each student has a name, a student id (`stil`), and an average grade. The students apply for various majors at colleges, where each college has a name, is situated in a given state, and has a given enrollment. Each application is for a specific major at a specific college, and it can be accepted, rejected, or pending.

- ▶ What classes and/or data structures would you use to keep track of this data in Python or Java?
- ▶ How do you list all applications for a given student?
- ▶ How do you list all applications for a given major at a given college?
- ▶ Approximately how many lines of Python/Java code do you need to list the three programs (`major@college`) with the highest grade average for the 'last' accepted student (assuming the students are accepted in order of decreasing grade averages)?
- ▶ ...



Att fundera på

- ▶ ...
- ▶ How would you persist your data, so it doesn't get lost if we have to restart the program?
- ▶ Can you come up with a way to let several programs /access/ the data simultaneously?
- ▶ If you solve the previous problem, and want to allow several programs to /update/ the data simultaneously, how do you make sure the data isn't corrupted?

After this first week, the solutions to problems 1-6 will be almost trivial (using SQL) – in week 3 we'll see how to use our SQL solutions from Python/Java code, and after week 5, problem 7 will also be trivial.

Kursinnehåll

- ▶ Vecka 1: SQL – frågespråk för relationsdatabaser
- ▶ Vecka 2: Modellering och design av relationsdatabas
- ▶ Vecka 3: Att koppla databasen till omvärlden
- ▶ Vecka 4: Normalisering – ett sätt att designa databaser
- ▶ Vecka 5: Transaktioner och triggers. Sätt att få databasen snabbare. Alternativ till relationsdatabaser (NoSQL).
- ▶ Vecka 6: Grafdatabaser
- ▶ Vecka 7: Projekt och repetition
- ▶ Vecka 8: Extentagenomgångar

Tenta

- ▶ Uppgift 1: Modellering och design av relationsdatabas (vecka 2)
- ▶ Uppgift 2: SQL – frågespråk för relationsdatabaser (vecka 1)
- ▶ Uppgift 3: Normalisering – ett sätt att designa databaser (vecka 4)
- ▶ Uppgift 4-6: Att koppla databasen till omvärlden (vecka 3)
- ▶ Uppgift 4-6: Transaktioner och triggers. Sätt att få databasen snabbare. Alternativ till relationsdatabaser, NoSQL (vecka 5)
- ▶ Uppgift 4-6: Grafdatabaser (vecka 6)
- ▶ Man måste bli godkänd på var och en av uppgift 1-3!