TDA357/DIT620 – Databases

Lecture 1 – Course Introduction, Tables

Jonas Duregård

Who am I?

- Jonas Duregård
- Senior lecturer at the Functional Programming division
- First time I'm giving this course!
    - Although I was involved in the course back when I was a PhD student
- Started at Computer Science (Datavetenskapligt program) back in 2004

Who are you?

- A *lot* of people, as it turns out. A mix of:
  - Computer Engineering (D, högskoleingenjör)
  - Software Engineering (IT)
  - Industrial Engineering and Management (I)
  - Biomedical Engineering (MSc)
  - Interaction Design (MSc)
  - Computers Science (DV) and other GU-students
  - Possibly other people that no one even told me about
- You have taken a course in Data Structures and you have learned (and not forgotten everything about) Set Theory and such.
- You have some knowledge of programming in Java
- Let me know if the description above does not apply to you

Why databases?

Let's get motivational

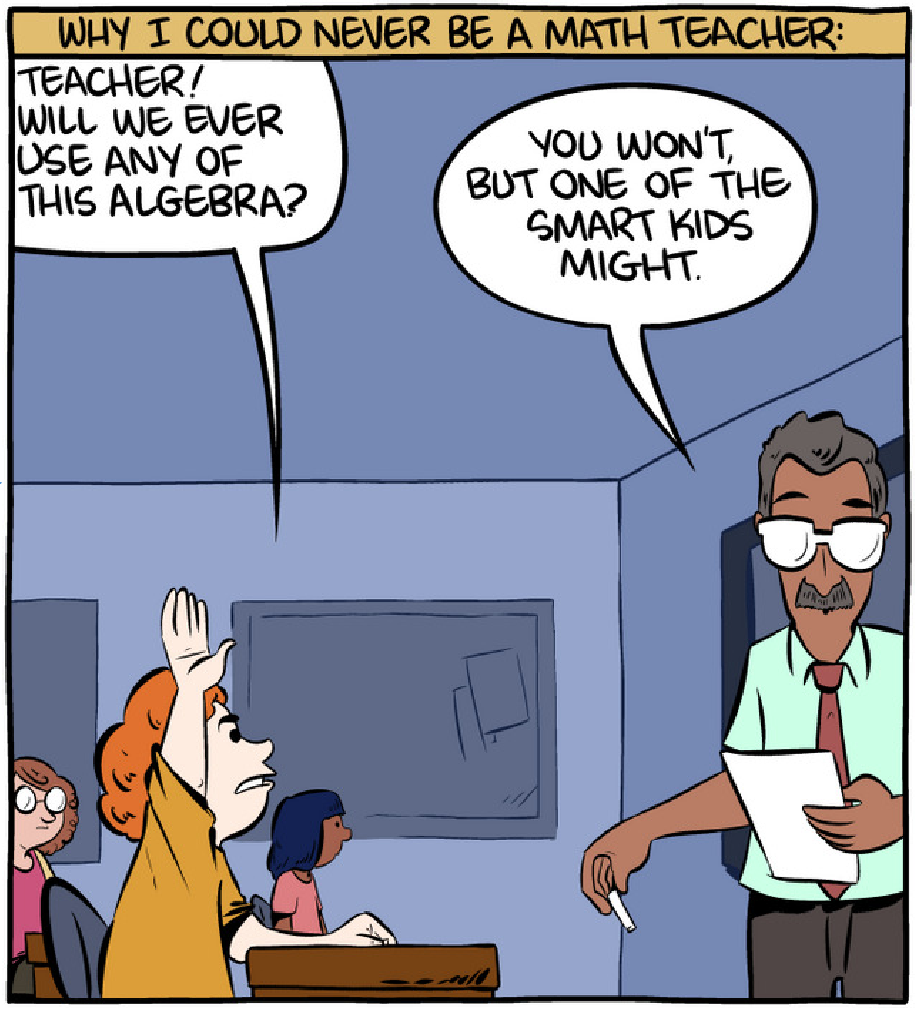
Optimism is the faith that leads to achievement. Nothing can be done without hope and confidence.
Helen Keller
BrainyQuote

Failure will never overtake me if my determination to succeed is strong enough.
OG MANDINO

MAKE UP YOUR MIND THAT NO MATTER WHAT COMES YOUR WAY, NO MATTER HOW DIFFICULT, NO MATTER HOW UNFAIR, YOU WILL DO MORE THAN SIMPLY SURVIVE. YOU WILL THRIVE IN SPITE OF IT.
JOEL OSTEEN
SUCCESS.com

Disclaimer

If my motivational speech is fails to capture your interest, it's because i'm like this guy, $\longrightarrow$
not because databases is a useless subject


WHY I COULD NEVER BE A MATH TEACHER:
TEACHER! WILL WE EVER USE ANY OF THIS ALGEBRA?
YOU WON'T, BUT ONE OF THE SMART KIDS MIGHT.

picture credit: smbc-comics.com

Why a whole course on databases?

- Short answer: Databases are everywhere
- Long answer: Databases are used in
  - WWW (Every page you visit is involves several database operations)
  - Finance (has driven the development of databases since the 60's)
  - Industry: Production control, Test data, Inventories, Sales, …
  - Research: Sensor data, Biological data, Demographical data…
  - It would be a challenge to go a single day of your life without causing a lot of databases updates

Wow, that was motivating – now what is a DB?

- Wide sense: Any collection of data that can be accessed digitally and is:
  - Structured – Data is stored in efficient structures
  - Persistent – Data is not lost without deliberate action
  - Mutable – data can be added/deleted/modified
- Slightly more useful: A database is a collection of data managed by a specialized software called a Database Management System (DBMS)

Some popular DBMS's:
Oracle, MS SQL Server, PostgreSQL, MySQL, ...

Why not use a file system?

- File systems are structured, persistant and mutable
- … but very inefficient and 'bulky' to work with

Modern DBMS

- Handle persistent data
- Give efficient access to huge amounts of data
- Give a convenient interface to users
- Guarantees *integrity constraints* on data
- Handles transactions and concurrent access to data

Relational databases

- Basically a bunch of tables with columns and rows
- Can also be viewed as mathematical relations, if you're into that
- Requires significant design-work
- SQL is a standardized language for manipulating relational databases
  - Common language supported by lots of different DBMS
  - Create, manipulate and query databases
  - Arguably one of the most used computer languages in existence
  - Fancy people pronounce it "Sequel"

Other database models

- The relational data model is so dominating, other approaches are commonly referred to as NoSQL-databases
- Semi-structured hierarchical models (XML, JSON, …)
- Key-value stores (Oracle NoSQL, Riak, …)
    - Easily distributed across multiple computers/data centers
    - Very simplistic data model (maps and lists)
- Usually NoSQL-databases are easier to design, sometimes more efficient, but also more limited when it comes to integrity constraints

Database system studies

- Design of databases
  - Entity relationship modelling
  - Relational data model
  - Dependencies and normalisation
  - Semi-structured data model
- Database Programming
  - Relational Algebra
  - Data manipulation and querying in SQL
  - Application programs in general purpose languages (like Java)
- Database implementation
  - Indexes, transaction management, concurrency, data recovery, …

Course Objectives

You will learn how to

- Design a database
- Construct a database from a schema
- Use a database through queries and updates
- Use a database from an external application

Course Objectives – Design

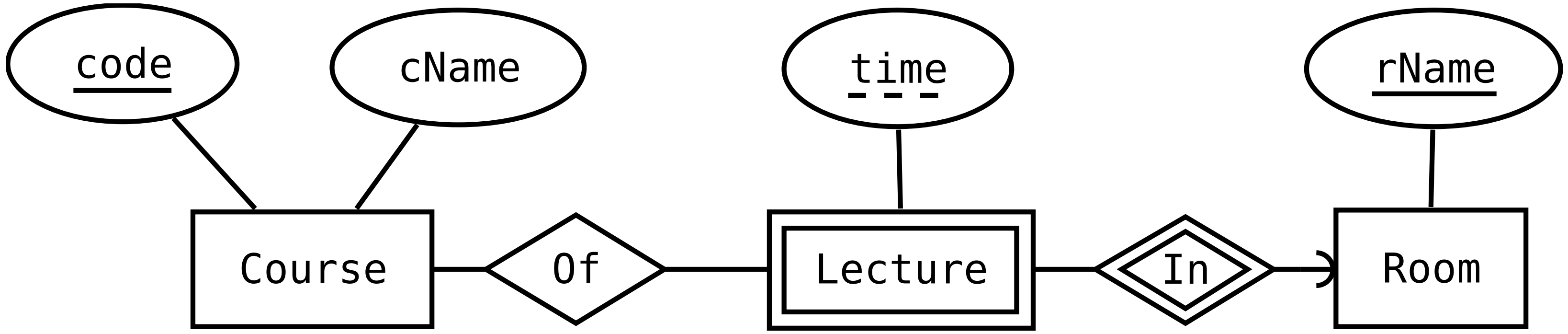
- When the course is through, you should
    - Given a domain, know how to design a database that correctly models the domain and its constraints

”We want a database that we can use for scheduling courses and lectures. This is how it’s supposed to work: …”

Course Objectives – Design

- Construct Entity-Relationship digagrams (ER)
- Determine functional dependencies (FD's)
- Compute normal forms

```
code → cName
code, time → rName
rName, time → code
```



Course Objectives – Construction

- When the course is over, you should
- Given a database schema with related constraints, implement the database in a relational DBMS

```
Courses(code, cname)
Rooms(rname)
Lectures(room, time, course)
    room   -> Rooms.rname
    course -> Courses.code
```

```sql
CREATE TABLE Courses(
  code CHAR(6) PRIMARY KEY,
  cname TEXT NOT NULL
);
CREATE TABLE Rooms (
  rname VARCHAR(100) PRIMARY KEY
);
CREATE TABLE Lectures (
  room VARCHAR(100) NOT NULL
    REFERENCES Rooms,
  course CHAR(6) NOT NULL
    REFERENCES Courses,
  time TIMESTAMP,
  PRIMARY KEY (room, time)
);
```

Course Objectives – Usage

- When the course is through, you should
  - Know how to query a database for relevant data using SQL
  - Know how to modify the contents of a database using SQL

```
SELECT time, room
  FROM Lectures
  WHERE course = 'TDA357';
```

```
INSERT INTO Rooms VALUES ('GD');
```

Course Objectives - Applications

- When the course is through, you should
  - Know how to connect to and use a database from external applications

Switchin' it up

- Traditionally this course is taught in the order you perform the tasks:
  design > construction > usage > application
- Problem: High level design concepts are difficult to learn and appreciate before you know about usage (Like learning object orientation and UML before knowing anything about programming)
- Instead, we will start with the gritty details (SQL), and then learn the abstract concepts on top of that
  construction > usage > design > (more usage/construction) > application

Course organization

- Scheduled lectures, lab sessions etc. up until December 21
- Exam is on January 16

Course language

- … is english in case you hadn't noticed
- Feel free to ask me questions in Swedish and I will answer in English

Course organization

- 1-3 lectures per week
- About 5 lab sessions per week (starting next week)
  - Opportunity to ask questions about the lab
  - Usually students attend one or two per week
  - No single session is mandatory, but you will have to attend at least one (near the end of the course to demonstrate your solution)
- 3 exercise sessions in weeks 1-6 of the course
  - Practice exam questions on the material from this weeks lectures
  - You attend at most one session per week
- As always, the schedule is available in Timeedit

Examination

The course has two parts:

- A lab assignment (graded pass/fail)
- A written exam (graded fail/3/4/5)

- Both must be passed to pass the course
- The grade of the exam will be your overall course grade

Lab organization

- Carried out in groups of two
  - NO! – you are not an exception to this ☺ (Unless you are…)
- The assignment is divided into five tasks:
  - Tasks 0-3 are handed in and graded
  - Task 4 is assessed on any lab session (don't wait till the last one)
  - You can and should proceed with the next task before the first is accepted
- All questions about groups etc. should be directed to my Head of Assignments Markus Aronsson (mararon@chalmers.se).
- Questions about the assignment text can be posted to the Google group (but do not post any part of your solution!)

Task 0

- You need to register in Fire and submit an empty submission <u>this week</u> to request a database username/password

Lab assignment overview

Goal: Construct a "student portal" application in Java+SQL

- Part I: Starting with a domain description and a draft schema, implement the schema in a database and write queries (views) for common operations
- Part 2: Use systematic design methods to add a few features, and find and eliminate flaws in the original schema
- Part 3: Create triggers to further increase the usefulness of the database
- Part 4: Connect to your database from a simple Java Application

Feedback

- You should get a response to your submission within 3 working days from submission
  - This is an ambition goal, so do not be to upset if we fail around deadlines
- If you are 'shotgunning' lots of poor submission, do not expect lots of useful feedback. We also reserve to right not to grade submissions at all if resubmissions do not show clear improvement.
- If you know your solution does not work when you submit it, always say so in your submission
- If you do not follow the submission instructions exactly, expect your submission to be summarily rejected without feedback (other than a polite request to follow the instructions)

About cheating

The assignment is to be carried in groups of two.

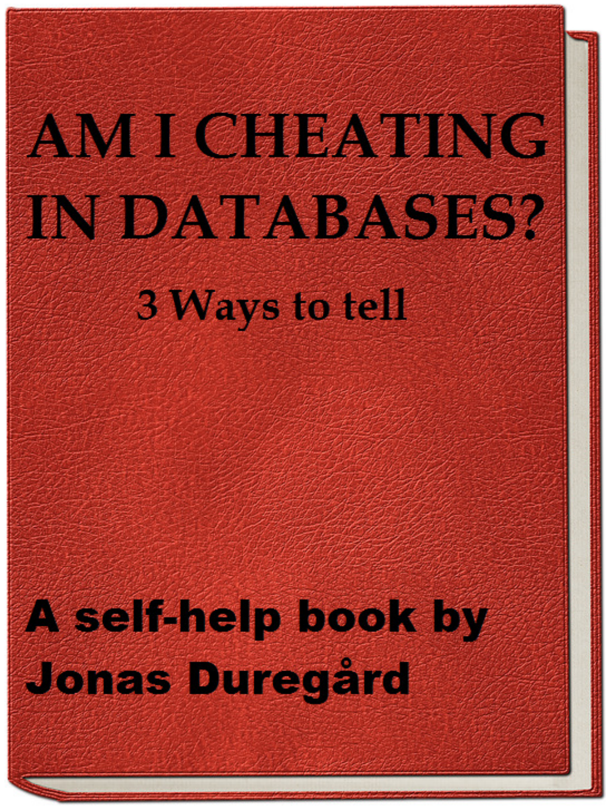
In the lab session, you will be working close to other groups on the same tasks. Helping each other is allowed within certain limits.

It's easy to accidentally cross the line into cheating

- Generally, talking about the assignment is not cheating
- If you show your code to other students, you are cheating
- If you look at code from other students, you are cheating

If we catch you cheating, we are obligated to report it to a disciplinary committe

Also: Do not put your solution in a public repository


AM I CHEATING IN DATABASES?
3 Ways to tell
A self-help book by Jonas Duregård

Literature

- The official course literature is Database Systems: The Complete Book by Garcia-Molina, Ullman and Widom.
    - This is a 1200 page volume. Reading it from cover to cover may not be the most productive use of your time…
- There is a more condensed manuscript by Jyrki Nummenmaa and Aarne Ranta currently called "Databases in 127 Pages" (link on the course page)
- Also, the slides from these lectures are published online
- For JSON (which is a new part of the course) there will be some additional links (although most of the XML-related chapters in the books above apply to JSON as well)

Web resources

- Course wep page has everything (including these slides):

http://www.cse.chalmers.se/edu/course/TDA357/HT2018/

(Or just google TDA357 if you're a lazy post-millenial)

- A Google group (tda357-ht2018), ask questions, read answers

Tools

- PostgreSQL (a.k.a. Postgres) is the DBMS we will be using in the course
    - Free stuff! Postgres is open source software
    - Works on most operating systems
    - Available on Chalmers computers
    - postgresql.org
- Dia is the recommended editor for ER-diagrams
    - More free stuff! Dia is also open source!
    - Available on Chalmers computers (i hope)
    - dia-installer.de
- You may want to have a text editor that supports SQL syntax highlighting

Teaching staff

- Examiner: Jonas Duregård (me!)
- A small army of Teaching Assistants:
    - Selpi
    - Markus Aronsson
    - Herbert Lange
    - Matthías "The Witch King" Páll
    - Natalia Jurczyńska
    - Tsigabu Mebrahtu Birhanu
    - Razmus Strandell
    - Emilia Vestlund
    - Gustav Grännsjö

Any questions on course organization?