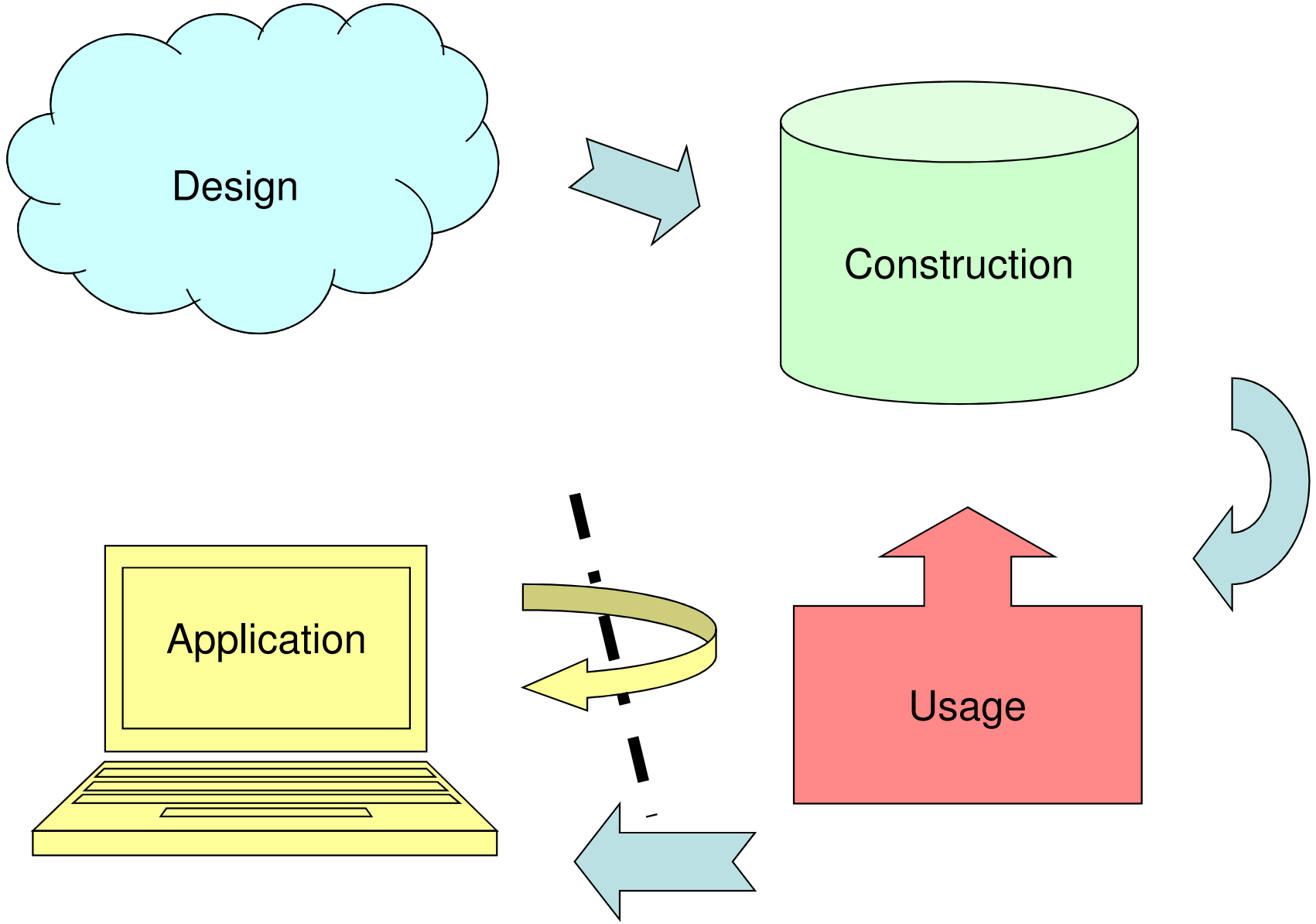
Databases Course Overview

Creating a Database System



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: …”

Workflow – Design

- Model the domain as an E-R diagram.
- Translate the diagram to a set of relation schemas, forming a stub schema for the database.
- Use FDs (and INDs) to identify relevant constraints, and adapt/complete your schema accordingly.

Course Objectives – Construction

When the course is through, you should

- Given a database schema with related constraints, implement the database in a relational (SQL) DBMS.
- SQL Data Definition Language

Workflow – Construction

- Implement the database schema in a DBMS.

- Implement auxiliary elements – views and triggers – to provide simple and safe interfaces, defined using privileges.

Course Objectives – Usage

When the course is through, you should

- Know how to query a database for relevant data using SQL.
- Know how to change the contents of a database using SQL.

- Relational Algebra
- SQL Data Manipulation Language

Course Objectives – Applications

When the course is through, you should

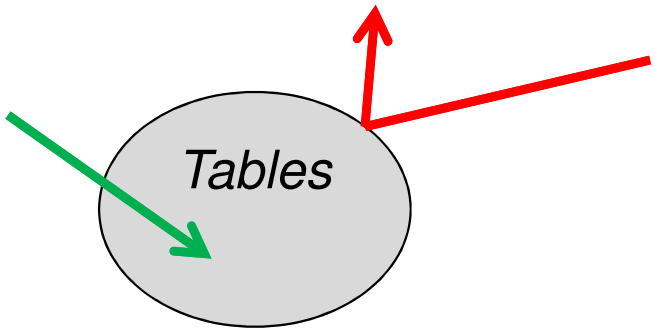
- Know how to connect to and use a database from external applications.
  - … using JDBC

Workflow – Application

- Write the external application using a provided interface with related priviliges.
  - Write the application in a ”real” programming language.
  - Communicate with the database via some special interface (CLI, ORM, …).
- Ensure through transaction management that your application is convenient, yet safe with respect to concurrency and system crashes.

Database core

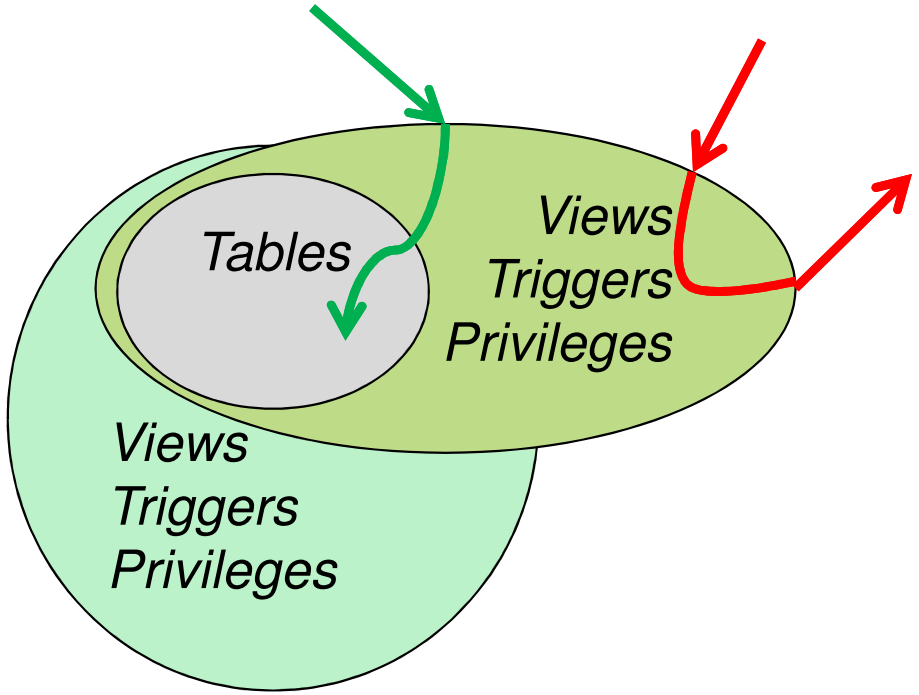
- Allow all ”possible” data
- Reject all ”unwanted” data
- As little redundancy as possible (and no anomalies!)
- ”Fool proof” – assume next layer will do stupid things.



Toolbox:
- E-R modeling
- FDs/INDs and the various NFs
- **`CREATE TABLE …`**
  - **`CONSTRAINT …`**

Database interface(s)

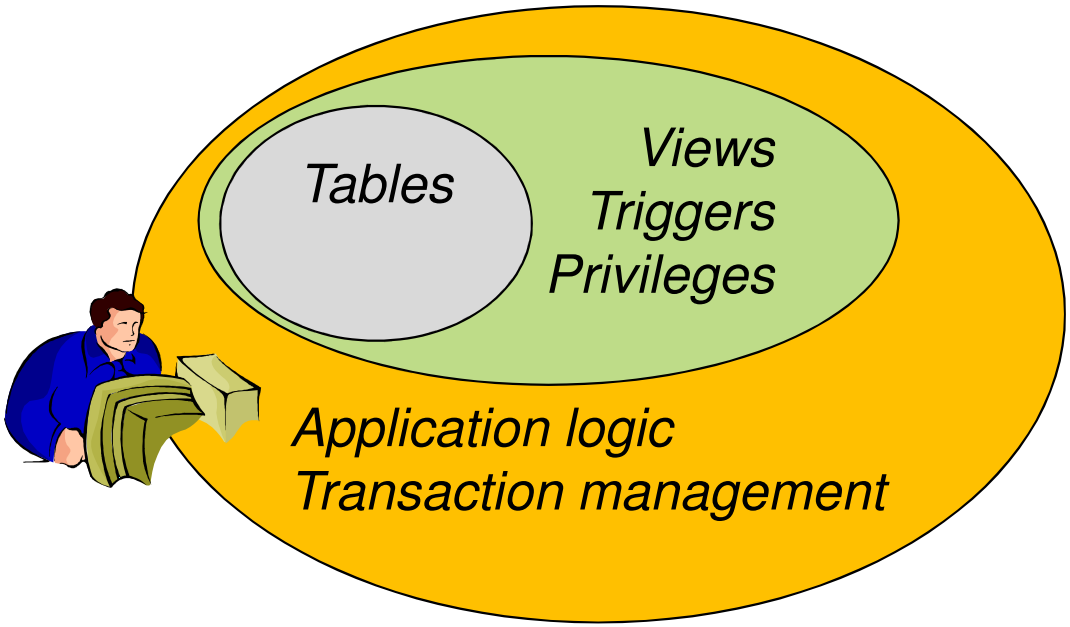
- Allow, and handle, all authorized manipulation.
- Reject forbidden manipulation.
- Consistent and safe interface(s).
- ”Fool proof” – assume application(s) will do stupid things.



Toolbox:
- SQL (views)
- SQL/PSM (triggers)
- SQL DCL (Data Control Language)

Database application

- Provide good user interface (standard programming principles apply)
- ”Fool proof” – assume user will do stupid things (e.g. cause exceptions)



Toolbox:
- JDBC (or other)
- ACID transactions
  - `SET TRANSACTION ISOLATION LEVEL …`

Lab Assignment

- Write a ”student portal” application in Java
  - Part I: Design
    - Given a domain description, design a database schema using an E-R diagram.
  - Part II: Design
    - Given a domain description, find and act on the functional dependencies of the domain to fix the schema from Part I.
  - Part III: Construction and Usage
    - Implement the schema from Part II in Oracle.
    - Insert relevant data.
    - Create views to support key operations.
  - Part IV: Construction
    - Create triggers to support key operations.
  - Part V: Interfacing from external Application
    - Write a Java application that uses the database from Part III.

Domain description

The domain that you will model in this assignment is that of courses and students at a university. So as not to make the task too large and unspecified, you will here get a description of the domain that restricts the problem somewhat. Note that the described domain is not identical to Chalmers or GU.

The university for which you are building this system is organized into departments for employees, such as the Dept. of Computing Science (CS), and study programmes for students, such as the Computer Science and Engineering programme (CSEP). Programmes are hosted by departments, but several departments may collaborate on a programme, which is the case with CSEP that is co-hosted by the CS department and the Department of Computer Engineering (CE). Department names and programme names are unique.

Each study programme is further divided into branches, for example CSEP has branches Computer Languages, Algorithms, Software Engineering etc. Note that branch names are unique within a given programme, but not necessarily across several programmes. For instance, both CSEP and a programme in Automation Technology could have a branch called Interaction Design. For each study programme, there are mandatory courses. For each branch, there are additional mandatory courses that the students taking that branch must read. Branches also name a set of recommended courses from which all students taking that branch must read a certain amount to fulfill the requirements of graduation, see below.

A student always belongs to a programme. Students must pick a single branch within that programme, and fulfill its graduation requirements, in order to graduate. Typically students choose which branch to take in their fourth year, which means that students who are in the early parts of their studies typically do not yet belong to any branch.

Courses are given by a department (e.g. CS gives the Databases course), and may be read by students reading any study programme. Some courses may be mandatory for certain programmes, but not so for others. Some, but not all, courses have a restriction on the number of students that may take the course at the same time. Each course gives a certain number of credits. For simplicity we assume all students get the same amount of credits for a given course, regardless of which study programme or branch they belong to. Courses can be classified as being mathematical courses, research courses or seminar courses. Not all courses need to be classified, and some courses may have more than one classification. Some courses have prerequisites, i.e. other courses that must be read before a student is allowed to register for the course at hand.

Students need to register for courses in order to read them. To be allowed to register, the student must first fulfill all prerequisites for the course. It should not be possible for a student to register to a course unless the prerequisite courses are already passed. It should not be possible for a student to register for a course which they have already passed.

If a course becomes full, subsequent registering students are put on a waiting list. If one of the previously registered students decides to drop out, such that there is an open slot on the course, that slot is given to the student who has waited the longest. When the course is finished, all students are graded on a scale of 'U', '3', '4', '5'. Getting a 'U' means the student has not passed the course, while the other grades denote various degrees of success.

A student administrator can override both course prerequisite requirements and size restrictions and add a student directly as registered to a course. (Note: you will not implement any front end application for student administrators, only for students. The database must still be able to handle this situation.)

For a student to graduate there are a number of requirements she must first fulfill. She must have passed (have at least grade 3) in all mandatory courses of the study programme she belongs to, as well as the mandatory courses of the particular branch that she must have chosen. Also she must have passed at least 10 credits worth of courses among the recommended courses for the branch. Furthermore she needs to have read and passed (at least) 20 credits worth of courses classified as mathematical courses, 10 credits worth of courses classified as research courses, and one seminar course. Mandatory and recommended courses that are also classified in some way are counted just like any other course. As an example, if one of the mandatory courses of a programme is also a seminar course, students of that programme will not be required to read any more seminar courses.

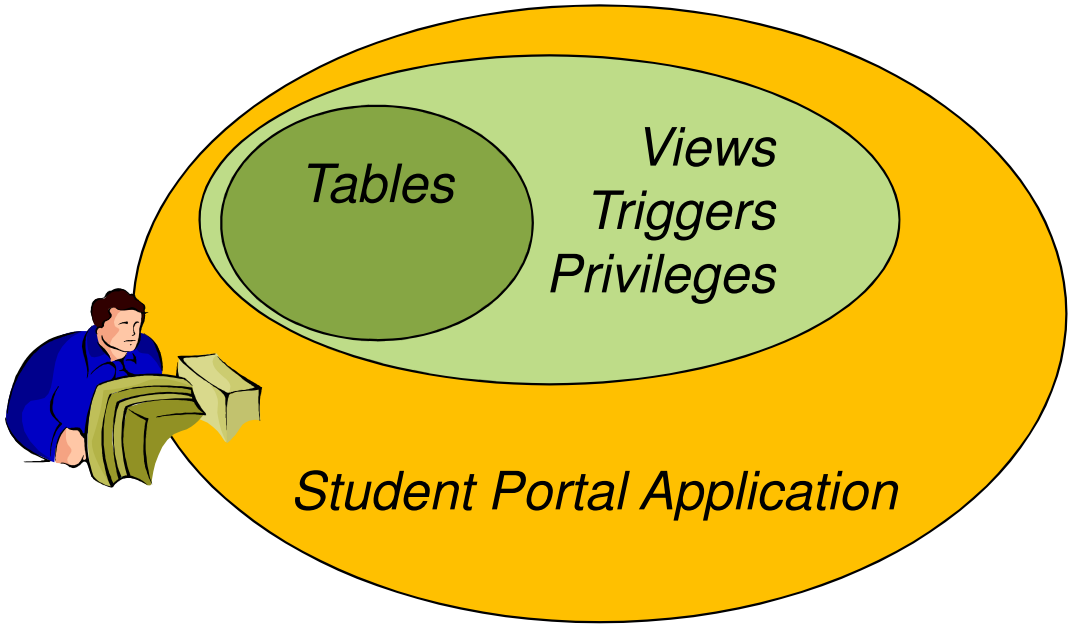
REDACTED

- 19 slides removed, showing parts of solution to assignment.

Application

- The application should only have access to the views, to give a consistent – and safe! – interface to the database.
  - No way to get info you shouldn't be privy to.
  - No way to bypass the triggers.
  - Other applications (e.g. student administrator) would have a different set of privileges (possibly with the ability to "bypass").

Student portal application



- … Questions?

Examination

- Written exam: Mar 14 (Fri) 8:30-12:30, J
  - 60 points (3/4/5 = 24/36/48, G/VG = 24/42)
  - Allowed aids: One handwritten A4 paper containing anything you like (on both sides), to be handed in together with the exam.
  - Divided into 7 distinct blocks:
    - E-R diagrams (12)
    - FDs and Normal Forms (12)
    - SQL DDL (8)
    - SQL (8)
    - Relational Algebra (6)
    - Transactions (6)
    - XML (8)

Non-standard Exam Structure

- Each block will have two or three sections: A, B and possibly C. Everyone is expected to know the A questions, while B and C questions are intended for those seeking higher grades.
- You can only get points from one section within each block!
  - Less time spent on blocks that you know well.
  - Harder to get ”stray” points.
  - A’s give ~35, B’s ~40, C’s 60.
  - If you answer more than one section, we give you the points from the one where you got the most points.

Block 1 – E-R diagrams (12)

”A small train company wants to design a booking system for their customers. …”

- Given the description, construct an E-R diagram.
- Translate a given E-R diagram into a database schema.

Note that all parts of E-R modelling belong in A!

B&C: More complex domain, analyze a diagram, reason about benefits and drawbacks of different designs for the domain, …

Block 2 – FDs and NFs (12)

”A car rental company has the following, not very successful, database. They want your help to improve it. …”

- Identify all functional dependencies you expect to hold in the domain.
- Indicate which of those dependencies violate BCNF with respect to the relations in the database.
- Do a complete decomposition of the database so that the resulting relations are in BCNF.

B&C: More complex domain, decompose to 3NF or 4NF, argue about the meaning of dependencies, analyze benefits and drawbacks of different normal forms for the domain, …

Block 3 – SQL DDL and Modifications (8)

”A grocery store wants a database to store information. After studying their domain you have come up with the following database schema. …”

- Write SQL statements that create the relations as tables in a DBMS, with all constraints.

B&C: Write a trigger for something, argue about the effects of some constraint, sketch a set of triggers that together achieve some effect, more complex variations, argue about indexes or authorization,…

You will not be required to write PSM code on the exam! But you may if you want to.

Block 4 – SQL Queries (8)

”The grocery store wants your help in getting proper information from their database. …”

- Write a number of SQL queries, given descriptions about what they should return.

Note that all parts of SQL belong in A!

B&C: more complex queries, argue about the differences of queries, rewrite a query where you must use some specific construct, …

Block 5 – Relational Algebra (6)

”Here is a schema for a database over persons and their employments. …”

- Explain (in words) what a given relational-algebraic expression will return.
- Translate a relational-algebraic expression to SQL.

Note that all operators of relational algebra belong in A!

B&C: More complex expressions, write a relational-algebraic expression given a text description, translate an SQL query into relational algebra, …

Block 6 – Transactions (6)

”Here are some transactions that run in parallel. …”

- What will the end results of these transactions be, and what could happen if they were not run as transactions?
- What isolation level should this transaction run in to ensure … ?

B&C: Show a transaction that would interfere with this under isolation level …, argue about benefits and drawbacks of using a particular isolation level for some transaction, …

Block 7 –XML (8)

”A medical research facility wants a database that uses a semi-structured model to represent different degrees of knowledge regarding the outbreak of epidemic diseases. …”

- Suggest how to model this domain as a DTD.
- Give an XML document that conforms to some DTD.
- Given this DTD, what does this XPath/XQuery expression compute?
- …

B&C: More complex queries to interpret, write an XQuery expression that computes something, discuss the benefits of the semi-structured data model for this particular domain, …

Things that will not appear on the exam!

- PSM, Embedded SQL
- JDBC (or any other interface)
- Calculations with indexes
- Grant diagrams

- Anything Oracle-specific
  - … but if you use Oracle-specific features, we will not consider that wrong.

Any more questions?

GOOD LUCK!