Lecture 1

Databases TDA357/DIT620

Pablo Picazo
pablop@chalmers.se

What's a database anyway?

A database is …

- Structured
- Persistant
- Changable
- Digital

- True to integrity constraints

DBMS

Database
==
Data collection managed by a
specialized software called a
Database Management System (DBMS)

Why a whole course in Databases?

Banking, ticket reservations, customer records, sales records, product records, inventories, employee records, address books, demographic records, student records, course plans, schedules, surveys, test suites, research data, genome bank, medicinal records, time tables, news archives, sports results, e-commerce, user authentication systems, web forums, www.imdb.com, the world wide web, …

Databases are everywhere!

Examples

- Banking
  - Drove the development of DBMS
- Industry
  - Inventories, personnel records, sales …
  - Production Control
  - Test data
- Research
  - Sensor data (25GB/h for a car)
  - Geographical data
  - Laboratory information management systems
  - Biological data (e.g. genome data)

Why not a file system?

File systems are

- Structured
- Persistant
- Changable
- Digital

… but oh so inefficient!

Modern DBMS

- Handle *persistent* data
- Give *efficient* access to huge amounts of data
- Give a *convenient* interface to users
- Guarantee *integrity* constraints
- Handle transactions and concurrency

Database Management Systems

- Hierarchical databases:
  - ”Easy” to design if only one hierarchy
  - Efficient access
  - Low-level view of stored data
  - Hard to write queries
- Network databases:
  - ”Easy” to design
  - Efficient access
  - Low-level view of stored data
  - Very hard to write queries

Database Management Systems

- Relational databases:
  - **Hard to design**
  - Use specialized storage techniques
  - Efficient access
  - Provides high-level views of stored data based on mathematical concepts
  - **Easy to write queries**
  - Not all data fit naturally into a tabular structure
- Other databases (”NoSQL”):
  - Some based on semantic data models
  - Object-oriented database management systems (OODBMS)
  - XML-based, Key-value based, …

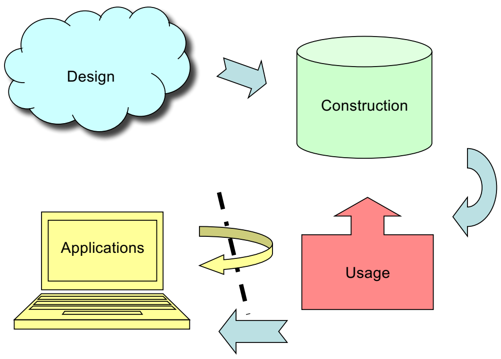
Relational DBMSs

- Very simple model
- Familiar tabular structure
- Has a good theoretical foundation from mathematics (set theory)
- Industrial strength implementations, e.g.
  - Oracle, Sybase, MySQL, PostgreSQL, Microsoft SQL Server, DB2 (IBM mainframes)
- Large user community

Database system studies

1. Design of databases, e.g.
   - Entity-Relationship modelling
   - relational data model
   - dependencies and normalisation
   - XML and its data model
2. Database programming, e.g.
   - relational algebra
   - data manipulation and querying in SQL
   - application programs
   - querying XML
3. Database implementation, e.g.
   - indexes, transaction management, concurrency control, recovery, etc.

Course Objectives



Course Objectives – Design

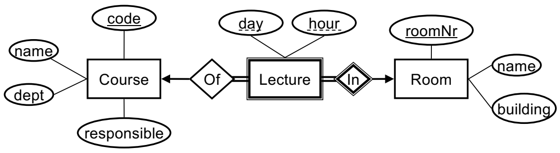
When the course is through, you should

– 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

- Entity-relationship (E-R) diagrams
- Functional Dependencies
- Normal Forms



Course Objectives – Construction

When the course is through, you should

– Given a database schema with related constraints, implement the database in a relational DBMS

```
Courses(code, name, dept, examiner)
Rooms(roomNr, name, building)
Lectures(roomNr, day, hour, course)
  roomNr -> Rooms.roomNr
  course -> Courses.code
```

Course Objectives – Construction

- SQL Data Definition Language (DDL)

```
CREATE TABLE Lectures (
  lectureId INT PRIMARY KEY,
  roomId REFERENCES Rooms(roomId),
  day INT check (day BETWEEN 1 AND 7),
  hour INT check (hour BETWEEN 0 AND 23),
  course REFERENCES Courses(code),
  UNIQUE (roomId, day, hour)
);
```

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

”Add a course ’Databases’ with course code ’TDA357’, given by …”

”Give me all info about the course ’TDA357’”

Course Objectives – Usage

- SQL Data Manipulation Language (DML)

```
INSERT INTO Courses VALUES
('TDA357', 'Databases','CS', Mickey');
```

- Querying with SQL

```
SELECT * FROM Courses WHERE code = 'TDA357';
```

Course Objectives – Applications

When the course is through, you should

– Know how to connect to and use a database from external applications

”We want a GUI application for booking rooms for lectures …”

Course Objectives – Applications

- JDBC

```
// Assemble the SQL command for inserting the
// newly booked lecture.
String myInsert = ”INSERT INTO Lectures ”
   + ”VALUES (” + room + ”, ”
   + day + ”, ” + hour + ”, ” + course + ”)”;

// Execute the SQL command on the database
Statement stmt = myDbConn.createStatement();
stmt.executeUpdate(myInsert);
```

Course Objectives - Summary

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 organisation

- 7 weeks
  - (Week 44-50: 31 October - 15 December)
- Lectures
- Exercise sessions
- Project with lab sessions
- Exam

Course organisation: Lectures

Week 44 (This week)

	Mon	Tue	Wed	Thu	Fri
08:00		JB	JB		
10:00		JB			JB
12:00					
14:00					
16:00					

Course organisation: Lectures

Week 45-end

	Mon	Tue	Wed	Thu	Fri
08:00			JB		
10:00		JB			
12:00					
14:00					
16:00					

Course organisation: Exercises

Week 45-end

	Mon	Tue	Wed	Thu	Fri
08:00					1JB
10:00					1JB+1LH
12:00					
14:00					
16:00					

Course organisation: Labs

Week 45-end

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
08:00		3JB			1LH		
10:00			3JB				
12:00					3JB		
14:00					3JB		
16:00							deadline

About the rooms and campuses

- Large amount of students (~200)
- Necessarily splitting over several rooms for labs/exercises
- Rooms change often! Check your schedules
- Doodles!!

Who prefers exercise/lab session at Johanneberg/Lindholmen?

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 PostgreSQL.
    - Insert relevant data.
    - Create views to support key operations.
  - Part IV: Construction
    - Create triggers to support key operations.
  - Part V: Interfacing from external Application (tests objective S7)
    - Write a Java application that uses the database from Part III.

Lab Assignment (cont.)

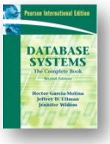
- The assignment is graded and is a requirement to pass the course
- Groups of 2 students
- First 4 tasks are graded using the Fire system, deadline each time on Sunday
- The final task is assessed on/before the last lab session.

As soon as you have the Assignment, make a demo before the last session

Course Book

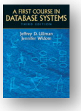
"Database Systems:
The Complete Book, 2E",
by Hector Garcia-Molina,
Jeffrey D. Ullman,
and Jennifer Widom

Approx. chapters 1-12



Alternative versions

"First Course in Database Systems,
A, 3/E" by Jeffrey D. Ullman and
Jennifer Widom



"Database Systems:
The Complete Book", by Hector
Garcia-Molina, Jeffrey D. Ullman,
and Jennifer Widom

Approx. chapters 1-8

Web Resources

- Website (Google TDA357, first hit)
  http://www.cse.chalmers.se/edu/course/TDA357/HT2017/
  - Slides of lectures + prev years (even course notes)
  - Exercise sessions + solutions
  - Lab assignment
  - Extra information
  - Old exam questions and solutions
- Google group
  https://groups.google.com/group/tda357-ht2017
  - Announcements, questions/answers, other information
  - **Sign up TODAY!**

Teaching staff

- Lecturer/Course responsible: Pablo Picazo
- Professor/Examiner: Graham Kemp
- Course assistants:
  - Markus Aronsson (mararon@chalmers.se)
  - Herbert Lange (herbert.lange@cse.gu.se)
  - Timon Lapawczyk (timon.lapawczyk@gmail.com)
  - Selpi (selpi@chalmers.se)
  - Andrea Vezzosi (vezzosi@chalmers.se)
  - Hamid Ebadi (hamid.ebadi@chalmers.se)
  - Evgeny Kotelnikov (evgenyk@chalmers.se)

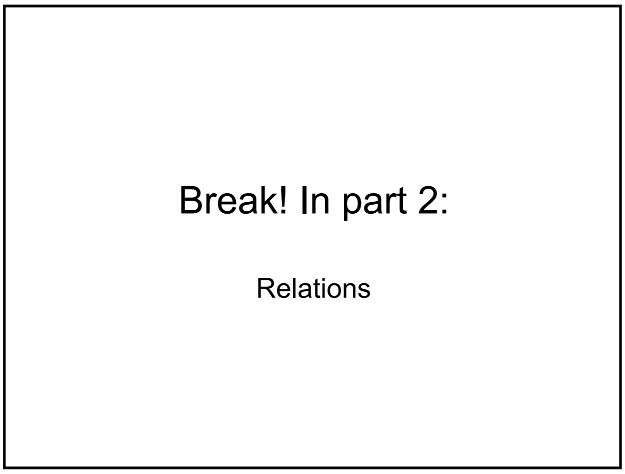
TODO for you

- Locate the course website
- Sign up the Google group
- Find a lab partner

**Failure is the key to success
Success is made of 99% failure**

make lots of mistakes and learn from them

but stop before the exam!



Break! In part 2:

Relations