Syllabus

Lecturer

- Werner Nutt, nutt@inf.unibz.it, Room POS 2.09

  Office hours: Tuesday, 14:00–16:00 and by appointment

  (If you want to meet up with me, send me an email or catch me after the lectures to make an appointment. In that way we can be sure that you don't have to wait until other students have finished their meetings.)

Teaching Assistants

- Lina Lubyte, lubyte@inf.unibz.it, Room POS 2.02

  Office hours: Friday, 16:00–17:30 and by appointment

- Marius Kaminskas, Marius.Kaminskas2@unibz.it

  Office hours: to be determined and by appointment

Timetable

Day	Time	Event	Room
Tuesday	8:30–10:30	Lecture	F0.03
Tuesday	14:00–16:00	Lab (Lubyte)	E431
Wednesday	10:30–12:30	Lab (Kaminskas)	E531
Thursday	10:30–12:30	Lecture	E422

- There will be two lab groups. Each student will be assigned to one group and will attend the lab sessions of that group.
- Lab sessions will start on 12/13 October.

Course Pages on the Web

We will use the Teleacademy e-learning environment system to run the course. You find it under

`http://www.teleacademy.it`

and can login using your university login and password. Then you have to register for the course "Introduction to Databases".

This year, we are using Teleacademy for the first time and we are not yet familiar with the services it offers. Therefore, in parallel, I will maintain some course pages on the Web at

`www.inf.unibz.it/˜nutt/IDBs1011,`

where I publish this syllabus, slides and other material. It may be that initially, there will be more material on these web pages than on Teleacademy. Please, consult both!

On the Web, you can also have a look at last year's material at

`http://www.inf.unibz.it/˜nutt/TeachingPast/IDBs0910`.

Aims

Virtually every software system has to manage significant amounts of data. Many tasks regarding the management and the manipulation of data can be executed by specialised software systems, called Database Management Systems (DBMS). To be able to use DBMS's successfully, one has to understand the concepts on which they are based. The aims of this course are

- to introduce you to the *basic concepts* underlying a DBMS;
- to show how they are realized in *specific systems* such as the PostgreSQL DBMS;
- to give you some *hands-on experience* in using the PostgreSQL DBMS.

Course Content

- Fundamental Database Concepts
- The Entity Relationship (= ER) Model
  - Conceptual Database Design
- The Relational Model
  - Relations and Integrity Constraints
    (Keys and Foreign Keys)
- Logical Database Design
  - ER to Relational Schemas
- Relational Algebra: an Algebraic Query Language for the Relational Model
- SQL: Querying and Manipulating Data
  - SQL Data Definition Language
  - Single Block Queries
  - Aggregation
  - Joins and Outer Joins
  - Nesting
  - Negation
- Transaction Management and Concurrency Control
- Database Access from a Programming Language: JDBC

- Data Storage and Indexing
  - File Organisation and Indexes
  - Tree-structured Indexing: B+-trees
  - Indexes in PostgreSQL
- Query Evaluation
  - Sorting
  - Evaluation of Relational Operators
  - Query Optimisation
  - Physical Database Design
- Query Plans in PostgreSQL
- Functional Dependencies and Normalisation

Format of the Course

The course has three main ingredients:

- Lectures
- Exercises
- Group Projects

Lectures

The lectures will introduce *new material.* The lectures will largely follow the structure of the book

> *A First Course in Database Systems*
> by Jeff Ullman and Jennifer Widom.

There are several copies in the library. Students are expected to work with the book to consolidate and revise the material of the lectures.

The sections on data storage, indexing and query execution will be based on

> *Database Management Systems*
> by Raghu Ramakrishnan and Johannes Gehrke.

As the lectures proceed, copies of the slides will be available at Teleacademy and at the course web pages.

Labs

There will be two kinds of activities during the labs:

- Solving exercises
- Presentations and discussions on the group projects.

During the lectures, a work sheet will be handed out with *exercises* on the subjects of the lectures. Students are expected to prepare for these exercises and may be asked to present their solution at the blackboard. Solutions will be discussed at the tutorials.

Some of the questions of the final exam will be very similar to the lab exercises. Thus, solving the exercises is an excellent preparation for the exam.

Students will also present work on their group projects and receive feedback and support.

Group Projects

The goal of the group projects is to apply the concepts and techniques presented in the lectures to a small database application. You will work on a project in groups of *three or four* students. (The size of the groups will depend on the number of students that register for the projects.) Each group will be supported by one of the two teaching assistants.

Organisation of Groups: In order to take part in the group projects, you will have to *register*. You can do that by entering your name (first and last) into the doodle poll at

`http://www.doodle.com/krvawphwhdvhaiy2`,

no later than *Tuesday, 5 October*.

On 6 October, we will define the groups. In the lecture of 7 October I will inform you who are the members of your group and who is the TA supporting your group.

Individual Database Application (IDA): Each group will develop an Individual Database Application (IDA). You are free to choose the topic of your IDA by yourself. However, no two groups can work on the same topic.

The groups have to inform us by *Monday, 11 October* on their topic. It may be that your topic will not be accepted because it is not suitable for a project or because it is to similar to the topic of another group. (Specific instructions how to do all this will follow.)

The project will consist of the following steps:

1. Specifying the application and writing up data requirements
2. Designing a conceptual model in the form of an Entity Relationship diagram (= conceptual design)
3. Translating the conceptual model into a relational schema
4. Implementing the relational schema in PostgreSQL and populating the database
5. Querying and modifying the database by suitable SQL statements
6. Optimising the access to data by adding indexes to the relational schema (= physical database design)
7. Writing a Java client that accesses the DB via JDBC, allowing a user to query, insert, delete, and modify data
8. Experimenting with PostgreSQL's concurrency control mechanisms.

Project Wiki: Each group will submit documents on their project and progress reports using Teleacademy. The work will be visible to all other students of the course so that you can learn from the work of other groups.

Milestones: There will be six *milestones* for the project:

Week 2: Group registered and topic of project defined;

Week 4: Data requirements and conceptual model;

Week 6: Translation into relational schema, implementation of the schema, population of the schema with data

Week 8: SQL queries over the database

Week 10: JDBC client that runs transactions on the database.

Week 12: Physical design, analysis of query execution plans, performance analysis

The *deadline* for each milestone is 11:30 pm on Monday of the following week. You submit your work by uploading it to Teleacademy. You will receive a mark for the work that can be found at that time at your home page. If there is no work, the mark will be 0.

Conflicting Topics: All groups have to work on different topics. If two groups choose the same topic, the group that registered second will be informed to choose a different one.

Project Groups and Lab Groups. Project groups will present and discuss their progress in the labs. Therefore, all members of the project should attend the same lab.

Presentations in the Labs: The labs in week 4 (19/20 October) will be devoted to the data requirements and conceptual model and the relational schema. Each group will give a short presentation. All participants of the tutorial are encouraged to discuss the project presented. The teaching assistant will ask questions about the planned project so that problems with the design can be identified as soon as possible.

More detailed instructions for the coursework will be given to you later during the term.

Reading

There are a number of good introductory textbooks around and most of them cover the more or less the same material. This list contains a small choice of them. For each book there is a web site that provides additional material.

- *A First Course in Database Systems* by Jeff Ullman and Jennifer Widom.
  This is the book on which by and large the lectures are based.

- *Database Management Systems* by Raghu Ramakrishnan and Johannes Gehrke.
  The sections on data storage, indexing and query execution will be based on this book.

- *Database Systems: The Complete Book* by Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom.
  This an extension of the book by Ullman and Widom by several chapters on the implementation of database management systems
- *Database Systems: Concepts, Languages and Architectures* by Paolo Atzeni, Stefano Ceri, Paolo Paraboschi and Riccardo Torlone.
- *Database System Concepts* by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan.
  This is book gives up to date information about recent technology. The part that discusses how database concepts are realized in various commercial systems is of particular interest.

Assessment

There will be a written exam for the course.

Moreover, students will receive a mark for the project work. First, each group will receive a mark for the project as a whole. By default, this will also be the mark of each group member. However, this default mark needs to be confirmed.

To this end, each group will have a final project discussion in January with their TA and myself where we ask questions about the work. We expect that each member of the group should be familiar with the outcome of the project and should be able to answer questions about it. The confirmed mark of a student can be higher or lower than the initial project mark.

The final mark will be computed as

$$\max\,\{0.3\,\text{project mark} + 0.7\,\text{exam mark},\\ 1.0\,\text{exam mark}\}.$$

In other words, the final mark is at least as good as the exam mark. If the project mark is better than the exam mark, then the final mark is composed of the mark for the project (contributing 30%) and the mark for the exam (contributing 70%).