

Information Systems (Informationssysteme)

Jens Teubner, TU Dortmund
`jens.teubner@cs.tu-dortmund.de`

Summer 2019

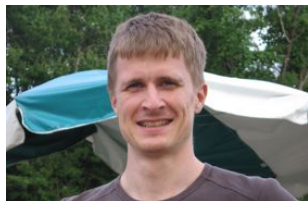
A Few Words About Me

Jens Teubner

DBIS Group (LS 6)

Otto-Hahn-Strasse 14, Room 333

jens.teubner@cs.tu-dortmund.de



1996–2001 Diploma in Physics, U Konstanz

2001–2005 Research assistant, DBIS Group, U Konstanz

2005–2007 Research assistant, Database Group, TU München

Oct 2006 PhD in Computer Science (XML query processing)

2007–2008 Postdoc, IBM T. J. Watson Research Center, NY, USA

2008–2013 Senior Researcher, Systems Group, ETH Zurich

since 4/2013 Full Professor, DBIS Group, TU Dortmund University

Topic: Database systems on modern computing hardware

Example: Library of Congress (<http://www.loc.gov/>)

In 2011:

- **151.8 million items** held
 - 34.5 million books
 - 66.6 million manuscripts
 - recordings, maps, sheet music, ...
- **22,000 items** received **per day**
($\approx 10,000$ are added to collection)
- 1.7 million on-site visitors
- website: 73.4 million visits,
512 million page views



The Information Age

Example: Google

- over **1 billion** searches per day
- response time: **1/4 second**
- **20–30 % of web content is new** every time Google crawls it
- **100 hours** of video uploaded to YouTube **every minute**

Imagine an engineer being paged at 4am, because there are only a few **petabytes** of storage space left.



Google Zurich location

In this course you'll learn how to

model, store, and process data

data in an efficient and scalable manner.

We'll look at

- good ways to **model** your data from an application perspective,
- the role of **database systems**,
- how you **access** and **query** them,
- how **multiple users** can access a database at the same time,
- how a database can guarantee **consistency** and **durability**, and
- what a database does to **find** your data quickly.

Lecture:

- Wednesdays, 16–18h, Room H.001, Seminarraumgebäude (SRG)
- Course website: <http://dbis.cs.tu-dortmund.de/cms/en/teaching/ss19/infosys/>

Please visit this website **regularly**. We will frequently post new information during the semester.

Exercises:

- Organizer: **Thomas Lindemann** (thomas.lindemann@cs.tu-dortmund.de)
- Register via **AsSESS** to one of the exercise groups.
- Exercises start **next week**.

Surviving the Exam

There will be a **written exam** (60 min) at the end of the semester.

- dates: **July 17, 2019**; 2nd date: **September 12, 2019**
- **material allowed**: one sheet of A4 paper, **handwritten**

Best preparation for the exam? Do the exercises!

- Do exercises **before** they are discussed in the group.

“I don’t understand this one thing. I need help!”

- Don’t hesitate to ask me or your TA.
- Speak up during the lecture!
- Visit the **HelpCenter** (OH 12, 4th floor, Mon AM/PM, Tue/Wed PM).

I will post all **lecture slides** on the course web site.¹

Good **text books**:

- A. Kemper and A. Eickler. *Datenbanksysteme*. Oldenbourg-Verlag.
- R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill.
- R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Prentice Hall. (in German: *Grundlagen von Datenbanksystemen*. Pearson Studium.)
- A. Heuer, K.-U. Sattler, and G. Saake. *Datenbanken: Konzepte und Sprachen*. mitp.

... and many more (this is a standard course, taught world-wide).

¹Except parts that I mark with  on the slide.

Experiment with a Database!

I **strongly** recommend you exercise the material of this course on a **real database system**.

Examples:

- **Oracle** (<http://www.oracle.com/us/products/database/>)
 - Used in the exercises for this course.
 - More details in the exercise groups.
- **IBM DB2** (<https://www.ibm.com/analytics/us/en/db2/>)
 - Full-featured, industry-strength database
 - Available **for free** (Win/Linux)
- **PostgreSQL** (<http://www.postgresql.org/>)
 - Very powerful and feature-rich **open source** database

- 1 Overview of database systems
- 2 Database design (3-tier architecture, ER diagrams)
- 3 The relational model (relational algebra, relational calculus)
- 4 SQL (Structured Query Language)
- 5 Normal forms
- 6 Transaction management (ACID properties, serializability)
- 7 Crash recovery (ARIES/write-ahead logging)
- 8 Semi-structured data (XML)
- 9 Database implementation (memory hierarchy, B-trees)