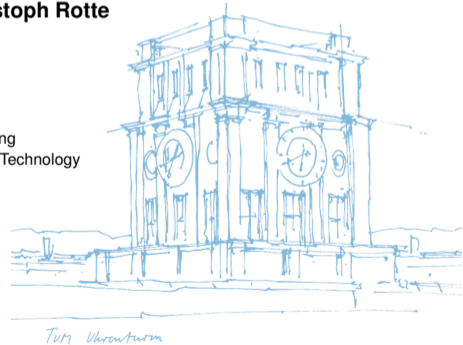


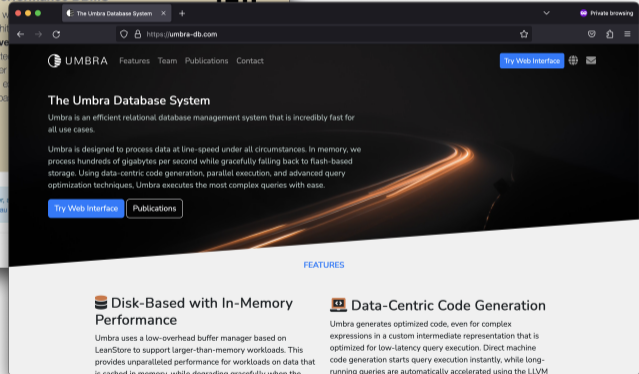
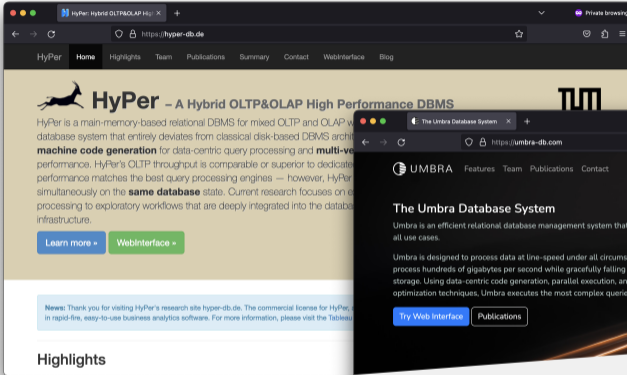
Practical Course: Database Implementation

Adrian Riedl, Simon Ellmann, Christoph Rotte

Tuesday 7th July, 2026

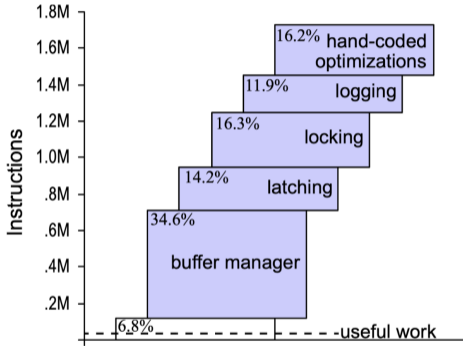
Chair of Data Science and Engineering
TUM School of Computation, Information and Technology
Technical University of Munich





- We build a tiny **main-memory compiling database system!**
- We focus on correctness and *speeeeed*¹ ...

¹ Warning: Query compilation will be slow



OLTP properties and new platforms	DBMS modification
logless architectures	remove logging
partitioning, commutativity	remove locking (when applicable)
one transaction at a time	single thread, remove locking, remove latching
main memory resident	remove buffer manager, directory
transaction-less databases	avoid transaction bookkeeping

Figure 1: OLTP Through the Looking Glass, and What We Found There by Harizopoulos et al. (SIGMOD '08)

Demo

- **Data representation** in memory and **query execution**
- Building a **shell** for our DBMS
- **Lexing** and **parsing** SQL
- **Algebra trees** and **query optimization**²
- **Code generation** and **execution**
- **Multithreading**

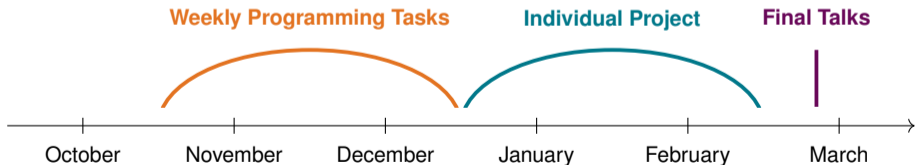
² Only predicate pushdown

Weekly Programming Tasks (~ 7 weeks):

- **Weekly meetings** and **assignments**

Individual Project (~ 7 weeks):

- Work on a state-of-the-art topic
- Present **your work** → **Final Talks**



Weekly Programming Tasks:

- **1-2 students present their solution** to the previous assignment
- Introduction to a **new topic + assignment**
- Code **should** be submitted before the next meeting

Individual Project:

- Try to **reproduce a paper**, *or*
- **Implement some fancy feature** in your database, *or*
- Work on a **topic provided by a PhD student** at the chair

→ No more weekly course meetings, instead individual ones with your advisor

Final Talk:

- **Present your project** in 10 to 15 minutes
- **On-site meeting** for all presentations + discussions (usually ~ 3 h)
- Date for the meeting is up to you (e.g., end of lectures vs. end of semester)



Tasks:

- We provide templates for the programming tasks in C++ and Rust
- The tasks build on each other → Decide early on the programming language!
- You may use helper libraries / crates (e.g., for text formatting, error handling, ...) that do not affect the overall implementation effort

Grading:

- Programming tasks and individual project (including the final talk) each count for roughly 50% of the grade
- We will grade after the final presentations (which is also the deadline for your code)

- ~~Solve the qualification challenge: <https://db.in.tum.de/teaching/ws2627/imlab/>~~
- Send us a **short e-mail** to imlab@db.cit.tum.de telling us:
 - **Why** you want to take part in the practical course (two to three sentences max)
 - What you **know about database systems** (attended lectures, practical experience, ...)
 - Your **experience with systems / low-level programming** (e.g., C/C++/Rust, memory management, performance tuning)
- **Deadline: 19th of July**
- Register via the matching system!

Questions?