# Seminar: Modern Database Systems

Organizational Meeting

Chair of Database Systems
Garching, January 28, 2020

# Overview

**Weekly Meeting**

- Wednesdays 16:00-18:00 (first session on 22.04.2020)
- Room 02.09.014
- 2 presentations per meeting
- There will be an attendance log

**Required Work**

- Seminar paper (5 pages) – 60% of the final grade
- Presentation (20 minutes + 10 minutes discussion) – 30% of the grade
- Moderate one discussion (act as the “devil‘s advocate“) – 10% of the grade

# Organization and Due Dates

**Check-in via email or personally**

1. Check in soon after matching for paper recommendations (preferences considered FCFS)
2. Check in when rough structure is planned
3. Check in when first draft is ready

**Due Dates**

- Structure: ca. 4 weeks prior to presentation date
- Presentation slides: 1 week prior to presentation date
- Seminar paper: 2 weeks after presentation date (strict!)

# Topics

**Block 1: OLTP**

- Hekaton: SQL Server‘s Memory-Optimized OLTP Engine (SIGMOD 2013)
- Speedy Transactions in Multicore In-Memory Databases (SOSP 2013)
- Fast Serializable Multi-Version Concurrency Control for Main-Memory Database Systems (SIGMOD 2015)
- SharedDB: Killing One Thousand Queries with One Stone (VLDB 2012)

**Block 2: OLAP**

- MonetDB/x100: Hyper-Pipelining Query Execution (CIDR 2005)
- Efficiently Compiling Efficient Query Plans for Modern Hardware (VLDB 2011)
- Relaxed Operation Fusion for In-Memory Databases: Making Compilation, Vectorization, and Prefetching Work Together at Last (VLDB 2018)
- Quickstep: A Data Platform Based on the Scaling Up Approach (VLDB 2018)
- Everything You Always Wanted to Know About Compiled and Vectorized Queries But Were Afraid to Ask (VLDB 2019)

# Topics

**Block 3: In the Cloud**

- Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases (SIGMOD 2017)
- The Snowflake Elastic Data Warehouse (SIGMOD 2016)
- Spanner: Google‘s Globally Distributed Database (OSDI 2012)
- Obladi: Oblivious Serializable Transactions in the Cloud (OSDI 2018)

**Block 4: Hybrid Transactional Analytical Processing (HTAP)**

- HyPer: A Hybrid OLTP&OLAP Main Memory Database System Based on Virtual Memory Snapshots (ICDE 2011)
- SAP HANA Database – Data Management for Modern Business Applications (SIGMOD 2011)
- BatchDB: Efficient Isolated Execution of Hybrid OLTP+OLAP Workloads for Interactive Applications (SIGMOD 2017)
- Columnstore and B+ tree - Are Hybrid Physical Designs Important? (SIGMOD 2018)

# Topics

**Block 5: Beyond Relational Databases**

- Naiad: A Timely Dataflow System (SOSP 2013)
- Differential Dataflow (CIDR 2013)
- EmptyHeaded: A Relational Engine for Graph Processing (SIGMOD 2016)
- LevelHeaded: A Unified Engine for Business Intelligence and Linear Algebra Querying (ICDE 2018)
- Froid: Optimization of Imperative Programs in a Relational Database (VLDB 2018)

# Why is reading research papers relevant?

**Understand the latest developments**

- Before any book covers them

**Learn how to write and communicate ideas**

**Read critically**

- Ask the right questions, challenge the assumptions

**Synthesize new ideas**

- Reading is the most effective way to generate novel research ideas

# Some model questions

**What is the research question that the paper addresses?**

- What is the motivation? Is it relevant? What is the impact if it is solved?

**What are the contributions?**

- How do they build on previous work? Is it something new? Can it be generalized?

**What do I learn by reading it?**

- For instance, even a good summary of related work can be worth a lot!

**How are the results substantiated?**

- Experiments, proofs, benchmarks, etc. Is the evaluation thorough?

**What are the conclusions and broader impact?**

- What can be built on top of it? What is an interesting follow-up / future work?

**Tip: You should be able to answer all of these questions before you present the paper!**

# How to read a paper

**Read a paper at least three times**

**First pass: get the general idea**

- 5-10 minutes
- Read abstract, introduction, conclusions, (sub-)section headings
- Briefly check the references

**Second pass: understand the content**

- Approximately 1 hour
- Read the full paper, ignore the details (e.g., proof, etc.)
- Find key points, take notes, check figures carefully to understand them
- Mark references for further reading

**Third pass: understand the depth**

- 4-5 hours
- Fully understand everything, attention to detail (check related work)
- Try to virtually re-implement the paper
- Question everything!
- Generate ideas for your work.

# Read the related work

**If you do not have the required background knowledge then check the related work**

**Find the related work**

- Check the related work section
- Try to find a survey paper of that area (or textbook)
- Check the author‘s more recent work
- Check top conferences in the field and recent publications
- Check later papers that cite the paper (e.g., Google scholar, ACM digital library)

# A good (research) talk

**Is centered around the audience (not you)**

- Teaches, engages, provokes, and excites the listeners

**Provides intuitions to the audiences**

**Should make them want to read the paper**

- But, not because they did not understand you

**It should not**

- Tell them all the technical details
- Cover everything you know about the topic
- Impress the audience how smart you are

# What to put in your talk

**Motivation (20%)**

- What should I listen to the talk?
- What is the problem?
- Is it an interesting problem?
- Is the proposed solution worthwile?

**The key idea (80%)**

- You must identify the key idea
- Be specific – do not leave it up to the audience to figure it out
- Be **very** specific – say „if you remember nothing else, remember this“
- Organize your talk around this specific goal. Remove everything that is irrelevant.

**Avoid shallow overviews**

- Cut to the technical „meat“ even if it covers only a part of the paper

**Use examples!**

- To motivate work, to convey basic intuition, to illustrate the idea, etc.

# What to omit

**Do not present talk outlines**

- It is not informative → a good talk has a storyline

**Do not present excessive related work**

- But, mention it in your slides. Acknowledge pre-cursors, praise the opposition…

**Do not present too many technical details**

- The audience may find it difficult to follow
- Put them in back-up slides in case somebody asks

**Do not exaggerate with animations**

- Animations are good, but can be also distracting

**Do not clutter your slides with graphics**

# Pointers for further reading

**How to read a paper:**

- S. Keshav – How to Read a Paper
- Philip W. L. Fong – How to Read a CS Research Paper?

**How to give a good (technical/research) talk:**

- Markus Puschel – How to give good technical presentations.
- Simon Peyton Jones – How to give a great research talk.