

DB 2

Summer 2024

Torsten Grust
Universität Tübingen, Germany

1 | Welcome all...

... to this course which is all about digging in the mud of database management system (DBMS) internals.

We will try to understand how DBMSs

1. **organize primary and secondary memory** (RAM and SSD/HDD) to store and query tables of data,
2. **layout tabular data** (e.g., in rows or columns) and how that influences the entire system architecture,
3. **use “data maps” (indexes)** to navigate huge tables, and
4. **internally represent, optimize, and evaluate complex queries** over tables.

But Why Would You Do That?

Insights into the DBMS kernel and its inner workings can explain, for example, ...

- why **query evaluation** takes (much) longer than expected (we are talking hours 🐢 vs. msec 😊 here),
- whether **database growth** (by factor $n > 1$) will slow down updates and/or queries by factor n (or n^2 or ...),
- whether the host's **resources** (CPU, cache, RAM, storage) are used effectively or if bottlenecks exist,
- how (known) **algorithms and data structures** hold up in the presence of huge inputs.

SQL is the Secret Key

I assume that you have working knowledge of **SQL**,¹ the ***“Intergalactic Dataspeak”*** used to manipulate and query tabular (or: relational) DBMSs.

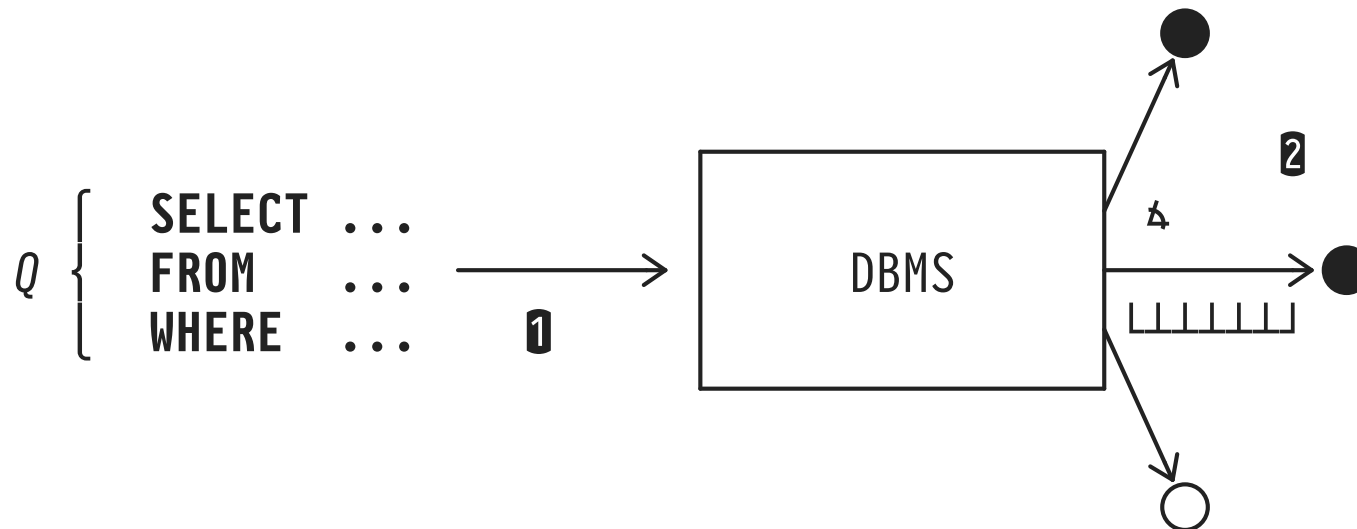
We will **submit a variety of SQL queries Q** to DBMSs and observe

- which components of the DBMS engage in processing Q ,
- how DBMS-internal data structures and algorithms contribute to execute Q efficiently.

¹ As taught in our courses “DB 1” or “Advanced SQL”, for example.

Sending “SQL Probes”

- 1 Submit SQL probe query Q to DBMS, then...
- 2 ... observe/interpret DBMS response.



Thankfully, DBMSs provide several hooks and facilities that support the observation of their operation.

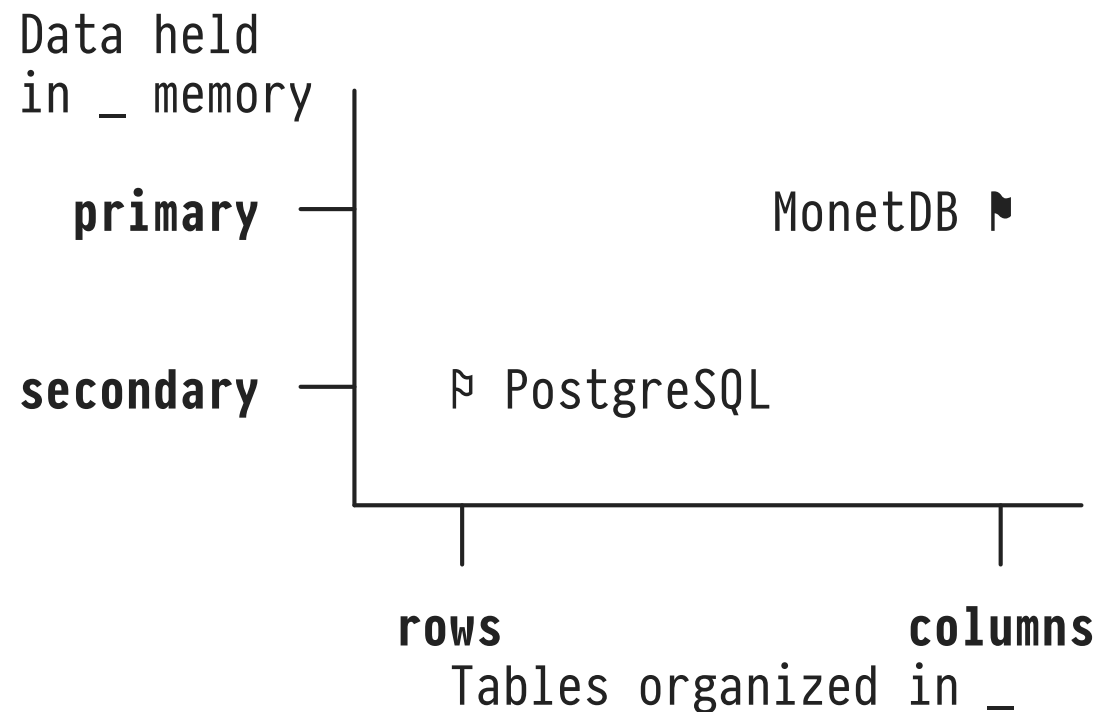
EXPLAIN Yourself, DBMS!

We will make extensive use of **EXPLAIN** facilities that act like a “DBMS X-ray.” Instead of Q , submit **EXPLAIN Q** to reveal

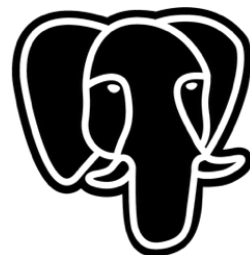
- the **evaluation order** chosen to process the expressions/statements in Q ,
- the **indexes** used to access tables,
- the **cost** and **resources** (# of I/O operations, memory buffers, CPU time) expended to process Q ,
- the **algorithms** selected to access tables, manipulate rows, and compute Q 's results.

2 : Two Ends of a Spectrum: PostgreSQL and MonetDB

There is **one relational data model** but a **variety of DBMSs** that implement it. We will focus on two DBMSs, **PostgreSQL** and **MonetDB**, whose internals deviate significantly:



PostgreSQL (postgresql.org)



- Row-wise table organization (☐)
- Data held in blocks on secondary storage (e.g., SSD/HDD), database size virtually unlimited (32 TB in single table)
- Extensible database kernel (types, operations), support for a rich SQL:2011/SQL:2016 dialect
- Actively developed since 1986 (derived from Ingres)
- In this course: **PostgreSQL 15.6** (or newer should do)
- *Open source*, available on macOS 🍏, Linux 🐧, Windows 🪟

MonetDB (monetdb.org, developed at CWI Amsterdam)



- Column-wise table organization (🗄)
- Data completely mapped into primary memory for processing
—once hot database size » RAM size, MonetDB struggles
- Optimized for CPU performance
- Actively used in database research since 1993 until today
(*"the column store pioneers"*)
- In this course: **MonetDB 5 (v11.35.19)**
- *Open source*, available on macOS 🍏, Linux 🐧, Windows 🪟

On the Bare Metal


Occasionally we will discuss/develop short **program fragments in C** to understand how DBMSs—MonetDB, in particular—process data internally.

There is no need to be a C whizz, but we will encounter:

- dynamic allocation of memory (`malloc(3)`)
- array types (`τ[]`), array access, array processing
- pointer types (`τ*`), pointer operations
- control flow (conditionals, loops)
- UNIX system calls (`open(2)`, `mmap(2)`, ...).


Torsten Grust?

Time Frame	Affiliation/Position
1989–1994	Diploma in Computer Science, TU Clausthal
1994–1999	Promotion (PhD), U Konstanz
2000	<i>Visiting Researcher</i> , IBM (USA)
2000–2004	Habilitation, U Konstanz
2004–2005	Professor Database Systems, TU Clausthal
2005–2008	Professor Database Systems, TU München
since 2008	Professor Database Systems, U Tübingen

- Web: <https://db.cs.uni-tuebingen.de/grust>
- Office: WSI, Sand 13, Room B318
- Best to catch me on the [DB group's Discord server](#) 

Administrivia

Weekday/Time	Slot	Room
Monday, 10:15–11:45	Lecture	Sand 1, A301
Tuesday, 10:15–11:45	Lecture	Sand 1, A301
Wednesday, 14:15–15:45	Tutorial	Sand 1, A301



-  **No** lectures/tutorials on
 - Wed, April 17 (tutorials start on April 24)
 - Wed, May 1 (International Worker's Day)
 - Mon/Tue/Wed, May 20/21/22 (Whitsun break)

Administrivia


End-Term Exam

- 90-min **written exam** on Monday, July 22, 14:00–16:00 in N7 (Morgenstelle).
- You may bring a DIN A4 double-sided *cheat sheet*.
- Passing earns you 9 ECTS.
- Everyone can take the exam (but do work on the weekly assignments to improve your chances to pass).

Weekly Assignments & Tutorial Sessions

- We will distribute, collect, and provide feedback on **weekly assignments** (Friday→Friday) via Github .
- You work on these in **teams of two**. Hand-in again via .

Organized and run by **Tim Fischer** and **Denis Hirn**:



- Web: <https://db.cs.uni-tuebingen.de/team>
- Offices: WSI, Sand 13, Room B314 + B312
- Find both on the [Discord server](#) 

Assignments start once we have collected the first batch of interesting material, probably on April 26.

Lecture Videos, Slides, and Pieces of Code

- These **slides** (PDF) and **code fragments** (SQL, C, ...) will be uploaded to a Github  repository:

<https://github.com/DBatUTuebingen-Teaching/db2-ss24> 

- For the 2020 edition of the course, I have produced **lecture videos**:
 - 83 videos, \approx 30-min fragments.
 - Playlist on YouTube : tinyurl.com/DB2-2020
 -  Since 2020, the course has moved on—material may be added, superseded, or shuffled.
 - We do aim to make your/our time in A301 worthwhile.

Discord

During this summer semester, the **DB2 Discord** is *the* course hub:

<https://db.cs.uni-tuebingen.de/discord> 

- ⚠ **Registration** (mandatory): [/verify](#), then [/join DB2](#)
- ❓ Questions and answers (do *not* post complete solutions)
- ☁ Download additional code examples (SQL, MAL, and C)
- 💬 General discussion
- 🗨 Quick turnaround (responses often within minutes)

Course Homepage

<https://db.cs.uni-tuebingen.de/teaching/ss24/db2> 

- **Organizational matters**

Curriculum. General announcements regarding the lecture, exams, or dates. Please surf by regularly. Thank you!

- **Contact information**

Turn to Discord first. But feel free to send e-mail if you seek specific help/need to discuss personal issues with us.

Material

This course is *not* based on a single textbook but instead draws from

- a variety of scientific papers,
- textbook excerpts,
- blog and mailing list postings, [Stack Exchange](#) Q&As,²
- SQL references/standards,
- DBMS docs for [PostgreSQL](#) (★★★★★) and [MonetDB](#) (★★★★☆),
- experience, and best practices.

² All of dba.stackexchange.com/questions/tagged/{postgresql,monetdb} are worth a look.

Material (on Index Design and Usage)



- To the point, clear, and actionable advice on how to design “data maps”—or: **indexes**—and how to recognize whether a query can (not) benefit from an index.
- See use-the-index-luke.com [free HTML version] and sql-performance-explained.com [PDF 9.95€, Paperback 29.95€].