DB 2

Summer 2020

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. use **"data maps" (indexes)** to navigate huge tables,
3. internally **represent, optimize, and evaluate complex queries** over tables, and
4. **manage concurrent access to data** while avoiding inconsistencies and confusion.

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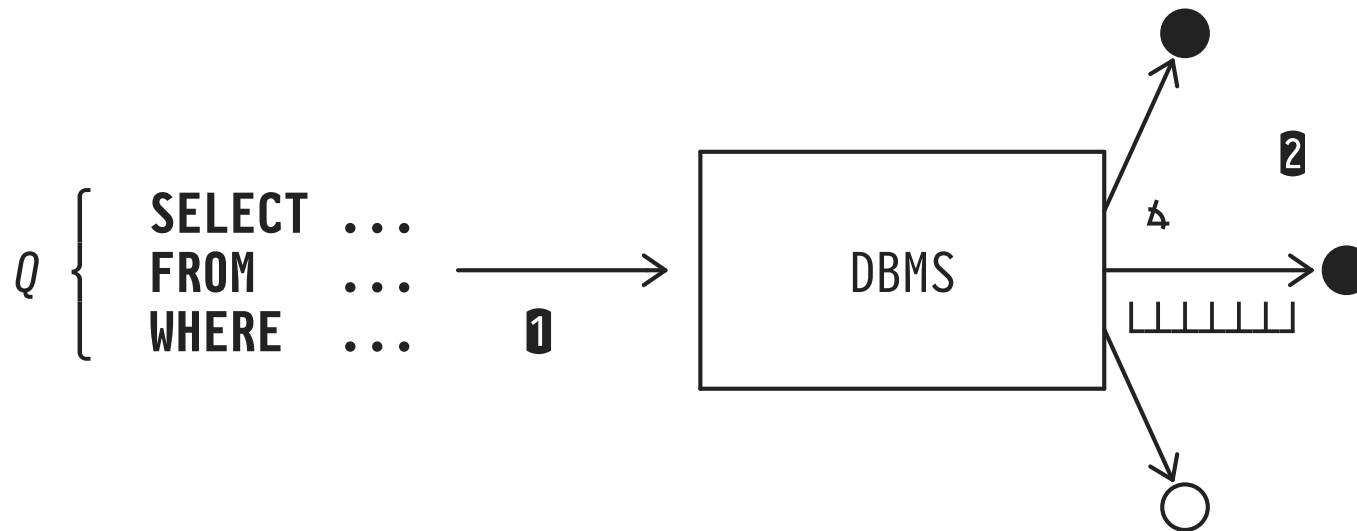
We will **submit a variety of SQL queries** Q to DBMSs and observe

- 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 "DB1" or "Advanced SQL", for example.

Sending "SQL Probes"

❶ Submit SQL probe query Q to DBMS, then...

❷ ... observe/interpret DBMS response.

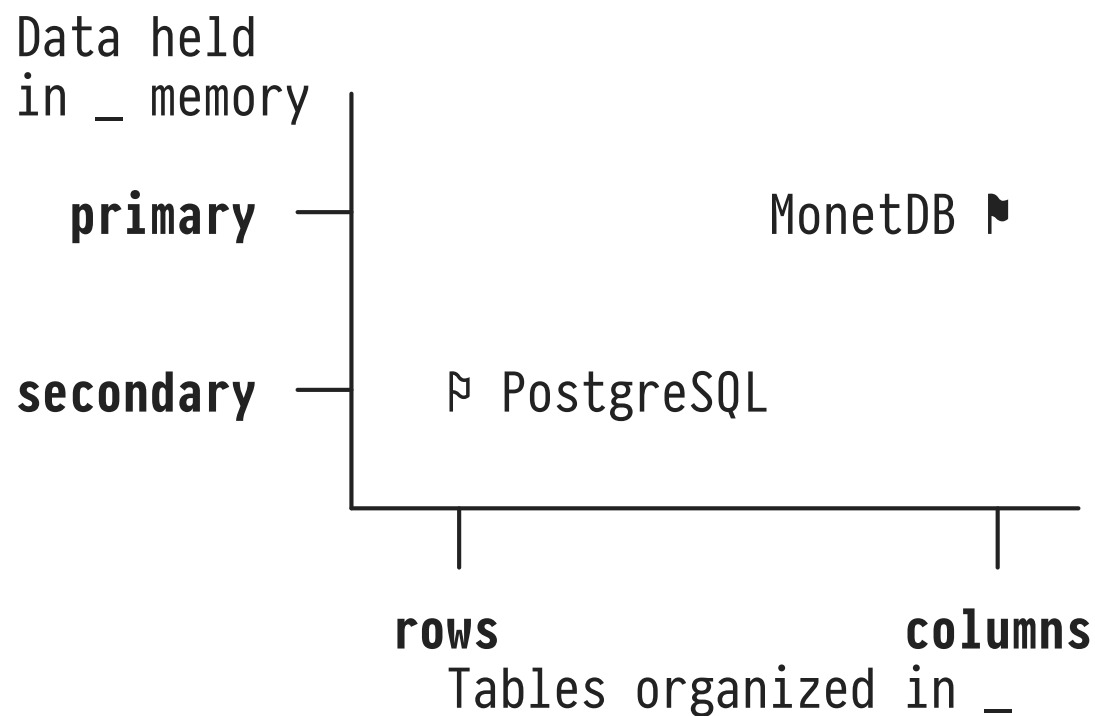Thankfully, DBMSs provide several hooks and facilities that support the observation of their operation.

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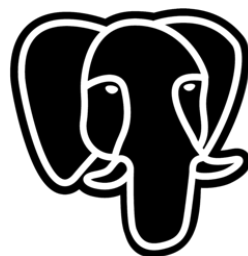
- the **evaluation order** chosen to process the expressions/statements in Q,
- the **indexes** used to access tables,
- the **cost** and **resources** (# of I/O operations, memory buffers, CPU time) expended to process Q,
- the **algorithms** selected to access tables, manipulate rows, and compute Q's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

Data held
in _ memory

primary —   MonetDB ⚑

secondary —   ⚑ PostgreSQL

rows   **columns**
Tables organized in _

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 dialect

- Actively developed since 1986 (derived from Ingres)
- In this course: **PostgreSQL 12.1** (≥ 11.x will 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**, "Nov2019-SP3")
- *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.

No need to be a C whiz, but we will encounter:

- dynamic allocation of memory (malloc(3))
- array types (τ[]), array access, array processing
- pointer types ($\tau*$), 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	*Visiting Researcher*, 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

- E-Mail: Torsten.Grust@uni-tuebingen.de
- Twitter: @Teggy (*Professor, likes database systems, programming languages, and SC Freiburg* ツ)
- WSI, Sand 13, Room B318 (home office!)

Administrativa

- To help keep the ongoing COVID-19 🦠 pandemic at bay, there will be **no in-class lectures,** at least until mid-June 2020 (*i.e.,* until the whitsun break).

- *In case* we will finish the semester in the class room, the following are the assigned lecture hall and time slots:

Weekday/Time	Slot	Room
Monday, 10:15-11:45	Lecture	Sand 6/7, F119
Tuesday, 10:15-11:45	Lecture	Sand 6/7, F119
Thursday, 14:15-15:45	Tutorial	Sand 6/7, F119

Lecture Videos, Slides, and Pieces of Code

- I will post **lecture videos** (≈ 15-min fragments) to a
  dedicated playlist on YouTube ▶. Those videos will
  - walk through the slides,
  - develop, run, and discuss SQL and C code snippets,
  - run live PostgreSQL and MonetDB experiments, and
  - expand on slide material.

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

  github.com/DBatUTuebingen/db2-ss20 ⬉

Weekly Assignments & Tutorial Sessions

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

Organized and run by **Denis Hirn** & **Benjamin Dietrich:**

- E-Mail: {Denis.Hirn,B.Dietrich}@uni-tuebingen.de
- WSI, Sand 13, Rooms B312 & B314 (home office!)

Assignments start once we have collected the first batch of interesting material, probably by the end of April. **Live, interactive tutorials** 👤←👥👥 will be announced in time.

Forum

During this loopy summer semester, the **DB2 forum** is *the* course hub and more important than ever:

forum-db.informatik.uni-tuebingen.de/c/ss20-db2

⚠ **Registration** (mandatory) and announcements
❓ Questions and answers (do *not* post complete solutions)
☁ Download additional code examples (SQL, MAL, and C)
💬 General discussion
⏱ Quick turnaround (responses often within minutes)

End-Term Exam

⚠️ Exactly when and how we can run the DB2 end-term exam is subject to contact restrictions and regulations of U Tübingen. Our current plans (as of April 20):

- 90-min **written exam** on July 27, 10:00 (Kupferbau HS 25).
- Score ≥ $\frac{2}{3}$ of the overall assignment points to be admitted to the exam and earn bonus points in the end-term exam.
- You may bring a DIN A4 double-sided *cheat sheet*.
- Passing earns you 9 ECTS.

Course Homepage

db.inf.uni-tuebingen.de/teaching/DB2SS2020.html

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

- **Contact information**
  Turn to the forum 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/{sql,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€].