

# Design and Implementation of Database System Internals

---

①

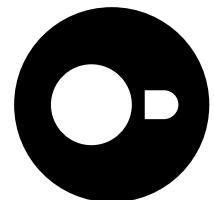
Welcome & Setup

Winter 2025/26

Torsten Grust  
Universität Tübingen, Germany

# Dissecting the Duck's Innards

---



①

Welcome & Setup

Winter 2025/26

Torsten Grust  
Universität Tübingen, Germany

# 1 | Welcome (Back)!

---

Welcome to this course which is all about digging deep into the internals of **tabular database management systems** (DBMSs).

Our tour through the DBMS kernel will touch on

- the efficient representation of data on **secondary** (SSD/HDD ) and **primary storage** (RAM )
- turning declarative SQL queries into efficient **flows of data**,
- a variety of interesting **data structures** for **sizable volumes** and associated **algorithms**,
- **modern CPUs** and how looping/branching code executes,
- various forms of **parallelism** on different levels (from single CPU instructions to threads), or
- ensuring **data integrity** under concurrent access or even if the host machine fails .

## Dissecting the Duck's Innards

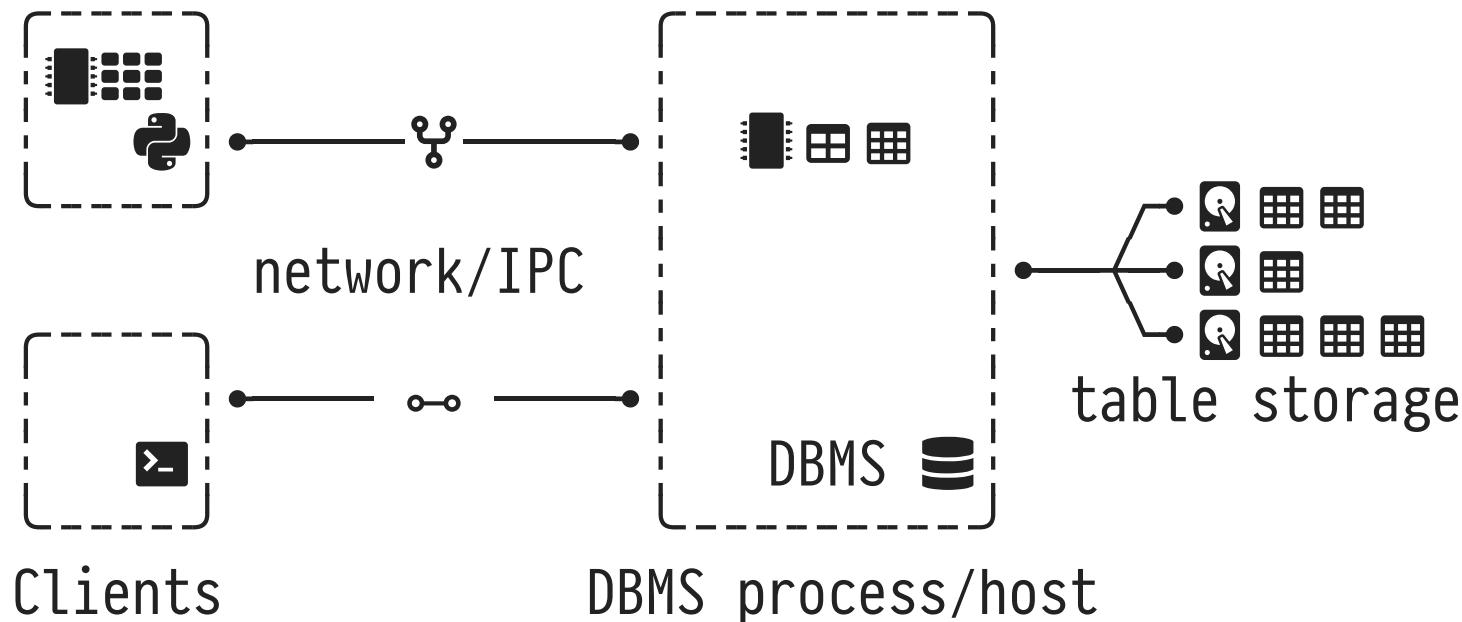
---

This semester we will focus on **DuckDB** , a contemporary tabular DBMS built for high-speed SQL-based data analytics.

- DuckDB is young, first released in June 2019 (while most DBMSs—like PostgreSQL—originate from the 1980s).
- DuckDB development is moving fast. The system both adopts newest research results and builds on established DB wisdom.
- DuckDB is developed in the open (, MIT license).<sup>1</sup> All code is available for inspection and tinkering.
- DuckDB provides a range of hooks to observe its internals.
- DuckDB comes with a CLI  and programming language APIs.
- DuckDB is easy to install and requires no maintenance.

<sup>1</sup> By now, quite a bit of DuckDB's kernel code has been developed here at U Tübingen.

## Not DuckDB: Dedicated DBMS Process/Host Isolated From Users



- DBMS controls family of disks holding table data and maps relevant table fragments into RAM buffer .

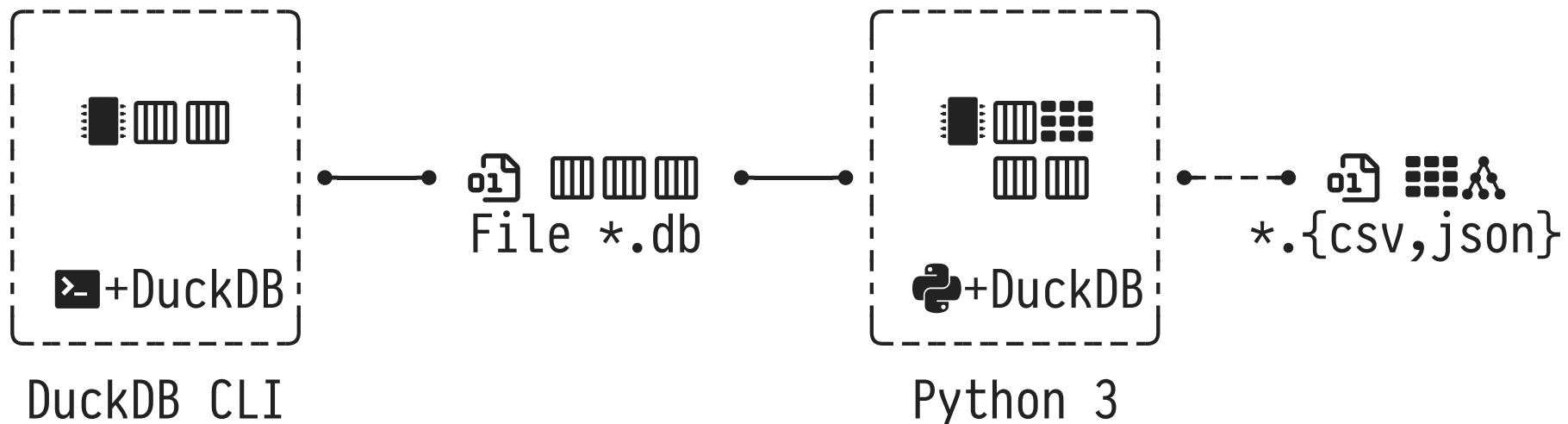
  - On-disk data organized in directories or in raw blocks.

- Client processes and DBMS process isolated, connected via network or inter-process communication .

  - Data needs to be de-/serialized after/before wire transfer.

- Data structures in clients are inaccessible by the DBMS.
- DBMS archetypes: PostgreSQL, MySQL, SQL Server™, Oracle®.

## DuckDB: A Tabular DBMS Inside Your Own Process



- DuckDB kernel and client share a **single process** [].
  - Python: `import duckdb`, C/C++: link with `libduckdb`.
- Table data resides in a **single database file** ① `*.db`, native DuckDB data formats ④ in file and in RAM are similar.
- DuckDB sees in-process client data and can **directly read/write client data structures** ③ using SQL (“**zero copy**”, replacement scans).
- DuckDB allocates sizable RAM buffers (but can use disk ② for temporary storage if required).

## 2 | This Course (*Dissecting the Duck's Innards*, short: *DiDi*)

---

- We will focus on DuckDB as a **tabular** SQL-based DBMS . There are other kinds of DBMSs (for graphs, key/value pairs, vectors, ...), but we will not discuss those here.
- We will get our hands dirty using **DuckDB**  and its extensive **SQL** dialect. Lots of SQL will be read and written.
- Whenever possible we try to observe DuckDB under load or use hooks to inspect its operation while SQL queries are processed.
- We thus assume basic familiarity with the tabular data model and SQL, e.g., as discussed in *Tabular Database Systems (TaDa)*.
- We will draw data and queries from a variety of sources and have **fun** along the way! 

# 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
- Bluesky : [@teggy.org](https://teggy.org)
- Best bet is to catch me on the DB group's Discord server 

## Administrivia

Weekday/Time	Slot	Room
Monday, 10:15–11:45 ⏱	Lecture	Lecture Hall TTR2 (MvL 6)
Thursday, 14:15–15:45 ⏱	Tutorial	Lecture Hall TTR2 (MvL 6)

- **⚠ No lecture/tutorials on**
  - Thu, October 16
  - (~~In this winter semester, we may miss Mon, Jan 19, 2026 and Thu, Jan 22, 2026—details will follow in due course.~~)
- **End-Term Exam (6 ECTS)**
  - Written exam on Mon, Feb 2, 2026, 10:00–12:00 (in Lecture Hall TTR2 in MvL6).
  - Score  $\geq \frac{2}{3}$  of the overall assignment points to be admitted to the exam.

## Slides and Further Lecture Material

---

These **slides** (PDF), **code fragments** (SQL, Python, C), and **sample data** will be uploaded to a GitHub  repository:

[github.com/DBatUTuebingen-Teaching/didi-ws2526](https://github.com/DBatUTuebingen-Teaching/didi-ws2526) 

- Slides point to relevant code files or extra material using tags like  #001:
  - Refers to a file named `001-*` on the GitHub repository (e.g., `001-sum-quantity.awk`).
- **NB.** Code and extra material provide essential content (e.g., details on SQL-based experiments or sample data).
  -  +  = : Only slides + code provide a complete picture.

## Weekly Assignments & Tutorial Sessions

---

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

Organized and run by **Denis Hirn**:

- Web: <https://db.cs.uni-tuebingen.de/team/members/denis-hirn/>
- E-Mail: db-lehre@cs.uni-tuebingen.de
- Office: WSI, Sand 13, Rooms B314
- Find us on the [Discord server](#) 

Assignments start once we have collected the first batch of interesting material, probably on Wednesday, October 22.

## Discord

---

During this winter semester, the **DB Discord** is *the* course hub:

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

- ⚠ **Registration** (do it!): use `/verify` with your e-mail address
- ❓ Questions and answers (do *not* post complete solutions)
- ⬇ Download additional code examples (e.g., SQL snippets)
- 💬 General discussion
- ⌚ Quick turnaround (responses often within minutes)

## DiDi's Course Homepage

---

<https://db.cs.uni-tuebingen.de/teaching/ws2526/didi/> 

- **Organizational matters**

Curriculum. General announcements regarding the lecture, exams, rooms, or dates. (Less important this semester.)

- **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. Rather, we build on

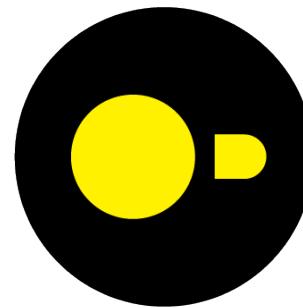
- a variety of scientific papers,
- textbook excerpts (few),
- the DuckDB  documentation at <https://duckdb.org/docs/>,
- Python/C/C++ code snippets (our own and from inside the )
- blog posts from a range of authors,
- SQL references/standards,
- experience, and best practices.

There is a plethora of books on tabular DBMSs (both usage and internals), sample SQL snippets (experiments, benchmarks, and idioms), or performance tweaks. If we will use such sources, we will provide pointers.

## Get Your Hands Dirty: Install DuckDB!

---

The tabular DBMS **DuckDB** will be the primary tool in this course:



# DuckDB

<https://duckdb.org>, version 1.4 (October 2025: 1.4.1)

- Implements an extensive SQL dialect, is highly performant, open to contributions, and generally awesome.
- Straightforward to install and use on macOS , Windows , Linux (x86 + ARM).

No DuckDB CLI ( SQL prompt/REPL) on iOS or Android.<sup>2</sup>

<sup>2</sup> Run the DuckDB CLI in the web browser: <https://shell.duckdb.org>. Suffices for quick SQL experiments.

### 3 | DuckDB? 🦆?

In case you were wondering:

**DuckDB** has been named after *Wilbur, the Duck*, which has been living as a pet with Hannes Mühleisen<sup>3</sup>—co-inventor of DuckDB with Mark Raasveldt—on Hannes' houseboat in Amsterdam.

Hannes (CEO) and Mark (CTO) run **DuckDB Labs**, a company that provides support and consultancy services around DuckDB. The labs are located in Amsterdam, The Netherlands.



Hannes and Wilbur (© Hannes Mühleisen)

<sup>3</sup> Hannes originally is from the Stuttgart area. Back then he used the license plate **SQL 1337**.