Programmierpraktikum: Datensysteme
01 Kickoff and Introduction

Prof. Dr. Matthias Boehm
Technische Universität Berlin
Berlin Institute for the Foundations of Learning and Data
Big Data Engineering (DAMS Lab)

Last update: Apr 13, 2025

Announcements / Org

- **#1 Hybrid & Video Recording**
  - Hybrid lectures (in-person, zoom) with optional attendance
    https://tu-berlin.zoom.us/j/9529634787?pwd=R1ZsN1M3SC9BOU1OcFdmem9zT202UT09
  - Zoom **video recordings**, links from website
    https://mboehm7.github.io/teaching/ss25_ppds/index.htm

- **#2 Course Registrations**
  - TU Berlin ISIS / Fak IV Meta registrations as of Apr 13 **~85**
  - Bachelor/Master ratio? CS/WINF/others ratio? **~32**

#3 Faculty IV - Team Awareness and Antidiscrimination

https://www.tu.berlin/eecs/awan

- **Goal**
  - **Low-barrier approachability** for spectrum of awareness and antidiscrimination issues
- **Team**
  - **Irene Hube-Achter** (MTSV)
  - **Matthias Boehm** (professors)
  - **Nadine Karsten** (scientific personnel)
  - **Tom Hersperger** (students)




- **Mission Statement**
  - Account for heterogeneity and complexity of modern societies at TU Berlin
  - All employees and students are committed to
    - **#1** Treat all persons with fairness and respect
    - **#2** Ensure a safe environment for all
    - **#3** Comply with our duty of care towards others
    - **#4** Actively support the implementation of the above guidelines and contribute

Contact: private email, eecs-TB-awareness@win.tu-berlin.de, or AwAn@dams.tu-berlin.de

Agenda

- Course Organization
- #1 Efficient Join Pipeline Executor (DAMS)
- #2 Duplicate Detection (D2IP)
- #3 Provenance Tracking in ML Pipelines (DEEM)
- Course Selection/Enrolment

Course Organization

Basic Course Organization

- **Language**
  - Lectures and slides: English (German if preferred)
  - Communication and presentations: English/German
  - Informal language (first name is fine)
  - Offline Q&A in forum, answered by teaching assistants

- **Course Format**
  - 6 ECTS (4 SWS) bachelor computer science / information systems
  - Every-other-week lectures (Mon 4.15pm sharp, including Q&A), attendance optional

- **Prerequisites**
  - Basic programming skills in languages such as C, C++, Java, Rust, Python, etc
  - Basic understanding of data management SQL / RA (or willingness to fill gaps)

Course Goals and Structure

- **Objectives**
  - **Apply basic programming skills** to more complex problem (in self-organized team work)
  - Technical focus on data management and data systems
  - Holistic programming projects: **prototyping, design, versioning, tests, experiments, benchmarks**

- **Grading: Pass/Fail**
  - **Project Implementation** (project source code) [**45%**]
  - **Component and Functional Tests** (test source code) [**10%**]
  - **Runtime Experiments** (achieve performance target) [**15%**]
  - **Documentation** (design document up to 5 pages / code documentation) [**15%**]
  - **Result Presentation** (10min talk) [**15%**]

- **Academic Honesty / No Plagiarism** (incl LLMs like ChatGPT)

Sub-Course Offerings

- **#1 Efficient Join Pipeline Executor**
  - Capacity: 28/60
  - Organized by DAMS group
  - Focus on query processing
  - Lectures every-other-week in **B 106**
- **#2 Duplicate Detection**
  - Capacity: 16/60
  - Organized by D2IP group
  - Focus on entity resolution
- **#2 Provenance Tracking in ML Pipelines**
  - Capacity: 16/60
  - Organized by DEEM group
  - Focus on ML pipelines

➔ **Admitted Students:**

- 32 + 28 on ISIS (incl duplicates)
- **Total registrations: up to 60**

  → 15 teams, 4 students each

#1 Efficient Join Pipeline Executor (DAMS)

About Me

- **Since 09/2022 TU Berlin**, Germany
  - University professor for Big Data Engineering (DAMS)

- **2018-2022 TU Graz**, Austria
  - BMK endowed chair for data management + research area manager
  - **Data management for data science** (DAMS), **SystemDS & DAPHNE**

- **2012-2018 IBM Research – Almaden**, CA, USA
  - Declarative large-scale machine learning
  - Optimizer and runtime of **Apache SystemML**

- **2007-2011 PhD TU Dresden**, Germany
  - Cost-based optimization of integration flows
  - Time series forecasting / in-memory indexing & query processing

DB group

History 1970/1980s
Relational Database Systems

SQL Standard
(SQL-**86**)

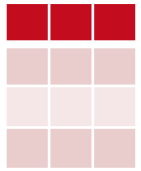
Oracle, IBM DB2,
Informix, Sybase
→ MS SQL

QUEL

SEQUEL

Ingres @ UC Berkeley
(Stonebraker et al.,
Turing Award '14)

System R @ IBM
Research – Almaden
(Jim Gray et al.,
Turing Award '98)



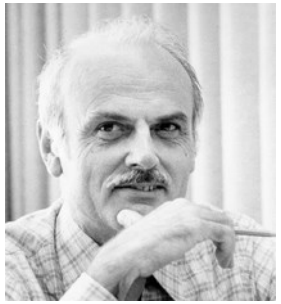
Tuple Calculus

Relational Algebra

Relational Model

Goal: Data Independence
(physical data independence)

- Ordering Dependence
- Indexing Dependence
- Access Path Dependence



Edgar F. "Ted" Codd @ IBM
Research (Turing Award '81)

[E. F. Codd: A Relational Model of Data for Large Shared Data Banks. Comm. ACM 13(6), **1970**]

Success of SQL / Relational Model

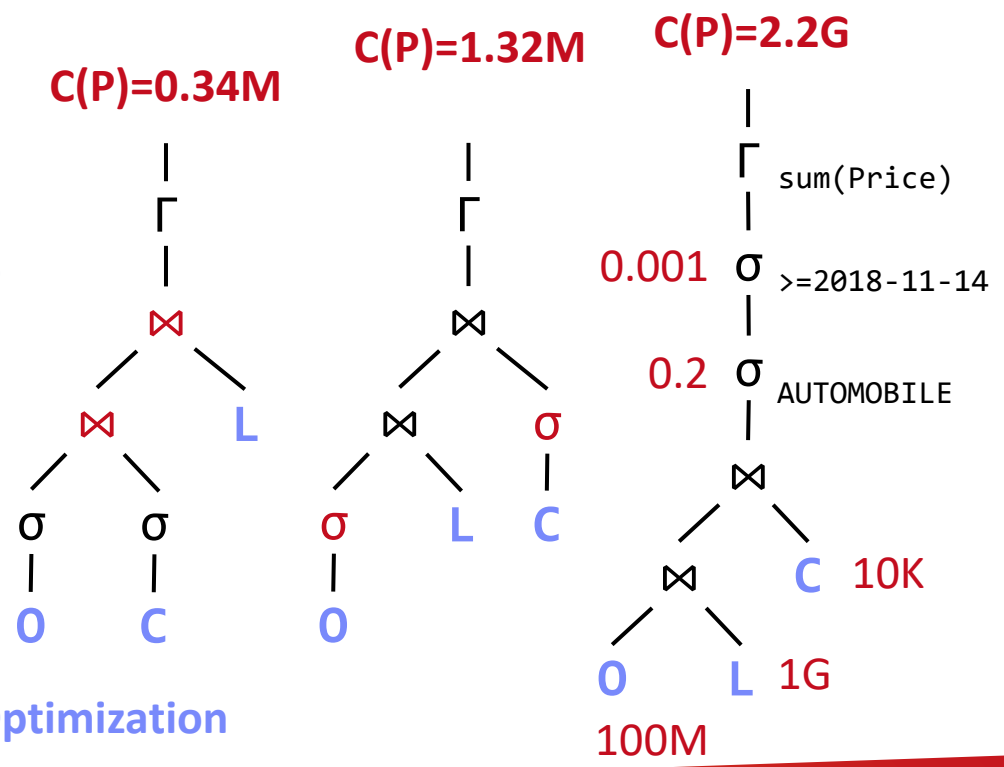
#1 Declarative:
what not how

#2 Flexibility:
closure property
→ composability

Query:

```
SELECT O_OID, sum(L_Price)
FROM Orders, Lineitem, Customer
WHERE O_OID = L_OID AND O_CID = C_CID
   AND O_Odate >= '2018-11-14'
   AND C_Msegment = 'AUTOMOBILE'
GROUP BY O_OID
```

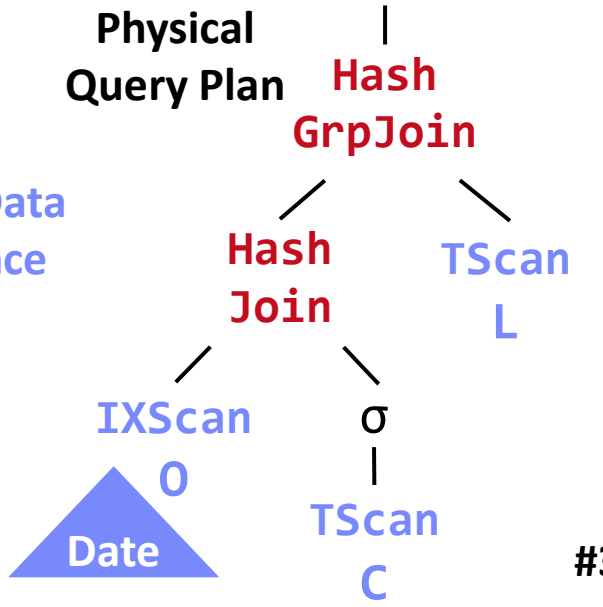
Logical Query Plans



#3 Automatic Optimization

#4 Physical Data Independence

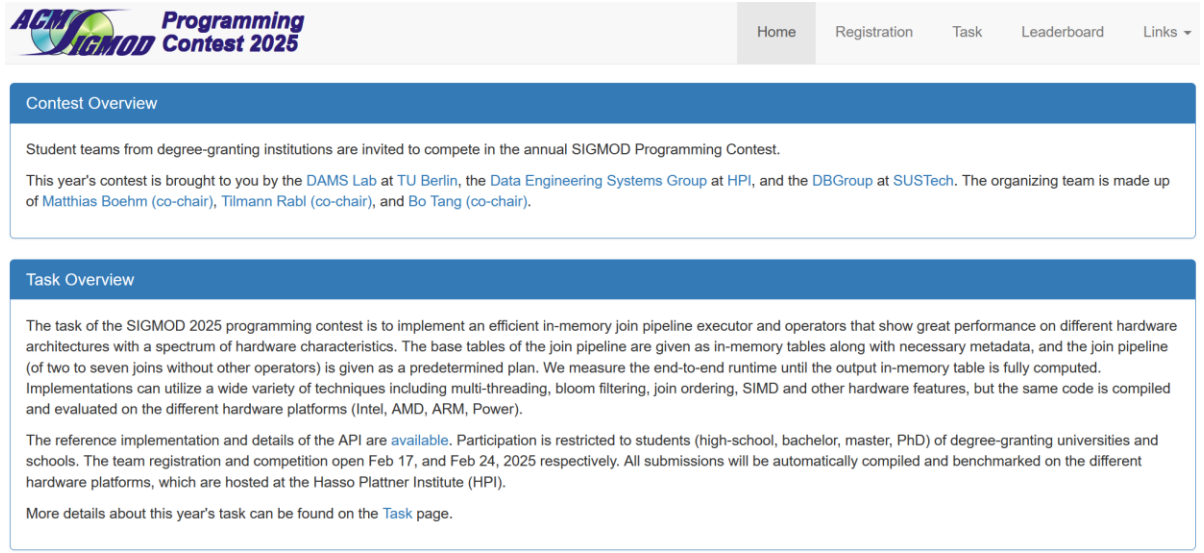
Physical Query Plan



Overview SIGMOD'25 Programming Contest

https://sigmod-contest-2025.github.io/index.html

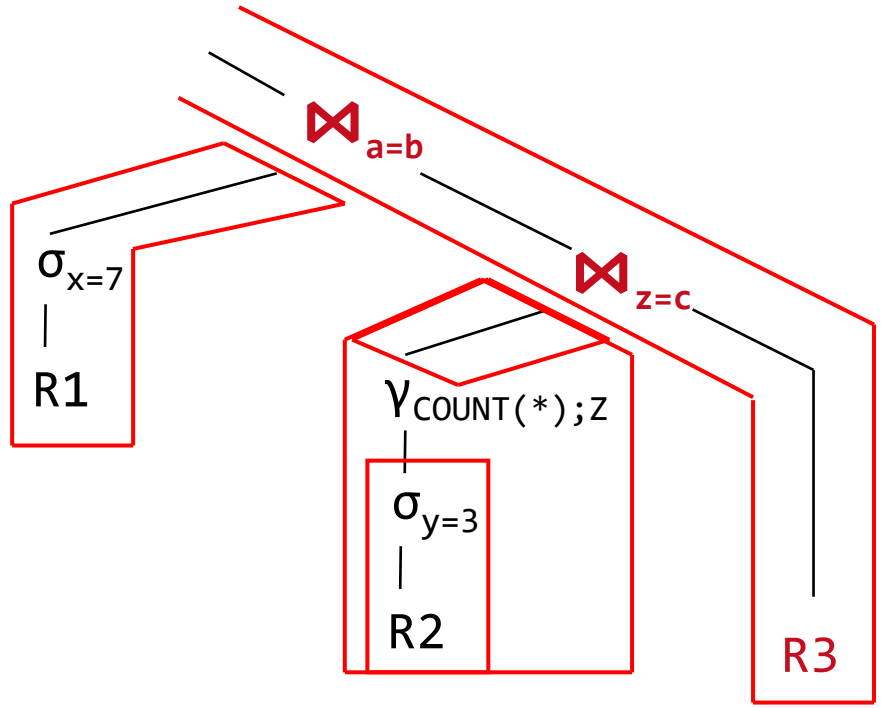
- **Task Overview**
  - Implement an efficient in-memory join pipeline executor
  - Aim performance on different hardware architectures
  - Base tables are given as columnar in-memory tables
  - Measures end-to-end runtime until outputs are computed



#	Team	Tests (Unit Tests / Query Plans)				Server Benchmarks (s)				Result (s)
		AMD #1	ARM #1	IBM #1	Intel #1	AMD #1	ARM #1	IBM #1	Intel #1	Geo. Mean
1	SortMergeJoins TU Munich (TUM)	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	0.38	0.3	0.75	0.6	**0.47** commit d014c4 (2025-04-01 05:34:31)
2	Kirara Beijing Institute of Technology and Xidian University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	1.37	1.07	3.23	2.5	**1.86** commit e36802 (2025-04-03 13:03:55)
3	Embryo Tsinghua University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	1.71	1.43	5.21	3.95	**2.67** commit babf2c (2025-04-01 07:12:30)
4	O.P.T. University of Athens	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	2.66	3.24	3.62	5.79	**3.67** commit 7d67d6 (2025-04-01 02:28:41)
5	Cross-Country Beijing Institute of Technology	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	3.21	3.07	6.27	4.64	**4.11** commit 66b66f (2025-03-31 22:00:47)
6	JobSeeking University of California, Riverside	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	2.91	2.38	8.9	5.61	**4.32** commit c21f27 (2025-04-01 04:25:56)
7	DB-Rush School of Data Science and Engineering, East China Normal University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	3.45	4.43	6.91	7.41	**5.29** commit 21f4fa (2025-03-31 13:27:07)
8	NanoPulse Nanjing University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	4.87	3.42	7.37	9.01	**5.77** commit b9bdd1 (2025-04-01 10:03:20)
9	HashStriker Zhejiang University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	6.3	4.17	21.51	9.4	**8.53** commit 69f0aa (2025-04-01 10:58:00)
10	DBRabbit East China Normal University	✓ / ✓	✓ / ✓	✓ / ✓	✓ / ✓	6.4	5.88	12.2	12.8	**8.75** commit 9fd90e (2025-04-01 02:42:46)

Overview Join Pipelines

```
SELECT *
   FROM R1, R3,
        (SELECT R2.z, count(*)
         FROM R2
         WHERE R2.y=3
         GROUP BY R2.z) R2
   WHERE R1.x=7
     AND R1.a=R3.b
     AND R2.z=R3.c
```



Application Programming Interface (API)

https://github.com/SIGMOD-25-Programming-Contest/base

```
namespace Contest {
  void* build_context();
  void  destroy_context(void*);

  ColumnarTable execute(
    const Plan& plan, void* context);
} // namespace Contest

./download_imdb.sh;
./build/build_database imdb.db;
./build/run plans.json;
```

```
struct Plan {
    std::vector<PlanNode>      nodes;
    std::vector<ColumnarTable> inputs;
    // std::vector<Table>         tables;
    size_t root;

    size_t new_join_node(bool                    build_left,
        size_t                                   left,
        size_t                                   right,
        size_t                                   left_attr,
        size_t                                   right_attr,
        std::vector<std::tuple<size_t, DataType>> output_attrs) {
        JoinNode join{
            .build_left = build_left,
            .left       = left,
            .right      = right,
            .left_attr  = left_attr,
            .right_attr = right_attr,
        };
        auto ret = nodes.size();
        nodes.emplace_back(join, std::move(output_attrs));
        return ret;
    }

    size_t new_scan_node(size_t                  base_table_id,
        std::vector<std::tuple<size_t, DataType>> output_attrs) {
        ScanNode scan{.base_table_id = base_table_id};
        auto     ret = nodes.size();
        nodes.emplace_back(scan, std::move(output_attrs));
        return ret;
    }

    size_t new_input(ColumnarTable input) {
        auto ret = inputs.size();
        inputs.emplace_back(std::move(input));
        return ret;
    }
};
```

Additional Course Logistics

- **Staff**
  - **Lecturer:** Prof. Dr. Matthias Boehm
  - **Teaching Assistant:** Christina Dionysio, and others if needed


Each teams gets a mentor
Q&A sessions on demand

- **Next Dates/Lectures**
  - **Apr 20:** Course Selection; team preferences, otherwise assignment
  - Apr 28: **Background Query Processing**
  - May 05: **Background Query Compilation and Parallelization**
  - May 26: **Background Query Optimization**
  - Jun 16: **Experiments and Reproducibility**
  - **Jul 08:** Project submissions (**performance target:** speedup > #pcores over reference)
  - **Jul 14:** Project presentations (10min per team, mandatory attendance)

- **Infrastructure**
  - Setup your own private Github/Gitlab repository

#2 Duplicate Detection (D2IP)

#3 Provenance Tracking in ML Pipelines (DEEM)

Course Selection/Enrolment

Select Your Course

- **#1 Efficient Join Pipeline Executor (DAMS)**
  - Capacity: 28/60
- **#2 Duplicate Detection (D2IP)**
  - Capacity: 16/60
- **#3 Provenance Tracking in ML Pipeline (DEEM)**
  - Capacity: 16/60

Thanks

Course Selection/Enrolment
by Apr 20 EOD

https://tinyurl.com/4kyhwz37