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, 2026

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/ss26_ppds/index.htm

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

26 meta
~65 all

#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)
  - **Ceenu George** (professors)
  - **Nadine Karsten** (scientific staff)
  - **Tom Hersperger** (students)





- **Mission Statement**
  - Account for heterogeneity and complexity of modern societies at TU Berlin
  - **#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)
- #4 Fairness Auditing for 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: 16/48
  - Organized by **DAMS** group
  - Focus on query processing
- **#2 Duplicate Detection**
  - Capacity: 16/48
  - Organized by **D2IP** group
  - Focus on entity resolution
- **#3 Provenance Tracking in ML Pipelines**
  #4 Fairness Auditing for ML Pipelines
  - Capacity: 16/48
  - Organized by **DEEM** group
  - Focus on ML pipelines

➔ **Admitted Students:**
- 67 on ISIS (incl duplicates)
- **Total registrations: up to 60**
  → 15 teams, 4 students each

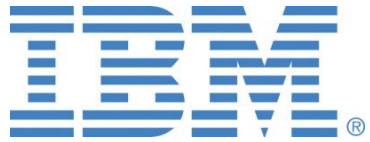
#1 Efficient Join Pipeline Executor (DAMS)

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**)

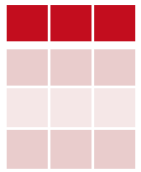
Oracle, IBM DB2,
Informix, Sybase
→ MS SQL

QUEL

SEQUEL

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

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



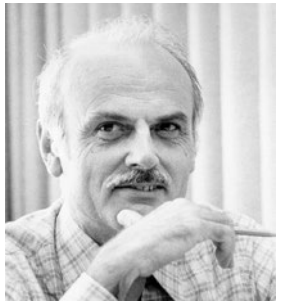
Tuple Calculus

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

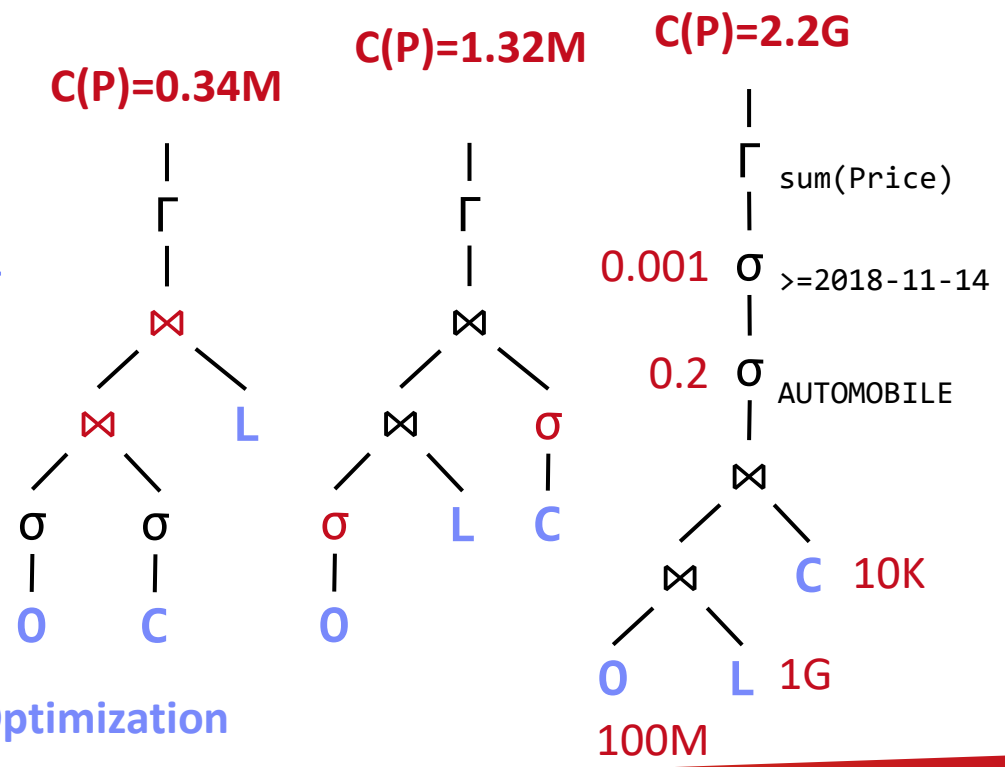
#1 Declarative:
what not how

#2 Flexibility:
closure property
→ composability

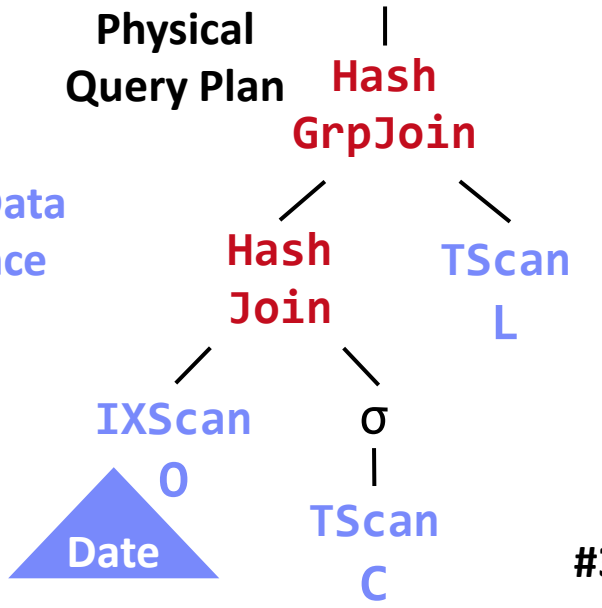
Query:

```
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



Physical Query Plan



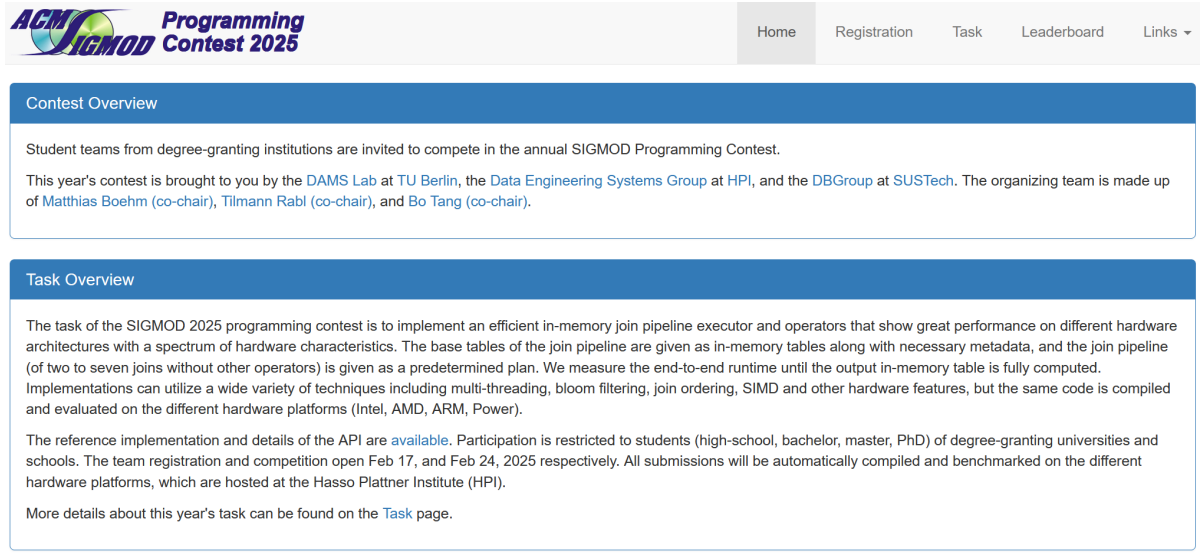
#3 Automatic Optimization

#4 Physical Data Independence

Overview SIGMOD'25 Programming Contest

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

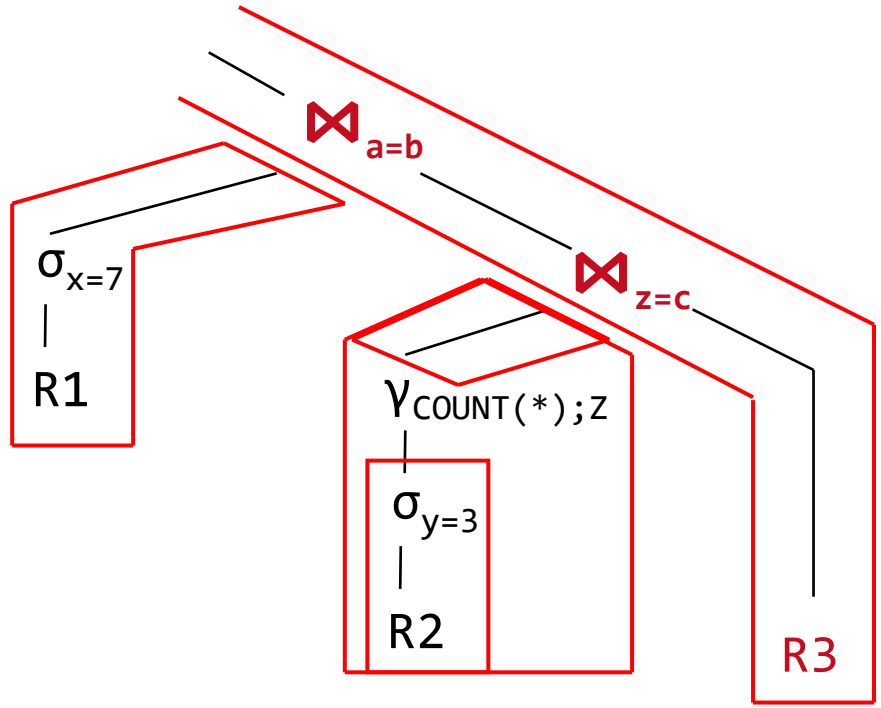
- **Task Overview**
  - Implement efficient in-memory join pipeline executor
  - Aim performance on different hardware architectures
  - Base tables are given as columnar in-memory tables
  - Measures end-to-end runtime until outputs computed



#	Team	Server Benchmarks (s)								Result
		AMD		ARM		IBM		Intel		Geo. Mean
		#1	#2	#1	#2	#1	#2	#1	#2	
1	SortMergeJoins TU Munich (TUM)	2.57	3.22	2.14	1.13	5.49	3.31	4.13	1.74	**2.67**
2	Kirara Beijing Institute of Technology and Xidian University	6.86	8.79	6.95	2.62	10.42	7.33	14.5	4.88	**7.03**
3	Embryo Tsinghua University	10.32	14.51	7.55	3.25	17.69	13.7	24.53	8.44	**10.8**
4	O.P.T. University of Athens	12.26	12.21	14.23	4.39	45.98	11.74	26.4	9.71	**13.77**
5	JobSeeking University of California, Riverside	20.37	33.06	16.81	5.15	71.26	31.88	37.19	16.74	**23.15**
6	DB-Rush School of Data Science and Engineering, East China Normal University	21.53	22.66	22.42	9.75	67.53	30.17	42.19	16.25	**24.92**
7	Cross-Country Beijing Institute of Technology	24.7	41.83	24.1	10.31	101.23	35.52	37.97	18.39	**29.93**
8	DBRabbit East China Normal University	30.1	33.17	23.47	9.66	176.14	26.07	60.83	19.76	**32.51**
9	ACID Trip LMU, TU Ilmenau, TUM	35.91	34.76	30.15	12.56	1468.73	29.52	60.07	27.76	**49.17**
10	Shiyu Zhejiang University	56.56	67.18	42.36	14.21	182.38	50.73	114.89	48.27	**57.37**
11	Optimus Join TU Dresden	127.56	115.05	208.28	84.04	556.73	97.78	201.37	113.74	**154.24**

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:** Philipp Ortner, and others if needed

Each teams gets a mentor
Q&A sessions on demand

- **Next Dates/Lectures**
  - **Apr 19:** Course Selection; team preferences, otherwise assignment
  - Apr 27: **Background Query Processing**
  - May 11: **Background Query Compilation and Parallelization**
  - Jun 08: **Background Query Optimization**
  - Jun 22: **Experiments and Reproducibility**
  - **Jul 01:** Project submissions (**performance target:** speedup > #pcores over reference)
  - **Jul 06:** 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)

#4 Fairness Auditing for ML Pipelines (DEEM)

Course Selection/Enrolment

Select Your Course

- **#1 Efficient Join Pipeline Executor (DAMS)**
  - Capacity: 16/48
- **#2 Duplicate Detection (D2IP)**
  - Capacity: 16/48
- **#3 Provenance Tracking in ML Pipeline (DEEM)**
  #4 Fairness Auditing for ML Pipelines (DEEM)
  - Capacity: 16/48

Thanks

Course Selection/Enrolment
by Apr 19 EOD

https://forms.gle/i6xFTMxHWSJabYHR8