Data Management

01 Introduction and Overview

Matthias Boehm
Graz University of Technology, Austria
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMVIT endowed chair for Data Management

Last update: Mar 02, 2020
Announcements/Org

- **#1 Video Recording**
  - Link in TeachCenter & TUbe (lectures will be public)

- **#2 Course Registration** (as of Mar 02)
  - Data Management VO: 457
  - Data Management KU: 451 😊
  - Databases VU: 89

- **#3 Bac::Mas Thesis Fair** (Mar 5, 10am-1pm, INF 25d Foyer)
  - STV-organized fair for open bachelor/master topics at institutes

- **#4 CS Talks x7** (Mar 10, 5pm, Aula Alte Technik)
  - Claudia Müller-Birn (Freie Universität of Berlin)
  - Title: Collaboration is Key – Human-Centered Design of Computational Systems
Agenda

- Data Management Group
- Course Motivation, Goals, and Outline
- Course Organization and Logistics
- History of Data Management
Data Management Group
About Me

- **09/2018 TU Graz, Austria**
  - BMVIT endowed chair for data management
  - **Data management for data science**
    (ML systems internals, end-to-end data science lifecycle)

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

- **2011 PhD TU Dresden, Germany**
  - Cost-based optimization of integration flows
  - Systems support for time series forecasting
  - In-memory indexing and query processing
Data Management Courses

- **Architecture of Database Systems** (ADBS, WS)
  - Master
  - DB system internals + prog. project

- **Architecture of ML Systems** (AMLS, SS)
  - Bachelor
  - ML system internals + prog. project in SystemDS:
    [github.com/tugraz-isds/systemds]

- **Data Integration and Large-Scale Analysis** (DIA, WS)
  - Distributed Data Management (usage and internals)

- **Data Management / Databases** (DM, SS+WS)
  - Data management from user/application perspective
Course Motivation, Goals, and Outline

Database Systems and Modern Data Management
Definition and Impact

- **Def: Database System**
  - Overall system of DBMS + DBs
  - DBMS: Database Management System (SW to handle DBs)
  - DBs: Database (data/metadata collection of conceptual mini-world)
  - Note: DB also a short for DBS/DBMS

- **Importance in Practice**
  - Market Volume: **10-100B $US**
  - Foundation of many applications in various domains

**“Relational databases are the foundation of western civilization”**
Motivation Database Systems

- **Application development and maintenance costs**
  - Declarative queries (what not how) and data independence
  - Efficient, correct, and independent data organization, size, access
- **Multi-user operations and access control**
  - Synchronization of concurrent user queries and updates
  - Enforce access control (e.g., permissions on tables, views)
- **Consistency and data integrity**
  - Eliminates redundancy and thus, enforces consistency
  - Enforces integrity constraints (e.g., semantic rules)
- **Logging and Recovery**
  - Recovery of consistent state after HW or SW failure
- **Performance and Scalability**
  - High performance for large datasets or high transaction throughput
  - Scale to large datasets with low memory requirements
Motivation and Goals

Goals

- **Course Goals**
  - **A:** Understanding of database systems (from user perspective)
  - **B:** Understanding of modern data management (from user perspective)

- **Meta Goals**
  - Understand, use, debug, and evaluate data management systems
  - Awareness of system alternatives and their tradeoffs
  - Fundamental concepts as basis for advanced courses and other areas
Part A: Database System Fundamentals

- 01 Introduction and Overview [Mar 02]
- 02 Conceptual Architecture and Design [Mar 09]
- 03 Data Models and Normalization [Mar 16]
- 04 Relational Algebra and Tuple Calculus [Mar 23]
- 05 Query Languages (SQL, XML, JSON) [Mar 30]
- 06 APIs (ODBC, JDBC, OR frameworks) [Apr 20]
- 07 Physical Design and Tuning [Apr 27]
- 08 Query Processing [May 04]
- 09 Transaction Processing and Concurrency [May 11]

Exercise 1: Data Modeling [Mar 31]
Exercise 2: Queries [Apr 28]
Exercise 3: Tuning [May 19]
Part B: Modern Data Management

- 10 NoSQL (key-value, document, graph, time series) [May 18]
- 11 Distributed file systems and object storage [May 25]
- 12 Data-parallel computation (MapReduce, Spark) [May 25]
- 13 Data stream processing systems [Jun 08]
- 14 Q&A and exam preparation [Jun 15]
- Final written exam [TBD; e.g., Jun 22, Jun 29]
Course Organization
Basic Course Organization

- **Staff**
  - **Lecturer:** Univ.-Prof. Dr.-Ing. Matthias Boehm, ISDS
  - **Assistant Lecturer:** M.Tech. Arnab Phani, ISDS
  - **Teaching Assistants:**
    - Alina Herderich, Dardan Dermaku, Olga Ovcharenko, Oliver Nikolic, Melanie Willfurth, Paul Mirtl

- **Language**
  - Lectures and slides: **English**
  - Communication and examination: **English/German**

- **Course Format**
  - DM VO + KU 2/1 (**3+1 ECTS**), DB VU 1/1 (**3(2) ECTS**)
  - Weekly lectures (**start 4.10pm**, including **Q&A**), attendance optional
  - 4/3 exercises (introduced in lecture) as individual assignments
Course Logistics

- **Communication**
  - *Informal language* (first name is fine)
  - Please, *immediate feedback* (unclear content, missing background)
  - *Newsgroup:* news://news.tugraz.at/tu-graz.lv.dbase (email for private issues)
  - *Office hours:* Mo 1pm, or after lecture

- **Website**
  - [https://mboehm7.github.io/teaching/ss20_dbs/index.htm](https://mboehm7.github.io/teaching/ss20_dbs/index.htm)
  - All course material (lecture slides, exercises) and dates

- **Exam**
  - *Completed mandatory exercises* (Mar 31, Apr 28, May 19, [Jun 16])
  - *Final written exam* (TBD, doodle voting)
  - *DB Grading* (30% exercises, 70% final)
Course Logistics, cont.

- **Exercises**
  - Written and programming assignments, submitted through TeachCenter
  - Assignments have 25 points + ? bonus points (capped for DB at 80/75)
  - Assignment completed if >50% points in total (but all submitted)
  - Deadlines are important (at most 7 late days in total)
  - Individual assignments (academic honesty / no plagiarism)

- **SW Tools and Languages**
  - Open Source PostgreSQL DBMS (setup on your own)
  - Distributed FS/object storage and Apache Spark for distributed computation
  - Languages for local/distributed programs (of your choice):
    e.g., Python, Java, Scala, C, C++, C#, Rust, Go, etc.
Exercises: DBLP Publications

- **Dataset**
  - CC0-licensed, derived (extracted, cleaned) from [DBLP](https://dblp.org) (Feb 1, 2020) for publication year ≥ 2011
  - Clone or download your copy from [https://github.com/tugraz-isds/datasets.git](https://github.com/tugraz-isds/datasets.git)

- **Exercises**
  - 01 Data modeling (relational schema)
  - 02 Data ingestion and SQL query processing
  - 03 Tuning, query processing, and transaction processing
  - 04 Large-scale data analysis (distributed data ingestions and query processing)
Exercises: DBLP Publications, cont.

- **DBLP Statistics**
  - 4,782,347 pubs
  - 2,438,282 authors
  - 75,435 PhD theses
  - 43,218 conferences

- **Our Exercise Dataset**
  - Subset w/ year ≥ 2011 and selected features
  - 2,607,587 pubs
  - 1,716,612 authors
  - 32,534 PhD theses
  - 22,730 conferences

438 MB in uncompressed text files (CSV)

[Credit: https://dblp.org/statistics/recordsindblp.html]
Course Organization

19

Literature

- **Not needed for lectures / exercises** (course is self-contained), but second perspective on covered topics of first part


History of Data Management
History 1960/70s (pre-relational)

- **Hierarchical Model**
  - Tree of records
  - E.g., IBM Information Management System (IMS) – IMS 15 (Oct 2017)

- **Network Model**
  - CODASYL (COBOL, DB interfaces)
  - Graph of records
  - Charles Bachman (Turing Award ’73)
  - E.g., Integrated Data Store (IDS)

- **Pros and Cons** (see NoSQL Doc-Stores)
  - Performance by *directly traversing static links*
  - *Duplicates* → inconsistencies on updates, *data dependence*
History of Data Management

History 1970/80s
(relational)

SQL Standard
(SQL-86)

Oracle, IBM DB2,
Informix, Sybase
→ MS SQL

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

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

Tuple Calculus

SEQUEL

SQL Standard

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

Relational Algebra

Goal: Data Independence
(physical data independence)
• Ordering Dependence
• Indexing Dependence
• Access Path Depend.

Edgar F. “Ted” Codd @ IBM
Research (Turing Award ‘81)

[Image of Edgar F. Codd]

Oracle, IBM DB2,
Informix, Sybase
→ MS SQL

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

relational Model

QUEL

RELATIONAL ALGEBRA

E. F. Codd @ IBM
Research (Turing Award ‘81)
**Success of SQL / Relational Model**

**Query:**

```sql
SELECT O_OID, sum(O_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
```

**#1 Declarative:**

what not how

**#2 Flexibility:**

closure property

→ composability

**Logical Query Plans**

**#3 Automatic Optimization**

**#4 Physical Data Independence**

**Physical Query Plan**
Excurus: PostgreSQL

- History of PostgreSQL (used in the exercises)
  - Postgres is the successor project of commercialized Ingres
  - Focus on abstract data types, commercialized as Illustra
  - Prototype w/ SQL open sourced as Postgres95 → PostgreSQL
  - Heavily used as basis for research projects / startups

- Recommended Reading
  - Video: http://www.youtube.com/watch?v=sEPTZVGk3WY
History 1980/90/2000s

- **Enterprise DBMS**
  - Heavy investment in research and development ➞ adoption
  - Oracle, IBM DB2, Informix, Sybase, MS SQL, PostgreSQL, MySQL
  - Other technologies: OODBMS, Multimedia, Spatiotemporal, Web, XML

- **Information/Data Warehousing (DWH)**
  - Workload separation into OLTP and OLAP
  - Classical DWH architecture: operational, staging, DWH, data marts + mining
  - ETL Process (Extract, Transform, Load)

- **Different Personas**
  - Domain Experts (e.g., BI Tools, SAP R/3)
  - DB Application Developers (e.g., ABAP)
  - DB Developers and DB Admins
History of Data Management

History 2000s / Early 2010s

- **Specialized Systems**
  - Column stores + compression for OLAP
  - Main memory systems for OLTP and OLAP
  - Data streaming, scientific and graph databases
  - Information extraction / retrieval, and XML

- **Other Research Trends**
  - Approximate QP / Adaptive QP / tuning tools
  - Large-scale data management (DFS, MR) / cloud computing

- **Toward Flexible, Large-Scale Data Management** (DWH ... a bygone era)
  - MAD Skills (magnetic, agile, deep), MADlib
  - Integration of R, Python in data analysis
  - Open data and its integration
  - Query processing over raw data files

---


History 2010s – Present

- **Two Key Drivers of DB Research**
  - *New analysis workloads* (NLP, key/value, RDF/graphs, documents, time series, ML) and applications
  - *New HW/infrastructure* (multi-/many-core, cloud, scale-up/scale-out, NUMA/HBM, RDMA, SSD/NVM, FPGA/GPU/ASIC)

- **Excursus: A retrospective view of specialized systems**
  - **Goal #1**: Avoid boundary crossing → *General-purpose*
  - **Goal #2**: New workload + Performance → *Specialized systems*
  - Some Examples

### New Workloads vs. New HW

- **New Workloads**
  - RDBMS
  - OODBMS
  - XML
  - Docs
  - OLAP
  - MR/Spark
  - RDF/graphs
  - NLP
  - Time
  - ML Sys

- **New HW**
  - OR
  - Hybrid
  - JSON Datatype
  - HTAP
  - SQL on Hadoop
  - In-DB alternatives
  - RDBMS

**History of Data Management**

Matthias Boehm, Graz University of Technology, SS 2020
History 2010s – Present (2)

- **Motivation NoSQL Systems**
  - Flexible schema (no upfront costs), scalability, or specific data types
  - Relaxed ACID (atomicity, consistency, isolation, durability) requirements
    ➔ BASE (basically available, soft state, eventual consistency)

- **Example NoSQL Systems** (local and distributed):
  - Key/Value-Stores: simple put/get/delete, massive scalability
  - Document-Stores: store nested documents (tree)
  - RDF Stores: store subject-predicate-object triples
  - Graph DBs: store nodes/edges/attributes, vertex-centric
  - Time Series DBs: store sequences of observations
History 2010s – Present (3)

- **Motivation Large-Scale Data Management**
  - **Massive scalability** (data/compute) on demand, **fault tolerance**, flexibility
  - Example Facebook 2014: 300PB DWH, 600TB daily ingest
  - Cost-effective commodity hardware
  - Error rate increases with increasing scale

- **Examples Large-Scale Data Management**
  - Distributed file systems w/ replication (e.g., GPFS, HDFS)
  - Cloud object storage (e.g., Amazon s3, OpenStack Swift)
  - Data-parallel data analysis with Spark/Flink, incl streaming
  - Automatic cloud resource elasticity (pay as you go)
Summary and Q&A

- **#1 Database Systems**
  - Mature and established technology ➔ broadly applicable & eco system
  - **General concepts**: abstraction, data modeling, query optimization & processing, transaction processing and recovery, physical design and tuning

- **#2 Modern Data Management**
  - Multiple specialized systems for specific scale / data types
  - General trend toward less upfront cost, flexibility, and higher scalability

- **Variety of data management tools ➔ Course meta goals**
  - **Understand, use, debug**, and evaluate data management systems
  - **Fundamental concepts** as basis for advanced courses and other areas

- **Upcoming**
  - **02 Conceptual Architecture and Design** [Mar 09] (ER Diagrams)
  - **03 Data Models and Normalization** [Mar 16] (ERD -> Relational Model)