Database Systems
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 04, 2019

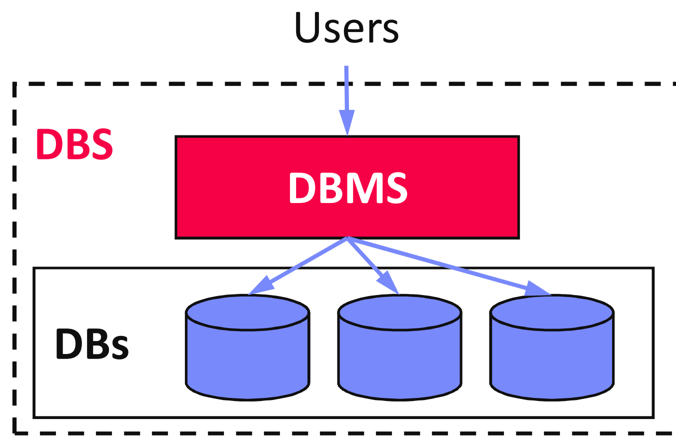
SCIENCE
PASSION
TECHNOLOGY

Agenda

- **Motivation and Goals**
- **Data Management Group**
- **Course Organization**
- **Course Outline**
- **History of Data Management**
- **Announcements**

Definition and Impact

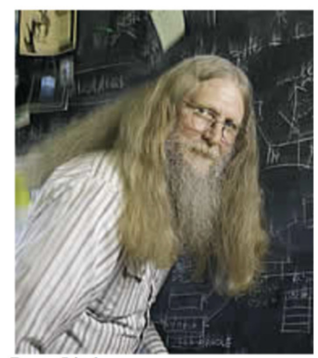
- **Def: Database System**
  - Overall system of DBMS + DBs
  - DBMS: Database Management System (SW to handle DBs)
  - DBs: Database (data/metadata of conceptual mini-world)
  - Note: DB also a short for DBS/DBMS



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

[M. Winslett: Bruce Lindsay speaks out: […]. SIGMOD Record 34(2), **2005**]

"Relational databases are the foundation of western civilization"



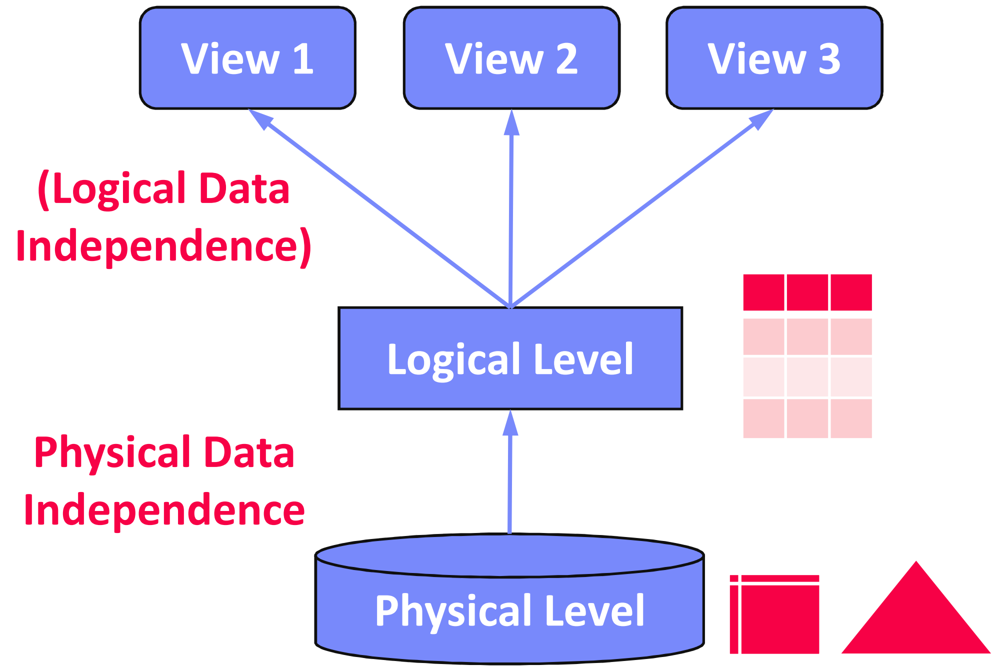
Bruce Lindsay

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

Data Independence

- **Three Layer ANSI-SPARC Architecture**
  - External schemas (external level)
  - Conceptual schema (logical level)
  - Internal schema (physical level)



- **Types of Data Independence**
  - Logical data independence (external views and applications independent of logical data model)
  - Physical data independence (logical data model independent of underlying data organization)

Goals

- **Course Goals**
  - Understanding of database systems **from user perspective** (conceptual design, relational model, physical design and tuning, query and transaction execution, APIs)
  - Understanding of modern means of data management **from user perspective** (NoSQL, distributed file systems, data-parallel frameworks, data streaming)

DB 1 (3 ECTS) — first item

DB (4 ECTS) — both items

- **Meta Goals**
  - Understand, use, debug, and evaluate data management tools / systems
  - Awareness of system alternatives and their tradeoffs
  - Fundamental concepts as basis for advanced courses and other areas

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)

https://github.com/tugraz-isds/systemds

- **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

DB group

Data Management Courses

- **SS: Databases / Databases 1 (DM)**
  - Data management from user/application perspective
  - VU 1.5/1.5 (4 ECTS), and VU 1/1 (3 ECTS)
- **SS: Architecture of ML Systems (AMLS)**
  - Internals of machine learning systems
  - VU 2/1 (5 ECTS), master, github.com/tugraz-isds/systemds

- **WS: Data Integration and Large-Scale Analysis (DIA)**
  - Distributed data and information systems
  - VU 2/1 (5 ECTS), bachelor/master
- **WS: Architecture of Database Systems (ADBS)**
  - Internals of database management systems
  - VU 2/1 (5 ECTS), master

Course Organization

Basic Course Organization

- **Staff**
  - Lecturer: Univ.-Prof. Dr.-Ing. Matthias Boehm, ISDS
  - Teaching Assistants: Dardan Dermaku, Ermal Gashi
- **Language**
  - Lectures and slides: **English**
  - Communication and examination: **English**/**German**
- **Course Format**
  - DB VU 1.5/1.5 (**4 ECTS**), DB1 VU 1/1 (**2/3 ECTS**)
  - **Weekly lectures** (**start 4.10pm**, including **Q&A**), **attendance optional**
  - **4/3 exercises** (introduced in lecture)
  - **Recommended papers** for additional reading on your own

Course Logistics

- **Exam**
  - **Completed mandatory exercises** (Apr 02, Apr 30, May 28, Jun 18)
  - **Final written exam** (Jun 24, 4.15pm-5.45pm – time/rooms TBD)
  - **Grading** (30% exercises, 70% final)

- **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: by appointment or after lecture

- **Website**
  - https://mboehm7.github.io/teaching/ss19_dbs/index.htm
  - All course material (lecture slides, exercises) and dates

Course Logistics (2)

- **Exercises**
  - Written and programming assignments, submitted through **TeachCenter**
    - DB: https://tc.tugraz.at/main/course/view.php?id=1821
    - DB1: https://tc.tugraz.at/main/course/view.php?id=1822
  - **Weekly office hours**, in addition to newsgroup
  - Assignment **completed if >50% points**
  - 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: e.g., C, C++, **Java**, **Scala** or **Python**

Literature

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

- Raghu Ramakrishnan, Johannes Gehrke: Database Management Systems (3. ed.). McGraw-Hill 2003, ISBN 978-0-07-115110-8, pp. I-XXXII, 1-1065
- Jeffrey D. Ullman, Jennifer Widom: A first course in database systems (2. ed.). Prentice Hall 2002, ISBN 978-0-13-035300-9, pp. I-XVI, 1-511
- Gerhard Weikum, Gottfried Vossen: Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery. Morgan Kaufmann 2002, ISBN 1-55860-508-8
- Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, 3rd Edition. Addison-Wesley-Longman 2000, ISBN 978-0-8053-1755-8, pp. I-XXVII, 1-955

Course Outline

Part A: Database System Fundamentals

- **01 Introduction and Overview** [Mar 04]
- **02 Conceptual Architecture and Design** [Mar 11]
- **03 Data Models and Normalization** [Mar 18]

→ Exercise 1: Data Modeling **[Apr 02]**

- **04 Relational Algebra and Tuple Calculus** [Mar 25]
- **05 Query Languages (SQL)** [Apr 01]
- **06 APIs (ODBC, JDBC, OR frameworks)** [Apr 08]

→ Exercise 2: Queries **[Apr 30]**

- **07 Physical Design and Tuning** [Apr 29]
- **08 Query Processing** [May 06]
- **09 Transaction Processing and Concurrency** [May 13]

→ Exercise 3: Tuning **[May 28]**

Part B: Modern Data Management

- **10 NoSQL (key-value, document, graph)** [May 20]
- **11 Distributed file systems and object storage** [May 27]
- **12 Data-parallel computation (MapReduce, Spark)** [Jun 03]
- **13 Data stream processing systems** [Jun 17]

→ Exercise 4: Spark **[Jun 18]**

- **Final Exam** **[Jun 24]** (room(s) and date(s) TBD)

Exercises: Soccer World Cup 1954-2014

- **Dataset**
  - Public-domain, derived (parsed, cleaned) from **Openfootball Worldcup Dataset**
  - Clone or download your copy from 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)

1954_2014_Squads.csv: The Squads file contains tl structure and examples look as follows.

```
#Year, Host_Country, Country, Jersey_Number,
1998,France,Austria,14,FW,Hannes Reinmayr,Stu
2014,Brazil,Germany,1,GK,Manuel Neuer,Bayern
2014,Brazil,Germany,11,FW,Miroslav Klose,Lazi
```

1954_2014_Matches.csv: The Matches file contains and examples look as follows.

```
#Year, Host_Country, Match_ID, Type, Date, Lo
2006,Germany,572,Group A,Wed Jun/14,Signal Id
2010,South Africa,684,Round of 16,Sun Jun/27
2014,Brazil,761,Final,Sun Jul/13 16:00,Estádi
```

1954_2014_Goals.csv: The Goals file contains the g time of the game. It's detailed structure and exam

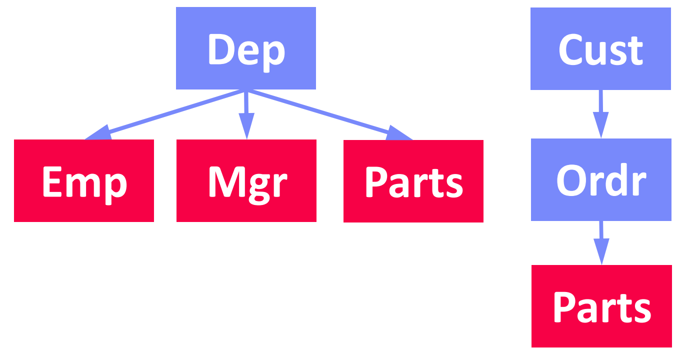
```
#Year, Host_Country, Match_ID, Team, Player,
2014,Brazil,760,Netherlands,Daley Blind,17
2014,Brazil,760,Netherlands,Georginio Wijnald
2014,Brazil,761,Germany,Mario Götze,113
```

History of Data Management

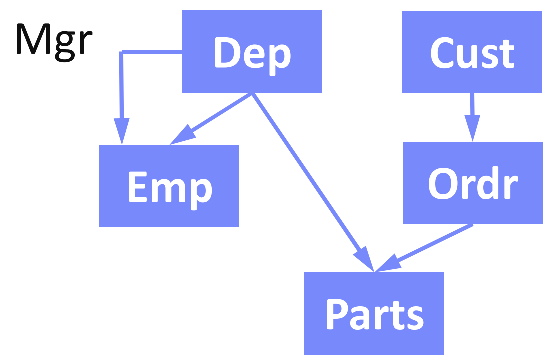
History 1960/70s (pre-relational)

CODASYL … Conference on Data Systems Languages

- **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 1970/80s (relational)

SQL Standard
(SQL-**86**)

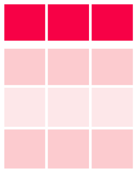
Oracle, IBM DB2, Informix, Sybase
→ MS SQL

SEQUEL

QUEL

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

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



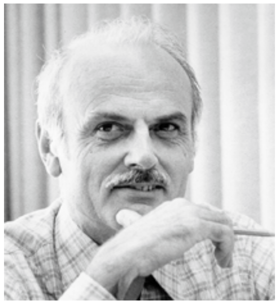
Tuple Calculus

Relational Algebra

Relational Model

Goal: Data Independence
(physical data independence)

- Ordering Dependence
- Indexing Dependence
- Access Path Depend.



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**]

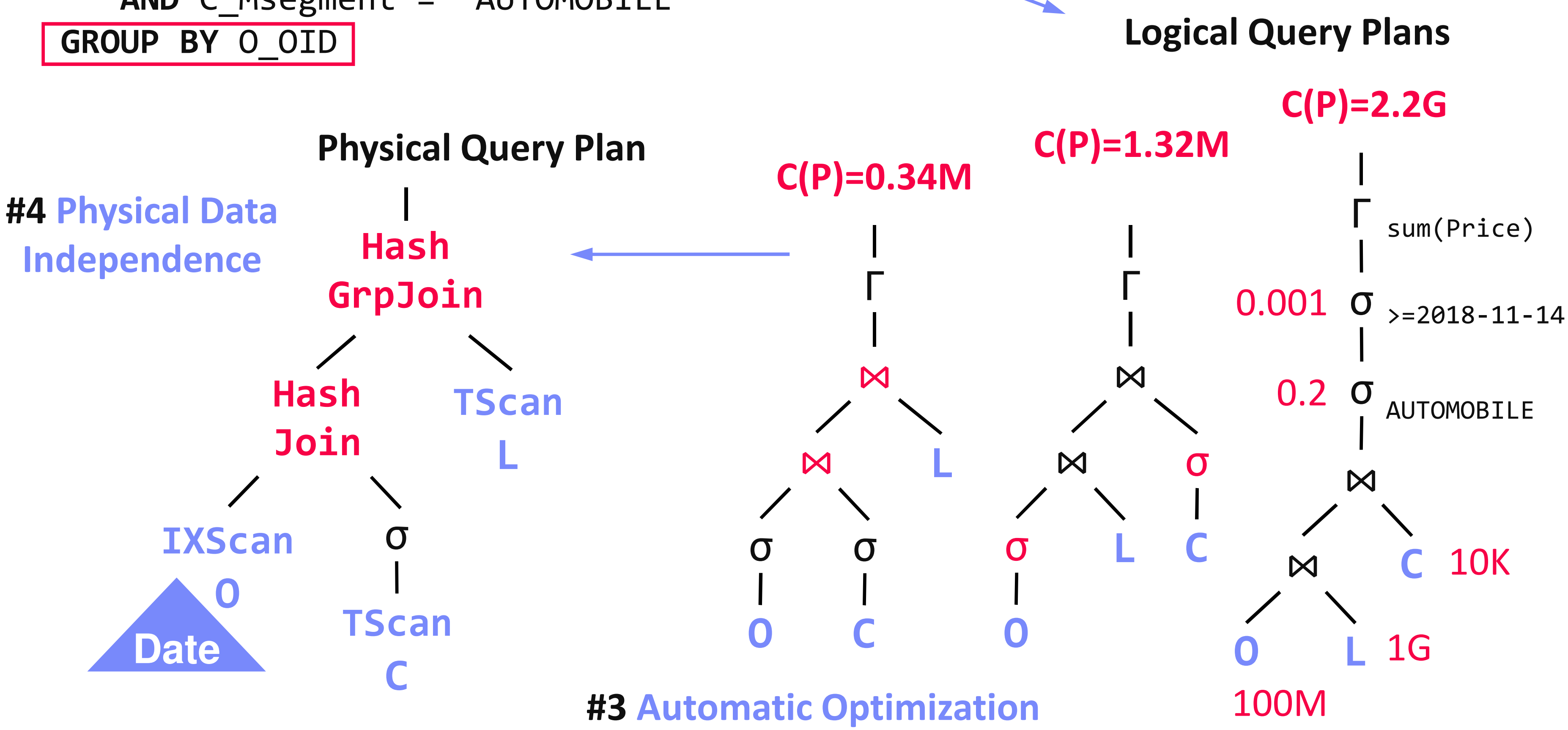
Success of SQL / Relational Model

Query:

```
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



Excursus: PostgreSQL

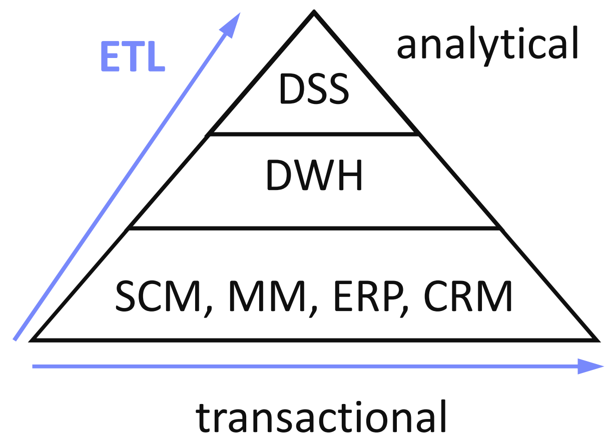
- **History of PostgreSQL** (used in the exercises)
  - Postgres is the successor project of commercialized Ingres
  - Focus on abstract data types, commercialized as Illustra
  - Prototype w/ SQL open sourced as **Postgres95** → **PostgreSQL**
  - Heavily used as **basis for research projects / startups**

- **Recommended Reading**
  - Michael Stonebraker: The land sharks are on the squawk box. Commun. ACM 59(2): 74-83 (2016), Turing Award Lecture, https://dl.acm.org/citation.cfm?doid=2886013.2869958
  - **Video:** http://www.youtube.com/watch?v=sEPTZVGk3WY
  - **Slides:** http://vldb.org/2015/wp-content/uploads/2015/09/stonebraker.pdf

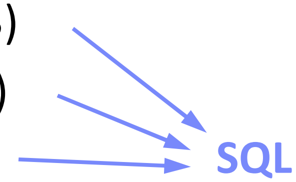
History 1980/90/2000s

OLTP … Online Transaction Processing
OLAP … Online Analytical Processing
ETL … Extract, Transform, Load

- **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)

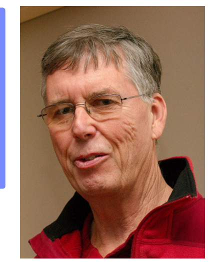


- **Different Personas**
  - Domain Experts (e.g., BI Tools, SAP R/3)
  - DB Application Developers (e.g., ABAP)
  - DB Developers and DB Admins



History 2000s / Early 2010s

[M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, P. Helland: **The End of an Architectural Era** (It's Time for a Complete Rewrite). **VLDB 2007**]



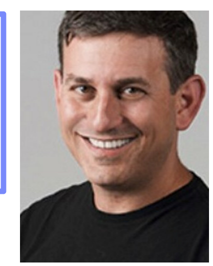
- **Specialized Systems**
  - **Column stores + compression** for OLAP
  - **Main memory systems** for OLTP and OLAP
  - Data streaming, scientific and graph databases
  - Information extraction / retrieval, RDF, and XML

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

RDF … Resource Description Framework

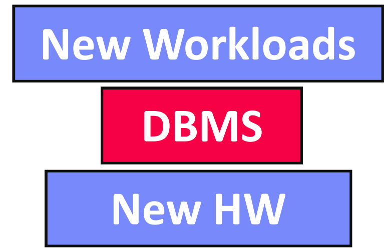
- **Toward Flexible, Large-Scale Data Management** (DWH … a bygone era)
  - **MAD Skills (magnetic, agile, deep)**, MADlib
  - Integration of R, Python in data analysis
  - Open data and its integration
  - Query processing over raw data files

[J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton: **MAD Skills**: New Analysis Practices for Big Data. **PVLDB 2(2) 2009**]

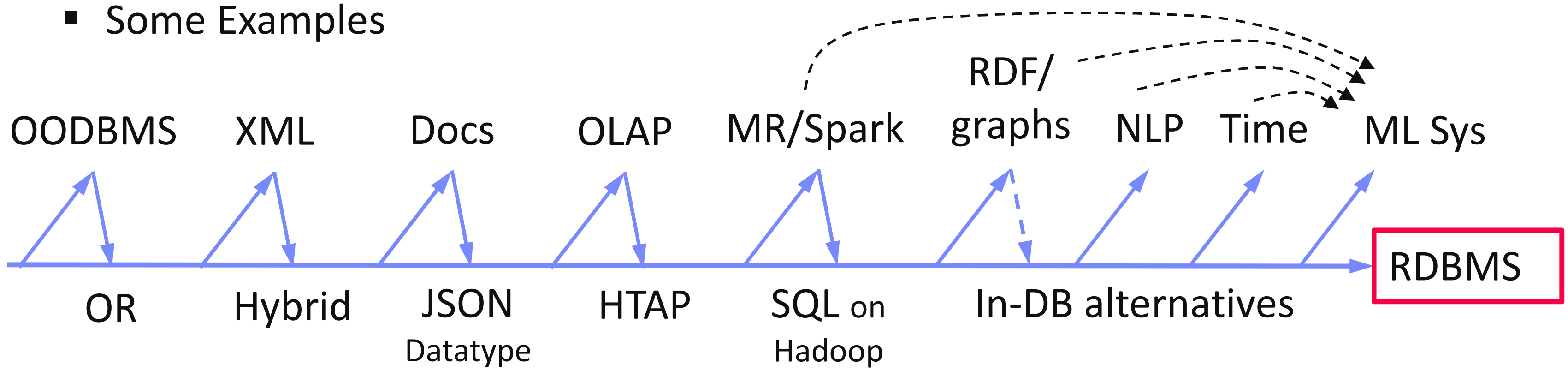


History 2010s – Present

- **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



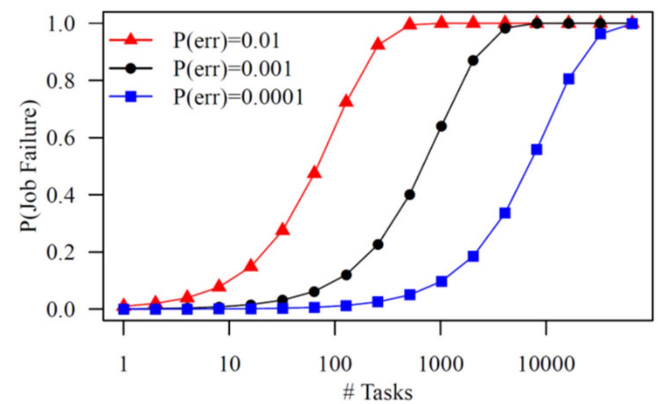
History 2010s – Present (2)

- **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 MapReduce and Spark, incl streaming
  - Automatic cloud resource elasticity (pay as you go)

Data Lake

Summary and Q&A

- **Database Systems**
  - Mature and established technology → broadly applicable & eco system
  - **General concepts:** abstraction, data modeling, query optimization & processing, transaction processing, logging & recovery, storage schemes and index structures, physical design and tuning

- **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 tools / systems
- Awareness of **system alternatives and their tradeoffs**
- **Fundamental concepts** as basis for advanced courses and other areas

Announcements

SIGMOD Programming Contest 2019

Extracurricular Activity

- **SIGMOD Programming Contest**
  - Since 2009, student teams of degree-granting institutions
  - Yearly contest, see last year http://sigmod18contest.db.in.tum.de/
  - Opportunity to compete and learn DB internals
  - Usually **prizes between $3.000-$7.000**
  - Current contest **not announced yet** (~ End Feb – End May)

- **Interested Students**
  - Should contact **Matthias Boehm** for mentoring
  - **Finalists** attend **SIGMOD 2019 in Amsterdam, NL** (we pay whatever is not covered by travel stipend)
  - Could be a great start into a research career and opportunity for networking

Extracurricular Activity

Experimentalphilosophische Studie **zur moralischen Intuition**

Mitterer Andreas, BA

Mag. iur. Galler Benjamin

andreas.mitterer@edu.uni-graz.at

Dauer: ~15 Minuten