

# 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  
BMK endowed chair for Data Management

# Announcements/Org

## ■ #1 Video Recording

- Link in [TeachCenter](#) & [TUbe](#) (lectures will be public)
- Optional attendance (independent of COVID)



## ■ #2 Course Registration (as of Oct 04)

- [Data Management VO](#): **138 (1)**
- [Data Management KU](#): 138
- [Databases VU](#): **68**

Total:

**206**

## ■ #3 COVID-19 Restrictions

- Max 25% room capacity (TC registrations)

## ■ #4 Startup Incubator “Gründungsgarage”

- [5min-overview](#) by Julia Harrer today
- If interested, apply by **Oct 11**



# Agenda

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

# Data Management Group

<https://damslab.github.io/>

# About Me

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



[https://github.com/  
apache/systemds](https://github.com/apache/systemds)

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

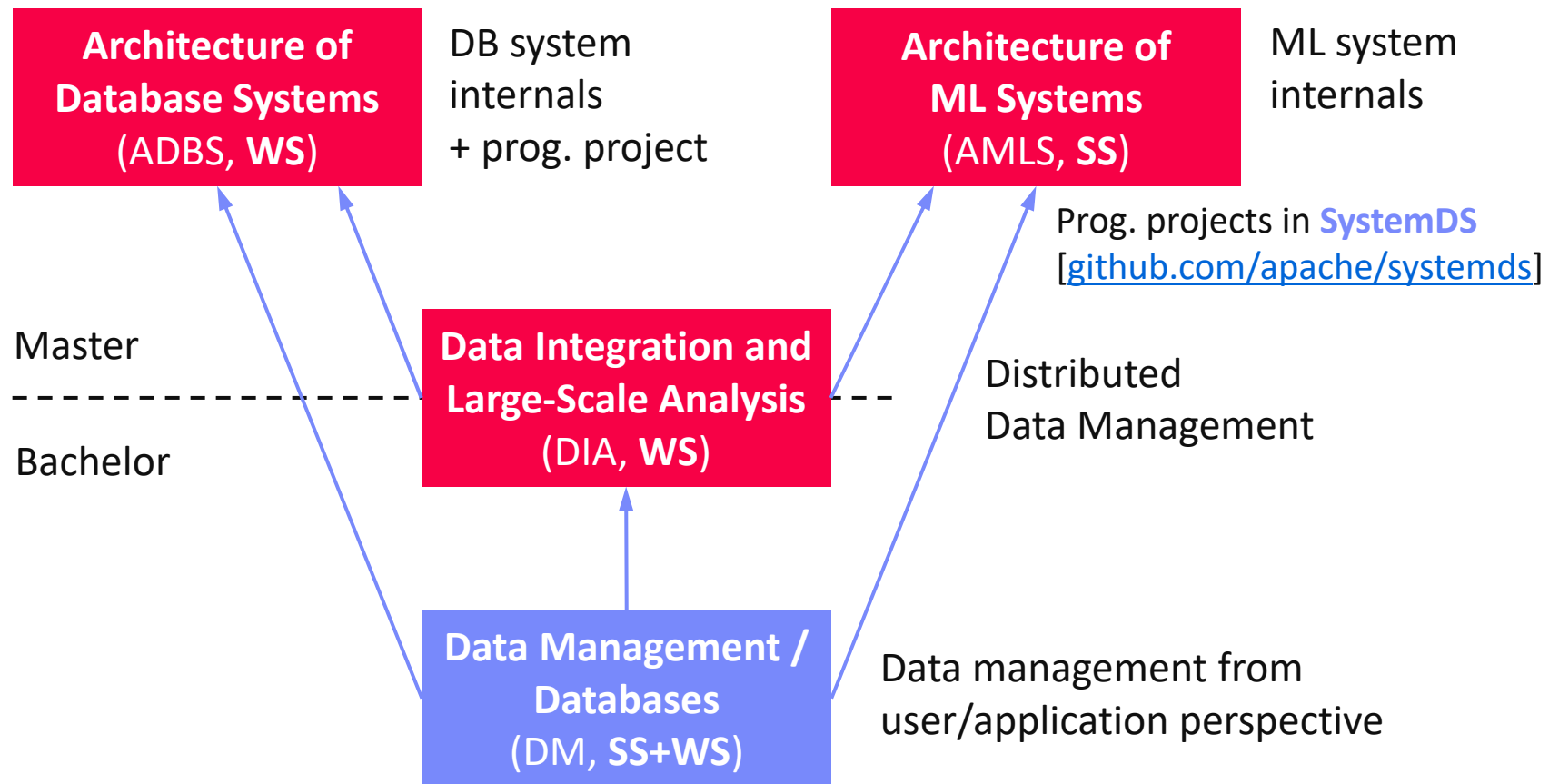


- **2011 PhD TU Dresden, Germany**
  - Cost-based optimization of integration flows
  - Systems support for time series forecasting
  - In-memory indexing and query processing



DB group

# Data Management Courses



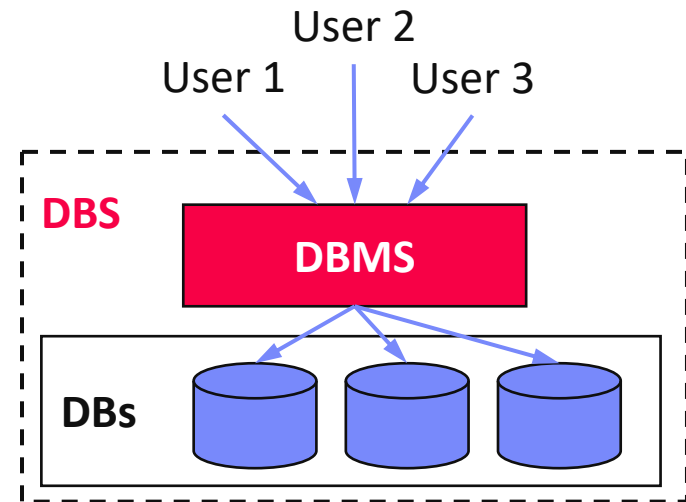
# Course Motivation, Goals, and Outline

**Database Systems** and Modern Data Management

# Definition and Impact

## Def: Database System

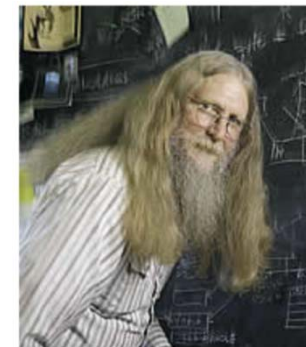
- Overall system of DBMS + DBs
- DBMS: Database Management System (SW to handle DBs)
- DBs: Database (data/metadata collection of conceptual mini-world)
- Note: DB also a short for DBS/DBMS



## Importance in Practice

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

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



Bruce Lindsay



**“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

# Goals

## Course Goals

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

	<b>INF.01017UF (VO) Data Mgmt.</b>	<b>INF.02018UF (KU) Data Mgmt.</b>
<b>706.010 (VU) Databases</b>	Part A 9 Lectures	Part A 3 Exercises
3(2) ECTS	Part B 3 Lectures	Part B 1 Exercise
	3 ECTS	1 ECTS

## 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** [Oct 05]
- **02 Conceptual Architecture and Design** [Oct 12]
- **03 Data Models and Normalization** [Oct 19]
- **04 Relational Algebra and Tuple Calculus** [Nov 09]
- **05 Query Languages (SQL, XML, JSON)** [Nov 16]
- **06 APIs (ODBC, JDBC, OR frameworks)** [Nov 23]
- **07 Physical Design and Tuning** [Nov 30]
- **08 Query Processing** [Dec 07]
- **09 Transaction Processing and Concurrency** [Dec 14]

→ Exercise 1:  
Data Modeling  
[Nov 03]

→ Exercise 2:  
Queries  
[Dec 01]

→ Exercise 3:  
Tuning  
[Dec 22]

For course Databases:  
part of Exercise 3 is **extra credit**

## Part B: Modern Data Management

- **10 NoSQL (key-value, document, graph, time series)** [Jan 11]
- **11 Distributed Storage and Data Analysis** [Jan 18]
- **12 Data Stream Processing Systems** [Jan 25]



Exercise 4:  
Spark (**extra credit**)  
**[Jan 26]**

- 
- **13 Q&A and exam preparation** [Jan 25]
  - **Final written exam** [TBD]

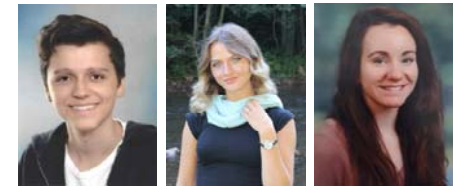
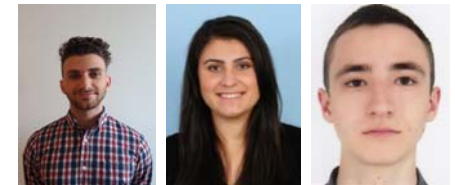
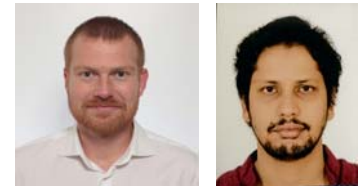
WS2020/21:  
**Automated Grading System**  
**for Exercises 2, 3, 4**  
(grading time, and  
consistent grading)

# Course Organization

# Basic Course Organization

## ■ Staff

- **Lecturer:** Univ.-Prof. Dr.-Ing. Matthias Boehm, ISDS  
**Assistant Lecturer:** M.Tech. Arnab Phani, ISDS
- **Teaching Assistants:**  
Dardan Dermaku, Nives Križanec, Artem Kroviakov,  
Thomas Mühlbacher, Ema Salkić, Fiona B. Trapp



## ■ Language

- Lectures and slides: **English**
- Communication and exams: **English/German**

## ■ Course Format

- DM VO + KU 2/1 (**3+1 ECTS**), DB VU 1/1 (**3(2) ECTS**)
- **Weekly lectures** (**start 6.10pm**, including **Q&A**), **attendance optional**
- **3+1 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 hour: Mo 12.30pm** (via <https://tugraz.webex.com/meet/m.boehm>), or after lecture

## ■ Website

- [https://mboehm7.github.io/teaching/ws2021\\_dbs/index.htm](https://mboehm7.github.io/teaching/ws2021_dbs/index.htm)
- All course material (lecture slides, exercises) and dates

## ■ Exam

- **Completed mandatory exercises** (Nov 03, Dec 01, Dec 22, [Jan 26])
- **Final written exam** (TBD, doodle for oral exams)
- **DB Grading** (30% exercises, 70% final), **DM Grading** (separate courses)

# Course Logistics, cont.

## ■ Exercises

- Written and programming assignments, submitted through **TeachCenter**
- Assignments **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: The Movies Dataset

## Dataset

- Derived (extracted, cleaned) from **The Movies Dataset** for movies year  $\geq 2011$
- Clone or download your copy from <https://github.com/tugraz-isds/datasets.git>
- Find CSV files in <datasets>/movies

## Exercises

- 01** Data modeling (relational schema)
- 02** Data ingestion and SQL query processing
- 03** Physical design tuning, query processing, and transaction processing
- 04** Large-scale data analysis (distributed query processing and ML model training)

**New**

### Movies

#### Overview

The following dataset was derived from a larger movie dataset - [The Movies Dataset](#). The original dataset contains metadata of over 45,000 movies as well as 26 million ratings from 270,000 users for all of the movies, which have been collected from [TMDB](#) and [GroupLens](#), respectively. From this large collection, data of around 12,000 movies from the past 10 years were extracted and stored as three denormalized CSV files (with ';' delimiter and simplified structure without the need for quoting): `Movies.csv` (cast members) and `Ratings.csv` (given to the movies by various users).

#### Structure:

`Movies.csv`: The movies file contains metadata on over 12,000 different movie titles. The datapoints included in the file are:

- `MovieID`: An ID that uniquely identifies each movie.
- `OriginalLanguage`: The language of the original movie title, denoted in its [ISO 639-1](#) language code equivalent.
- `OriginalTitle`: The title of the movie in its original language.
- `EnglishTitle`: The english equivalent of the original title.
- `Budget`: The amount of money invested into making the movie.
- `Revenue`: The amount of money generated by the movie.
- `Homepage`: A link to the movies website.
- `Runtime`: The duration of the movie in minutes.
- `ReleaseDate`: The date on which the movie was/will be released.
  - Format: yyyy-mm-dd
- `Genres`: A list of genres that the movie is categorized under.
  - Format: genre|genre2|...|genreN
- `CastID`: A list of 24 character long IDs, belonging to the movies cast members.
  - Format: castid1|castid2|...|castidN
- `ProductionCompanies`: A list of production companies involved with the movie.
  - Format: company1|company2|...|companyN
- `ProductionCountries`: A list of countries in which the movie was filmed, paired with their [ISO 3166-1](#) country code.
  - Format: code1-country1|code2-country2|...|codeN-countryN
- `SpokenLanguages`: A list of languages spoken in the movie, paired with their [ISO 639-1](#) language code.
  - Format: code1-language1|code2-language2|...|codeN-languageN

The following is an excerpt from the `Movies.csv` file which is representative of the dataset's structure:

```
MovieID,OriginalLanguage,OriginalTitle,EnglishTitle,Budget,Revenue,Homepage,Runtime,ReleaseDate,Genres,CastID,ProductionComp
136558,en,Kingdom Come,Kingdom Come,,http://www.kingdomcomefilm.com/#kingdomcome,88,2011-09-01,Comedy,...
118428,fr,Camille Claudel 1915,Camille Claudel 1915,3512454,115086,,95,2013-03-13,Drama,Canada|Arte France Cinéma|3B Product
62775,fi,Havukka-Ahon Ajattelijat,Havukka-Ahon Ajattelijat,2223000,,2018-01-15,Comedy|Drama,,fi-suomi
12477,en,When in Rome,When in Rome,,36699403,,91,2018-01-20,Fantasy|Comedy|Romance,Kranorra|Foster-Productions|Touchstone PI
12281,en,Edge of Darkness,Edge of Darkness,80000000,74001330,,117,2010-01-29,Crime|Drama|Mystery|Thriller,,Icon Productions|I
37034,sh,Su Qi-Er,True Legend,20000000,,115,2010-02-09>Action|Fantasy,,Shanghai Film Group|Focus Features|EKO Film|Eko Ma
```

`Persons.csv`: The persons file contains information about the cast members and their characters in the movies. The datapoints included in the file are:

- `MovieID`: An ID referencing the movie in which the character appeared.
- `CastID`: An ID that uniquely identifies each character played by the cast member.
- `Name`: The name of the cast member.
- `Gender`: The gender of the cast member (1 = female, 2 = male)
- `Character`: Full name of the movie's character.

The following is an excerpt from the `Persons.csv` file which is representative of the dataset's structure:

```
MovieID,CastID,Name,Gender,Character
136558,52f4c18c3a368484e1a6d23,Daniel Gillies,2,Himself
136558,52f4c18c3a368484e1a6d27,Rachael Leigh Cook,1,Meriel
118428,52f4e406c3a368484e0978029,Juliette Binoche,1,Camille Claudel
118428,52f4e406c3a368484e0978069,Jean-Luc Vincent,Paul Claudel
62775,52f4e406c3a368484e0978069,Kari Lehtinen,2,Konsta Pykäläinen
62775,52f4e406c3a368484e0978073,Tomi Korpela,1,maisteri Kronberg
```

`Ratings.csv`: The ratings file contains information about the ratings given to the movies by different users. The datapoints included in the file are:

- `UserID`: An ID identifying the user that published the rating.
- `MovieID`: An ID referencing the movie that had been rated.
- `Rating`: The user's rating of the movie on a scale from 1.0 to 5.0.
- `Timestamp`: Timestamp at which the user's rating was published.

The following is an excerpt from the `Ratings.csv` file which is representative of the dataset's structure:

```
UserID,MovieID,Rating,Timestamp
1,58559,4,0,1425942007
1,98621,4,0,1425941392
7,58559,5,0,1486235675
7,68744,1,5,1486235974
11,58559,4,5,1216701920
15,49539,3,5,1346002547
```

**Movies.csv**

**Persons.csv  
(Cast)**

**Ratings.csv**

# 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
- 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
- Alfons Kemper, André Eickler: Datenbanksysteme - Eine Einführung, 10. Auflage. De Gruyter Studium, de Gruyter Oldenbourg 2015, ISBN 978-3-11-044375-2, pp. 1-879

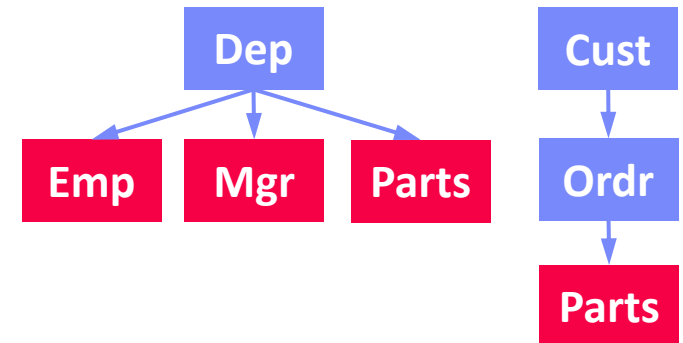
# History of Data Management

# History 1960/70s (pre-relational)

CODASYL ... Conference on  
Data Systems Languages

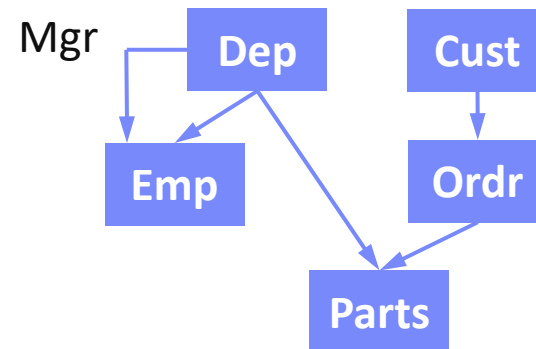
## ■ Hierarchical Model

- 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)

Oracle, IBM DB2, Informix, Sybase  
→ MS SQL



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

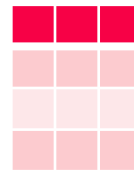
**System R @ IBM**  
Research – Almaden  
(Jim Gray et al.,  
**Turing Award '98**)



**Tuple Calculus**

**Relational Algebra**

**Relational Model**



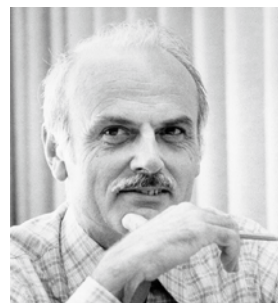
**QUEL**

**SEQUEL**

**SQL Standard (SQL-86)**

**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(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
```

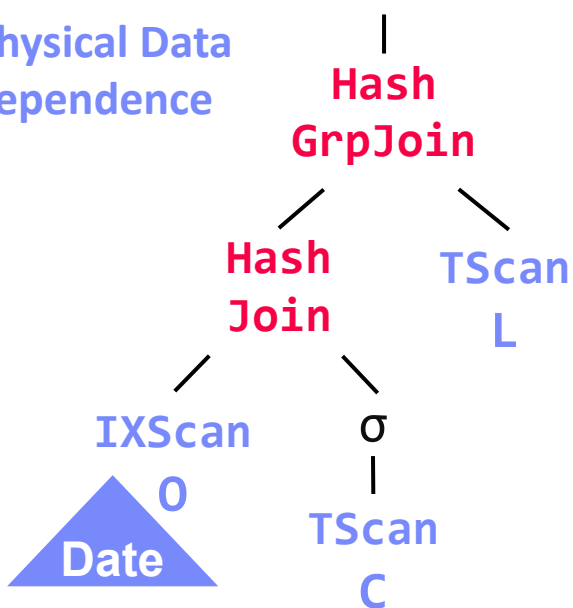
#1 **Declarative:**  
what not how

#2 **Flexibility:**  
closure property  
→ composability

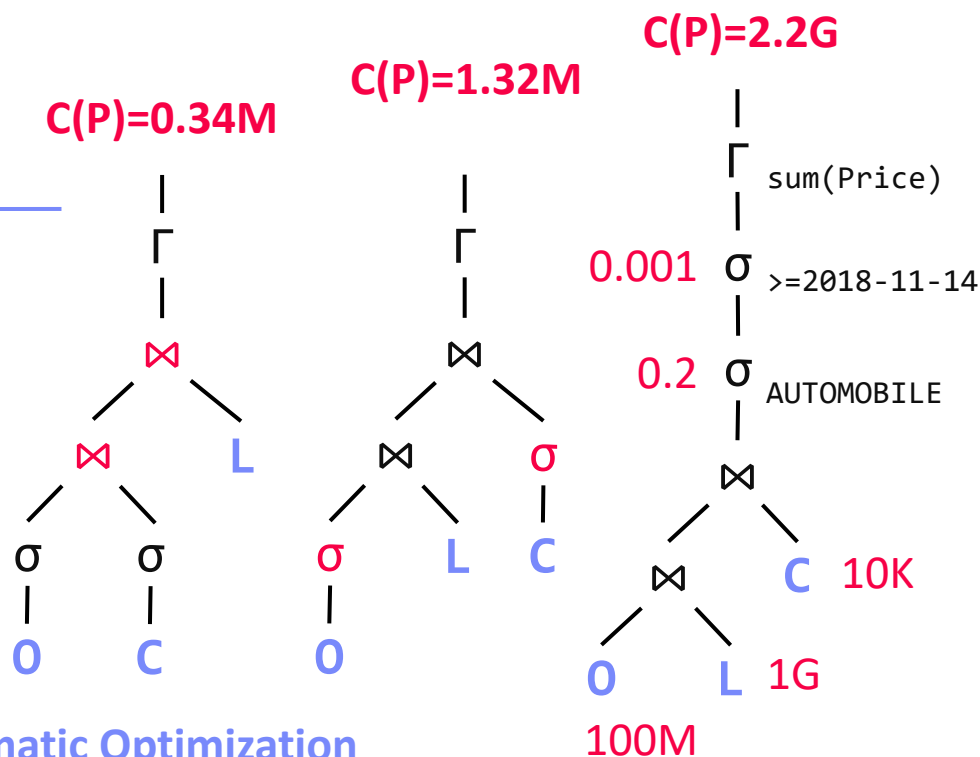
Logical Query Plans

#4 **Physical Data Independence**

Physical Query Plan



#3 **Automatic Optimization**



# 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

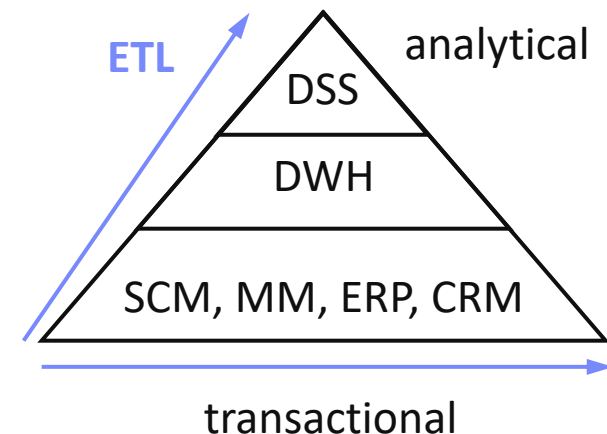
OLTP ... Online Transaction Processing  
 OLAP ... Online Analytical Processing  
 ETL ... Extract, Transform, Load

## ■ 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

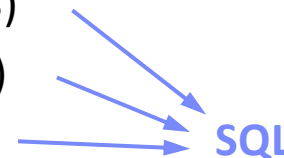
## ■ Information/Data Warehousing (DWH)

- **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, 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)

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

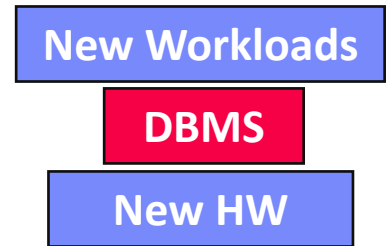
- **MAD Skills (magnetic, agile, deep)**, MADlib
- Integration of R, Python in data analysis
- Open data and its integration
- Query processing over raw data files



# History 2010s – Present

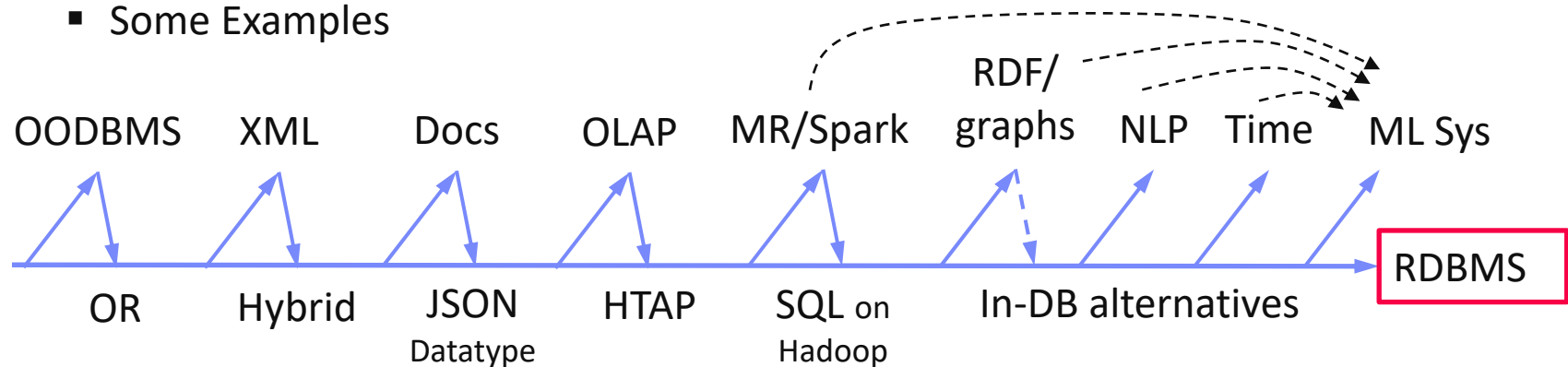
## Two Key Drivers of DB Research

- **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



## 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)

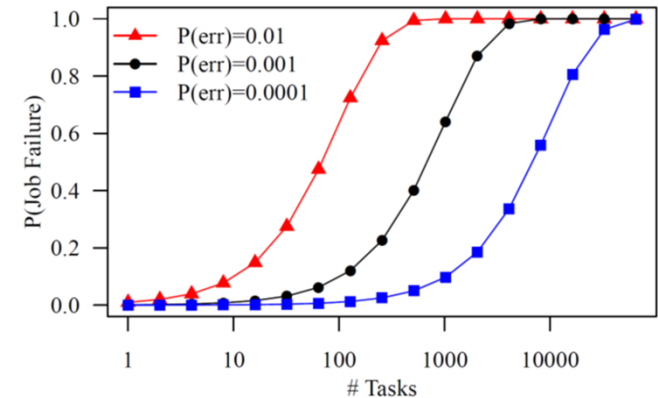
### ■ Example NoSQL Systems (local and distributed):

- **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)

} Data Lake

# 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

## ■ Next Lectures

- **02 Conceptual Architecture and Design** [Oct 12] (ER Diagrams)
- **03 Data Models and Normalization** [Oct 19] (ERD → Relational Model)