Universität Rostock Traditio et Innovatio

Evolution Management of Multi-Model Data (Position Paper)

Irena Holubova
Charles University Prague, Czech Republic

Meike Klettke
University of Rostock, Germany

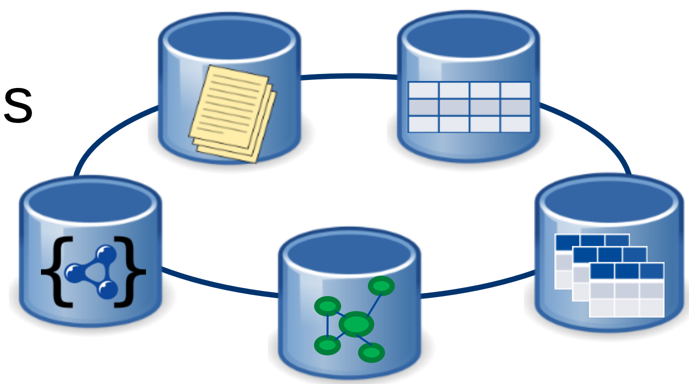
Uta Störl
University of Applied Sciences Darmstadt, Germany

Motivation /1

- Big Data movement changes many technologies
- Most challenging issue: **Variety** of data
  - Variety within one system (heterogeneous data)
  - Data in multiple types and formats (structured, semi-structured, and unstructured)

⇨ multi-model systems

⇨ Polystores

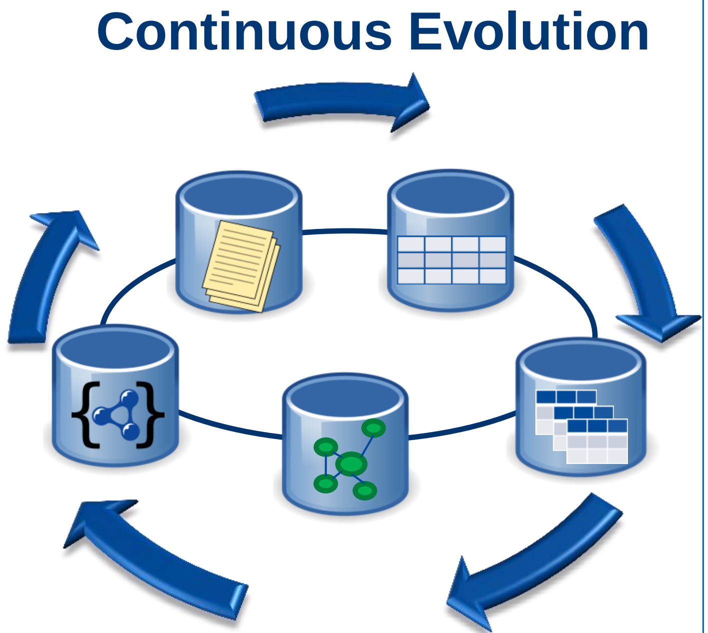


Motivation /2

"Software aging will occur in all successful products"

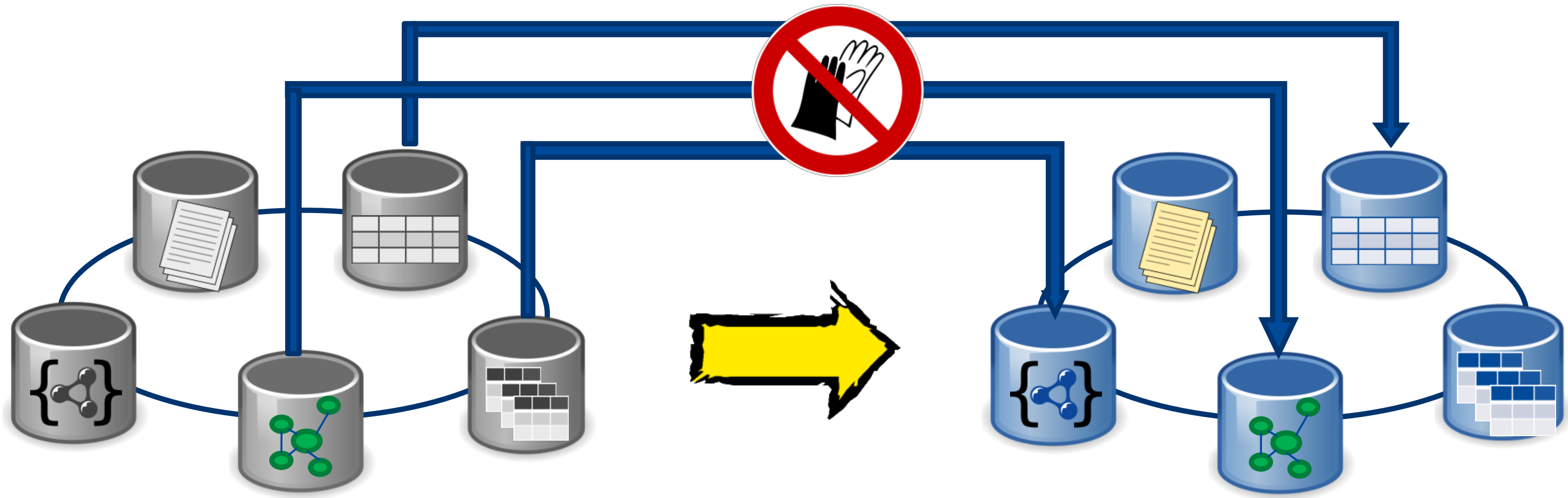
(David Lorge Parnas, 1994)

- ➡ Continuous redevelopment necessary
- ➡ Evolution of the databases in Polystores
  - – One of the most challenging tasks
  - – In Polystores new requirements arise from the **variety** of data



Motivation /3

- Avoid: manually written scripts for data migration



Motivation /4

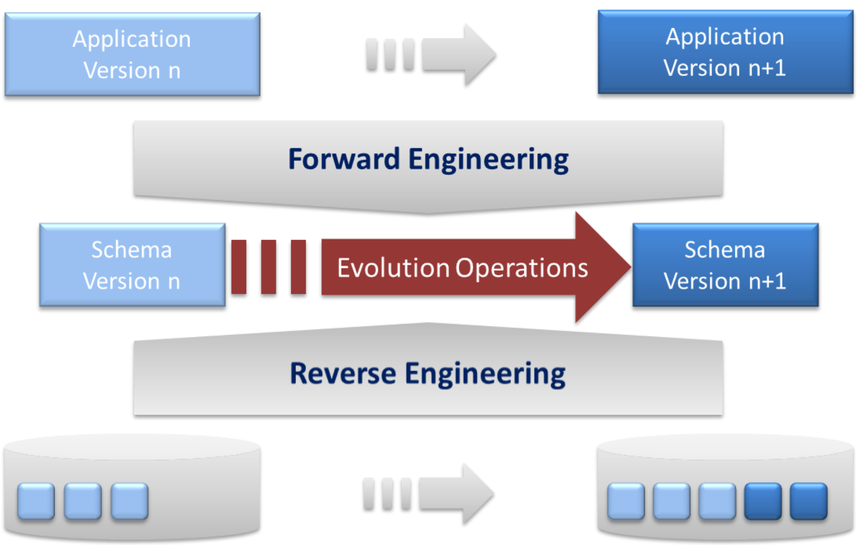
- We need the "**alter table"** for Polystores
- *Lingua franca* for all models in Polystores

State of the Art:

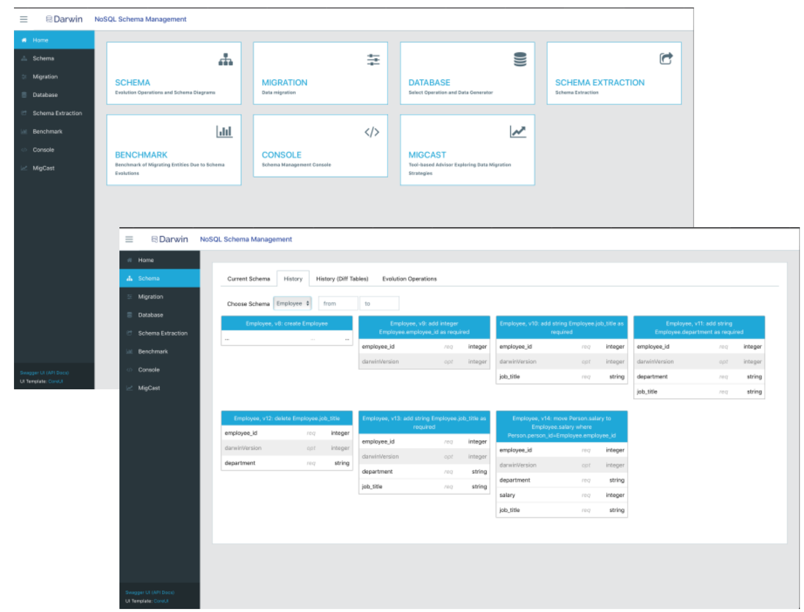
- Relational databases: (simple) schema evolution operations available
- XML: some research prototypes are available *(Challenge: Separation of schema and data)*
- NoSQL: first research activities (e.g. Darwin) *(Challenge: No schema information and Variety of data)*

Our Background

- Evolution for NoSQL databases
- Considering variety (within one database) and volume



Darwin



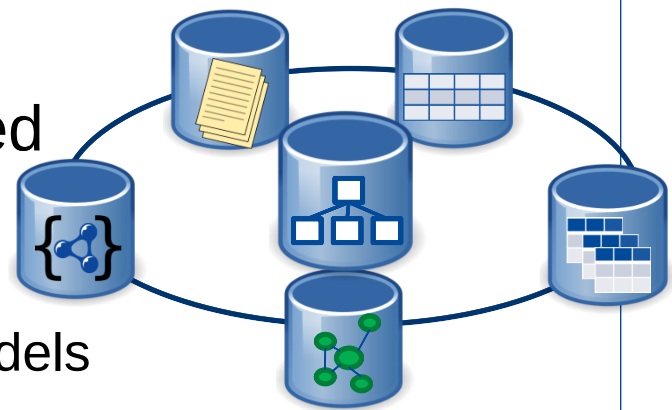
- *Meike Klettke, Uta Störl, Manuel Shenavai, Stefanie Scherzinger:* ***NoSQL Schema Evolution and Big Data Migration at Scale****, 4th Scalable Cloud Data Management Workshop SCDM @ IEEE Big Data Conference, Washington, D.C., USA, 2016*
- *Uta Störl, Daniel Müller, Julian Stenzel, Alex Tekleab, Stephane Tolale, Meike Klettke, Steffi Scherzinger:* ***Curating Variational Data in Application Development****, Demo presentation, ICDE 2018, Paris*

Challenges of Evolution in Polystores

1. Modeling of Multi-Model Data
2. Defining Evolution Operations
    1. Intra vs. Inter-Model Operations
    2. Single-Type vs. Multi-Type Operations
    3. Global vs. Local Evolution Operations
3. Executing Data Migration Operations (Eager, Lazy and Hybrid Approaches)
4. Inference of a Multi-Model Schema
5. ...

1. Modeling of Multi-Model Data

- Requirement: **Unified model** of a Polystore as starting point for evolution
- Object **modeling language** is needed
  - for relational data, objects, documents, graphs, streams, ..
  - representing **links** between different models

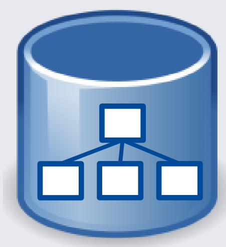


- Additionally
  - Translation from global object model into specific model (design)
  - Translation of model changes to a specific model (evolution)

➡ **Same methods for design/ evolution**

Modeling of Multi-Model Data

Challenge	State of the Art	Multi-Model Open Issues
Modelling	• Conceptual modeling languages (UML, ER, ...) are tailored for certain models • Translations between different models	• Formal definition of **intra-/ inter-model links**
		• **Multi-model conceptual modeling language** • Multi-model **schema definition** language
		• Support for **inter-model links, cross-model redundancy**



- *Alberto Hernández Chillón, Diego Sevilla Ruiz, Jesús García Molina, Severino Feliciano Morales: A Model-Driven Approach to Generate Schemas for Object-Document Mappers. IEEE Access 7: 59126-59142 (2019)*
- *Irena Holubova, Martin Svoboda, Jiaheng Lu: Unified Management of Multi-Model Data. ER '19, Salvador, Bahia, Brazil, 2019*

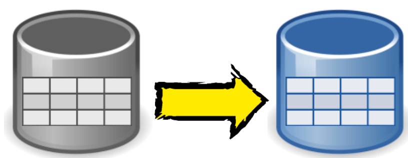
2. Schema Evolution Operations

- Schema evolution language for defining evolution operations **in all data models**
- Single-type operations
  - **add, delete, rename**
- Multi-type operations
  - **move/copy**
  - **split/merge**
- Details in the article 

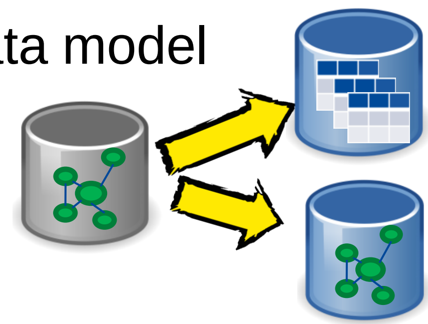
```
evolutionop = typeop | propertyop;

typeop     = createtype | droptype | renametype | split | merge;
createtype = "create type" kind;
droptype   = "drop type" kind;
renametype = "rename type" kind "to" kind;
split      = "split" kind "into" kind ":" pnames and kind ":" pnames;
merge      = "merge" kind ":" pnames "and" kind ":" pnames complexcond "into" kind;

propertyop = add | delete | rename | move | copy;
add        = "add" [datatype] property [defaultValue];
delete     = "delete" property;
rename     = "rename" property "to" pname;
move       = "move" property "to" ( kind | property ) complexcond;
copy       = "copy" property "to" ( kind | property ) complexcond;

complexcond = "where" joinconds ["and" conds];
joinconds   = joincond {"and" joincond};
joincond    = property "=" property;
conds       = cond {"and" cond};
cond        = property "=" value;

pnames   = pname ["as" pname] {"," pname ["as" pname]};
property = kind "." pname;
kind     = [mname "."] kname;
kname    = identifier;
mname    = identifier;
pname    = identifier;

defaultvalue = value;
```

2.1. Intra vs. Inter-Model Operations

Different kinds of evolution operations:



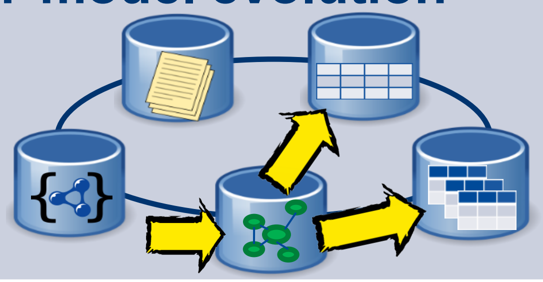
- **Intra-model operations** are translated into evolution operations within **one model**
- **Inter-model operations** change data in **different models**:
  - Evolving **links** between different models
  - Evolving data which are **redundant in different data models**
  - **Refactoring** a Polystore
    - Reorganization: moving data into another data model
    - Duplication of data (redundant storage for performance reasons)
    - Introduction of new models in a Polystore



Universität Rostock

Traditio et Innovatio

Intra vs. Inter-Model Operations

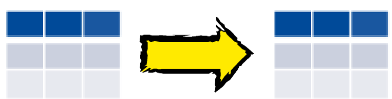
Challenge	State of the Art	Multi-Model Open Issues
Intra-/Inter-Model Operations	• Intra-model operations supported in some, but not all models	• Implementation of **intra-model operations** in **all models/ systems**
		• Implementation of **inter-model evolution operations**
		• **Discovering inter-model evolution operations** • Propagate them in a Polystore 

• *Stefanie Scherzinger, Meike Klettke, and Uta Störl: Managing Schema Evolution in NoSQL Data Store, The 14th International Symposium on Database Programming Languages DBPL@VLDB, Italy, 2013*

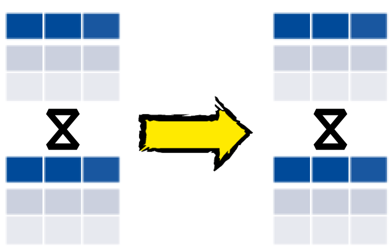
• *Michal Vavrek, Irena Holubova, Stefanie Scherzinger: MM-evolver: A Multi-Model Evolution Management Tool. EDBT 2019*

2.2. Single-type vs. Multi-type Evolution Operations

- Single-type operations
  - covering one table/ object/ document
  - e.g. **`add, delete, rename`**



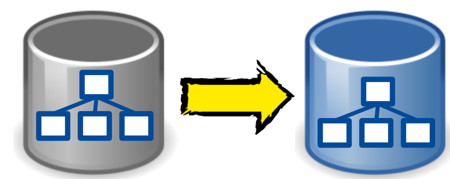
- Multi-type operations
  - cover two or more tables/objects/documents
  - e.g. **`move/copy`**, or **`split/merge`**
  - for complex operations or refactoring within one model



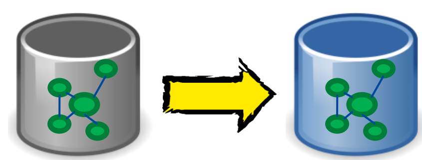
2.3. Global vs. Local Evolution Operations

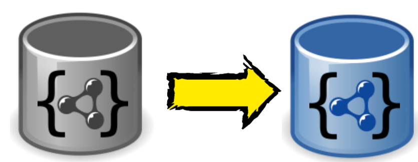
Evolution operation can be specified

- on the **global model**



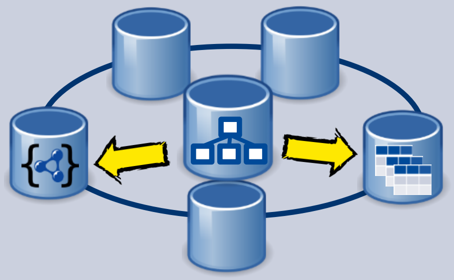
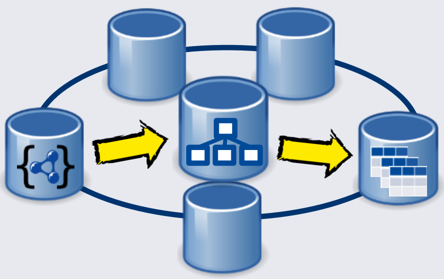
- on a certain **local model** of the Polystore





After executing evolutions: **Consistent state** of a Polystore has to be guaranteed

Global vs. Local Evolution Operations

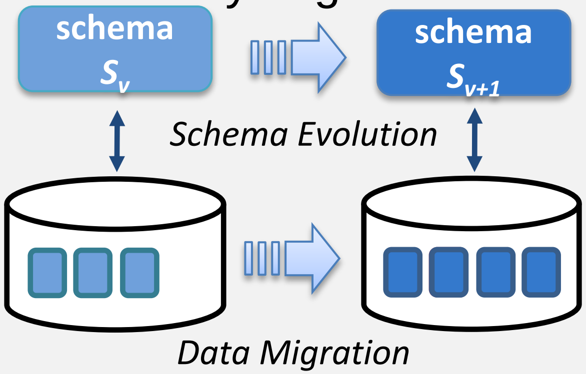
Challenge	State of the Art	Multi-Model Open Issues
Global/ Local Operations	• Local single-type evolution operations available in most systems • Multi-type operations (like split, merge, copy, move) only in research prototypes	• Propagation of **global operations** 
		• Detection and propagation of **local operations** 
		• Discovering **schema changes** from schema snapshots

Universität Rostock
Traditio et Innovatio

3. Basic Strategies of Data Migration

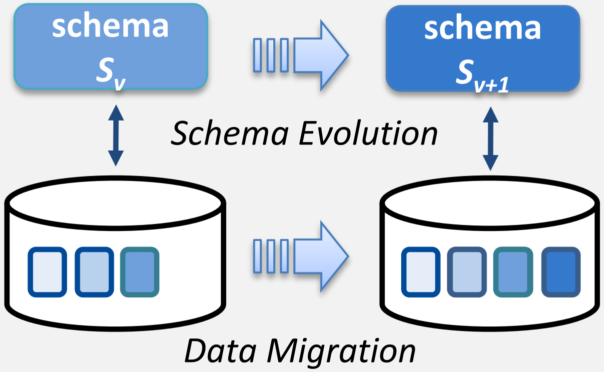
Eager Migration

- after introduction of *a new schema version*, all datasets are immediately migrated



Lazy Migration

- evolution operations are stored and data migration is done **on request**



Hybrid Approaches

- combination of both, **intelligent prediction** for data migration

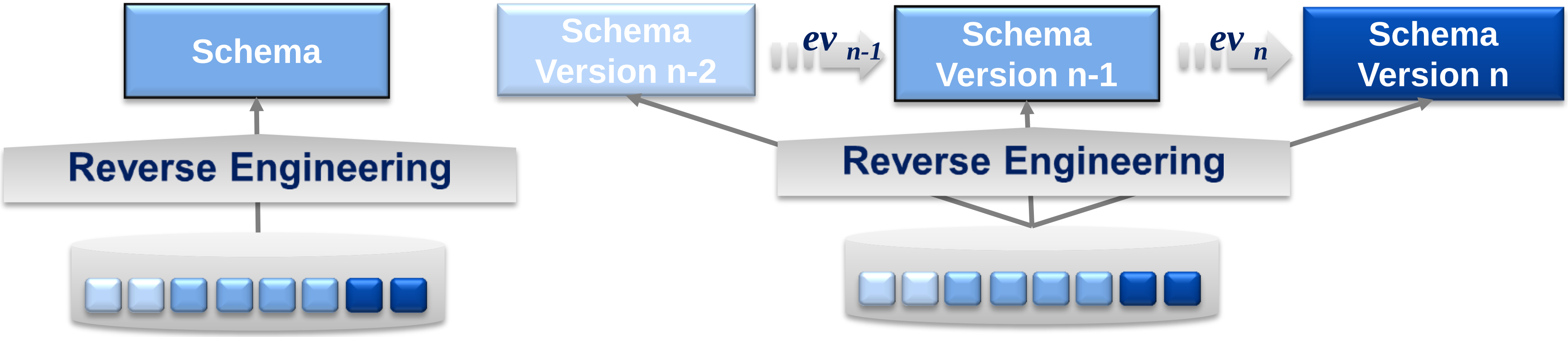
Eager vs. Lazy Migration in Polystores

Challenge	State of the Art	Multi-Model Open Issues
Eager/Lazy Migration	⇨ Eager/lazy migration is supported by some systems for single-type operations ⇨ Eager/lazy migration for multi-type operations only in research prototypes	⇨ Realizing **lazy migration** over different models / systems in case of inter-model operations
		⇨ **Synchronizing eager migration** over different models /systems ⇨ needs **version numbers** in all data sets or **timestamps**

- *Uta Störl, Alex Tekleab, Meike Klettke, Steffi Scherzinger: In for a Suprise When Migating NoSQL Data, Lightning Talk on the ICDE 2018, Paris*
- *Andrea Hillenbrand, Maksym Levchenko, Uta Störl, Stefanie Scherzinger, Meike Klettke: MigCast: Putting a Price Tag on Data Model Evolution in NoSQL Data Stores. SIGMOD Conference 2019, Amsterdam*

4. Schema Inference: Reverse Engineering

- Available for all data models, e.g. JSON
  - Deriving implicit structure
  - Deriving integrity constraints
  - Deriving changes over time (schema versions and evolution operations)



Schema Inference: Reverse Engineering

Challenge	State of the Art	Multi-Model Open Issues
Schema Inference	⇨ Single-model inference approaches (XML, JSON, RDF hierarchies, ...)	⇨ Multi-model (i.e., target) schema definition
		⇨ Multi-model inference approaches (heuristic / grammar-inferring)
		⇨ Mutual enrichment of single model subschemas (using, e.g., **inter-model links, redundancy**, ..)

- *Francisco Javier Bermudez Ruiz, Jesús García Molina, Oscar Díaz García: On the application of model-driven engineering in data reengineering. Inf. Syst. 72: 136-160, 2017*
- *Meike Klettke, Uta Störl, Stefanie Scherzinger: Schema Extraction and Structural Outlier Detection for NoSQL Data Stores. BTW 2015*
- *Meike Klettke, Hannes Awolin, Uta Störl, Daniel Müller, Stefanie Scherzinger: Uncovering the Evolution History of Data Lakes, 6th Scalable Cloud Data Management Workshop (SCDM) @ IEEE Big Data Conference, Boston, USA, 2017*
- *Irena Holubova, Stefanie Scherzinger: Unlocking the Potential of NextGen Multi-Model Databases for Semantic Big Data Projects, SBD@SIGMOD, Amsterdam, Netherlands, 2019*

Conclusion and Future Work

- Handling schema evolution and data migration in Polystores
- Most technologies are available for single-model systems
  - Either as state-of-the-art or
  - Research projects
- Most challenging:
  - Adding "inter-model features"
  - Guarantee data consistency for evolution

Literature

- P. Atzeni, F. Bugiotti, and L. Rossi. Uniform Access to NoSQL Systems. Information Systems, 43:117 - 133, 2014
- M.-A. Baazizi, D. Colazzo, G. Ghelli, and C. Sartiani. Parametric Schema Inference for Massive JSON Datasets. The VLDB Journal, Jan. 2019
- F. Bugiotti, L. Cabibbo, P. Atzeni, and R. Torlone. Database Design for NoSQL Systems. In Conceptual Modeling, pages 223-231, 2014
- C. Curino, H. J. Moon, L. Tanca, and C. Zaniolo. Schema Evolution in Wikipedia - Toward a Web Information System Benchmark. In ICEIS, Barcelona, Spain, pages 323-332, 2008
- G. Daniel, G. Sunye, and J. Cabot. UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases. In Conceptual Modeling, pages 430-444, 2016
- D. J. DeWitt, A. Halverson, R. V. Nehme, S. Shankar, J. Aguilar-Saborit, A. Avanes, M. Flasza, and J. Gramling. Split Query Processing in Polybase. SIGMOD, New York, USA, pages 1255-1266, 2013
- K. Herrmann, H. Voigt, J. Rausch, A. Behrend, and W. Lehner. Robust and Simple Database Evolution. Information Systems Frontiers, 20(1):45{61, 2018
- J. Lu and I. Holubova. Multi-Model Databases: A New Journey to Handle the Variety of Data. ACM Comput. Surv., 52(3):55:1-55:38, June 2019
- J. Lu, I. Holubova, and B. Cautis. Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges. In CIKM 2018, Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pages 2301-2302, 2018

Literature

- J. Lu and I. Holubova. Multi-model Data Management: What's New and What's Next? In EDBT 2017: Proceedings of the 20th International Conference on Extending Database Technology, pages 602-605, 2017 J. Pokorny. Conceptual and Database Modelling of Graph Databases. In IDEAS, pages 370-377, New York, USA, 2016
- J. Rumbaugh, I. Jacobson, and G. Booch. Unified Modeling Language Reference Manual. Pearson Higher Education, 2004
- K. Saur, T. Dumitras, and M. W. Hicks. Evolving NoSQL Databases Without Downtime. CoRR, abs/1506.08800, 2015
- S. Scherzinger, M. Klettke, and U. Störl. Managing Schema Evolution in NoSQL Data Stores. In Proceedings of DBPL 2013: Proceedings of the 14th International Symposium on Database Programming Languages, 2013
- J. Schildgen, T. Lottermann, and S. Deßloch. Cross-system NoSQL Data Transformations with NotaQL. In Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond, BeyondMR, pages 5:1-5:10, New York, USA, 2016
- D. Sevilla Ruiz, S. F. Morales, and J. Garca Molina. Inferring Versioned Schemas from NoSQL Databases and Its Applications. In Conceptual Modeling, pages 467-480, 2015