IDEAS 2021

Montreal, Canada

Multi-Model Data Modeling and Representation: State of the Art and Research Challenges

Pavel Čontoš, Irena Holubová, Martin Svoboda

svoboda@ksi.mff.cuni.cz

July 14, 2021

Charles University, Prague, Czech Republic

Data Variety

Structure of data

- **Logical models**
  - Relational, key/value, wide column, document, graph, …
- **Data formats**
  - XML or JSON for the document model, …
- **Schemas**
  - DTD or XML Schema schema languages, …
- **Vocabularies**
  - Names of XML elements or attributes, …

Other aspects

- **Technologies**: implementations, interfaces, protocols, …
- **Query languages**: syntax, constructs, expressive power

Database Systems

Traditional approach

- **Relational databases**
  - Primary option for decades
- Alternatives
  - Native XML databases, RDF stores, …

NoSQL databases

- Core models
  - Key/value, wide column, document, graph
- Finding the best model respecting the nature of data / queries
  - Not always possible

Multi-model databases

- Multiple models supported within just a single system

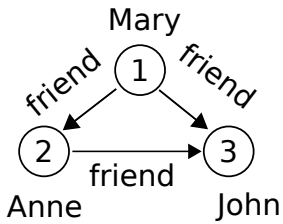
Sample Database

Multi-model scenario

relational table $\mathcal{T}$

customer	name	address	credit
1	Mary	...	3 000
2	Anne	...	2 000
3	John	...	5 000

property graph $\mathcal{G}$



document collection $\mathcal{D}$

```
{ order : 220,
  paid : true,
  items : [
    { product : T1, name : toy,
      price : 200, quantity : 2 },
    { product : B4, name : book,
      price : 150, quantity : 1 } ] }
```

wide-column table $\mathcal{W}$

customer	orders
1	[220, 230, 270, ...]
2	[10, 217]
3	[370, 214, 94, 137]

key/value pairs $\mathcal{K}$

customer	cart
1	product: T1, name: toy, quantity: 2 product: B4, name: book, quantity: 1
2	product: G1, name: glasses, quantity: 1 product: B2, name: book, quantity: 1
3	product: B3, name: book, quantity: 2

Multi-Model Databases

Multi-model databases

- One database for several different data models at a time
  - Provides a fully integrated backend
- More than 20 representatives
  - E.g.: OrientDB, ArangoDB, MarkLogic, Virtuoso, …

Issues and challenges

- **Underlying models**
  - Number of supported models, non-equal roles, …
- **Cross-model processing**
  - Links between the models, querying, indexing, …
- **Formal background**
  - Proprietary solutions (often not well documented)

Paper Objectives

Formal **unifying framework** is necessary

- Solid theoretical background
- But still user-friendly enough

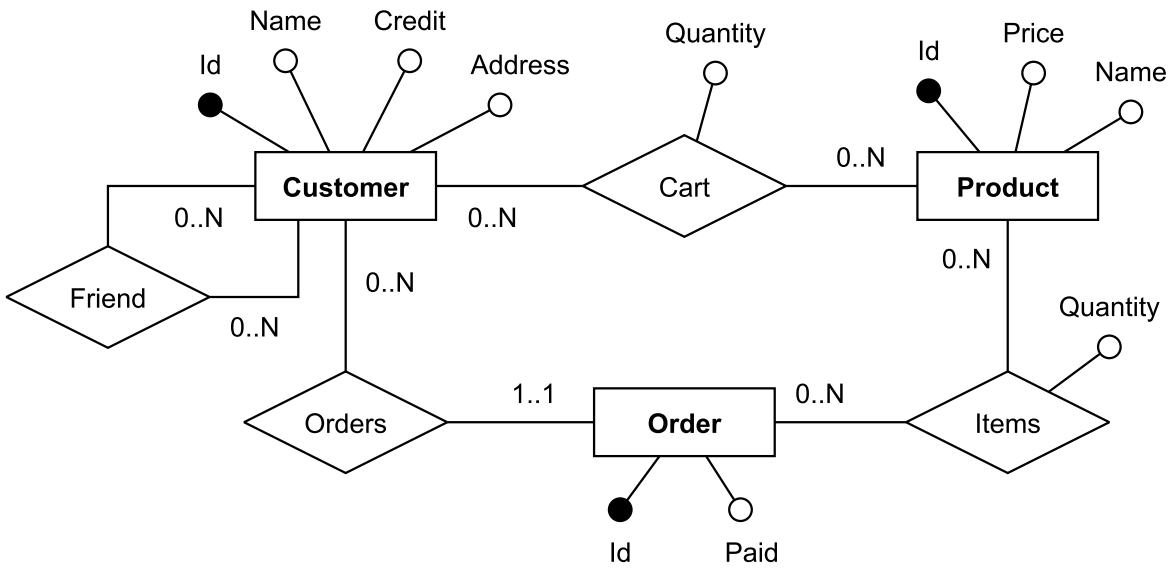
Our **objective**

- Survey of existing approaches that could be exploited
  - Conceptual modeling
  - Data representation
  - Integrity constraints
  - Evolution management
  - …

Conceptual Modeling

ER (**Entity-Relationship model**)

- Entity types, relationship types, attributes, identifiers, …

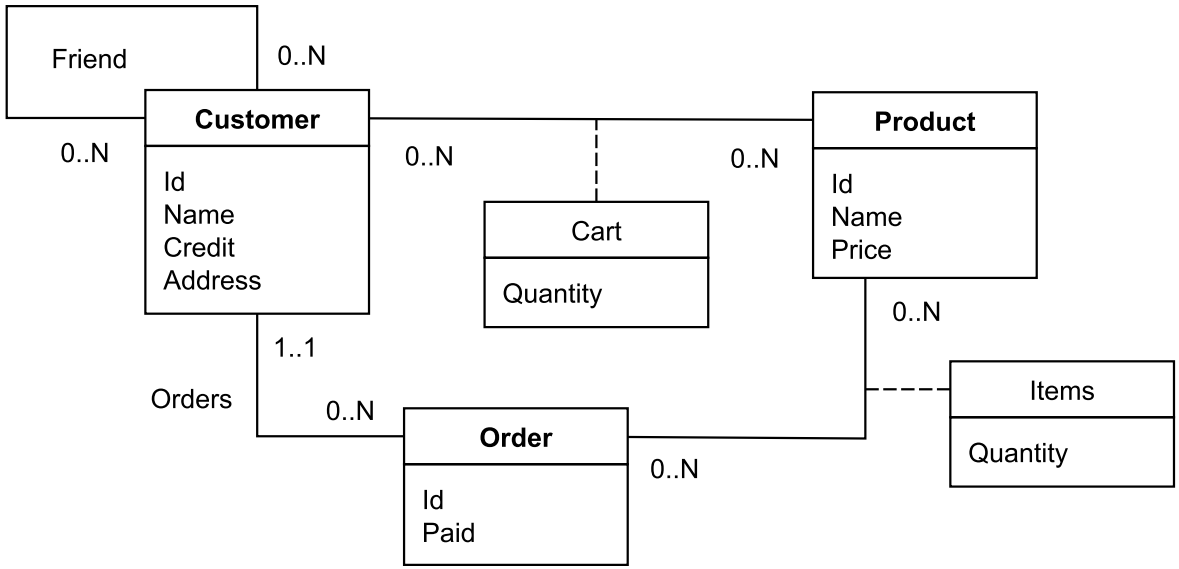


- Not standardized, various notations, structured attributes, identifiers for relationship types, participants of weak relationship types, non-unique or ordered values, …

Conceptual Modeling

UML (**Unified Modeling Language**)

- Classes, associations, attributes, …

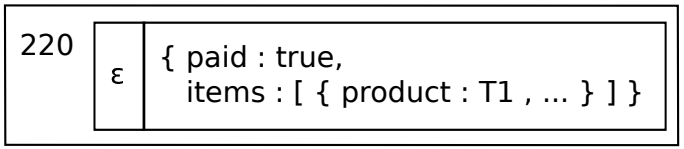


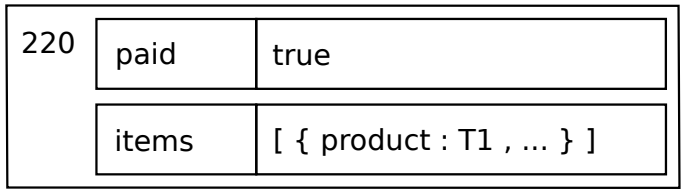
- Standardized, data oriented (conceals details such as weak entity types), …

Data Representation

NoAM (**NoSQL Abstract Model**)

- Data model
  - **Database** = **set of collections**, each with a unique name
  - **Collection** = **set of blocks**, each with a unique identifier
  - **Block** = **set of entries**, each with a unique key
  - **Entry** = **key / value pair**, values can be simple or complex
- Different strategies
  - Entry per Aggregate Object / Entry per Top-Level Field





- Aggregate-oriented models only (key/value, wide column, document), considered separately

Data Representation

Associative Arrays

- 2-dimensional matrix
  - $A : K_1 \times K_2 \rightarrow \mathbb{V}$
  - Mapping from row and column keys to values

relational table T

	name	address	credit
1	Mary	...	3 000
2	Anne	...	2 000
3	John	...	5 000

property graph G

	1	2	3
1	0	1	1
2	0	0	1
3	0	0	0

document collection D

	order	paid	items/product	items/name	...
001	220	true			
001/001			T1	toy	
001/002			B4	book	

- Not straightforward for all models, matrix operations

Integrity Constraints

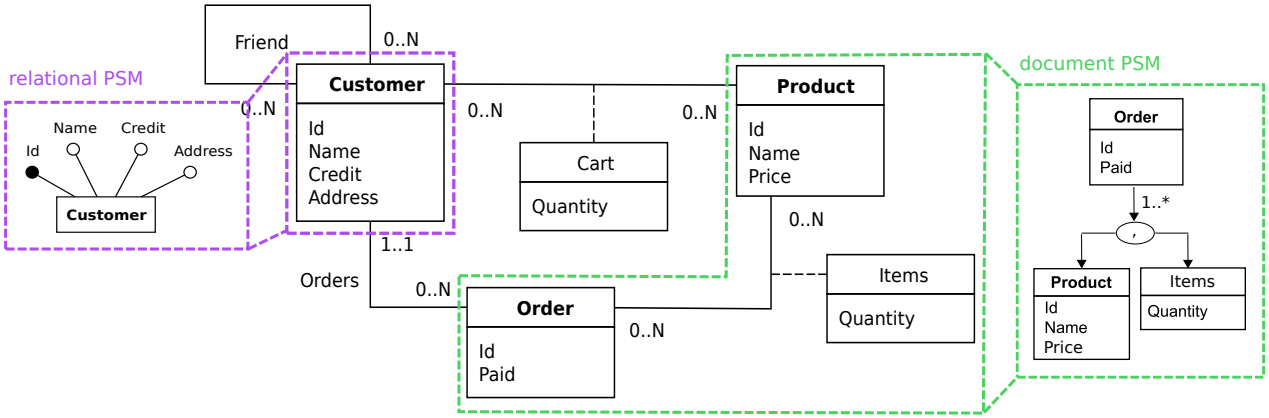
OCL (**Object Constraint Language**)

- Constructs
  - Pre-conditions and post-conditions for methods and operations
  - Rules for initial or derived values of attributes
  - **Invariants** = assertions data instances must satisfy
- Example
  - Each order must have at least one ordered item
  - ```
    context Customer inv :
      self.Orders->forAll( o | o.Items->size() >= 1 )
    ```
- Observations
  - Complex, conceptual layer
    ```

Evolution Management

DaemonX

- Evolution management framework
  - **Platform-independent model** (PIM)
  - Individual single model **platform-specific models** (PSMs)
  - Schema, operational, and extensional levels
- Correct and complete propagation of evolution changes



- Without inter-model links, without cross-model queries

Broader Generalization

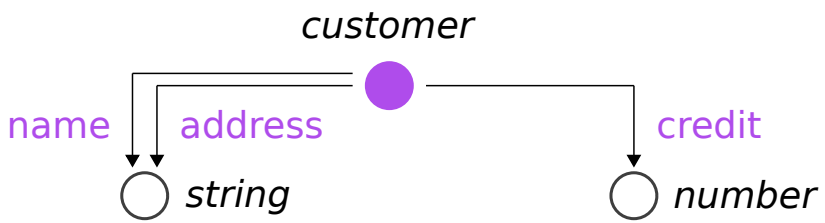
Category theory

- **Category** $\mathbf{C} = (\mathcal{O}, \mathcal{M}, \circ)$
  - Set of **objects** $\mathcal{O}$ (acting as multigraph vertices)
  - Set of **morphisms** $\mathcal{M}$ (acting as directed edges)
    - Each modeled as an arrow $f : A \to B$ with objects A, B
  - **Composition** operation $\circ$ for the morphisms
- Requirements
  - **Transitivity**: $g \circ f \in \mathcal{M}$ for any suitable morphisms f, g
  - **Associativity**: $h \circ (g \circ f) = (h \circ g) \circ f$ for any suitable f, g, h
  - **Identities**: identity morphism 1_A for any object A such that $f \circ 1_A = f = 1_B \circ f$ for any suitable morphism f
- Example
  - $\mathbf{Set}$: objects are sets, morphisms functions between them

Broader Generalization

Spivak 2009

- Description of a **schema of a relational database**
  - **Objects** for tables and generalized data types
  - **Morphisms** for attributes and foreign keys
    - And respective identity morphisms



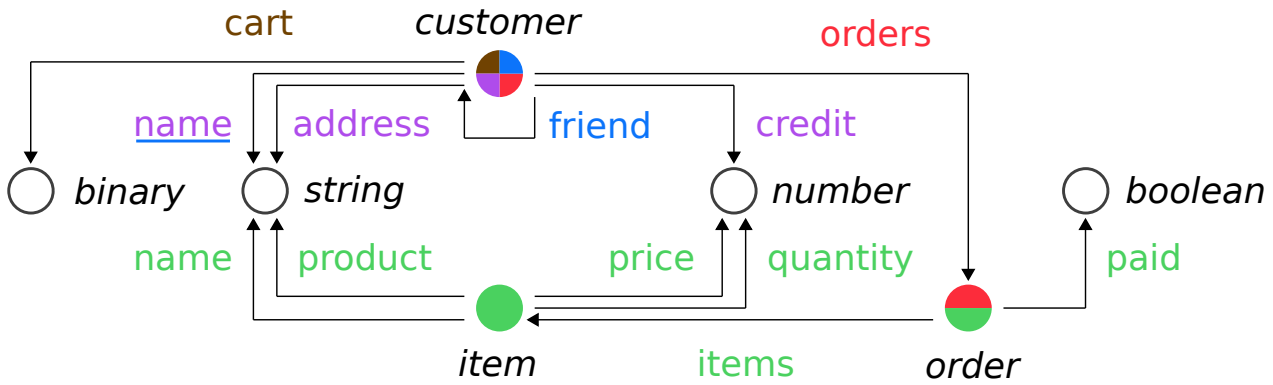
Observations

- Compulsory single-column primary key, relational model only

Broader Generalization

Multi-model scenario

- Draft of a possible extension to the previous approach



Challenges

- Contents of key/value pairs (black boxes), ordered collections (JSON arrays), embedded structures (JSON subdocuments), shared morphisms across models (names of customers), directions of morphisms, compound primary keys, …

Conclusion

Observations

- **Multi-model systems** grow in importance
- **Unifying conceptual framework** is necessary
  - Single-model solutions exist
  - But they cannot be straightforwardly adopted

Particular **challenges**

- **Schema design**
- **Data representation**
- **Unified querying**
- **Evolution management**
- **Autonomous database**

Thank you for your attention…