A Categorical Framework on Multi-Model Databases

Jiaheng Lu
University of Helsinki, Finland

Oracle ST-Seminar 23.08.2019

Purpose of this talk

Category theory can be used to model multi-model databases.

- Category theory provides a unified data model for multi-model databases.
- Categorical framework can provide powerful query language for multi-model DB.

A grand challenge on Variety

- Big data: Volume, Variety, Velocity, Veracity
- Variety: hierarchical data (XML, JSON), graph data (RDF, property graphs, networks), tabular data (CSV), etc

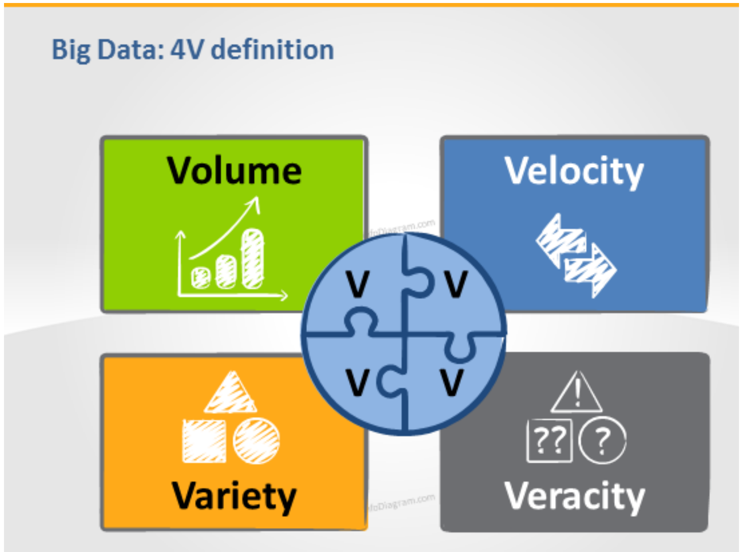
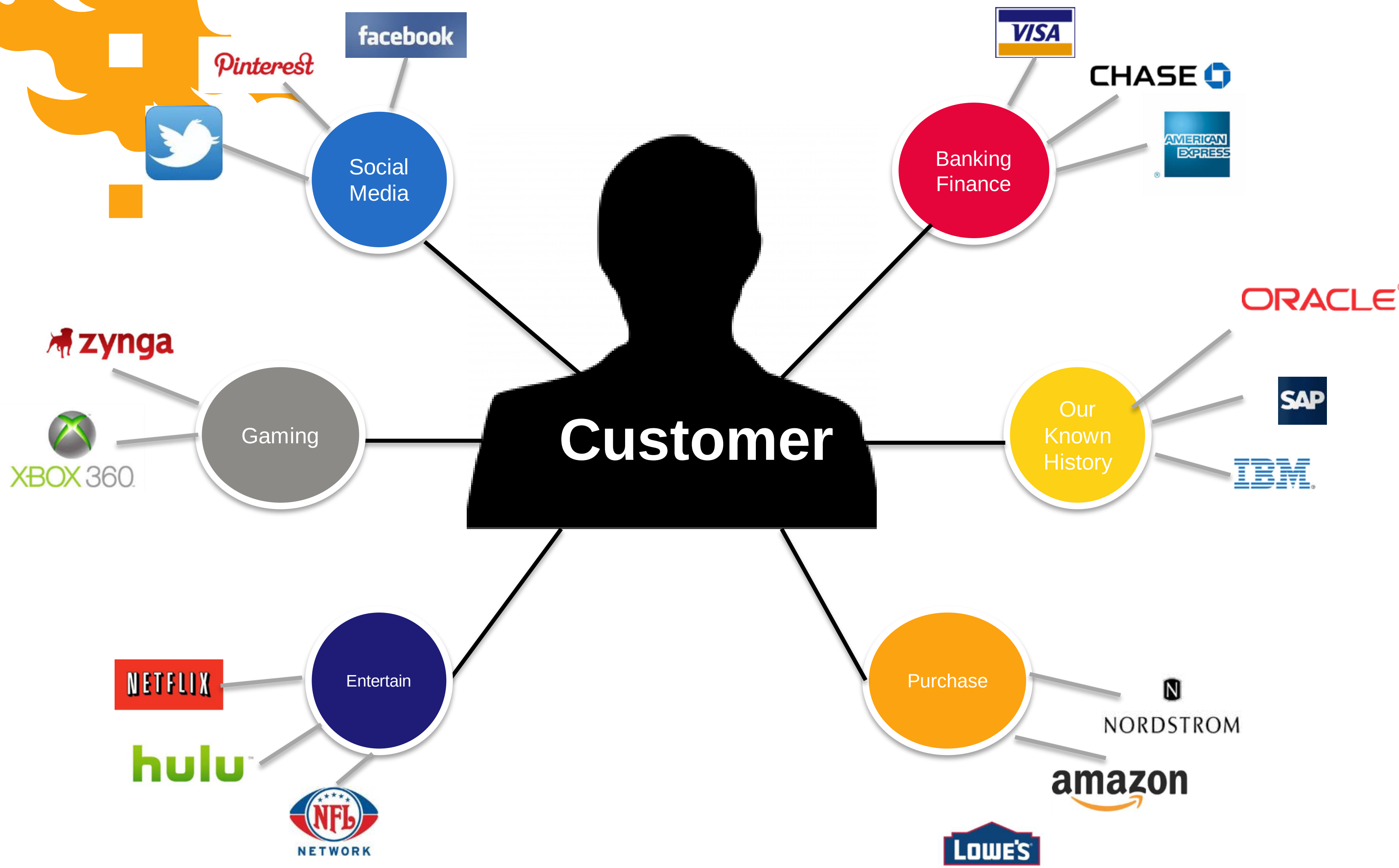


Photo downloaded from: https://blog.infodiagram.com/2014/04/visualizing-big-data-concepts-strong.html

Motivation: E-commerce

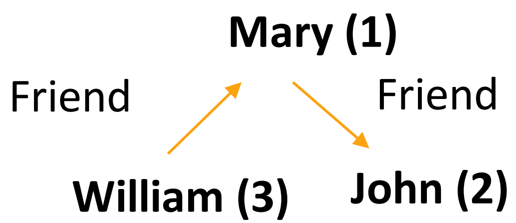


Motivation: one application to include multi-model data

- Relational data: customer databases
- Graph data: social networks
- Hierarchical data: catalog, product
- Text data: customer review
- ……

An E-commerce example with multi-model data

An example of multi-model data and query



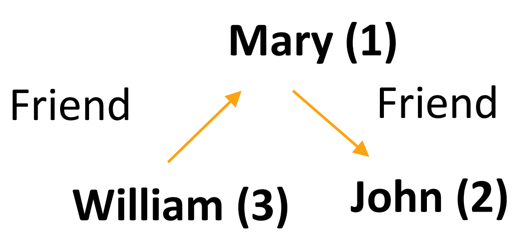
"1" -- > "34e5e759"

"2"-- > "0c6df508"

ID	Name	Credit	Phone
1	Mary	5,000	{ "type": "home", "number": "212 555-1234"}
2	John	3,000	{"type": "fax", "number": "646 555-4567"}
3	William	2,000	{ "type": “work", "number": "2198 -111-4321",

```
{"Order_no":"0c6df508",
 “Orderlines": [
    { "Product_no":"2724f”
      “Product_Name":“Toy",
      "Price":66 },
    { "Product_no":“3424g”,
      "Product_Name":“Book",
      "Price":40 } ]
}
```

Q: Return all products which are ordered by a friend of a customer whose credit>3000




```
"1" -- > "34e5e759"
"2"-- > "0c6df508"
```

ID	Name	Credit	Phone
1	Mary	5,000	{ "type": "home", "number": "212 555-1234"}
2	John	3,000	{"type": "fax", "number": "646 555-4567"}
3	William	2,000	{ "type": “work", "number": "2198 -111-4321",

```
{"Order_no":"0c6df508",
 “Orderlines": [
   { "Product_no":"2724f”
     “Product_Name":“Toy",
     "Price":66 },
   { "Product_no":“3424g”,
     "Product_Name":“Book",
     "Price":40 } ]
}
```

An example of multi-model query (ArangoDB)

```
Let CustomerIDs =(FOR Customer IN Customers FILTER
Customer.CreditLimit > 3000 RETURN Customer.id)

Let FriendIDs=(FOR CustomerID in CustomerIDs FOR
Friend IN 1..1 OUTBOUND CustomerID Knows return
Friend.id)

For Friend in FriendIDs

For Order in 1..1 OUTBOUND Friend Customer2Order

Return Order.orderlines[*].Product_no
```

Recommendation query:

Return all products which are ordered by a friend of a customer whose credit>3000

Multi-model databases

- A multi-model database is designed to support multiple data models against a single, integrated backend.

- Document, graph, relational and key-value models are examples of data models that may be supported by a multi-model database.

Multi-model databases: One size fits multi-data-model

ORACLE

mongoDB

MariaDB

Azure Cosmos DB

ArangoDB

OrientDB

APACHE DRILL

FOUNDATIONDB

MarkLogic

InterSystems CACHÉ

AsterixDB

...

Need theoretical foundation for multi-model databases

Call for a unified theory for manipulating & transforming multi-model data

The theory of relations (150 years old) needs to be extended to mathematically describe transformation in multi-model databases

Categories Defined

- A category C is ….
  - a collection of objects ob(C) .. {X,Y, Z ….}
  - a collection of morphisms {f, g ….}
    - A set of morphisms from object X into Y is denoted by Hom (X, Y) or X→Y.

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[dr, "g \circ f"'] & Y \arrow[d, "g"] \\ & Z \end{tikzcd}$$

Categories Defined (con't)

- The category must satisfy the following rules
- Composition of morphisms:
- for any three objects X, Y, and Z, we have
- If f: $X \to Y$, g: $Y \to Z$,
- then there is a composed morphism $f \circ g: X \to Z$.
- Having associative and identity laws

An example of multi-model data

Mary (1)

Friend

Friend

William (3)

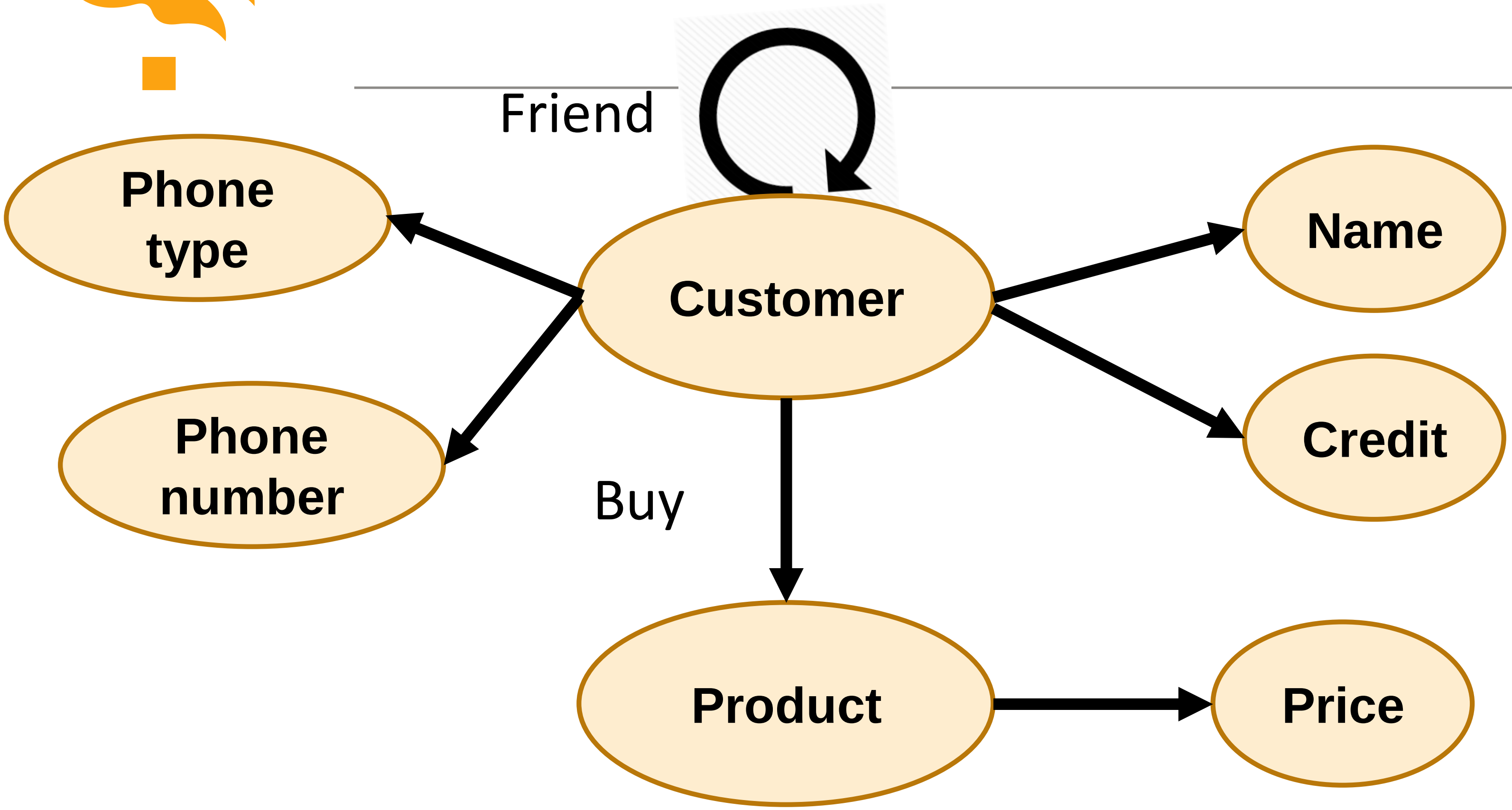
John (2)

```
"1" -- > "34e5e759"
"2"-- > "0c6df508"
```

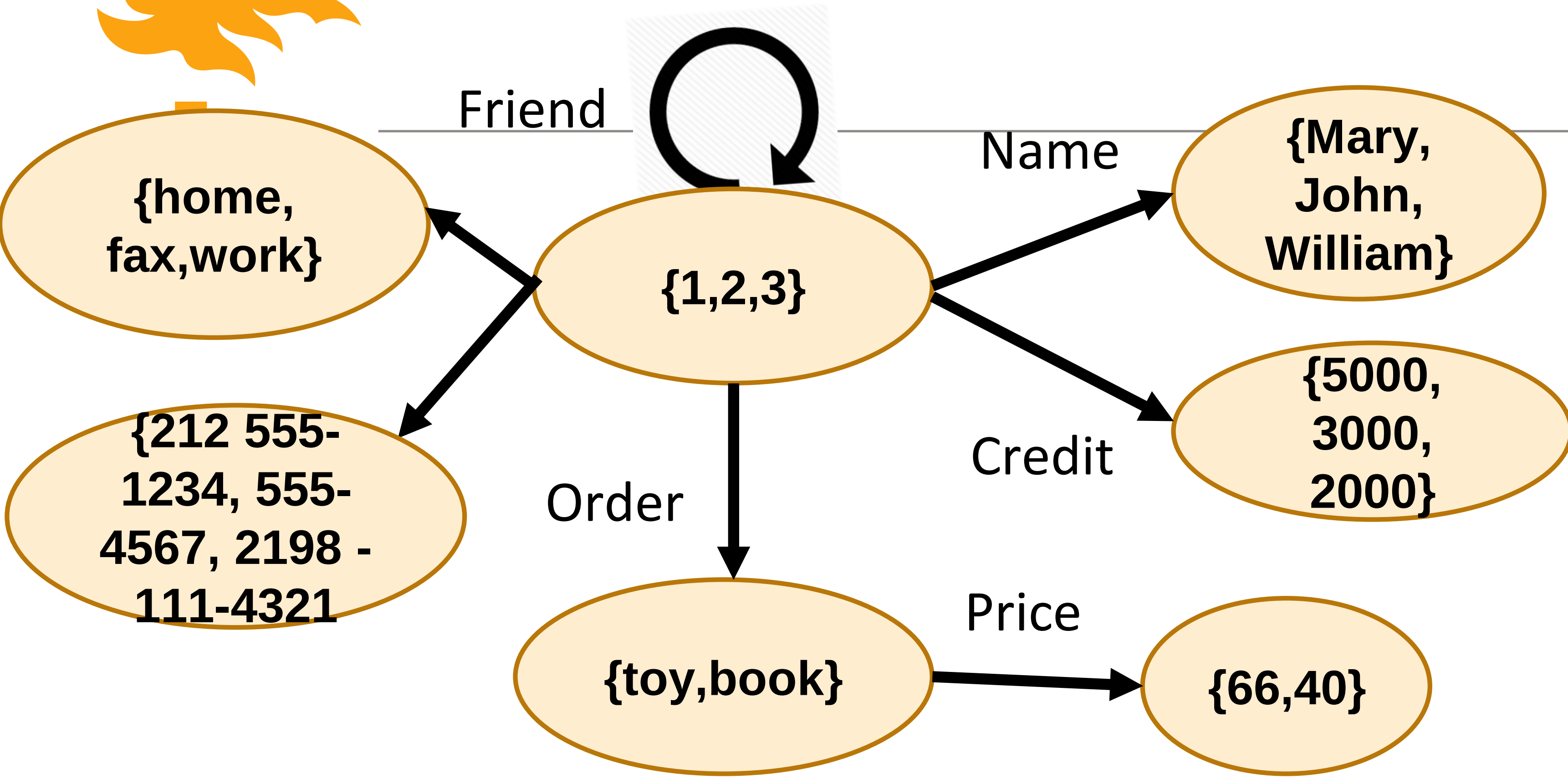
ID	Name	Credit	Phone
1	Mary	5,000	{ "type": "home", "number": "212 555-1234"}
2	John	3,000	{"type": "fax", "number": "646 555-4567"}
3	William	2,000	{ "type": “work", "number": "2198 -111-4321",

```
{"Order_no":"0c6df508",
 “Orderlines": [
   { "Product_no":"2724f”
     “Product_Name":“Toy",
     "Price":66 },
   { "Product_no":“3424g”,
     "Product_Name":“Book",
     "Price":40 } ]
}
```

Categorical model for multi-model data



Instance categorical model



Morphisms:
Friend: 1 → 2, 3 → 2;
Name: 1 → Mary, 2 → John, 3 → William …….

Difference between category model and Entity-Relation model (I)

- 1. Category model supports the composition of morphisms (relations)
- Friend (A,B), Friend(B,C) $\Rightarrow$ Friend (A,C)
- Can define the reachability query in graph databases
- Can define the descendant axes in XML databases

SQL and logic

- E.F. Codd, 1970:
- A SQL query can be defined as a first order logic.

- For example: "*List the set of employees' names and their department whose chair is named Cher*"

- ***Select*** *Employee_name*
- ***From*** *Dep, Employee*
- ***Where*** *Employee.depID=Dep.ID and Dep.chair="Cher"*

- $\exists x, \exists y$, Dep(x,"Cher") $\wedge$ Employee(x,y)

- Fagin, 1976: **Graph reachability is not expressible in first-order logic!**

There is no first-order formula ϕ(x, y) that says there is a path in graph G from node x to node y.

Categorical framework can present the reachability query.

- $\exists x, \exists y, \exists f,\ f(x,y)$
  where f is a composed morphism in the category

Computation complexity

- Categorical query can support other complicated path query:
- Find persons x and z such that x is parent of y and x know one of teachers of z and y (recursively) knows z
- $\exists x, \exists z \ \ p(x,y) \wedge k^*(y,z) \wedge (k \circ t)(x,z)$

Query fragment	Evaluation	Containment
Categorical Path Query	NLOGSPACE-complete	PSPACE-complete
Categorical Path Query with Counting	#P-complete	???

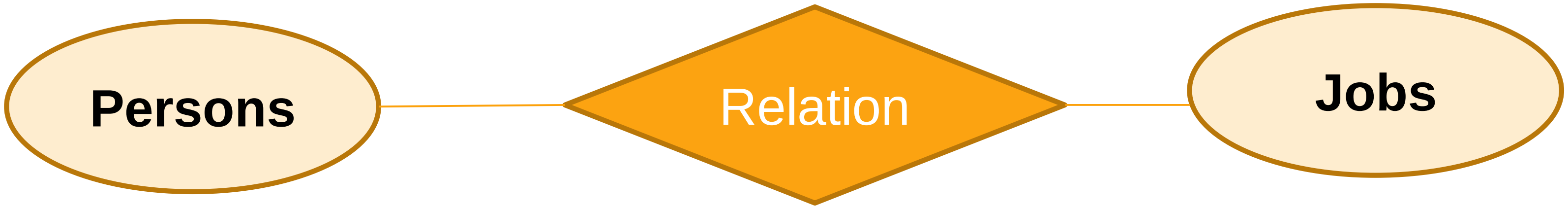
Difference between category model and Entity-Relation model (II)

- 2. Category model can define exponential object.
- Given two objects, an exponential object is an object X^Y equipped with an evaluation map $ev: X^Y \times Y \to X$ which is universal in the sense that, given any object Z and map $e: Z \times Y \to X$, there exists a unique map $u: Z \to X^Y$

Difference between category model and Entity-Relation model (II)

Difference between category model and Entity-Relation model (II)

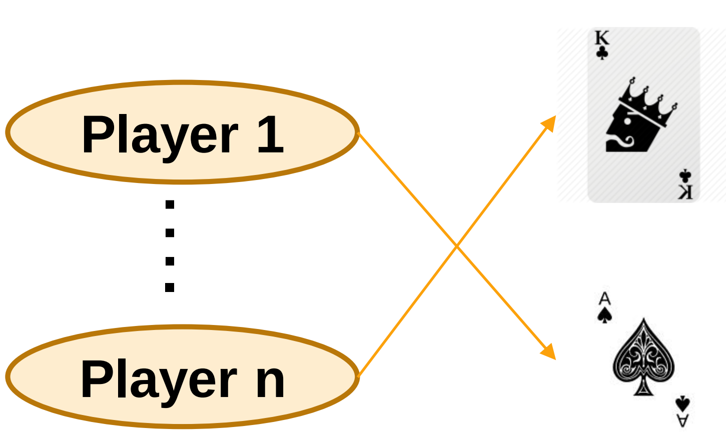
- 2. Category model can define exponential object. Exponential object is different from the Relation attribute in ER.



$\neq \mathrm{Jobs}^{\mathrm{Persons}}$

Exponential object includes all morphisms between two objects, but relation attribute defines only one morphism.

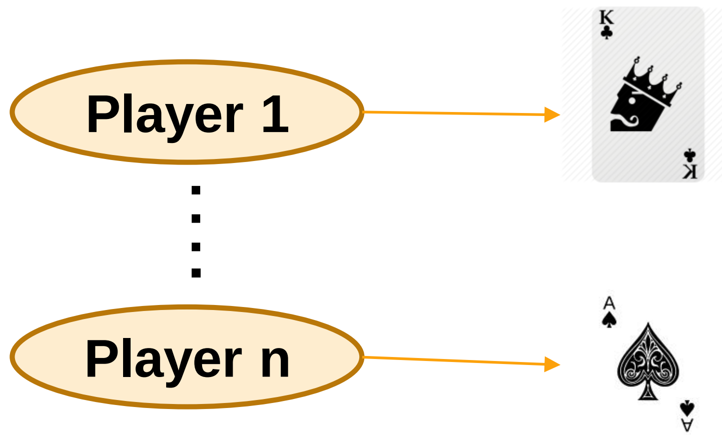
Multi-rounds of games



Player	Card	Start_time	End_time
P1	9♥,	10.00	10.02
P2	7♠	10.02	10.03

Project 1

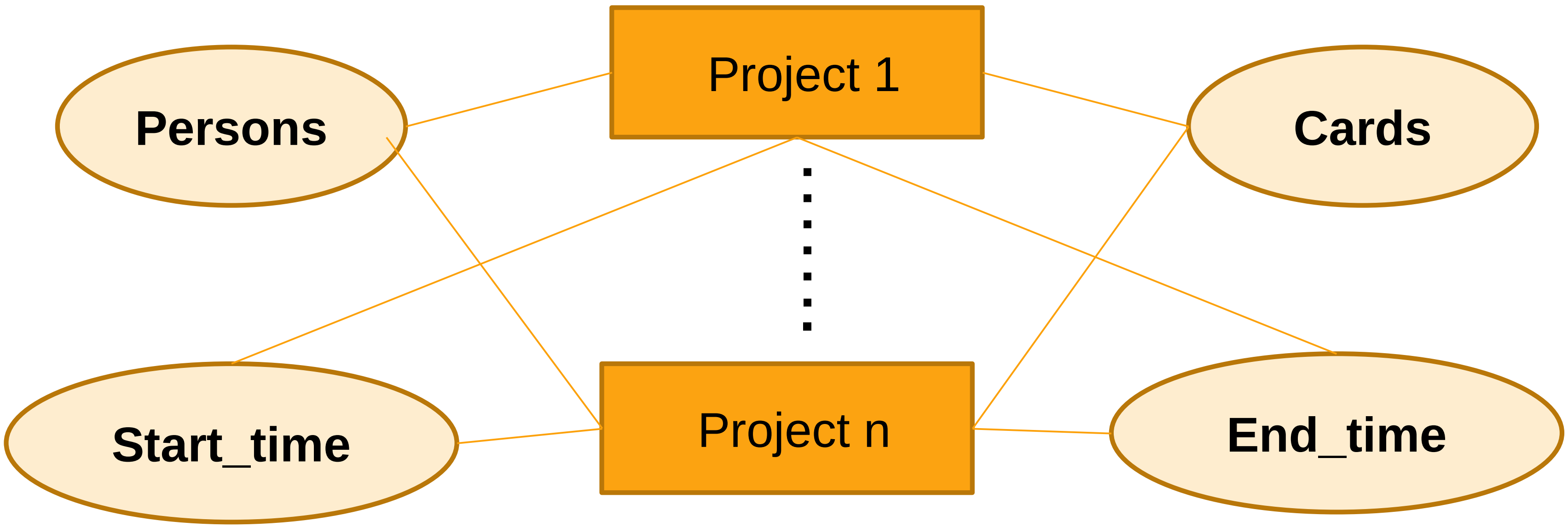
……



Player	Card	Start_time	End_time
P1	5♦	11.00	11.01
P2	A♥	11.02	11.03

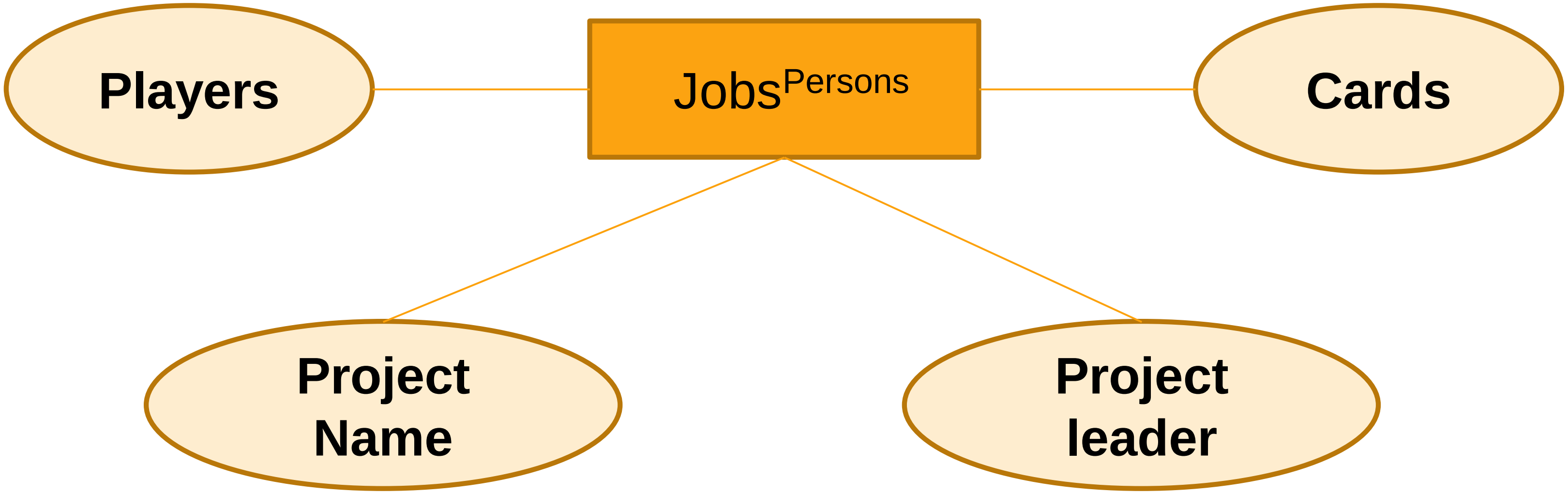
Round n

ER for Multi-rounds of games



Category model for Multi-rounds of games

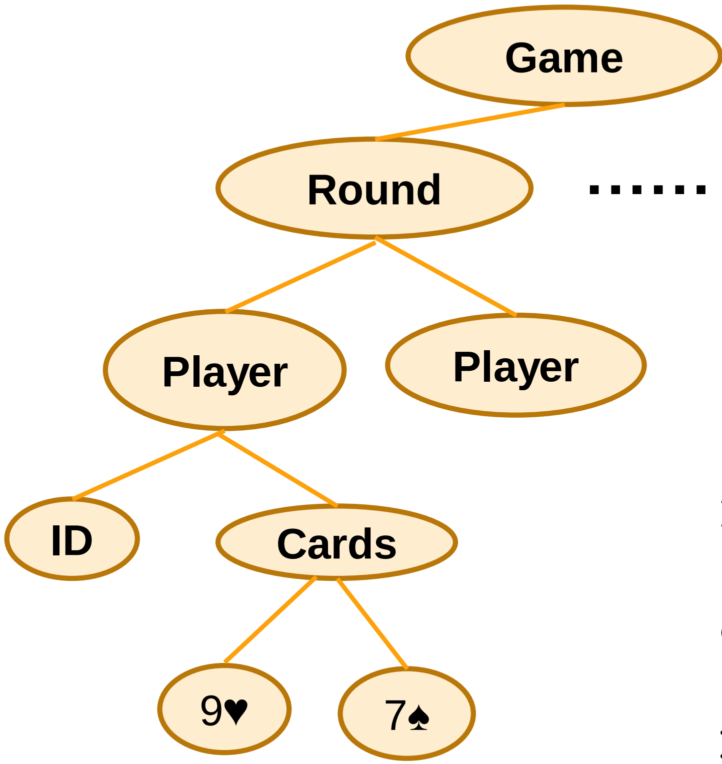
- Category model can define exponential object.



Complicated query with exponential object

- Categorical query can support some complicated queries with exponential object
- Find two projects where nobody is assigned the same job
- Let ev: $(\text{Cards}^{\text{Players}} \times \text{Players}) \rightarrow \text{Cards}$
- $\exists r1, \exists r2, \forall p,\ ev(r1,p) \neq ev(r2,p) \wedge E(r1) \wedge E(r2)$
- E stands for the exponential object $\text{Cards}^{\text{Players}}$
- (This is NOT a rigorous formal expression. We need to introduce Heyting algebra to give a formal definition)

XPath 3.1 with higher-order function and categorical query



∃r1, ∃r2, ∀p, f(r1,p) ≠ f (r2,p) ∧ E(r1) ∧ E(r2)

E stands for the exponential object $\mathrm{Cards}^{\mathrm{Players}}$

```
Let $b := TRUE
for-each-pair(game/round/player/cards,
game/round/player/cards, fn($a,$b)
(if ($a== $b} then let  $b := FLASE)
Return $b
```

Cartesian closed category

- The category C is called Cartesian closed if and only if it satisfies the following properties:

- Any two objects X and Y of C have a product X×Y in C.
- Any two objects Y and Z of C have an exponential Z^Y in C.

- A multi-model database can be modeled as a Cartesian closed category.

Query category with higher order logic

- SQL query is the first-order logic query language. Sql can be extended

- Based on Cartesian closed category (and topos theory), we can define higher-order logic query for XPath 3.1 and reachability graph query.

- Categorical framework build a unified query language

- More powerful, but more expensive to compute

Limitation of this category framework

- Provide a unified framework to incorporate different data models together
- But no hints for efficient algorithms
- Many problems are NP-hard, EXPTIME or even undecidable
- Efficient implementation and tractable subclasses for the future work

Related systems and work

- Multidatabase systems(or federated systems)
  - a few databases (less than 10)
  - Powerful queries (transaction and updates)
- Object Relation database systems
  - supports extension of the data model with custom data types and methods
  - Relation model is the first-class citizen, and ad-hoc support for other models of data
- Polyglot persistence
  - Integrated access to multiple, heterogeneous cloud data stores, such as NoSQL, HDFS, and RDBMS

Related systems and work (con’t)

- Category theory and databases
  - David I. Spivak propose: category = database schema
  - The data is a functor: $C \rightarrow Set$
  - But our work focuses on multi-model data and the unified query language

Conclusion

- Category theory can serve as a theoretical framework for multi-model databases
- Powerful high-order logic query interface

Conclusion (Cont.)

- We will demo how to use a functional programming language ***Haskell*** to query multi-model data

- Category theory is a theoretical foundation for functional programming.

Acknowledgement

- Thanks for the review and feedback from Dieter Gawlick, Zhen Hua Liu, Souripriya Das, Heli Helskyaho and Greg Pogossiants

- This research is partially funded by Academy of Finland and Oracle gift award.

ORACLE

SUOMEN AKATEMIA
FINLANDS AKADEMI
ACADEMY OF FINLAND

References

1) Zhen Hua Liu , Jiaheng Lu, Dieter Gawlick, Heli Helskyaho, Gregory Pogossiants,   Zhe Wu: Multi-Model Database Management Systems (MMDBMS) - a Look Forward (Position paper) VLDB workshop 2018, Poly

2) Jiaheng Lu, Irena Holubová, Bogdan Cautis: Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges. CIKM 2018: 2301-2302 (Tutorial)

3) Jiaheng Lu, Irena Holubová: Multi-model Data Management: What's New and What's Next? EDBT 2017:   602-605 31-51 (2012)

References

4) Chao Zhang, Jiaheng Lu, Pengfei Xu, Yuxing Chen: UniBench: A Benchmark for Multi-model Database Management Systems. TPCTC 2018: 7-23 (Benchmark system work)
5) Mac Lane, Saunders. Categories for the working mathematician. Vol. 5. Springer Science & Business Media, 2013.
6) David I. Spivak. Category theory for the sciences[M]. MIT Press, 2014.
7) David I. Spivak: Functorial data migration. Information and. Computation. 217