It's not Just Tables Anymore

Building Reliable Knowledge Graphs

Maximilian K. Egger
October 23, 2025

Why Graphs Matter

What if your data could consider relationships?

Why Graphs Matter

What if your data could consider relationships?

WIKIPEDIA

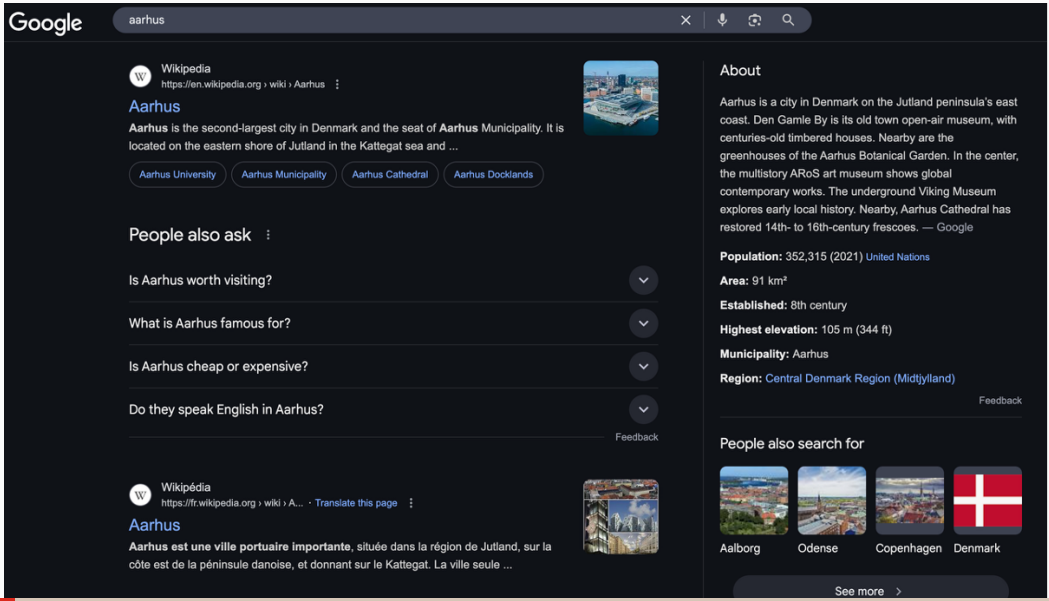
Why Graphs Matter

What if your data could consider relationships?

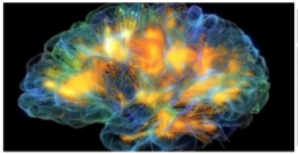
WIKIPEDIA

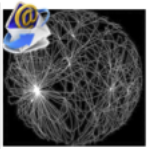
Google

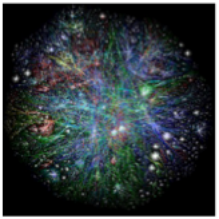
Example: Google Search



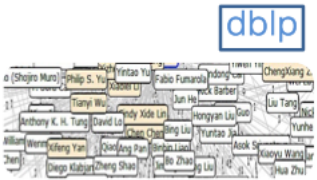
Examples

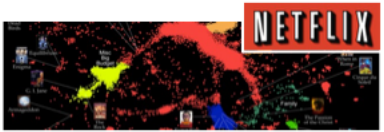







You
Tube


dblp
Philip S. Yu
Yintao Yu
Fabio Fumarola
Jun He
Liu Tang
Anthony K. H. Tung
David Lo
Hongyan Liu
Bing Liu
Yuntao Jia
Kifeng Yan
Diego Klabjan
Zheng Shao
Bo Zhao
Asok S
Kaoyu Wang
Hua Zhu


NETFLIX

Relational Databases

Relational Databases: Structured and Familiar

- Data stored in tables
- Schema provides structure
- Entries identifiable with primary keys
- Relationships handled by foreign keys and joins

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
W. Mozart	35	Salzburg	Composer
S. Freud	83	Vienna	Neurologist
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

Rigid Schemas

- Columns defined upfront
- All rows must conform
- New attribute requires schema edit

Person

Name	Age	Address	Occupation	CPR
M. Egger	28	Aarhus	PhD student	REDACTED
W. Mozart	35	Salzburg	Composer	NULL
S. Freud	83	Vienna	Neurologist	NULL
⋮	⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer	NULL
C. Waltz	69	Los Angeles	Actor	NULL

Linking Data

- Relationships stored in separate tables
- Joins needed
- Complex with multiple relations

Compositions

Title	Year	Type
Magic Flute	1791	Opera
Blue Danube	1866	Waltz
Die Fledermaus	1874	Operetta
⋮	⋮	⋮

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
W. Mozart	35	Salzburg	Composer
S. Freud	83	Vienna	Neurologist
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

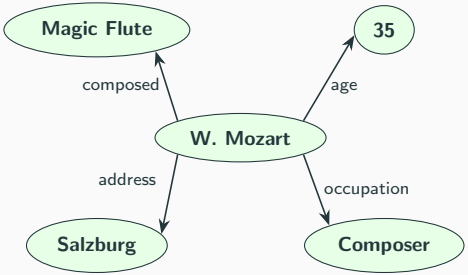
ComposedBy

Composer	Title

Introducing Graphs

Knowledge Graphs

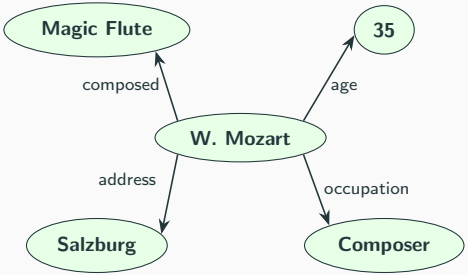
- Nodes represent entities
- Edges are relationships
- Based on RDF
  (Resource Description Framework)

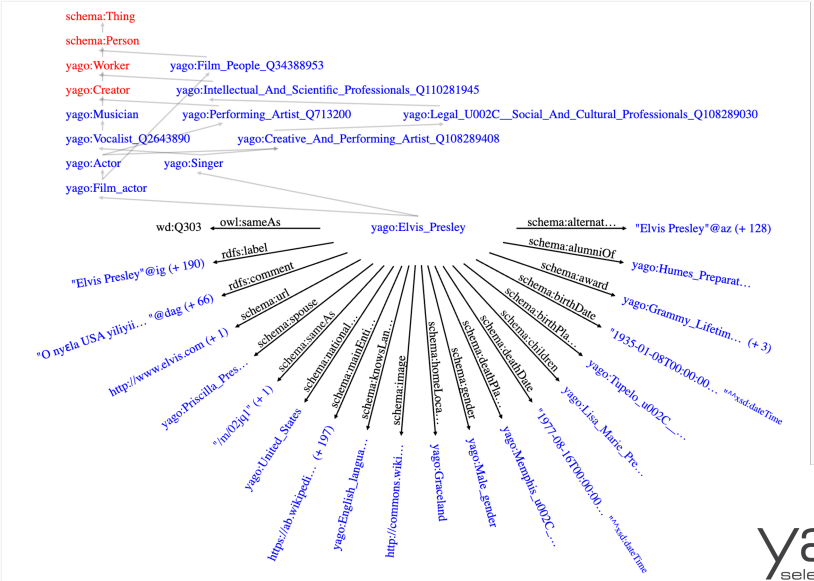


Knowledge Graphs

- Nodes represent entities
- Edges are relationships
- Based on RDF
  (Resource Description Framework)

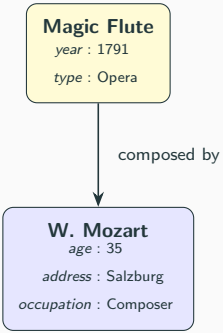
W. Mozart, composed, Magic Flute;
W. Mozart, address, Salzburg;
W. Mozart, age, 35;
W. Mozart, occupation, Composer;




schema:Thing
schema:Person
yago:Worker
yago:Film_People_Q34388953
yago:Creator
yago:Intellectual_And_Scientific_Professionals_Q110281945
yago:Musician
yago:Performing_Artist_Q713200
yago:Legal_U002C__Social_And_Cultural_Professionals_Q108289030
yago:Vocalist_Q2643890
yago:Creative_And_Performing_Artist_Q108289408
yago:Actor
yago:Singer
yago:Film_actor
wd:Q303
owl:sameAs
yago:Elvis_Presley
schema:alternat…
"Elvis Presley"@az (+ 128)
schema:alumniOf
yago:Humes_Preparat…
schema:award
yago:Grammy_Lifetim… (+ 3)
schema:birthDate
"1935-01-08T00:00:00… "^^xsd:dateTime
schema:birthPla…
yago:Tupelo_u002C_…
schema:children
yago:Lisa_Marie_Pre…
schema:deathDate
"1977-08-16T00:00:00… "^^xsd:dateTime
schema:deathPla…
yago:Memphis_u002C_…
schema:gender
yago:Male_gender
schema:homeLoca…
yago:Graceland
schema:image
http://commons.wiki…
schema:knowsLan…
yago:English_langua…
schema:mainEnti…
https://ab.wikipedi… (+ 197)
schema:nationali…
yago:United_States
schema:sameAs
"/m/02jq1" (+ 1)
schema:spouse
yago:Priscilla_Pres…
schema:url
http://www.elvis.com (+ 1)
rdfs:comment
"O uyela USA yüliyii… "@dag (+ 66)
rdfs:label
"Elvis Presley"@ig (+ 190)

Property Graphs

- Nodes and edges can have properties
- Allows for complex data
- Representation compact

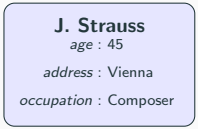


Transforming Tables

Name	Age	Address	Occupation
W. Mozart	35	Salzburg	Composer
J. Strauss	45	Vienna	Composer

Transforming Tables

Name	Age	Address	Occupation
W. Mozart	35	Salzburg	Composer
J. Strauss	45	Vienna	Composer



Transforming Tables

Compositions

Title	Year	Type

W. Mozart
age : 35
address : Salzburg
occupation : Composer

J. Strauss
age : 45
address : Vienna
occupation : Composer

ComposedBy

Composer	Title

Magic Flute
year : 1791
type : Opera

Die Fledermaus
year : 1874
type : Operetta

Blue Danube
year : 1866
type : Waltz

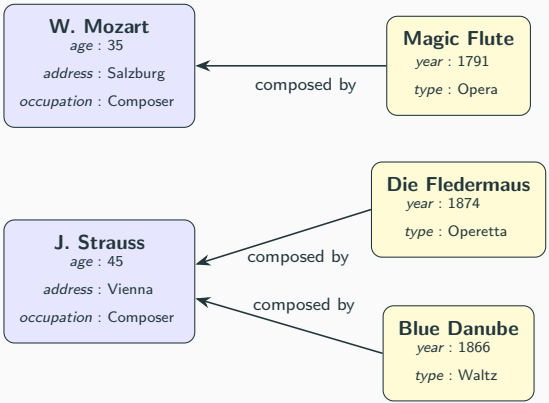
Transforming Tables

Compositions

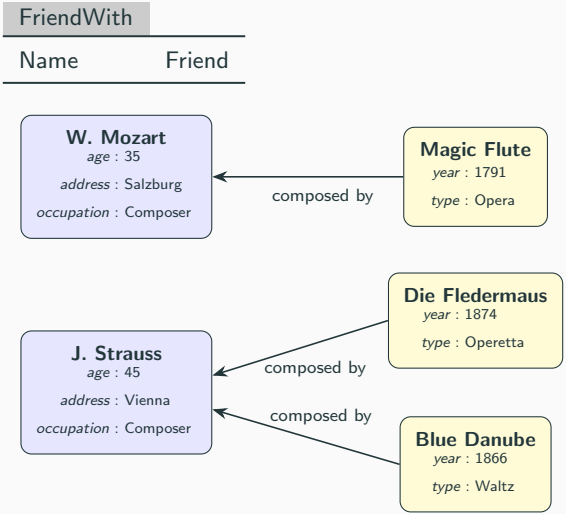
Title	Year	Type

ComposedBy

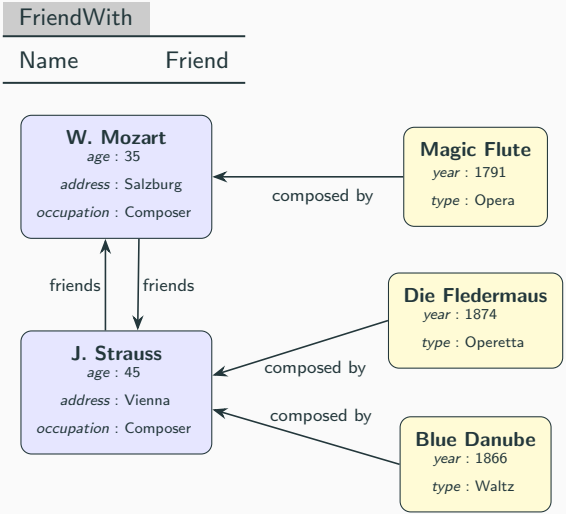
Composer	Title



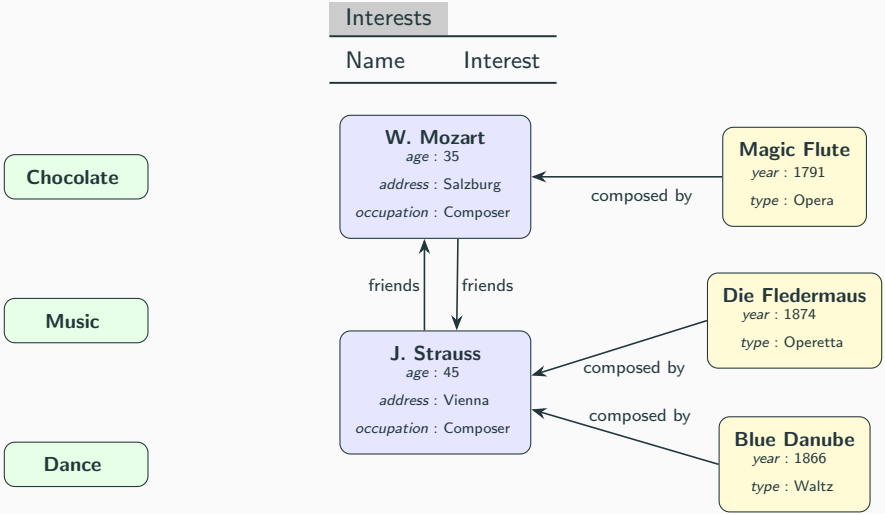
Transforming Tables



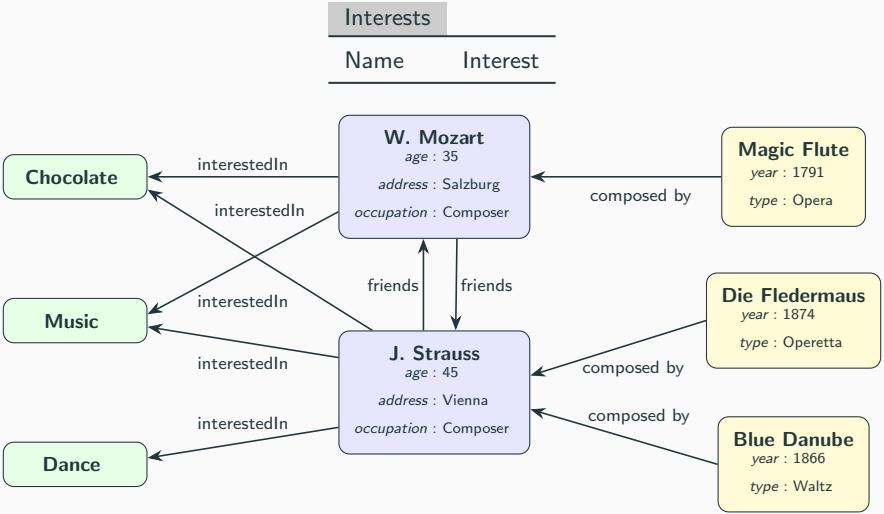
Transforming Tables



Transforming Tables



Transforming Tables



Graph Schema

Graph Schema

- no schema (or implicit)
  - no rules
  - data describes itself

Graph Schema

- no schema (or implicit)
  - no rules
  - data describes itself
- lightweight schema
  - define node and edge types
  - some common properties

Graph Schema

- no schema (or implicit)
  - no rules
  - data describes itself
- lightweight schema
  - define node and edge types
  - some common properties
- strong schema (ontology)
  - strong types
  - attributes are well defined

Schema Considerations

- node types

Schema Considerations

- node types
- edge types
  - names
  - directionality
  - semantics
  - cardinality expections

Schema Considerations

- node types
- edge types
  - names
  - directionality
  - semantics
  - cardinality expections
- property definitions
  - allowed properties for node/edge type
  - data types
  - cardinality
  - required or optional

Schema Considerations

- node types
- edge types
  - names
  - directionality
  - semantics
  - cardinality expections
- property definitions
  - allowed properties for node/edge type
  - data types
  - cardinality
  - required or optional
- keys and identifiers

Schema Considerations

- node types
- edge types
  - names
  - directionality
  - semantics
  - cardinality expections
- property definitions
  - allowed properties for node/edge type
  - data types
  - cardinality
  - required or optional
- keys and identifiers
- domain constraints

Schema Validation

Schema Validation

- cypher

```
CREATE CONSTRAINT ON (p:Person) ASSERT p.globalId IS UNIQUE;
CREATE CONSTRAINT ON (c:Composition) ASSERT exists(c.title);
```

Schema Validation

- cypher

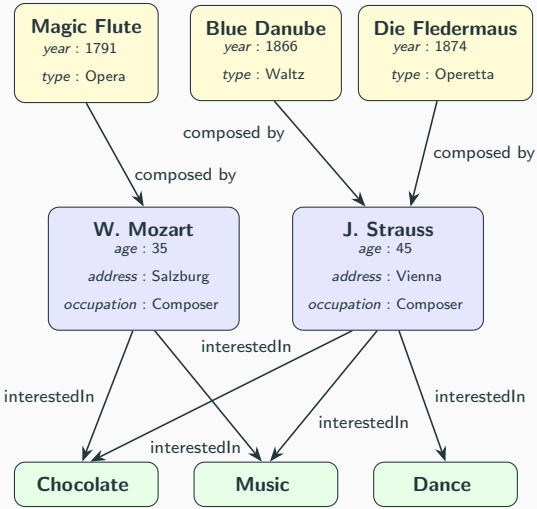
```
CREATE CONSTRAINT ON (p:Person) ASSERT p.globalId IS UNIQUE;
CREATE CONSTRAINT ON (c:Composition) ASSERT exists(c.title);
```

- SHACL

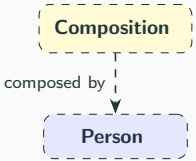
```
:PersonShape a sh:NodeShape ;
sh:targetClass :Person ;
sh:property [ sh:path :birthYear ; sh:datatype xsd:integer ; sh:minCount 0 ;
sh:maxCount 1 ] .
```

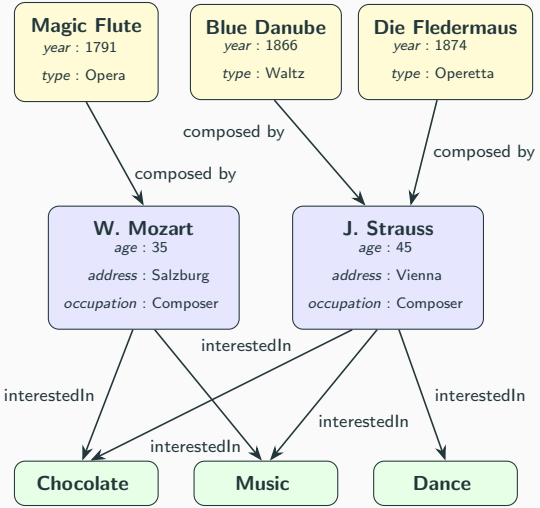
Tools

Graph Pattern

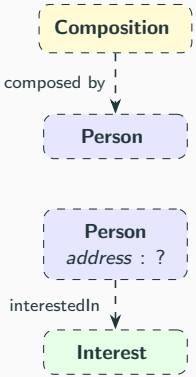


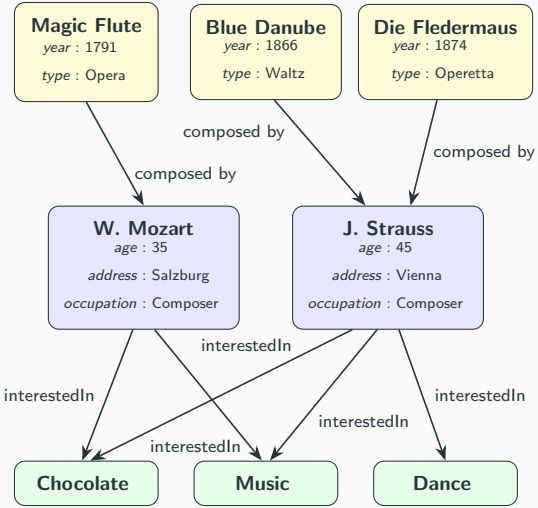
Graph Pattern



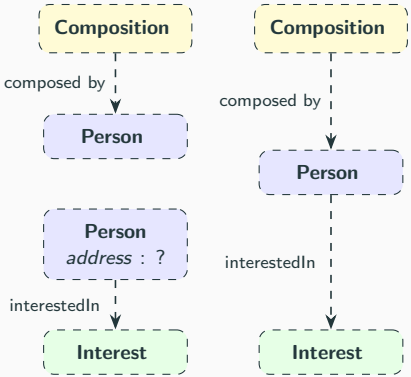


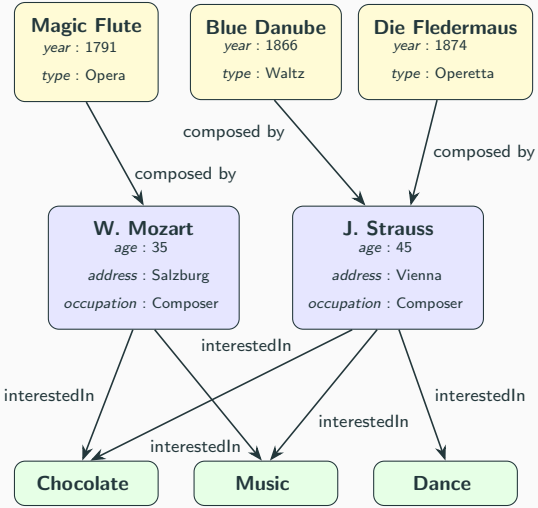
Graph Pattern



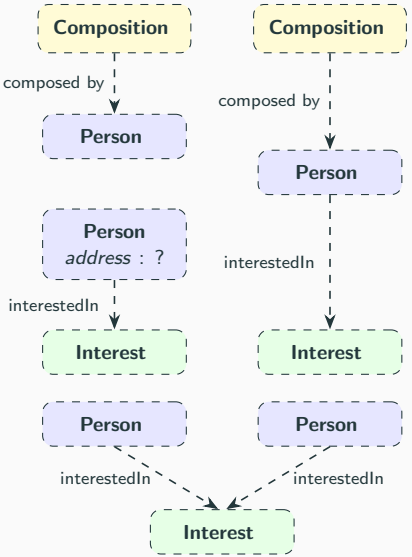


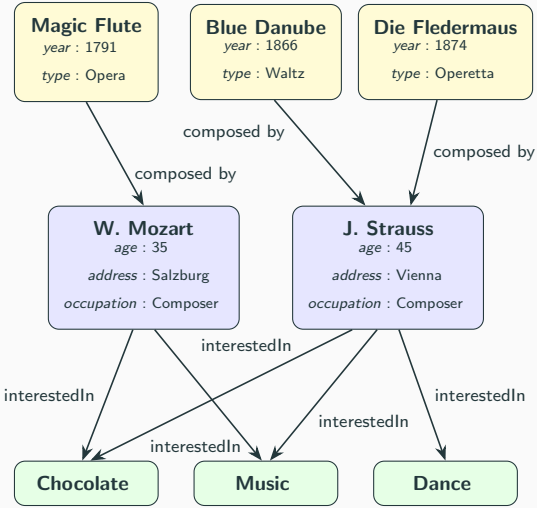
Graph Pattern





Graph Pattern





Queries

Give me all release years of music compositions by artists from a specific city?

Queries

Give me all release years of music compositions by artists from a specific city?

Person

Name	Age	Address	Occupation

Compositions

Title	Year	Type

ComposedBy

Composer	Title

Queries

Give me all release years of music compositions by artists from a specific city?

```
SELECT c.year
FROM Person p
JOIN ComposedBy cb ON p.name = cb.composer
JOIN Compositions c ON cb.title = c.title
WHERE p.city = 'Vienna';
```

Person

Name	Age	Address	Occupation

Compositions

Title	Year	Type

ComposedBy

Composer	Title

Queries

Give me all release years of music compositions by artists from a specific city?

```
SELECT c.year
FROM Person p
JOIN ComposedBy cb ON p.name = cb.composer
JOIN Compositions c ON cb.title = c.title
WHERE p.city = 'Vienna';
```

Person

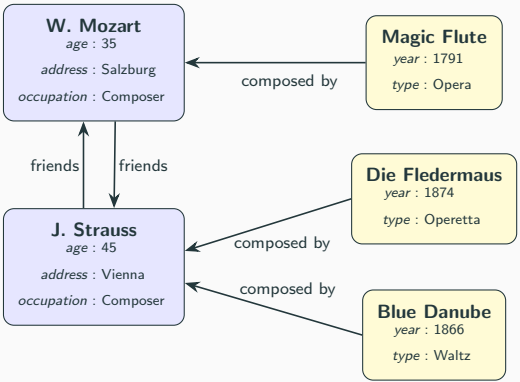
Name	Age	Address	Occupation

Compositions

Title	Year	Type

ComposedBy

Composer	Title



Queries

Give me all release years of music compositions by artists from a specific city?

```
SELECT c.year
FROM Person p
JOIN ComposedBy cb ON p.name = cb.composer
JOIN Compositions c ON cb.title = c.title
WHERE p.city = 'Vienna';
```

Person

Name	Age	Address	Occupation

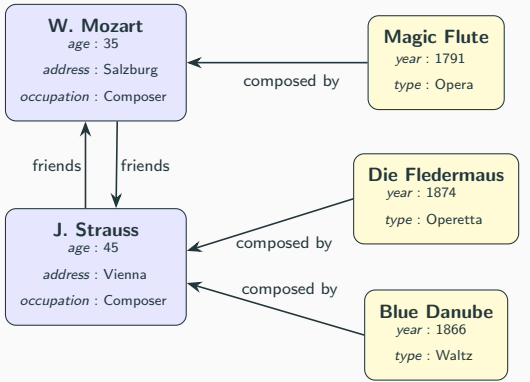
Compositions

Title	Year	Type

ComposedBy

Composer	Title

```
MATCH (c:Composition)-[:composed by]->(p:Person)
WHERE p.city = 'Vienna'
RETURN c.year
```



Queries

Give me all release years of music compositions by artists from a specific city?

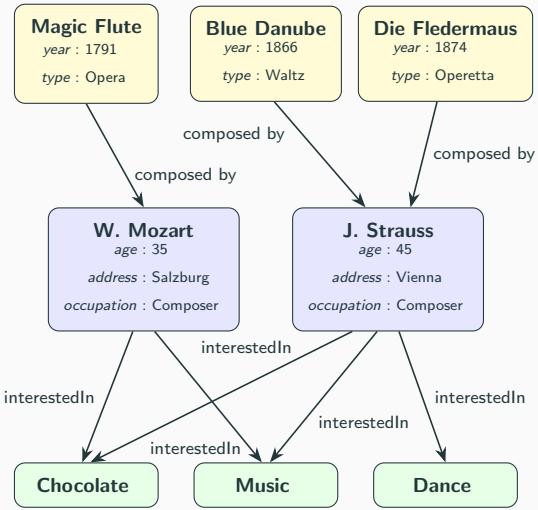
```
SELECT c.year
FROM Person p
JOIN ComposedBy cb ON p.name = cb.composer
JOIN Compositions c ON cb.title = c.title
WHERE p.city = 'Vienna';
```

```
MATCH (c:Composition)-[:composed by]->(p:Person)
WHERE p.city = 'Vienna'
RETURN c.year
```

Feature	Relational (SQL)	Graph (Cypher)
Joins required	Yes	No
Schema	Yes	Optional
Readability	Moderate	High

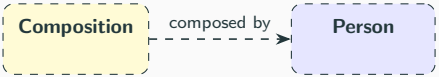
Functional Dependencies

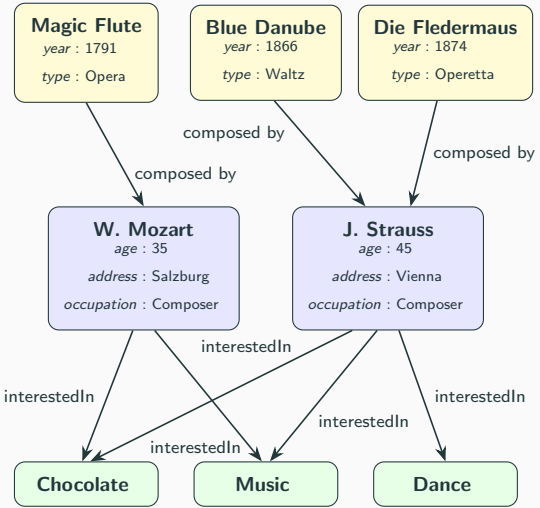
How can we validate certain dependencies between data that we know about?



Functional Dependencies

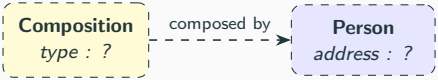
How can we validate certain dependencies between data that we know about?

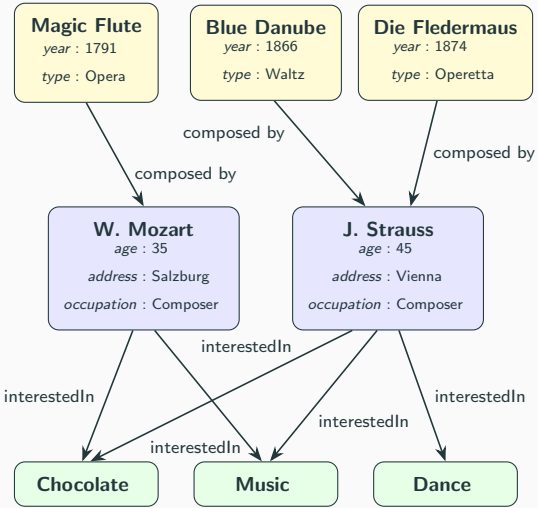




Functional Dependencies

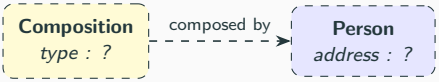
How can we validate certain dependencies between data that we know about?



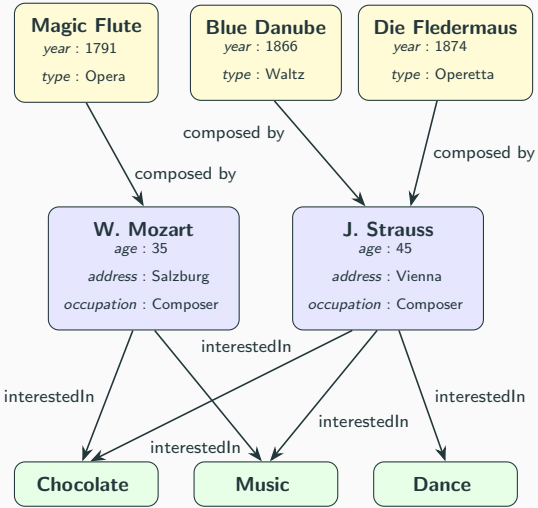


Functional Dependencies

How can we validate certain dependencies between data that we know about?

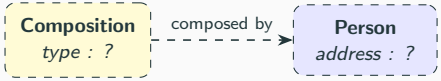


$c.type \Rightarrow p.address$

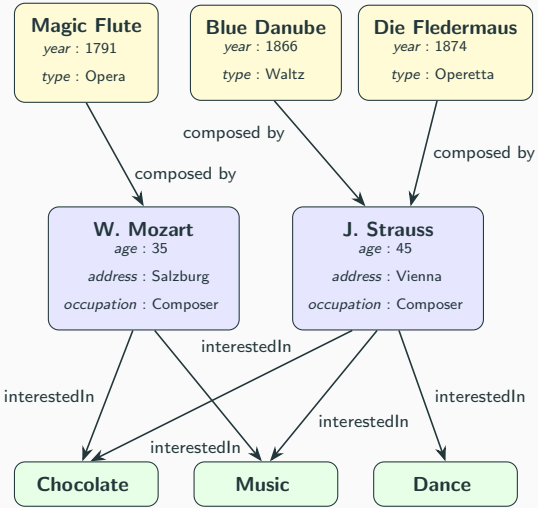


Functional Dependencies

How can we validate certain dependencies between data that we know about?

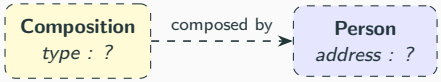


$c.type \Rightarrow p.address$



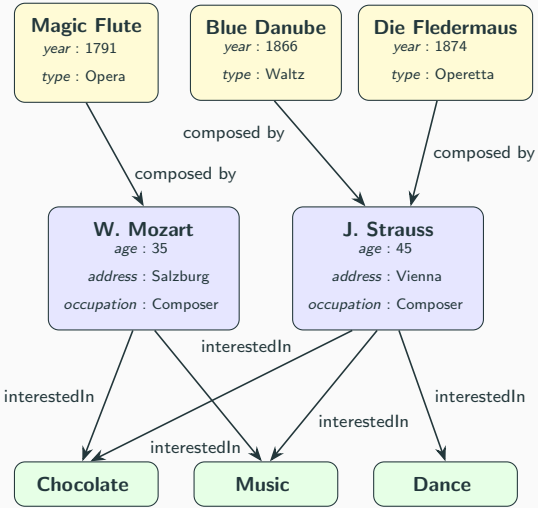
Functional Dependencies

How can we validate certain dependencies between data that we know about?



$c.type \Rightarrow p.address$

$\emptyset \Rightarrow p.occupation$

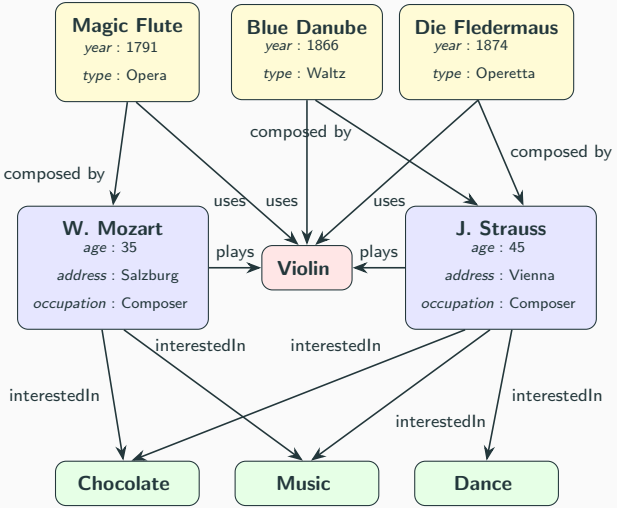


Rules

If condition is met does the result exist?

Rules

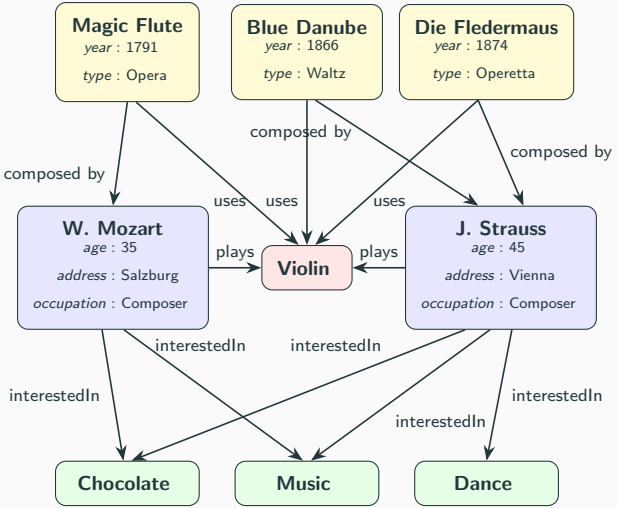
If condition is met does the result exist?



Rules

If **body** is met does the result exist?

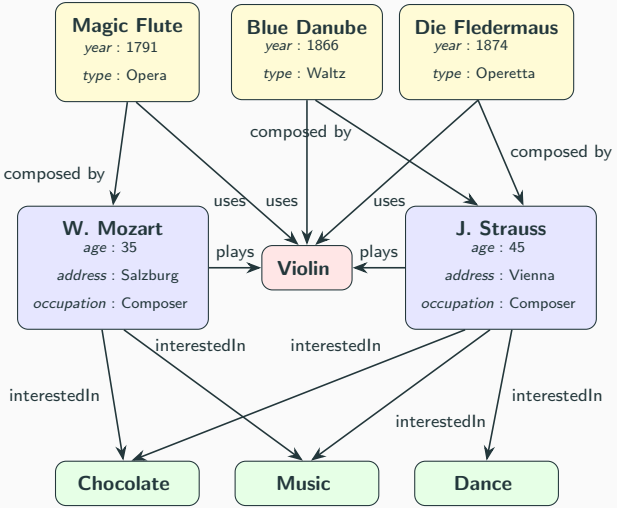
Body



Rules

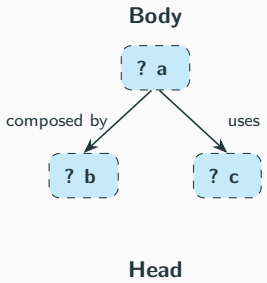
If **body** is met does the **head** exist?

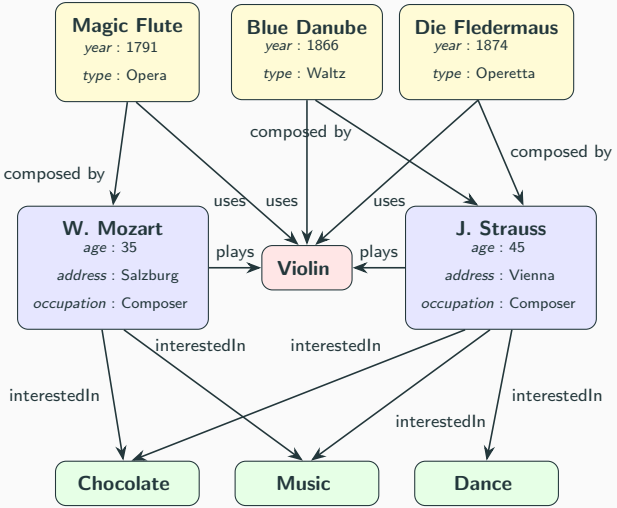
Body



Rules

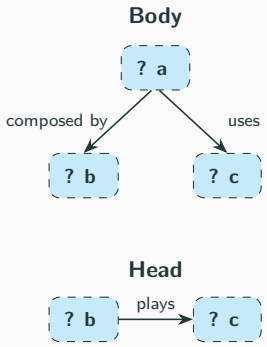
If **body** is met does the **head** exist?

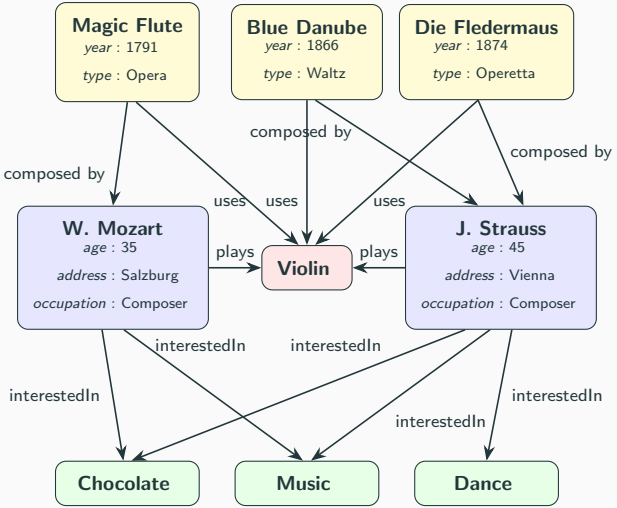




Rules

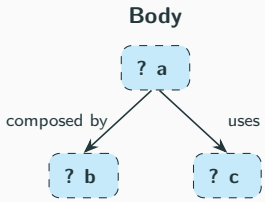
If **body** is met does the **head** exist?



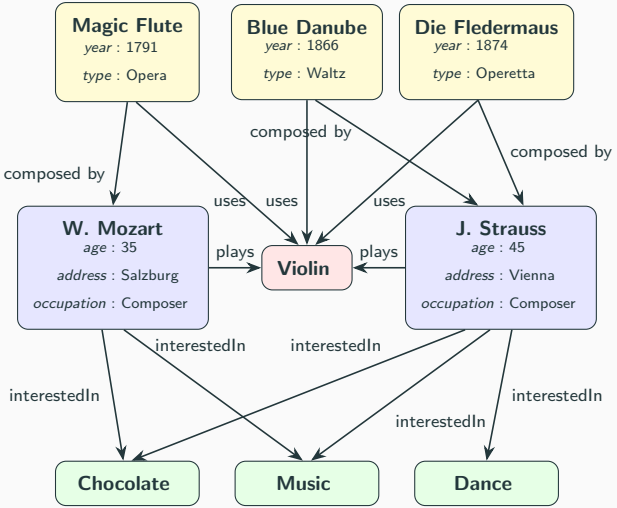


Rules

If **body** is met does the **head** exist?



$$\frac{\#(body \wedge head)}{\#body}$$



Rule-Mining

How does one find rules?

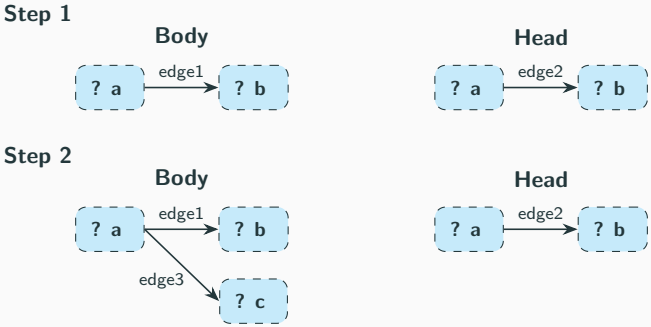
Rule-Mining

How does one find rules?

Step 1

Rule-Mining

How does one find rules?



Rule-Mining

How does one find rules?

Step 1

Step 2

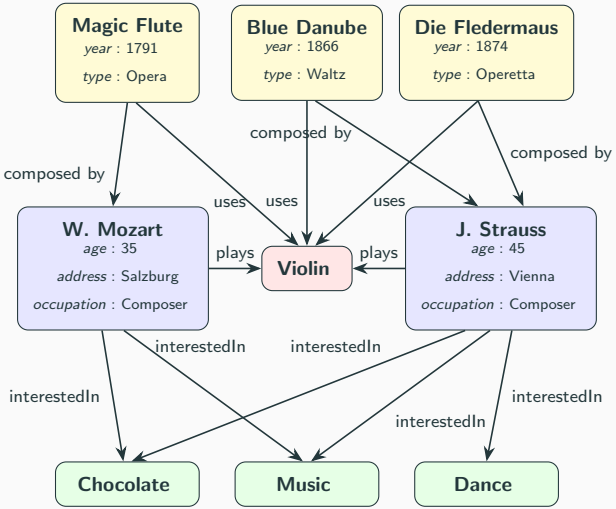
Step 3: Repeat

Data Driven; bad data leads to bad rules!

Embeddings

How to use graphs in numerical space?

$$\begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,d} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \cdots & e_{n,d} \end{bmatrix}$$

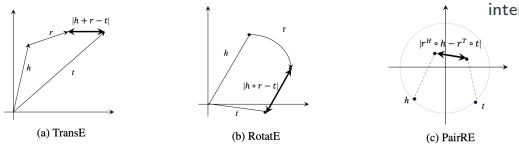


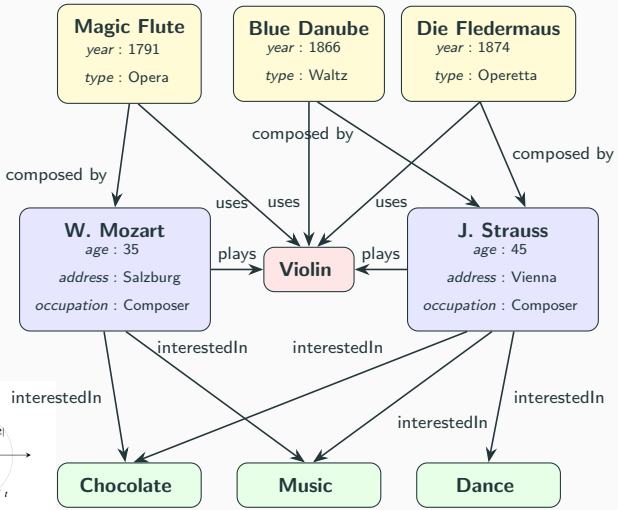
Embeddings

How to use graphs in numerical space?

$$\begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,d} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n,1} & e_{n,2} & \cdots & e_{n,d} \end{bmatrix}$$

- TransE
- ComplEx
- RotatE
- ...





Applications

Compliance

What if someone wants all their information removed?

Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

ActedIn

Actor	Movie	Role

ComposedBy

Composer	Title

FriendWith

Name	Friend
⋮	

Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

ActedIn

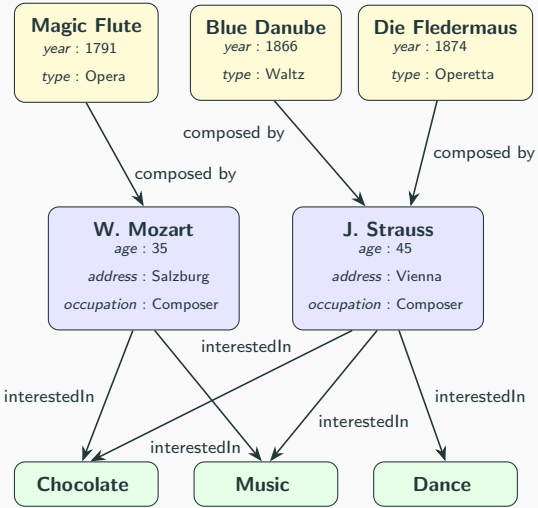
Actor	Movie	Role

ComposedBy

Composer	Title

FriendWith

Name	Friend
⋮	



Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

ActedIn

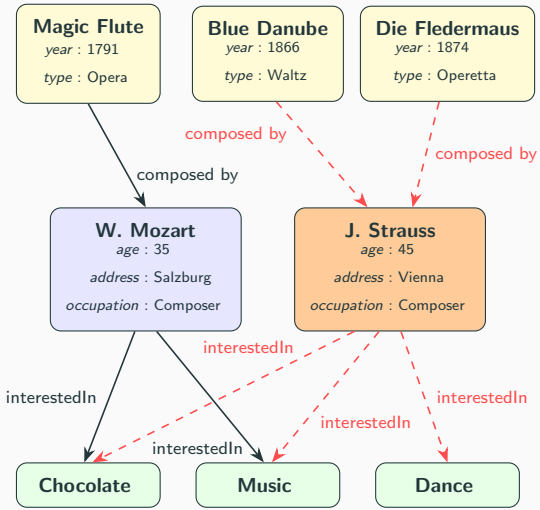
Actor	Movie	Role

ComposedBy

Composer	Title

FriendWith

Name	Friend
⋮	



Compliance

What if someone wants all their information removed?

Person

Name	Age	Address	Occupation
M. Egger	28	Aarhus	PhD student
⋮	⋮	⋮	⋮
J. Strauss	45	Vienna	Composer
C. Waltz	69	Los Angeles	Actor

ActedIn

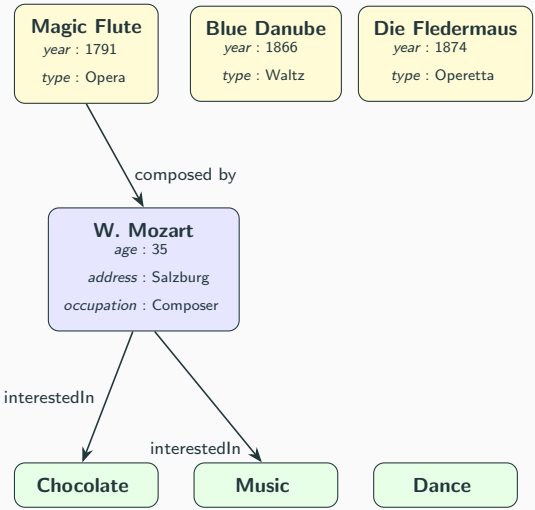
Actor	Movie	Role

ComposedBy

Composer	Title

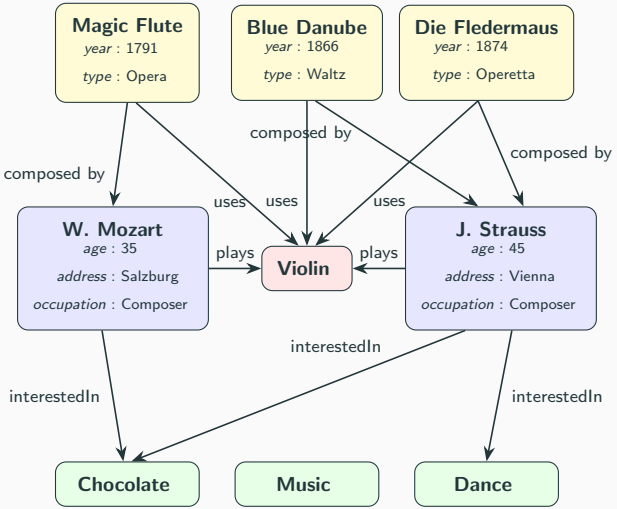
FriendWith

Name	Friend
⋮	



Inference

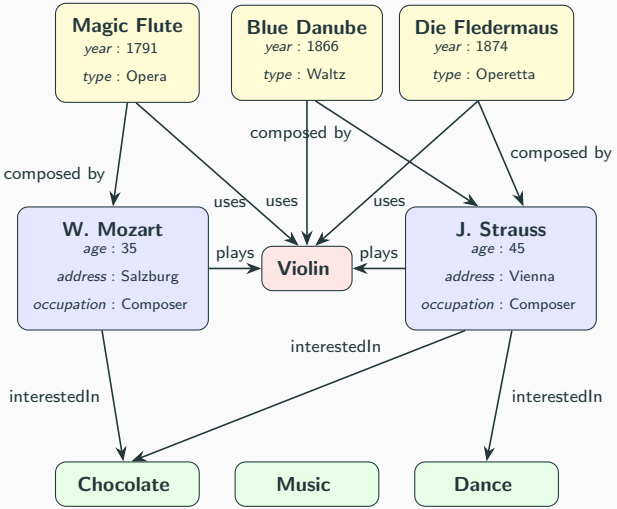
Is there a way to fill in missing information?



Inference

Is there a way to fill in missing information?

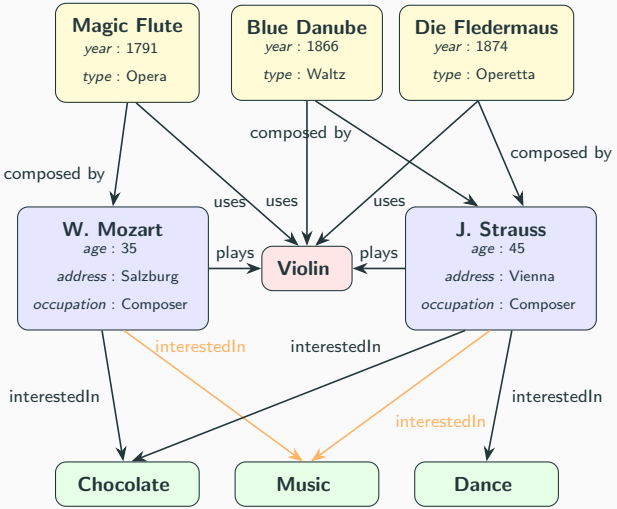
- Rules
- Functional Dependencies
- Embeddings



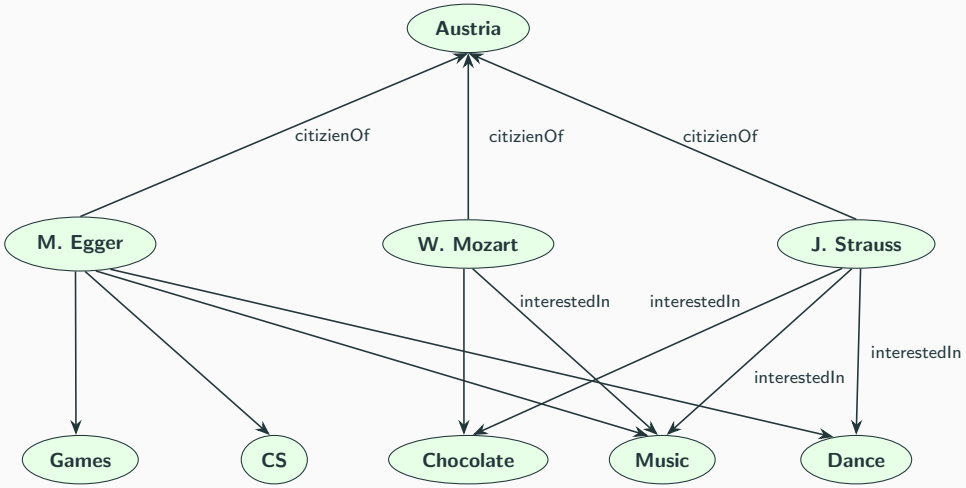
Inference

Is there a way to fill in missing information?

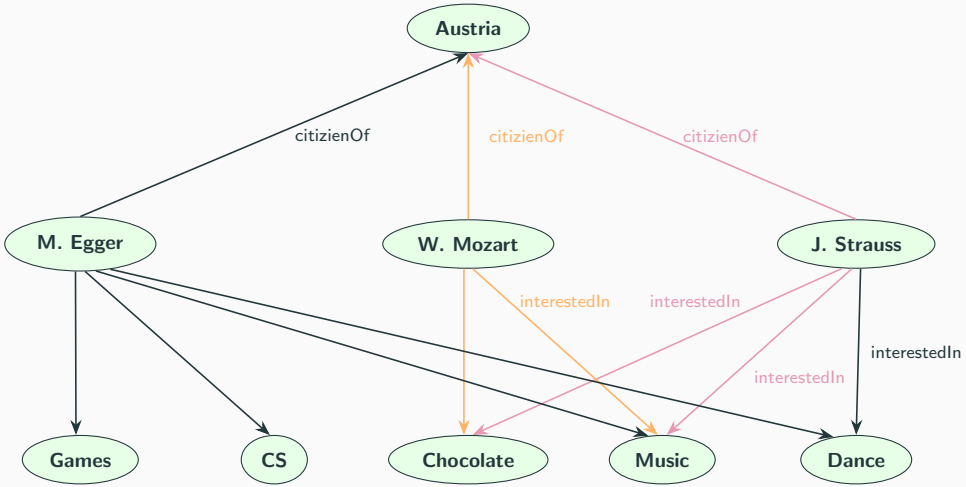
- Rules
- Functional Dependencies
- Embeddings



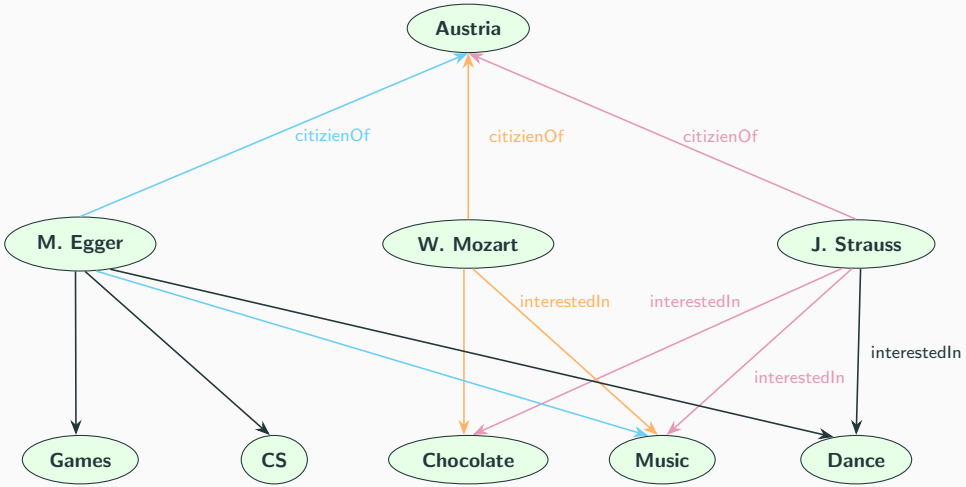
How to recommend someone new interests?



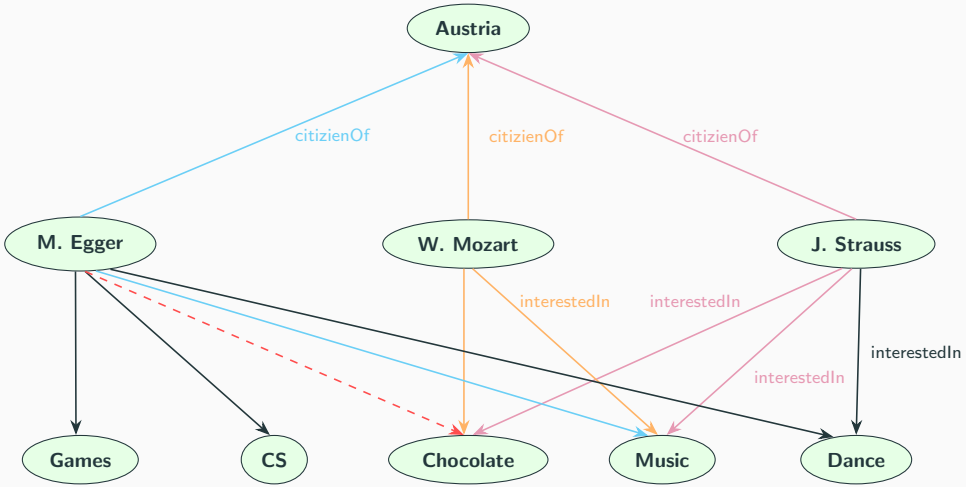
How to recommend someone new interests?



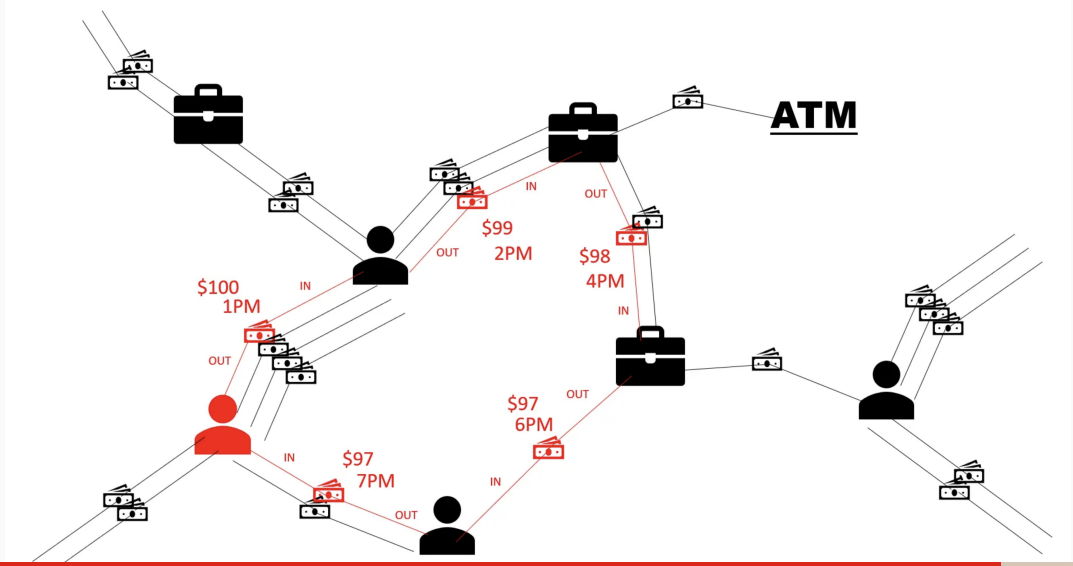
How to recommend someone new interests?



How to recommend someone new interests?



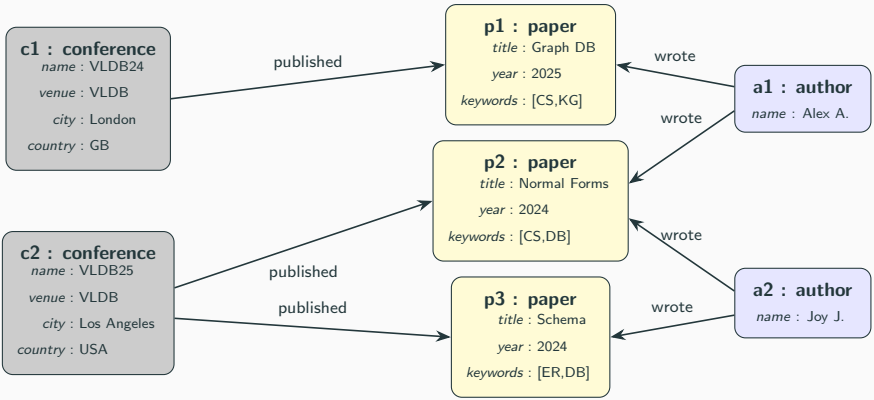
Money Laundering Detection



Research

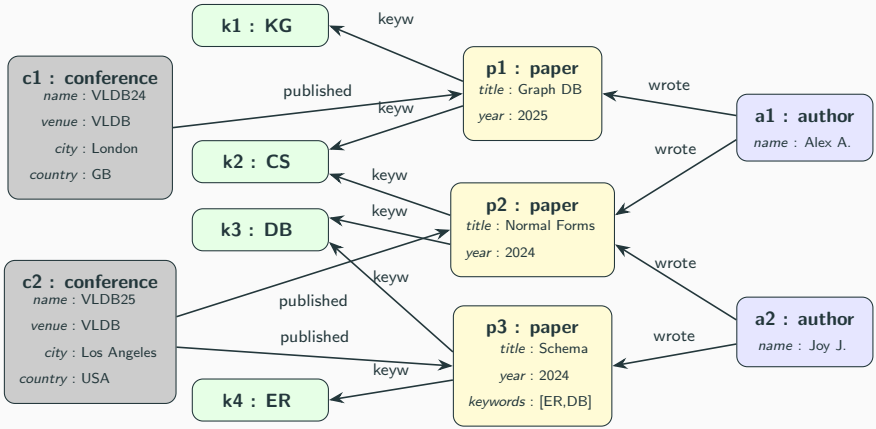
Normalization

In relational model **normalization** reduces redundancy, can we do the same in graphs?



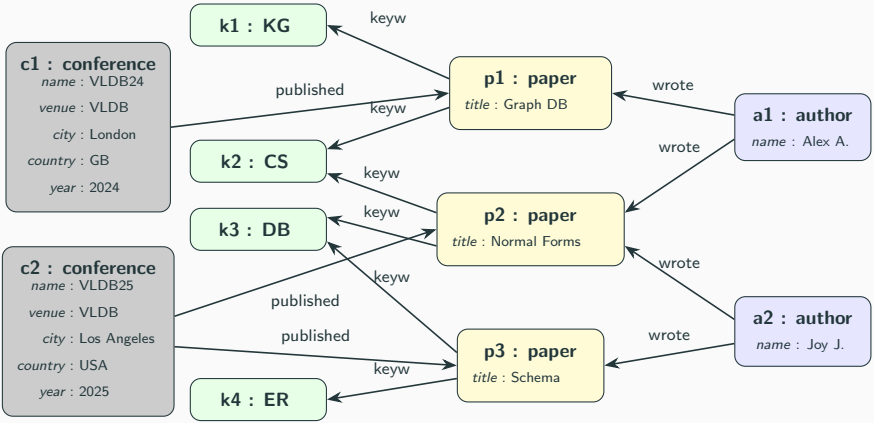
Normalization

In relational model **normalization** reduces redundancy, can we do the same in graphs?



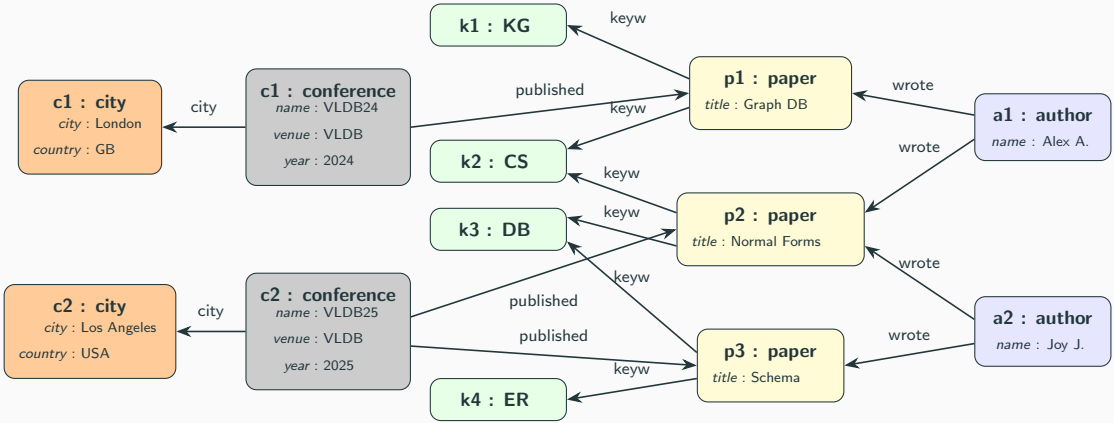
Normalization

In relational model **normalization** reduces redundancy, can we do the same in graphs?



Normalization

In relational model **normalization** reduces redundancy, can we do the same in graphs?



Summary

We learned about:

- limits of relational databases
- how to construct knowledge and property graphs
- the differences
- different tools in the graph world

Questions?