Graphs and graph databases

Introduction to graph databases

NoSQL: Lecture 1, part 1

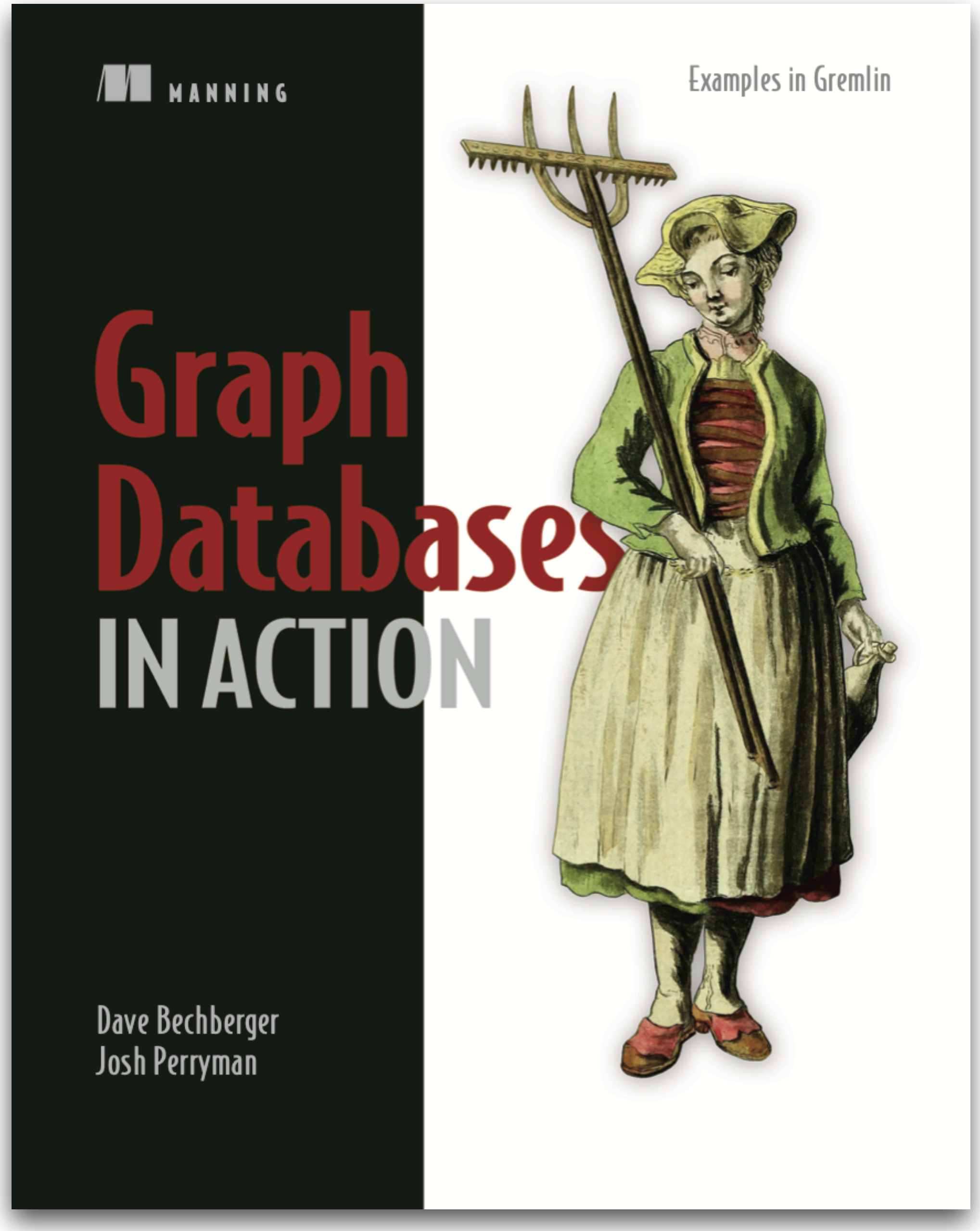
Piotr Fulmański

FACULTY OF MATHEMATICS
AND COMPUTER SCIENCE
University of Lodz

Graph databases In Action

by Dave Bechberger and Josh Perryman

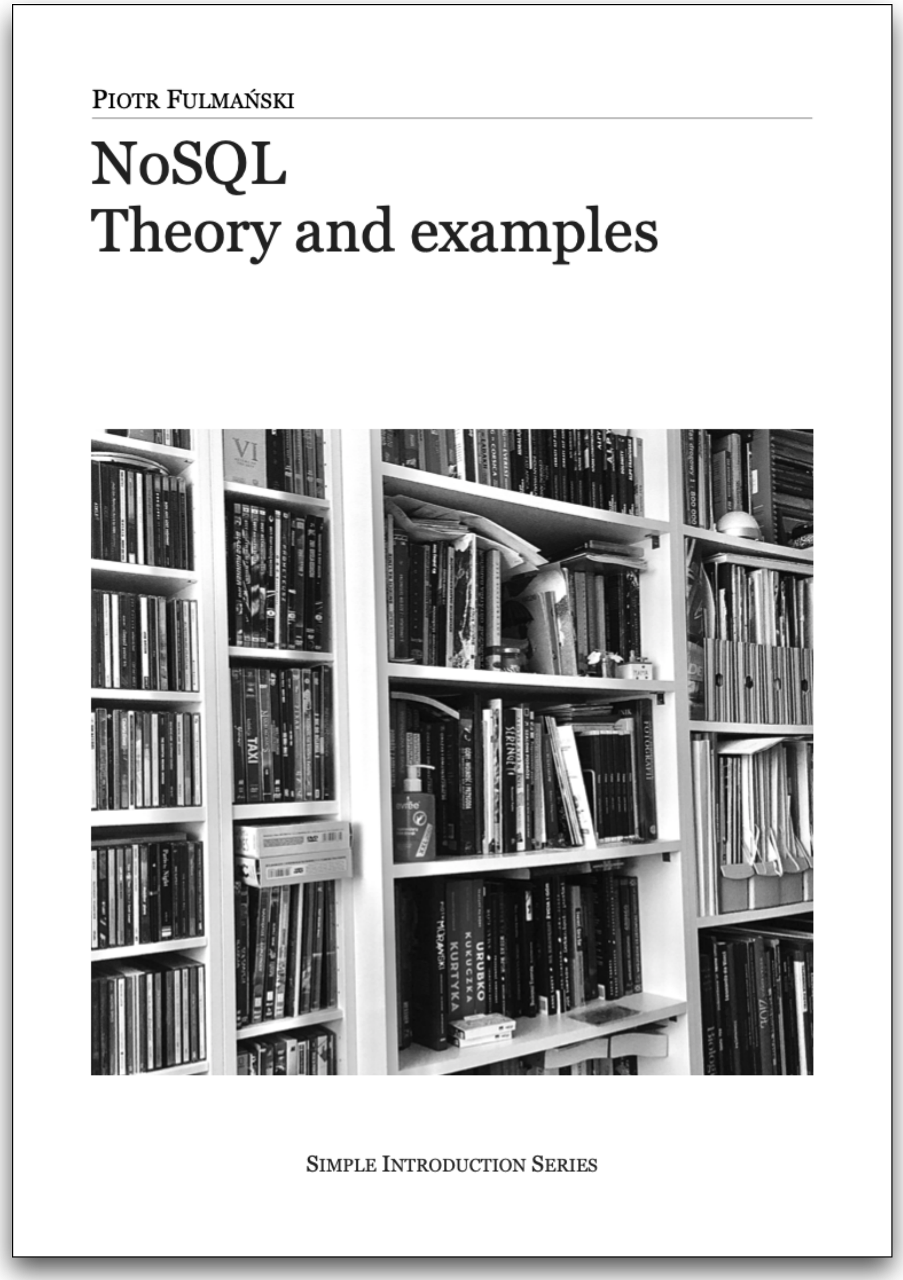
Manning Publications, 2020



NoSQL Theory and examples

by Piotr Fulmański

Piotr Fulmański, 2021



PIOTR FULMAŃSKI

NoSQL
Theory and examples

SIMPLE INTRODUCTION SERIES

Graphs and graph terminology

Graphs and graph terminology

- Vertices (singular: vertex) a.k.a. nodes.

Graphs and graph terminology

- Vertices (singular: vertex) a.k.a. nodes.
- Edges a.k.a. relationships, links, or connections.

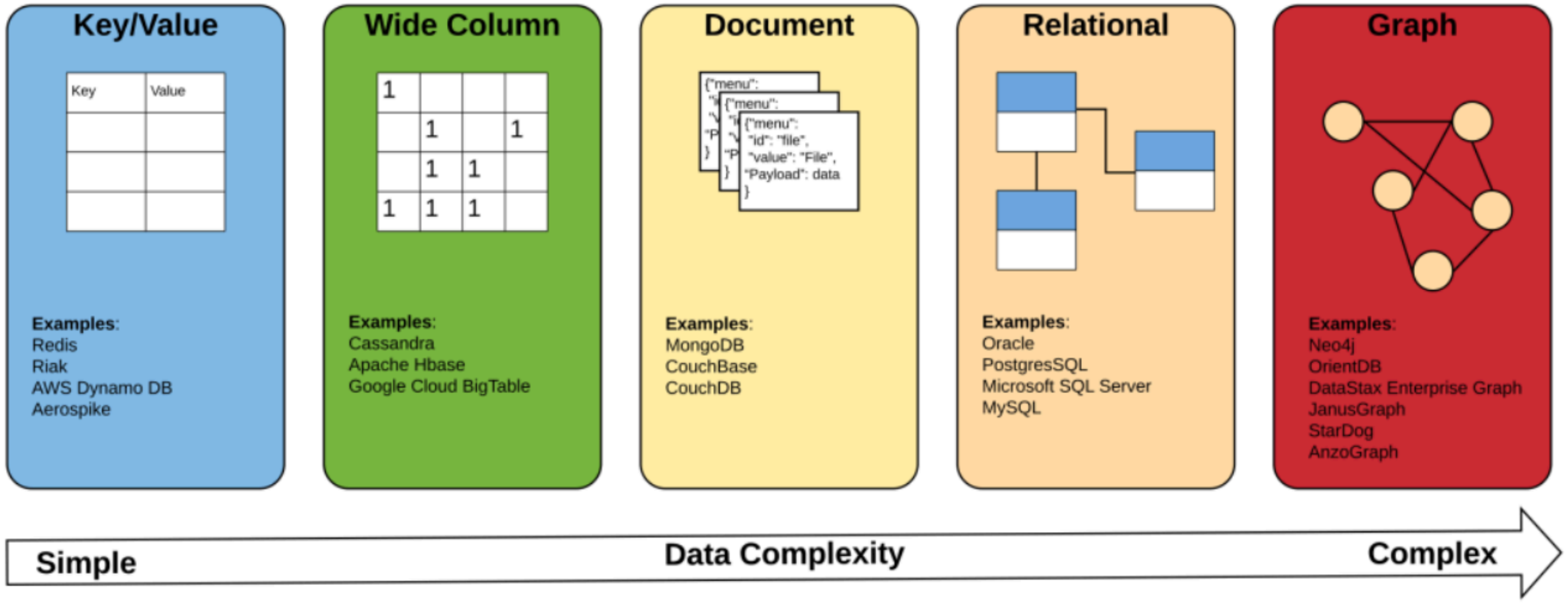
Graphs and graph terminology

- Vertices (singular: vertex) a.k.a. nodes.
- Edges a.k.a. relationships, links, or connections.
- Properties.

Graphs and graph terminology

- Vertices (singular: vertex) a.k.a. nodes.
- Edges a.k.a. relationships, links, or connections.
- Properties.
- Graph - A set of vertices and edges along with their properties.

Comparison with other types of databases



Key/Value

Examples:
Redis
Riak
AWS Dynamo DB
Aerospike

Wide Column

Examples:
Cassandra
Apache Hbase
Google Cloud BigTable

Document

Examples:
MongoDB
CouchBase
CouchDB

Relational

Examples:
Oracle
PostgresSQL
Microsoft SQL Server
MySQL

Graph

Examples:
Neo4j
OrientDB
DataStax Enterprise Graph
JanusGraph
StarDog
AnzoGraph

Simple **Data Complexity** **Complex**

Database engine types ordered by data complexity. Source: [Bec]

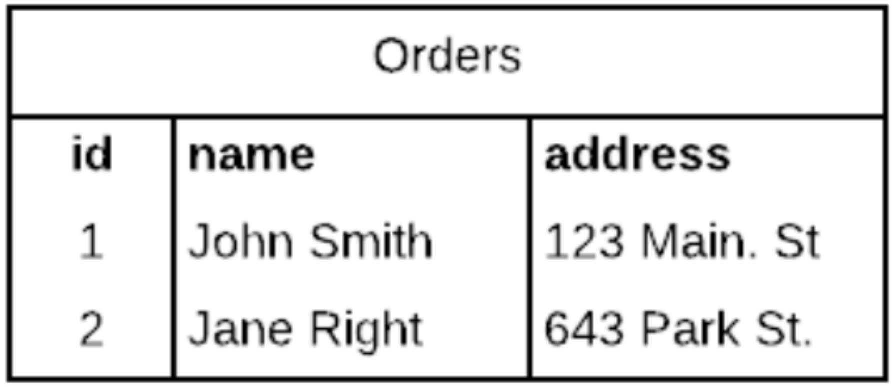
Do we really need another one database?

RECURSIVE QUERIES

- EXAMPLE

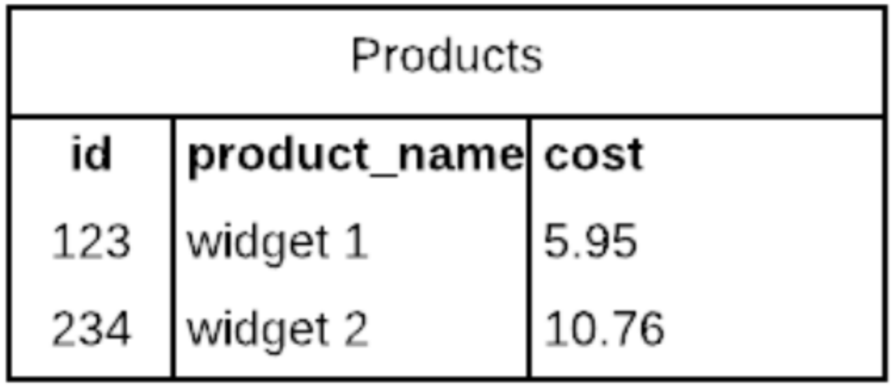
Do we really need another one database?

DIFFERENT RESULT TYPES



Orders

id	name	address
1	John Smith	123 Main. St
2	Jane Right	643 Park St.

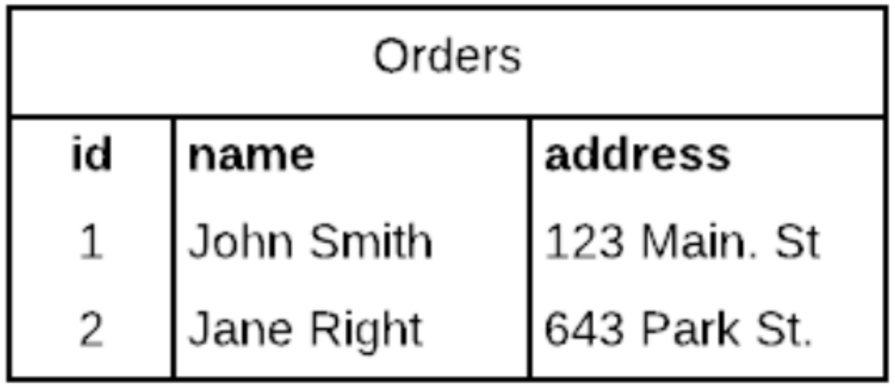


Products

id	product_name	cost
123	widget 1	5.95
234	widget 2	10.76

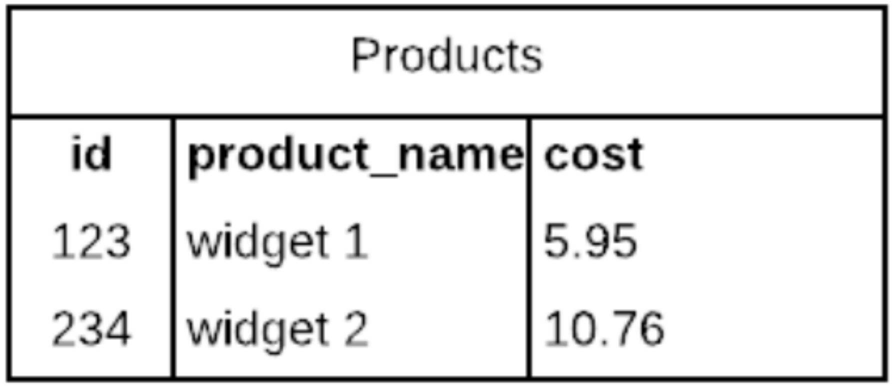
Do we really need another one database?

DIFFERENT RESULT TYPES



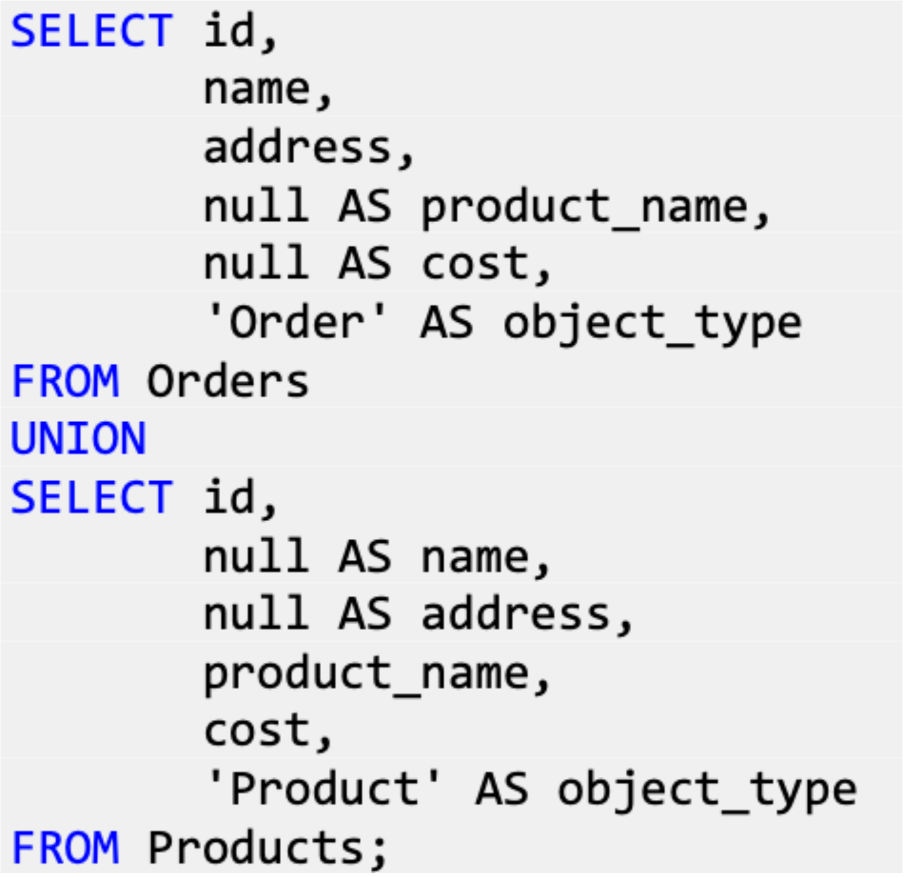
Orders

id	name	address
1	John Smith	123 Main. St
2	Jane Right	643 Park St.



Products

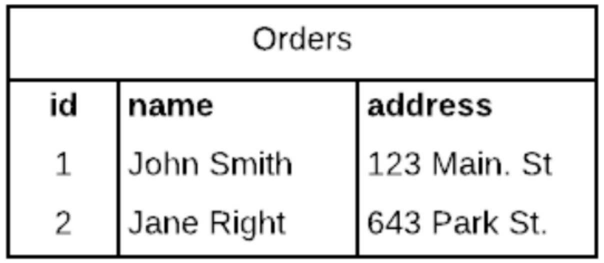
id	product_name	cost
123	widget 1	5.95
234	widget 2	10.76



```sql
SELECT id,
       name,
       address,
       null AS product_name,
       null AS cost,
       'Order' AS object_type
FROM Orders
UNION
SELECT id,
       null AS name,
       null AS address,
       product_name,
       cost,
       'Product' AS object_type
FROM Products;
```

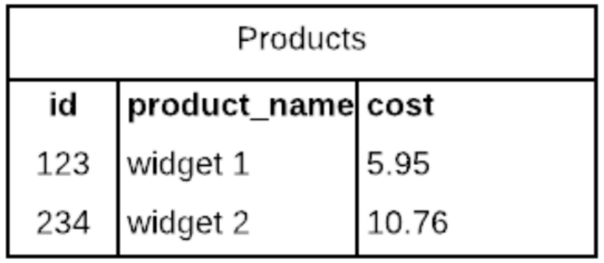
Do we really need another one database?

DIFFERENT RESULT TYPES



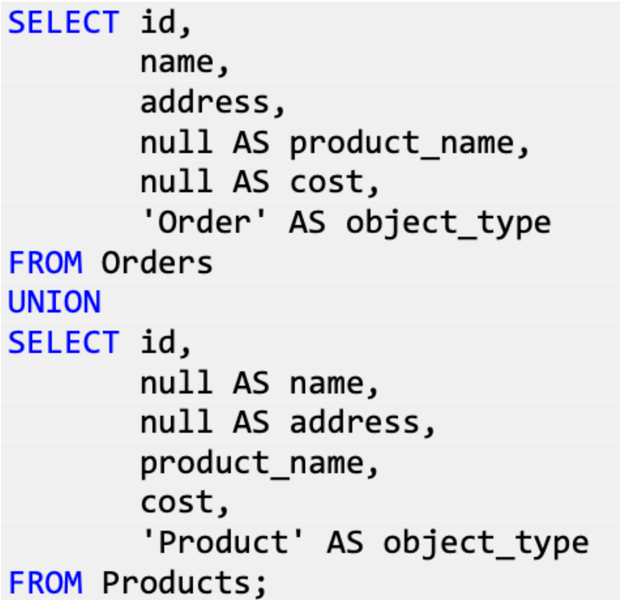
Orders

id	name	address
1	John Smith	123 Main. St
2	Jane Right	643 Park St.

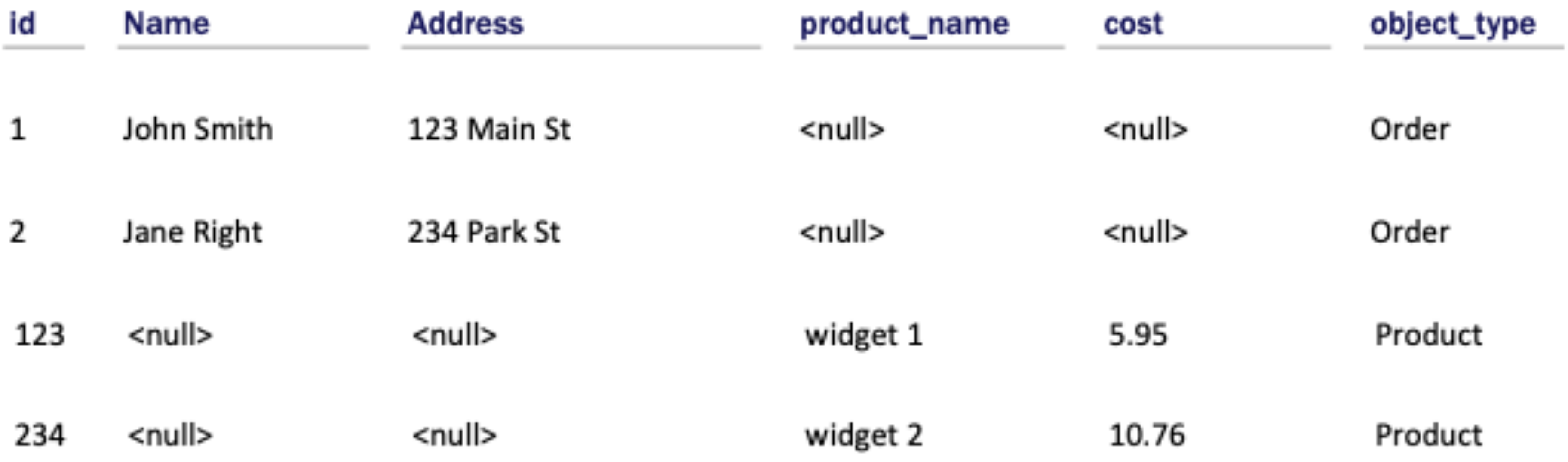


Products

id	product_name	cost
123	widget 1	5.95
234	widget 2	10.76



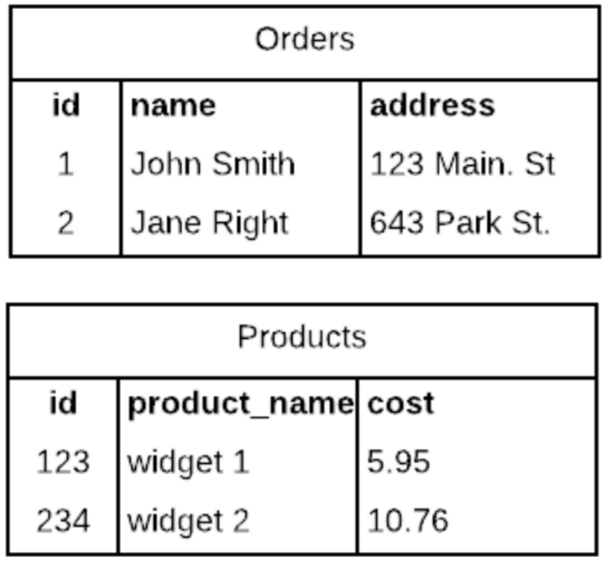
```sql
SELECT id,
       name,
       address,
       null AS product_name,
       null AS cost,
       'Order' AS object_type
FROM Orders
UNION
SELECT id,
       null AS name,
       null AS address,
       product_name,
       cost,
       'Product' AS object_type
FROM Products;
```



id	Name	Address	product_name	cost	object_type
1	John Smith	123 Main St	<null>	<null>	Order
2	Jane Right	234 Park St	<null>	<null>	Order
123	<null>	<null>	widget 1	5.95	Product
234	<null>	<null>	widget 2	10.76	Product

Do we really need another one database?

DIFFERENT RESULT TYPES

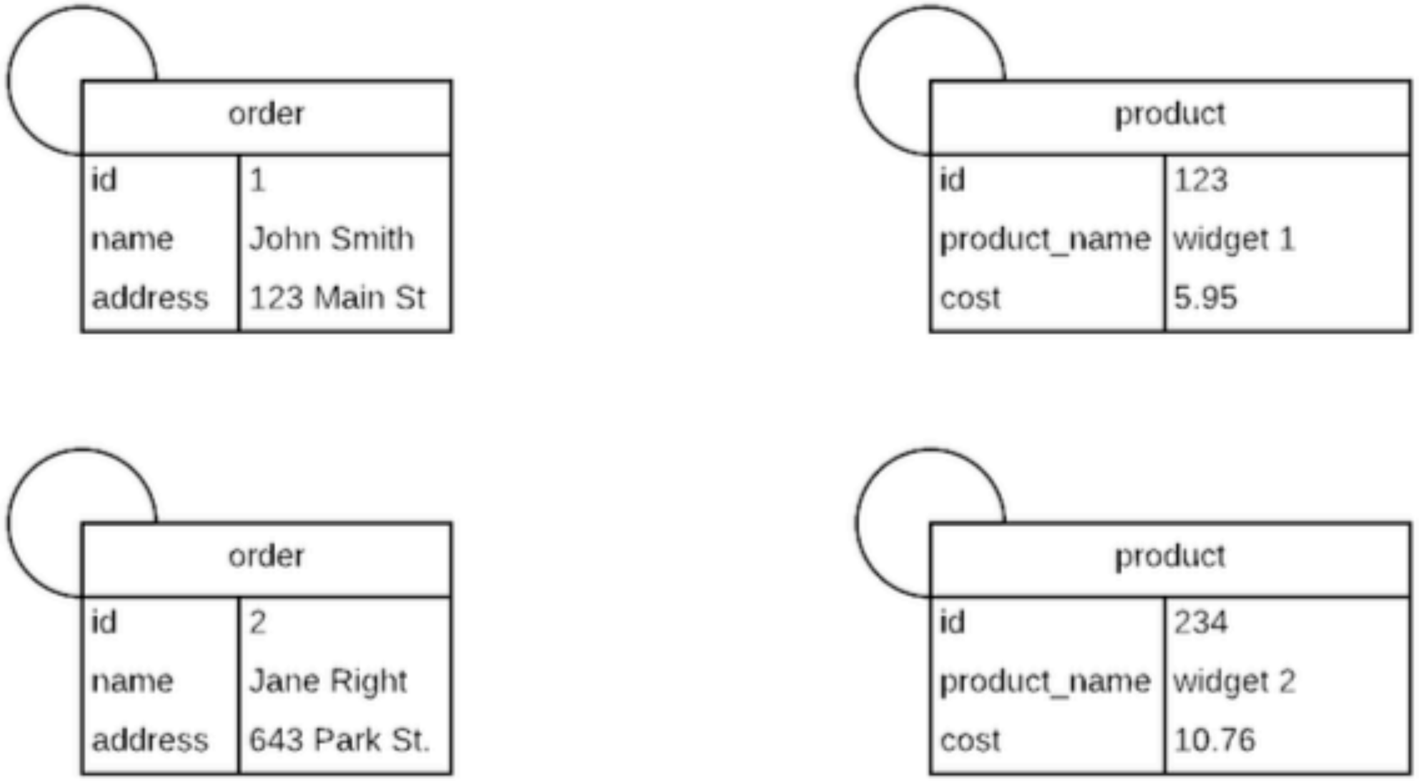


Orders

id	name	address
1	John Smith	123 Main. St
2	Jane Right	643 Park St.

Products

id	product_name	cost
123	widget 1	5.95
234	widget 2	10.76



order

id	1
name	John Smith
address	123 Main St

product

id	123
product_name	widget 1
cost	5.95

order

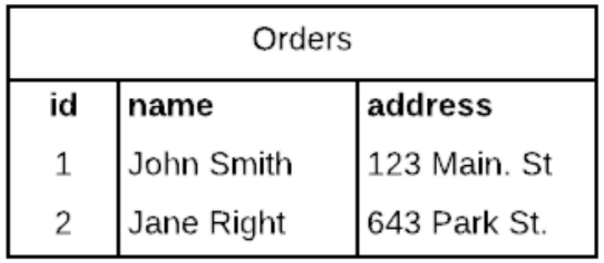
id	2
name	Jane Right
address	643 Park St.

product

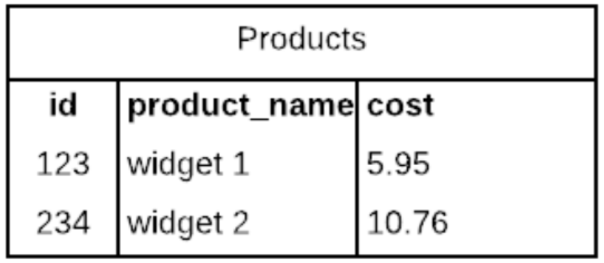
id	234
product_name	widget 2
cost	10.76

Do we really need another one database?

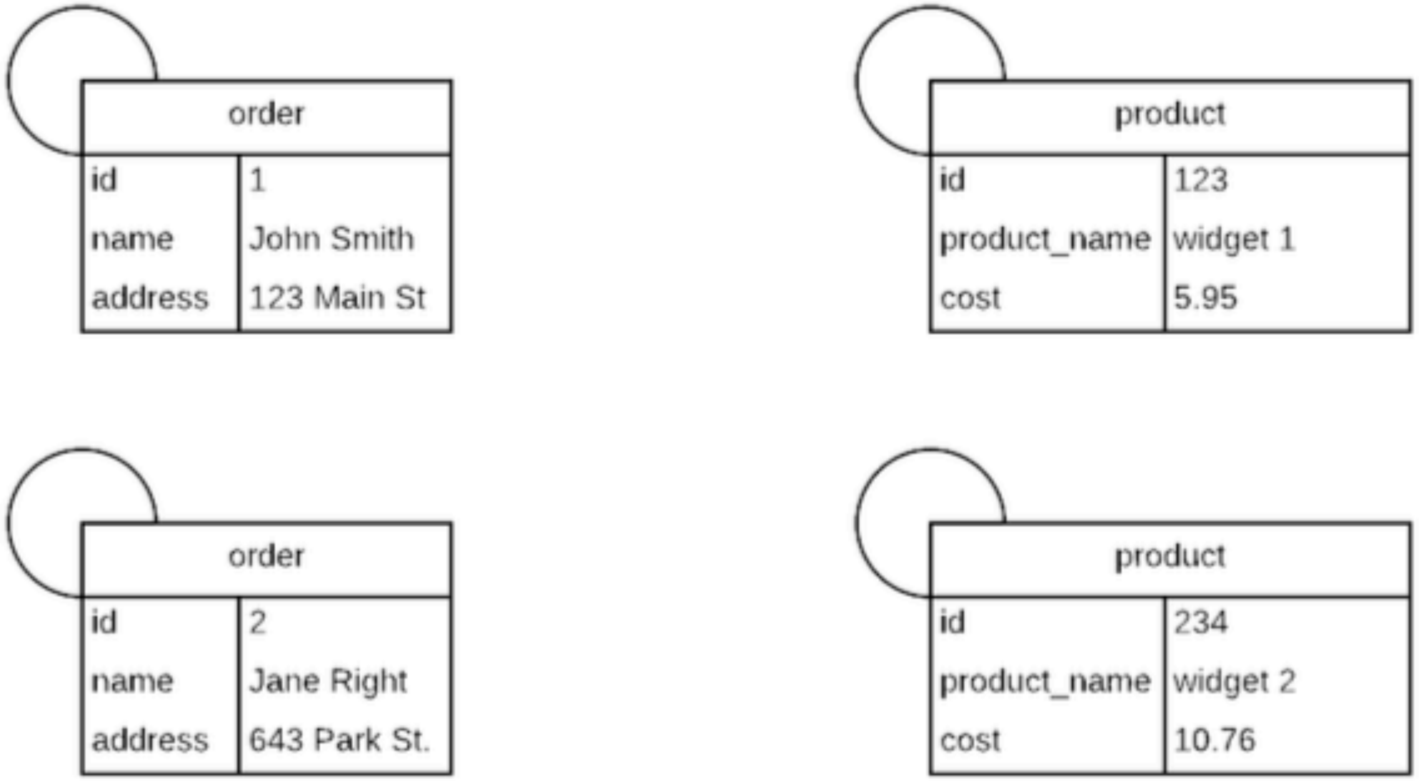
DIFFERENT RESULT TYPES



Orders		
id	**name**	**address**
1	John Smith	123 Main. St
2	Jane Right	643 Park St.



Products		
id	**product_name**	**cost**
123	widget 1	5.95
234	widget 2	10.76

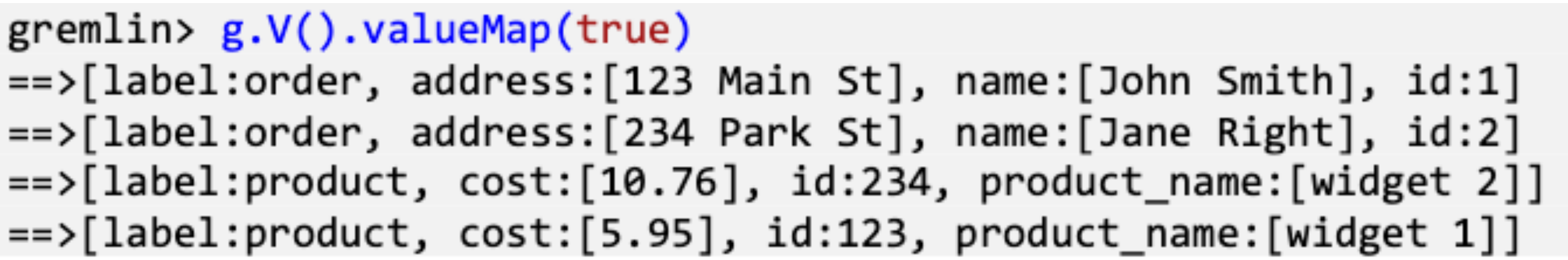


order	
id	1
name	John Smith
address	123 Main St

product	
id	123
product_name	widget 1
cost	5.95

order	
id	2
name	Jane Right
address	643 Park St.

product	
id	234
product_name	widget 2
cost	10.76



```
gremlin> g.V().valueMap(true)
==>[label:order, address:[123 Main St], name:[John Smith], id:1]
==>[label:order, address:[234 Park St], name:[Jane Right], id:2]
==>[label:product, cost:[10.76], id:234, product_name:[widget 2]]
==>[label:product, cost:[5.95], id:123, product_name:[widget 1]]
```

Do we really need another one database?

PATHS

River crossing puzzle: we have a fox, a goose, and a bag of barley that must be transported across a river by a farmer on a boat. However, this movement is bound by the following constraints:

- The boat can only carry one item in addition to the farmer on each trip.
- The farmer must go on each trip.
- The fox cannot be left alone with the goose or it will eat it.
- The goose cannot be left alone with the grain or it will eat it.

Do we really need another one database?

PATHS

Let’s start by modeling the initial state of our system as a vertex in our graph, which we’ll call: **TGFB_** with each character representing part of the problem:

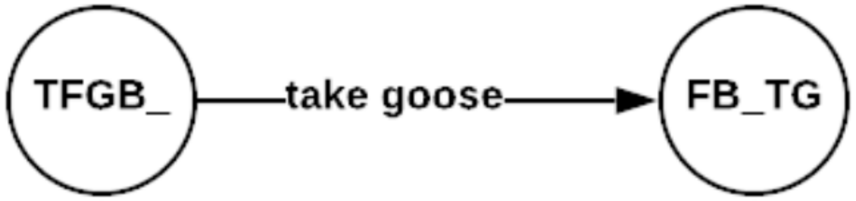
T – the boat & farmer

G – the goose

F – the fox

B – the barley

_ – the river



Do we really need another one database?

PATHS

River crossing puzzle full graph:

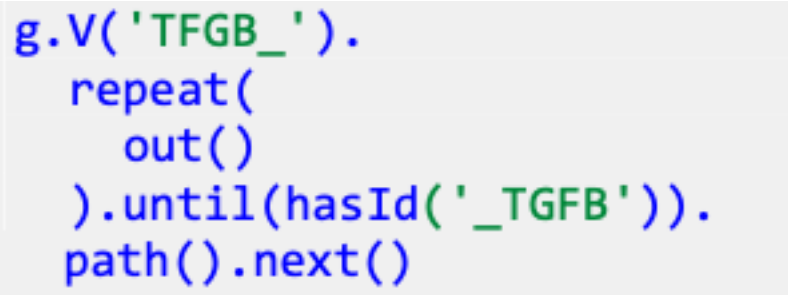
[draw it here]

Do we really need another one database?

PATHS

River crossing puzzle full graph:

[draw it here]



```
g.V('TFGB_').
  repeat(
    out()
  ).until(hasId('_TGFB')).
  path().next()
```

Ask yourselves

Is my problem a graph problem?

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?
    - Selection / search

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?
    - Selection / search
    - Related or recursive data

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?
    - Selection / search
    - Related or recursive data
    - Aggregation

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?
  - Selection / search
  - Related or recursive data
  - Aggregation
  - Pattern matching, influence

Ask yourselves

Is my problem a graph problem?

- What problem are we trying to solve?
  - Selection / search
  - Related or recursive data
  - Aggregation
  - Pattern matching, influence
- …

Ask yourselves

Is my problem a graph problem?

Ask yourselves

Is my problem a graph problem?

- Do we care about the relationships between entities as much or more than the entities themselves?

Ask yourselves

Is my problem a graph problem?

- Do we care about the relationships between entities as much or more than the entities themselves?
- Does my sql query perform multiple joins on the same table or require a recursive CTE?

Ask yourselves

Is my problem a graph problem?

- Do we care about the relationships between entities as much or more than the entities themselves?
- Does my sql query perform multiple joins on the same table or require a recursive CTE?
- Is the structure of my data continuously evolving?

Ask yourselves

Is my problem a graph problem?

- Do we care about the relationships between entities as much or more than the entities themselves?
- Does my sql query perform multiple joins on the same table or require a recursive CTE?
- Is the structure of my data continuously evolving?
- Is my domain a natural fit for a graph?

Bibliography

- [Bec] Dave Bechberger, Josh Perryman, *Graph Databases in Action*, Manning Publications, 2020
- [Ful] Piotr Fulmański, *NoSQL. Theory and examples*, Piotr Fulmański, 2021