



# **OrientDB**

**graph database  
in practice**

**Aurelijus Banelis**

# About me

**Aurelijus Banelis**

**aurelijus@banelis.lt**  
**aurelijus.banelis.lt**

**Software Engineer at NFQ**



# You will learn

## WHAT

Graph  
Graph database

## WHY

Real world  
example

## HOW

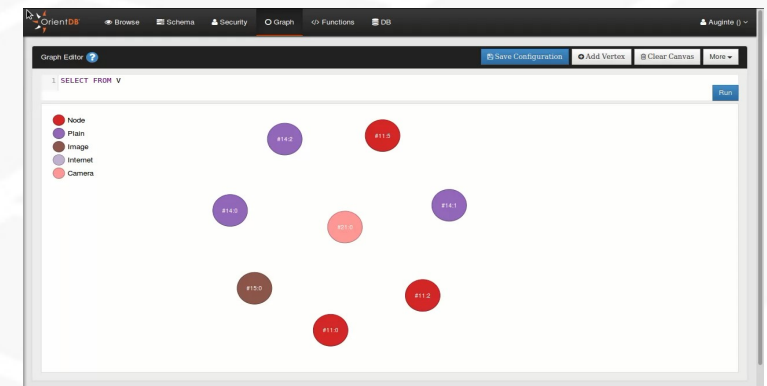
With PHP  
With relational database  
Best practice



# WHAT: Graph



Source: [http://commons.wikimedia.org/wiki/File:Tiskevicius\\_Juozapas\\_1835-1891.JPG](http://commons.wikimedia.org/wiki/File:Tiskevicius_Juozapas_1835-1891.JPG)



**Not Graf**

mathematical **graph**

Read more at: [http://en.wikipedia.org/wiki/Graph\\_%28mathematics%29](http://en.wikipedia.org/wiki/Graph_%28mathematics%29)

# Graph - vertices, edges and properties

Graph Editor ? Save Configuration Add Vertex Clear Canvas More

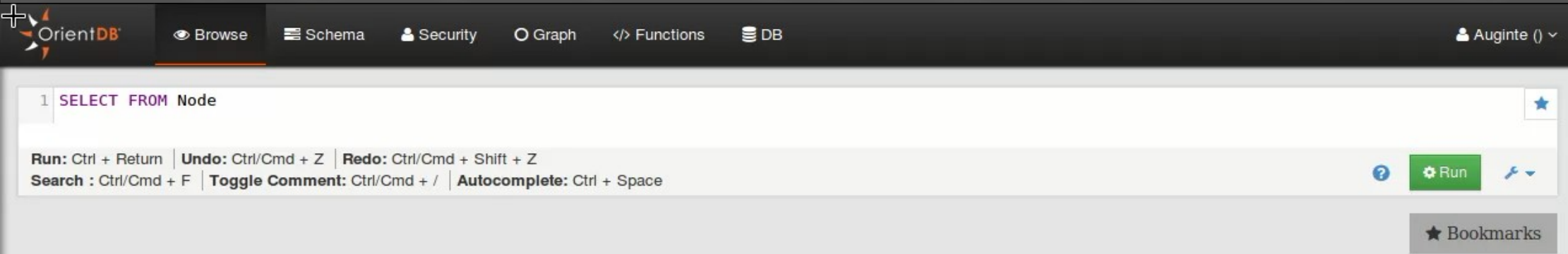
```
1 SELECT FROM V
```

Run

- Node
- Plain
- Image
- Internet
- Camera

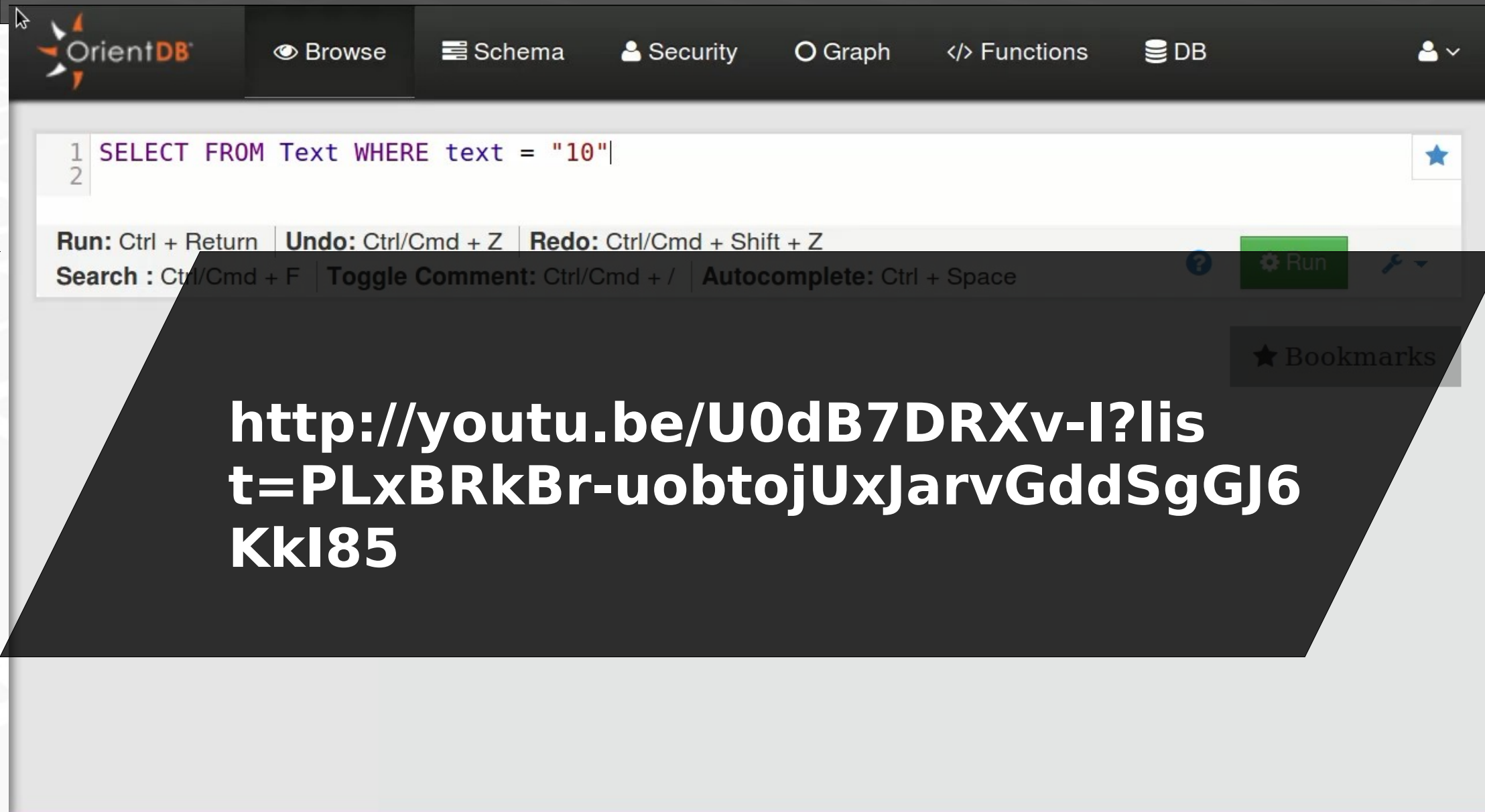
<http://youtu.be/0nQj5ruYoAs?list=PLxBRkBr-uobtojUxJarvGddSgGJ6KkI85>

# Data visualization: table, document, graph



[http://youtu.be/mA\\_1McQ4ifl?list=PLxBRkBr-uobtojUxJarvGddSgGJ6KkI85](http://youtu.be/mA_1McQ4ifl?list=PLxBRkBr-uobtojUxJarvGddSgGJ6KkI85)

# Graph database is optimized for many relations



<http://youtu.be/U0dB7DRXv-I?list=PLxBRkBr-uobtojUxJarvGddSgGJ6KkI85>

# Summary (1/3)

## WHAT

**Graph**  
**Graph database**

## WHY

Real world  
example

## HOW

With PHP  
With relational database  
Best practice

## Graph

Vertices and edges  
As data visualization

## Graph database

Optimized for many relations





# WHY: Common operations

Auginte

<http://youtu.be/ZcEX9n4-NRM?list=PLxBRkBr-uobtojUxJarvGddSgGJ6Kkl85>

v0.6

v0.7

Long-term plan

- Infinity zooming
- Source tracking
- Distributed architecture

Short-term plan

- Stable infinity zooming
- Self-organizing teams
- Self-organizing teams

Backlog



Calendar

**License:** Apache 2 (free + for commercial)

**Structures:** Schema-hybrid, with extend

**Scalability:** Sharding, replication, WAL

**Used by:** Cisco, Ericsson

**API:** JavaLib, binary, REST

**Query languages:**  
WEB, SQL, Gremlin

**Results:**  
Synchronous, asynchronous

**Storage:**  
File, memory, remote

**Documentation:**  
Tutorials, groups, source

**Text search:**  
Lucene full text index

**Transactions:**  
ACID, MVCC

**Attributes:**  
Read-only, metadata

**Time for  
DEMO**

# Summary (2/3)

## WHAT

Graph  
Graph database

## WHY

Real world  
example

## HOW

With PHP  
With relational database  
Best practice

**Structures/Operations**  
Trees with cycles = graph  
Traverse like operations

**Tools**  
Debugging complex structures

**Business logic**  
List or association based

# HOW: Notes for developers



## PHP

**OrientDB in PHP world**

## Migration

**and learning cost**

## Relational

**database integration**

## Design

**structures in database**

# HOW: in PHP world

**Official**  
PHP >=5.4

**PhpOrient**  
OrientDB >=1.7.4

```
"require": {  
  "ostico/phporient": "1.1.*"  
}
```

**First/old OrientDB-PHP**  
PHP >=5.3 API getting outdated

**OrientDB-PHP**

```
"require": {  
  "ostico/ostico-orientdb-php/orientdb-php": "dev-master"  
}
```

## Not recommend

No complex results = no graph

Use PhpOrient instead

**Doctrine**  
**Symfony**  
PHP >=5.3

**OrientDB-ODM**  
On top of OrientDB-PHP

```
"require": {  
  "ostico/ostico-orientdb-odm": "dev-master"  
}  
  
"require": {  
  "concept-it/orient-db-bundle": "dev-master"  
}
```

**Slower**

**REST/HTTP API**

<http://0.0.0.0:2480>

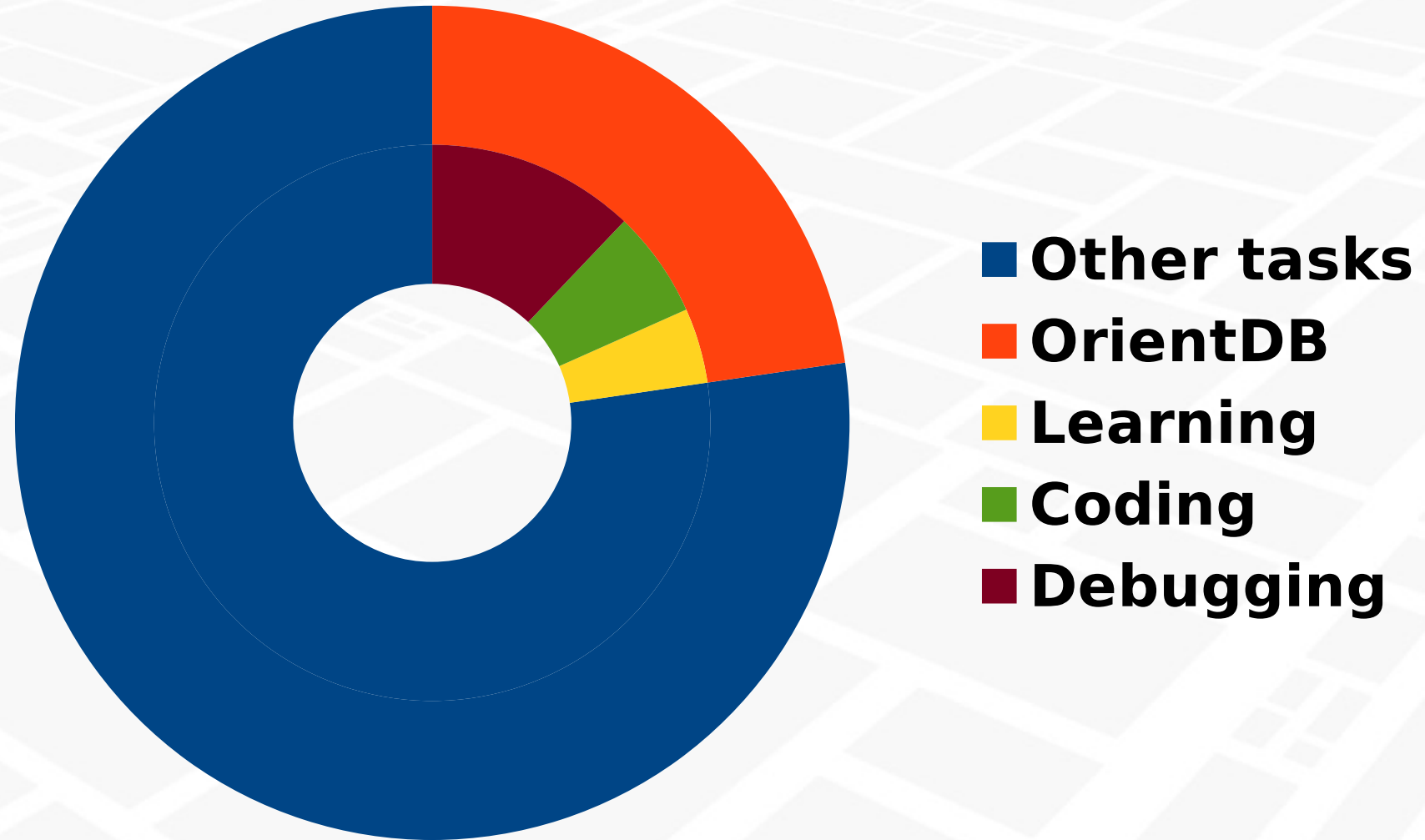
# HOW: PhpOrient

```
/** @var Record[] $results */  
$results = $client->query(  
    'SELECT FROM Testas WHERE c.d.e = 132'  
);  
foreach ($results as $res) {  
    print "{$res->getRid()} {$res['c']['d']['f']}";  
}
```

```
/** @var Bag $inParent */ /** @var ID $id */  
$inParent = $res['in_Parent'];  
foreach ($inParent as $id) {  
    $child = $client->recordLoad($id);  
}
```



# HOW: Migration costs



\* Based on time used in my personal project

# HOW: Integrate with relational database

## **OrientDB ETL**

**Native tool to import  
Too hard for n-n relations**

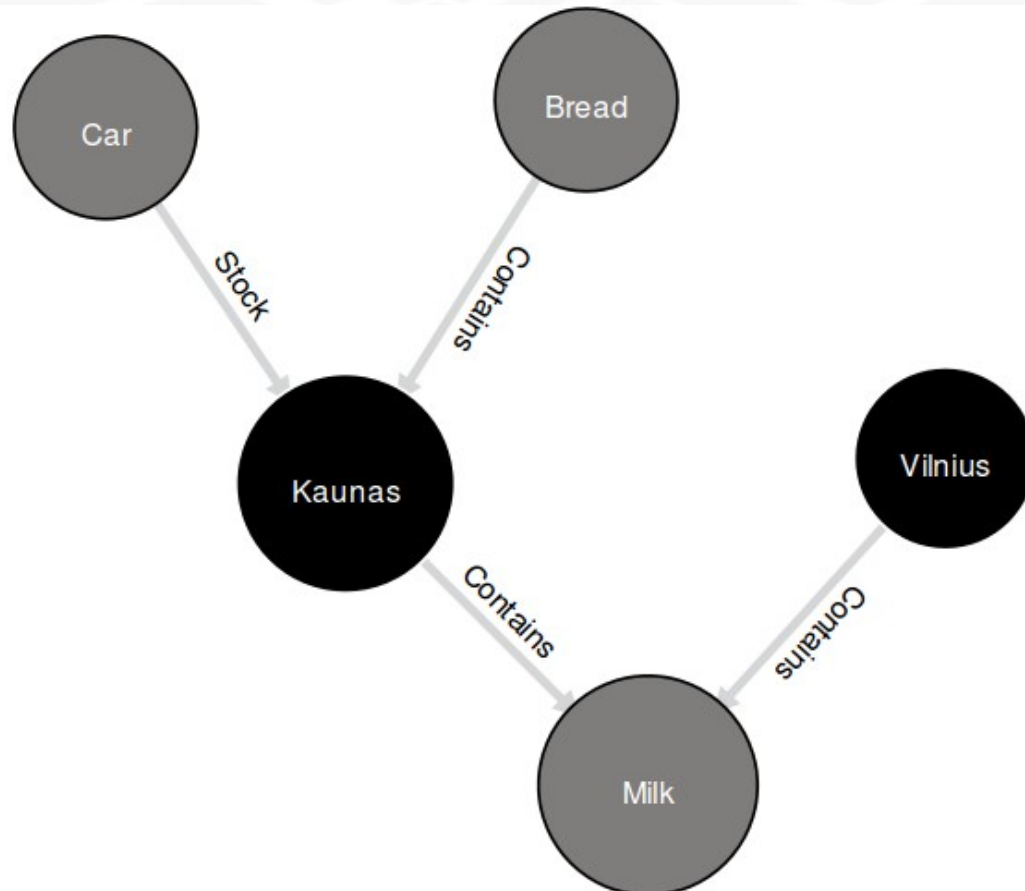
## **SQL Batch**

**Execute multiple queries at  
the server side**

```
LET v1 = CRAETE VERTEX Article SET name="123"  
LET v2 = CRAETE VERTEX Action SET name="456"  
LET v3 = CRAETE VERTEX Stock SET house="A", amount=2  
CREATE EDGE Discount FROM $v1 TO $v2  
CREATE EDGE Contains FROM $v3 TO $v1  
  
RETURN [$v1, $v2, $v3]
```

# HOW: Design structure in database

## Where to put parameters?



### In edges

- Not shown in Graph view
- Strange Traverse behavior

### Only in vertices

- Easier to implement caching
- Easier to extend relations
- Imitating edge parameter with inheritance of vertices

# Summary (3/3)

## WHAT

Graph  
Graph database

## WHY

Real world  
example

## HOW

With PHP  
With relational database  
Best practice

**PHP wrapper**  
PhpOrient

**Migration cost**  
Time for debugging/tests

**Relational database**  
Use SQL Batch

**Design database**  
Use light edges

# Questions?

**WHAT**

Graph  
Graph database

**WHY**

Real world  
example

**HOW**

With PHP  
With relational database  
Best practice



# References and useful links

- **<http://www.orienttechnologies.com>**
- <https://github.com/orientechnologies/orientdb/wiki/SQL-Functions>
- <https://github.com/orientechnologies/orientdb/wiki/SQL-Traversal>
- <http://www.orienttechnologies.com/docs/latest/orientdb.wiki/Programming-Language-Bindings.html>
- <https://github.com/orientechnologies/orientdb/wiki/Document-Database#prepared-query>
- <https://github.com/orientechnologies/orientdb/wiki/Fetching-Strategies>
- <http://neo4j.com/>
- <http://auginte.com>