Introduction to Graph Databases with Neo4j

Michael Moussa | @michaelmoussa

http://legacy.joind.in/event/view/4525

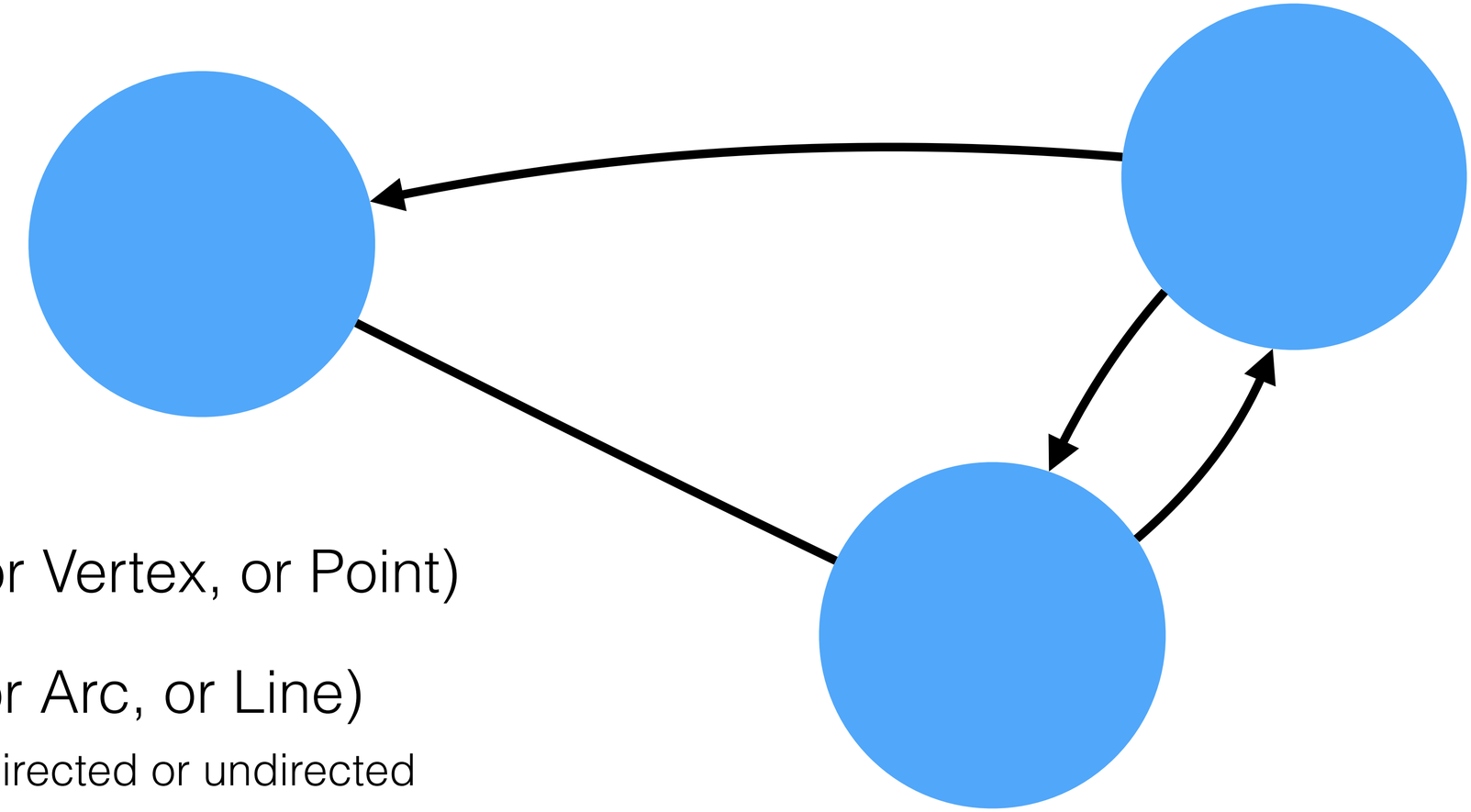
https://github.com/michaelmoussa/talks/tree/master/intro-to-graph-databases-with-neo4j

About me

- Web application developer for 16 years
- Senior Software Engineer - Payments, LINIO
- Neo4j Certified Professional

What is a Graph?

Structure that models relationships between objects



Node (or Vertex, or Point)

Edge (or Arc, or Line)

*Can be directed or undirected

Graph Theory

Study of graphs and their applications

Used to model various problems in biology, chemistry, physics, etc.

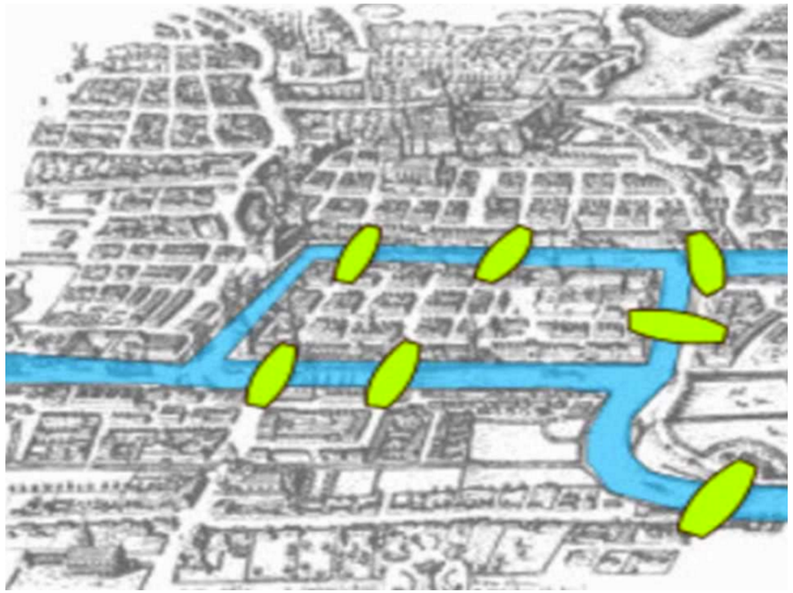
Leonhard Euler (1707-1783) considered the "father" of graph theory

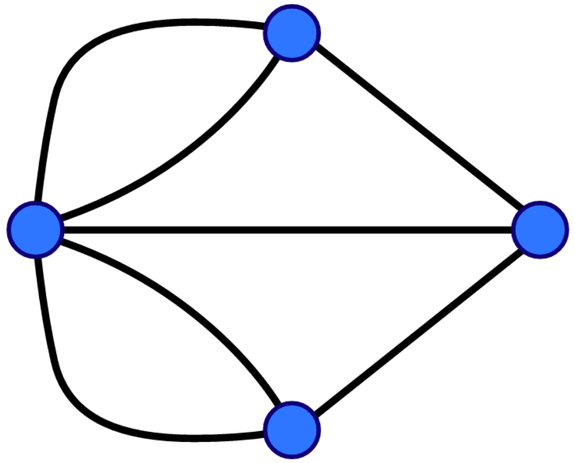
Seven Bridges of Königsberg

Can you walk through the city crossing each and every bridge exactly once?

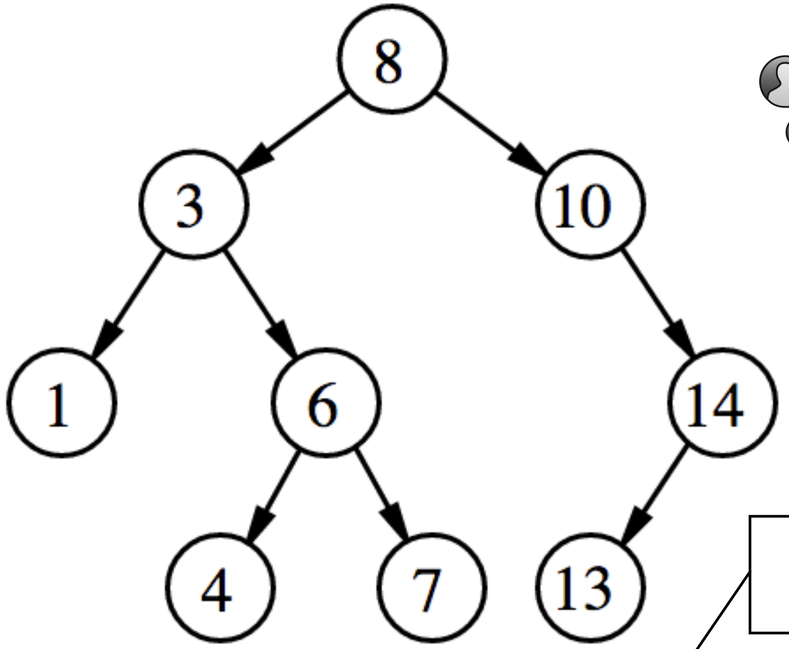
Euler proved in 1736 that the answer is "No".

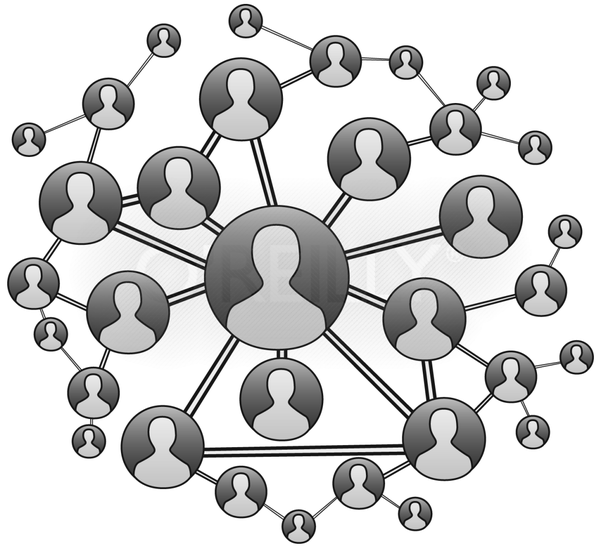
His solution is the first theorem of graph theory.

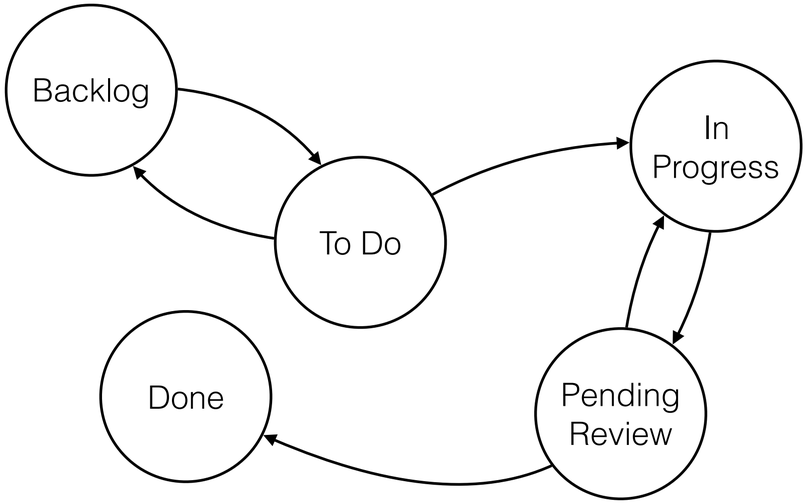


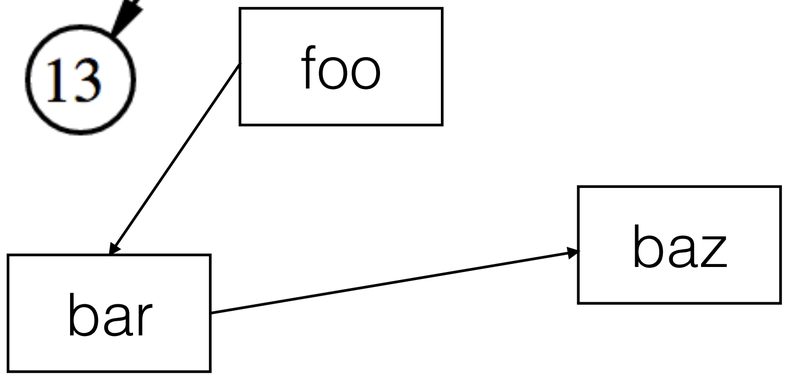


Graphs are Everywhere!



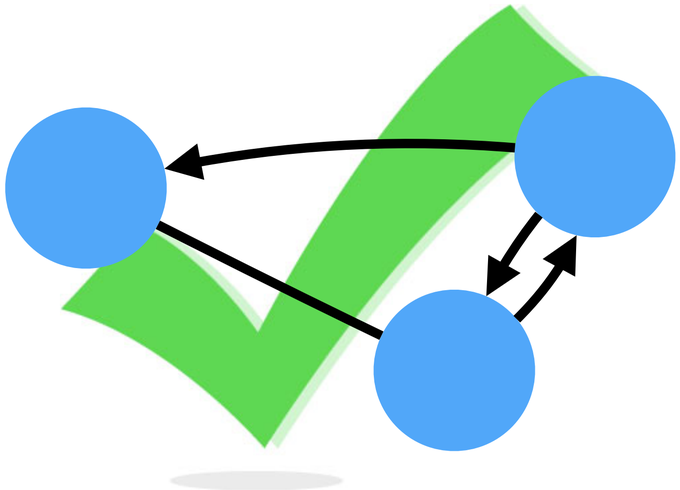






- Driving directions
- Product recommendations
- Financial fraud detection
- Identity and access management

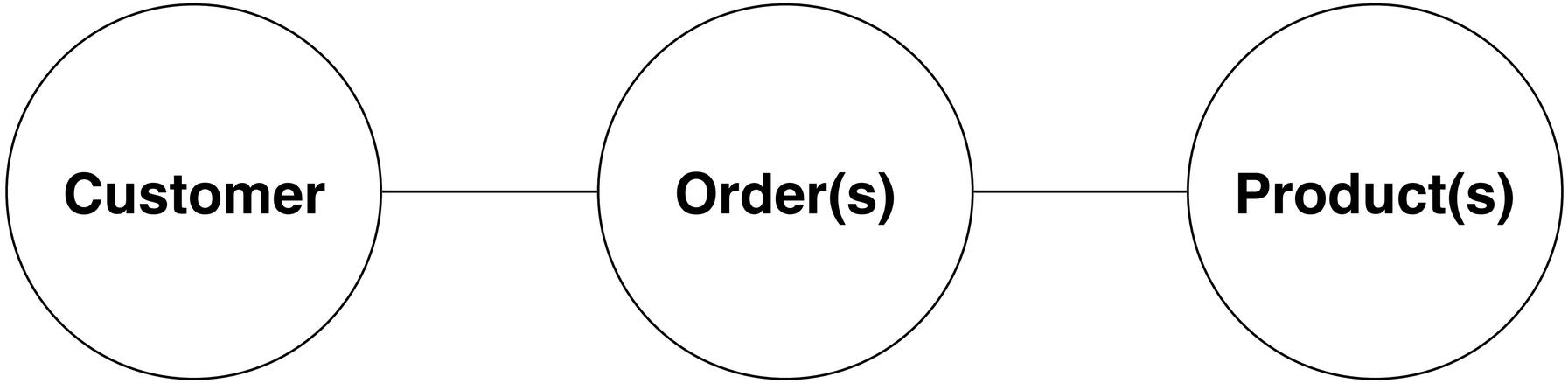
Graph Database



- Store information as nodes and relationships
- Relationships between data are <u>first-class citizens</u>

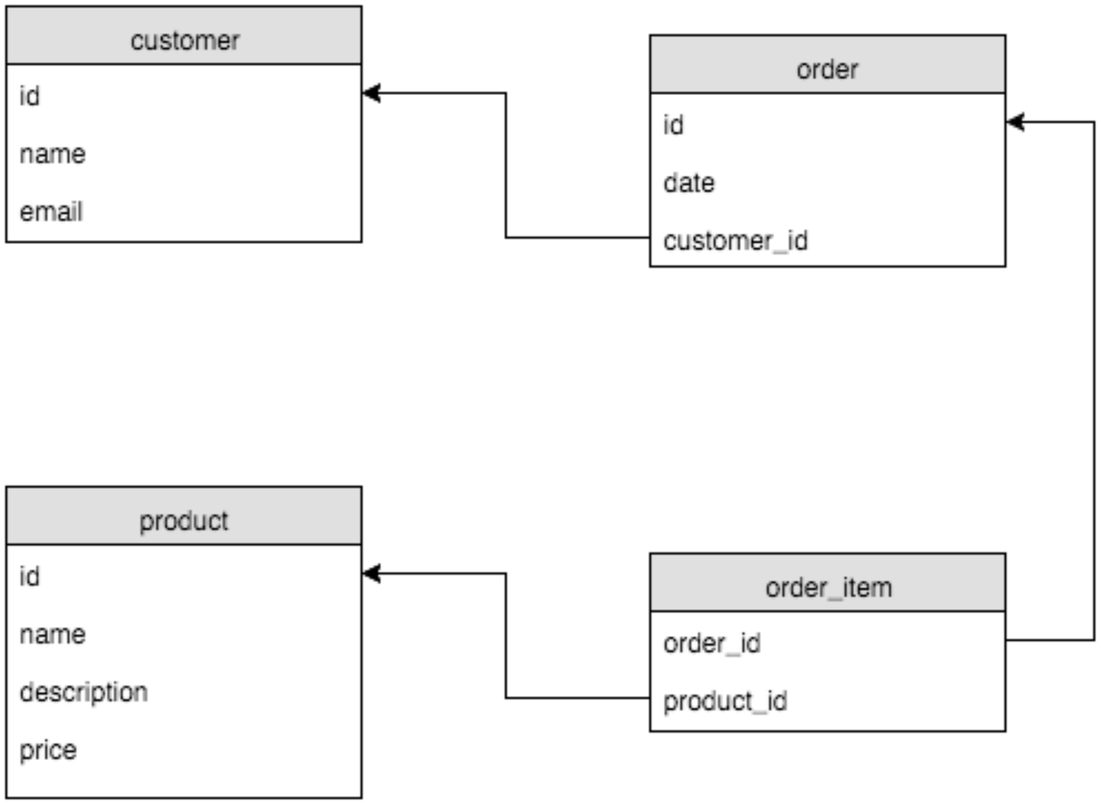
E-commerce Example

A **customer** places **order(s)** consisting of **product(s)**



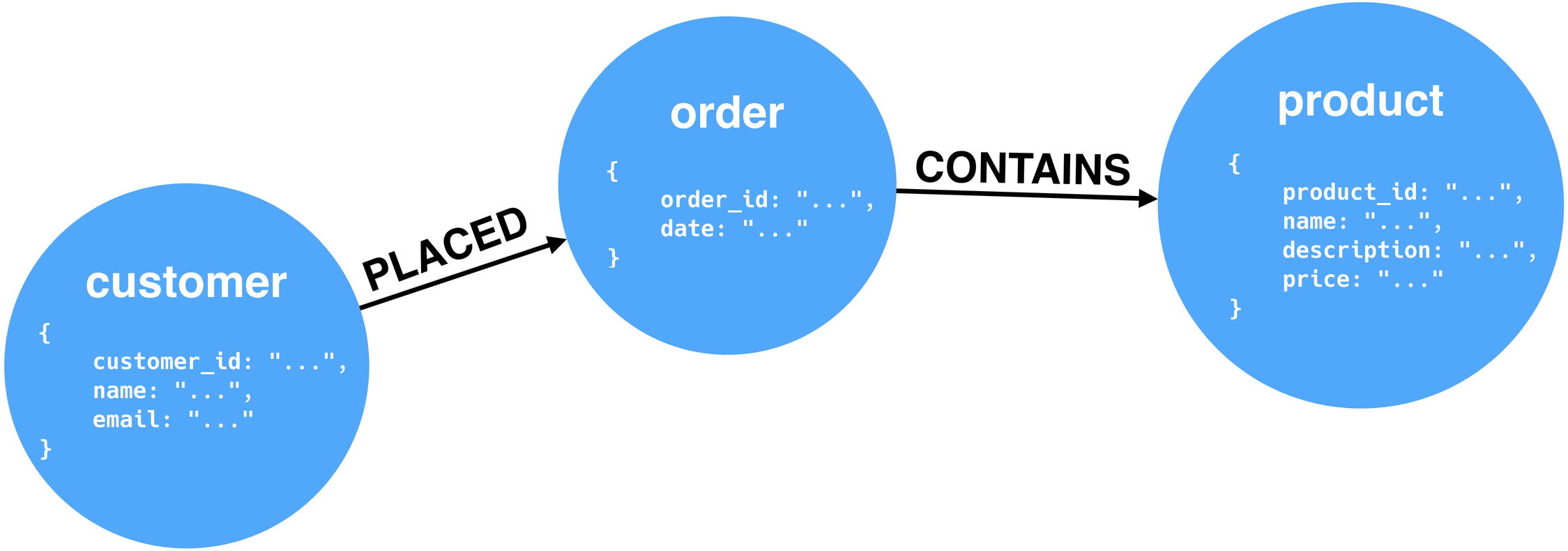
E-commerce Example

Modeling in a relational database



E-commerce Example

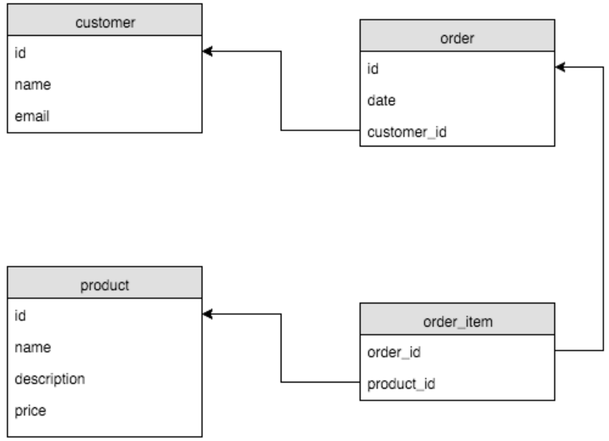
Modeling in a graph database



E-commerce Example

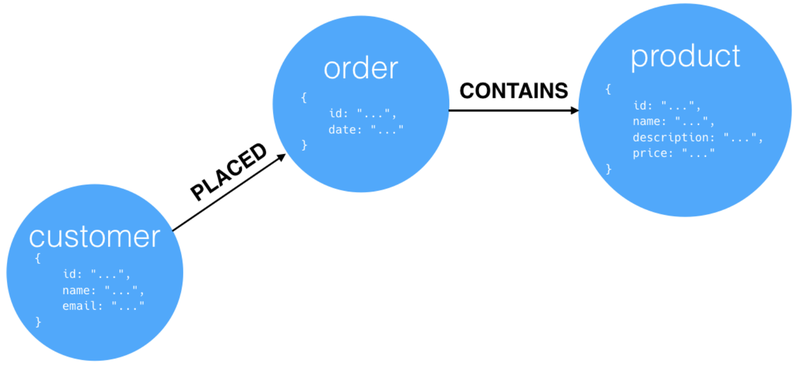
RDBMS

- Highly-structured
- Entities related via JOIN tables and foreign keys



Graph DB

- Schemaless, flexible
- Relationships are as important as the data itself



neo4j

The world's most popular graph database

- Native graph database
- Open source
- ACID compliant
- Powerful and expressive query language
- Excellent documentation
- Active community

Labeled Property Graph

- Entities are **nodes** containing various **properties**.
- **Relationships** connect those nodes to others, and are enhanced by **properties** of their own.
- Nodes are grouped with like nodes using **labels**.

Capacity

Nodes: ~34 billion

Relationships: ~34 billion

Properties: 68 billion - 274 billion (depending on datatype)

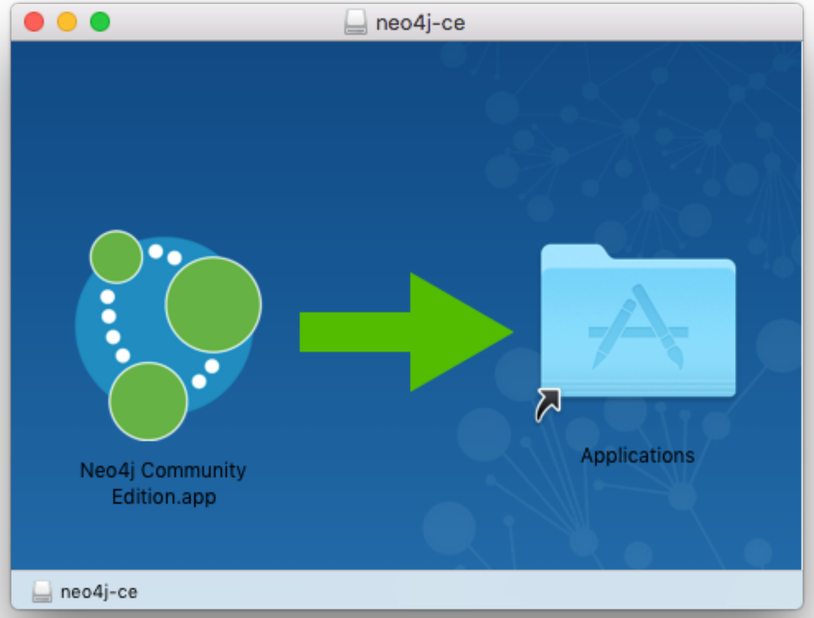
Relationship types: 65,000

onefinestay
ZEPHYR HEALTH INC
CISCO
53
gamesys
Global 500 Media
Global 500 Logistics
doximity Linking Medical Minds
SNAP INTERACTIVE
mallowstreet
Lufthansa Systems
Walmart
WANDERU
TOMTOM
telenor
SharePractice
meetic
tallcat
AXON ACTIVE Focusing on your decisions
webshots wallpaper & screensaver
InfoJobs
zeebox
ebay
megree
Glowbl
DingLicom
Mapillary
Global 500 Manufacturing
CrunchBase
eHarmony
pitney bowes
Global 2000 Adidas
wazoku
classmates
hp
Hinge
Care.com There for you
LIFECHURCH.TV
Aol.
UBS
maaii Let's connect
shutl
entropy
CareerArcGroup
spring
UNIFIED
M
CenturyLink
NII 国立情報学研究所 National Institute of Informatics
aikux.com
Curaspan HEALTH GROUP
ePals
<fuseworks/>
gen
dop
MONSANTO
ALERTLOGIC Security Compliance Cloud
Legacy.com
VIRTUAL INSTRUMENTS
ICE
janssen
kitedesk
NexLP
OPEN Tree of Life
ThingWorx A PTC Business
ECOLUTIS
PERIGEE
graphAlchemist
WILLERSOFT
Pro Integra S.L.
SERENA
SODIFRANCE CONSEIL, TECHNOLOGIES & SERVICES IT
timeBlend!
Zenoss
LA BANQUE POSTALE
myWave
workjam
nine#
LinkedIn
die Bayerische
Accenture
StarHub
noble group
SMARTMONSTERS
Pondera
SUPPLYHUB
compete
EQUILAR
Spree
NATIONAL GEOGRAPHIC
Livestation
NMBS B SNCB
KiwiRail
3 Three.com
YELAGO
wayblazer
HealthUnlocked
r
PERIGEE

Cypher

- Neo4j's query language
- Declarative
- SQL-inspired
- Open source!
  - http://opencypher.org

Installing Neo4j



http://neo4j.com/download/

DEMO

Neo4j-PHP-Client

https://github.com/graphaware/neo4j-php-client

```
composer require graphaware/neo4j-php-client
```

It's a REST API behind the scenes...

```
$ curl --header "Authorization: Basic <.....>" http://localhost:7474/db/data/
{
  "extensions" : { },
  "node" : "http://localhost:7474/db/data/node",
  "node_index" : "http://localhost:7474/db/data/index/node",
  "relationship_index" : "http://localhost:7474/db/data/index/relationship",
  "extensions_info" : "http://localhost:7474/db/data/ext",
  "relationship_types" : "http://localhost:7474/db/data/relationship/types",
  "batch" : "http://localhost:7474/db/data/batch",
  "cypher" : "http://localhost:7474/db/data/cypher",
  "indexes" : "http://localhost:7474/db/data/schema/index",
  "constraints" : "http://localhost:7474/db/data/schema/constraint",
  "transaction" : "http://localhost:7474/db/data/transaction",
  "node_labels" : "http://localhost:7474/db/data/labels",
  "neo4j_version" : "2.3.2"
}
```

```php
<?php

require_once 'vendor/autoload.php';

use Neoxygen\NeoClient\ClientBuilder;

$client = ClientBuilder::create()
    ->addConnection(
        'default', 'http', 'localhost', 7474,
        true, 'username', 'password'
    )
    ->setAutoFormatResponse(true)
    ->build();
```

```php
$cypher = 'MATCH (bob:Customer {email: { email } })
                -[:PLACED]->(order)
                -[:CONTAINS]->(product)
           RETURN order.date,
                  COLLECT(product.name) AS product_list';

$client->sendCypherQuery(
    $cypher,
    ['email' => 'bob@example.com']
);

$result = $client->getRows();
```

```
Array
(
    [order.date] => Array
        (
            [0] => 2016-02-05
        )

    [product_list] => Array
        (
            [0] => Array
                (
                    [0] => Dog biscuits
                    [1] => Squeaky dog toy
                )
        )
)
```

```php
$cypher = 'MATCH (bob:Customer {email: { email } })
                -[:PLACED]->(order)
                -[:CONTAINS]->(product)
           RETURN order.date,
                  COLLECT(product.name) AS product_list';

$client->sendCypherQuery(
    $cypher,
    ['email' => 'bob@example.com']
);

$result = $client->getResult()->getTableFormat();
```

```
Array
(
    [0] => Array
        (
            [order.date] => 2016-02-05
            [product_list] => Array
                (
                    [0] => Dog biscuits
                    [1] => Squeaky dog toy
                )
        )
)
```

Reference Material

- Official Documentation
  - http://neo4j.com/docs/stable
- Cypher Refcard
  - http://neo4j.com/docs/stable/cypher-refcard
- Online Training
  - http://neo4j.com/graphacademy/online-training
- Neo4j Certification
  - http://neo4j.com/graphacademy/neo4j-certification
- GraphGists
  - http://neo4j.com/graphgists
- Slack
  - https://neo4j-users-slack-invite.herokuapp.com
- Twitter
  - https://twitter.com/neo4j
- Chef Cookbook
  - https://supermarket.chef.io/cookbooks/neo4j

USE THE RIGHT TOOL FOR THE JOB

http://legacy.joind.in/event/view/4525

```
MATCH (:Attendee)-[:HAS]->(q:Question)
RETURN q.text;
```