B4M36DS2, BE4M36DS2: **Database Systems 2**

https://cw.fel.cvut.cz/b221/courses/b4m36ds2/

Practical Class 10

Neo4j

Yuliia Prokop
prokoyul@fel.cvut.cz

1. 12. 2025

Author: Martin Svoboda
(martin.svoboda@matfyz.cuni.cz)

Czech Technical University in Prague, Faculty of Electrical Engineering

Data Model

Database system structure

Instance ⟶ single **graph**

Property graph = directed labeled multigraph

- Collection of vertices (nodes) and edges (relationships)

Node

- Internal identifier
- Set of labels, set of properties

Relationship

- Internal identifier
- **Direction**, start and end node
- Exactly one type, set of properties

First Steps

Connect to our NoSQL server

- SSH / PuTTY and SFTP / WinSCP
- **nosql.felk.cvut.cz**

Start Cypher shell

- `cypher-shell`

Get familiar with basic commands

- `help`
- `:exit`

Sample data (common database)

- See `/home/DS2/neo4j/data.cypher`

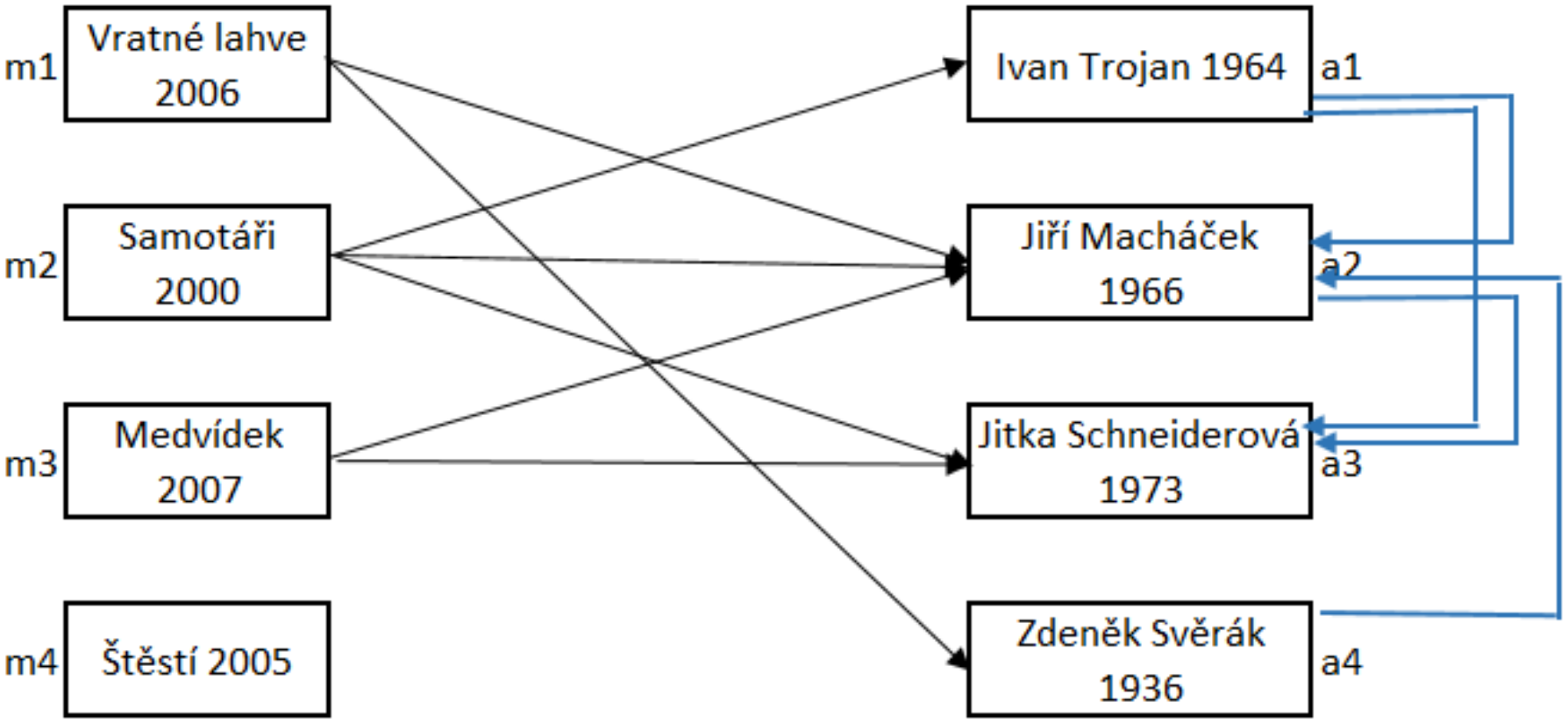
Source file

```
CREATE
  (m1:MOVIE { id: "vratnelahve", title: "Vratne lahve", year: 2006 }),
  (m2:MOVIE { id: "samotari", title: "Samotari", year: 2000 }),
  (m3:MOVIE { id: "medvidek", title: "Medvidek", year: 2007 }),
  (m4:MOVIE { id: "stesti", title: "Stesti", year: 2005 }),

  (a1:ACTOR { id: "trojan", name: "Ivan Trojan", year: 1964 }),
  (a2:ACTOR { id: "machacek", name: "Jiri Machacek", year: 1966 }),
  (a3:ACTOR { id: "schneiderova", name: "Jitka Schneiderova", year: 1973 }),
  (a4:ACTOR { id: "sverak", name: "Zdenek Sverak", year: 1936 }),
  (a5:ACTOR { id: "novak", name: "Jan Novak", year: 1970 }),
  (a6:ACTOR { id: "svoboda", name: "Petr Svoboda", year: 1965 }),
  (a7:ACTOR { id: "kral", name: "Lukas Kral", year: 1980 }),
  (a8:ACTOR { id: "novotny", name: "Martin Novotny", year: 1975 }),
  …
```

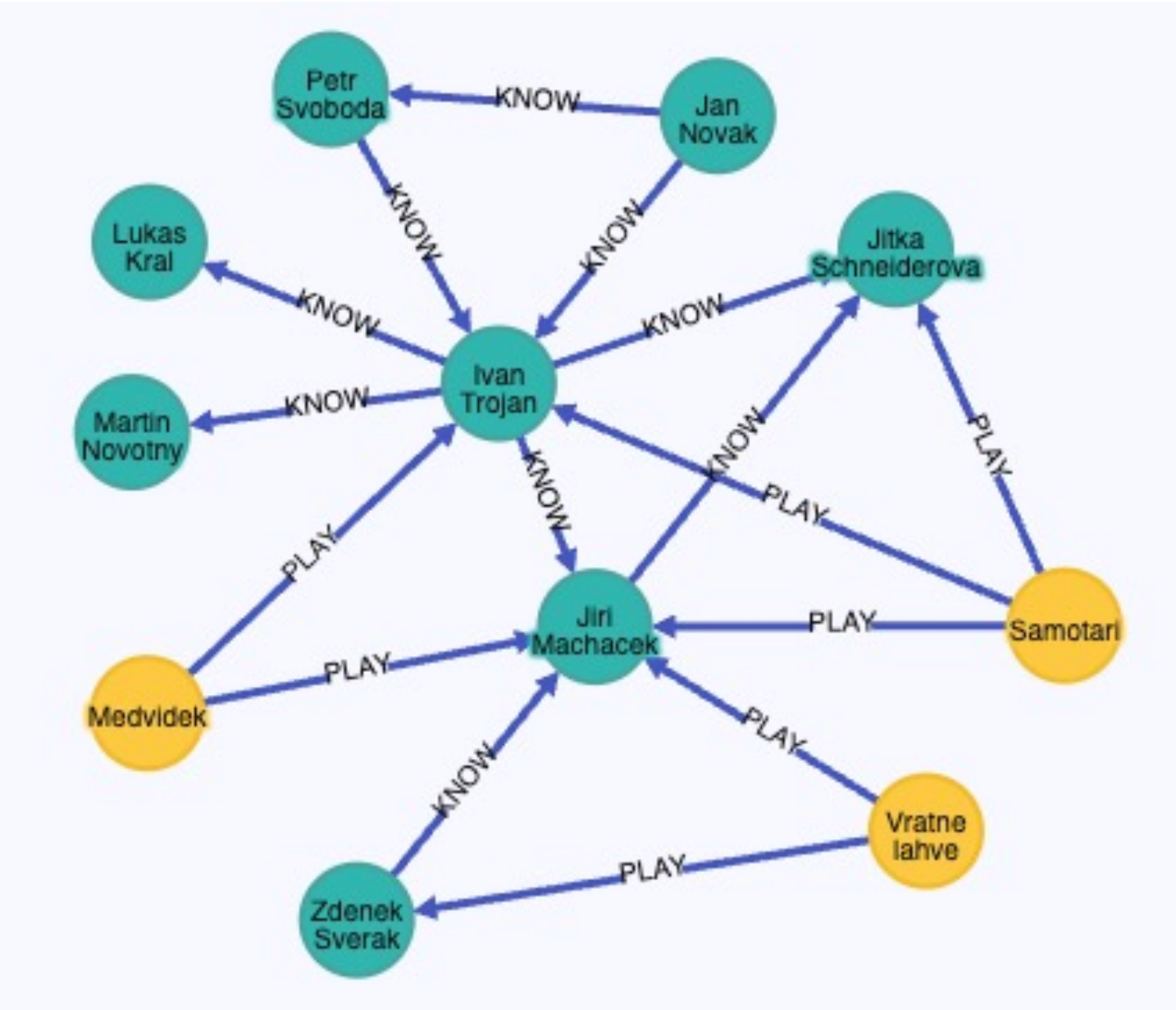
Source file

```
…
 (m1)-[c1:PLAY { role: "Robert Landa" }]->(a2),
 (m1)-[c2:PLAY { role: "Josef Tkaloun" }]->(a4),
 (m2)-[c3:PLAY { role: "Ondrej" }]->(a1),
 (m2)-[c4:PLAY { role: "Jakub" }]->(a2),
 (m2)-[c5:PLAY { role: "Hanka" }]->(a3),
 (m3)-[c6:PLAY { role: "Ivan" }]->(a1),
 (m3)-[c7:PLAY { role: "Jirka", award: "Czech Lion" }]->(a2),

 (a1)-[f1:KNOW]->(a2),
 (a1)-[f2:KNOW]->(a3),
 (a2)-[f3:KNOW]->(a3),
 (a4)-[f4:KNOW]->(a2),
 (a5)-[f5:KNOW]->(a6),
 (a6)-[f6:KNOW]->(a1),
 (a5)-[f7:KNOW]->(a1),
 (a1)-[f8:KNOW]->(a7),
 (a1)-[f9:KNOW]->(a8);
```

Source file


m1
Vratné lahve
2006
m2
Samotáři
2000
m3
Medvídek
2007
m4
Štěstí 2005
Ivan Trojan 1964
a1
Jiří Macháček
1966
a2
Jitka Schneiderová
1973
a3
Zdeněk Svěrák
1936
a4

Source file


Petr Svoboda
KNOW
Jan Novak
KNOW
KNOW
Lukas Kral
KNOW
Jitka Schneiderova
KNOW
Ivan Trojan
KNOW
Martin Novotny
KNOW
PLAY
KNOW
PLAY
PLAY
Jiri Machacek
PLAY
Samotari
PLAY
Medvidek
PLAY
PLAY
KNOW
Vratne lahve
PLAY
Zdenek Sverak

Exercise 1

Express the following Cypher query

- **Find movies with identifier *medvidek***
- Return movie nodes together with title properties

```
+-----------------------------------------------------------------------+
| m                                                          | m.title  |
+-----------------------------------------------------------------------+
| (:MOVIE {id: "medvidek", title: "Medvidek", year: 2007}) | "Medvidek" |
+-----------------------------------------------------------------------+
```

Exercise 1 - Solution

Express the following Cypher query

- **Find movies with identifier *medvidek***
- Return movie nodes together with title properties

```
MATCH (m:MOVIE {id: "medvidek"})
RETURN m, m.title;

MATCH (m:MOVIE)
  WHERE m.id = "medvidek"
RETURN m, m.title;
```

Exercise 2

Express the following Cypher query

- **Find actors born in *1965* or later**
- Return actor names and years they were born
- Sort the result using years (in descending order) and then names (in ascending order)

a.name	a.year
"Lukas Kral"	1980
"Martin Novotny"	1975
"Jitka Schneiderova"	1973
"Jan Novak"	1970
"Jiri Machacek"	1966
"Petr Svoboda"	1965

Exercise 2 - Solution

Express the following Cypher query

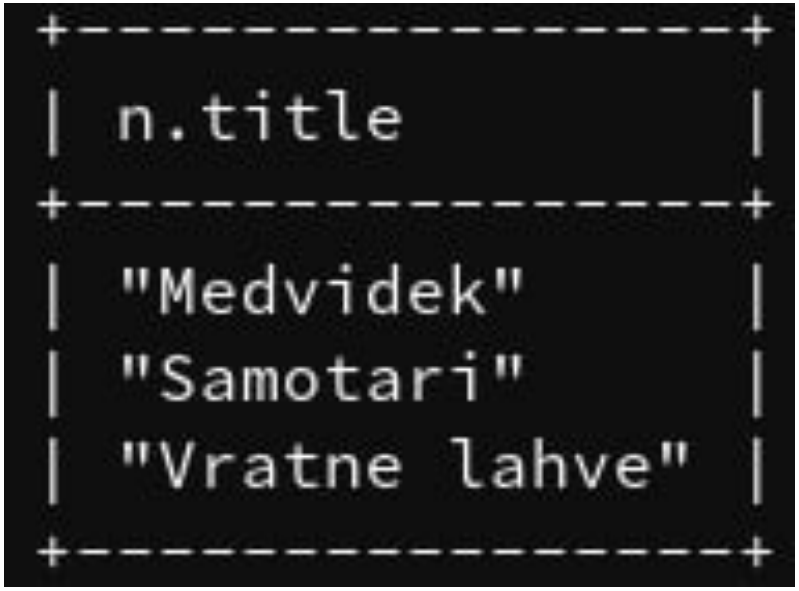
- **Find actors born in *1965* or later**
- Return actor names and years they were born
- Sort the result using years (in descending order) and then names (in ascending order)

```
MATCH (a:ACTOR)
  WHERE a.year >= 1965
RETURN a.name, a.year
  ORDER BY a.year DESC, a.name ASC;

... ORDER BY a.year DESCENDING, a.name ASCENDING;

... ORDER BY a.year DESCENDING, a.name;
```

Exercise 3

Express the following Cypher query

- **Find titles of movies in which *Jiri Machacek* played**



Exercise 3 - Solution

Express the following Cypher query

- **Find titles of movies in which *Jiri Machacek* played**

```
MATCH (:ACTOR {name: "Jiri Machacek"})<-[:PLAY]-(n:MOVIE)
RETURN n.title;

MATCH (n:MOVIE)-[:PLAY]->(:ACTOR {name: "Jiri Machacek"})
RETURN n.title;

MATCH (n:MOVIE)-[:PLAY]->(a:ACTOR)
  WHERE a.name = "Jiri Machacek"
RETURN n.title;
```

Exercise 3 - Solution

Express the following Cypher query

- **Find titles of movies in which *Jiri Machacek* played**

```
MATCH (:ACTOR {name: "Jiri Machacek"})<--(n:MOVIE)
RETURN n.title;

MATCH (:ACTOR {name: "Jiri Machacek"})--(n:MOVIE)
RETURN n.title;

MATCH (a:ACTOR {name: "Jiri Machacek"})
MATCH (n:MOVIE)-[:PLAY]->(a)
RETURN n.title;

MATCH (a:ACTOR {name: "Jiri Machacek"}), (n:MOVIE)-[:PLAY]->(a)
RETURN n.title;
```

Exercise 4

Express the following Cypher query

- **Find all actors whom Jiří Macháček knows**

Exercise 4 – Solution

```
MATCH (a:ACTOR { name: "Jiri Machacek" })-[:KNOW]->(knownActors:ACTOR)
RETURN knownActors.name AS Known_Actors;
```

```
+----------------------+
| Known_Actors         |
+----------------------+
| "Jitka Schneiderova" |
+----------------------+
```

```
MATCH (a:ACTOR { name: "Jiri Machacek" })-[:KNOW]-(knownActors:ACTOR)
RETURN knownActors.name AS Known_Actors;
```

```
+----------------------+
| Known_Actors         |
+----------------------+
| "Ivan Trojan"        |
| "Jitka Schneiderova" |
| "Zdenek Sverak"      |
+----------------------+
```

Exercise 5

Express the following Cypher query

- **Find movies where at least one actor played**

```
+-------------------------------------------------------------------------+
| m                                                                       |
+-------------------------------------------------------------------------+
| (:MOVIE {id: "vratnelahve", title: "Vratne lahve", year: 2006})         |
| (:MOVIE {id: "samotari", title: "Samotari", year: 2000})                |
| (:MOVIE {id: "medvidek", title: "Medvidek", year: 2007})                |
+-------------------------------------------------------------------------+
```

Exercise 5 - Solution

Express the following Cypher query

- **Find movies where at least one actor played**

```
MATCH (m:MOVIE)-[:PLAY]->(:ACTOR)
RETURN DISTINCT m;

MATCH (m:MOVIE)
  WHERE SIZE( [ (m)-[:PLAY]->(a:ACTOR) | a] ) >= 1
RETURN m;

MATCH (m:MOVIE)
  WHERE EXISTS( (m)-[:PLAY]->(:ACTOR) )
RETURN m;

MATCH (m:MOVIE)
  WHERE (m)-[:PLAY]->(:ACTOR)
RETURN m;
```

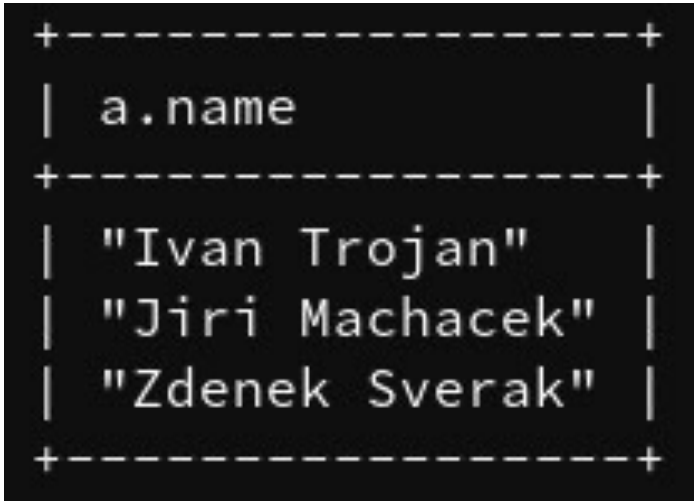
Exercise 5 - Solution

```
MATCH (m:MOVIE)
WITH m, SIZE( [(m)-[:PLAY]->(a:ACTOR) | a] ) AS actors
  WHERE actors >= 1
RETURN m;

MATCH (m:MOVIE)-[:PLAY]->(a:ACTOR)
WITH m, COUNT(a) as actors
  WHERE actors >= 1
RETURN m;

MATCH (m:MOVIE)-[:PLAY]->(a:ACTOR)
WITH m, COUNT(a) as actors
RETURN m;

MATCH (m:MOVIE), (a:ACTOR)
  WHERE (m)-[:PLAY]->(a)
RETURN DISTINCT m;
```

Exercise 6

Express the following Cypher query

- **Find actors who starred in movies released after 2005**



Exercise 6 - Solution

```
//Incorrect - creating new variable 'm' in pattern expression:
MATCH (a:ACTOR)
WHERE EXISTS((a)<-[:PLAY]-(m:MOVIE) WHERE m.year > 2005)
RETURN a.name


//CORRECT
MATCH (a:ACTOR)
WITH a, SIZE([(a)<-[:PLAY]-(m:MOVIE) WHERE m.year > 2005 | m]) as
movieCount
WHERE movieCount > 0
RETURN a.name;

MATCH (a:ACTOR)
MATCH (a)<-[:PLAY]-(m:MOVIE)
WHERE m.year > 2005
RETURN DISTINCT a.name;
```

Exercise 7

Express the following Cypher query

- **Find actors who played with *Ivan Trojan***

```
+-----------------------------------------------------------------------------+
| a                                                                           |
+-----------------------------------------------------------------------------+
| (:ACTOR {name: "Jiri Machacek", id: "machacek", year: 1966})                |
| (:ACTOR {name: "Jitka Schneiderova", id: "schneiderova", year: 1973}) |
+-----------------------------------------------------------------------------+
```

Exercise 7 - Solution

Express the following Cypher query

- **Find actors who played with *Ivan Trojan***

```
MATCH
  (s:ACTOR {name: "Ivan Trojan"})
    <-[:PLAY]-(m:MOVIE)-[:PLAY]->
  (a:ACTOR)
RETURN DISTINCT a;

MATCH
  (s:ACTOR {name: "Ivan Trojan"})<-[:PLAY]-(m:MOVIE),
  (m)-[:PLAY]->(a:ACTOR)
RETURN DISTINCT a;
```

Exercise 7 - Solution

Express the following Cypher query

- **Find actors who played with *Ivan Trojan***

```
MATCH (s:ACTOR {name: "Ivan Trojan"})<-[:PLAY]-(m:MOVIE)
MATCH (m)-[:PLAY]->(a:ACTOR)
  WHERE a <> s
RETURN DISTINCT a;

 ... WHERE a.name <> "Ivan Trojan"

MATCH (a:ACTOR)
  WHERE
    (a)<-[:PLAY]-(:MOVIE)-[:PLAY]->(:ACTOR {name: "Ivan Trojan"})
RETURN a;
```

Exercise 8

Express the following Cypher query

- **Find all friends of actor *Ivan Trojan***
- Include friends of friends etc.
- Return actor names

```
+----------------------+
| a.name               |
+----------------------+
| "Jiri Machacek"      |
| "Jitka Schneiderova" |
| "Zdenek Sverak"      |
| "Jan Novak"          |
| "Petr Svoboda"       |
| "Lukas Kral"         |
| "Martin Novotny"     |
+----------------------+
```

Exercise 8 - Solution

- **Find all friends of actor *Ivan Trojan***
- Include friends of friends etc.
- Return actor names

```
MATCH (s:ACTOR {name: "Ivan Trojan"})-[:KNOW *]-(a:ACTOR)
  WHERE s <> a
RETURN DISTINCT a.name;

MATCH (s:ACTOR {name: "Ivan Trojan"})-[:KNOW *1..]-(a:ACTOR)
  WHERE s <> a
RETURN DISTINCT a.name;

MATCH (a:ACTOR)
  WHERE
    EXISTS( (a)-[:KNOW *]-(:ACTOR {name: "Ivan Trojan"}) )
    AND
    (a.name <> "Ivan Trojan")
RETURN a.name;
```

Exercise 9

Express the following Cypher query

- **Find pairs of movies and their actors**
- Include movies without actors as well

m.title	a.name
"Vratne lahve"	"Zdenek Sverak"
"Vratne lahve"	"Jiri Machacek"
"Samotari"	"Jitka Schneiderova"
"Samotari"	"Jiri Machacek"
"Samotari"	"Ivan Trojan"
"Medvidek"	"Jiri Machacek"
"Medvidek"	"Ivan Trojan"
"Stesti"	NULL

Exercise 9 - Solution

Express the following Cypher query

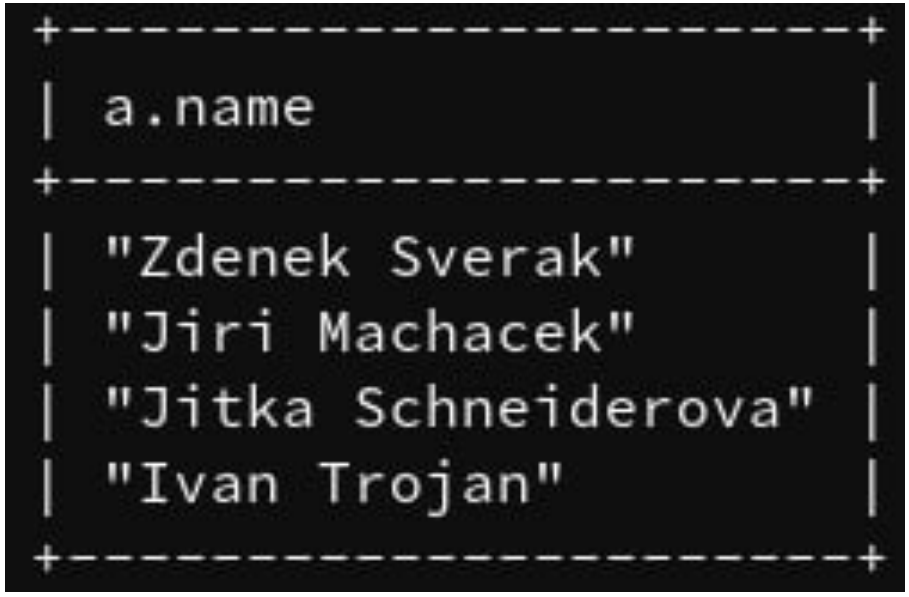
- **Find pairs of movies and their actors**
- Include movies without actors as well

```
MATCH (m:MOVIE)
OPTIONAL MATCH (m)-[:PLAY]->(a:ACTOR)
RETURN m.title, a.name;
```

Exercise 10

Express the following Cypher query

- **Find actors who played in movies having above average number of actors**
- Return actor names



```
+-----------------------+
| a.name                |
+-----------------------+
| "Zdenek Sverak"       |
| "Jiri Machacek"       |
| "Jitka Schneiderova"  |
| "Ivan Trojan"         |
+-----------------------+
```

Exercise 10 - Solution

Express the following Cypher query

- **Find actors who played in movies having above average number of actors**
- Return actor names

```
MATCH (m:MOVIE)
WITH m, SIZE( [(m)-[:PLAY]->(a:ACTOR) | a] ) AS actors
WITH AVG(actors) AS average
MATCH (m:MOVIE)
  WHERE SIZE( [(m)-[:PLAY]->(a:ACTOR) | a] ) > average
MATCH (m)-[:PLAY]->(a:ACTOR)
WITH DISTINCT a
RETURN a.name;
```

Exercise 10 - Solution

Express the following Cypher query

- **Find actors who played in movies having above average number of actors**
- Return actor names

```
MATCH (m:MOVIE)
OPTIONAL MATCH (m)-[:PLAY]->(a:ACTOR)
WITH m, COUNT(a) AS actors
WITH AVG(actors) AS average
MATCH (m:MOVIE)
  WHERE SIZE( [(m)-[:PLAY]->(a:ACTOR) | a] ) > average
MATCH (m)-[:PLAY]->(a:ACTOR)
WITH DISTINCT a
RETURN a.name;
```

Exercise 11

Express the following Cypher query

- **Find all actors who have played in the same movie as Jiří Macháček or know him**
- **Display actor names and movies, in which they played together**

Exercise 11 - Solution

```
MATCH (a1:ACTOR {name: "Jiri Machacek"})-[:PLAY]-(m:MOVIE)-[:PLAY]-(a2:ACTOR)
    WITH a2.name AS Actor, m.title AS Movies
    RETURN Actor, Movies;
```

Actor	Movies
"Zdenek Sverak"	"Vratne lahve"
"Ivan Trojan"	"Medvidek"
"Jitka Schneiderova"	"Samotari"
"Ivan Trojan"	"Samotari"

Exercise 11 - Solution

```
MATCH (a1:ACTOR {name: "Jiri Machacek"})-[:PLAY]-(m:MOVIE)-[:PLAY]-(a2:ACTOR)
    WITH a2.name AS Actor, COLLECT(DISTINCT m.title) AS Movies
    RETURN Actor, Movies;
```

Actor	Movies
"Zdenek Sverak"	["Vratne lahve"]
"Ivan Trojan"	["Medvidek", "Samotari"]
"Jitka Schneiderova"	["Samotari"]