

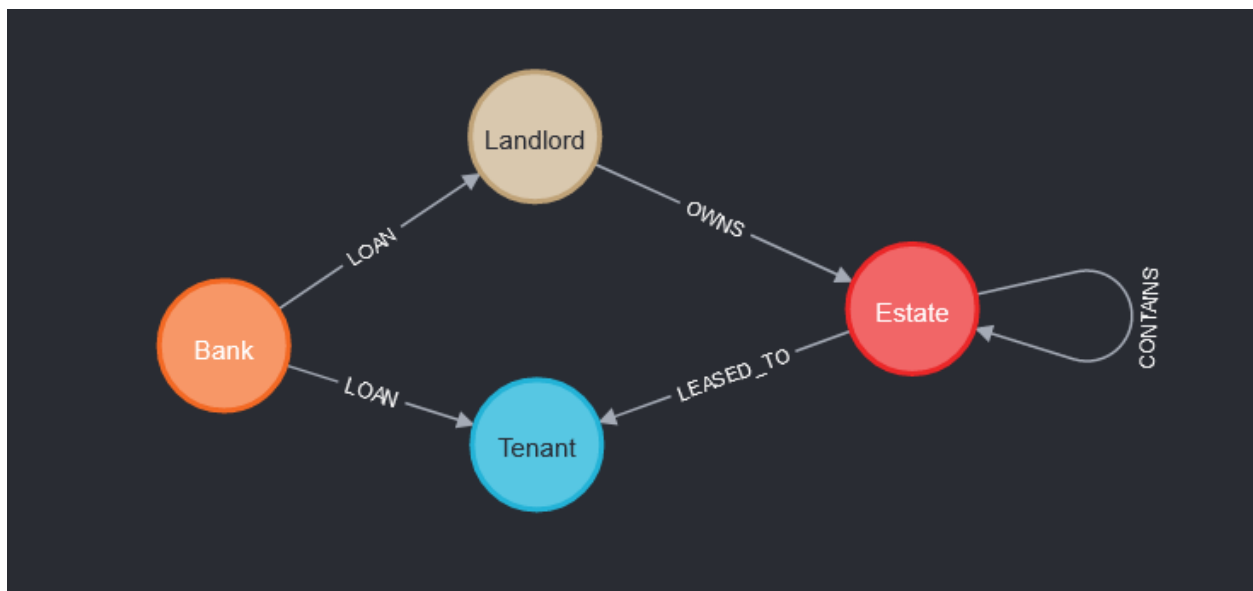
## Cvičení Neo4j

### Pracovní databáze

Pro cvičení jsem vymyslel jednoduchou databázi nájemníků (tenant) a pronajímatelů (landlord) v Praze a Brně. V databázi si držíme banky (id, název, kód), nájemníky a pronajímatele (jméno, příjmení, email, telefon) a nemovitosti (typ, adresa, cena). Některé osoby jsou jak nájemníci tak pronajímatelé. Banky můžou půjčovat peníze pronajímatelům (hypotékou či přes nemovitostní fond) a nájemníkům (jednorázová půjčka). U půjček nás zajímá částka a úroková sazba. Pronajímatel může vlastnit nemovitosti a nájemník jí může mít pronajatou (s danou měsíční cenou). Nemovitost může být byt, dům, garáž, sklep nebo pozemek, je daná nějakou adresou, a má nějakou cenu. Nemovitost může mít navíc další nemovitost jako součást sama sebe (například k bytu patří nějaká externí garáž). Databáze je udělaná primárně jako grafová, ale zároveň je dělaná tak, aby všechny úlohy šly udělat i v MongoDB pro srovnání. Z tohoto důvodu zde nebude moc úloh orientovaných čistě na práci s grafy.

Create script je v souboru: neo4j.txt

### Schéma databáze

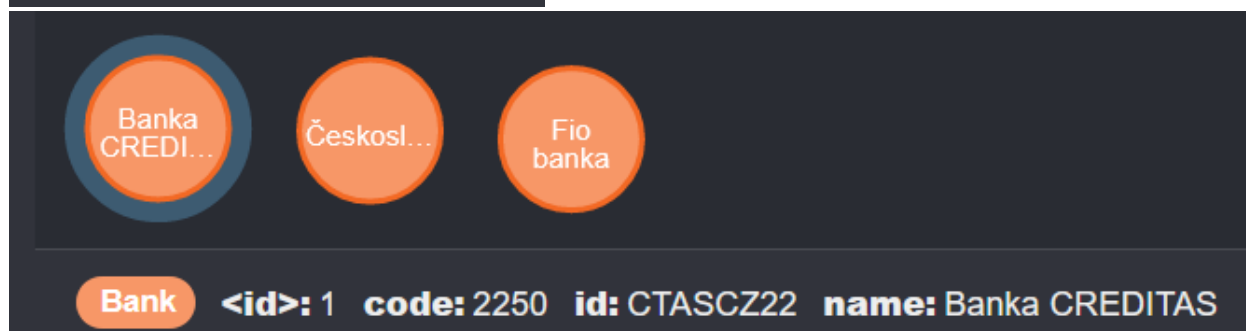


## Úkol 1 – Seznámení s daty

Postupně vypište všechny banky, pronajímatele, nájemníky a nemovitosti. Podívejte se, jaké mají data a poté najděte pronajímatele s telefonním číslem 745082420.

Banky:

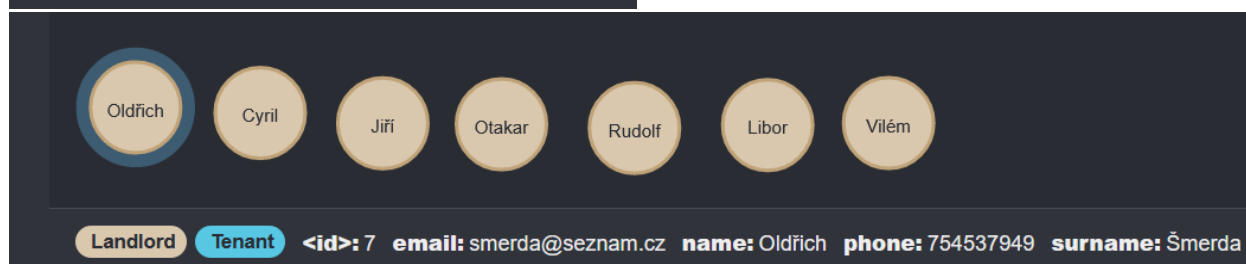
```
1 MATCH (m:Bank)
2 return m
```



Visual representation of the query results for the 'Bank' nodes. It shows three orange circular nodes labeled 'Banka CREDI...', 'Českosl...', and 'Fio banka'. The first node is highlighted with a blue border. Below the nodes, a summary bar shows the selected node type 'Bank' and its properties: <id>: 1, code: 2250, id: CTASCZ22, name: Banka CREDITAS.

Pronajímatelé:

```
1 MATCH (m:Landlord)
2 return m
```



Visual representation of the query results for the 'Landlord' nodes. It shows seven tan circular nodes labeled 'Oldřich', 'Cyril', 'Jiří', 'Otakar', 'Rudolf', 'Libor', and 'Vilém'. The first node 'Oldřich' is highlighted with a blue border. Below the nodes, a summary bar shows the selected node type 'Landlord' and its properties: <id>: 7, email: smerda@seznam.cz, name: Oldřich, phone: 754537949, surname: Šmerda.

Nájemníci:

```
1 MATCH (m:Tenant)
2 return m
```

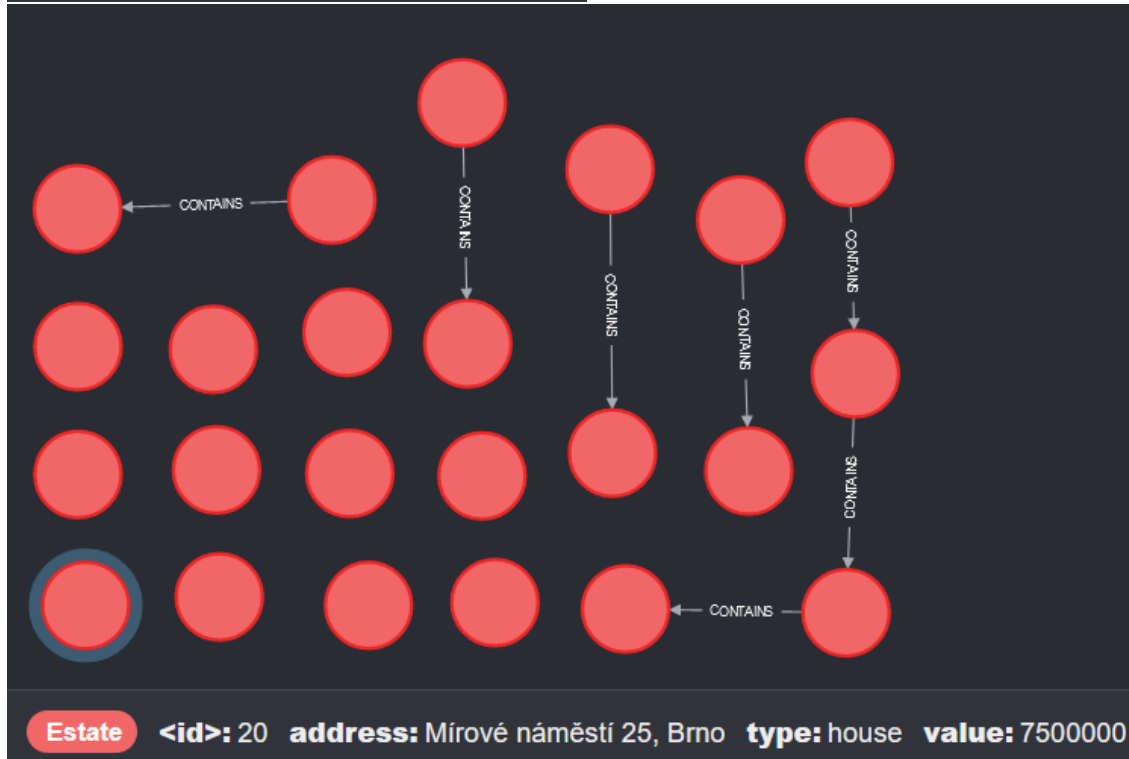


Visual representation of the query results for the 'Tenant' nodes. It shows two rows of circular nodes. The first row has five light blue nodes: 'Marcel', 'Viktor', 'Vratislav', 'Ivo', and 'Vilém'. The second row has six tan nodes: 'Oldřich', 'Otakar', 'Cyril', 'Viktor', 'Kristián', and 'Rostislav'. The first node 'Oldřich' in the second row is highlighted with a blue border. Below the nodes, a summary bar shows the selected node type 'Tenant' and its properties: <id>: 7, email: smerda@seznam.cz, name: Oldřich, phone: 754537949, surname: Šmerda.

V databázi máme 3 osoby (Oldřich, Otakar, Cyril), kteří jsou pronajímateli i nájemníky.

Nemovitosti:

```
1 MATCH (m:Estate)
2 return m
```



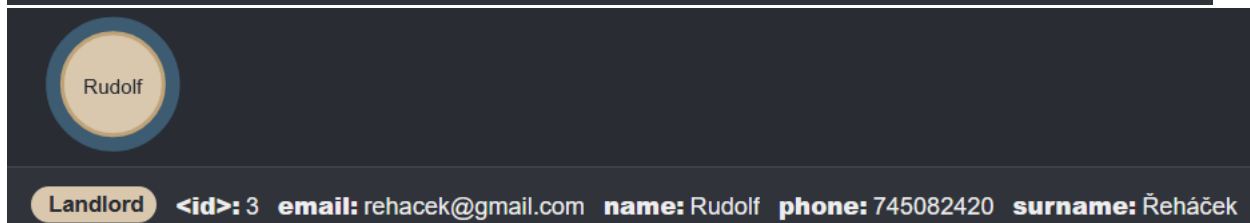
V databázi máme 4 nemovitosti (byty), které pod sebou mají další nemovitost (garáž) a jednu supernemovitost (byt + pozemek + garáž + sklep).

Pronajímatel s telefoním číslem 745082420:

```
1 MATCH (p:Landlord)
2   WHERE p.phone = "745082420"
3 RETURN p
```

nebo

```
1 MATCH (p:Landlord {phone: "745082420"})
2 RETURN p;
```



Úkol 2 – Najděte nemovitosti dražší než 4 miliony a seřadte je podle ceny od největší

```
1 match(e:Estate)
2   where e.value > 4000000
3 return e order by e.value desc
```

"e"

{"address":"Lipová 81, Praha","type":"house","value":9400000}

{"address":"Mírové náměstí 25, Brno","type":"house","value":7500000}

{"address":"Javorová 12, Brno","type":"house","value":5700000}

{"address":"Višňová 46, Brno","type":"flat","value":5130000}

{"address":"Třebblatská 131, Praha","type":"flat","value":4940000}

{"address":"Lesní 92, Praha","type":"land","value":4855000}

{"address":"Ploní 15, Brno","type":"land","value":4660000}

Úkol 3 – Najděte byty, které vlastní Jiří Bravenec

```
1 match(l:Landlord {name:"Jiří", surname:"Bravenec"})-[:OWNS]-(e:Estate)
2 return e
```

nebo

```
1 match(e:Estate)-[:OWNS]-(l:Landlord {name:"Jiří", surname:"Bravenec"})
2 return e
```

nebo

```
1 match(e:Estate)--(l:Landlord {name:"Jiří", surname:"Bravenec"})
2 return e
```

"e"

{"address":"Ondřejovská 86, Praha","type":"flat","value":3100000}

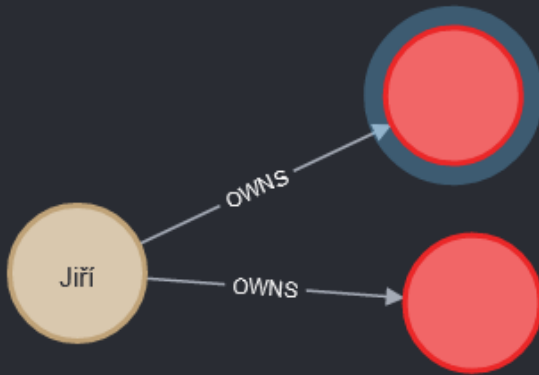
{"address":"Javorová 12, Brno","type":"house","value":5700000}

Grafově:

```
1 match(l:Landlord {name:"Jiří", surname:"Bravenec"})-[:OWNS]→(e:Estate)
2 return l, e
```

nebo

```
1 match(l:Landlord {name:"Jiří", surname:"Bravenec"})--(e:Estate)
2 return l, e
```



**Estate** <id>: 17 **address:** Javorová 12, Brno **type:** house **value:** 5700000

Úkol 4 – Najděte byty, které nikdo nevlastní (neznáme majitele), podnikovitosti neuvažujte

```
1 MATCH (e:Estate)
2   WHERE not EXISTS( (:Landlord)-[:OWNS]→(e) )
3   AND not EXISTS( (:Estate)-[:CONTAINS]→(e) )
4 return e;
```

nebo

```
1 MATCH (e:Estate)
2   WHERE not (:Landlord)-[:OWNS]→(e)
3   AND not (:Estate)-[:CONTAINS]→(e)
4 return e;
```

nebo

```
1 MATCH (e:Estate)
2   WHERE SIZE((:Landlord)-[:OWNS]→(e)) = 0
3   AND SIZE((:Estate)-[:CONTAINS]→(e)) = 0
4 return e;
```

Podmínka AND ... filtruje pryč podnemovitosti, které nemáme uvažovat. Pokud bychom je chtěli uvažovat, stačilo by k returnu přidat vypsání všech podnemovitostí (tím by se vypsali jen podnemovitosti nevlastněné nemovitosti). V Mongu by to už ale bylo horší.

"e"
{"address":"Mírové náměstí 25, Brno","type":"house","value":7500000}
{"address":"Sluneční 199, Brno","type":"flat","value":2850000}
{"address":"Višňová 46, Brno","type":"flat","value":5130000}

Úkol 5 – Najděte pronajímatele, kteří vlastní alespoň 1 byt

```
1 MATCH (l:Landlord)
2   WHERE SIZE((l)-[:OWNS]→(:Estate)) > 1
3 RETURN l;
```

Kdybychom ovšem chtěli vypsát i konkrétní počet, údaj už nemáme, co zkusit count:

```
1 MATCH (l:Landlord)-[:OWNS]→(e:Estate)
2   WHERE count(e) > 1
3 RETURN l, count(e);
```

Pokud to tak zkusíme, dostaneme chybu, že agregační funkci takto použít nelze. Musíme použít klauzuli WITH, a výsledek dotazu „poslat“ do dalšího dotazu, kde ji vyfiltrujeme přes WHERE:

```
1 MATCH (l:Landlord)-[:OWNS]→(e:Estate)
2   WITH l, count(e) as pocet_nemovitosti
3   WHERE pocet_nemovitosti > 1
4 RETURN l, pocet_nemovitosti;
```

"l"	"pocet_nemovitosti"
{"name": "Jiří", "phone": "740698061", "surname": "Bravenec", "email": "bravenec@seznam.cz"}	2
{"name": "Rudolf", "phone": "745082420", "surname": "Řeháček", "email": "rehacek@gmail.com"}	3
{"name": "Libor", "phone": "752693555", "surname": "Šašek", "email": "sasek@seznam.cz"}	2
{"name": "Vilém", "phone": "756881693", "surname": "Kotík", "email": "kotik@seznam.cz"}	2
{"name": "Otakar", "phone": "739986862", "surname": "Vacula", "email": "vacula@gmail.com"}	2

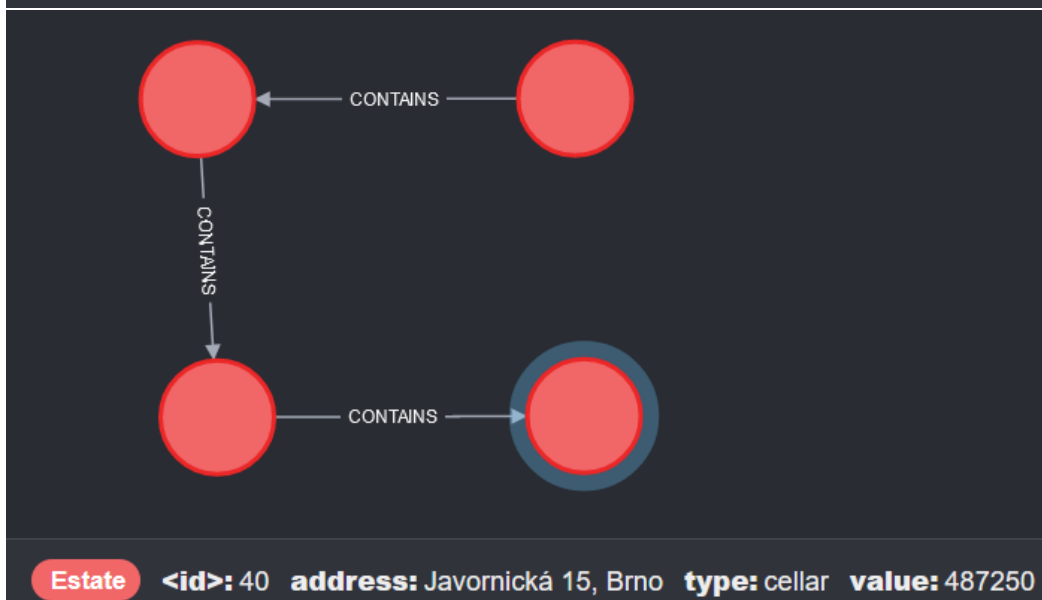
Úkol 6 – Najděte všechny supernemovitosti (nemovitosti, které pod sebou mají další nemovitosti a ty mají další a tak dále hloubky 4 a více) např. dům->pozemek->garáž->sklep a zobrazte celou tuto nemovitost jako graf

Nemovitosti hloubky 4 a více mají 3 a více „CONTAINS“ uzlů, můžeme proto napsat:

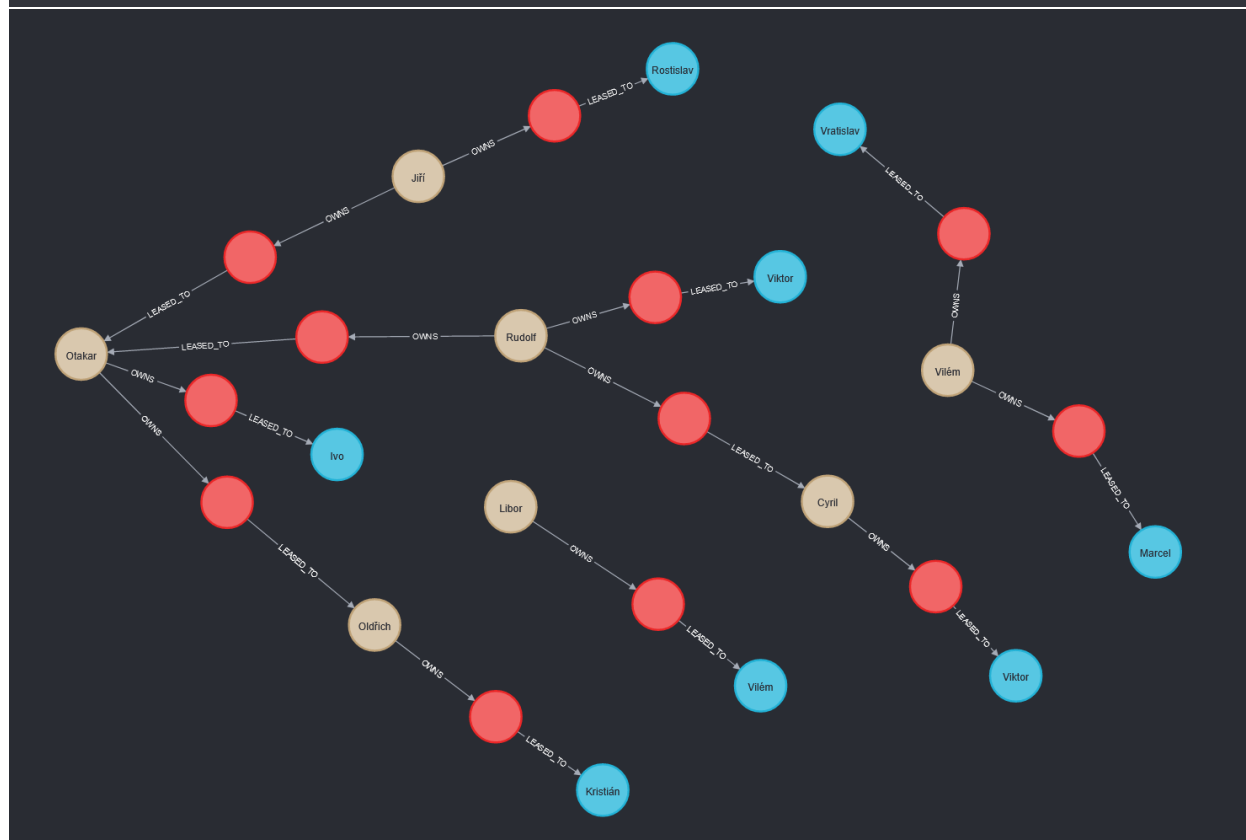
```
1 MATCH (e:Estate)-[:CONTAINS *3.. ]->(f:Estate)
2 return e, f;
```

Tím ovšem dostaneme pouze krajní hrany. Navíc nemovitost může být „strom“ a mít další podnemovitosti menší hloubky, které chceme vypsát. Celou nemovitost tedy vypíšeme jako:

```
1 MATCH (e:Estate)-[:CONTAINS *3.. ]->(f:Estate)
2 return (e)-[:CONTAINS *]-();
```



Úkol 7 – Vypište vztahy mezi pronajímateli a nájemníky do grafu

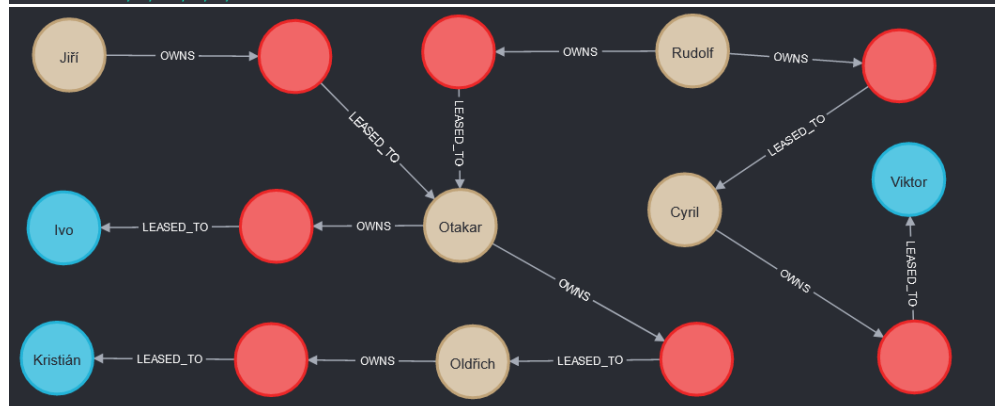


Úkol 8 – Vypište takové lidi, kteří zároveň vlastní nemovitost a jsou nájemníky v jiné  
Pokud by stačilo pouze vypsat osoby, stačí velmi jednoduchý:

```
1 MATCH (lt:Landlord:Tenant)
2 return lt;
```

Kdybychom chtěli i informace o tom, kdo je pronajal a komu osoba pronajímá, můžeme využít například:

```
1 MATCH (l:Landlord)-[:OWNS]→(e:Estate)-[:LEASED_TO]→(lt:Landlord)-[:OWNS]→(f:Estate)-[:LEASED_TO]→(t:Tenant)
2 return l,e,lt,f,t;
```





Úkol 9 – Vypište lidi, kteří profitují na tom, že jsou nájemníky někde jinde a vypište jejich profit, daně zanedbejte

Zajímá nás tedy rozdíl v příjmech z pronájmu vs ztráty z nájmu, první nástřel:

```
1 MATCH (l:Landlord)-[:OWNS]-(e:Estate)-[loss:LEASED_TO]-(lt:Landlord)-[:OWNS]-(f:Estate)-[profit:LEASED_TO]-(t:Tenant)
2 WHERE profit.monthly > loss.monthly
3 return lt, (profit.monthly - loss.monthly);
```

Je špatně, protože jeden člověk může vlastnit i bydlet ve více nemovitostech, tohle by fungovalo pouze pokud by mohl vlastnit maximálně jednu. Zkusíme tedy využít sumu:

```
1 MATCH (l:Landlord)-[:OWNS]-(e:Estate)-[loss:LEASED_TO]-(lt:Landlord)-[:OWNS]-(f:Estate)-[profit:LEASED_TO]-(t:Tenant)
2 WITH lt, sum(profit.monthly) as profit, sum(loss.monthly) as loss
3 WHERE profit > loss
4 return lt, profit, loss;
```

V tomto případě je to také špatně, protože se cesty překrývají a profity i ztráty se započítávají vícekrát.

Musíme tedy využít distinct:

```
1 MATCH (l:Landlord)-[:OWNS]-(e:Estate)-[loss:LEASED_TO]-(lt:Landlord)-[:OWNS]-(f:Estate)-[profit:LEASED_TO]-(t:Tenant)
2 WITH lt, sum(distinct profit.monthly) as profit, sum(distinct loss.monthly) as loss
3 WHERE profit > loss
4 return lt, profit - loss as free_money;
```

"lt"	"free_money"
{"name": "Oldřich", "phone": "754537949", "surname": "Šmerda", "email": "smerda@seznam.cz"}	19830

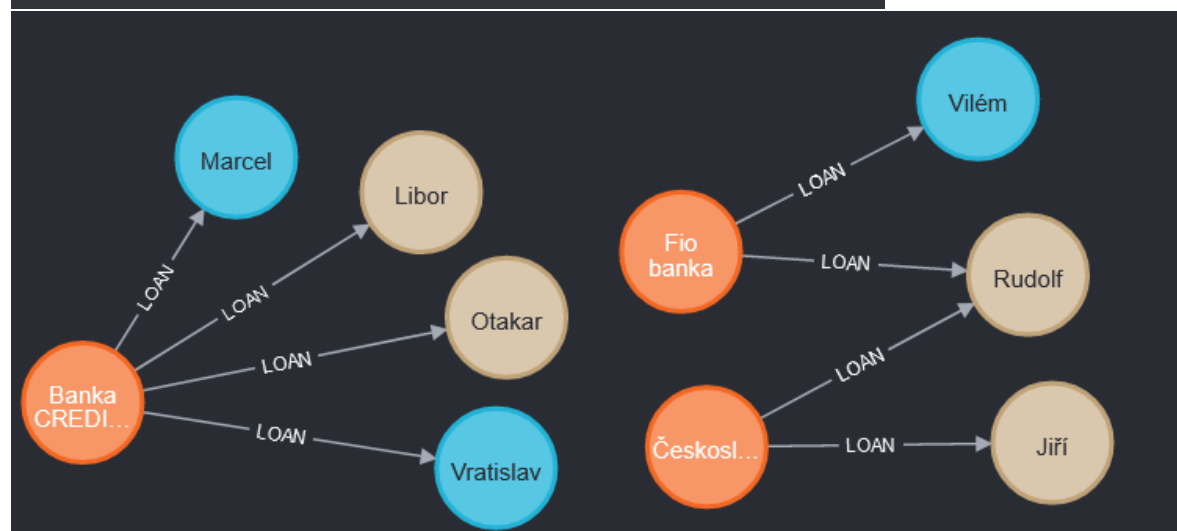
Úkol 10 – Zobrazte v grafu všechny půjčky

Půjčky si můžou vzít jak nájemníci, tak pronajímatelé, můžeme zkusit:

```
1 MATCH (b:Bank)-[:LOAN]-(lt:Landlord|Tenant)
2 return b, lt;
```

Takto ovšem nedostaneme „Landlord or Tenant“, ale „Landlord and Tenant“, jelikož chceme všechny, můžeme použít obecné x:

```
1 MATCH (b:Bank)-[:LOAN]-(x)
2 return b, x;
```



Úkol 11 – Zjistěte průměrný úrok u všech půjček a u nemovitostních fondů

Obecně:

```
1 MATCH (b:Bank)-[l:LOAN]→(x)
2 return avg(l.interest);
```

avg(l.interest)

0.03875

Fond (drobné problémy s doublem):

```
1 MATCH (b:Bank)-[l:LOAN {type: "Nemovitostní fond"}]→(x)
2 return avg(l.interest);
```

avg(l.interest)

0.037500000000000006

Úkol 12 – Zjistěte, jestli pronajímatelé profitují na svých půjčkách, výdaje zanedbejte, daň z příjmu předpokládejte 15% bez odpisů

Budeme postupovat podobně jako u zjišťování profitů nájemník – pronajímatel, akorát musíme udělat trochu jiné výpočty:

```
1 MATCH (:Bank)-[loss:LOAN]→(la:Landlord)-[:OWNS]→(:Estate)-[profit:LEASED_TO]→(:Tenant)
2 WHERE loss.type = "Nemovitostní fond" or loss.type = "Hypotéka"
3 WITH la, sum(distinct profit.monthly * 0.85) as prof, sum(distinct loss.amount * loss.interest / 12) as los
4 return la, prof as profit, los as loss, prof-loss as total;
```

Zajímají nás tedy půjčky z fondu nebo hypotéka, a potřebujeme zjistit profit, což je součet všech pronajatých bytů (bez duplicit) \* 0.85 což nám strhne 15%. Ztrátu vypočítáme jako celkovou výši půjčky \* úrok / 12, protože úroky jsou ročně. Pak už vše stačí jen vypsát.

"la"	"profit"	"loss"	"total"
{"name":"Jiří","phone":"740698061","surname":"Bravenec","email":"bravenec@seznam.cz"}	51807.5	16666.666666666668	35140.83333333333
{"name":"Rudolf","phone":"745082420","surname":"Řeháček","email":"rehacek@gmail.com"}	64132.5	25000.0	39132.5
{"name":"Libor","phone":"752693555","surname":"Šašek","email":"sasek@seznam.cz"}	19125.0	5625.0	13500.0
{"name":"Otakar","phone":"739986862","surname":"Vacula","email":"vacula@gmail.com"}	35190.0	7500.0	27690.0

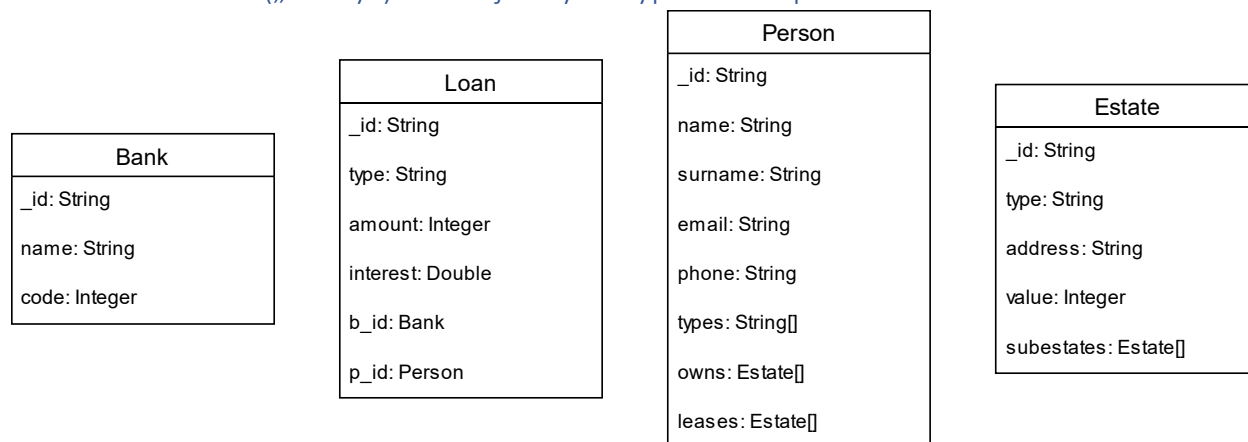
## Cvičení MongoDB

### Pracovní databáze

Pro srovnání MongoDB a Neo4j zůstává databáze téměř stejná (snažil jsem se ji co nejlépe převést do JSONu, jak si myslím, že by to mělo být). Převedení z Neo4j do MongoDB je následující: Uzly (Bank, Tenant, Landlord, Estate) jsou kolekce „entit“. Jelikož v Neo4j může být uzel Tenant is Landlord zároveň, rozhodl jsem se udělat kolekci People, která má vždy atribut type, což je pole stringů, které určuje, jestli je Person „tenant“ nebo „landlord“ nebo obojí. V Neo4j jsme měli také několik „relací“ (hran). Hrana LOAN odpovídá many-to-many vazbě a proto je v MongoDB řešena jako samostatná kolekce, která má vždy id banky a id osoby + ostatní vlastnosti, které byly obsaženy v hraně. Hraný OWNS, LEASED\_TO a CONTAINS odpovídají one-to-many (byt patří vždy pouze jedné osobě a může ho pronajímat vždy jen jedna osoba, nemovitost patří vždy pouze pod jednu nemovitost [ale nemovitost má více podnemovitostí]) a proto je tohle řešené pomocí pole, kde každá položka je vazba (má id na osobu / nemovitost a další atributy dle potřeby hrany). Databáze byla dělaná pro Neo4j a proto některé úlohy mají poměrně krkolomné řešení, na které se Mongo nehodí (nebo jsem nepřišel na jednodušší řešení).

Create script je v souboru: mongo.txt

### Schéma databáze („entity“) neboli jak by to vypadalo v aplikaci



### Úkol 1 – Seznámení s daty

Postupně vypište všechny banky, pronajímatele, nájemníky a nemovitosti. Podívejte se, jaké mají data a poté najděte pronajímatele s telefonním číslem 745082420.

Banky:

```
> db.banks.find().pretty()
{
  "_id" : "CEKOCZPP",
  "name" : "Československá obchodní banka",
  "code" : 300
}
{ "_id" : "FIOBCZPP", "name" : "Fio banka", "code" : 2010 }
{ "_id" : "CTASCZ22", "name" : "Banka CREDITAS", "code" : 2250 }
```

## Pronajímatelé:

```
> db.people.find({types:"landlord"}).pretty()
{
  "_id" : "24",
  "name" : "Jiří",
  "surname" : "Bravenec",
  "email" : "bravenec@seznam.cz",
  "phone" : "740698061",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "1"
    },
    {
      "e_id" : "4"
    }
  ],
  "leases" : [
    {
    }
  ]
}

{
  "_id" : "25",
  "name" : "Rudolf",
  "surname" : "Reháček",
  "email" : "rehacek@gmail.com",
  "phone" : "745082420",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "2"
    },
    {
      "e_id" : "19"
    },
    {
      "e_id" : "20"
    }
  ],
  "leases" : [
    {
    }
  ]
}

{
  "_id" : "26",
  "name" : "Libor",
  "surname" : "Šašek",
  "email" : "sasek@seznam.cz",
  "phone" : "752693555",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "5"
    },
    {
      "e_id" : "6"
    }
  ],
  "leases" : [
    {
    }
  ]
}

{
  "_id" : "27",
  "name" : "Vilém",
  "surname" : "Kotík",
  "email" : "kotik@seznam.cz",
  "phone" : "756881693",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "8"
    },
    {
      "e_id" : "9"
    }
  ],
  "leases" : [
    {
    }
  ]
}

{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "email" : "vacula@gmail.com",
  "phone" : "739986862",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "10"
    },
    {
      "e_id" : "18"
    }
  ],
  "leases" : [
    {
      "e_id" : "1",
      "monthly" : 34500
    },
    {
      "e_id" : "20",
      "monthly" : 13230
    }
  ]
}

{
  "_id" : "29",
  "name" : "Oldřich",
  "surname" : "Šmerda",
  "email" : "smerda@seznam.cz",
  "phone" : "754537949",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "11"
    }
  ],
  "leases" : [
    {
      "e_id" : "18",
      "monthly" : 12650
    }
  ]
}

{
  "_id" : "30",
  "name" : "Cyril",
  "surname" : "Honzík",
  "email" : "honzik@gmail.com",
  "phone" : "749415819",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "17"
    }
  ],
  "leases" : [
    {
      "e_id" : "19",
      "monthly" : 15470
    }
  ]
}
```

## Nájemníci:

```
> db.people.find({types:"tenant"}).pretty()
{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "email" : "vacula@gmail.com",
  "phone" : "739986862",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "10"
    },
    {
      "e_id" : "18"
    }
  ],
  "leases" : [
    {
      "e_id" : "1",
      "monthly" : 34500
    },
    {
      "e_id" : "20",
      "monthly" : 13230
    }
  ]
},
{
  "_id" : "29",
  "name" : "Oldřich",
  "surname" : "Šmerda",
  "email" : "smerda@seznam.cz",
  "phone" : "754537949",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "11"
    }
  ],
  "leases" : [
    {
      "e_id" : "18",
      "monthly" : 12650
    }
  ]
},
{
  "_id" : "30",
  "name" : "Cyril",
  "surname" : "Hončík",
  "email" : "honzik@gmail.com",
  "phone" : "749415819",
  "types" : [
    "landlord",
    "tenant"
  ],
  "owns" : [
    {
      "e_id" : "17"
    }
  ],
  "leases" : [
    {
      "e_id" : "19",
      "monthly" : 15470
    }
  ]
},
{
  "_id" : "31",
  "name" : "Viktor",
  "surname" : "Steklý",
  "email" : "stekly@seznam.cz",
  "phone" : "737047921",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    },
    {
      "e_id" : "2",
      "monthly" : 46750
    }
  ],
  "leases" : [
    {
    }
  ]
},
{
  "_id" : "32",
  "name" : "Rostislav",
  "surname" : "Šeda",
  "email" : "seda@gmail.com",
  "phone" : "739814131",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "4",
      "monthly" : 26450
    }
  ]
},
{
  "_id" : "33",
  "name" : "Vilém",
  "surname" : "Balek",
  "email" : "balek@seznam.cz",
  "phone" : "744019103",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "6",
      "monthly" : 22500
    }
  ]
},
{
  "_id" : "34",
  "name" : "Vratislav",
  "surname" : "Krejša",
  "email" : "krejsa@gmail.com",
  "phone" : "754276241",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "8",
      "monthly" : 18700
    }
  ]
},
{
  "_id" : "35",
  "name" : "Marcel",
  "surname" : "Uhlik",
  "email" : "uhlik@seznam.cz",
  "phone" : "752183651",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "9",
      "monthly" : 27500
    }
  ]
},
{
  "_id" : "36",
  "name" : "Ivo",
  "surname" : "Chlumský",
  "email" : "chlumsky@hotmail.com",
  "phone" : "747283444",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "10",
      "monthly" : 28750
    }
  ]
},
{
  "_id" : "37",
  "name" : "Kristián",
  "surname" : "Zabka",
  "email" : "zabka@seznam.cz",
  "phone" : "752055869",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "11",
      "monthly" : 32480
    }
  ]
},
{
  "_id" : "38",
  "name" : "Viktor",
  "surname" : "Slány",
  "email" : "slany@seznam.cz",
  "phone" : "753350534",
  "types" : [
    "tenant"
  ],
  "owns" : [
    {
    }
  ],
  "leases" : [
    {
      "e_id" : "17",
      "monthly" : 6440
    }
  ]
}
```

V databázi máme 3 osoby (Oldřich, Otakar, Cyril), kteří jsou pronajímateli i nájemníky.

Nemovitosti:

```
> db.estates.find()
{ "_id" : "2", "type" : "house", "address" : "Lipová 81, Praha", "value" : 9400000, "subestates" : [ ] }
{ "_id" : "3", "type" : "house", "address" : "Mírové náměstí 25, Brno", "value" : 7500000, "subestates" : [ ] }
{ "_id" : "4", "type" : "flat", "address" : "Ondřejovská 86, Praha", "value" : 3100000, "subestates" : [ ] }
{ "_id" : "5", "type" : "flat", "address" : "Pionýrská 15, Brno", "value" : 2985000, "subestates" : [ ] }
{ "_id" : "6", "type" : "flat", "address" : "Podhornická 139, Praha", "value" : 2423000, "subestates" : [ ] }
{ "_id" : "7", "type" : "flat", "address" : "Sluneční 199, Brno", "value" : 2850000, "subestates" : [ ] }
{ "_id" : "8", "type" : "flat", "address" : "Spojovací 42, Praha", "value" : 2350000, "subestates" : [ ] }
{ "_id" : "17", "type" : "land", "address" : "Vládní 15, Praha", "value" : 1970000, "subestates" : [ ] }
{ "_id" : "18", "type" : "land", "address" : "Družstevní 133, Brno", "value" : 3320000, "subestates" : [ ] }
{ "_id" : "19", "type" : "land", "address" : "Lesní 92, Praha", "value" : 4855000, "subestates" : [ ] }
{ "_id" : "20", "type" : "land", "address" : "Ploní 15, Brno", "value" : 4660000, "subestates" : [ ] }
{ "_id" : "9", "type" : "flat", "address" : "Struhařovská 84, Brno", "value" : 3240000, "subestates" : [ { "e_id" : "13" } ] }
{ "_id" : "13", "type" : "garage", "address" : "Struhařovská 84, Praha", "value" : 450000, "subestates" : [ ] }
{ "_id" : "10", "type" : "flat", "address" : "Školní 31, Praha", "value" : 3480000, "subestates" : [ { "e_id" : "14" } ] }
{ "_id" : "14", "type" : "garage", "address" : "Školní 31, Praha", "value" : 295000, "subestates" : [ ] }
{ "_id" : "11", "type" : "flat", "address" : "Třeblátská 131, Praha", "value" : 4940000, "subestates" : [ { "e_id" : "15" } ] }
{ "_id" : "15", "type" : "garage", "address" : "Třeblátská 131, Praha", "value" : 248000, "subestates" : [ ] }
{ "_id" : "12", "type" : "flat", "address" : "Višňová 46, Brno", "value" : 5130000, "subestates" : [ { "e_id" : "16" } ] }
{ "_id" : "16", "type" : "garage", "address" : "Višňová 46, Brno", "value" : 345000, "subestates" : [ ] }
{ "_id" : "1", "type" : "house", "address" : "Javorová 12, Brno", "value" : 5700000, "subestates" : [ { "e_id" : "21" } ] }
{ "_id" : "21", "type" : "land", "address" : "Javornická 15, Brno", "value" : 495000, "subestates" : [ { "e_id" : "22" } ] }
{ "_id" : "22", "type" : "garage", "address" : "Javornická 15, Brno", "value" : 325000, "subestates" : [ { "e_id" : "23" } ] }
{ "_id" : "23", "type" : "cellar", "address" : "Javornická 15, Brno", "value" : 487250, "subestates" : [ ] }
```

V databázi máme 4 nemovitosti (byty), které pod sebou mají další nemovitost (garáž) a jednu supernemovitost (byt + pozemek + garáž + sklep).

Pronajímatel s telefoním číslem 745082420:

```
> db.people.find({types:"landlord", phone:"745082420"}).pretty()
{
  "_id" : "25",
  "name" : "Rudolf",
  "surname" : "Řeháček",
  "email" : "rehacek@gmail.com",
  "phone" : "745082420",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "2"
    },
    {
      "e_id" : "19"
    },
    {
      "e_id" : "20"
    }
  ],
  "leases" : [
    {
    }
  ]
}
```

Úkol 2 – Najděte nemovitosti dražší než 4 miliony a seřadte je podle ceny od největší

```
> db.estates.find({value:{>:4000000}}).sort({value:-1})
{ "_id" : "2", "type" : "house", "address" : "Lipová 81, Praha", "value" : 9400000, "substates" : [ ] }
{ "_id" : "3", "type" : "house", "address" : "Mírové náměstí 25, Brno", "value" : 7500000, "substates" : [ ] }
{ "_id" : "1", "type" : "house", "address" : "Javorová 12, Brno", "value" : 5700000, "substates" : [ { "e_id" : "21" } ] }
{ "_id" : "12", "type" : "flat", "address" : "Višňová 46, Brno", "value" : 5130000, "substates" : [ { "e_id" : "16" } ] }
{ "_id" : "11", "type" : "flat", "address" : "Třebelatská 131, Praha", "value" : 4940000, "substates" : [ { "e_id" : "15" } ] }
{ "_id" : "19", "type" : "land", "address" : "Lesní 92, Praha", "value" : 4855000, "substates" : [ ] }
{ "_id" : "20", "type" : "land", "address" : "Ploní 15, Brno", "value" : 4660000, "substates" : [ ] }
```

Úkol 3 – Najděte byty, které vlastní Jiří Bravenec

Jelikož Person a Estates mají one-to-many vazbu, musíme udělat něco jako join, tedy vytahat si z Estates všechny nemovitosti, které se shodují s e\_id v owns[]. Pak už stačí jednoduchý selekt a projekce.

```
> db.people.aggregate([
...   {$lookup:{
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "nemovitosti"
...   }},
...   {$match:{
...     name: "Jiří",
...     surname: "Bravenec"
...   }},
...   {$project:{
...     nemovitosti: 1
...   }}
... ]).pretty()
{
  "_id" : "24",
  "nemovitosti" : [
    {
      "_id" : "1",
      "type" : "house",
      "address" : "Javorová 12, Brno",
      "value" : 5700000,
      "substates" : [
        {
          "e_id" : "21"
        }
      ]
    },
    {
      "_id" : "4",
      "type" : "flat",
      "address" : "Ondřejovská 86, Praha",
      "value" : 3100000,
      "substates" : [ ]
    }
  ]
}
```

Pokud neuděláme projekci, dostaneme data i s informacemi o vlastníkovi:

```
> db.people.aggregate([
...   {$lookup:{
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "nemovitosti"
...   }},
...   {$match:{
...     name: "Jiří",
...     surname: "Bravenec"
...   }}
... ]).pretty()
{
  "_id" : "24",
  "name" : "Jiří",
  "surname" : "Bravenec",
  "email" : "bravenec@seznam.cz",
  "phone" : "740698061",
  "types" : [
    "landlord"
  ],
  "owns" : [
    {
      "e_id" : "1"
    },
    {
      "e_id" : "4"
    }
  ],
  "leases" : [
    {

    }
  ],
  "nemovitosti" : [
    {
      "_id" : "1",
      "type" : "house",
      "address" : "Javorová 12, Brno",
      "value" : 5700000,
      "substates" : [
        {
          "e_id" : "21"
        }
      ]
    },
    {
      "_id" : "4",
      "type" : "flat",
      "address" : "Ondřejovská 86, Praha",
      "value" : 3100000,
      "substates" : [ ]
    }
  ]
}
```



Úkol 4 – Najděte nemovitosti, které nikdo nevlastní (neznáme majitele), podnemovitosti neuvažujte

Uděláme si proměnou `unwanted_a`, do které dáme id všech nemovitostí, které někdo vlastní. V agregaci používáme `unwind`, který z pole nemovitostí udělá jednotlivé hodnoty a ostatní nakopíruje. Pokud bychom měli třeba dvojici `{_id: 1, nemovitosti: [5, 6]}`, `unwind` na nemovitost by udělal dvě samostatné dvojice `(1, 5)`, `(1, 6)` a díky tomu k nim jde jednoduše přistupovat přes tečkovou notaci.

Poté uděláme v podstatě to samé pro `unwanted_b`, akorát tam přijdou id všech podnemovitostí. Vezmeme tedy každou nemovitost, podíváme se, jestli má nějakou podnemovitost a pokud ano, přidáme ji do výsledku.

Na konec stačí projet kolekci `estates` a vyfiltrovat všechny nemovitosti, které mají id v `unwanted_a` nebo `unwanted_b` a zbydou nám pouze ty, které nikdo nevlastní.

```
> var unwanted_a = db.people.aggregate([
...   {$lookup:{
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "nemovitosti"
...   }},
...   {$unwind: "$nemovitosti"},
...   {$project:{
...     _id: 0,
...     "nemovitosti._id": 1
...   }}
... ]).toArray().map( function(u) { return u.nemovitosti._id; } )
>
> var unwanted_b = db.estates.aggregate([
...   {$match:{
...     "substates.0":{$exists:true}
...   }},
...   {$unwind: "$substates"},
...   {$project:{
...     _id: 0,
...     substates: 1
...   }}
... ]).toArray().map( function(u) { return u.substates.e_id; } )
>
> db.estates.find(
...   {
...     $and: [
...       { _id: { $nin: unwanted_a } },
...       { _id: { $nin: unwanted_b } }
...     ]
...   }
... )
{ "_id" : "12", "type" : "flat", "address" : "Višňová 46, Brno", "value" : 5130000, "substates" : [ { "e_id" : "16" } ] }
{ "_id" : "3", "type" : "house", "address" : "Mírové náměstí 25, Brno", "value" : 7500000, "substates" : [ ] }
{ "_id" : "7", "type" : "flat", "address" : "Sluneční 199, Brno", "value" : 2850000, "substates" : [ ] }
```

Úkol 5 – Najděte pronajímatele, kteří vlastní více než 1 byt

Uděláme opět join lidí s vlastněnými nemovitostmi. V tomto joinu budeme mít pole nemovitostí. Více než jeden je v podstatě 2 a více, a jelikož se indexuje od nuly, pokud na indexu 1 v poli nemovitostí existuje hodnota, má pronajímatel alespoň 2 byty. Pak dle libosti vyfiltrujeme jen zajímavé parametry.

```
> db.people.aggregate([
...   {$lookup:{
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "nemovitosti"
...   }},
...   {$match:{
...     "nemovitosti.1":{$exists:true}
...   }},
...   {$project:{
...     name: 1,
...     surname: 1,
...     email: 1,
...     phone: 1,
...     "nemovitosti.type": 1,
...     "nemovitosti.address": 1,
...     "nemovitosti.value": 1
...   }}
... ]).pretty()
{
  "_id" : "24",
  "name" : "Jiří",
  "surname" : "Bravenec",
  "email" : "bravenec@seznam.cz",
  "phone" : "740698061",
  "nemovitosti" : [
    {
      "type" : "house",
      "address" : "Javorová 12, Brno",
      "value" : 5700000
    },
    {
      "type" : "flat",
      "address" : "Ondřejovská 86, Praha",
      "value" : 3100000
    }
  ]
}

{
  "_id" : "25",
  "name" : "Rudolf",
  "surname" : "Řeháček",
  "email" : "rehacek@gmail.com",
  "phone" : "745082420",
  "nemovitosti" : [
    {
      "type" : "land",
      "address" : "Lesní 92, Praha",
      "value" : 4855000
    },
    {
      "type" : "house",
      "address" : "Lipová 81, Praha",
      "value" : 9400000
    },
    {
      "type" : "land",
      "address" : "Ploní 15, Brno",
      "value" : 4660000
    }
  ]
}

{
  "_id" : "26",
  "name" : "Libor",
  "surname" : "Šašek",
  "email" : "sasek@seznam.cz",
  "phone" : "752693555",
  "nemovitosti" : [
    {
      "type" : "flat",
      "address" : "Pionýrská 15, Brno",
      "value" : 2985000
    },
    {
      "type" : "flat",
      "address" : "Podhornická 139, Praha",
      "value" : 2423000
    }
  ]
}

{
  "_id" : "27",
  "name" : "Vilém",
  "surname" : "Kotík",
  "email" : "kotik@seznam.cz",
  "phone" : "756881693",
  "nemovitosti" : [
    {
      "type" : "flat",
      "address" : "Spojovací 42, Praha",
      "value" : 2350000
    },
    {
      "type" : "flat",
      "address" : "Struhařovská 84, Brno",
      "value" : 3240000
    }
  ]
}

{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "email" : "vacula@gmail.com",
  "phone" : "739986862",
  "nemovitosti" : [
    {
      "type" : "flat",
      "address" : "Školní 31, Praha",
      "value" : 3480000
    },
    {
      "type" : "land",
      "address" : "Družstevní 133, Brno",
      "value" : 3320000
    }
  ]
}
```

Úkol 6 – Najděte všechny supernemovitosti (nemovitosti, které pod sebou mají další nemovitosti a ty mají další a tak dále hloubky 4 a více) např. dům->pozemek->garáž->sklep a zobrazte celou tuto nemovitost

Typická úloha pro grafy a Neo4j, jak se s tím popere Mongo? Mongo má kupodivu funkci graphLookup, která umí rekurzivně procházet „entity“ a přidávat všechny, které projde do pole. Stačí tedy vhodně napsat, jak má průchod vypadat. Procházíme kolekci estates a mapujeme vždy subestates.e\_id na \_id v estates, výsledek dáme do additional. Takto hezky rekurzivně projdeme vše (pozor, mongo má na tohle dle dokumentace limit cca 100 MB paměti).

```
> db.estates.aggregate([
...   {$graphLookup: {
...     from: "estates",
...     startWith: "$subestates.e_id",
...     connectFromField: "subestates.e_id",
...     connectToField: "_id",
...     as: "additional"
...   }},
...   {$match: {
...     "additional.2": {$exists: true}
...   }}
... ]).pretty()
{
  "_id" : "1",
  "type" : "house",
  "address" : "Javorová 12, Brno",
  "value" : 5700000,
  "subestates" : [
    {
      "e_id" : "21"
    }
  ],
  "additional" : [
    {
      "_id" : "22",
      "type" : "garage",
      "address" : "Javornická 15, Brno",
      "value" : 325000,
      "subestates" : [
        {
          "e_id" : "23"
        }
      ]
    },
    {
      "_id" : "23",
      "type" : "cellar",
      "address" : "Javornická 15, Brno",
      "value" : 487250,
      "subestates" : [ ]
    },
    {
      "_id" : "21",
      "type" : "land",
      "address" : "Javornická 15, Brno",
      "value" : 495000,
      "subestates" : [
        {
          "e_id" : "22"
        }
      ]
    }
  ]
}
```

## Úkol 7 – Vypište vztahy mezi pronajímateli a nájemníky

Musíme udělat 2x join, poprvé z People na Estates (person owns estate) a poté znovu z People na Estates (person rents owned estate). V prvním joinu tedy spojíme osoby s vlastněnými byty a v druhém pak všechny tyto data spojíme znovu s osobami přes vlastněnou nemovitost na pronájem nemovitosti. Pak už stačí vyfiltrovat vše co chceme vypsat, aby toho nebylo moc. Tímto dostaneme vztahy pronajímatelů s nájemníky, když chceme opak, stačí joiny vhodně otočit.

Vztahy pronajímatelů s nájemníky:

```
> db.people.aggregate([
...   {$match: {
...     types: "landlord"
...   }},
...   {$lookup: {
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "vlastni"
...   }},
...   {$project: {
...     name: 1,
...     surname: 1,
...     "vlastni._id": 1,
...     "vlastni.type": 1,
...     "vlastni.address": 1
...   }},
...   {$lookup: {
...     from: "people",
...     localField: "vlastni._id",
...     foreignField: "leases.e_id",
...     as: "najemnici"
...   }},
...   {$project: {
...     name: 1,
...     surname: 1,
...     "vlastni._id": 1,
...     "vlastni.type": 1,
...     "vlastni.address": 1,
...     "najemnici._id": 1,
...     "najemnici.name": 1,
...     "najemnici.surname": 1,
...     "najemnici.email": 1
...   }}
... ]).pretty()
{
  "_id" : "24",
  "name" : "Jiří",
  "surname" : "Bravenec",
  "vlastni" : [
    {
      "_id" : "1",
      "type" : "house",
      "address" : "Javorová 12, Brno"
    },
    {
      "_id" : "4",
      "type" : "flat",
      "address" : "Ondřejovská 86, Praha"
    }
  ],
  "najemnici" : [
    {
      "_id" : "28",
      "name" : "Otakar",
      "surname" : "Vacula",
      "email" : "vacula@gmail.com"
    },
    {
      "_id" : "32",
      "name" : "Rostislav",
      "surname" : "Šeda",
      "email" : "seda@gmail.com"
    }
  ]
}

{
  "_id" : "25",
  "name" : "Rudolf",
  "surname" : "Řeháček",
  "vlastni" : [
    {
      "_id" : "19",
      "type" : "land",
      "address" : "Lesní 92, Praha"
    },
    {
      "_id" : "2",
      "type" : "house",
      "address" : "Lipová 81, Praha"
    },
    {
      "_id" : "20",
      "type" : "land",
      "address" : "Ploní 15, Brno"
    }
  ],
  "najemnici" : [
    {
      "_id" : "28",
      "name" : "Otakar",
      "surname" : "Vacula",
      "email" : "vacula@gmail.com"
    },
    {
      "_id" : "30",
      "name" : "Cyril",
      "surname" : "Hončík",
      "email" : "honzik@gmail.com"
    },
    {
      "_id" : "31",
      "name" : "Viktor",
      "surname" : "Steklý",
      "email" : "stekly@seznam.cz"
    }
  ]
}

{
  "_id" : "26",
  "name" : "Libor",
  "surname" : "Šašek",
  "vlastni" : [
    {
      "_id" : "5",
      "type" : "flat",
      "address" : "Pionýrská 15, Brno"
    },
    {
      "_id" : "6",
      "type" : "flat",
      "address" : "Podhornická 139, Praha"
    }
  ],
  "najemnici" : [
    {
      "_id" : "33",
      "name" : "Vilém",
      "surname" : "Balek",
      "email" : "balek@seznam.cz"
    }
  ]
}
```

```

{
  "_id" : "27",
  "name" : "Vilém",
  "surname" : "Kotík",
  "vlastni" : [
    {
      "_id" : "8",
      "type" : "flat",
      "address" : "Spojovací 42, Praha"
    },
    {
      "_id" : "9",
      "type" : "flat",
      "address" : "Struhařovská 84, Brno"
    }
  ],
  "najemnici" : [
    {
      "_id" : "34",
      "name" : "Vratislav",
      "surname" : "Krejša",
      "email" : "krejsa@gmail.com"
    },
    {
      "_id" : "35",
      "name" : "Marcel",
      "surname" : "Uhlík",
      "email" : "uhlik@seznam.cz"
    }
  ]
},
{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "vlastni" : [
    {
      "_id" : "10",
      "type" : "flat",
      "address" : "Školní 31, Praha"
    },
    {
      "_id" : "18",
      "type" : "land",
      "address" : "Družstevní 133, Brno"
    }
  ],
  "najemnici" : [
    {
      "_id" : "29",
      "name" : "Oldřich",
      "surname" : "Šmerda",
      "email" : "smerda@seznam.cz"
    },
    {
      "_id" : "36",
      "name" : "Ivo",
      "surname" : "Chlumský",
      "email" : "chlumsky@hotmail.com"
    }
  ]
}

```

```

{
  "_id" : "29",
  "name" : "Oldřich",
  "surname" : "Šmerda",
  "vlastni" : [
    {
      "_id" : "11",
      "type" : "flat",
      "address" : "Třebelatská 131, Praha"
    }
  ],
  "najemnici" : [
    {
      "_id" : "37",
      "name" : "Kristián",
      "surname" : "Žabka",
      "email" : "zabka@seznam.cz"
    }
  ]
},
{
  "_id" : "30",
  "name" : "Cyril",
  "surname" : "Honzíček",
  "vlastni" : [
    {
      "_id" : "17",
      "type" : "land",
      "address" : "Vládní 15, Praha"
    }
  ],
  "najemnici" : [
    {
      "_id" : "38",
      "name" : "Viktor",
      "surname" : "Slaný",
      "email" : "slany@seznam.cz"
    }
  ]
}

```

## Vztahy nájemníků s pronajímateli:

```
> db.people.aggregate([
...   {$match: {
...     types: "tenant"
...   }},
...   {$lookup: {
...     from: "estates",
...     localField: "leases.e_id",
...     foreignField: "_id",
...     as: "pronajal_si"
...   }},
...   {$project: {
...     name: 1,
...     surname: 1,
...     "pronajal_si.id": 1,
...     "pronajal_si.type": 1,
...     "pronajal_si.address": 1
...   }},
...   {$lookup: {
...     from: "people",
...     localField: "pronajal_si.id",
...     foreignField: "owns.e_id",
...     as: "poskytovatel"
...   }},
...   {$project: {
...     name: 1,
...     surname: 1,
...     "pronajal_si.id": 1,
...     "pronajal_si.type": 1,
...     "pronajal_si.address": 1,
...     "poskytovatel.id": 1,
...     "poskytovatel.name": 1,
...     "poskytovatel.surname": 1,
...     "poskytovatel.email": 1
...   }}
... ]).pretty()
{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "pronajal_si" : [
    {
      "_id" : "1",
      "type" : "house",
      "address" : "Javorová 12, Brno"
    },
    {
      "_id" : "20",
      "type" : "land",
      "address" : "Ploní 15, Brno"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "24",
      "name" : "Jiří",
      "surname" : "Bravenec",
      "email" : "bravenec@seznam.cz"
    },
    {
      "_id" : "25",
      "name" : "Rudolf",
      "surname" : "Řeháček",
      "email" : "rehacek@gmail.com"
    }
  ]
}
```

```
{
  "_id" : "29",
  "name" : "Oldřich",
  "surname" : "Šmerda",
  "pronajal_si" : [
    {
      "_id" : "18",
      "type" : "land",
      "address" : "Družstevní 133, Brno"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "28",
      "name" : "Otakar",
      "surname" : "Vacula",
      "email" : "vacula@gmail.com"
    }
  ]
},
{
  "_id" : "30",
  "name" : "Cyril",
  "surname" : "Honzík",
  "pronajal_si" : [
    {
      "_id" : "19",
      "type" : "land",
      "address" : "Lesní 92, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "25",
      "name" : "Rudolf",
      "surname" : "Řeháček",
      "email" : "rehacek@gmail.com"
    }
  ]
},
{
  "_id" : "31",
  "name" : "Viktor",
  "surname" : "Steklý",
  "pronajal_si" : [
    {
      "_id" : "2",
      "type" : "house",
      "address" : "Lipová 81, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "25",
      "name" : "Rudolf",
      "surname" : "Řeháček",
      "email" : "rehacek@gmail.com"
    }
  ]
},
{
  "_id" : "32",
  "name" : "Rostislav",
  "surname" : "Šeda",
  "pronajal_si" : [
    {
      "_id" : "4",
      "type" : "flat",
      "address" : "Ondřejovská 86, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "24",
      "name" : "Jiří",
      "surname" : "Bravenec",
      "email" : "bravenec@seznam.cz"
    }
  ]
}
```

```

{
  "_id" : "33",
  "name" : "Vilém",
  "surname" : "Balek",
  "pronajal_si" : [
    {
      "_id" : "6",
      "type" : "flat",
      "address" : "Podhornická 139, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "26",
      "name" : "Libor",
      "surname" : "Šásek",
      "email" : "sasek@seznam.cz"
    }
  ]
}
{
  "_id" : "34",
  "name" : "Vratislav",
  "surname" : "Krejša",
  "pronajal_si" : [
    {
      "_id" : "8",
      "type" : "flat",
      "address" : "Spojovací 42, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "27",
      "name" : "Vilém",
      "surname" : "Kotík",
      "email" : "kotik@seznam.cz"
    }
  ]
}
{
  "_id" : "35",
  "name" : "Marcel",
  "surname" : "Uhlík",
  "pronajal_si" : [
    {
      "_id" : "9",
      "type" : "flat",
      "address" : "Struhařovská 84, Brno"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "27",
      "name" : "Vilém",
      "surname" : "Kotík",
      "email" : "kotik@seznam.cz"
    }
  ]
}
{
  "_id" : "36",
  "name" : "Ivo",
  "surname" : "Chlumský",
  "pronajal_si" : [
    {
      "_id" : "10",
      "type" : "flat",
      "address" : "Školní 31, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "28",
      "name" : "Otakar",
      "surname" : "Vacula",
      "email" : "vacula@gmail.com"
    }
  ]
}

```

```

{
  "_id" : "37",
  "name" : "Kristián",
  "surname" : "Žabka",
  "pronajal_si" : [
    {
      "_id" : "11",
      "type" : "flat",
      "address" : "Třebelatská 131, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "29",
      "name" : "Oldřich",
      "surname" : "Šmerda",
      "email" : "smerda@seznam.cz"
    }
  ]
}
{
  "_id" : "38",
  "name" : "Viktor",
  "surname" : "Slaný",
  "pronajal_si" : [
    {
      "_id" : "17",
      "type" : "land",
      "address" : "Vládní 15, Praha"
    }
  ],
  "poskytovatel" : [
    {
      "_id" : "30",
      "name" : "Cyril",
      "surname" : "Honzík",
      "email" : "honzik@gmail.com"
    }
  ]
}

```

Úkol 8 – Vypište takové lidi, kteří zároveň vlastní nemovitost a jsou nájemníky v jiné. Jednoduchá verze, stačí se podívat že Person je jak landlord tak tenant. Kdybychom tuto informaci neměli, museli by jsme opět joinovat.

```
> db.people.find(
...   {$and: [
...     { types: "landlord" },
...     { types: "tenant" }
...   ]},
...   {
...     name: 1,
...     surname: 1,
...     email: 1
...   }
... ).pretty();
{
  "_id" : "28",
  "name" : "Otakar",
  "surname" : "Vacula",
  "email" : "vacula@gmail.com"
}
{
  "_id" : "29",
  "name" : "Oldřich",
  "surname" : "Šmerda",
  "email" : "smerda@seznam.cz"
}
{
  "_id" : "30",
  "name" : "Cyril",
  "surname" : "Honzík",
  "email" : "honzik@gmail.com"
}
```



Úkol 9 – Vypište lidi, kteří profitují na tom, že jsou nájemníky někde jinde a vypište jejich profit, daně zanedbejte

Chceme tedy Person, který je landlord i tenant. Pak si k němu joinem přidáme vlastněné byty a nájemníky těchto bytů. Unwindem zajistíme, že dostaneme něco jako kartézský součin, z kterého pak vyfiltrujeme pouze ty hodnoty, kde nájemník opravdu platí za vlastněný byt (protože nájemník může mít pronajatých více bytů od více osob, vyfiltrujeme to pouze na hodnoty, kde se id vlastněného bytu rovná id pronajatého bytu). Tyto zbylé hodnoty pak opět sjednotíme podle id vlastníka a sečteme měsíční příjmy a ztráty. Na konec uděláme rozdíl a vyfiltrujeme jen ty, které mají profit. Pokud bychom chtěli jméno a příjmení, kromě id, museli bychom ještě joinout s Person (pravděpodobně to jde i jinak, ale nepodařilo se mi přes \$group propasovat další data).

```
> db.people.aggregate([
...   {$match: {
...     $and: [
...       {types: "landlord"},
...       {types: "tenant"}
...     ]
...   }},
...   {$lookup:{
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "vlastni"
...   }},
...   {$lookup:{
...     from: "people",
...     localField: "vlastni._id",
...     foreignField: "leases.e_id",
...     as: "najemnici"
...   }},
...   {$unwind: '$owns'},
...   {$unwind: '$leases'},
...   {$unwind: '$najemnici'},
...   {$unwind: '$najemnici.leases'},
...   {$match: {
...     $expr: {$eq: ["$najemnici.leases.e_id", "$owns.e_id"]}
...   }},
...   {$group: {
...     _id: "$_id",
...     pos: {$sum: "$najemnici.leases.monthly"},
...     neg: {$sum: "$leases.monthly"}
...   }},
...   {$project: {
...     _id: 1,
...     profit: {$subtract: [ "$pos", "$neg" ] }
...   }},
...   {$match: {
...     profit: {$gt: 0}
...   }},
... ]).pretty()
{ "_id" : "29", "profit" : 19830 }
```

## Úkol 10 – Zobrazte všechny půjčky

Jelikož půjčky jsou řešeny přes many-to-many vazbu s Bank a Person. Musíme si k půjčce najoinovat obě data a pak vyfiltrovat, co chceme vypsát.

```
> db.loans.aggregate([
...   {$lookup:{
...     from: "banks",
...     localField: "b_id",
...     foreignField: "_id",
...     as: "banka"
...   }},
...   {$lookup:{
...     from: "people",
...     localField: "p_id",
...     foreignField: "_id",
...     as: "zakaznik"
...   }},
...   {$project: {
...     _id: 1,
...     type: 1,
...     amount: 1,
...     interest: 1,
...     banka: 1,
...     "zakaznik._id": 1,
...     "zakaznik.name": 1,
...     "zakaznik.surname": 1,
...     "zakaznik.email": 1
...   }},
... ]).pretty()
{
  "_id" : "39",
  "type" : "Nemovitostní fond",
  "amount" : 5000000,
  "interest" : 0.04,
  "banka" : [
    {
      "_id" : "CEKOCZPP",
      "name" : "Československá obchodní banka",
      "code" : 300
    }
  ],
  "zakaznik" : [
    {
      "_id" : "24",
      "name" : "Jiří",
      "surname" : "Bravenec",
      "email" : "bravenec@seznam.cz"
    }
  ]
}
{
  "_id" : "40",
  "type" : "Nemovitostní fond",
  "amount" : 4000000,
  "interest" : 0.035,
  "banka" : [
    {
      "_id" : "CEKOCZPP",
      "name" : "Československá obchodní banka",
      "code" : 300
    }
  ],
  "zakaznik" : [
    {
      "_id" : "25",
      "name" : "Rudolf",
      "surname" : "Řeháček",
      "email" : "rehacek@gmail.com"
    }
  ]
}
```

```
{
  "_id" : "41",
  "type" : "Hypotéka",
  "amount" : 8000000,
  "interest" : 0.02,
  "banka" : [
    {
      "_id" : "FIOBCZPP",
      "name" : "Fio banka",
      "code" : 2010
    }
  ],
  "zakaznik" : [
    {
      "_id" : "25",
      "name" : "Rudolf",
      "surname" : "Řeháček",
      "email" : "rehacek@gmail.com"
    }
  ]
}
{
  "_id" : "42",
  "type" : "Hypotéka",
  "amount" : 2700000,
  "interest" : 0.025,
  "banka" : [
    {
      "_id" : "CTASCZ22",
      "name" : "Banka CREDITAS",
      "code" : 2250
    }
  ],
  "zakaznik" : [
    {
      "_id" : "26",
      "name" : "Libor",
      "surname" : "Šašek",
      "email" : "sasek@seznam.cz"
    }
  ]
}
```

```
{
  "_id" : "43",
  "type" : "Hypotéka",
  "amount" : 3000000,
  "interest" : 0.03,
  "banka" : [
    {
      "_id" : "CTASCZ22",
      "name" : "Banka CREDITAS",
      "code" : 2250
    }
  ],
  "zakaznik" : [
    {
      "_id" : "28",
      "name" : "Otakar",
      "surname" : "Vacula",
      "email" : "vacula@gmail.com"
    }
  ]
}
{
  "_id" : "44",
  "type" : "Jednorázová půjčka",
  "amount" : 200000,
  "interest" : 0.05,
  "banka" : [
    {
      "_id" : "FIOBCZPP",
      "name" : "Fio banka",
      "code" : 2010
    }
  ],
  "zakaznik" : [
    {
      "_id" : "33",
      "name" : "Vilém",
      "surname" : "Balek",
      "email" : "balek@seznam.cz"
    }
  ]
}
```

```
{
  "_id" : "45",
  "type" : "Jednorázová půjčka",
  "amount" : 250000,
  "interest" : 0.05,
  "banka" : [
    {
      "_id" : "CTASCZ22",
      "name" : "Banka CREDITAS",
      "code" : 2250
    }
  ],
  "zakaznik" : [
    {
      "_id" : "34",
      "name" : "Vratislav",
      "surname" : "Krejsa",
      "email" : "krejsa@gmail.com"
    }
  ]
}
{
  "_id" : "46",
  "type" : "Jednorázová půjčka",
  "amount" : 300000,
  "interest" : 0.06,
  "banka" : [
    {
      "_id" : "CTASCZ22",
      "name" : "Banka CREDITAS",
      "code" : 2250
    }
  ],
  "zakaznik" : [
    {
      "_id" : "35",
      "name" : "Marcel",
      "surname" : "Uhlík",
      "email" : "uhlik@seznam.cz"
    }
  ]
}
```

Úkol 11 – Zjistěte průměrný úrok u všech půjček a u nemovitostních fondů  
Stačí projet všechny půjčky (group by null) a zavolat průměrovací funkci na úrok.

```
> db.loans.aggregate([
...   {$group:{
...     _id: null,
...     prumerny_urok: { $avg: "$interest" }
...   }},
...   {$project: {
...     _id: 0,
...     prumerny_urok: 1
...   }}
... ])
{ "prumerny_urok" : 0.03875 }
```

Úkol 12 – Zjistěte, jestli pronajímatelé profitují na svých půjčkách, výdaje zanedbejte, daň z příjmu předpokládejte 15% bez odpisů

Musíme si úlohu rozdělit na dvě části, protože kartézský součin přes tolik věcí by produkoval duplicitní data (museli bychom vymyslet chytřejší group by než přes id). Rozdělíme si úlohu na profits a losses. Profits už máme z předchozí úlohy, jenom se trochu upraví a přidá se násobení 0.85 kvůli dani. Ztráty jsou jednodušší, stačí vzít všechny vlastníky nějaké nemovitosti, najoinovat si k nim jejich půjčky a unwindnout je. Pak stačí obyčejný group by podle id (akorát násobíme úrokem a dělíme 12 měsíci). Na konec by asi šlo vymyslet něco chytřejšího, ale pro ukázkou co vše v Mongu jde jsem to umlátil JavaScriptem.

```
> var profits = db.people.aggregate([
...   {$match: {
...     types: "landlord"
...   }},
...   {$lookup: {
...     from: "estates",
...     localField: "owns.e_id",
...     foreignField: "_id",
...     as: "vlastni"
...   }},
...   {$lookup: {
...     from: "people",
...     localField: "vlastni_id",
...     foreignField: "leases.e_id",
...     as: "najemnici"
...   }},
...   {$unwind: '$owns'},
...   {$unwind: '$najemnici'},
...   {$unwind: '$najemnici.leases'},
...   {$match: {
...     $expr: {$eq: ["$najemnici.leases.e_id", "$owns.e_id"]}
...   }},
...   {$group: {
...     _id: "$_id",
...     pos: {$sum: {$multiply: ["$najemnici.leases.monthly", 0.85]}}
...   }}
... ]).toArray();
>
> var losses = db.people.aggregate([
...   {$match: {
...     types: "landlord"
...   }},
...   {$lookup: {
...     from: "loans",
...     localField: "_id",
...     foreignField: "p_id",
...     as: "pujcky"
...   }},
...   {$unwind: '$pujcky'},
...   {$group: {
...     _id: "$_id",
...     neg: {$sum: {$divide: [{$multiply: ["$pujcky.amount", "$pujcky.interest"]}, 12]}}
...   }}
... ]).toArray();
>
> var result = []
> for (i = 0; i < losses.length; ++i) {
...   for (j = 0; j < profits.length; ++j) {
...     if (losses[i]["_id"] == profits[j]["_id"]) {
...       result.push({_id: losses[i]["_id"], neg: losses[i]["neg"], pos: profits[j]["pos"], total: (profits[j]["pos"] - losses[i]["neg"])});
...     }
...   }
... }
4
> result
[
  {
    "_id" : "24",
    "neg" : 16666.666666666668,
    "pos" : 51807.5,
    "total" : 35140.83333333333
  },
  {
    "_id" : "26",
    "neg" : 5625,
    "pos" : 19125,
    "total" : 13500
  },
  {
    "_id" : "28",
    "neg" : 7500,
    "pos" : 35190,
    "total" : 27690
  },
  {
    "_id" : "25",
    "neg" : 25000,
    "pos" : 64132.5,
    "total" : 39132.5
  }
]
```

## Závěr

Vytvořil jsem databázi pro Neo4j tak, aby šla rozumně převést do MongoDB. Všechny dotazy (kromě „ukážte jako graf“) jdou udělat jak v Neo4j, tak v MongoDB. Tím že byly dotazy vymyšlené pro Neo4j a tím, že MongoDB moc není dělané na časté joiny a obecně má jiný přístup, tak bylo psaní dotazů pro Mongo o dost náročnější. Práce ukazuje, že je velmi důležité zvolit pro daný problém vhodnou databázi. Obecně se mi pracovalo lépe s Neo4j, ne kvůli typu dotazů, ale kvůli syntaxi. JavaScript syntaxe má strašně symbolů a závorek a byla pro mě méně přehlednější. Na druhou stranu mi připadlo, že MongoDB dotazy šlo psát celkem mechanicky, kdežto u Neo4j jsem nad tím musel přemýšlet.