**MongoDB**

**1.**

db.collection.find({},

{

"user\_id": 1,

"firstName": 1,

"lastname": 1,

"\_id": 0

})

**2.**

db.collection.find({

"firstName": "Grace"

},

{

"email": 1,

"password": 1,

"\_id": 0

})

**4.**

**11.**

db.receptek.find({"title": {$regex: /Tacos/}}, {}).pretty()

**12.**

db.receptek.aggregate ([{$group:

... {\_id: "$type", total: {$sum: "$cook\_time"}}}])

**13.**

db.receptek.find({$and:  
... [{servings: 4}, {tags: {$in: ["quick", "easy"]}}]}).count()

**14.**

db.receptek.updateOne(

... {"\_id": ObjectId("5e878f5220a4f574c0aa56db")},

... {$set: {"cook\_time": 33}})

**15.**

db.receptek.updateOne(

... {\_id: ObjectId("5e5e9c470d33e9e8e3891b35")},

... {$push: {likes: 200}})

**Neo4J**

**17.**

match (p:Person)

where p.born = 1964 or p.born = 1965

return p.name, p.born

**18.**

match (m:Movie)

where m.title starts with 'A'

return m.title, m.released

order by m.released desc

**19.**

match (p:Person)-[:PRODUCED]->(m:Movie)

where p.name = 'Joel Silver'

return m.title

**20.**

match (p:Person)-[:DIRECTED]->(m:Movie)

with p.name as `nev`, count(\*) as `db`

where db > 1

return nev, db

**21.**

match (p:Person)-[:ACTED\_IN]->(m:Movie)<-[:DIRECTED]-(p:Person)

return p

**22.**

match (p:Person)-[:ACTED\_IN]->(m:Movie)

return m.title, count(\*)

order by count(\*) desc, m.title

limit 10

**23.**

create index ipersonborn

for (p:Person)

on (p.name, p.born)

:schema

**24.**

profile

match (p:Person)

where p.born >= 1980 and p.born <= 2000

return p.name, p.born

**28.**

match (tar:tanar), (tulo:tanulo)

where tar.nev = 'Toth Otto' and tulo.nev = 'Kiss Bela'

create (tar)-[:tanit]->(tulo)

**29.**

match (t:tanulo {nev: 'Nagy Ilona'})

set t.atlag = 5.0

return (t)