Francisco Carvalho

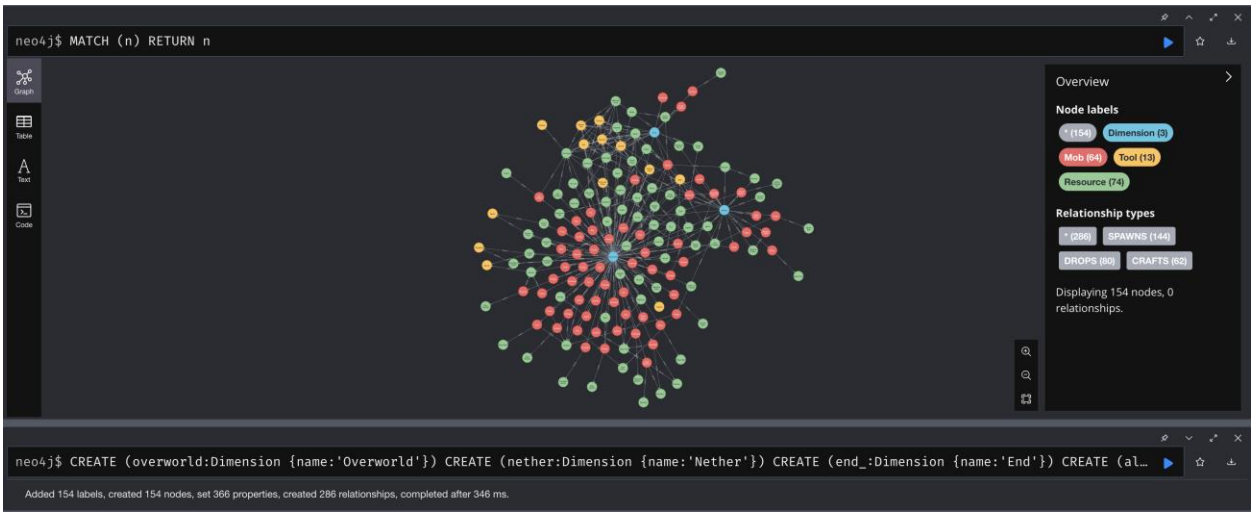
Dr. Villa

CSC 434

Project 2: Minecraft Database

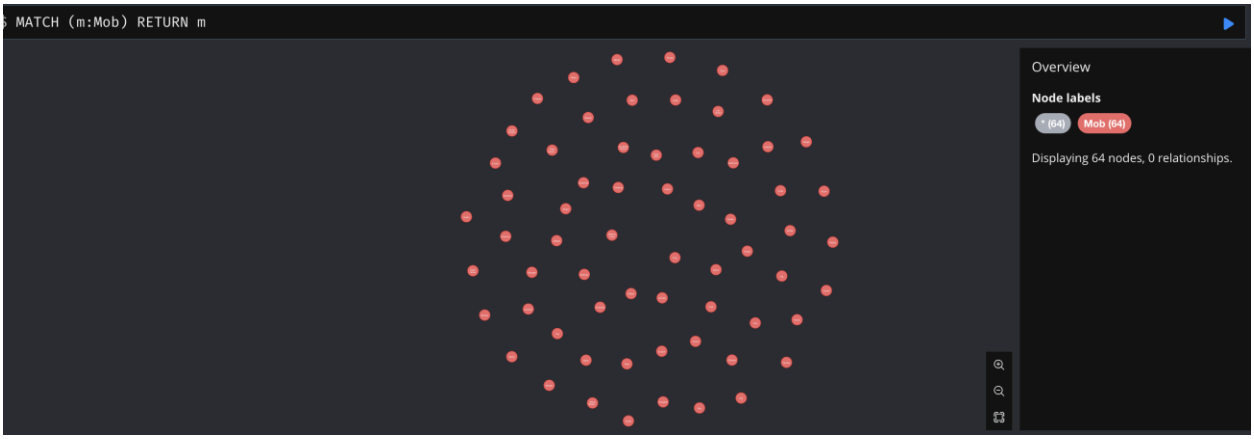
For this project I will be doing a Minecraft database. There will be four different types of nodes. These will be blocks and/or resources, tools, “mobs”, and dimension. Blocks and resources are what you use in the game for building and crafting. You can either harvest these materials or have them be dropped but different creatures. Tools in the game can include things like a sword, pickaxe, or shovel, that you can use to fight or build with. Mobs is a term used to describe the creatures in the game. These can be hostile mobs, like zombies or spiders, or passive creatures, like cows or chickens. Finally, there are three different dimensions or “realms” in the base game. These are the Overworld, where the player spawns, and then the Nether and the End. These dimensions are accessed through certain portals the player can either build or find and each dimension has a different set of blocks, tools, and creatures that don’t spawn in the other dimensions. For the relationships, I will have a “crafts” relationship, “spawn” relationship, and a “drops” relationship. The “crafts” relationship connects a resource to another block or tools that can be used to craft that specific block or tool (ex. Stick ---crafts--> Sword). Note that more than one specific block or resource can be used to craft a certain block or tool. The “drop” relationship is supposed to show what resource is dropped by the mob when either killed or acted upon, by some action (ex. Spider –drops--> Spider Eye). Finally, the “spawn” relationship shows where a specific mob, block, resource, or tool (sometimes found in chests), can be found in. For simplicity reasons, I decided to group certain things together that have variants. For instance, there are 12 different wood variants, each with a plank version, a stripped version, and a block version. This includes other things like mobs, tools, and other blocks. So, I decided to group these together, so it isn’t so much writing. As I have just mentioned, certain tools have different tiers (swords have a wood, stone, gold, iron, diamond, and netherite variants). So, I will have relationships from those resources to each respective tools that this situation applies to. This is the only instance that I believed was necessary for having multiple relationships leading to one item.

Match (n) Return n Output:

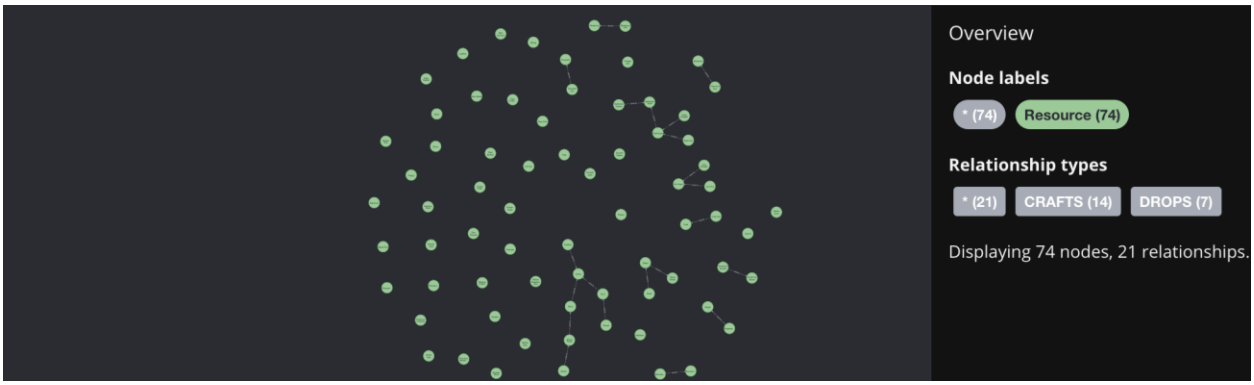


1. Find all the tools.

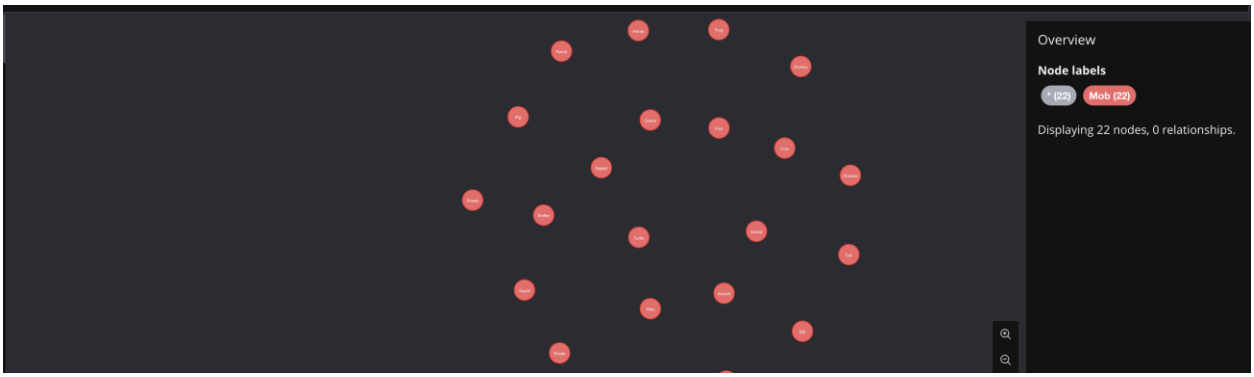
2. Find all the mobs.



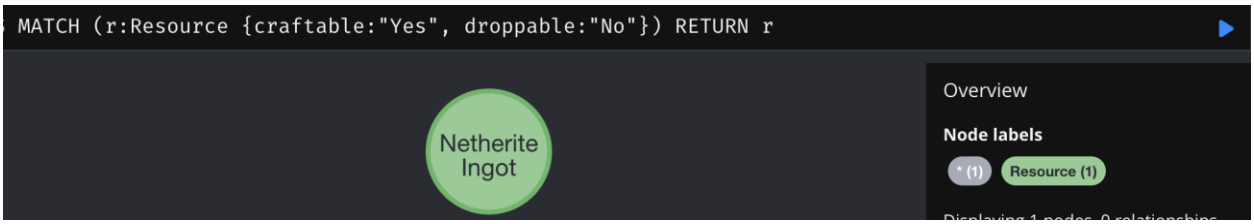
3. Find all the resources.



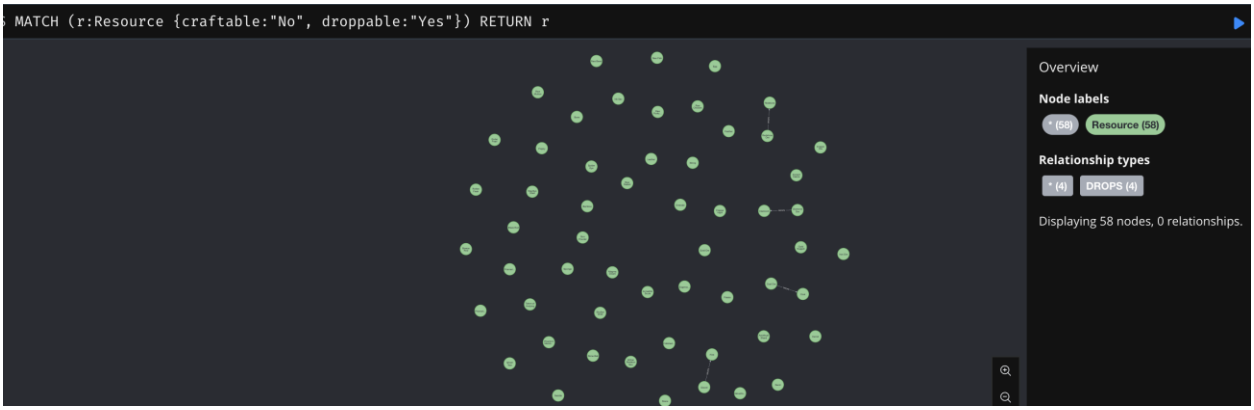
4. Find all the passive mobs.



5. Find all the resources that you can craft but can't be dropped.



6. Find all the resources that you can't craft but can be dropped.



7. Find all the mobs that can be found in the nether.

```
j$ MATCH (m:Mob) ←[:SPAWNS]- (n:Dimension {name:"Nether"}) RETURN m.name,n.name
```

	m.name	n.name
1	"Strider"	"Nether"
2	"Enderman"	"Nether"
3	"Piglin"	"Nether"
4	"Zombified Pigman"	"Nether"
5	"Blaze"	"Nether"
6	"Ghast"	"Nether"

8. Find all the resources that spawn in the overworld.

```
MATCH (r:Resource) ←[:SPAWNS]- (n:Dimension {name:"Overworld"}) RETURN r.name,n.name
```

r.name	n.name
"Wood"	"Overworld"
"Wood Plank"	"Overworld"
"Stick"	"Overworld"
"Stone"	"Overworld"
"Cobblestone"	"Overworld"
"Gravel"	"Overworld"

9. Find all the hostile mobs that spawn in the overworld.

```
MATCH (m:Mob {type:"Hostile"}) <-[:SPAWNS]- (n:Dimension {name:"Overworld"}) RETURN m.name,n.n…
```

m.name	n.name
"Breeze"	"Overworld"
"Creaking"	"Overworld"
"Creeper"	"Overworld"
"Elder Guardian"	"Overworld"
"Evoker"	"Overworld"
"Guardian"	"Overworld"

10. Find all the neutral mobs and where they spawn.

```
MATCH (m:Mob {type:"Neutral"}) <-[:SPAWNS]- (n:Dimension)
RETURN m.name,m.type, n.name
```

	m.name	m.type	n.name
1	"Bee"	"Neutral"	"Overworld"
2	"Dolphin"	"Neutral"	"Overworld"
3	"Drowned"	"Neutral"	"Overworld"
4	"Enderman"	"Neutral"	"Overworld"
5	"Enderman"	"Neutral"	"Nether"
6	"Enderman"	"Neutral"	"End"

11. Find items that are craft-able, and what items are used to craft them.

```
MATCH (r1:Resource {craftable:"Yes"}) -[:CRAFTS]→ (r2:Resource) RETURN r2.name as Item, r1.na…
```

Item	Crafted_With
"Stick"	"Wood Plank"
"Glass Bottle"	"Glass"
"Netherite Ingot"	"Gold Ingot"
"Arrow"	"Stick"

12. What items are dropped by the drowned mob?

```
MATCH (d:Mob {name:"Drowned"}) -[:DROPS]→ (r:Resource) RETURN d.name, r.name
```

d.name	r.name
"Drowned"	"Copper Ingot"
"Drowned"	"Rotten Flesh"
"Drowned"	"Nautilus Shell"

13. What resources are dropped when you break a block?

```
$ MATCH (r1:Resource) -[:DROPS]→ (r2:Resource) RETURN r1.name, r2.name
```

r1.name	r2.name
"Gravel"	"Flint"
"Glowstone"	"Glowstone Dust"
"Amethyst Cluster"	"Amethyst Shard"
"Coal Ore"	"Coal"
"Diamond Ore"	"Diamond"
"Redstone Ore"	"Redstone"

14. What resources do you need to craft an arrow?

```
$ MATCH (r1:Resource) -[:CRAFTS]→ (r2:Resource {name:"Arrow"}) RETURN r1.name, r2.name
```

r1.name	r2.name
"Stick"	"Arrow"
"Flint"	"Arrow"
"Feather"	"Arrow"

15. How many hostile mobs are there in each dimension?

```
MATCH (d:Dimension) -[:SPAWNS]→ (m:Mob {type:"Hostile"}) RETURN d.name, count(m)
```

d.name	count(m)
"Nether"	7
"Overworld"	18
"End"	3

16. What mobs drop leather?

```
MATCH (m:Mob) -[:DROPS]→ (r:Resource {name:"Leather"}) RETURN m.name
```

m.name
"Cow"
"Donkey"
"Horse"
"Llama"
"Hoglin"

17. How many total passive mobs are there?

```
MATCH (m:Mob {type:"Passive"}) RETURN m.type, count(m)
```

m.type	count(m)
"Passive"	22

18. What tools need string to be crafted?

```
MATCH (s:Resource {name:"String"}) -[:CRAFTS]→ (t:Tool) RETURN t.name
```

t.name
"Bow"
"Fishing Rod"

19. What craft-able items are crafted using non-craft-able items?

```
MATCH (r1:Resource {craftable:"No"}) -[:CRAFTS]→ (r2:Resource {craftable:"Yes"}) RETURN r2.na…
```

Item	Crafted_With
"Wood Plank"	"Wood"
"Glass"	"Sand"
"Sugar"	"Sugarcane"
"Iron Ingot"	"Iron Ore"
"Iron Ingot"	"Iron Nugget"
"Gold Ingot"	"Gold Ore"

20. What mobs drop raw meat?

```
MATCH (m:Mob) -[:DROPS]→ (r:Resource) WHERE r.name CONTAINS 'Raw' RETURN m.name, r.name
```

m.name	r.name
"Chicken"	"Raw Chicken"
"Fish"	"Raw Fish"
"Dolphin"	"Raw Fish"
"Polar Bear"	"Raw Fish"
"Elder Guardian"	"Raw Fish"
"Guardian"	"Raw Fish"