DAY-23

NODE4J.TXT

Nodes

======

CREATE (). --- Creates Node without Label

CREATE (:Student) -- Creates Node with Label

CREATE (:Student {name : 'Reddy'})

MATCH (a) return a -- Retrieves all Nodes

CREATE (:Student :Person)

CREATE (st:Student:Person {name : 'John', birth_year: 1990}) return st. --- This creates a node with both Labels and properties

CREATE (st:Student {name : 'Doe', birth_year: 1990})

CREATE (st:Student {name : 'SAM', birth_year: 1985})

match (p:Person) return p -- Retrieves Label as Person

match (s:Student) return s -- Retrieves Label as Student

match (s:Student {birth_year:1990}) return s. --- Birth Year = 1990 for Student

match (s:Student {name: 'John', birth_year:1990}) return s. -- Searching for Name on Birth Year

match (s:Student) where s.birth_year >1985 return (s). - Display birth_year > 1985

match (s:Student) where s.name ='John' and s.birth_year >1985 return (s)

MATCH (s:Student) where ID(s) = 2 RETURN (s)

MATCH (s:Student) where ID(s) = 2 set s.birth_year = 1998 RETURN (s) --- Update Record Based on ID

MATCH (s:Student) where ID(s) = 2 remove s:Person RETURN s-- Remvove Person Label

MATCH (s:Student) where ID(s) = 2 remove s.birth_year RETURN s ---- Removes the Property

MATCH (n)

DETACH DELETE n

Relationships

=====

(Creating BOth Nodes at same time and creating relationship)

==========

create (st:Student {name:'Vijay'}), (sub:Subject {sub_name: 'Gen AI'})

create (st)-[:IS_LEARNING]-> (sub)

return st,stub

For existin Nodes

=====

MATCH (st:Student {name: 'Doe'}), (sub:Subject {sub_name: 'Gen AI'}) CREATE (st)-[:IS_LEARNING]->(sub)

```
MATCH (st:Student {name: 'Reddy'}), (sub:Subject {sub_name: 'Gen AI'})
CREATE (st)-[:IS_LEARNING]->(sub)
return st, sub

MATCH (st:Student) , (sub:Subject)
where ID(st) = 4 and ID(sub) = 6
CREATE (st)-[:IS_LEARNING]->(sub)
return st, sub
```

Constraints

=====

```
CREATE CONSTRAINT t_name IF NOT EXISTS
FOR (t:Training)
REQUIRE t.name IS UNIQUE

CREATE (:Training {name : 'Gen AI'})
```

Property constraint

==========

```
CREATE CONSTRAINT t_name_exists
FOR (t:Training)
REQUIRE t.name IS NOT NULL
```

Aggregation

======

Count

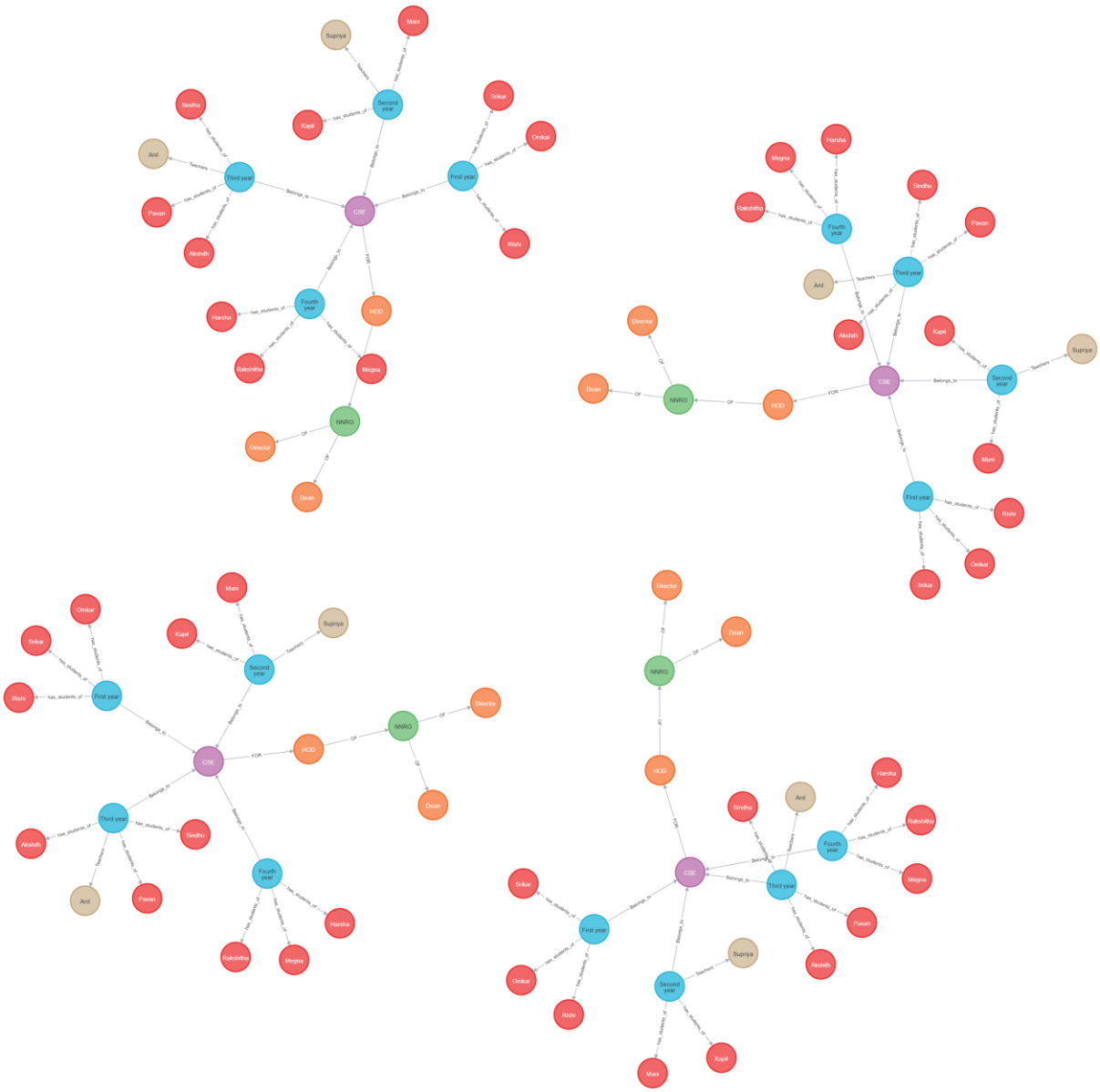
MATCH (s:Student) RETURN count(s.name)

Distinct

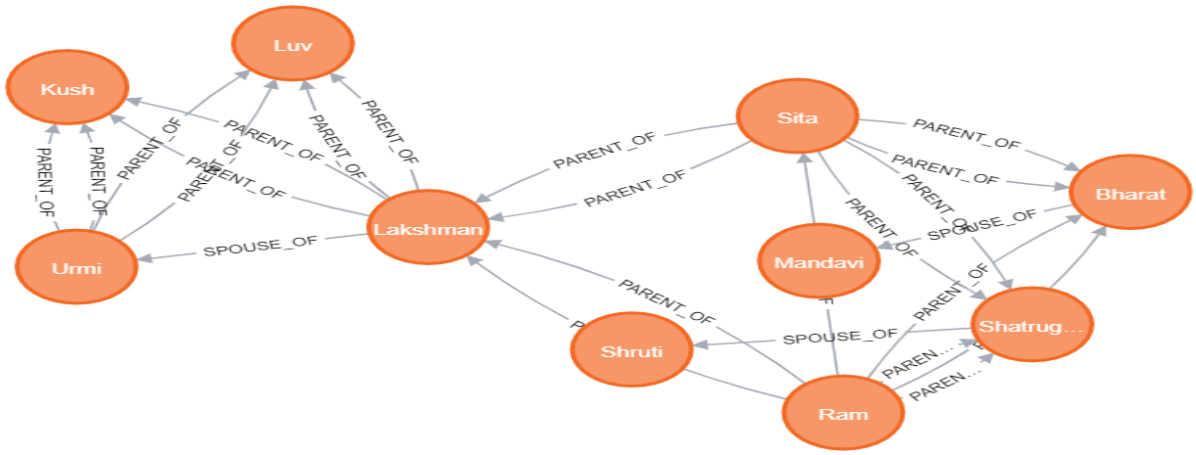
====

match (s:Student) with distinct s.name as Name return Name

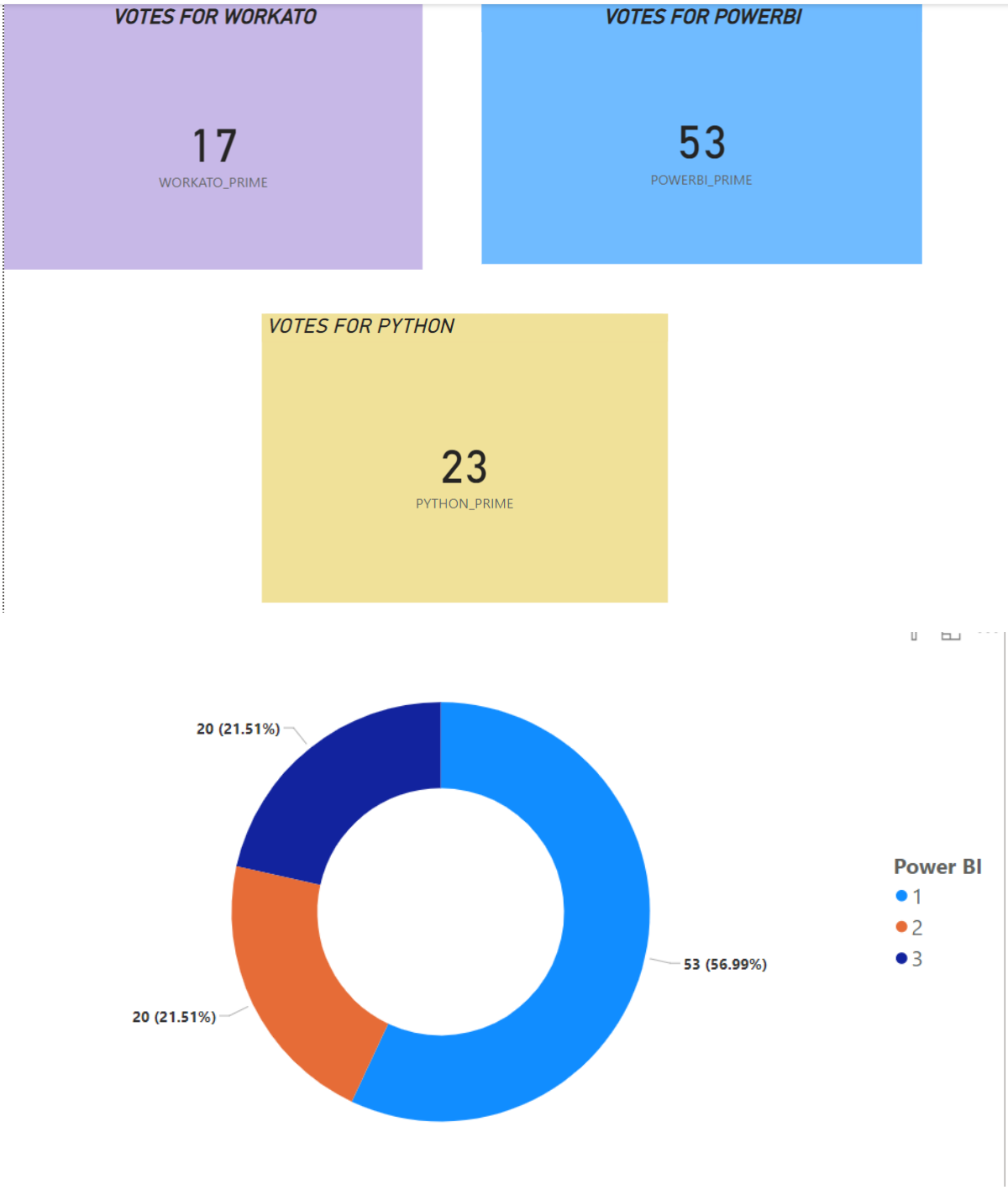
NNRG DETAILS

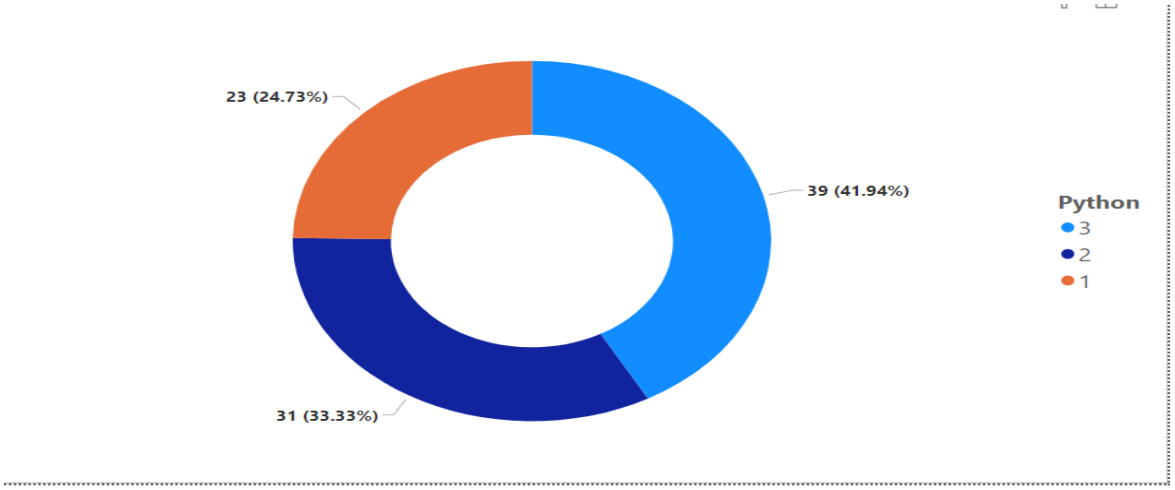


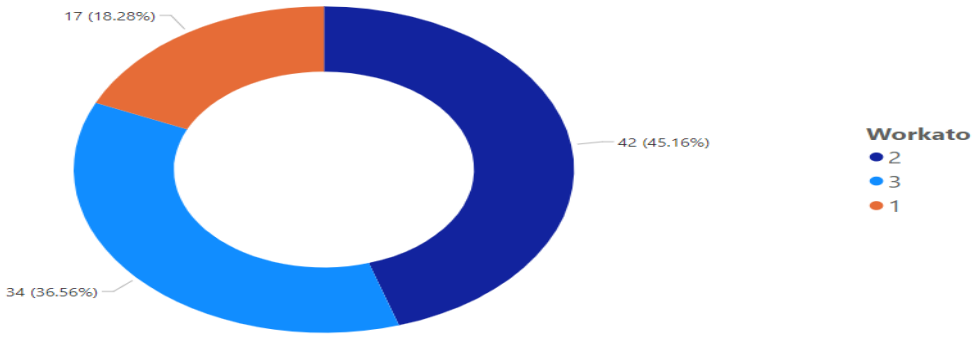
FAMILY DETAILS

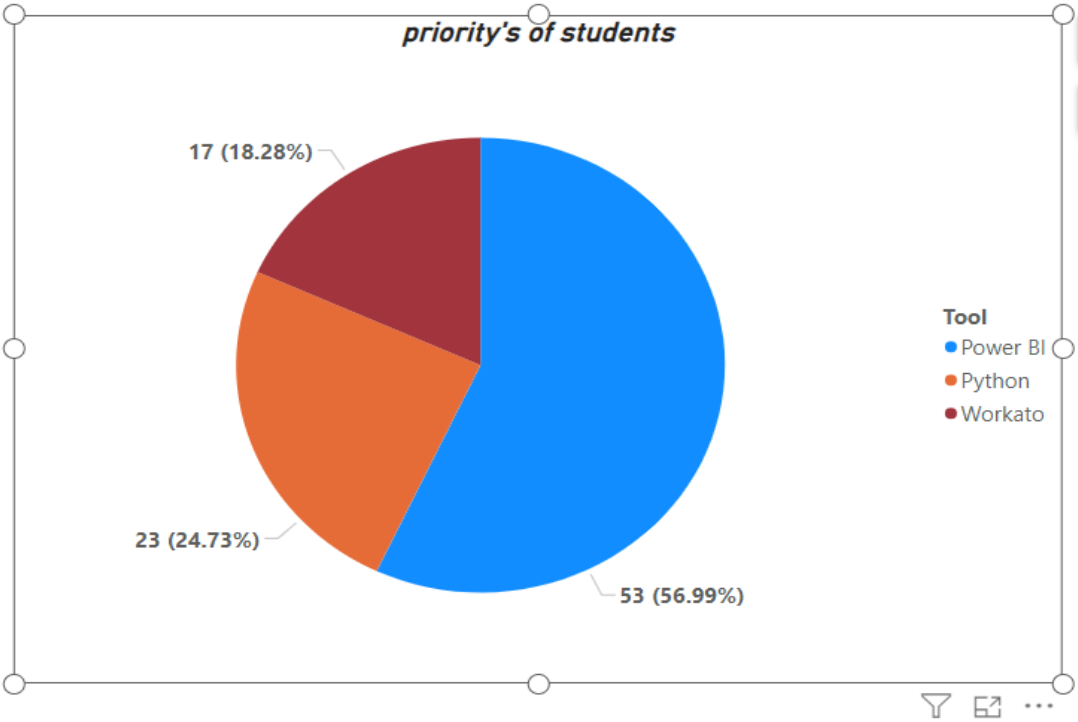


DAY-24


VOTES FOR WORKATO
17
WORKATO_PRIME
VOTES FOR POWERBI
53
POWERBI_PRIME
VOTES FOR PYTHON
23
PYTHON_PRIME
20 (21.51%)
53 (56.99%)
20 (21.51%)
Power BI
1
2
3


23 (24.73%)
39 (41.94%)
31 (33.33%)
Python
3
2
1


17 (18.28%)
42 (45.16%)
34 (36.56%)
Workato
2
3
1


priority's of students
17 (18.28%)
23 (24.73%)
53 (56.99%)
Tool
Power BI
Python
Workato

Count of Name of the Student by Tool