Statement explanation

- **Important notes:**

- \"{}\" this part will replace with input data in all statement.
- session.run(statemant.format(inputs)) this part used to run statement and give inputs used in statement.
- (e:names) this part refers to all nodes in database have “names” label.
- -[w:youraddress]-> this part refers to there is relationship called “youraddress” between two nodes .

Statements :

1- **(match (e:names)-[w:youraddress]->(d:addresses{{full_add: \"{}\" }}) return e.state)**

- Firstly, we will give this statement the address as input.
- And it will search for all node has “names” label and node has “addresses” label with this address.
- And return states of all users have this address.

2- **(match (e:names)-[w:workplace] ->(d:work{{ place:\"{}\" }}) return e.state)**

- Firstly, we will give this statement the company name as input.
- And it will search for all node has “names” label and node has “work” label with this company name.
- And return states of all users have this company.

3- **(create(e:names{{codeid: \"{}\" ,firstname: \"{}\" ,surname: \"{}\" ,dob: \"{}\" ,email: \"{}\" ,mobil: \"{}\" ,state: \"{}\" }}))**

- Firstly, we will give this statement the (code, first name, surname, dob, email, Mobil, state) as inputs.
- And it will create new node has “names” label and this info as node parameters.

4- **(match (e:work{{place:\"{}\"}}) return e)**

- Firstly, we will give this statement the company name as input.
- And it will search for all node has “work” label with this company name

5- **(create(e:work{{place:\"{}\"}}))**

- Firstly, we will give this statement the (company name) as inputs.
- And it will create new node has “work” label and this company name as node parameters.

Statement explanation

6- (match(e:names{{firstname:\"{}\"}}) ,(d:work{{place:\"{}\"}}) create(e)- [w:workplace]->(d))

- Firstly, we will give this statement the (first name, company name) as inputs.
- And it will create relationship between node has “names” label with this first name and node has “place” label with this company name.

7- (match (e:addresses{{full_add:\"{}\"}}) return e)

- Firstly, we will give this statement the address as input.
- And it will search for all node has “addresses” label with this address.

8- (match (e:names{{firstname:\"{}\"}})-[w:youraddress]->(d:addresses) return d.full_add)

- Firstly, we will give this statement the (first name, address) as input.
- And it will search for all node has “names” label with this first name and node has “work” label with this company name.
- And return address of this users.

9- (match (e:names{{firstname:\"{}\"}}) set e.state=\"{}\" return e)

- Firstly, we will give this statement the (first name, state) as input.
- And it will search for all node has “names” label with this first name.
- Once git this node it will update this node state to this new state.

- Note: I think the rest statement the same structure only parameters are different.

****