

23SA63 – BIGDATA & MODERN DATABASE SYSTEMS

PACKAGE

GRAPH DATABASE – Neo4j

// Create Person nodes

```
CREATE (rosa:Person {name: "Rosa", gender: "F"})
```

```
CREATE (karl:Person {name: "Karl", gender: "M"})
```

```
CREATE (fred:Person {name: "Fred", gender: "M"})
```

// Create Place nodes

```
CREATE (berlin:Place {city: "Berlin", country: "Germany", lives_since: 2020})
```

```
CREATE (paris:Place {city: "Paris", country: "France", lives_since: 1980})
```

```
CREATE (london:Place {city: "London", country: "UK"})
```

// Create FRIEND relationships

```
CREATE (rosa)-[:FRIEND]->(karl)
```

```
CREATE (karl)-[:FRIEND]->(fred)
```

// Create LIVES_IN relationships

```
CREATE (rosa)-[:LIVES_IN]->(berlin)
```

```
CREATE (karl)-[:LIVES_IN]->(paris)
```

```
CREATE (fred)-[:LIVES_IN]->(london)
```

// Create Project nodes and contributed_to relationships

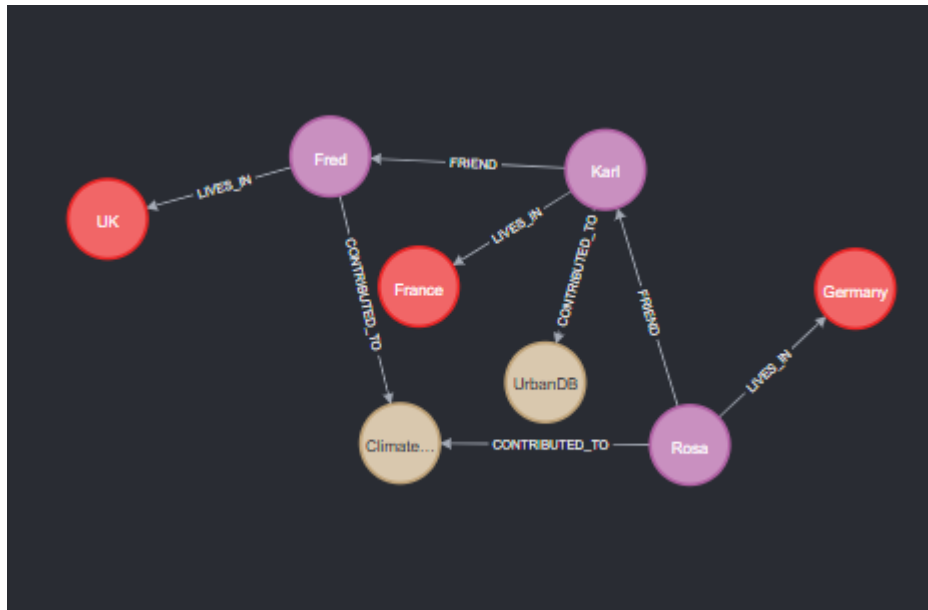
```
CREATE (proj1:Project {name: "ClimateGraph"})
```

```
CREATE (proj2:Project {name: "UrbanDB"})
```

```
CREATE (rosa)-[:CONTRIBUTED_TO {DBs_used: ["Neo4j", "MongoDB"]}]->(proj1)
```

```
CREATE (karl)-[:CONTRIBUTED_TO {DBs_used: ["PostgreSQL"]}]->(proj2)
```

```
CREATE (fred)-[:CONTRIBUTED_TO {DBs_used: ["Neo4j", "Redis"]}]->(proj1)
```



```
MATCH (p:Person {name: "Rosa"})
```

```
SET p.dob = date("1990-05-12")
```

Set 1 property, completed after 3 ms.

```
MATCH (n)
```

```
SET n.years_experience = 5
```

Set 11 properties, completed after 2 ms.

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place {city: "Berlin"})
```

```
RETURN p.name, p.gender
```

p.name	p.gender
"Rosa"	"F"

```
MATCH (rosa:Person {name: "Rosa"})-[:FRIEND]->(:Person)-[:FRIEND]->(fof:Person)
```

```
WHERE fof.name <> "Rosa"
```

```
RETURN DISTINCT fof.name, fof.gender
```

fof.name	fof.gender
"Fred"	"M"

```
MATCH (p:Person)
```

```
WHERE p.name STARTS WITH "Ka"
```

```
RETURN p.name
```

p.name
"Karl"

```
MATCH (p:Person)
```

```
WHERE p.name CONTAINS "os"
```

```
RETURN p.name
```

p.name
"Karl"

```
MATCH (p:Person)
```

```
WHERE NOT p.name ENDS WITH "d"
```

```
RETURN p.name
```

p.name
"Tom Hanks"
"Robert Zemeckis"
"Rosa"
"Karl"

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place {city: "Berlin"})
WHERE p.name IN ["Rosa", "Karl"]
RETURN p.name
```

p.name
"Rosa"

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place {city: "Berlin"})
MATCH (p)-[:FRIEND]->(f1)-[:FRIEND]->(f2)
RETURN DISTINCT f1.name AS Friend, f2.name AS FriendOfFriend
```

Friend	FriendOfFriend
"Karl"	"Fred"

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place)
WITH pl.city AS city, COUNT(p) AS population
RETURN city, population
```

ORDER BY population DESC

city	population
"Berlin"	1
"Paris"	1
"London"	1

MATCH (p:Person)

WHERE p.years_experience >= 9 AND p.years_experience <= 12

RETURN p.name, p.years_experience

(no changes, no records)

MATCH (p:Person)-[r:CONTRIBUTED_TO]->(:Project)

WHERE "Neo4j" IN r.DBs_used

RETURN DISTINCT p.name

p.name
"Rosa"
"Fred"

MATCH (p:Person)-[r:CONTRIBUTED_TO]->(:Project)

WHERE "MongoDB" IN r.DBs_used

MATCH (p)<-[:FRIEND]-(f:Person)

RETURN DISTINCT f.name

(no changes, no records)

```
MATCH (rosa:Person {name: "Rosa"})-[:FRIEND]->(f:Person)
WHERE NOT (f)-[:CONTRIBUTED_TO]->(:Project)
RETURN f.name
```

(no changes, no records)

```
MATCH (:Person)-[:CONTRIBUTED_TO]->(p:Project {name: "project A"})
RETURN COUNT(*) AS member_count
```

member_count
0

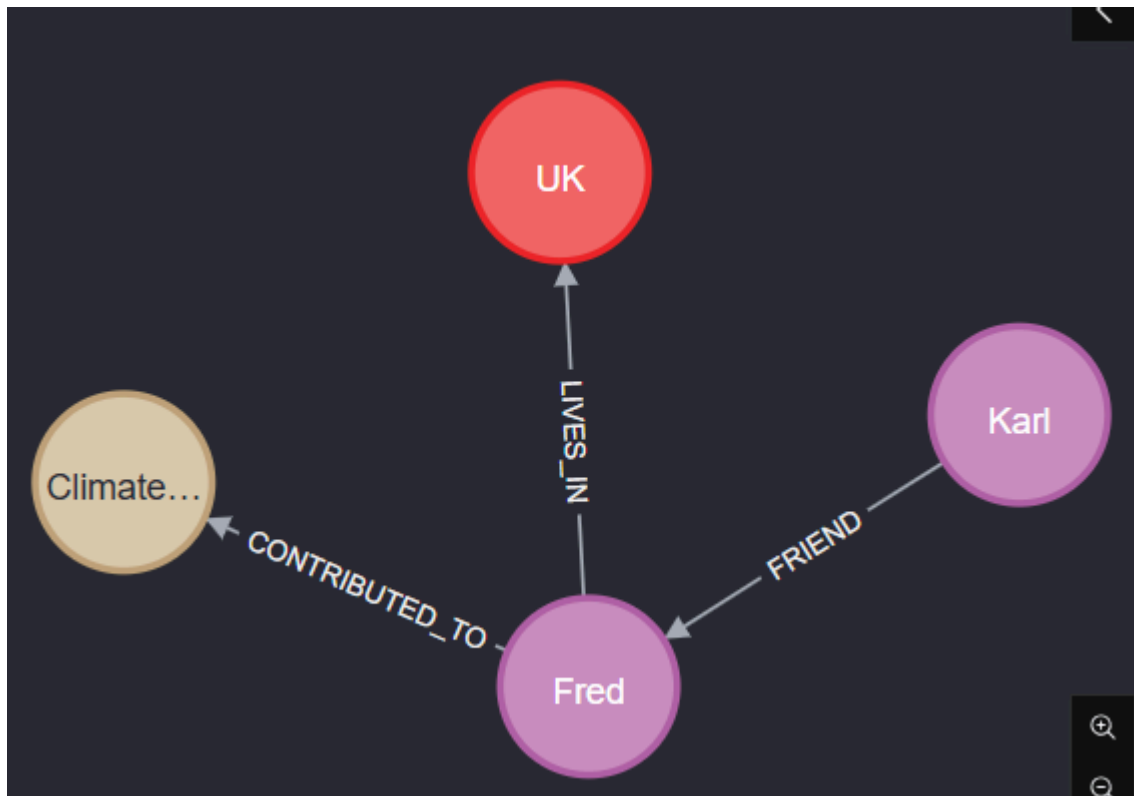
```
MATCH (:Person {name: "Rosa"})-[:FRIEND]->(f:Person)
RETURN f.name
ORDER BY f.name DESC
```

f.name
"Karl"

```
MATCH (karl:Person {name: "Karl"})-[:FRIEND]->(f:Person)-[r:CONTRIBUTED_TO]->(:Project)
WHERE "Neo4j" IN r.DBs_used
RETURN DISTINCT f.name
```

f.name
"Fred"

```
MATCH (fred:Person {name: "Fred"})-[r]-(n)
RETURN fred, r, n
```



```
MATCH (p:Person)
```

```
RETURN DISTINCT p.name
```

p.name
"Tom Hanks"
"Robert Zemeckis"
"Rosa"
"Karl"
"Fred"

```
// UNION DISTINCT
```

```

MATCH (p:Person)
RETURN p.name AS label

UNION

MATCH (pl:Place)
RETURN pl.city AS label

```

label	"Paris"
"Tom Hanks"	"London"
"Robert Zemeckis"	
"Rosa"	
"Karl"	
"Fred"	
"Berlin"	

```

CALL {
  MATCH (p:Person)
  RETURN p.name AS label
  UNION ALL
  MATCH (pl:Place)
  RETURN pl.city AS label
}
RETURN label, COUNT(*) AS occurrences
ORDER BY occurrences DESC

```


	label	occurrences
1	"Tom Hanks"	1
2	"Robert Zemeckis"	1
3	"Rosa"	1
4	"Karl"	1
5	"Fred"	1
6	"Berlin"	1
7	"Paris"	1
8	"London"	1

MATCH (p:Person)

SET p.age = 44,

p.email = p.name + "@gmail.com"

Set 10 properties, completed after 3 ms.

MATCH (:Person {name: "Rosa"})-[:FRIEND]->(f:Person)

WHERE f.age > 30

RETURN f.name, f.age

f.name	f.age
"Karl"	44

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place {city: "Berlin"})
WHERE p.name STARTS WITH "R"
RETURN p.name
```

p.name
"Rosa"

```
MATCH (p:Person)
WHERE p.email =~ ".*@gmail\\.com"
RETURN p.name, p.email
```

	p.name	p.email
1	"Tom Hanks"	"Tom Hanks@gmail.com"
2	"Robert Zemeckis"	"Robert Zemeckis@gmail.com"
3	"Rosa"	"Rosa@gmail.com"
4	"Karl"	"Karl@gmail.com"
5	"Fred"	"Fred@gmail.com"

```
MATCH (p:Person)-[:FRIEND]->(f:Person)
WITH p, COUNT(f) AS friend_count
WHERE friend_count > 1
```

```
RETURN p.name, friend_count
```

```
(no changes, no records)
```

```
MATCH (p:Person)-[:LIVES_IN]->(pl:Place {city: "London"})
```

```
RETURN DISTINCT p.name
```

```
p.name
```

```
"Fred"
```

```
MATCH (p:Person)
```

```
RETURN
```

```
  COUNT(p) AS total_persons,
```

```
  MAX(p.age) AS max_age,
```

```
  MIN(p.age) AS min_age,
```

```
  percentileCont(p.age, 0.5) AS median_age,
```

```
  percentileDisc(p.age, 0.75) AS percentile_75,
```

```
  SUM(p.years_experience) AS total_experience,
```

```
  STDEV(p.years_experience) AS experience_stdev
```

total_persons	max_age	min_age	median_age	percentile_75	total_experience	experience_stdev
5	44	44	44	44	25	0.0

```
MATCH (n)
```

```
WHERE n.age > 50
```

```
SET n.category = 'senior'
```

```
WITH n
```

```
WHERE n.age >= 30 AND n.age <= 50
```

```
SET n.category = 'middle age'
```

```
WITH n
```

```
WHERE n.age < 30
```

SET n.category = 'junior'

(no changes, no records)

```
:param people => [  
  {name: "Tom", age: 54},  
  {name: "Alex", age: 44},  
  {name: "Susan", age: 20}  
]
```

REDIS – Redis Cloud

The image displays a sequence of six Redis Cloud command-line interface (CLI) sessions, each showing a command entered in the left pane and its output in the right pane.

- Session 1:** Commands include `SET user:username "Kiran Kumar"`, `SET user:email "24pa12@psgtech.ac.in"`, `SET user:login_count 0`, `INCR user:login_count`, `GET user:username`, and `GET user:email`. The output for `GET user:login_count` is `"1"`.
- Session 2:** Commands include `LPUSH task_queue "Task 1: Check login details"`, `LPUSH task_queue "Task 2: Save your data"`, `LPUSH task_queue "Task 3: Make daily report"`, `LPUSH task_queue "Task 4: Delete old files"`, and `LPUSH task_queue "Task 5: Send message to user"`. The output for `LPUSH task_queue Task 5...` is `15`.
- Session 3:** Commands include `LPOP task_queue` and `LPOP task_queue`. The output for `LPOP task_queue` is `"Task 4: Delete old files"`.
- Session 4:** Command is `LRANGE task_queue 0 -1`. The output is an array: `["Task 3: Make daily report", "Task 2: Save your data"]`.
- Session 5:** Commands include `SADD post1:tags "tech" "redis" "NoSQL"`, `SADD post2:tags "tech" "database" "cache"`, and `SMEMBERS post1:tags`. The output for `SMEMBERS post1:tags` is an array: `["tech", "redis"]`.
- Session 6:** Command is `SINTER post1:tags post2:tags`. The output is an array: `["tech"]`.

Between the fourth and fifth sessions, there are two buttons: `+ Code` and `+ Markdown`.

```
1 SUNION post1:tags post2:tags
```

Command SUNION post1:tags post2...

```
[
  "tech",
  "redis",
]
```

```
1 HSET user:1 name "Kiran" email: "24pa12@psgtech.ac.in"
  age "21"
2 HGETALL user:1
```

Command HGETALL user:1

```
{
  "name": "Kiran",
  "email": "24pa12@psgtech.ac.in",
}
```

```
1 HSET user:1 email "newmail@example.com"
2 HDEL user:1 age
3 HLEN user:1
```

Command HLEN user:1

```
3
```

```
1 ZADD leaderboard 150 "Player1" 200 "Player2" 170
  "Player3"
```

Command ZADD leaderboard 150 Pl...

```
3
```

```
1 ZREVRANGE leaderboard 0 2 WITHSCORES
```

Command ZREVRANGE leaderboard 0...

```
[
  "Player2",
  "200",
]
```

```
1 ZINCRBY leaderboard 20 "Player1"
```

Command ZINCRBY leaderboard 20 ...

```
"170"
```

```
1 ZREVRANGE leaderboard 0 -1 WITHSCORES
```

Command ZREVRANGE leaderboard 0...

```
[
  "Player2",
  "200",
]
```

```
1 ZRANGE leaderboard 0 0 WITHSCORES
```

Command ZRANGE leaderboard 0 0 ...

```
[
  "Player1",
  "170"
]
```

```
1 ZREM leaderboard "Player1"
```

Command ZREM leaderboard Player1

```
1
```

```
1 SUBSCRIBE chat:room1
```

Command SUBSCRIBE chat:room1

CHANNEL	MESSAGE	TIME AGO
chatroom1	"Hello from user A"	2m

```
1 PUBLISH chat:room1 "Hello from user A"
```

Command PUBLISH chat:room1 Hell...

1

```
1 XADD chat:room1 * sender "UserA" message "Hello everyone"
2 XREAD COUNT 5 STREAMS chat:room1 0
```

Command XREAD COUNT 5 STREAMS c...

[
 [
 "chat:room1",

```
1 SUBSCRIBE chat:room2
```

Command SUBSCRIBE chat:room2

CHANNEL	MESSAGE	TIME AGO
chatroom2	"Hi from user B"	2m

```
1 PUBLISH chat:room2 "Hi from user B"
```

Command PUBLISH chat:room2 Hi f...

1

```
1 XADD chat:room2 * sender "UserB" message "Welcome to room 2"
```

Command XADD chat:room2 * sende...

"1761124227854-0"

```
1 XREAD COUNT 5 STREAMS chat:room2 0
```

Command XREAD COUNT 5 STREAMS c...

[
 [
 "chat:room2",

```
1 PFADD users:asia "user1" "user2" "user3"
```

Command PFADD users:asia user1 ...

0

```
1 PFADD users:europe "user4" "user5"
```

Command PFADD users:europe user...

0

1 PFADD users:america "user1" "user5"

Command PFADD users:america use...
1

1 PFMERGE users:global users:asia users:europa
users:america

Command PFMERGE users:global us...
"OK"

1 PFCOUNT users:global

Command PFCOUNT users:global
5

1 XADD notifications * message_id "msg1" content "promo"
1" recipient "UserA"

Command XADD notifications * me...
"1761124477881-0"

1 XADD notifications * message_id "msg2" content "Alert"
recipient "UserB"

Command XADD notifications * me...
"1761124499033-0"

1 XADD notifications * message_id "msg3" content "Update"
recipient "UserC"

Command XADD notifications * me...
"1761124522802-0"