* **Import Data:** MATCH (n) DETACH DELETE n;

**#LOAD CSV WITH HEADERS FROM:**

'https://vbatushkov.bitbucket.io/log\_of\_calls.csv' AS line

MERGE (c1:City { name: line.from\_city })

MERGE (p1:Person { name: line.from\_name, number: line.from\_number, gender: line.from\_gender })

MERGE (p1)-[:FROM]->(c1)

MERGE (c2:City { name: line.to\_city })

MERGE (p2:Person { name: line.to\_name, number: line.to\_number, gender: line.to\_gender })

MERGE (p2)-[:FROM]->(c2)

CREATE (c:Call { from: datetime(line.from\_dt),

to: datetime(line.to\_dt),

duration: duration.between(datetime(line.from\_dt), datetime(line.to\_dt)).minutes })

CREATE (p1)-[:OUT]->(c)<-[:IN]-(p2);

* **To get all results: MATCH (all) RETURN all**
* **To get 5 values from your data: MATCH (all) RETURN all LIMIT 3**
* **To get 5 values from your data: MATCH (all) RETURN all LIMIT 5**
* **To get values from your data:** MATCH (all) RETURN all LIMIT 7
* **To get the call relation of a city :** MATCH (node1) –[rel] (node2) RETURN node1, rel1, node2 LIMIT 1
* **Calls made during month 1 with duration of 1 minute:**

MATCH (c:Call)

WHERE c.from.month = 5 and c.duration = 1

RETURN count( c ) as calls

* **Persons(MALE), who got a call from TIFFANY during month 5 of the log period.**

MATCH (p1:Person)-[:OUT]->(c:Call)<-[:IN]-(p2:Person)

WHERE p1.name = “TIFFANY” and c.from.month=5 and p2.gender = “MAN”

RETURN p2

* **Persons**, who got a call from “**Robert**”

MATCH (p1:Person)-[:OUT]->(:Call)<-[:IN]- (p2:Person)

WHERE p1.name = "Robert"

RETURN p2

* **Calls received or dialed by NATALIA**

MATCH (person:Person {name: ‘NATALIA’})

RETURN person

* **Incoming or outgoing calls from Pattaya**

MATCH (person:Person {name: ‘NATALIA’})

MATCH (person1:Person {name: ‘HADLEIGH’})

RETURN person, person1

* **Individual call from city Pattaya**

MATCH (p:Person)-[:FROM]->(c:city)

WHERE p.name= “NATALIA”

Return p,c

* **How many calls were missed in May?**

MATCH (c:Call)

WHERE c.from.month = 5 AND c.duration = 0

RETURN (c)

* **Find a name of man, who received a call from Tiffany in May?**

MATCH (c:City)<-[:FROM]-(p1:Person)-[:OUT]->(cc:Call)<-[:IN]-(p2:Person)-[:FROM]->(c)

RETURN c.name, count(cc) as total

ORDER BY total

* **How many women from Pattaya received calls from Bangkok men?**

MATCH (:City { name: "Pattaya" })<-[:FROM]-(p:Person { gender: "Woman" })-[:IN]->(:Call)<-[:OUT]-(:Person { gender: "Man" })-[:FROM]->(:City { name: "Bangkok" })

RETURN c

* **On date 25 of April, find the woman who has least total duration of conversations?**

MATCH (p:Person)-[]->(c:Call)

WHERE p.gender = "Woman" AND c.from.month = 4 AND c.from.day = 25 AND c.duration> 0

RETURN p.name as name, sum(c.duration) as minutes

ORDER BY minutes

LIMIT 10

* **On date 25 of April, find the woman who has highest call duration of conversations?**

MATCH (p:Person)-[]->(c:Call)

WHERE p.gender = "Woman" AND c.from.month = 4 AND c.from.day = 25 AND c.duration>10

RETURN p.name as name, sum(c.duration) as minutes

ORDER BY minutes

LIMIT 10

* **How many individual pairs of people called to each other?**

MATCH (p1:Person)-[:OUT]->(:Call)<-[:IN]-(p2:Person)

WHERE (p1)-[:IN]->(:Call)<-[:OUT]-(p2) AND id(p1) > id(p2)

RETURN p1

* **Top 10 persons with highest number of calls from Bangkok:**

MATCH (c:City { name: "Bangkok" })<-[:FROM]-(p1:Person)-[:OUT]->(cc:Call)<-[:IN]-(p2:Person)

RETURN p1.name, count(cc) as total

ORDER BY total DESC

LIMIT 10

* **All persons received 10 minutes calls in July and made 5 minutes calls:**

MATCH (c1:Call { duration: 10 })<-[:IN]-(p:Person)

WHERE (p)-[:OUT]->(:Call { duration: 5 }) AND c1.from.month = 7

RETURN p

* **Find a city with the lowest number of internal city calls?**

Match (c1:City)<-[:FROM]-(p1:Person)-[:OUT]->(c:Call)<-[:IN]-(p2:Person)-[:FROM]->(c2:City) where c1 = c2 return c1,count(c) as total order by total

LIMIT 1

* **Find a city with the highest number of internal city calls?**

Match (c1:City)<-[:FROM]-(p1:Person)-[:OUT]->(c:Call)<-[:IN]-(p2:Person)-[:FROM]->(c2:City) where c1 > c2 return c1,count(c) as total order by total

LIMIT 1

* **How many calls Robert did , and to whom?**

MATCH (p1:Person)-[:OUT]->(:Call)<-[:IN]- (p2:Person)

WHERE p1.name = "Robert"

RETURN p2

* **How many cities are we making use for analyzing the data of calling?**

MATCH (callerCity:City)<-[:FROM]-(caller:Person)-[:OUT]->(c:Call)<-[:IN]-(receiver:Person)-[:FROM]->(receiverCity:City)

WHERE callerCity.name = receiverCity.name RETURN callerCity.name as name, count(c) as total

* **Total Number of call for all the cities?**

MATCH (callerCity:City)<-[:FROM]-(caller:Person)-[:OUT]->(c:Call)<-[:IN]-(receiver:Person)-[:FROM]->(receiverCity:City)

WHERE callerCity.name = receiverCity.name RETURN callerCity.name as name, count(c) as total

* **CREATE**

Create CREATE (friend:Person {name: 'Mark'})

RETURN friend

* **VIEW/READ**

MATCH (:Person {name: 'Jane'})-[:FROM]->(C:City)  
RETURN C

* **View Without Graph**

MATCH (:Person {name: 'Jane'})-[:FROM]->(C:City)  
RETURN C.name

* **Updating Data with Cypher**

MATCH (p:Person {name: 'Jane'})  
SET p.birthdate = date('1980-01-01')  
RETURN p

* **DELETE**

MATCH (j:Person {name: 'Jane'})-[r:IS\_FRIENDS\_WITH]->(m:Person {name: 'Mark'})  
DELETE r