Neo4j

- מערכת ניהול נתונים המבוססת על מבנה של גרף (Graph Store).
- מורכב משני אלמנטים, קודקודים וקשתות.
- משמש לשמירת מידע הקשור לרשתות וקשרים חברתיים.
- מאפשר למצוא חברים של קודקוד וחברים של חברים עד דור 10.

Compare to SQL

RDBMS	Neo4J
Table	Graph
Row	Node
Column and Data	Property and its values
Constraint	Relationship
Join	Traversal

CQL - Cypher Query Language

- שפת השאילתות לבסיסי נתונים של Neo4j.
- זוהי שפה דקלרטיבית
- התחביר פשוט וקריא
- השפה משמשת כדי ליצור ולאחזר את היחסים בין הנתונים מבלי להשתמש בשאילתות מורכבות.

Common CQL Clauses

CQL Clause	Usage
CREATE	Create nodes, relationships and properties
MATCH	Retrieve data about nodes, relationships and properties
RETURN	Return query results
WHERE	Provide conditions to filter retrieval data
DELETE	Delete nodes and relationships
REMOVE	Delete properties of nodes and relationships
ORDER BY	Sort retrieved data
SET	Add or update labels

- נשתמש באתר המדמה לנו עבודה מול Neo4j

http://Console.neo4j.org

- כדי שיהיה לנו מסודר בעיניים נמחוק קודם את המבנה שקיים כבר באתר:

- MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r

- אנו ניצור מבנה גרף חדש המתאר משפחה:

הורים: מיכל וקובי

ילדים: יואב ותותי

חיות מחמד: לאסי, קטי ופישי.

Create

- יצירת קודקוד (Node)
- Create (<node-name>:<label-name> {

  <Property1-name>:<Property1-Value>,

  ..... ,

  <PropertyN-name>:<PropertyN-Value> })

←בפקודת Create אחת ניתן ליצור רשימה של צמתים.

Create

```
Create (Michal:Person {name:'Michal', age: 36,gender:"female"}),
        (Koby:Person {name:'Koby', age: 39,gender:"male"}),
        (Yoav:Person {name:'Yoav', age: 10,gender:"male"}),
        (Tuti:Person {name:'Tuti', age: 4,gender:"female"}),
        (Lassy:Pet {name:'Lassy', age: 2,gender:"female" ,type: "dog"}),
        (Caty:Pet {name:'Caty', age: 4,gender:"male", type: "cat"}),
        (Fishy:Pet {name:'Fishy', age: 6,gender:"male", type: "fish"})
```

Create


```
Query:
CREATE (Michal:Person {name:'Michal', age: 36,gender:"female"}),        (Koby:Person {name:'Koby', age: 39,gender:"male"}),     (Yoav:Person {name:'Yoav', age: 10,gender:"male"}),
(Tuti:Person {name:'Tuti', age: 4,gender:"female"}),    (Lassy:Pet {name:'Lassy', age: 2,gender:"female" ,type: "dog"}),       (Caty:Pet {name:'Caty', age: 4,gender:"male", type:
"cat"}),        (Fishy:Pet {name:'Fishy', age: 6,gender:"male", type: "fish"})


Query took 74 ms and returned no rows.
Updated the graph - created 7 nodes set 24 properties Result Details
```

Relations

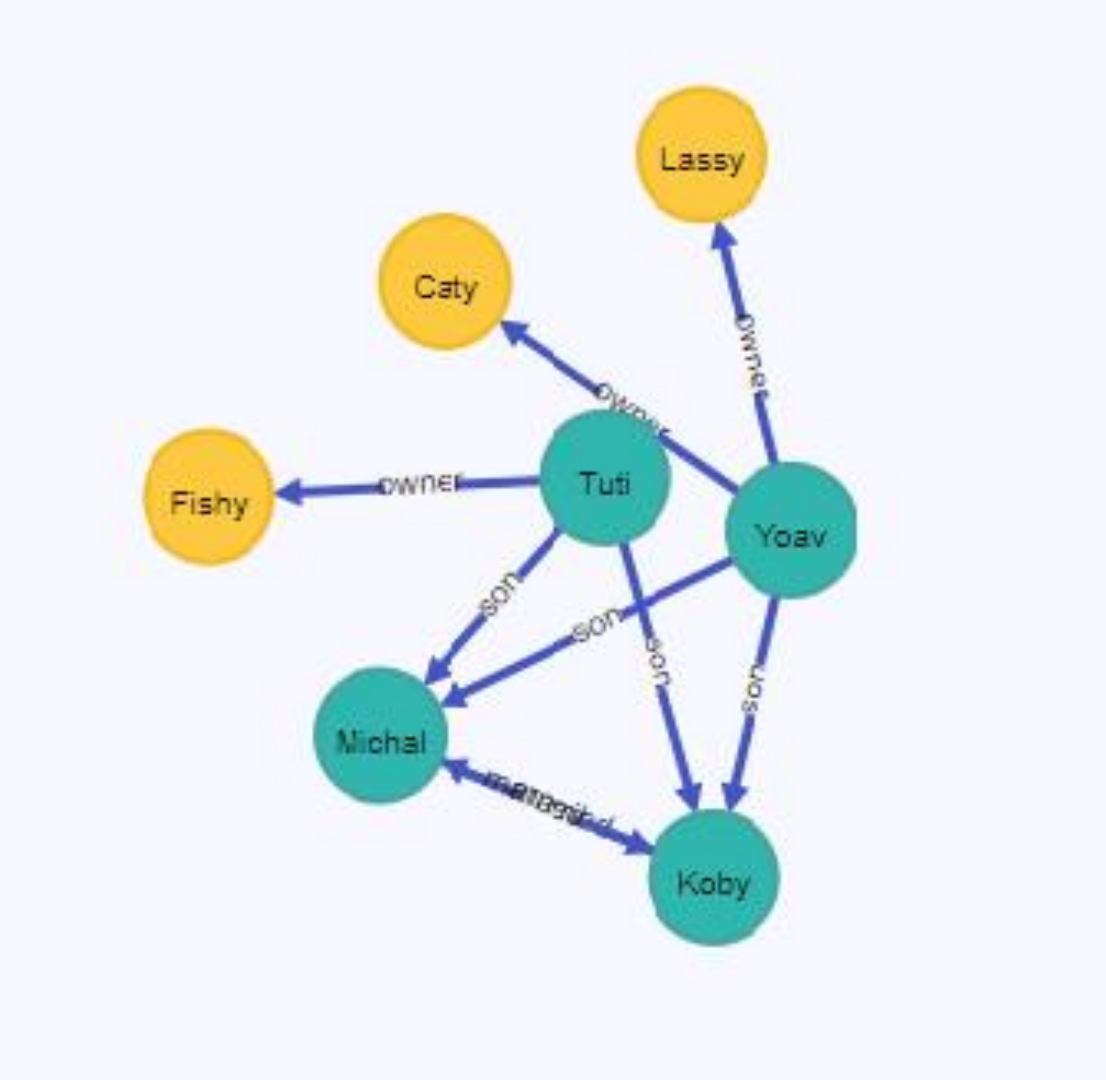
- יצירת יחסי גומלין בין הקודקודים:

Create (node1)-[:RelationshipType]->(node2)

←שוב, בפקודת Create אחת ניתן ליצור רשימה של יחסים.

Relations

```
Create (Koby)-[:married]->(Michal),
        (Michal)-[:married]->(Koby),
        (Yoav)-[:son]->(Michal),
        (Tuti)-[:son]->(Michal),
        (Yoav)-[:son]->(Koby),
        (Tuti)-[:son]->(Koby),
        (Yoav)-[:owner]->(Lassy),
        (Yoav)-[:owner]->(Caty),
        (Tuti)-[:owner]->(Fishy)
```


Lassy
Caty
owner
owner
Fishy
owner
Tuti
Yoav
son
son
son
son
Michal
Koby

Match

- בעזרת הפקודה Match מאחזרים מידע מבסיס הנתונים

```
MATCH (node:label)
RETURN node
```

לדוגמא, החזר את כל הקודקודים שהם מסוג Person

```
MATCH (n:Person)
RETURN n
```

Match

- בנוסף, ניתן להחזיר את כל הקודקודים לפי סוג היחס אליהם:

```
MATCH (node:label)-[: Relationship]->(n)
RETURN n
```

- מה תהיה התוצאה של השאילתא הבאה?

```
MATCH (p:Person {gender:"male"})-[:owner]->(n)
RETURN n.name
```

- דרך נוספת לכתוב את השאילתא:

```
MATCH (n)<-[:owner]-(p:Person {gender:"male"})
RETURN n.name
```

More Examples

- Match (n:Person) return n.name

n.name
Michal
Koby
Yoav
Tuti

More Examples

- Match (a)-->(b{name:'Michal'}) RETURN a.name

a.name
Tuti
Yoav
Koby

More Examples

- Match (a{name:'Yoav'})-->(b:Pet) RETURN b.name

b.name
Lassy
Caty

More Examples

- Match (a{name:'Koby'})-[:married]->(b) RETURN b.name

b.name
Michal

More Examples

- רשימת האנשים הקשורים ל-koby (לא משנה מהות הקשר):

```
MATCH (a:Person)-->(b{name:‘Koby'}) RETURN a
```

- כל הקשרים שיש לקובי (לא רק קשרים ישירים):

```
MATCH (a {name:'Koby'})-[*]-(b) RETURN DISTINCT b
```

- כל הקשרים מסוג son ומסוג married:

```
MATCH (a)-[:son|:married]-(b) RETURN DISTINCT b
```

Where

- לפי תכונות-

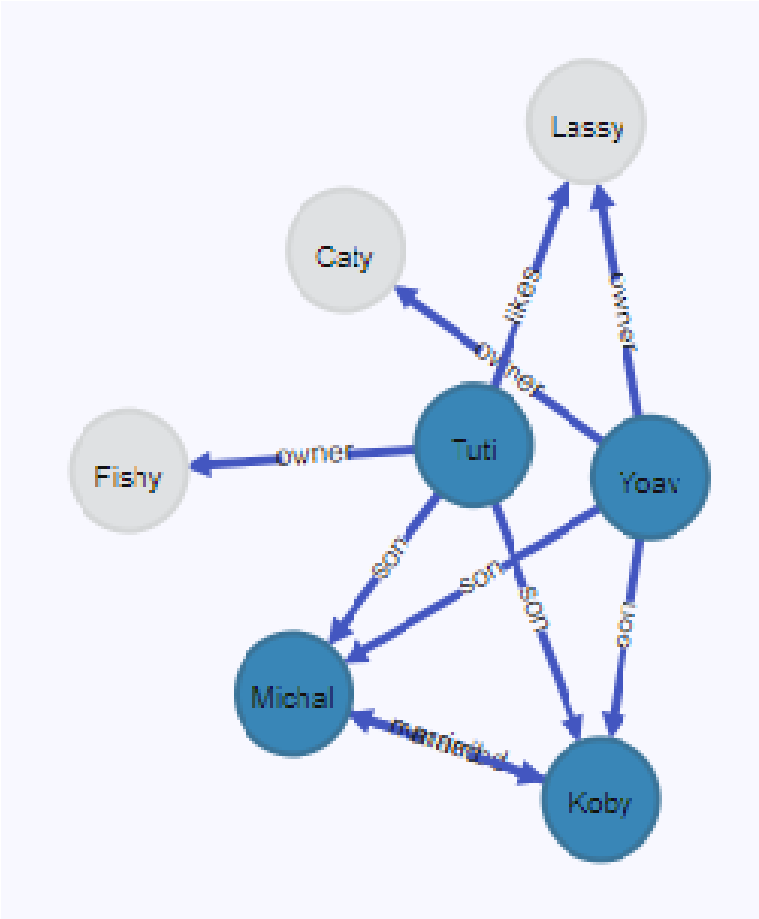
```
MATCH (Person)
WHERE Person.gender = "male" AND Person.age>=18
RETURN Person
```

- לפי קשרים-

```
MATCH (n)
WHERE (n)<-[:son]-({name: "Yoav"})
RETURN n
```

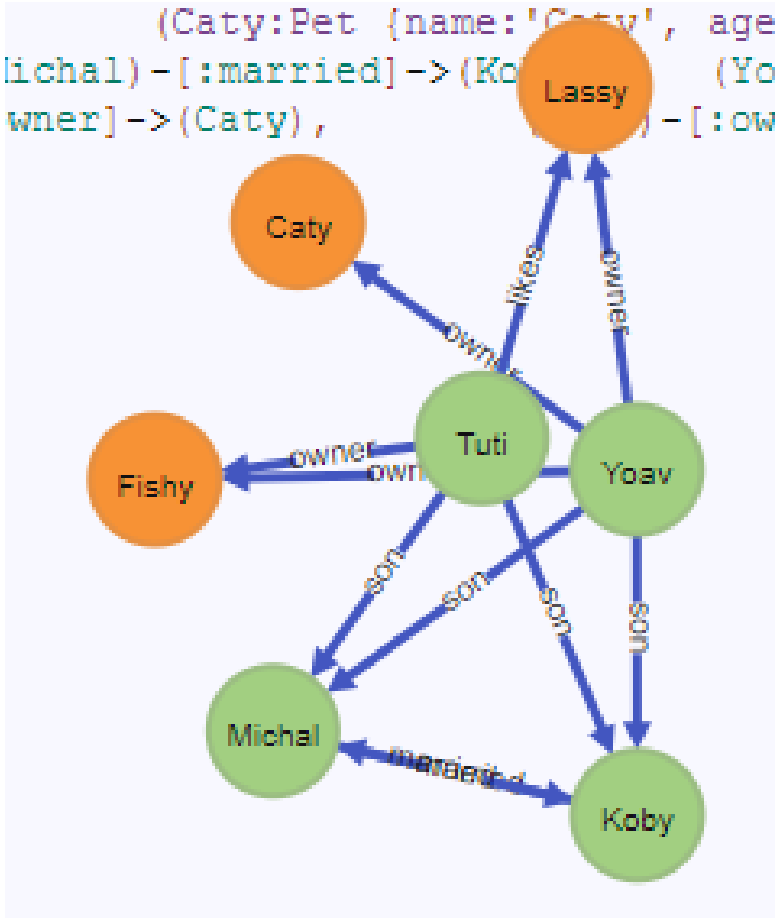
Add Relation

- MATCH (a:Person),(b:Pet) WHERE a.name = 'Tuti' AND b.name = 'Lassy' CREATE (a)-[r1:likes]->(b)



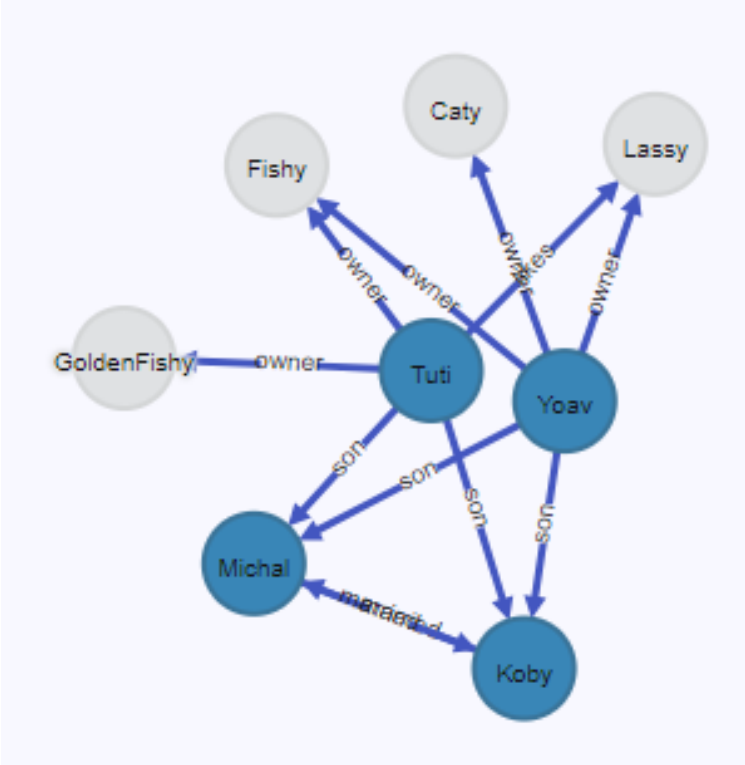
Add Relation

- MATCH (a:Person),(b:Pet) WHERE a.name = 'Yoav' AND b.name = 'Fishy' CREATE (a)-[r1:owner]->(b)



Add Relation

- CREATE (GoldenFishy:Pet {name: "GoldenFishy", age:0.5,gender:"male", type: "fish"})
- MATCH (a:Person),(b:Pet) WHERE a.name = 'Tuti' AND b.name = 'GoldenFishy' CREATE (a)-[r1:owner]->(b)



Remove

- מאפשרת מחיקת תכונות של קודקוד

MATCH (node:label{properties . . . . })

REMOVE node.property RETURN node

- לדוגמא:

MATCH (GoldenFishy:Pet {name: "GoldenFishy", age:0.5,gender:"male", type: "fish"}) REMOVE GoldenFishy.gender

MATCH (n:Pet) RETURN n

n
(11:Pet {age:2, gender:"female", name:"Lassy", type:"dog"})
(12:Pet {age:4, gender:"male", name:"Caty", type:"cat"})
(13:Pet {age:6, gender:"male", name:"Fishy", type:"fish"})
(14:Pet {age:0.5, name:"GoldenFishy", type:"fish"})

Delete

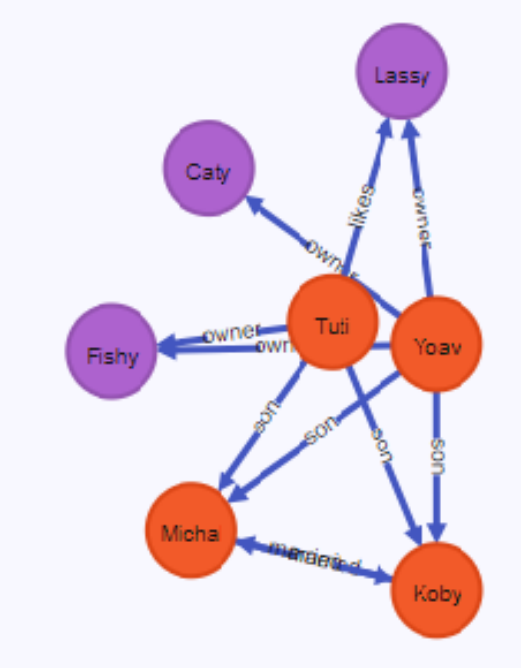
- מחיקת כל הקודקודים:

MATCH (n) detach delete n

- מחיקה של קודקוד ספציפי (כולל הקשרים שלו):

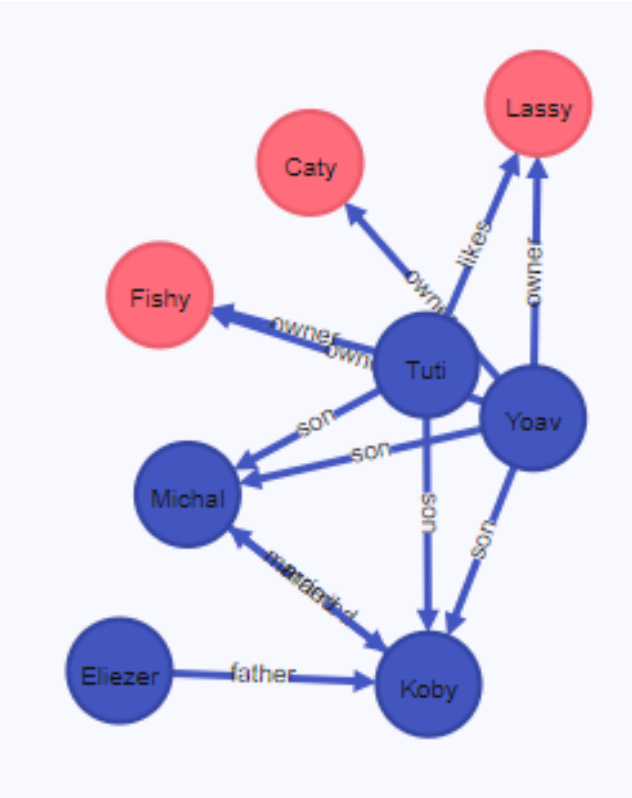
MATCH (GoldenFishy:Pet {name: "GoldenFishy", age:0.5,gender:"male", type: "fish"})

DETACH DELETE GoldenFishy



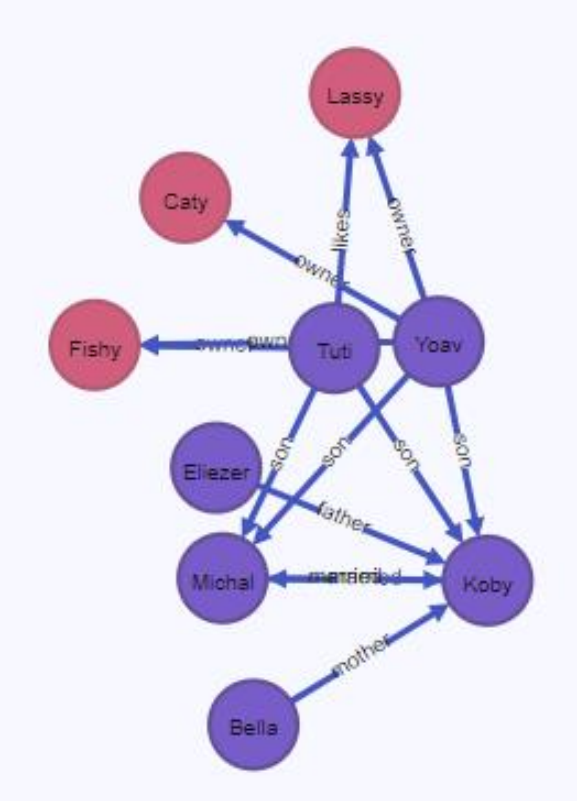
Add Generation

- Create(Eliezer:Person {name:'Eliezer', age:82,gender:"male"})
- MATCH (a:Person),(b:Person) WHERE a.name = 'Eliezer' AND b.name = 'Koby' CREATE (a)-[r1:father]->(b)



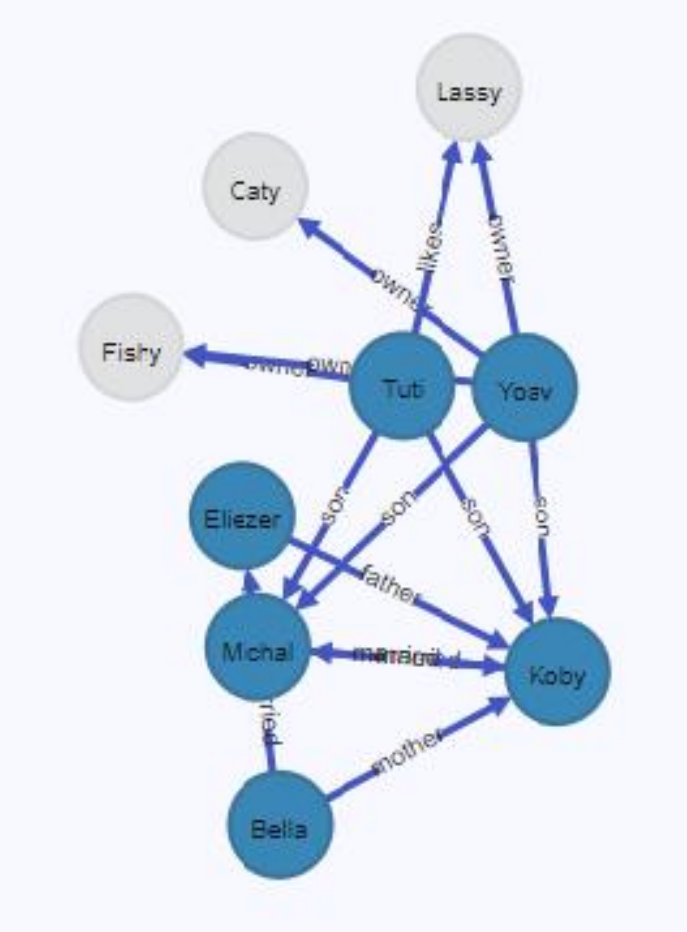
Add Generation

- create(Bella:Person {name:'Bella', age:78,gender:"female"})
- MATCH (a:Person),(b:Person) WHERE a.name = 'Bella' AND b.name = 'Koby' CREATE (a)-[r1:mother]->(b)



Add Generation

- MATCH (a:Person),(b:Person) WHERE a.name = 'Bella' AND b.name = 'Eliezer' CREATE (a)-[r1:ex_married]->(b)



Basic Aggregation Functions

Function	Description
Count	Returns the number of rows returned by MATCH command
Max	Returns the maximum value from a set of rows returned by MATCH command
Min	Returns the minimum value from a set of rows returned by MATCH command
Sum	Returns the summation value of all rows returned by MATCH command
Avg	Returns the average value of all rows returned by MATCH command

Aggregation Functions

- ממוצע גילאים:

Match (n:Person) Return avg(n.age)

avg(n.age)
41.5

- רשימת כל הגילאים:

Match (n:Person) Return collect(n.age)

collect(n.age)
[36, 39, 10, 4, 82, 78]

With

MATCH (n) WITH n ORDER BY n.property RETURN collect(n.property)

- לדוגמא, רשימת כל השמות בסדר יורד עם הגבלה ל-3 שמות:

MATCH (n) WITH n ORDER BY n.name DESC LIMIT 3 RETURN collect(n.name)

collect(n.name)
[Yoav, Tuti, Michal]

With

- מה מבצעת השאילתא הבאה?

```
MATCH (c:Pet) WITH COLLECT(c) AS Pets MATCH (s:Person) WHERE ALL
(x IN Pets WHERE (s)-[:owner]->(x)) RETURN s.name
```

TF – IDF

TF-IDF

- מדד שמאפשר לנו לדרג משפטים לפי רמת הרלוונטיות שלהם לנושא.

$$tfidf(d) = \sum_{k=0}^{|Q|} \frac{\#k\ in\ d}{|d|} \log\left(\frac{|D|}{\#D\ with\ k}\right)$$

דוגמא:

- שאלה: מי אכל תפוח וגם בננה?

- משפט 1: "תפוח זהו פרי שקיים באדום וגם בירוק וגם בצהוב"
- משפט 2: "אם יוסי אכל בננה וגם תפוח הוא יהיה בריא וחזק מאוד כשיגדל"
- משפט 3: "אתמול יוסי אכל תפוח, שני אפרסקים וגם בננה"
- משפט 4: "אבא קנה אתמול תפוח, אפרסק וגם בננה"
- משפט 5: "אתמול ראיתי שהקוף אכל בננה"

מס' המילים במשפט	בננה	וגם	תפוח	אכל	מי	
9	0	2	1	0	0	משפט 1
12	1	1	1	1	0	משפט 2
8	1	1	1	1	0	משפט 3
7	1	1	1	0	0	משפט 4
5	1	0	0	1	0	משפט 5
	4	4	4	3	0	

TF-IDF (משפט 1): 1/9*log(5/4)+2/9*log(5/4) = 0.0323

TF-IDF (משפט 2): 1/12*log(5/3)+1/12*log(5/4)+1/12*log(5/4)+1/12*log(5/4) = 0.04271

TF-IDF (משפט 3): 1/8*log(5/3)+1/8*log(5/4)+1/8*log(5/4)+1/8*log(5/4) = 0.064

TF-IDF (משפט 4): 1/7*log(5/4)+1/7*log(5/4)+1/7*log(5/4)= 0.04153

TF-IDF (משפט 5): 1/5*log(5/3)+1/5*log(5/4)= 0.06375

שאלת מבחן:

- דרגו את המשפטים הבאים לפי דרגת הTF-IDF.

- Q: apples and apes
- D1: **apes and** monkeys eat bananas
- D2: a man walked down the street **and** saw an elephant.
- D3: monkeys eat **apples and** bananas
- D4: **apes** like to eat **apples**
- D5: would you like to eat **apples and** bananas?

שאלת מבחן נוספת:

- דרגו את המשפטים הבאים לפי דרגת הTF-IDF.

- Q: elephants with horns
- D1: Mary went on a date **with** Larry
- D2: Google will lock **horns with** Microsoft
- D3: I never knew **elephants** had **horns**
- D4: I saw **elephants with** big trunks
- D5: Bears **with** blue ears also have **horns**

ועוד אחת...

- Q: she has a nice home
- D1: I saw that **she** did not come **home** before I left to work
- D2: **she** must come **home** early tomorrow night
- D3: **she** is looking for **a nice home**
- D4: **she** said that **she** is smart
- D5: **she** is **nice** but not that smart