Presenters: Itay Rafee & Almog Jakov בס"ד

מטלה 3 – מסדי נתונים

(mongoDB)

א.

```
var beginners =
[
{"id":0,"Dep":"Industrial engineering","age":50,"Courses": [{"name":"math",grade:95},{"name":"database","grade":7},{"name":"algebra","grade":14}]},
{"id":1,"Dep":"CS","age":5,"Courses": [{"name":"math","grade":46}]},
{"id":2,"Dep":"CS","age":29,"Courses": [{"name":"math","grade":91},{"name":"database","grade":21},{"name":"algebra","grade":60}]},
{"id":3,"Dep":"Electrical Engineer","age":8,"Courses": [{"name":"math","grade":88},{"name":"database","grade":10},{"name":"algebra","grade":68},{"name":"logic","grade":33}]},
{"id":4,"Dep":"Constructor","age":26,"Courses":
[{"name":"math","grade":86},{"name":"database","grade":37},{"name":"algebra","grade":26},{"name":"logic","grade":95},{"name":"history","grade":32}]},
{"id":5,"Dep":"Industrial engineering","age":10,"Courses": [{"name":"math","grade":87},{"name":"database","grade":11}]},
{"id":6,"Dep":"Electrical Engineer","age":4,"Courses": [{"name":"math","grade":46},{"name":"database","grade":86},{"name":"algebra","grade":95}]},
{"id":7,"Dep":"Industrial engineering","age":52,"Courses": [{"name":"math","grade":82},{"name":"database","grade":48},{"name":"algebra","grade":68}]},
{"id":8,"Dep":"Constructor","age":53,"Courses":
[{"name":"math","grade":23},{"name":"database","grade":47},{"name":"algebra","grade":93},{"name":"logic","grade":48},{"name":"history","grade":67},{"name":"Chemistry","grade":48}]},
{"id":9,"Dep":"Industrial engineering","age":21,"Courses": [{"name":"math","grade":48},{"name":"database","grade":53},{"name":"algebra","grade":100},{"name":"logic","grade":22}]},
{"id":10,"Dep":"Industrial engineering","age":39,"Courses": [{"name":"math","grade":96},{"name":"database","grade":93},{"name":"algebra","grade":62}]},
{"id":11,"Dep":"Constructor","age":46,"Courses": [{"name":"math","grade":5},{"name":"database","grade":0},{"name":"algebra","grade":24},{"name":"logic","grade":63}]},
{"id":12,"Dep":"CS","age":15,"Courses": [{"name":"math","grade":22},{"name":"database","grade":54}]},
{"id":13,"Dep":"Constructor","age":13,"Courses": [{"name":"math","grade":82},{"name":"database","grade":67}]},
{"id":14,"Dep":"Constructor","age":21,"Courses": [{"name":"math","grade":14},{"name":"database","grade":13},{"name":"algebra","grade":2}]},
{"id":15,"Dep":"Electrical Engineer","age":35,"Courses": [{"name":"math","grade":66},{"name":"database","grade":41},{"name":"algebra","grade":64},{"name":"logic","grade":89}]},
{"id":16,"Dep":"Electrical Engineer","age":25,"Courses": [{"name":"math","grade":18},{"name":"database","grade":77},{"name":"algebra","grade":44},{"name":"logic","grade":4}]},
{"id":17,"Dep":"Electrical Engineer","age":38,"Courses": [{"name":"math","grade":67},{"name":"database","grade":26},{"name":"algebra","grade":86},{"name":"logic","grade":43}]},
{"id":18,"Dep":"CS","age":49,"Courses": [{"name":"math","grade":53},{"name":"database","grade":48}]},
{"id":19,"Dep":"Industrial engineering","age":59,"Courses": [{"name":"math","grade":4},{"name":"database","grade":76},{"name":"algebra","grade":0}]}
];
db.workers.insert(beginners);
```

ב.

```
db.workers.mapReduce (
          function () {
            for (var idx = 0; idx < this.Courses.length; idx++) {
              if (this.Dep == "CS" || this.Dep == "Electrical Engineer") {
              var key = {
                DEP: this.Dep,
                COURSE: this.Courses[idx].name
              }
              var value = {
                sum: this.Courses[idx].grade,
                count: 1
              }
              emit(key,value);
              }
            }
  },
          function(Ckey,Cvalues) {
            var reduced_var = {sum:0,count:0}
            for (var idx = 0; idx < Cvalues.length; idx++) {
              reduced_var.sum += Cvalues[idx].sum;
              reduced_var.count += Cvalues[idx].count;
            }
            return reduced_var;
  },
          {
                    out: "average_result",
                    finalize: function(Ckey,Cvalues) {
              var reduced_var = {[Ckey.COURSE]:Cvalues.sum/Cvalues.count}
              return reduced_var;
    }
          }
);
db.average_result.aggregate([{$group:{_id:{ "Dep" : "$_id.DEP"}, Averages: {$mergeObjects:"$value"} } },{ $out : "res" }]);
db.average_result.drop();
db.res.find().forEach( doc =>  print( tojson(doc._id.Dep)+tojson(doc.Averages) ) );
```

(Neo4j)

```
MATCH (dani:Person{name:'Dani'})-[:liked|:watched]->(movie:Movie)
WITH COLLECT(movie) AS dani_movies
MATCH (s:Person)
WHERE (s)-[:friend*1..2]-(:Person{name:'Dani'}) AND ALL (x IN dani_movies WHERE (s)-[:watched]->(x))
RETURN s
```

(elasticSearch) .1

```
curl -XPOST "http://localhost:9200/library/books" -H "Content-Type: application/json" -d "{\"BookName\": \"Harry Potter\",
\"AuthorName\": \"J.K. Rowling\", \"Genre\": \"Fantasy\", \"Publisher\": \"Bloomsbury Publishing\", \"PublishedYear\":
\"1997\", \"Synopsis\": \"Harry Potter, a young wizard who discovers his magical heritage on his eleventh birthday.\"}"
```

(elasticSearch) .2

```
curl -XGET http://localhost:9200/library/books/_search?pretty -H "Content-Type: application/json" -d "{\"query\": {\"bool\":
{\"must\":[{\"match\": {\"Genre\":\"Science Fiction\"}},{\"range\":{\"PublishedYear\":{\"gte\":2000}}},{\"match\":
{\"Synopsis\":\"Science Fiction\"}}, {\"match\": {\"Synopsis\":\"reality\"}}]}}}"
```

(X-path) .3

```
Cities/City[sum(institution/@num)>1000000]/@name
```

(Stream) .4

```
// should import:
// import java.util.stream.IntStream;
// import java.math.BigInteger;

public static void primeNumbersTillN(int n) {
      String s = IntStream.range(2, n).filter(value -> n % value == 0).
            filter(a -> BigInteger.valueOf(a).isProbablePrime(100)).
            mapToObj(String::valueOf).reduce((x, y) -> x + "\n" + y).orElse("");
      System.out.println(s);
   }
```

5. (RDF & SPARQL)

א.

ttr:Dani	dbp:id	23
ttr:Dani	dbp:age	70
ttr:Dani	dbo:parent	80
ttr:Michal	dbp:id	12
ttr:Michal	dbp:age	23
ttr:Michal	dbo:parent	23
ttr:Yaron	dbp:id	45
ttr:Yaron	dbp:age	49
ttr:Yaron	dbo:parent	67

ובקובץ XML:

```
<?xml version="1.0"?>

<rdf:RDF   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:ID="http://www.Person/ID#">

           <rdf:Description        rdf:about="http://www.Person/ID/23">
                      <ID:name>Dani</ID:name>
                      <ID:age>70</ID:age>
                      <ID:Father_id>80</ID:Father_id>
           </rdf:Description>

           <rdf:Description        rdf:about="http://www.Person/ID/12">
                      <ID:name>Michal</ID:name>
                      <ID:age>23</ID:age>
                      <ID:Father_id>23</ID:Father_id>
           </rdf:Description>

           <rdf:Description        rdf:about="http://www.Person/ID/45">
                      <ID:name>Yaron</ID:name>
                      <ID:age>49</ID:age>
                      <ID:Father_id>67</ID:Father_id>
           </rdf:Description>

</rdf:RDF>
```

ב. (שליפה ע"פ הטבלה העליונה בחלק א')

```
SELECT ?person WHERE {
?person dbo:parent / dbo:parent ttr:Dani .
}
```

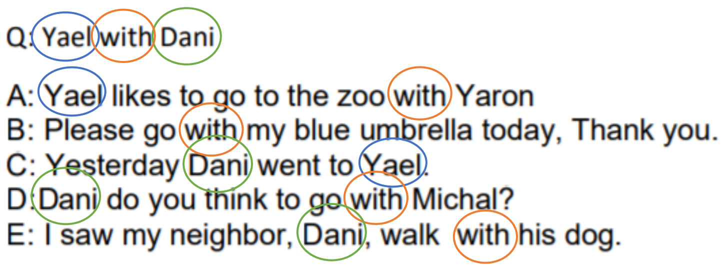
6. (TF-IDF)

חישוב ע"פ נוסחה:

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

A = (1/9)*log(5/2) + (1/9)*log(5/4) = 0.126
B = (1/9)*log(5/4) = 0.024
C = (1/5)*log(5/3) + (1/5)*log(5/2) = 0.285
D = (1/8)*log(5/3) + (1/8)*log(5/4) = 0.091
E = (1/9)*log(5/3) + (1/9)*log(5/4) = 0.081

חישוב ע"פ מערכת שיקולים:


Q: Yael with Dani

A: Yael likes to go to the zoo with Yaron
B: Please go with my blue umbrella today, Thank you.
C: Yesterday Dani went to Yael.
D: Dani do you think to go with Michal?
E: I saw my neighbor, Dani, walk with his dog.

<u>דרך חישוב</u>
נבדוק איזה מילה נמצאת ביותר משפטים:
Yael – 2
With – 4
Dani – 3
ככל שהמספר יותר קטן כך הדירוג יותר גבוהה.
ל-C יש את שני המילים הכי חשובות לכן הוא במקום הראשון, ל-A יש שני מילים מהמשפט ואחת מהם היא המילה הכי חשובה ולכן הוא במקום השני, D ו-E נמצאים בתיקו מבחינת שווי המילים אך מספר המילים ב-D קטן ממספר המילים ב-E ולכן D במקום השלישי ו-E במקום הרביעי, ולבסוף יש את המשפט B במקום האחרון עם מילה אחת שנמצאת בדירוג הכי נמוך.
סה"כ נקבל:

1.	C
2.	A
3.	D
4.	E
5.	B