

**מבחן במסדי נתונים**

**מירב שקרון**

**7028010**

סמסטר קיץ מועד א' כט תשרי התשע"ח, 19.10.2017

**הנחיות כלליות:**

* משך הבחינה: 180 דקות.
* **יש לענות בגוף השאלון!** המחברת תשמש כטיוטא בלבד. על מענה במחברת יורדו נקודות!
* במבחן 11 שאלות, שימו לב שאתם עוברים על כולם
* אין להכניס שום חומר עזר.
* השימוש במחשבון **אסור**.
* בסיום הבחינה - נא למסור את השאלון ואת המחברת.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Total | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |  |
| 100 | 13 | 9 | 9 | 5 | 10 | 7 | 5 | 8 | 13 | 5 | 16 | Max points |
|  |  |  |  |  |  |  |  |  |  |  |  | Grade |

**ב ה צ ל ח ה!**

1. SQL (16 נק')

נתון מבנה הנתונים הנ"ל, של אולימפיאדת מקסיקו 2016:

Participant (ParticipantId, name, gender, yearOfBirth, country) טבלת משתתפים

CompetitionTypes (competitionTypesId, name)  טבלת תחרויות

Prizes (prizeId, name, points)  טבלת סוגי פרסים וניקוד

CompetitionParticipant (competitionTypeId, ParticipantId, dayAsInt, prizeId)

טבלת משתתפים בתחרות והפרס (הפרס יכול להיות null)

ניתן להניח שdayAsInt הוא מספר שלם המייצג את יום המשחק מאז שהחלה האולימפיאדה.

1. כתבו שאילתה שמחזירה את מס' הנשים שזכו בפרסים ואת מס' הגברים שזכו בפרסים ומה סה"כ הניקוד של כל אחד מהם (של הגברים והנשים בנפרד)

**SELECT** Participant.gender, **COUNT(distinct** Participant.participantId**) as** counter, **SUM(**Prizes.points**) as** sum **FROM** Participant  
**JOIN** CompetitionParticipant   
**ON** Participant.ParticipantId = CompetitionParticipant.ParticipantId  
**JOIN** Prizes  
**ON** Prizes.prizeId = CompetitionParticipant.prizeId  
**GROUP BY** Participant.gender

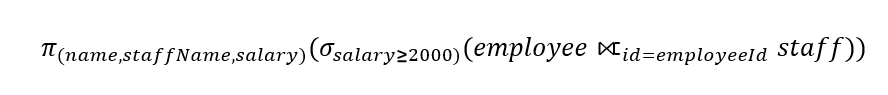
1. כתבו שאילתה שמחזירה עבור כל יום את **גילו** של הצעיר ביותר שלא זכה באף תחרות
2. Select cp.dayAsInt, min(2021- year(p.yearOfBirth)) as youngest\_looser
3. from Participants as p join CompetitionParticipant as cp
4. on(p.ParticipantId = cp.ParticipantId)
5. Where cp.prizeId = NULL
6. group by cp.dayAsInt;

אם רוצים לסנן לפי מי שלא זכה באף תחרות **ספציפית באותו היום**, אפשר לשנות אחרי הwhere ל:

1. Where count(cp.prizeId) = 0

2. Relational algebra (5 נק')

כתבו שאילתת SQL השקולה לביטוי הרלציוני הבא:



Select name, staffName, salary

from employee as e join staff as s

on(e.id = s.employeeId)

where salary>=2000;

3. Normalization (13 נק')

נתונה הרלציה הבאה:

R (U, V, W, X, Y, Z)

נתונות התלויות שלה:

Z🡪W

Y🡪{X, Z}

{W, X}🡪Y

{U, Y, Z}🡪V

1. מצא את כל ה-candidate keys האפשריים (5 נק')

**{U,W,X} , {U,Y} , {U,Z,X}**

1. מהם תכונות ה- prime? (1 נק')

U,W,X,Y,Z

1. מהי רמת הנירמול (NF) של הרלציה הנ"ל (1NF, 2NF, 3NF, BCNF, 4NF).

**נמק** למה הרלציה לא יכולה להתאים ל-NF גבוה יותר (7 נק')

The only non-prime key is V, and it depends on a prime-key (U,Y,Z) then 1,2,3 NF holds.

R violates BCNF since V doesn’t depend on a candidate key

4.XML and XSD (8 נק')

כתבו את **כל** הסיבות מדוע ה- XML הבא, לא בתוקף (not valid) לפי הגדרת ה-XSD להלן:

XML:

<pencilCase>

<pencil>HB</pencil>

<pen>BLUE</pen>

<scissors>AAA</scissors>

<eraser>STAEDTLER</eraser>

<ruler>15 cm</ruler>

<sharpener>KANEX</sharpener>

</pencilCase>

XSD:

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="pencilCase">

<xs:complexType>

<xs:sequence>

<xs:element name="pencil" minOccurs="2" maxOccurs="4"/>

<xs:element name="pen" type="xs:int" />

<xs:element name="eraser" />

<xs:element name="scissors" />

<xs:element name="ruler" />

<xs:element name="colors" maxOccurs="10"/>

<xs:element name="sharpener" />

</xs:sequence>

<xs:attribute name="forWhom" default="students"/>

</xs:complexType>

</xs:element>

</xs:schema>

1. Pencil occurs only once and the minimum is 2.
2. Pen is a string and not int.
3. Eraser comes after scissors.
4. Colors not defined.
5. Note: although forWhom not defined, the default sets it automatically to “students”…

5.NoSQL (5 נק')

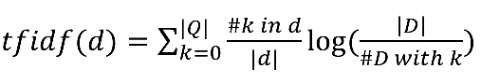
מה ההבדל העיקרי בין בסיס נתונים של Redis לבסיס נתונים של MongoDB?

1. Redis מאחסן נתונים בצורה Key-Value ו-MongoDB מבוסס על Wide column store.
2. ב- Redis ניתן לאחסן רק ערכים פשוטים כמו string, int וכו' בעוד שב-MongoDB ניתן לשמור גם ערכים בצורה של xml או json
3. ב- Redis ניתן לשלוף נתונים לפי ה-key שלהם בעוד שב-mongoDB ניתן לתשאל גם את הנתונים עצמם.
4. MongoDB הרבה יותר מהיר מ-Redis.

6. TF-IDF (7 נק')

**דרגו** את המשפטים הבאים לפי הניקוד של ה- TF-IDF שלהם בהינתן ה-query הבא. במקום הראשון- את התשובה הרלוונטית ביותר ובמקום האחרון את התשובה הכי פחות רלוונטית. נמק את תשובתך.

(אסור להשתמש במחשבון וגם אין צורך בו)

להזכירכם נוסחת ה- TF-IFD היא:

Q: horses and cows

D1: **horses** gallop while **cows** moo

D2: zebras **and** **horses** are similar

D3: Milk all **cows** **and** goats

D4: Jessica **and** Daniel are a couple, married last summer

D5: Ariel **and** Dvora love to ride **horses**

First (**most** relevant): \_\_D1\_\_\_\_

Second: \_\_D3\_\_\_\_

Third: \_\_D2\_\_\_\_

Forth: \_\_D5\_\_\_\_  
Fifth (**least** relevant): \_\_D4\_\_\_\_

7.Cassandra (10 נק')

נתונות 2 הטבלאות הבאות- T1, T2.

T1 מוגדרת כך:

CREATE TABLE T1 (W INT, X INT, Y INT, Z INT, PRIMARY KEY ((W, X), Y, Z));

T2 מוגדרת כך:

CREATE TABLE T2 (W INT, X INT, Y INT, Z INT, PRIMARY KEY (W, X, Y));

סמן עבור כל אחד מהשאילתות ה-CQL האם היא legal, illegal או syntax error

1. SELECT \* FROM T1 WHERE Y<412 AND W=152 AND X=741 \_\_\_legal\_
2. SELECT X FROM T1 WHERE W=174 AND X<890 \_\_\_illegal\_\_\_
3. SELECT X FROM T2 WHERE W=174 AND X<890 \_legal\_\_\_\_
4. SELECT W FROM T2 JOIN T1 ON T2. Z=T1.Z WHERE W=154 \_syntax\_\_\_\_
5. SELECT \* FROM T2 WHERE W>59 \_\_\_illegal\_\_\_

8.Logistic regression (5 נק')

איזה מהבעיות הבאות מתאימה ביותר לפתור בעזרת Logistic regression (לעומת Linear regression ו-Naïve bayes)

1. ניבוי ציוניו של סטודנט לתואר שני על סמך נוכחותו בשיעורים והגשת מטלות בית
2. ניבוי הרווחים של חנות כלי בית על סמך מיקום החנות וגודל האוכלוסייה
3. ניבוי מיקום חנות כלי בית על סמך רווחי החנות וגודל האוכלוסייה
4. ניבוי האם חנות כלי בית תהיה רווחית או הפסדית על סמך מיקום החנות וגודל האוכלוסייה

9.Mongo DB (9 נק')

בהינתן רשימת פרטי תלמידים הבאה:

db.student.insert([

{student\_id:"111", name:"Dani", course\_name:"Databases", grade:42},

{student\_id:"222", name:"Dafna", course\_name:"Operating systems", grade:81},

{student\_id:"333", name:"Miri", course\_name:"Software structure", grade:78},

{student\_id:"444", name:"Nati", course\_name:"Databases", grade:52},

{student\_id:"555", name:"Yaffa", course\_name:"Software structure", grade:62},

{student\_id:"666", name:"Zohar", course\_name:"Operating systems", grade:95}

{student\_id:"777", name:"Ari", course\_name:"Operating systems", grade:48}

{student\_id:"888",name:"Miki",course\_name:"Databases",grade:65})

] )

ופונקציית ה-mapReduce הבאה:

db.student.mapReduce(

function (){

if ( this.grade > 59 ) {

emit(this.course\_name, this.grade);

}

},

function(key, val1) {

var total =0;

for (var i = 0; i < val1.length; i++) {

total += val1[i];

}

return total / val1.length;

},

{out: "my\_out"});

מה תכיל my\_out לאחר הרצת הקוד לעיל (כלומר, מה נקבל כשנקליד **db.my\_out.find()**):

[

{

\_id:”Operating systems”,

Value: 88

},

{

\_id:”Software structure”,

Value: 70

},

{

\_id:”Databases”,

Value: 65

}

]

https://lh6.googleusercontent.com/3XpcoJ4CjAchf3zlaoBtbvtlgsTaT7XRBnua7zwOoQnEJlPjhdKJxuDh7kGJ8BHP0y9nuD_nRYtasoBXciRjU4c_PvwtADrhhDbnh2kJcwvYrmF9c2giMMan6_-FBZauV6pgHM5E01. Linear regression + JavaStreams (9 נק')

יש לממש את פונקציית ה-loss שלמדנו ב-Linear regression

בעזרת JavaStreams.

לצורך כך, הוגדר מבנה הנתונים הבא:

class Data {  
 double x;  
 double y;  
  
 Data(double x, double y){  
 this.x = x;  
 this.y = y;  
 }  
}

הניחו כי w=0.1 ו-b=5

public static double linReg(Data[] dataset){

return Arrays.stream(dataset)

.map(d->Math.pow(5+0.1\*d.x-d.y,2))

.reduce(0.0, (s,v)->s+v)/(2\*dataset.length);

}

11. Linear regression (13 נק')

נזכיר שעבור linear regression הגדרנו את פונקציית ה-loss הבאה:

https://lh6.googleusercontent.com/3XpcoJ4CjAchf3zlaoBtbvtlgsTaT7XRBnua7zwOoQnEJlPjhdKJxuDh7kGJ8BHP0y9nuD_nRYtasoBXciRjU4c_PvwtADrhhDbnh2kJcwvYrmF9c2giMMan6_-FBZauV6pgHM5E

נזכיר שקו הניבוי ב- linear regression הוא: h(x) = xw + b

בהינתן הנתונים הבאים:

X = [2, 4, 0, -2]

Y = [3, 1, 1, -3]

יש להניח שאנו משתמשים ב- Gradient Descent

1. מה ערכי ה-  **gradient** (לפי w ולפי b) כאשר w=0 ו-b=0 ? (יש לתת ערך מספרי) (6 נק')
2. הניחו ש- α=0.1 , מה יהיו הערכים של w ו-b באיטרציה הבאה?

(השתמשו בגרדיאנט שחישבתם בסעיף א') (3 נק')

1. לפי הערכים שמצאתם ל-w ו-b בסעיף הקודם, מה יהיה הניבוי ל-x=2?

מתי הניבוי יותר מדויק- עם w=0, b=0 או עם הערכים שמצאתם בסעיף הקודם?

(יש להראות חישוב) (4 נק')

כמובן שיש שיפור