מס' מחבררת:___________ ת.ז.:_________________

מבחן במערכות בסיסי נתונים

טובה מילוא ועמית סומך

סמסטר ב' מועד א'

22 ביוני, 2016

<u>הוראות (נא לקרוא!)</u>

- משך הבחינה שלוש שעות, חלקו את זמנכם ביעילות.
- אסור השימוש בחומר עזר כלשהו (אין צורך במחשבונים).
- יש לענות על כל השאלות בגוף הבחינה במקום המיועד לכך. יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תיפסל. תשובות במחברת הבחינה לא תיבדקנה. במידת הצורך ניתן לכתוב בגב טופס הבחינה.
- יש למלא מספר סידורי (מס' מחברת) ומספר ת.ז על כל דף של טופס הבחינה.

לשימוש הבודקים בלבד:

חלק	ציון
1	
2	
3	
סה"כ	

חלק 1 [40 נק']

עבור בסיסי הנתונים של תכנית הריאליטי "מטבחי הכס", נתונה הסכמה הבאה:

Chef (<u>id</u> : *INT*, name : *VARCHAR*, specialization: Varchar*, team:VARCHAR,* birthDate*: DATE***)**

Team(<u>name</u> : *VARCHAR*, leader : *INT* **)**

CulinarySpecialization(<u>name</u>: *VARCHAR***)**

Show(<u>number</u> : INT, theme: *VARCHAR*, team: *VARCHAR,* winner: *INT* , Loser : *INT* , showDate: *DATE***)**

בסכמה זו, השדות (attributes) המודגשים בקו תחתון הם מפתחות ראשיים (primary keys). בנוסף, מוגדרים המפתחות הזרים (foreign keys) הבאים:

Chef(specialization) -> CulinarySpecialization(name)

Chef(team)-> Team(name)

Team(leader) -> Chef(id)

Show(winner) -> Chef(id)

Show(Loser) -> Chef(id)

Show(theme) -> CulinarySpecialization(name)

ניתן להניח שהטבלאות אינן מכילות ערכי NULL, אך גם לא מוגדרים NULL CONSTRAINTS על אף אחד מהשדות.

בטבלת Chef נשמרים הנתונים עבור השפים שמשתתפים בתוכנית. כל שף מזוהה על ידי מספר סידורי (id), ובטבלה מצויים ההתמחות הקולינרית של השף, שם הקבוצה ותאריך לידה.

השפים בתוכנית מחולקים לקבוצות (team), כל קבוצה מזוהה על פי שם הקבוצה (name) ויש לה מנהיג (leader) שהוא אחד השפים המשתתפים בתוכנית.

בכל תוכנית (show) , המזוהה על ידי מספר התוכנית (number), יש "נושא" קולינרי (theme). **בכל תוכנית משתתפת רק אחת הקבוצות** (team) ויש לה שף מנצח (winner) ושף מפסיד (loser).

א. [10 נק']

כיתבו שאילתת SQL המחזירה את שם הקבוצה בה ממוצע גיל המשתתפים הוא הנמוך ביותר.
להזכירכם, הפקודה datediff מחשבת את ההפרש בין שני תאריכים, והפקודה DATE(NOW()) מחזירה את התאריך של היום.

תשובה קומפקטית:

```
SELECT c.team, AVG(DATEDIFF(Date(now()),c.birthDate)) as avg_chef_age FROM Chef c
GROUP BY c.team HAVING

avg_chef_age <= ALL (SELECT AVG(DATEDIFF(Date(now()),c2.birthDate)) as avg_chef_age2
FROM Chef c2 GROUP BY c2.team)
```

ב. [10 נק']

i. כתבו שאילתה המחזירה את הid של כל מנהיגי הקבוצות שניצחו לפחות בתכנית אחת (ללא חזרות)

```
SELECT DISTINCT Chef.id FROM Chef , Team , show WHERE Team.leader = Chef.id AND Show.winner=
Chef.id
```

ii. כתבו שאילתה המחזירה את הID של כל מנהיגי הקבוצות שהפסידו לפחות פעם אחת בתכנית שהנושא שלה הוא תחום ההתמחות שלהם (ללא חזרות)

SELECT DISTINCT Chef.id from Chef, Team, Show WHERE Team.leader = Chef.id AND Show.theme = Chef.specialization and Show.loser=Chef.id

iii. כתבו שאילתה המחזירה את ה*שמות* של כל מנהיגי הקבוצות שניצחו לפחות בתכנית אחת, אך מעולם לא הפסידו בתכנית שהנושא שלה הוא תחום ההתמחות שלהם. **הוראה: אין להשתמש בפקודות IN, EXIST, EXCEPT, INTERSECT אלא בפקודות מסוג join בלבד : LEFT JOIN,RIGHT JOIN, FULL OUTER JOIN.** (רמז: ניתן להשוות לערך NULL)

SELECT A.name FROM (SELECT Chef.name, Chef.id FROM Chef, Team, Show WHERE

Team.leader = Chef.id AND Show.winner= Chef.id) as a LEFT JOIN (SELECT Chef.id as id from Chef, Show WHERE Show.theme = Chef.specialization and Show.loser=Chef.id) AS B

ON A.id=B.id AND B.id=NULL

ג. **[6 נק']**
כיתבו את השאילתה של סעיף ב' iii באלגברה רלציונית. ניתן להשתמש בכל האופרטורים שנלמדו בכיתה.

PART I:

$\prod_{c.id} (((Chef\ c \bowtie_{c.id=t.leader} Team\ t)) \bowtie_{c.id= s.winner\ and} Show\ s)$

PART II:

$\prod_{c.id} (Chef\ c \bowtie_{c.id= s.loser\ AND\ c.specialization=s.theme} Show\ s)$

FINAL ANSWER:

$\prod_{c.name} (Chef\ c \bowtie_{c.id= r.id} (<PART\ 1> - <PART\ II>)$

ד. **[14 נק']**
נתון ה-view הבא, מעל אותה הסכמה:

```
CREATE VIEW my_view AS

SELECT Chef.Name as chef_name, Show.theme as show_theme, count(*) as wcount FROM Chef INNER JOIN Show ON Show.winner=Chef.ID GROUP BY chef_name, show_theme
```

i) האם הmy_view הוא בר עדכון (updatable)? הקיפו בעיגול **כן / לא**. אם כן, כתבו לכל סוג עדכון (delete, update, insert) הצהרת sql (סדרת פקודות) המעדכנת את בסיס הנתונים במידע זה, ואם לא הסבירו למה.

לא, משום שקיים מצב בו יש שני שפים בשם "אייל משה", כאשר לכל אחד מהם תחום התמחות שונה ולכן לא נדע איזה ID של chef להכניס לשדה winner בטבלה Show.

ii) כתבו שאילתה המשתמשת **my_view בלבד** (ללא שימוש בטבלאות עצמן), המחשבת עבור כל שם של שף את מספר התכניות שבהן ניצח שף בעל שם זה.

```
SELECT chef_name, SUM(wcount) FROM my_view GROUP BY chef_name
```

חלק 2 [30 נק']

א. [18 נק']

נתונה הטבלה R(A, B, C, D, E,G) עם התלויות הבאות:

A,B —> C

A —> D

C —> E

B —> G

G —> A,D

B —> E

i. מצאו את כל המפתחות המינימאליים של R.

B

ii. בצעו פירוק (lossless) של R ל- BCNF כפי שנלמד בכיתה. התשובה צריכה לכלול את צעדי הגזירה (עם צעדים והתלות שהיוותה הפרה ל-BCNF בכל צעד), ואת הטבלאות שנוצרו עם השדות וכל המפתחות המינימאליים שלהן.

R1 = (B,G,C) , B—> GC , B

R2 = (A,D) , A—>d , A

R3 = (C,E) , C—>E , C

R4 = (G,A), G—>A ,G

* All dependencies break the BCNF beside B —> CG

iii. האם בטבלה המקורית R קיימת תלות שמפרה את BCNF אך לא את 3NF? הקיפו בעיגול כן/ לא והסבירו:

אין, השדה B הוא המפתח היחיד ואינו מופיע בצד ימין באף תלות.

ב. [12 נק']
הפעם, על אותה סכמה R(A,B,C,D,E,G) נתונות התלויות הבאות:

A—>BCD, BC—>DE, B—>D, D—>E

הקיפו בעיגול את התשובה הנכונה וכתבו הסבר קצר. **הערה**: תשובות נכונות ללא הסבר לא יבדקו, וכן סימון תשובה הנכונה עם הסבר שגוי לא יזכה בניקוד.

i. האם A הוא מפתח של R? כן/ לא . הסבר:

לא, AG הוא המפתח המינימאלי כיוון שA וG לא מופיעים מצד ימין באף תלות

ii. האם התלות G,A—>A,B,C,D,G נובעת מהתלויות הנתונות? כן/ לא . הסבר:

כן, ag<-ag זה טריוויאלי, וa->bcd היא תלות נתונה

iii. האם התלות **D,G—>C,B,A,D,G** נובעת מהתלויות הנתונות? כן/ לא . הסבר:

לא. ניתן לראות דוגמה שמפרה את התלות אך מספקת את שאר התלויות. :

A1 B1 C1 D1 E1 G1

A2 B2 C2 D1 E1 G1

חלק 3 [30 נק']

א. [5 נק']

נתונות הטבלאות R(A) ו S(B) , עם מספר דפים B(R) וB(S) ומס' שורות T(R) וT(S) (הניחו כי מספר הדפים בכל טבלה **קטן ממש** ממספר השורות בכל הטבלאות). בנוסף נתון שהערכים בשדה A הם כולם "4" וכן גם הערכים בשדה B כל הערכים הם "4" .

עבור השאילתה $Q=R\bowtie_{A=B} S$, איזה מהאלגוריתמים הבאים הכי יעיל במקרה מבחינת סיבוכיות i/o? **ניתן לסמן יותר מתשובה אחת נכונה**

1. page-oriented nested loop join
2. index nested loop join (unclustered Hash index on B, probing cost is 0 i/o)
3. index nested loop join (clustered B tree index on B, probing cost 2 i/o)
4. hash join with 2 hash functions h1 (partition phase),h2 (matching phase)
5. sort merge join (Both relation can be sorted in 1 pass)
6. לא ניתן לדעת

תשובה 1, כיוון שהסיבוכיות היא B(R) + B(R)·B(S).

בכל המקרים יש לבצע מכפלה קרטזית של לכל הפחות B(R)·B(S) , וכן לבצע חישוב ב"עלות נוספת" לפי הפירוט הבא:

אופציה 2: B(R)+T(R)(B(S) . עלות החיפוש באינדקס היא 0 אך הוא מכיל מצביע לכל הדפים כיוון שיש כיוון שT(R)גדול ממש מB(r) אופציה זו אינה עדיפה

אופציה 3: B(R)+T(R)2(B(S) . עלות החיפוש באינדקס היא 2 אך כיוון שכל הערכים שווים, נאלץ לסרוק את כל הטבלה (s) בכל חיפוש

אופציה 4: 2(B(R)+B(S) + B(R)(B(S) . הפעלת פונקציית h1 לוקחת 2(B(R)+B(S)) אך כל הערכים נמצאים באותו partition. כעת גם שנפעיל את h2 עבור הpartition, היא לא תיכנס בזיכרון כי גודלה לפחות כמו הטבלה R, לכן שלב הmatching יקח לכל הפחות B(R)*B(S)

אופציה 5: B(R)+B(S) + B(R)*B(S)מיון יתבצע בעלות של B(R)+B(S) ושלב הmerge יקח לפחות B(R)*B(S) לפי האלגוריתם merge שנלמד בכיתה.

ב. [5 נק']
מעל הטבלה R(A,B) נתונות שתי השאילתות הבאות:

q1= SELECT * FROM R WHERE A = “FLOW%”

q2= SELECT * FROM R WHERE B = “FLOWER”

שני השדות A וB בעלי אותו גודל והתפלגות הערכים אחידה. הניחו שהאינדקסים מאוכסנים בכללותם בזיכרון

אילו אינדקסים עלינו לבחור כדי לשפר את ביצועי הריצה של <u>**שתי**</u> השאילתות? **ניתן לסמן יותר מתשובה אחת נכונה.**

1. Clustered BTree Index on A , unclustered B tree index on B
2. Clustered Btree index on A, Clustered hash index on B
3. Clustered Btree index on A, Unclustered hash index on B
4. Clustered hash index on A, Clustered hash index on B
5. Clustered hash index on A, Unclustered hash index on B
6. Unclustered hash index on (A,B) - זהו אינדקס על שני השדות ביחד
7. לא ניתן לדעת .

התשובה הנכונה היא 3, ו1: כיוון ששאילתה 1 תוכל להתבצע בb(r) ושאילתה 2 גם היא בb(r) כיוון ששני האינדקסים בזיכרון. לכן אין זה משנה שהאינקס על B הוא hash או Btree.

לגבי שאר האופציות:

2 לא אפשרי כי אי אפשר לקלסטר לפי יותר משדה אחד, 4 ו5 לא יעילים עבור שאילתה 1, כיוון שיש להם hash index ובשאילתה נדרש range. 6 לא משמעותי כיוון שאין שאילתה המשתמשת בשני השדות.

ג. [20 נק']

נתונות שלוש טבלאות $R1(\underline{A}, B)$ $R2(\underline{C}, D)$ $R3(\underline{E}, F, G)$ והשאילתה הבאה:

$$P = (\sigma_{B=0}(R1 \bowtie_{A=C} R2)) \bowtie_{D=F} (\sigma_{G>6}(R3))$$

הערות:

- שימו לב לסדר הפעולות המוכתב ע"י הסוגריים, אשר יהיה קבוע בכל תוכניות הביצוע להלן.
- האינדקסים יושבים בזיכרון בשלמותם, כך שגישה לאינדקס לוקחת 0 פעולות IO.
- השאילתה מחושבת ב- pipeline, כלומר תוצאת אופרטור אחד ניתנת ישירות לאופרטור הבא (בלי להיכתב לדיסק בדרך).
- ניתן להניח שאפשר למיין את כל הטבלאות, כולל תוצאות ביניים, בשני מעברים (phases), כאשר תוצאת המיון נכתבת לדיסק ולא מועברת ישירות לאופרטור הבא.
- בכל מקום בו משתמשים ב- hash join, ניתן להניח שכמות הזיכרון וגדלי ה- buckets מאפשרים את הרצת האלגוריתם כפי שנלמד בכיתה.
- לא נתון מידע על התפלגות ערכי השדות.

נתון המידע הסטטיסטי הבא:

ערך נמוך וגבוה בכל שדה	מס' ערכים בטבלה בעמודה נתונה	מספר דפים בטבלה	מס' רשומות בטבלה	
High(A),Low(A)	V(R1, A) V(R1, B)	B(R1)	T(R1)	R1
	V(R2, C) V(R2, D)	B(R2)	T(R2)	R2
High(G),Low(G)	V(R3, E) V(R3, F) V(R3, G)	B(R3)	T(R3)	R3

בשלושת הסעיפים i-iii הבאים עליכם לחשב את התשובה במונחים של פעולות I/O כפונקציה של הסטטיסטיקות הנתונות, ע"פ הנוסחאות שנלמדו בכיתה. הקפידו על סדר פעולות חשבון וסוגריים בביטוי שתכתבו כדי להבטיח את קריאותו ונכונותו.

<u>לדוגמא</u>: B(R1)B(R2)/V(R1,A).

i. בטאו את הגודל המשוער בדפים של תוצאת הביטוי הבא $(\sigma_{B=0}(R1 \bowtie_{A=C} R2))$

(B(R1) * B(R2) / (MAX(V(R1,A),V(R2,C)) * V(R1,B)))

ii. בטאו את הגודל המשוער בדפים של תוצאת ה- selection על R3 , $\sigma_{G>6}(R3)$.

B(R3) * ((High(G) - 6) / High(G) - Low(G)

iii. עבור השאילתה הנתונה בתחילת השאלה $(P = (\sigma_{B=0}(R1 \bowtie_{A=C} R2)) \bowtie_{D=F} (\sigma_{G>6}(R3)))$ עליכם להעריך את מחיר תוכנית הביצוע במונחים של פעולות I/O, לא כולל את כתיבת התוצאה הסופית לדיסק. ניתן להיעזר בתוצאות שחישבתם בסעיפים i ו- ii. הסבירו כל אחד מצעדי החישוב בקצרה.

הנחות לסעיף זה: מוגדר אינדקס מסוג Btree (Clustered) על C, ואינדקס מסוג hash(unclustered) על A וגם על G. שני ה- join ממומשים כ- sort-merge.

$P = (\sigma_{B=0}(R1 \bowtie_{A=C} R2)) \bowtie_{D=F} (\sigma_{G>6}(R3))$

(1) **$\sigma_{B=0}(R1 \bowtie_{A=C} R2)$:** Sort R1= 4(B(R1) + Sort R2=0 because it's already sorted. Merge = B(R1) + B(R2) ==== >**5B(R1) + B(R2)** Because the selection is "free" because the results of the join operator are in memory already.

(2) **σG>6(R3)**: We have to scan the entire table: ===> **B(R3)**

(3) $\bowtie_{D=F}$: The size of $(\sigma_{B=0}(R1 \bowtie_{A=C} R2))$ is (B(R1) * B(R2) / (MAX(V(R1,A),V(R2,C)) * V(R1,B))), denoted as (i) . we have to sort it, but can read only once so it's **3 *(i)**

Next we sort $\sigma_{G>6}(R3)$, of size B(R3) * ((High(G) - 6) / High(G) - Low(G), denoted by **(ii)**

which is in memory so it's **3(ii)**

adding the merge phase: **(i) + (ii)**

Final result is Merge ====> **4 (B(R1) * B(R2) / (MAX(V(R1,A),V(R2,C)) * V(R1,B))) + 4B(R3) * ((High(G) - 6) / High(G) - Low(G)**

TOTAL (1) + (2) + (3): 5B(R1) + B(R2) + B(R3) + 4 (B(R1) * B(R2) / (MAX(V(R1,A),V(R2,C)) * V(R1,B))) + 4B(R3) * ((High(G) - 6) / High(G) - Low(G)

בהצלחה!