



**הטכניון - מכון טכנולוגי לישראל**  
**הפקולטה למדעי המחשב**

ד"ר אלדר פישר  
סעאב מנסור  
לינה זריבץ'

חורף תשס"ז  
20, מרץ 2007

**מערכות מסדי נתונים – 236363**

**מועד ב' - פתרון**

הזמן: 3 שעות  
במבחן זה 9 עמודים

| שאלה                 | נקודות |
|----------------------|--------|
| שאלה 1 – ERD         | 11     |
| שאלה 2 – שפות שאילתא | 38     |
| שאלה 3 – Design      | 28     |
| שאלה 4 – XML         | 23     |
| סה"כ                 | 100    |

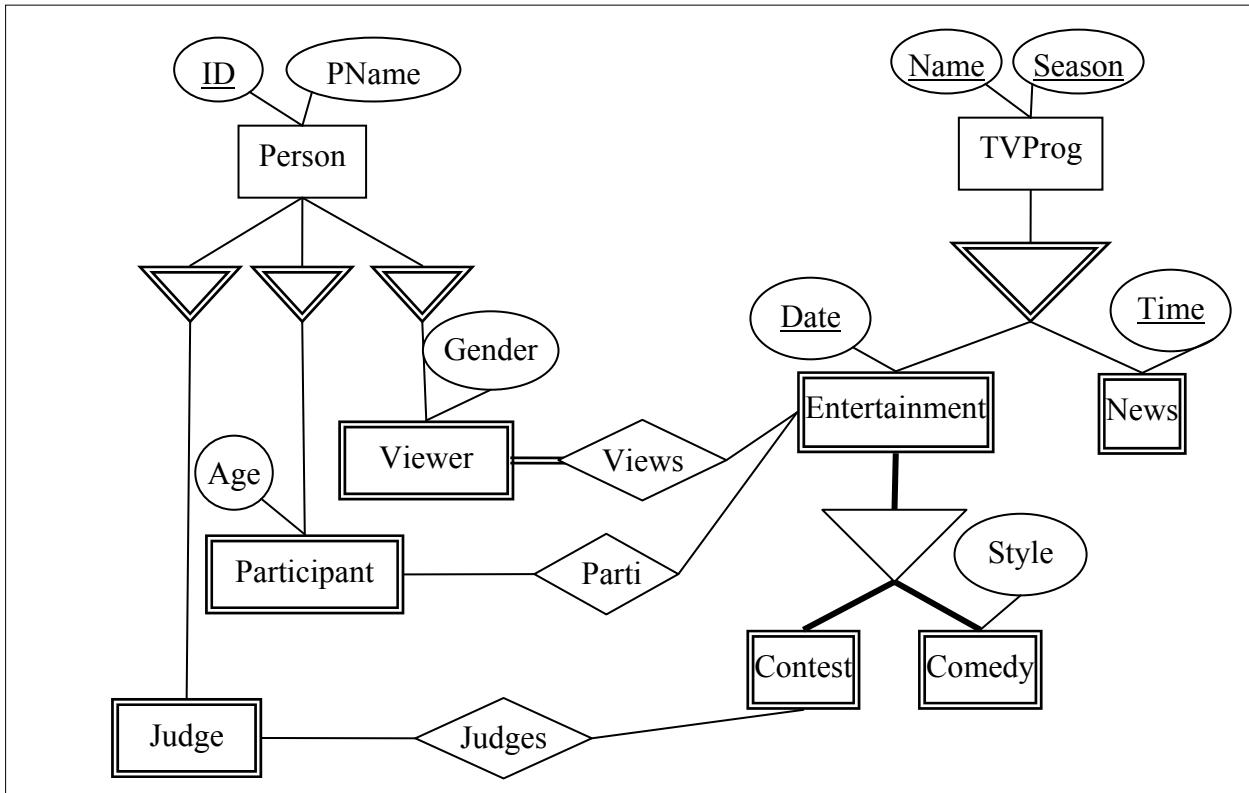
**הנחיות:**

1. יש לענות על כל השאלות **בטופס הבחינה**.
2. חומר עזר מותר: רק דברים שעשויים מנייר.
3. **אין להחזיק מכשיר אלקטרוני כלשהו לרבות מחשב כיס.**
4. קראו היטב את ההוראות שבתחילת כל שאלה ואת ההסברים לסכמות.
5. מומלץ שתתכננו היטב את זמנכם, **לא תינתנה הארכות.**
6. ערעורים יש להגיש תוך שבועיים ממועד פרסום התוצאות.
7. לא יתקבלו ערעורים בנוסח "בדיקה מחמירה מדי".

**בהצלחה**

## שאלה 1 – ERD (11 נק')

נתונה סכמת ה-ERD הבאה שמתארת מסד נתונים של תוכניות טלוויזיה:



### ישראל:

**TVProg** – מייצג תוכניות טלוויזיה. לכל תוכנית יש שם (Name) ועונה (Season) שמייחדים אותה. קיימים שני סוגים של תוכניות:

- **News** – חדשות – שידור של תוכנית חדשות שכוללת את התאריך וזמן השידור בשדה Time.
- **Entertainment** – בידור – מייצג שידור של תוכנית בידור שכולל את תאריך השידור (Date). לתוכניות בידור יש שני סוגים:
  - **Contest** – תחרות – תוכניות תחרותיות.
  - **Comedy** – קומדיה – תוכניות קומדיות שכוללות מידע לגבי הסגנון (Style). סגנון יכול להיות סאטירה, פרודיה וכו'.

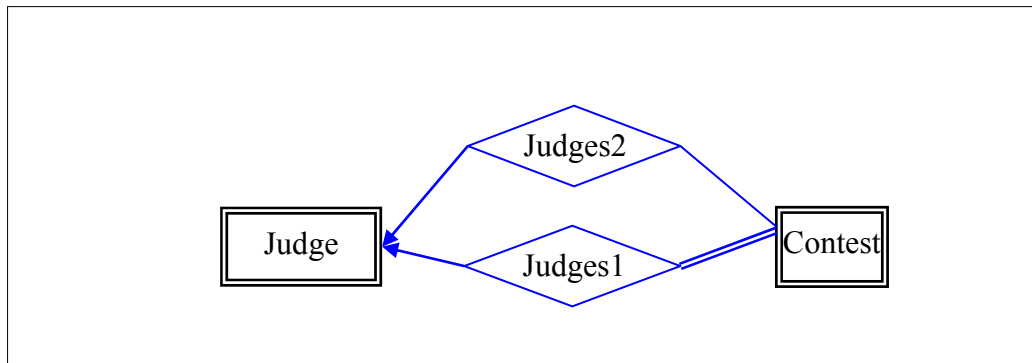
**Person** – מייצג בנאדם במסד. לכל בנאדם קיים מזהה ייחודי (ID) ושם (PName). קיימים מספר סוגים של בני-אדם שונים:

- **Viewer** – צופה – עבור כל צופה נשמר המין שלו (Gender).
- **Participant** – משתתף – מייצג משתתפים בתוכניות טלוויזיה. לכל משתתף נשמר את הגיל (Age) שלו.
- **Judge** – שופט – מייצג שופטים בתחרויות.

### יחסים:

- **Views** – היחס שומר איזה צופים צפו באיזה שידורים של תוכניות בידור.
- **Parti** – היחס שומר באיזה שידורי בידור משתתפים המשתתפים.
- **Judges** – היחס שומר באיזה שידורי תחרויות השופטים שפטו.

א. (5 נק') שנו את הסכמה, כך שלכל תחרות יהיה או שופט אחד או שניים. ציירו את השינוי על החלק המובא להלן. שימו לב שמותר לשנות את מספר הטיפוסים בסכמה, ולא חייבים לאכוף ששני השופטים יהיו שונים.



ב. (6 נק') תארו במילים פשוטות אילו שינויים יש לעשות בסכמה המקורית, כך שה-ERD החדש יאפשר לשופט, בנוסף לשפיטת תחרויות, גם לתת ציון למשתתפי תוכניות בידור. השופט כמובן יכול לתת ציון שונה לאותו משתתף בכל שידור שונה.

נוסיף הקבצה על Parti, ונחבר אותה ל-Judge באמצעות קשר חדש JudgesWS.

נוסיף שדה Score ל-JudgesWS שאינו מפתח.

## שאלה 2 – שפות שאילתא (38 נק')

הערה: בסעיפים הבאים הניחו תרגום סטנדרטי של סכמת ה-ERD משאלה 1 לסכמות רלציוניות: Person, Viewer, Participant, Judge, TVProg, Entertainment, News, Contest, Comedy, Views, Parti, Judges.

א. (6 נק') צרו מבט ב-SQL המכיל עבור כל תחרות ומשתתף בתחרות זו, את מספר השידורים שבהם השתתף במהלך התחרות. סכימת התוצאה תהיה Res(Name,Season,Id,Num).

```

Create view Res as
Select p.name, p.season, p.id, count(p.id) as Num
From Parti P, Contest C
Where P.Name=C.Name and P.Season=C.Season and P.Date=C.Date
Group by p.name, p.season, p.id

```

ב. (6 נק') כתבו שאילתא ב- **SQL** אשר בודקת אם בתחרות "נולד לרקוד" עונה 1 יש מתאם בין גיל להשתתפות: החזירו את כל המתחרים שעבורם קיים משתתף אחר צעיר יותר שהשתתף ביותר שידורים. ניתן להשתמש במבט שיצרתם בסעיף הקודם.

```
Select id
From Res c, participant p
Where name="נולד לרקוד" and season=1 and c.id=p.id
And exists (select id from Res c1, participant p1
            Where c1.name=c.name and c1.season=c.season and
                  c1.id=p1.id and c1.Num>c.Num and p1.Age<p.Age)
```

ג. (7 נק') נניח שנתונה טבלת רייטינג Rating(Name,Season,Date,Num) המכילה לכל שידור של תוכנית את מספר הצופים בה (Num). כתבו במילים פשוטות את משמעות השאילתא הבאה:

$$\{ \langle \text{name}, \text{season} \rangle \mid \text{TVProg}(\text{name}, \text{season}) \wedge \forall \text{date1}, \text{num1}, \text{date2}, \text{num2} \left( \left( \text{Rating}(\text{name}, \text{season}, \text{date1}, \text{num1}) \wedge \text{Rating}(\text{name}, \text{season}, \text{date2}, \text{num2}) \wedge \text{date2} > \text{date1} \right) \rightarrow \text{num2} > \text{num1} \right) \}$$

השאילתא מחזירה את התוכניות (שם ועונה) שהרייטינג שלהן עולה ממש במהלך העונה.

ד. (7 נק') כתבו שאילתא ב-**RA** שמחזירה את התוכניות שהמשתתפים בהן, לא מכסים את כל טווח האפשרויות של גיל. ניתן להניח שגיל המשתתפים האפשרי נע בין 16-36, ולשם כך ניתן להשתמש בטבלה הקבועה  $\text{Age}(n) = \{16, 17, \dots, 36\}$ . יש לכתוב שאילתא שלמה (ללא קיצורים) כך שמספר התווים בה יהיה לכל היותר 100.

$$\pi_{\text{name, season}}(\text{Parti} \bowtie \text{Participant}) \setminus$$

$$\left( \left( \rho_{\text{Age} \rightarrow n} \pi_{\text{Age, name, season}}(\text{Parti} \bowtie \text{Participant}) \right) \div \text{Age} \right)$$

הסעיף הבא הינו בלתי תלוי בסעיפים הקודם.

ה. (12 נק') נתונות תוכניות ה-Datalog הבאות, עם הפרדיקטים המפורשים a, b, c. קבעו האם קיימות עבור כל אחת מהן תוכניות Datalog **שקולות** ולא רקורסיביות. אם כן, רשמו את התוכניות השקולות. אם לא, נמקו.

תוכנית 1:

$q(X, Y) \leftarrow a(X, Y), q(Z, Y).$   
 $q(X, Y) \leftarrow b(X, Y).$

$q(X, Y) \leftarrow a(X, Y), b(Z, Y)$

---

$q(X, Y) \leftarrow b(X, Y)$

---



---



---

תוכנית 2:

$q(X, Y) \leftarrow c(X, Z), q(Z, Y).$   
 $q(X, Y) \leftarrow b(X, Y).$

לא קיימת תוכנית שקולה לא רקורסיבית עבור התוכנית הנתונה.

---

הסיבה: התוכנית מחשבת סגור טרנזיטיבי של הפרדיקט c וכפי שראינו בהרצאות, לא

---

ניתן לבטא תוכניות כאלה ללא רקורסיה.

---



---

### שאלה 3 – Design (28 נק')

א. (12 נק') תהי  $F$  קבוצה כלשהי המכילה תלויות פונקציונאלית עם אטריביוט בודד בצד ימין.

תזכורת:

נגיד שתלות  $X \rightarrow A \in F$  הינה מיותרת אם  $F \setminus \{X \rightarrow A\} \vdash X \rightarrow A$ .

נגיד שאטריביוט  $B \in X$  הינו מיותר בתלות  $X \rightarrow A \in F$  אם  $F \vdash (X \setminus B) \rightarrow A$ .

i. (5 נק') תנו דוגמא ל- $F$  ולתלות  $f \in F$  כך ש- $f$  היא גם תלות מיותרת וגם בעלת אטריביוט מיותר.

$F = \{A \rightarrow B, B \rightarrow D, D \rightarrow C, AB \rightarrow C\}, f = AB \rightarrow C$

אפשר אף דוגמא פשוטה יותר:  $F = \{AB \rightarrow C, B \rightarrow C\}, f = AB \rightarrow C$

ii. (7 נק') תהי  $f = X \rightarrow A$  תלות שאינה מיותרת אך יש בה אטריביוט מיותר. כמו כן יהי  $B \in X$  אטריביוט מיותר כלשהו.  
הוכיחו כי בכל סדרת הוכחה של  $F \vdash (X \setminus \{B\}) \rightarrow A$  משתמשים בתלות  $X \rightarrow A$ .

הוכחה:

נניח בשלילה שקיימת סדרת הוכחה שאינה משתמשת בתלות  $X \rightarrow A$ .

כלומר מתקיים ש-  $F \setminus \{X \rightarrow A\} \vdash (X \setminus \{B\}) \rightarrow A$ . בגלל נאותות ושלמות

של אקסיומות ארמסטרונג נוכל להסיק כי  $F \setminus \{X \rightarrow A\} \vdash X \rightarrow A$ .

דרך יותר מפורטת: נוכל לבנות סדרת הוכחה ל-  $F \setminus \{X \rightarrow A\} \vdash X \rightarrow A$  ע"י

הוספת השורות הבאות לסוף ההוכחה הקיימת:

(k-1)  $(X \setminus \{B\}) \rightarrow A$  (last line of proof)

(k)  $X \rightarrow AB, A2$

(k+1)  $AB \rightarrow A, A1$

(k+2)  $X \rightarrow A, (k), (k+1), A3$

לפי כך נוכל להסיק כי  $f$  תלות מיותרת בסתירה לנתון.

הסעיף הבא הינו בלתי תלוי בסעיף הקודם.

ב. (16 נק') נתונה הסכמה  $R(A, B, C, D, E)$  וקבוצת תלויות פונקציונאליות  $F = \{ B \rightarrow C, B \rightarrow D, DE \rightarrow A, C \rightarrow E, A \rightarrow B \}$

i. (5 נק') רשמו את כל המפתחות הקבילים של  $R, F$ .

---

$A, B, DE, CD$

---

ii. (5 נק') בצעו שינוי אחד ב- $F$  כך ש- $R, F$  תהיה ב-BCNF. שינויים אפשריים הינם מחיקת תלות או הוספת תלות חדשה.

---

נוסיף תלות חדשה  $C \rightarrow A$ .

---

iii. (6 נק') מצאו פירוק משמר מידע ותלויות ל-BCNF המכיל מספר קטן ככל האפשר של תתי-סכמות.

---

$BCDA, DEA, CE$

---

## שאלה 4 – XML (23 נק')

א. (13 נק') נתונים הקבצים הבאים:

קובץ progs.xml המציית ל-DTD הבא עם צומת מסמך list:

```
<!ELEMENT list (prog*)>
<!ELEMENT prog (name, host*)>
<!ELEMENT host (name)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST prog id ID #REQUIRED
              type CDATA #IMPLIED>
<!ATTLIST host experience CDATA "unknown">
```

הקובץ מתאר תוכניות (prog) בעלי שם (name) ורשימת מנחים (host).

קובץ channels.xml המציית ל-DTD הבא עם צומת מסמך list:

```
<!ELEMENT list (channel*)>
<!ELEMENT channel EMPTY>
<!ATTLIST channel num CDATA #REQUIRED
                 progs CDATA #REQUIRED>
```

הקובץ מתאר ערוצים (channel) ואת שידורי התוכניות. לכל ערוץ קיימת התכונה progs אשר אמורה להכיל רשימה של מזהי תוכניות מופרדים ברווחים.

i. (5 נק') מדוע ב-DTD של channels.xml השתמשנו בטיפוס CDATA עבור התכונה progs במקום ב-IDREFS?

---

כי אין תכונה שהיא ID ב-DTD של channels.xml

---

יותר בפירוט: התכונה progs מתייחסת לתכונות ID שכתובות בקובץ אחר, בעוד

---

ש-IDREFS יכול להתייחס רק לתכונות ID באותו קובץ.

---

ii. (8 נק') כתוב שאילתת XQuery 1.0 אשר תייצר קובץ שלישי המשלב את שני הקבצים. ה-DTD שהוא מציית לו זהה לזה של progs.xml, פרט לכך שהשורה השניה שם מוחלפת בשלוש השורות הבאות:

```
<!ELEMENT prog (name, host*, channel*)>
<!ELEMENT channel EMPTY>
<!ATTLIST channel num CDATA #REQUIRED>
```



```
xquery version "1.0";
<list>{
  for $p in doc("progs.xml")//prog
  let $cs:= (for $c in doc("channels.xml")//channel
             where contains($c/@progs, $p/@id)
             return $c)
  return
  <prog>{$p/@id,$p/@type,$p/name,$p/host}
        {for $c in $cs return <channel>{$c/@num}</channel>}
}
</list>
```

הערה: בתשובה כאן הנחנו כי אף אחד ממזחי התוכניות אינו מוכל במזהה אחר כתת-מחרוזת  
ללא הנחה זו אפשר להחליף למעלה את הביטוי "contains(\$c/@progs,\$p/@id)" בביטוי  
"\$p/@id=id(\$c/@progs,doc("progs.xml"))/@id".

ב. (10 נק') עבור כך אחד מזוגות השאילתות הבאות ב-XPath 1.0, כתוב אם הן  
שקולות או לא. נמק בכל אחד מהמקרים, ובפרט כתוב את ההבדל ביניהן אם הן לא  
שקולות. בשני המקרים מותר להניח שהן מבוצעות על הקובץ progs.xml מהשאלה  
הקודמת.

i. (5 נק') השאילתה `/list/prog[host][2]` והשאילתה `/list/prog[2][host]`

השאילתות לא שקולות. למשל מסד שמכיל שני progs והשני בהם מכיל host.

השאילתא הראשונה תחזיר תוצאה ריקה, והשאילתא השנייה תחזיר את ה-prog

השני. זאת מכיוון שהשאילתה הראשונה בוחרת את התוכנית השניה מבין אלו

שיש להן מנחים, והשניה פשוט בודקת אם התוכנית השניה בכלל היא עם מנחים.

ii. (5 נק') השאילתה `/list/prog[host[last()<2]]` והשאילתה  
`/list/prog[count(host)<2]`

השאילתות לא שקולות. השאילתא הראשונה מחזירה כל prog עם host אחד בדיוק,

זאת מכיוון שבמקרה של אפס מנחים תת השאילתה `host[...]` תחזיר קבוצה ריקה.

לעומת זאת, השאילתא השנייה מחזירה כל prog עם אפס או אחד host.