

סמסטר חורף התשע"ג

מרצח : פרופ' יוהן מקובסקי

מתרגלים : מר עומר כ"ץ  
מר דניאל גרפונקל**מערכות מסדי נתונים  
236363**

מועד א' (י' באדר התשע"ג, 20 בפברואר 2013)

**מס' סטודנט:****פירוט השאלות והניקוד:**

מס'	נושא	ניקוד
1	ERD	28
2	שאליות מידע	30
3	פירוקים ותלויות פונקציונליות	30
4	XML	16
סה"כ		104

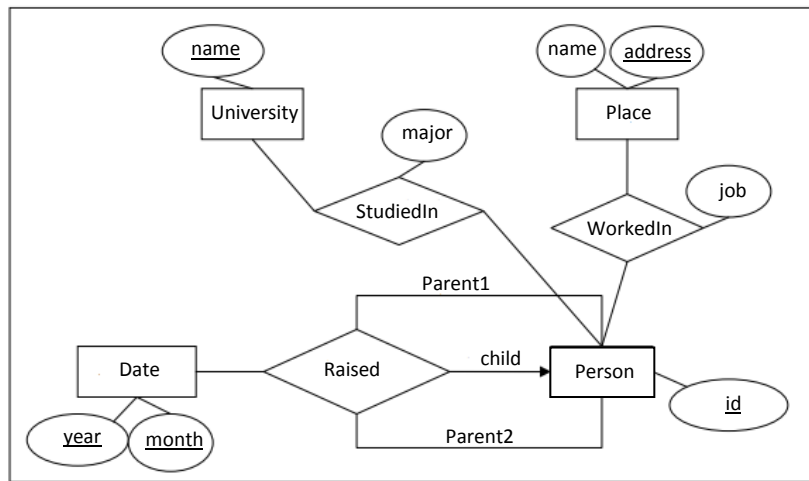
**הנחיות לנבחנים**

1. כתבו את התשובות אך ורק בטופס הבחינה, המחברת מיועדת לטייטה בלבד.
2. מותר ומומלץ לכתוב את התשובות בעפרון.
3. בדף האחרון יש מקום נוסף לתשובות. אם צריך מקום נוסף לתשובות, השתמשו במקום זה תוך ציון הדבר ליד השאלה המקורית.
4. כל חומר עזר כתוב על נייר מותר בשימוש.
5. אין לקבל או להעביר חומר עזר כלשהו בזמן הבחינה.
6. בבחינה ארבע שאלות ללא בחירה. יש לענות עליהן במלואן.
7. בכל מקום שלא נאמר אחרת, יש לנמק את התשובות בקצרה. **תשובות לא מנומקות לא תתקבלנה , למעט במקומות שבהם אתם מתבקשים לכתוב שאלתה.**
8. יש להשתמש רק בסימנים או פונקציות שנלמדו בתרגול או בהרצאה או שמופיעות בשקפים של הקורס. כל שימוש בסימון שאינו כזה מחייב הסבר מלא של משמעות הסימון.
9. משך הבחינה שלוש שעות. תכננו את הזמן בהתאם. **לא תינתנה הארכות זמן במהלך המבחן.**
10. הבחינה כוללת 9 דפים (כולל דף זה), **בהם 17 עמודים**. נא לוודא שיש בידכם את כל הטופס.
11. כאשר ניקוד תתי הסעיפים אינו מצוין, ניקוד הסעיף מתחלק שווה ביניהם.
12. הניקוד אינו נועד לשקף את קושי השאלה ולכן מומלץ לקרוא קודם את כל השאלות.

**בהצלחה**

## שאלה 1 – ERD (28 נק')

נתונה דיאגרמת ה-ERD הבאה :



דיאגרמת ה-ERD מתארת אנשים (Person) המגדלים ילד (אדם נוסף) החל מחודש מסויים (Date).

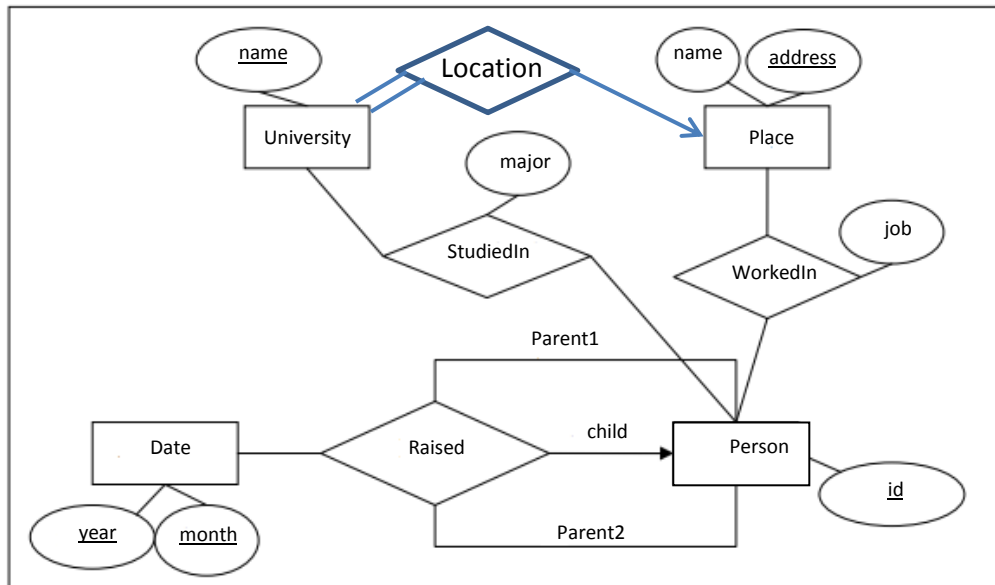
**שאלות:**

א. (10 נק') השלימו את הטבלאות הבאות המתקבלות מתרגום ישיר של ה-ERD.

שם הטבלה : StudiedIn
שדות : name, id, major
מפתחות : name, id
תלויות פונקציונליות : name, id -> major

שם הטבלה : Raised
שדות : Parent1, Parent2, year, month, child
מפתחות : Parent1, Parent2, year, month
תלויות פונקציונליות : Parent1, Parent2, year, month -> child

ב. 8 נק') שנו את הדיאגרמה המקורית כך שלכל אוניברסיטה יהיה מיקום (ללא הוספת אטריביוטים חדשים).



ג. ענו על השאלות הבאות על פי דיאגרמת ה-ERD המקורית. חובה לנמק (בקצרה)  
 i. 5 נק') האם הורים יכולים לגדל שלישיה שנולדה באותו יום?

לא.  
 לפי היחס Raised ניתן לראות שמה שמגדיר ילד זה הצירוף של הורה 1, הורה 2, שנה וחודש. מכיוון שהשלישיה נולדה באותו יום יש להם אותם ערכים לשנה וחודש. מכאן שההבדל האפשרי היחיד ברשומות יהיה בהורה 1 והורה 2. מכיוון שאנו מעוניינים באותו זוג הורים ניתן לכל היותר רק להחליף את הסדר ביניהם, כלומר יתכנו 2 אפשרויות ולכן לא ניתן לגדל שלישיה.

ii. 5 נק') כמה הורים יכולים לגדל ילד נמקו?

אין הגבלה למספר הרשומות בהן יכול להופיע ילד ספציפי ולכן אין הגבלה למספר ההורים שיכולים להיות לילד.

## שאלה 2 – שאילות מידע (30 נק')

השאלות בסעיף זה מתייחסות לדיאגרמת ה-ERD שניתנה בשאלה 1 ולטבלאות שמהוות תרגום ישיר ממנה.

א. (6 נק') תרגמו את השאילתה הבאה לשפה טבעית:

SELECT Parent1, Parent2 FROM Raised

GROUP BY Parent1, Parent2

HAVING COUNT(\*)>=3

השאילתה מחזירה את כל זוגות ההורים שגידלו ביחד לפחות 3 פעמים (יתכן שגידלו את אותו ילד מספר פעמים)

ב. (6 נק') תרגמו את השאילתה מסעיף א' (לא את תשובתכם לסעיף א') לDRC (ציינו את סדר המשתנים של הטבלאות בהם השתמשתם):

$$\{ \langle Parent1, Parent2 \rangle \mid \exists child1, child2, child3, y1, y2, y3, m1, m2, m3 \\ \wedge Raised(Parent1, Parent2, child1, y1, m1) \\ \wedge Raised(Parent1, Parent2, child2, y2, m2) \\ \wedge Raised(Parent1, Parent2, child3, y3, m3) \\ \wedge (y1 \neq y2 \vee m1 \neq m2) \\ \wedge (y1 \neq y3 \vee m1 \neq m3) \\ \wedge (y3 \neq y2 \vee m3 \neq m2) \}$$

ג. 6 נק') כתבו שאילתת SQL המחזירה את תאריך הלידה (תאריך הלידה הוא החודש שמוגדר בטבלה Date, הנח שיש תאריך אחד כזה עבור כל אדם) של האנשים (child) אשר למדו את אותו major בלפחות שלוש אוניברסיטאות שונות וגם עבדו בלפחות חמש עבודות שונות באותו מקום (כתובת).

```
SELECT month,year FROM Raised
WHERE child IN
(
    (SELECT id FROM StudiedIn
    GROUP BY id, major
    HAVING COUNT(*)>=3)
    INTERSECT
    (SELECT id FROM WorkedIn
    GROUP BY id, address
    HAVING COUNT(*)>=5)
);
```

ד. (6 נק') נניח שמסע בזמן הוא אפשרי. נגדיר כלופר (looper) אדם שחזר בזמן וגידל את אחד מאבותיו (ancestor) או את עצמו. כתבו תכנית Datalog שמוצאת את כל האנשים שאינם לופרים.

```

ancestor(p1, p2) ← Raised(p1, _, _, p2)
ancestor(p1, p2) ← Raised(_, p1, _, p2)
ancestor(p1, p3) ← ancestor(p1, p2), ancestor(p2, p3)
looper(p) ← ancestor(p, p)
notLooper(p) ← ¬looper(p), Person(p)

```

ה. (6 נק') בהינתן מסד הנתונים הבא :

*Worker(workerID, workerName)*

*Manager(managerID, managerName)*

*WorkedUnder(workerID, managerID)*

רשמו שאילתה ב-RA המחזירה את כל העובדים אשר עבדו תחת מנהל אשר ניהל את כל העובדים.

$Worker \triangleright < (WorkedUnder \triangleright < (WorkedUnder \div \pi_{workerID}(Worker)))$

הערה: השאילתה תחזיר את כל העובדים עם קיים מנהל כזה שעבד עם כולם, או קבוצה ריקה אם לא קיים מנהל שעבד עם כולם. לכן, הפתרון Workers לא נכון.

### שאלה 3 – פירוקים ותלויות פונקציונליות (30 נק')

תזכורת:

1. בהינתן סכמה  $R$  וקבוצת תלויות  $F$ , פירוק של  $R$  לתת-סכמות  $R_1$  ו  $R_2$  הוא משמר מידע אם ורק אם מתקיים  $R_1 \cap R_2 \rightarrow R_1$  או  $R_1 \cap R_2 \rightarrow R_2$ .
2. בהינתן סכמה  $R$  קבוצת תלויות  $F$  ופירוק של  $R$  לתת-סכמות  $R_1, \dots, R_n$ , אם מתקיים  $(\bigcup_{i=1}^n \pi_{R_i} F)^+ = F^+$  וקיים  $i$  כלשהו עבורו מתקיים ש  $R_i$  הוא מפתח-על של  $R$  אזי הפירוק הוא משמר מידע.

א. 6 נק') נתונה הטענה הבאה: בהינתן התלויות  $x \rightarrow y, z \rightarrow w$  גם התלות  $xz \rightarrow wy$ . הוכח (באמצעות כללי ההיסק ואקסיומות ארמסטרונג שנלמדו) או הפרך.

$x \rightarrow y$	נתון
$\Downarrow$	
$xz \rightarrow zy$	הכללה
$\Downarrow$	
$z \rightarrow w$	נתון
$\Downarrow$	
$zy \rightarrow wy$	הכללה
$\Downarrow$	
$xz \rightarrow wy$	טרנזיטיביות

ב. 6 נק') נתונות התלויות הפונקציונליות  $x \rightarrow y, wy \rightarrow z$ .

i. מצא  $u$  עבורו לא מתקיימת התלות  $xu \rightarrow z$ .

עבור  $x = u$  נקבל  $xu = x$  ומכיון שלא ניתן להסיק מהנתונים  $x \rightarrow z$  אז גם לא ניתן להסיק  $xu \rightarrow z$ .

כל פתרון כך ש  $u \notin z, z \notin u$  התקבל כפתרון נכון.

ii. מצא  $u$  עבורו מתקיימת התלות  $xu \rightarrow z$ .

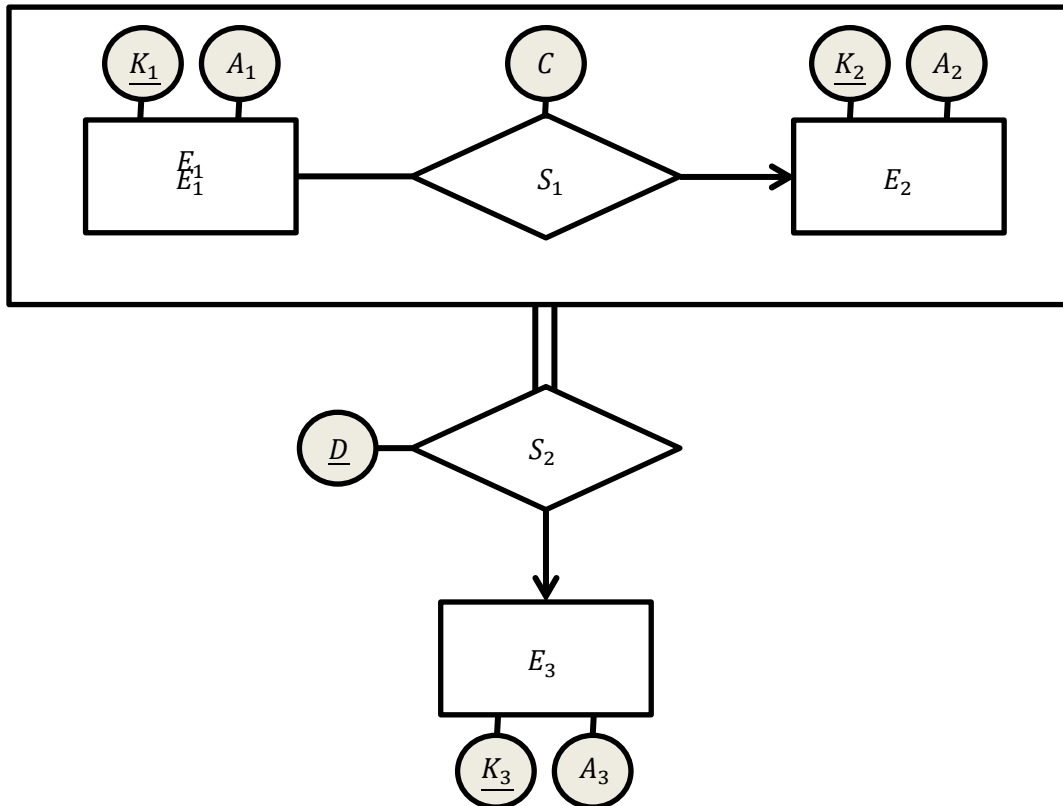
עבור  $u = w$  נקבל  $u \rightarrow w$  ולפי סעיף א' והנתון  $x \rightarrow y$  נקבל  $xu \rightarrow wy$  ומטרנזיטיביות עם  $wy \rightarrow z$  נקבל  $xu \rightarrow z$ .

עבור  $u = z$  נקבל  $xz \rightarrow z$  וזאת תלות טריוויאלית.

ג. נתונה הסכמה  $R[K_1A_1K_2A_2K_3A_3CD]$  ונתונה קבוצת תלויות פונקציונליות

$$F = \left\{ \begin{array}{l} K_1 \rightarrow A_1 \\ K_2 \rightarrow A_2 \\ K_3 \rightarrow A_3 \\ K_1 \rightarrow CK_2 \\ K_1D \rightarrow K_3 \end{array} \right\}$$

המתאימים לדיאגרמת ה-ERD הבאה :





i. (9 נק') נתון הפירוק  $P_1 = \{E_1[K_1A_1], E_2[K_2A_2], E_3[K_3A_3], S_1[K_1K_2C], S_2[K_1K_3D]\}$  הנובע ישירות מהדיאגרמה.

a. הוכח שהפירוק  $P_1$  הוא משמר מידע.

לפי התלויות הנתונות ניתן לרשום שמתקיים  $K_1D \rightarrow R$ , כלומר  $S_2$  היא מפתח על של  $R$ .  
 לפי סעיף ב' אנו רואים שהפירוק הוא משמר תלויות, כלומר מתקיים  $(\cup_{i=1}^n \pi_{R_i} F)^+ = F^+$ .  
 לפי הסעיף השני של התזכורת בעמוד 7 אנו מקבלים שהפירוק הוא משמר מידע

b. הוכח שהפירוק  $P_1$  הוא משמר תלויות.

התלות  $K_1 \rightarrow A_1$  נשמרת כולה בסכמה  $E_1$ .  
 התלות  $K_2 \rightarrow A_2$  נשמרת כולה בסכמה  $E_2$ .  
 התלות  $K_3 \rightarrow A_3$  נשמרת כולה בסכמה  $E_3$ .  
 התלות  $K_1 \rightarrow CK_2$  נשמרת כולה בסכמה  $S_1$ .  
 התלות  $K_1D \rightarrow K_3$  נשמרת כולה בסכמה  $S_2$ .  
 כל התלויות המקוריות נשמרות בתת הסכמות החדשות ולכן בפרט גם הסגור שלהן נשמר.

c. הוכח שהפירוק  $P_1$  הוא בצורה הנורמלית BCNF.

ההיטל של F על  $E_1$  הוא התלות  $K_1 \rightarrow A_1$ .  $K_1$  הוא מפתח על של  $E_1$  ולכן  $E_1$  היא ב-BCNF.

ההיטל של F על  $E_2$  הוא התלות  $K_2 \rightarrow A_2$ .  $K_2$  הוא מפתח על של  $E_2$  ולכן  $E_2$  היא ב-BCNF.

ההיטל של F על  $E_3$  הוא התלות  $K_3 \rightarrow A_3$ .  $K_3$  הוא מפתח על של  $E_3$  ולכן  $E_3$  היא ב-BCNF.

ההיטל של F על  $S_1$  הוא התלות  $K_1 \rightarrow CK_2$ .  $K_1$  הוא מפתח על של  $S_1$  ולכן  $S_1$  היא ב-BCNF.

ההיטל של F על  $S_2$  הוא התלות  $K_1D \rightarrow K_3$ .  $K_1D$  הוא מפתח על של  $S_2$  ולכן  $S_2$  היא ב-BCNF.

כל תת הסכמות החדשות הן ב-BCNF ולכן הפירוק כולו הוא ב-BCNF.

ii. 9 נק') נתון פירוק נוסף עבור  $R$ :  $P_2 = \{E_1[K_1A_1], E_2[K_2A_2], E_3[K_3A_3], S'[K_1K_2K_3CD]\}$ .

a. האם הפירוק  $P_2$  הוא משמר מידע? הוכח או הפרך.

לפי התלויות הנתונות ניתן לרשום שמתקיים  $K_1D \rightarrow R$ , כלומר  $S'$  היא מפתח על של  $R$ .

לפי סעיף ב' אנו רואים שהפירוק הוא משמר תלויות, כלומר מתקיים  $(\cup_{i=1}^n \pi_{R_i} F)^+ = F^+$ .

לפי הסעיף השני של התזכורת בעמוד 7 אנו מקבלים שהפירוק הוא משמר מידע

b. האם הפירוק  $P_2$  הוא משמר תלויות? הוכח או הפרך.

התלות  $K_1 \rightarrow A_1$  נשמרת כולה בסכמה  $E_1$ .

התלות  $K_2 \rightarrow A_2$  נשמרת כולה בסכמה  $E_2$ .

התלות  $K_3 \rightarrow A_3$  נשמרת כולה בסכמה  $E_3$ .

התלות  $K_1 \rightarrow CK_2$  נשמרת כולה בסכמה  $S'$ .

התלות  $K_1D \rightarrow K_3$  נשמרת כולה בסכמה  $S'$ .

כל התלויות המקוריות נשמרות בתת הסכמות החדשות ולכן בפרט גם הסגור שלהן נשמר.

c. האם הפירוק  $P_2$  הוא בצורה הנורמלית BCNF? הוכח או הפרך.

ההיטל של F על  $S'$  הוא התלויות  $K_1 \rightarrow CK_2$  ו  $K_1D \rightarrow K_3$ .  $K_1$  אינו מפתח על של  $S'$  ולכן  $S'$  היא אינה ב-BCNF.

קיימת תת סכמה שאינה ב-BCNF ולכן הפירוק כולו אינו ב-BCNF.

## שאלה 4 – XML (14 נק')

נתון DTD עבור מסד הנתונים של פיצריה:

```
<!ELEMENT orders order*>
<! ELEMENT order (pizza+,sidedish*)>
<!ATTLIST order
    datetime CDATA #REQUIRED
    client CDATA #REQUIRED
    delivery (no | yes) "no"
>
<! ELEMENT pizza topping+>
<! ELEMENT topping topping*>
<! ATTLIST topping
    numOfSlices CDATA #REQUIRED
    name ID #REQUIRED
>
<! ELEMENT sidedish EMPTY>
<! ATTLIST sidedish
    price CDATA #REQUIRED
    name CDATA #REQUIRED
>
```

### שאלות:

א. ענו על השאלות הבאות לגבי ה-DTD ונמקו בקצרה:

i. (2 נק') האם חובה להזמין מנת צד (sidedish) והאם ניתן להזמין רק מנת צד (sidedish)?

מכיוון שההגדרה של order היא `<! ELEMENT order (pizza+,sidedish*)>` ניתן לראות שמנות הצד (sidedish) מסומנות באמצעות \* ולכן יתכן שלא יהיו בכלל מנות צד בהזמנה, כלומר לא חובה להזמין מנת צד.

לעומת זאת, הפיצות מסומנות באמצעות + ולכן חייבת להיות לפחות פיצה אחת בהזמנה ומכאן שלא ניתן להזמין רק מנת צד, בלי פיצה.

- ii. 2 נק') האם יתכן ששני לקוחות יזמינו את אותה מנת צד (sidedish) במחירים שונים. אם כן, הסבר מדוע וכיצד ניתן לתקן זאת? אם לא, הסבר מדוע וכיצד ניתן לתקן זאת?

יתכנו שני אלמנטים מסוג מנת צד בעלי אותו שם ומחירים שונים. הסיבה לכך היא שלכל אלמנט יש מחיר משלו ואין מניעה שיהיו מספר אלמנטים שמייצגים את אותה מנת צד (כלומר בעלי אותו שם).

```
<ELEMENT orders (order*,sidedish*)>
<! ELEMENT order (pizza+)>
<ATTLIST order
    datetime CDATA #REQUIRED
    client CDATA #REQUIRED
    delivery (no | yes) "no"
    sidedishes IDREFS #IMPLIED
>
<ATTLIST sidedish
    price CDATA #REQUIRED
    name ID #REQUIRED
>
```

על מנת לתקן זאת נעשה את השינויים הבאים:

(השינויים באדום)

(הגדרות שלא מופיעות כאן לא התשנו)

- iii. 1 נק') ללא השינוי בסעיף ii, האם ניתן להזמין פיצה עם תוספת כפולה כלשהי (שימו לב, בהנחה שקיימת תוספת מסוג X, תוספת X כפולה תוגדר כתוספת X על גבי תוספת X)?

מכיוון שהשם של תוספת הוא ID שלה, לא יתכן שהמסמך יכיל פעמיים את אותה תוספת ולכן בפרט לא יתכן שתהיה תוספת כפולה.

- ב. 5 נק') כתבו שאילתת XPath1.0 המחזירה את כל ההזמנות שכוללות מגשים לא מלאים, כלומר מספר המשולשים שהוזמנו אינו כפולה של 8 (לצורך ספירת המשולשים יש להתייחס רק לתוספות הראשיות, כלומר לא לספור תוספת על גבי תוספת). הניחו שהפעולות האריתמטיות מוגדרות בשפה.

```
//order[pizza[sum(number(topping/@numOfSlices)) % 8 != 0]]
```

- ג. ענו על השאלה הבאה על פי ה DTD המקורי שניתן בתחילת השאלה ללא השינויים מסעיף א'.  
 i. (4 נק') כתבו תוכנית XQuery המקבלת מסמך pizza.xml התקף עבור ה-DTD המקורי הנתון בתחילת השאלה ומדפיסה מסמך המפרט לכל שם של מנת צד (sidedish) בכמה הזמנות הוזמנה מנת צד עם שם זה ומדפיסה את פרטי ההזמנות הללו. אלמנט השורש של המסמך החדש יקרא sidedishes.

```
<sidedishes>
  {
    for $dish in distinct-values(doc("books.xml")//sidedish/@name)
    let $orders := (
      for $o in doc("reviews.xml")//order
      where ($o/sidedish/@name = $dish)
      return $o
    )
    return
      <sidedish count={count($orders)}>
      {
        for $order in $orders
        return
          <order
            datetime={$order/@datetime}
            client={$order/@client}
            delivery={$order/@delivery}
          />
      }
    </sidedish>
  }
</sidedishes>
```

שימו לב שלא נדרש להדפיס גם את שם המנה.  
 כמו כן, פרטי ההזמנה הנדרשים הם התכונות שלה, לא הבנים שלה.

- ii. (2 נק') כתבו DTD המתאים לפלט התוכנית שכתבתם בסעיף הקודם.

```
<!ELEMENT sidedishes (sidedish*)>
<!ELEMENT sidedish (order+)>
<!--ATTLIST sidedish
      count CDATA #REQUIRED
-->
<!ELEMENT order EMPTY>
<!--ATTLIST order
      datetime CDATA #REQUIRED
      client CDATA #REQUIRED
      delivery (no | yes) "no"
-->
```

### מקום נוסף לתשובות

אם אתם משתמשים בדף זה, ציינו זאת ליד השאלה/השאלות המקוריות, וציינו כאן את מספר/י השאלה/השאלות.

שאלה: \_\_\_\_\_ סעיף: \_\_\_\_\_

שאלה: \_\_\_\_\_ סעיף: \_\_\_\_\_



שאלה: \_\_\_\_\_  
סעיף: \_\_\_\_\_

