



ד"ר אלדר פישר
סעאב מנסור
לינה זריבץ'

אביב תשס"ה
22 בספטמבר 2005

מערכות מסדי נתונים

מועד ב'

הזמן: 3 שעות
במבחן זה 11 עמודים

שם פרטי: _____

שם משפחה: _____

מס' סטודנט: _____

פקולטה: _____

שאלה	נקודות	מתוך
שאלה 1 – ERD + ODL		24
שאלה 2 – Embedded SQL + DATALOG		19
שאלה 3 – RA + RC		18
שאלה 4 – Design		23
שאלה 5 – XML		16
סה"כ		100

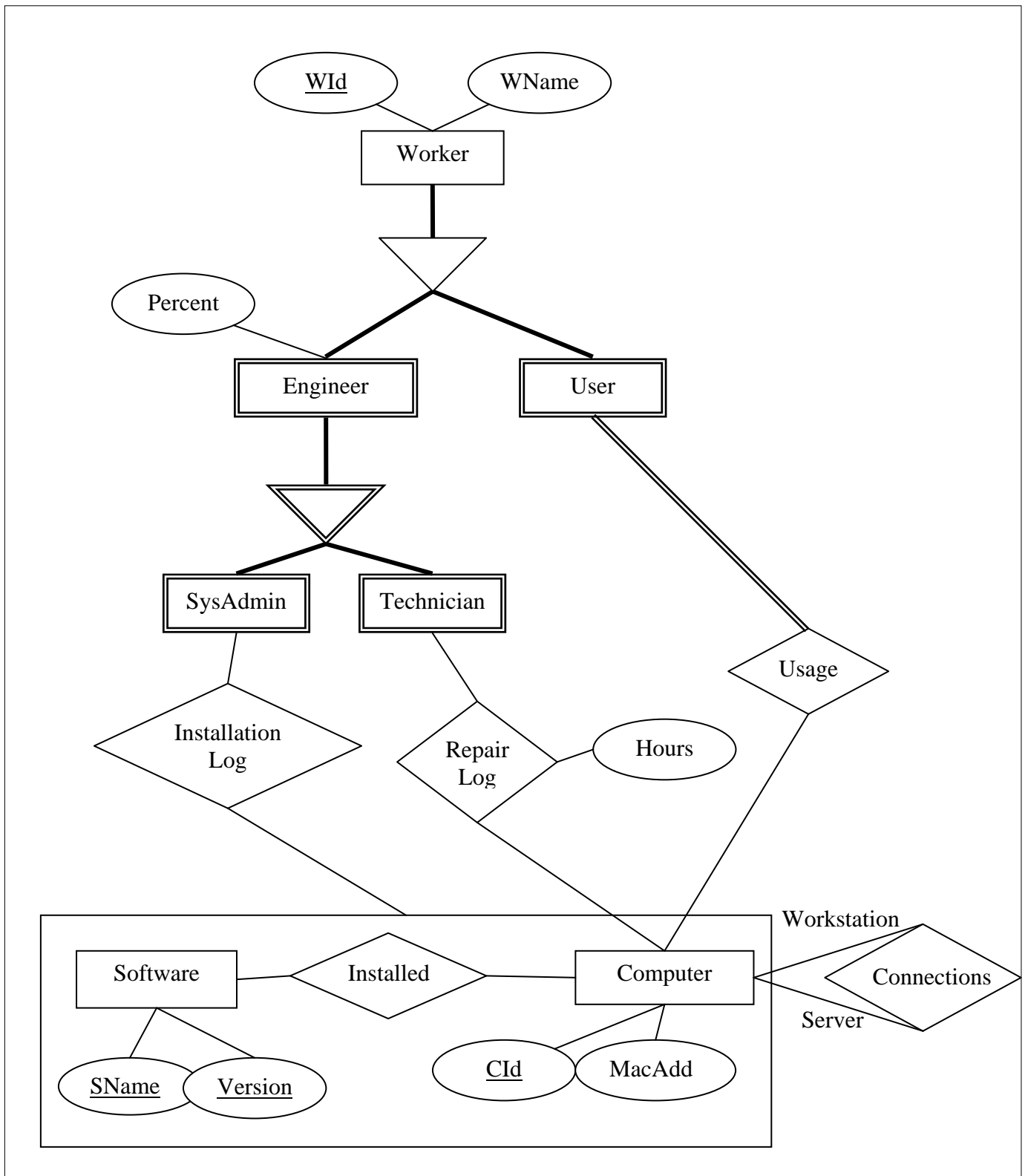
הערות:

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

בהצלחה

שאלה 1 – ERD + ODL (24 נק')

נתונה סכימת ה-ERD הבאה שמתארת מסד נתונים של חוות מחשבים:



הסברים לדיאגרמה:

ישויות:

Worker – מייצג עובד בחוות המחשבים. לכל עובד יש שם (WName) ומספר תעודת זהות (WId) ייחודי. קיימים כמה סוגים של עובדים שונים:

- משתמש – **User**
- מהנדס – **Engineer** – עבור כל מהנדס יש מידע לגבי אחוז המשרה בה מועסק (Percent).
- טכנאי – **Technician** – אחראי על תיקוני חומרה במחשבים.
- מנהל – **SysAdmin** – תפקידו בעיקר לבצע התקנות שונות על מחשבים.

Computer – מייצג מחשב השייך לחווה. לכל מחשב יש מספר מזהה ייחודי (CId) וכתובת ה-Mac (MacAdd).

Software – ישות שמתארת תוכנה. לכל תוכנה יש שם שמתאר אותה (SName) ומספר הגרסה (Version).

יחסים:

Installed – יחס שמתאר אילו תוכנות מותקנות באילו מחשבים.

Connections – יחס שמייצג רשת מחשבים בחווה, כלומר תחנות עבודה (Workstations) המחוברות לשרתים (Servers).

Usage – יחס שמייצג מול אילו מחשבים עובדים המשתמשים.

InstallationLog – יחס שמייצג את כל ההתקנות שבוצעו בחווה.

RepairLog – יחס שמתאר את כל העבודות אשר בוצעו ע"י הטכנאים על מחשבי החווה והזמן שגזלה כל עבודה.

א. (10 נק') מלאו את הטבלאות הבאות בהתייחס לדיאגרמה הנתונה:

Worker	
שדות	WName, WId
מפתחות	WId
Engineer	
שדות	WId, WName, Percent
מפתחות	WId
Technician	
שדות	WId
מפתחות	WId
Connections	
שדות	CIdWorkstation, CIdServer
מפתחות	CIdWorkstation, CIdServer
InstallationLog	
שדות	WID, CID, SName, Version
מפתחות	WID, CID, SName, Version

ב. (8 נק') הוחלט להרחיב את מסד הנתונים המקורי ולהוסיף מידע אודות הרשאות (**Privileges**) של משתמשים לעבודה מול המחשבים. דוגמא להרשאות - הרשאת Admin (המשתמש יכול לעבוד על המחשב ללא הגבלות כלשהן), הרשאה רגילה (המשתמש יכול להפעיל את כל התוכנות, אך לא יכול להתקין שום דבר על המחשב) וכו'. בפניכם עומדים שני תכנונים אפשריים:

1. להוסיף **Privileges** כתכונה של ישות User.
2. להוסיף **Privileges** כתכונה של יחס Usage.

באופן טבעי, כל אחת מהאפשרויות מתאימה לתרחיש אחר באשר לאופן ההתארגנות של החווה. תארו בקצרה איזה תרחיש מתאים לאפשרות הראשונה ואיזה תרחיש מתאים לאפשרות השנייה.

בתכנון הראשון לכל משתמש מותאם סוג הרשאה אחד בעבודה מול כל המחשבים בחווה.

התכנון השני לעומת זאת מאפשר להתאים לאותו משתמש הרשאות שונות בעבודה מול

מחשבים שונים.

ג. (6 נק') בתרגום של דיאגרמת ה-ERD ל-ODL, ציינו את שמות ה-Interfaces שאינם ישויות ב-ERD.

Installed, RepairLog

שאלה 2 – Embedded SQL + DATALOG (19 נק')

שאלה זאת מתייחסת לדיאגרמת ה-ERD משאלה 1. בכל הסעיפים הבאים הניחו תרגום סטנדרטי של סכמת ה-ERD משאלה 1 לסכמות רלציוניות.

נגיד שמשתמש (User) יכול לגשת למחשב מסויים אם הוא עובד מולו (לפי רלצית Usage) או שזהו מחשב שניתן להגיע אליו מהמחשב מולו הוא עובד (לפי רלצית Connections).

א. (10 נק') בסעיף זה נרצה למצוא עבור כל המשתמשים את המחשבים אליהם הם יכולים לגשת. לדוגמא, אם משתמש א' עובד מול מחשב מס' 1 ובנוסף מחשב מס' 1 מחובר למחשב מס' 2 ומחשב מס' 2 מחובר למחשב מס' 3, אזי משתמש א' יכול לגשת למחשבים 1, 2 ו-3. על התוצאה להימצא בטבלא זמנית שתקרא Result. אנו מרשים ש-Result תכיל כפילויות.

לצורך משימה זאת נכתבה תוכנית בשפת C עם הוראות SQL. עליכם להשלים את המקומות החסרים הדרושים להשלמת המשימה.
שימו לב, אתם נדרשים לכתוב רק הוראות SQL או חלקים מהן!

```
#include <libpq-fe.h>
PGconn *conn;
int toContinue = 1;          /*initialize the flag used for looping*/
PGresult *res = NULL;

int main(void) {

    /*initialize connection*/
    conn = ...

    /*create and initialize the Result table with a copy of the contents of Usage*/
    PQexec(conn, "CREATE TABLE Result
                  (WId Integer, CId Integer)");

    PQexec(conn, "INSERT INTO Result
                  (SELECT * FROM Usage)");

    while(toContinue != 0){
        /*update the Result table*/
        PQexec(conn, "INSERT INTO Result

                        SELECT WId,CIdServer FROM Result R, Connections C

                        WHERE R.Cid = C.CIdWorkStation

                    ");
```

```

/* check whether one more iteration is necessary*/
res = PQexec(conn, "
    SELECT * FROM Result R, Connections C
    WHERE R.Cid = C.CidWorkStation
    AND (WId, CIdServer) NOT IN (SELECT * FROM Result) ");

toContinue = PQntuples(res); /* calculates the number of tuples in res (0 to stop) */
}
return 0;
}

```

ב. (9 נק') נגיד שמשתמש הוא משתמש מתקדם (PowerUser) אם הוא יכול לגשת לכל המחשבים הנמצאים בחווה. כתבו תוכנית **Datalog** בטוחה וניתנת לריבוד אשר מחזירה את המשתמשים המתקדמים בתוך פרדיקט סתום שנקרא PowerUser. הניחו כי הסדר של התכונות בפרדיקטים המפורשים הוא אלפביתי.

$result(C,W) \leftarrow Usage(C,W).$

$result(S,W) \leftarrow result(C,W), Connection(S,W).$

$NotPower(U) \leftarrow User(U, _), Computer(C, _), \neg result(C,U).$

$PowerUser(U) \leftarrow User(U, _), \neg NotPower(U).$

הערה: בתשובה למעלה התכונות מסודרות אלפביתית (למשל $Usage[Cid,Wid]$), אבל בבדיקה

התקבלו גם תשובות שהניחו סדר שונה של התכונות.

שאלה 3 – RA + RC (18 נק')

נתונה רלציה Div(Number, Divisor) המתארת את המחלקים של כל המספרים הטבעיים עד 1000. להלן 8 השורות הראשונות ברלציה Div הממוינות לפי Number מיון ראשוני ולפי Divisor מיון משני. שימו לב שברלציה זאת מופיעות גם שורות עם מחלקים טריוויאליים כגון 1 והמספר עצמו.

Number	Divisor
1	1
2	1
2	2
3	1
3	3
4	1
4	2
4	4

א. (8 נק') כתבו שאילתא ב-RA אשר מוצאת את כל המספרים הראשוניים עד 1000.

$\pi_{\text{Number}} \text{Div} \setminus \pi_{\text{Number}} \sigma_{\text{Divisor} \neq 1 \wedge \text{Divisor} \neq \text{Number}} \text{Div}$

ב. (10 נק') נתונה שאילתת RA הבאה:

$\pi_{\lambda}(\pi_{N1, N2}((\rho_{\text{Number} \rightarrow N1, \text{Divisor} \rightarrow D} \text{Div}) \bowtie (\rho_{\text{Number} \rightarrow D, \text{Divisor} \rightarrow N2} \text{Div})) \setminus (\rho_{\text{Number} \rightarrow N1, \text{Divisor} \rightarrow N2} \text{Div}))$

1. רשמו במילים פשוטות מה משמעות השאילתא.

רמז: השאילתא מבצעת מעין בדיקת אמינות של הטבלא.

השאילתא בודקת לכל מספר ב-Div אם כל מחלק של מחלק שלו הוא גם מחלק של המספר עצמו.

אם קיים מספר שמפר תנאי זה אז הטבלא אינה אמינה ותוחזר שורה ריקה (שמתאימה לערך

אמת בוליאני), ואחרת תוחזר טבלא ריקה (חסרת שורות).

2. תרגמו את השאילתא ל-DRC.

$\neg (\forall N1,D1,D2 (Div(N1,D1) \wedge Div(D1,D2)) \rightarrow Div(N1,D2))$

או:

$\exists N1,D1,D2 Div(N1,D1) \wedge Div(D1,D2) \wedge \neg Div(N1,D2)$

שאלה 4 – DESIGN (23 נק')

א. (8 נק') תן דוגמא לסכמה R,F בעלת 3 תכונות לכל היותר הנמצאת ב-3NF, כך שלכל תלות $X \rightarrow Y$ הנמצאת ב-F, X אינו על מפתח.

למשל

$R=\{A,B,C\}$ $F=\{A \rightarrow B, B \rightarrow A\}$

ב. (15 נק') נתונה סכמה R[A, B, C, D] וקבוצת תלויות פונקציונליות F. ידוע כי המפתחות הקבילים של R,F הם ABC ו-ABD בלבד.

1. האם ייתכן מצב בו R נמצאת בצורה נורמלית BCNF יחסית ל-F?
אם כן, תן דוגמא ל-F. אם לא, נמק!

כן, למשל

$F=\{ABC \rightarrow D, ABD \rightarrow C\}$

2. האם ייתכן מצב בו R נמצאת בצורה נורמלית 3NF אך לא ב-BCNF יחסית ל-F?
אם כן, תן דוגמא ל-F. אם לא, נמק!

כן, למשל

$F=\{C \rightarrow D, D \rightarrow C\}$

3. האם ייתכן מצב בו R לא נמצאת בצורה נורמלית 3NF יחסית ל-F?
אם כן, תן דוגמא ל-F. אם לא, נמק!

לא ייתכן מצב כזה כיוון שכל תכונה של R נמצאת במפתח קביל כל שהוא.

שאלה 5 – XML (16 נק')

כל השאלות הבאות מתייחסות לאילן יוחסין המתואר ע"י שני קבצים, fathers.xml ו-mothers.xml. שני הקבצים מצייתים ל-DTD הפשוט הבא, כאשר הבן היחיד של השורש הוא צומת המסמך ancestors:

```
<!ELEMENT ancestors person+>
<!ELEMENT person person*>
<!ATTLIST person name ID #REQUIRED>
```

הרעיון הוא שבנים של איש או אישה מופיעים כצמתים בנים בעץ ה-XML של הקובץ. הקובץ fathers.xml מתאר את שושלת האבות (כך שהנשים המופיעות במשפחות הנידונות יכולות להופיע רק בעלים), והקובץ mothers.xml מתאר את שושלת האימהות. הביטוי בדוגמא הבאה עבור חלק ממשפחתו של אברהם.

fathers.xml:	mothers.xml:
<pre><?xml version="1.0"?> <ancestors> <person name="Avraham"> <person name="Itzchak"> <person name="Yaakov"/> </person> <person name="Ishmael"/> </person> </ancestors></pre>	<pre><?xml version="1.0"?> <ancestors> <person name="Sarah"> <person name="Itzchak"/> </person> <person name="Hagar"> <person name="Ishmael"/> </person> <person name="Rivka"> <person name="Yaakov"/> </person> </ancestors></pre>

א. (8 נק') שאלה זו מתייחסת רק לקובץ mothers.xml. נגיד ששני אנשים הם בעלי קשר אימהי אם יש בשושלת האמהות אם קדמונית משותפת לשניהם. כתבו שאילתת XPath 1.0 המוצאת את כל האנשים שיש להם ע"פ הקובץ קשר אימהי לאיש ששמו Brian.

שימו לב כי צומת המסמך ancestors אינו מציין אדם כל שהוא (מאף מין) אלא רק מכיל רשימה של אימהות.

[//person\[ancestor-or-self::person/descendant-or-self::person/@name="Brian"\]](#)

או:

[id\("Brian"\)/ancestor-or-self::person/descendent-or-self::person](#)

הערה: כאן התשובות מגדירות גם קשר בין אם ובנה כקשר אימהי, אולם בבדיקה התקבלו גם

תשובות המחפשות רק קשרים שבהם האם הקדמונית המשותפת היא אדם שלישי.

ב. (8 נק') שאלה זו מתייחסת לשני הקבצים. זוג אנשים יקרא "מרובה-ילדים" אם יש לו לפחות חמישה ילדים משותפים. כתבו שאילתת XQuery אשר מוצאת את שמות כל הזוגות מרובי הילדים. כל זוג כזה יש להחזיר כצומת חדש מהצורה
`<pair father="..." mother="..." />`. **אין להשתמש בפונקצית id().**

`for $father in doc("fathers.xml")//person, $mother in doc("mothers.xml")//person`

`where count($father/person[@name = $mother/person/@name])>=5`

`return <pair father="{ $father/@name}" mother="{ $mother/@name}" />`
