

## מטלה מ'ס 6 : Transaction

שאלה 1 – מועד ב

נתונות שלוש תנועות כלדקמן:

T1: R(b); R(p); R(c); W(c);  
T2: R(b); W(c); R(c);  
T3: R(b); W(p); R(c); W(p);

נתון התזמון הבא (משמאל לימין):

R1(b); R2(b); R1(p); R1(c); R3(b); W1(c); W2(c); W3(p); R3(c); R2(c); W3(p);

קבעו איזו הגדרה התזמון מקיים:

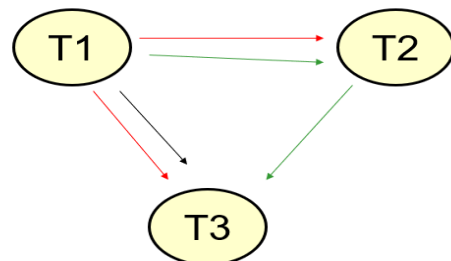
1. תזמון סידרתי
2. תזמון בר סידרתי קונפליקטים
3. תזמון בר סידרתי מבטים
4. התזמון אינו שקול לתזמון סידרתי
5. לא ניתן לקבוע מבלי לדעת מהו סוג הפרוטוקול PL2 המתקיים במקרה זה (רגיל, strict או static)

תזמון בר-סדרתי קונפליקטים וגם תזמון בר סדרתי מבטים.

הסבר:

ניתן לראות בקלות שתזמון זה הינו תזמון בר-סדרתי קונפליקטים ע"י יצירת גרף G כך שכל תנועה  $i$  תיוצג ע"י צומת  $T_i$  בגרף. נמתח קשתות  $T_i \rightarrow T_j$ , ונראה שיתקבל גרף DAG, ז"א גרף חסר מעגלים.

T1	T2	T3
R(b)		
	R(b)	
R(p)		
R(c)		
		R(b)
W(c)		
	W(c)	
		W(p)
		R(c)
		W(p)



כמו כן ניתן גם לראות שתזמון זה הינו תזמון בר-סדרתי מבטים כיוון שהוא שקול לתזמון הסדרתי:

T1	T2	T3
R(b)		
R(p)		
R(c)		
W(c)		
	R(b)	
	W(c)	
	R(c)	
		R(b)
		W(p)
		R(c)
		W(p)

## שאלה 2 – 2021 מועד ב

נתונה הטבלה הבאה Authors(auid, name), ריקה ושתי טרנזקציות

```
INSERT INTO Authors VALUES  
(42,'Emily Dickinson')  
DELETE FROM Authors WHERE name  
LIKE 'John%'
```

```
INSERT INTO Authors VALUES  
(43,'John Keats')  
DELETE FROM Authors WHERE name  
LIKE 'Emi%'  
INSERT INTO Authors VALUES  
(44,'Robert Frost')
```

שתי הטרנזקציות הופעלו במקביל על ידי שני משתמשים שונים עם Serializable Isolation level. בסיס הנתונים קרס בסיום הטרנזקציות, לאחר ביצוע Commit. כאשר בסיס הנתונים חזר לפעולה, הטבלה הכילה את הרשומה של אמילי דיקנסון בלבד. מה ניתן לומר?

1. בסיס הנתונים עומד ב-ACID.
2. בסיס הנתונים לא עומד ב-ACID – אין Consistency.
3. בסיס הנתונים לא עומד ב-ACID – אין Durability.
4. בסיס הנתונים לא עומד ב-ACID – אין Atomicity.
5. בסיס הנתונים לא עומד ב-ACID – אין Availability.

## התשובה היא (3)

הסבר: נתון שבסיס הטרנזקציה בוצע committed, ז"א בסיס הנתונים התחייב לכך שהטרנזקציה הסתיימה כמו שצריך ושהנתונים ישמרו לאחר ביצועה ביעילות. אולם, אין Durability כיוון שהאפקט של התנועה שהסתיימה כן נעלם, היות ונתון שהרשומה הכילה את אימילי דיקנסון בלבד ללא שאר הנתונים שהיו אמורים להכנס לבסיס הנתונים. isolation כן מקתיים כיוון שהפעולות שמבצעות התנועות אינן תלויות אחת בשנייה, כמו כן Atomicity גם חל כיוון שהתנועה אכן מבוצעת כשמקשה אחת, לבסוף גם Availability מתקיים כיוון שהתנועה אינה מפרה את האילוצים על מבנה הנתונים.

## שאלה 3- 2017 מועד א

DBMS מקבל לביצוע את הפעולות הבאות מתנועות שונות ומנסה לבצע אותן משמאל לימין:

$T1:R(X), T2:W(Y), T1:R(Y), T3:R(Z), T2:W(Z), T3:W(Z)$

אפשר להניח שאם שתי תנועות מחכות לאותו המנעול שישתחרר אז זאת שחיכתה ראשונה תזכה במנעול קודם.

- א. הנח כי הבסיס נתונים משתמש ב-2 phase locking ושכל תנועה משחררת את המנעולים שלה מיד כשמתאפשר (בלי לוותר על ה-2 phase locking). תחת הנחות אלו, לאיזה תזמון סדרתי שקול הרצת הפעולות הנ"ל? **הסבירו את תשובתכם.**
- ב. הנח כי הבסיס נתונים משתמש ב-Straight 2 phase locking ושכל תנועה משחררת את המנעולים שלה מיד כשמתאפשר. בנוסף, הנח כי כל תנועה מנסה לבצע commit מיד לאחר הפעולה האחרונה שלה. תחת הנחות אלו, לאיזה תזמון סדרתי שקול הרצת הפעולות הנ"ל? **הסבירו את תשובתכם.**

א. התזמון הסדרתי הוא  $T3 \rightarrow T2 \rightarrow T1$  שהוא בר סדרתיות קונפליקטית. בכל תזמון אחר נכנס למצב של דדלוק-  $T1$  מבקשת מנעול קריאה על  $Y$  אך  $T2$  מחזיקה עליו כבר מנעול כתיבה ולכן צריך לחכות שהיא תסתיים. כמו כן,  $T2$  מעוניינת לקבל מנעול כתיבה על  $Z$  אך  $T3$  כבר מחזיקה עליו מנעול קריאה ולכן צריכה לחכות שהיא תסתיים. על כן התזמון היחיד בו אנו לא נכנסים למצב של דדלוק הוא כאשר הסדר הנ"ל הוא הסדר בו מתבצעות הטרנזקציות.

ב. הרצת הפעולות הנ"ל שקול לתזמון בר התאוששות, התזמון הסדרתי יהיה  $T1, T1 \rightarrow T3 \rightarrow T2$  מקבלת מנעול משותף על  $X, Y$  בתחילת הריצה שלה.  $T2$  מחכה ש- $T1$  תסתיים ותעשה commit בכדי שתוכל להתחיל לרוץ, מכיוון שהיא צריכה מנעול על  $Y$ . בזמן זה,  $T3$  מתחילה לרוץ, היא צריכה מנעול משותף רק על  $Z$  ולכן אין לה בעיה לרוץ. ברגע שתנועה  $T1$  מסתיימת,  $T2$  צריכה להמתין כעת לתנועה  $T3$  שתסתיים כדי לקבל מנעול משותף על  $Z$ . ברגע שמסתיימת תנועה  $T3$  יכולה תנועה  $T2$  להתחיל לרוץ.