

תרגיל 6 : Transaction Management

תאריך הגשה : 55:23, 12.06.22.

הוראות הגשה:

בתרגיל זה אתם נדרשים להגיש קובץ zip בודד שיכלול את הקבצים הבאים :

- ex6.pdf עם התשובות מפורטות לשאלות.
- README שמכיל שורה בודדת ובו ה-login של הסטודנט שמגיש את התרגיל. אם התרגיל מוגש בזוגות, על שורה זאת להכיל את שני ה-login מופרדים בפסיק.

שימו לב:

- נא לקרוא על הדרישות המנהליות של הקורס בלינק באתר הקורס כדי למלא אחר ההוראות להגשה של קבצים סרוקים!
- תרגיל מוקלד יזכה ב- 2 נקודות בונוס!

על מנת להקל על הבדיקה של התרגיל הזה, אתם מתבקשים לענות עליו בגוף התרגיל עצמו. זאת גם הסיבה שהתרגיל נראה כל כך ארוך (למרות שמבחינת השאלות הוא אינו ארוך).

שאלה 1: (36 נקודות)

נתון התזמון :

	T1	T2	T3
1		R(Z)	
2		R(Y)	
3			R(X)
4	R(X)		
5			R(Z)
6			R(Y)
7			Commit
8		W(Z)	
9		W(Y)	
10	R(Z)		
11	W(Z)		
12	Commit		
13		R(X)	
14		Commit	

ענה על השאלות הבאות, ונמק בקצרה את תשובתך.

1

א. כמה קריאות מלוכלכות (dirty reads) יש בתזמון?

נימוק: קריאה מלוכלכת מתבצעת כאשר טרזנקציה קוראת אובייקט שנכתב על ידי טרזנקציה אחרת, שעוד לא ביצעה commit. הקריאה R(Z) של T1 מלוכלכת.

0

ב. כמה קריאות שלא ניתנות לשחזור (nonrepeatable reads) יש בתזמון?

נימוק: אפס. קריאה שלא ניתנת לשחזור מתרחשת כאשר טרנזקציה קוראת את אותו אובייקט פעמיים ומגלה שטרנזקציה אחרת שינתה אותה. בתזמון הזה אין אף טרנזקציה שקוראת אובייקט פעמיים.

לא

ג. האם התזמון נמנע מ-cascading aborts? הקיף את התשובה הנכונה: כן

נימוק: תזמון נמנע מ-cascading aborts אם אין קריאות מלוכלכות. ראינו כבר בסעיף א שיש קריאה מלוכלכת.

לא

ד. האם התזמון הוא בר-התאוששות (recoverable)?

נימוק: תזמון הוא בר-התאוששות אם טרנזקציות מבצעות commit רק לאחר שהטרנזקציות שאת שינוייהם הם קראו, מבצעות commit. T1 קורא את השינוי של T2 ומבצע commit לפניו, לכן התזמון אינו בר-התאוששות.

לא

ה. האם התזמון בר-סידור קונפליקטים (conflict serializable)?

נימוק: בגרף הקדימויות יש צלע מ-T1 ל-T2 ול-T3, ומ-T2 ל-T1. אין מעגל ולכן התזמון בר-סידור קונפליקטים.

לא

ו. האם התזמון יכול להיווצר על ידי פרוטוקול 2PL?

נימוק: ניתן להוסיף פעולות של נעילה ושחרור לתזמון כך שהתזמון יקיים את תנאי פרוטוקול 2PL.

לא

כן

ז. האם התזמון יכול להיווצר על ידי פרוטוקול strict 2PL ?

נימוק: T2 חייב לשחרר מוקדם את המנעול על Z.

ח. האם התזמון יכול להיווצר על ידי פרוטוקול חותמות הזמן כאשר

$$TS(T1) = 1, TS(T2) = 2, TS(T3) = 3$$

לא

כן

הקיף את התשובה הנכונה :

8

אם ענית לא, באיזה שורה הפרוטוקול ייכשל?

ט. האם התזמון יכול להיווצר על ידי פרוטוקול חותמות הזמן כאשר

$$TS(T1) = 3, TS(T2) = 2, TS(T3) = 1$$

לא

כן

הקיף את התשובה הנכונה :

אם ענית לא, באיזה שורה הפרוטוקול ייכשל?

שאלה 2 (30 נקודות)

בתזמון הבא ציינו את זמני ההתחלה של הטרנזקציות, בקשות למנעול משותף (למשל T4 מבקש מנעול משותף על A) ובקשות למנעול אקסלוסיבי (למשל T3 מבקש מנעול אקסלוסיבי על B). שימו לב שהטרנזקציות אינם מבקשות לשחרר מנעולים.

T1	T2	T3	T4
			BEGIN
	BEGIN		
		BEGIN	
			S(A)
		X(B)	
BEGIN			
	S(A)		
	S(B)		
X(C)			
X(B)			
			X(C)

הערה: אין צורך להתחשב בריצה מחודשת של טרנזקציות שנופלות.

א. נניח שמנהל המנעולים משתמש בשיטת **wait-die**. בחר את כל התשובות הנכונות.

1. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T1 (כלומר יעשה לו abort)
2. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T1 (כלומר יגרום לו לחכות)
3. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T2 (כלומר יעשה לו abort)
4. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T2 (כלומר יגרום לו לחכות)
5. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T3 (כלומר יעשה לו abort)
6. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T3 (כלומר יגרום לו לחכות)
7. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T4 (כלומר יעשה לו abort)
8. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T4 (כלומר יגרום לו לחכות)
9. אף אחת מהתשובות האחרות איננה נכונה

ב. נניח שמנהל המנעולים משתמש בשיטת **wound-wait**. בחר את כל התשובות הנכונות.

1. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T1 (כלומר יעשה לו abort)
2. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T1 (כלומר יגרום לו לחכות)
3. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T2 (כלומר יעשה לו abort)
4. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T2 (כלומר יגרום לו לחכות)
5. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T3 (כלומר יעשה לו abort)
6. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T3 (כלומר יגרום לו לחכות)
7. במהלך ביצוע התזמון, מנהל המנעולים יפיל את T4 (כלומר יעשה לו abort)
8. במהלך ביצוע התזמון, מנהל המנעולים ישהה את T4 (כלומר יגרום לו לחכות)
9. אף אחת מהתשובות האחרות איננה נכונה

שאלה 3 (34 נקודות)

למדנו שניתן להריץ טרנזקציות ברמות בידוד שונות, ובהתאם, התנהגות הטרנזקציות עלולה להיות שונה. הבנה טובה של רמות בידוד הוא קריטי באפליקציה אמתית. בחירת רמת הבידוד יכול להשפיע גם על נכונות הנתונים במסד, וגם על יעילות האפליקציה. בשאלה זו, אתם תתנסו בהרצה של אותו קוד ברמות בידוד שונות, ותדרשו לנמק את ההבדלים בתוצאות.

נתונים 3 תזמונים. לפני הרצת כל אחד מהתזמונים, מייצרים טבלה ומכניסים שורות:

```
CREATE TABLE grades(id integer primary key, name varchar, grade integer);  
INSERT INTO grades VALUES(1, 'alice', 80), (2, 'bob', 90), (3, 'claire', 100);
```

ולאחר הרצת כל אחד מהתזמונים, הטבלה נמחקת. **שימו לב:** פקודות עדכון (insert או update) שמסתיימים ב * returning, מחזירות למשתמש את השורות שהשתנו על ידי פעולת העדכון. כמו כן, שימו לב שאנחנו נתעניין בעיקר בתוצאות של השורות המודגשות בצהוב.

תזמון 1:

	<u>T1</u>	<u>T2</u>
1	Select * from grades;	
2		Select * from grades where id = 1;
3	update grades set grade = grade +10 where id = 1 returning *;	
4		Select * from grades where id = 1;
5	Commit;	
6		Select * from grades where id = 1;
7		Commit;

תזמון 2:

	<u>T1</u>	<u>T2</u>
1	Select * from grades;	
2		Select * from grades where grade = 100;
3	insert into grades values(4, 'dan', 100) returning *;	
4		Select * from grades where grade = 100;
5	Commit;	
6		Select * from grades where grade = 100;
7		Commit;

תזמון 3:

	<u>T1</u>	<u>T2</u>	<u>T3</u>
1	Select * from grades;		
2	insert into grades select 5, 'trans1', avg(grade) from grades returning *;		
3	Select * from grades;		
4		Select * from grades;	
5		insert into grades select 6, 'trans2', avg(grade) from grades returning *;	
6		Select * from grades;	
7	Commit;		
8		Commit;	
9			Select * from grades;

עליכם להריץ את :

- תזמון 1 ברמות בידוד read committed ו repeatable read
 - תזמון 2 ברמות בידוד read committed ו repeatable read
 - תזמון 3 ברמות בידוד repeatable read ו serializable
- יש 2 דרכים שונות להריץ את התזמונים, ותוכלו לבחור בדרך הנוחה לכם :

- הרצה ידנית: תפתחו חלון של postgres עבור כל טרנזקציה. בחלון הראשון, תרשמו את הפקודות של T1 בחלון הראשון ובחלון השני תרשמו את הפקודות של T2. שימו לב להפעיל את הפקודות לפי הסדר שרשום בתזמון, וכן להשתמש בפקודת BEGIN TRANSACTION ISOLATION LEVEL עם רמת הבידוד הדרושה. **הערה: השיטה הזאת פחות מומלצות, בגלל הקלות לטעות במהלך הכנסת הפקודות.**

- הרצה בעזרת תוכנית run-schedules.py: על מנת להקל עליכם, כתבנו תוכנית python שמתחבר למסד נתונים שלכם ומריץ את התזמונים. התוכנית רושמת את הפלט של כל אחד מהפקודות למסך. כדי להריץ את run-schedules.py, הורידו אותה מאתר הקורס לחשבון שלכם באוניברסיטה. התחברו לחשבון linux שלכם באוניברסיטה. בתיקה שבו שמרתם את התוכנית, הריצו :

```
python run-schedules.py <user-name> <schedule-num> <isolation-level>
```

כאשר

- user-name הוא שם המשתמש שלכם ב linux,
- schedule-num הוא מספר 1, 2, או 3
- Isolation-level הוא RC (בשביל read committed), RR (בשביל repeatable read), או S (בשביל serializable)

להזכירכם, תצטרכו להריץ את התוכנית 6 פעמים, עם הפקודות :

- python run-schedules.py <user-name> 1 RC
- python run-schedules.py <user-name> 1 RR
- python run-schedules.py <user-name> 2 RC
- python run-schedules.py <user-name> 2 RR
- python run-schedules.py <user-name> 3 RR
- python run-schedules.py <user-name> 3 S

לאחר שתריצו את התזמונים, ענו על השאלות הבאות. בהסברים שלכם, עליכם להתייחס לרמת הבידוד ולמושגים כגון dirty write, dirty read, nonrepeatable read, phantom, serialization anomaly שנלמדו בהרצאות TM1 ו TM2.

תזמון 1, רמת בידוד read committed:

- מה מוחזר על ידי שורה 4?
(1, alice, 80)

- מה מוחזר על ידי שורה 6?
(1, alice, 90)

- האם 2 השורות החזירו את אותם תוצאות? כן
- אם כן, הסבר כיצד זה קשור לרמת הבידוד בו רץ השאילתה. לא

- אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read, serialization anomaly, phantom?
Non-repeatable read

- האם שני הטראנסקציות הצליחו לבצע commit? כן
- אם לא, מדוע? לא

תזמון 1, רמת בידוד repeatable read:

- מה מוחזר על ידי שורה 4?
(1, alice, 80)

- מה מוחזר על ידי שורה 6?
(1, alice, 80)

- האם 2 השורות החזירו את אותם תוצאות? כן
- אם כן, הסבר כיצד זה קשור לרמת הבידוד בו רץ השאילתה. לא
- רמת הבידוד אינו מאפשר non-repeatable read ולכן T2 ממשיך לראות את אותו ערך.

- אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read, serialization anomaly, phantom?

- האם שני הטראנסקציות הצליחו לבצע commit? כן
- אם לא, מדוע? לא

תזמון 2, רמת בידוד read committed:

- מה מוחזר על ידי שורה 4?

(3, claire, 100)

- מה מוחזר על ידי שורה 6?

(3, claire, 100), (4, dan, 100)

לא

כן

- האם 2 השורות החזירו את אותם תוצאות?

- אם כן, הסבר כיצד זה קשור לרמת הבידוד בו רץ השאילתה.

- אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read,

phantom, serialization anomaly?

Phantom

לא

כן

- האם שני הטראנסקציות הצליחו לבצע commit?

- אם לא, מדוע?

תזמון 2, רמת בידוד repeatable read:

- מה מוחזר על ידי שורה 4?

(3, claire, 100)

- מה מוחזר על ידי שורה 6?

(3, claire, 100)

לא

כן

- האם 2 השורות החזירו את אותם תוצאות?

- אם כן, הסבר כיצד זה קשור לרמת הבידוד בו רץ השאילתה.

רמת הבידוד אינו מאפשר phantom ולכן T2 ממשיך לראות את אותו קבוצה של ערכים.

- אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read,

phantom, serialization anomaly?

לא

כן

- האם שני הטראנסקציות הצליחו לבצע commit?

- אם לא, מדוע?

תזמון 3, רמת בידוד repeatable read :

- מה מוחזר על ידי שורה 3?
(1, alice, 80), (2, bob, 90), (3, claire, 100), (5, trans1, 90)
- מה מוחזר על ידי שורה 6?
(1, alice, 80), (2, bob, 90), (3, claire, 100), (6, trans2, 90)
- מה מוחזר על ידי שורה 9?
(1, alice, 80), (2, bob, 90), (3, claire, 100), (5, trans1, 90), (6, trans2, 90)
- האם התוצאות שקולות לריצה סדרתית כלשהו של הטראנסקציות שביצעו commit?
כן לא
- אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read, phantom, serialization anomaly?
- האם שני הטראנסקציות הצליחו לבצע commit?
כן לא
- אם לא, מדוע?

תזמון 3, רמת בידוד serializable :

- מה מוחזר על ידי שורה 3?
(1, alice, 80), (2, bob, 90), (3, claire, 100), (5, trans1, 90)
 - מה מוחזר על ידי שורה 6?
(1, alice, 80), (2, bob, 90), (3, claire, 100), (6, trans2, 90)
 - מה מוחזר על ידי שורה 9?
(1, alice, 80), (2, bob, 09), (3, claire, 100), (5, trans1, 90)
 - האם התוצאות (כלומר מצב הטבלה בסיום הריצה) שקולות לריצה סדרתית כלשהו של הטראנסקציות שביצעו commit?
כן לא
 - אם לא, איזה מהתופעות הבאות התרחשה, dirty write, dirty read, nonrepeatable read, phantom, serialization anomaly?
 - האם שני הטראנסקציות הצליחו לבצע commit?
כן לא
 - אם לא, מדוע?
- ברמת בידוד Serializable לא ניתן לאפשר serialization anomaly והיה יכו להיווצר כזו תופעה.
רק במקרה בגלל השימוש בפונקציית avg התוצאה היתה שקולה לריצה סדרתית.