תרגיל 6 : Transaction Management

תאריך הגשה : 17.01.21 ,23:55.

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!

<u>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).</u>

<u>שאלה 1:</u> (36 נקודות)

נתון התזמון :

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

ענה על השאלות הבאות, ונמק בקצרה את תשובתך.

א. כמה קריאות מלוכלכות (dirty reads) יש בתזמון? 1

<u>נימוק:</u> קריאה מלוכלכת מתבצעת כאשר טרזנקציה קוראת אובייקט שנכתב על ידי טרזנקציה אחרת, שעוד לא ביצעה commit. הקריאה R(X) של T3 מלוכלכת.

ב. כמה קריאות שלא ניתנות לשחזור (nonrepeatable reads) יש בתזמון? 0

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.

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

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

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

נימוק: תזמנון הוא בר התאוששות אם טרנזקציות מבצעות commit רק לאחר שהטרנזקציות שאת שינוייהם הם קראו, מבצעות commit. T3 קורא את השינוי של T1 ומבצע commit אחריו.

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

נימוק: בגרף הקדימויות יש צלע מT2 לT1 ומT1 לT3. אין מעגל ולכן התזמון בר סידור קונפליקטים.

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

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

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

נימוק: T1 חייב לשחרר מוקדם את המנעול על X.

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

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

הקיף את התשובה הנכונה: כן לא

אם ענית לא, באיזה שורה הפרוטוקול ייכשל? 7

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

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

הקיף את התשובה הנכונה: כן לא

אם ענית לא, באיזה שורה הפרוטוקול ייכשל?

שאלה 2 (30 נקודות)

א. תן דוגמה *קצרה* של תזמון שהוא בר סידור קונפליקטים אך אינו ניתן להשגה על ידי 2PL.

רמז: אפשר למצוא תזמון שמקיים את הדרישה שיש בו 3 טרנזקציות, 2 פעולות קריאה ו2 פעולות כתיבה בסה"כ. מלאו את הטבלה הבאה עם הפתרון שלכם.

T1	T2	T3
R(A)		
	W(A)	
		W(B)
R(B)		

ב. הוכח שכל תזמנון שיש בו 2 טרנזקציות בלבד הוא בר סידור קונפליקטים אם ורק אם ניתן להשיג את התזמנון על ידי 2PL.

כבר הוכחנו בכיתה שאם התזמון ניתן להשגה על ידי 2PL אז הוא בר סידור קונפליקטים. נוכח את הכיוון שני. יהי S תזמנון בר סידור קונפליקטים שמכיל שתי טרנזקציות T1 ו T2.

נתבונן על גרף הקדימויות. מכיוון ש S בר סידור קונפליקטים, אין מעגל בגרף. נניח ללא הגבלת הכלליות שאין צלע מ T2 ל T1. כלומר, בכל קונפליקט, T1 מופיע לפני T2. לכן, T1 יכול לבקש את כל המנעולים אליו הוא יהיה זקוק בתחילת התזמנון, ויכול לשחרר כל מנעול בתום השימוש בו. לאחר מכן, T2 יכול לקבל את המנעול. מכיוון שאין טרנזקציות נוספות, T2 יכול לשחרר את כל המנעולים בסוף התזמון.

מכיוון שהטרנזקציות אף פעם לא מבקשות מנעול לאחר שחרור, התזמנון מקיים את פרוטוקול 2PL.

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 accounts(id integer primary key, owner varchar, balance integer);
INSERT INTO accounts VALUES(1, 'alice', 100), (2, 'bob', 100), (3, 'claire', 100);
```

ולאחר הרצת כל אחד מהתזמונים, הטבלה נמחקת. **שימו לב:** פקודות עדכון (update או insert) 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:

	T1	T2
1	Select * from accounts;	
2		Select * from accounts where id = 1;
3	update accounts set balance = balance – 10 where id = 1 returning *;	
4		Select * from accounts where id = 1;
5	Commit;	
6		Select * from accounts where id = 1;
7		Commit;

תזמון 2:

	T1	T2
1	Select * from accounts;	
2		Select * from accounts where balance = 100;
3	insert into accounts values(4, 'dan', 100) returning *;	
4		Select * from accounts where balance = 100;
5	Commit;	
6		Select * from accounts where balance = 100;
7		Commit;

תזמון 3:

	T1	T2	T3
1	Select * from accounts;		
2	insert into accounts select 5, 'trans1', sum(balance) from accounts returning *;		
3	Select * from accounts;		
4		Select * from accounts;	
5		insert into accounts select 6, 'trans2', sum(balance) from accounts returning *;	
6		Select * from accounts;	
7	Commit;		
8		Commit;	
9			Select * from accounts;

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. בתיקיה 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, 100)

- מה מוחזר על ידי שורה 6?

(1, alice, 90)

- האם 2 השורות החזירו את אותם תוצאות? כן לא
- 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?

Non-repeatable read

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

תזמון 1, רמת בידוד repeatable read:

- מה מוחזר על ידי שורה 4?

(1, alice, 100)

- מה מוחזר על ידי שורה 6?

(1, alice, 100)

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

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, phantom, serialization anomaly?

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

תזמון 2, רמת בידוד read committed:

- מה מוחזר על ידי שורה 4?

(1, alice, 100), (2, bob, 100), (3, claire, 100)

- מה מוחזר על ידי שורה 6?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (4, dan, 100)

- האם 2 השורות החזירו את אותם תוצאות? כן לא
- 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

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

תזמון 2, רמת בידוד repeatable read:

- מה מוחזר על ידי שורה 4?

(1, alice, 100), (2, bob, 100), (3, claire, 100)

- מה מוחזר על ידי שורה 6?

(1, alice, 100), (2, bob, 100), (3, claire, 100)

- האם 2 השורות החזירו את אותם תוצאות? כן לא
- 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?

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

תזמון 3, רמת בידוד repeatable read:

- מה מוחזר על ידי שורה 3?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (5, trans1, 300)

- מה מוחזר על ידי שורה 6?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (6, trans2, 300)

- מה מוחזר על ידי שורה 9?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (5, trans1, 300), (6, trans2, 300)

- האם התוצאות שקולות לריצה סדרתית כלשהו של הטרנזקציות שביצעו commit?

כן לא

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

Serialization anomaly.

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

תזמון 3, רמת בידוד serializable:

- מה מוחזר על ידי שורה 3?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (5, trans1, 300)

- מה מוחזר על ידי שורה 6?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (6, trans2, 300)

- מה מוחזר על ידי שורה 9?

(1, alice, 100), (2, bob, 100), (3, claire, 100), (5, trans1, 300)

- האם התוצאות שקולות לריצה סדרתית כלשהו של הטרנזקציות שביצעו commit?

כן לא

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

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

ברמת בידוד Serializable לא ניתן לאפשר serialization anomoly
