Transaction

Thursday, 16 October 2025 20:14

Terpenuhinya seluruh properti dari transaksi (*ACID Property*) merupakan tanggung jawab DBMS sepenuhnya

- ○ true
- ● false

Properti transaksi mana yang dideskripsikan oleh kalimat berikut ini?
Semua perubahan yang dilakukan terhadap basis data oleh sebuah transaksi yang *commit* harus bertahan di basis data dan tidak boleh hilang seandainya terjadi kegagalan sistem

- ○ Atomicity
- ○ Consistency
- ○ Isolation
- ● Durability

Properti *isolation* dari transaksi pada basis data memberikan kesan seolah-olah hanya ada 1 transaksi yang berjalan di dalam sistem meskipun pada saat itu ada sejumlah transaksi yang sedang berjalan secara bersama-sama

- ● true
- ○ false

Diketahui *schedule* transaksi sebagai berikut.

	T1	T2
1	read(X)	
2	X := X – 100	
3	write(X)	
4		read(X)
5		temp := X * 0.2
6		X := X – temp
7		write(X)
8	read(Y)	
9	Y := Y + 100	
10	write(Y)	
11		read(Y)
12		Y := Y + temp
13		write(Y)

Jika pada kondisi awal nilai X = 1000 dan Y = 2000 serta kondisi konsisten dinyatakan sebagai X + Y = 3000, mana sajakah pernyataan yang benar terkait *schedule* konkuren di atas?

- ○ Schedule di atas tidak menjaga konsistensi basis data.
- ● Schedule di atas conflict equivalent dengan schedule serial T1 -> T2.
- ○ Schedule di atas conflict equivalent dengan schedule serial T2 -> T1.
- ○ Schedule di atas tidak conflict serializable.

Berikut ini adalah *schedule* hasil eksekusi 3 buah transaksi secara konkuren. Pada *schedule* Ry(A) dan Wy(A) menyatakan operasi yang dilakukan oleh transaksi Ty terhadap item data A, Cy menyatakan commit transaksi Ty.
R1(X); R3(Y); R3(X); W3(Y); R2(Y); R1(Y); W3(X); R2(X); C1; C2; C3;
Schedule tersebut adalah *schedule* yang:

- ○ Serializable, recoverable, dan cascadeless
- ○ Serializable, recoverable, tapi tidak cascadeless
- ○ Serializable tapi tidak recoverable
- ● Tidak serializable

Berikut ini adalah *schedule* hasil eksekusi 3 buah transaksi secara konkuren. Pada *schedule* Ry(A) dan Wy(A) menyatakan operasi yang dilakukan oleh transaksi Ty terhadap item data A, Cy menyatakan commit transaksi Ty.
R1(X); R2(Z); R1(Z); R3(X); R3(Y); W1(X); C1; W3(Y); C3; R2(Y); W2(Z); W2(Y); C2;
Schedule tersebut adalah *schedule* yang:

- ● Serializable, recoverable, dan cascadeless
- ○ Serializable, recoverable, tapi tidak cascadeless
- ○ Serializable tapi tidak recoverable
- ○ Tidak serializable

Berikut ini adalah *schedule* hasil eksekusi 3 buah transaksi secara konkuren. Pada *schedule* Ry(A) dan Wy(A) menyatakan operasi yang dilakukan oleh transaksi Ty terhadap item data A, Cy menyatakan commit transaksi Ty.
R1(X); R2(Z); R1(Z); R3(X); R3(Y); W1(X); W3(Y); R2(Y); W2(Z); W2(Y); C1; C2; C3;
Schedule tersebut adalah *schedule* yang:

- ○ Serializable, recoverable, dan cascadeless
- ○ Serializable, recoverable, tapi tidak cascadeless
- ● Serializable tapi tidak recoverable
- ○ Tidak serializable

Berikut ini adalah *schedule* hasil eksekusi 2 buah transaksi secara konkuren. Pada *schedule* Ry(A) dan Wy(A) menyatakan operasi yang dilakukan oleh transaksi Ty terhadap item data A, Cy menyatakan commit transaksi Ty.
R1(Z); W1(Y); R2(Y); W2(X); W1(X); W2(X); C1; C2;
Schedule tersebut adalah *schedule* yang:

- ○ Serializable, recoverable, dan cascadeless
- ○ Serializable, recoverable, tapi tidak cascadeless
- ○ Serializable tapi tidak recoverable
- ● Tidak serializable

Diantara masalah yang harus dicegah saat eksekusi transaksi adalah: (1) *dirty read*, (2) *lost update*, (3) *transaction failure*, dan (4) *inconsistent database state*. Apabila dua transaksi T1 dan T2 yang berjalan secara konkuren dan melakukan perubahan terhadap item data yang sama dieksekusi tanpa menggunakan mekanisme kontrol konkurensi, maka masalah yang dapat muncul adalah:

- ○ (1), (2), dan (3)
- ○ (3) dan (4)
- ○ (1) dan (2)
- ● (1), (2), dan (4)

Pernyataan tentang level konsistensi yang didefinisikan pada SQL-92 berikut ini benar, kecuali:

- ● *Serializable*: seluruh transaksi dijalankan secara serial untuk memastikan terjaganya properti *isolation*.
- ○ *Repeatable read*: hanya data yang sudah *committed* yang dapat dibaca dan hasil pembacaan sebuah item selalu sama. Memungkinkan terjadinya *phantom phenomena*.
- ○ *Read committed*: hanya data yang sudah *committed* yang dapat dibaca. Jika sebuah item data dibaca beberapa kali, nilai yang diperoleh mungkin tidak sama.
- ○ *Read uncommitted*: data yang belum *committed* tetap boleh dibaca.

1. Perlihatkan bahwa setiap eksekusi serial dari kedua transaksi tersebut akan selalu menjaga konsistensi basis data
   a. Serial T1 → T2

T1	T2
read(A);	
read(B);	
if A=0 then B := B+1;	
write(B);	
	read(B);
	read(A);
	if B=0 then A := A+1;
	write(A)

T1 → A = 0; B = 1
T2 → B ≠ 0; A = 0 → KONSISTEN

   b. Serial T2 → T1

T1	T2
	read(B);
	read(A);
	if B=0 then A := A+1;
	write(A)
read(A);	
read(B);	
if A=0 then B := B+1;	
write(B);	

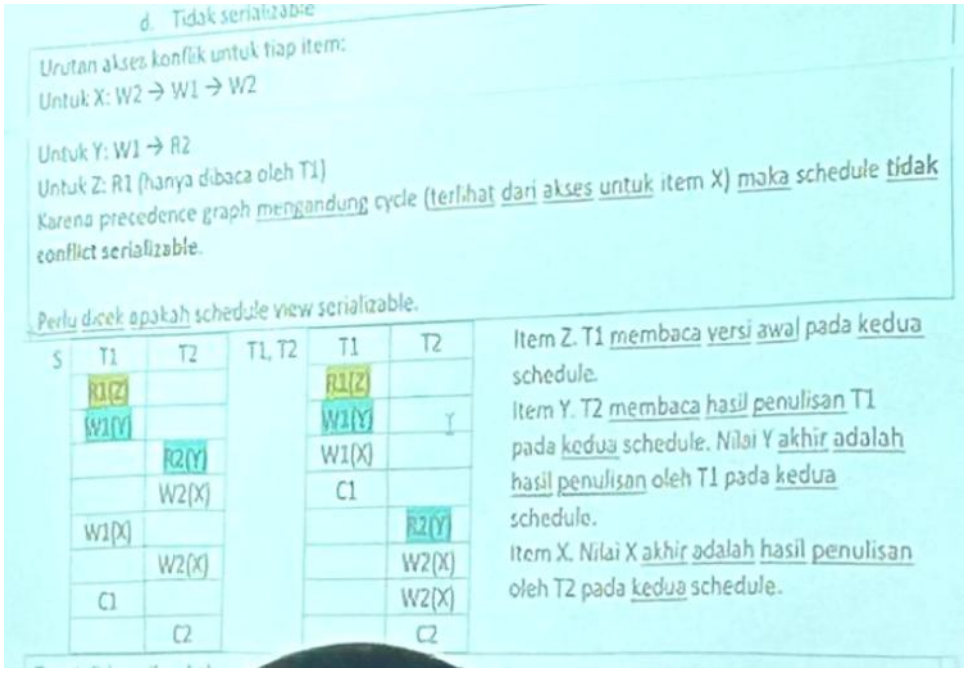
T2 → B = 0; A = 1
T1 → A ≠ 0; B = 0 → KONSISTEN

2. Berikan satu contoh *schedule* eksekusi konkuren kedua transaksi tersebut yang tidak *serializable*. Jelaskan jawaban Anda.

T1	T2
	read(B);
read(A);	
read(B);	
if A=0 then B := B+1;	
write(B);	
	read(A);
	if B=0 then A := A+1;
	write(A)

Hasil akhir A = 1; B = 1
Pada A, T1 → T2
Pada B, T2 → T1
Sehingga, T1 ↔ T2, tidak *conflict-serializable*, tidak *serializable*

3. Apakah dimungkinkan eksekusi konkuren kedua transaksi tersebut menghasilkan *serializable schedule*?
Tidak dimungkinkan karena terdapat karena R2(B) tidak bisa menyeberang W1(B) lewat serangkaian *swap non-conflicting*, siklus tetap ada.


d. Tidak serializable

Urutan akses konflik untuk tiap item:
Untuk X: W2 → W1 → W2

Untuk Y: W1 → R2
Untuk Z: R1 (hanya dibaca oleh T1)
Karena precedence graph mengandung cycle (terlihat dari akses untuk item X) maka schedule tidak conflict serializable.

Perlu dicek apakah schedule view serializable.

S	T1	T2
	R1(Z)	
	W1(Y)	
		R2(Y)
		W2(X)
	W1(X)	
		W2(X)
	C1	
		C2

T1, T2	T1	T2
	R1(Z)	
	W1(Y)	
	W1(X)	
	C1	
		R2(Y)
		W2(X)
		W2(X)
		C2

Item Z. T1 membaca versi awal pada kedua schedule.
Item Y. T2 membaca hasil penulisan T1 pada kedua schedule. Nilai Y akhir adalah hasil penulisan oleh T1 pada kedua schedule.
Item X. Nilai X akhir adalah hasil penulisan oleh T2 pada kedua schedule.