Veritabanı Yönetim Sistemleri (335)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

L23- Eşzamanlılık - (Concurreny)

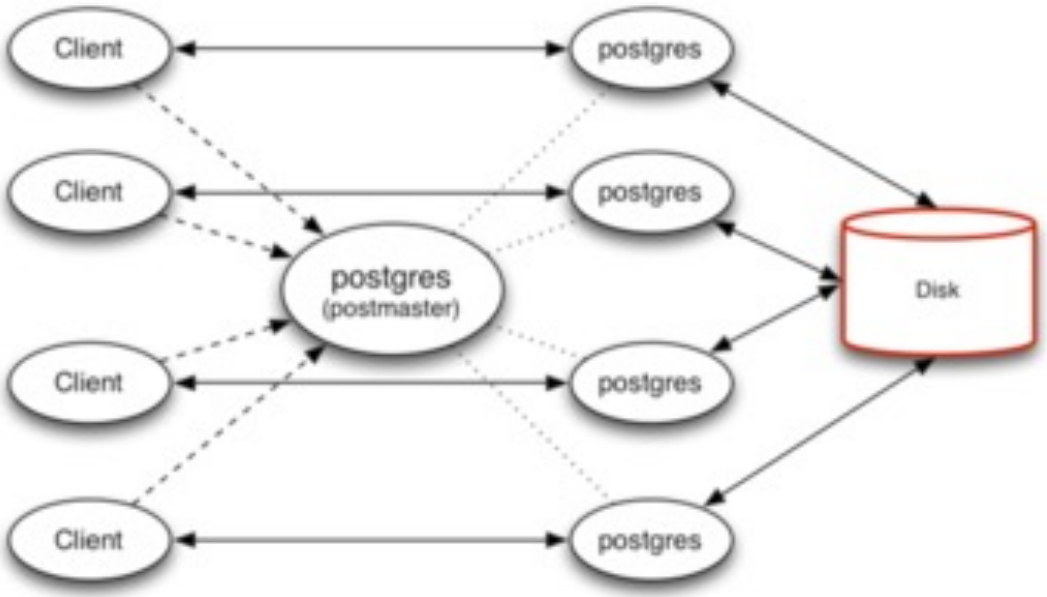
GÜZ -2022

Sorular

- Hareket (transaction) nedir?
- ACID properties ?
- WAL prensibi nedir ?
- Rollback ?

Eşzamanlılık (Concurrency) nedir ?

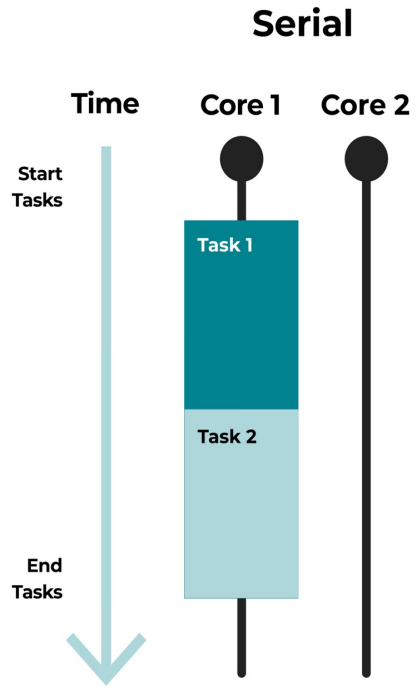
Veritabanı yönetim sistemlerinde **birden fazla hareketin** aynı zaman dilimi içerisinde gerçekleştirilmesinin koordinasyonunu eşzamanlılık (concurrency) ile sağlanır.



https://webcms3.cse.unsw.edu.au/COMP9315/16s1/resources/2352

Seri (sequential) –Paralel (parallel) – Eş zamanlı (Concurrent)

https://openclassrooms.com/en/courses/5684021-scale-up-your-code-with-java-concurrency/5684028-identify-the-advantages-of-concurrency-and-parallelism

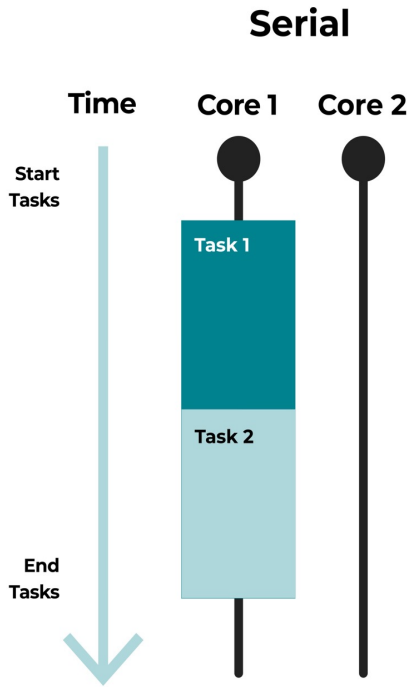


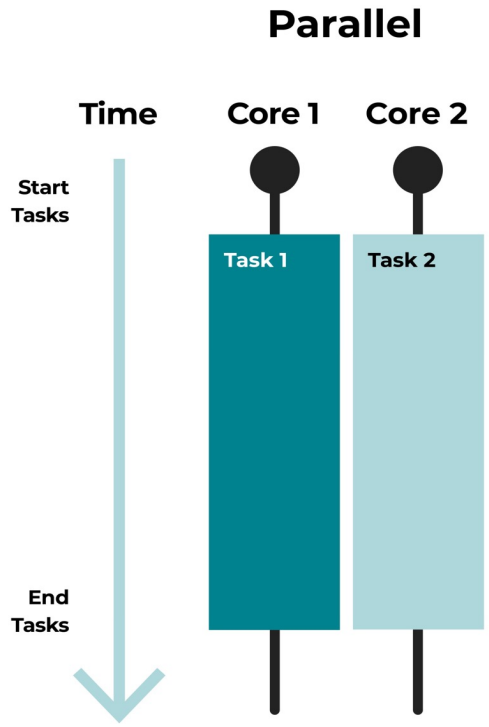
İşlemler sırasıyla gerçekleştirilir.

T1 tamamlanır sonrasında T2 başlar ve biter

Seri (sequential) –Paralel (parallel) – Eş zamanlı (Concurrent)

https://openclassrooms.com/en/courses/5684021-scale-up-your-code-with-java-concurrency/5684028-identify-the-advantages-of-concurrency-and-parallelism



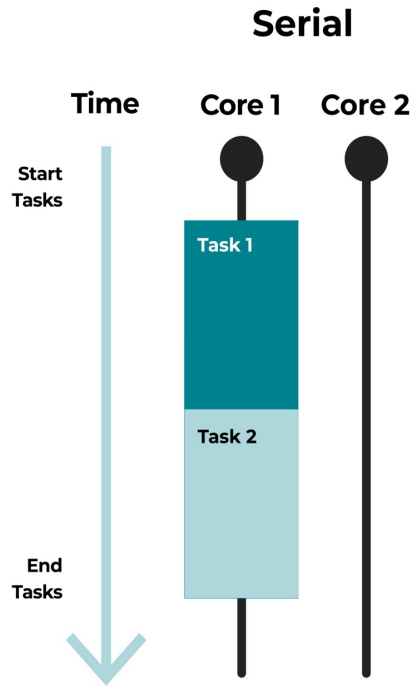


İşlemler sırasıyla gerçekleştirilir. T1 tamamlanır sonrasında T2 başlar ve biter

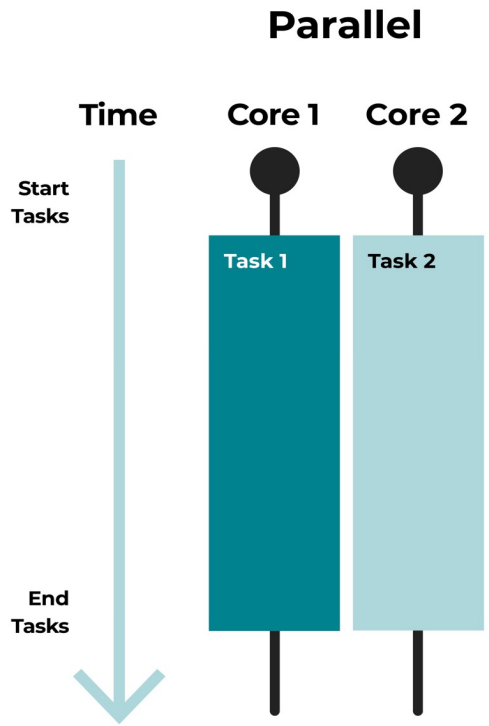
T1 ve T2 aynı anda farklı CPU core'ları ile gerçekleştirilir.

Seri (sequential) –Paralel (parallel) – Eş zamanlı (Concurrent)

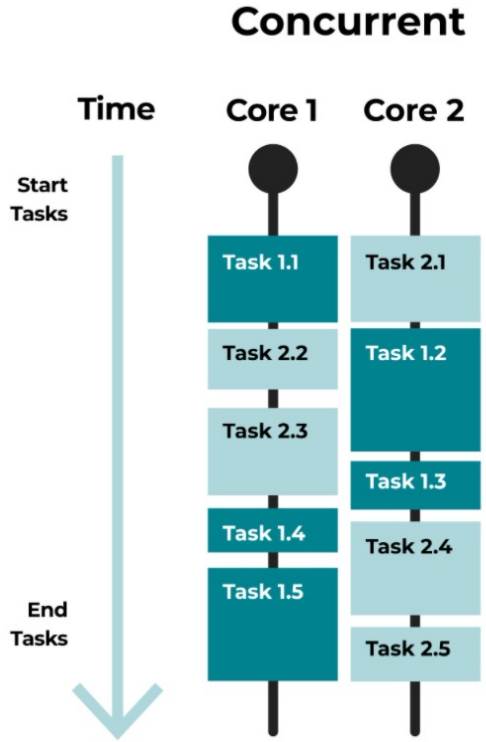
https://openclassrooms.com/en/courses/5684021-scale-up-your-code-with-java-concurrency/5684028-identify-the-advantages-of-concurrency-and-parallelism



İşlemler sırasıyla gerçekleştirilir. T1 tamamlanır sonrasında T2 başlar ve biter

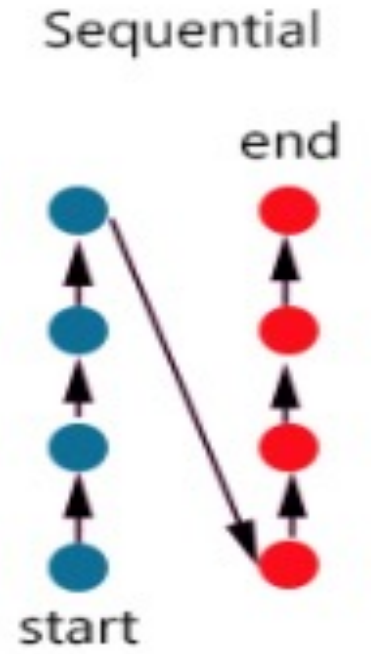


T1 ve T2 aynı anda farklı CPU core'ları ile gerçekleştirilir.

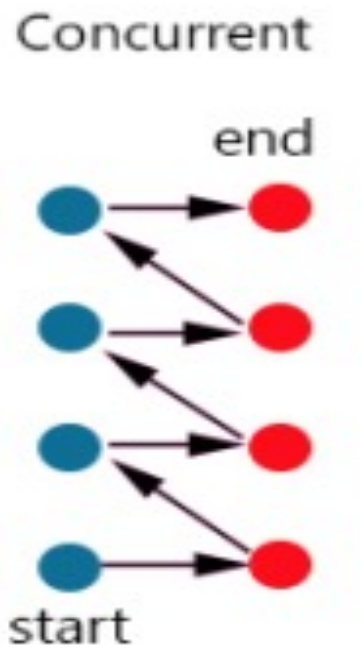


Alt aşamalara bölünebilen görevleri dönüşümlü olarak uygun olan işlemci çekirdeğinde eş zamanlı olarak çalıştırır.

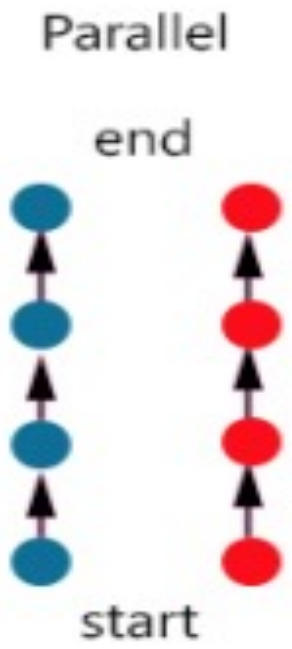
Seri (sequential) –Paralel (parallel) – Eş zamanlı (Concurrent)



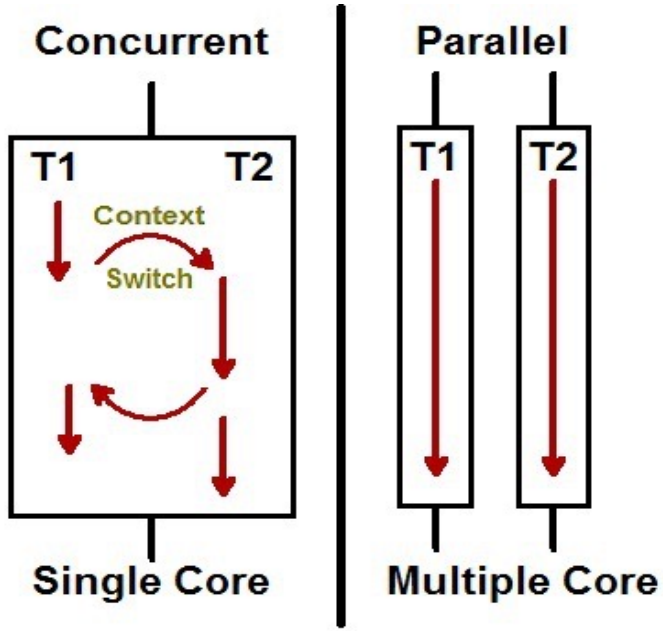
İşlemler sırasıyla gerçekleştirilir.



Dönüşümlü olarak işlemler gerçekleştirilir.

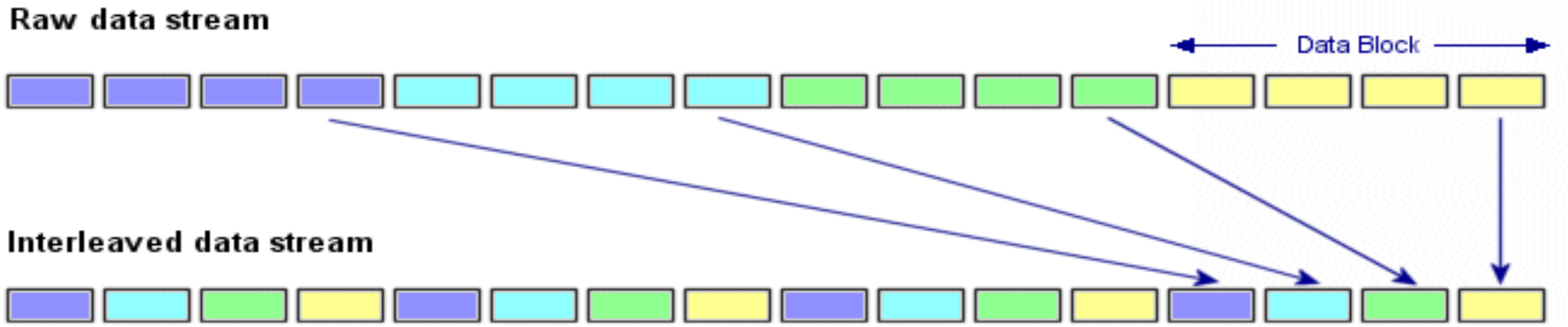


Aynı anda farklı işlemci çekirdeklerinde gerçekleştirilir



Eşzamanlılık (Concurrency) nedir ?

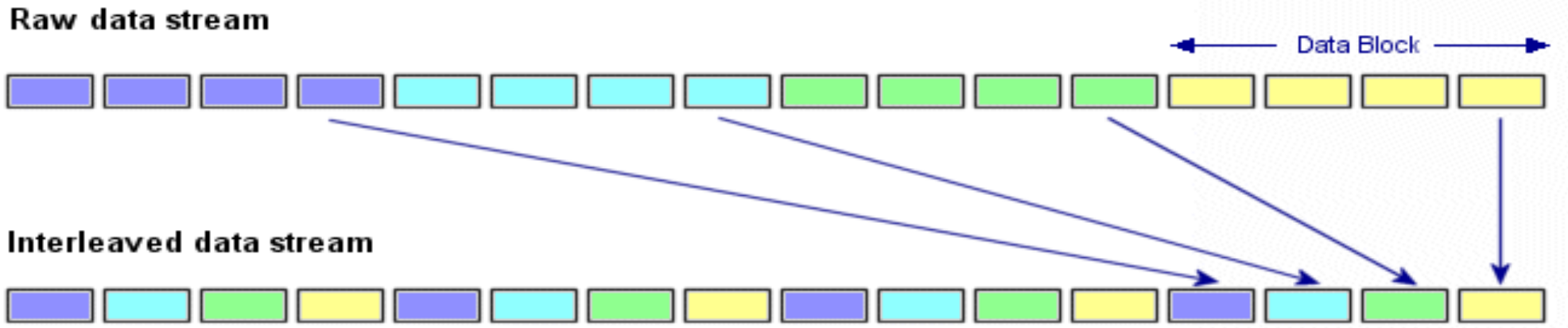
- Eşzamanlılık alt aşamalara bölünebilen problemleri sonucu değişmeyecek bir biçimde belirli bir sıraya uyulmaksızın gerçekleştirilmesine imkan sağlar.



https://kitz.co.uk/adsl/interleaving.htm

Eşzamanlılık (Concurrency) nedir ?

Eşzamanlılık alt aşamalara bölünebilen problemleri sonucu değişmeyecek bir biçimde belirli bir sıraya uyulmaksızın gerçekleştirilmesine imkan sağlar.



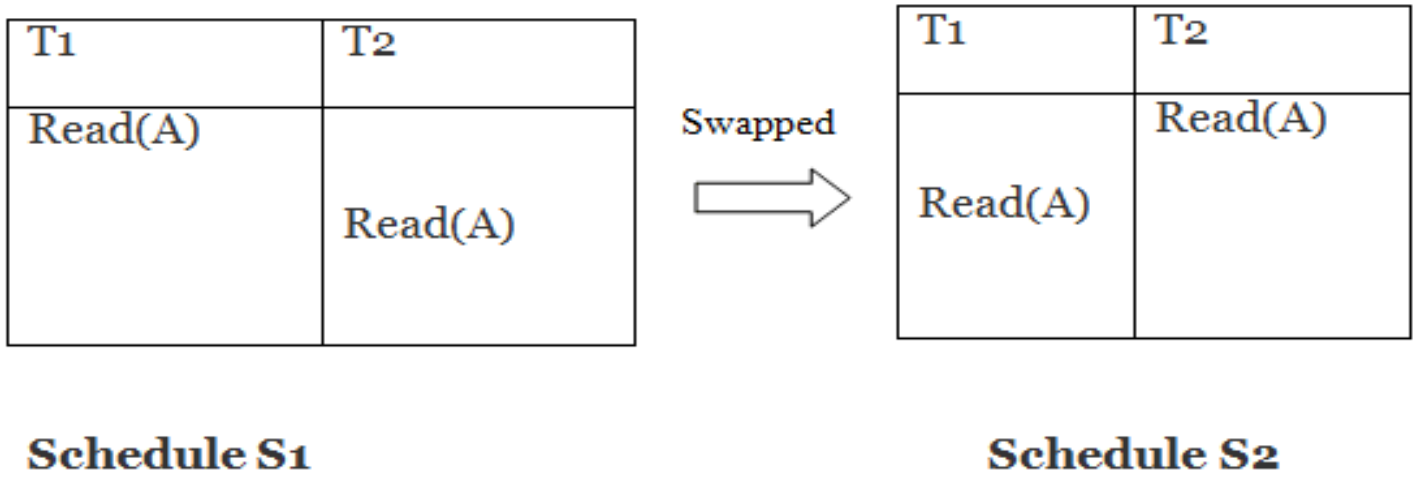
https://kitz.co.uk/adsl/interleaving.htm

İstenilen performası gerçekleştirmek için VTYS hareketler arasında interlaving (dönüşümlü çalıştırmak) tekniğini kullanmaktadır.

Eşzamanlılık (Concurrency)

Birden fazla hareket eş zamanlı (concurrent) çalıştırıldığında veritabanının tutarlılığına zarar verecek durumlar ortaya çıkabilir ve bu durumlar **çelişkiler (conflict)** olarak isimlendirilir.

1. T1: Read(A) T2: Read(A)



Schedule S1 **Schedule S2**

S1=S2 olduğundan bir çakışma söz konusu değil

https://www.javatpoint.com/dbms-conflict-serializable-schedule

Eşzamanlılık (Concurrency)

Birden fazla hareket eş zamanlı (concurrent) çalıştırıldığında veritabanının tutarlılığına zarar verecek durumlar ortaya çıkabilir ve bu durumlar **çelişkiler (conflict)** olarak isimlendirilir.

2. T1: Read(A) T2: Write(A)

T1	T2
Read(A)	
	Write(A)

Schedule S1

Swapped

T1	T2
	Write(A)
Read(A)	

Schedule S2

S1 != S2 olduğundan çakışma söz konusu

https://www.javatpoint.com/dbms-conflict-serializable-schedule

Eşzamanlılık (Concurrency)

Eşzamanlılık kontrolünün amacı, çok kullanıcılı bir veritabanı ortamında çakışmayı **(conflict)** engelleyip işlemlerin **sıralanabilirliğini (serializability)** sağlamaktır.

Non-serial schedule

T1	T2
Read(A)	
Write(A)	
	Read(A)
	Write(A)
Read(B)	
Write(B)	
	Read(B)
	Write(B)

Schedule S1

Serial Schedule

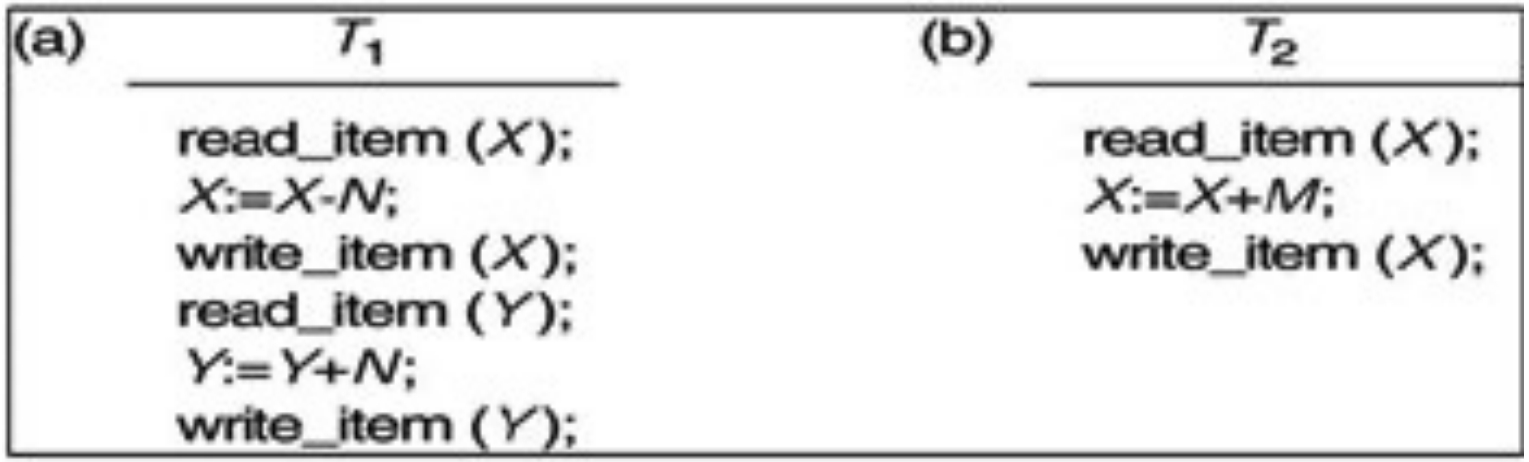
T1	T2
Read(A)	
Write(A)	
Read(B)	
Write(B)	
	Read(A)
	Write(A)
	Read(B)
	Write(B)

Schedule S2

https://www.javatpoint.com/dbms-conflict-serializable-schedule

Eşzamanlılık (Concurrency)

Concurrency (eşzamanlılık) kontrolü ile veritabanı yönetim sistemlerinin birden fazla hareket çalıştırırken ortaya çıkabilecek **sistem hatalarından korunmasını** (recovery from failure) sağlar.



(a) T_1

```
read_item (X);
X:=X-N;
write_item (X);
read_item (Y);
Y:=Y+N;
write_item (Y);
```

(b) T_2

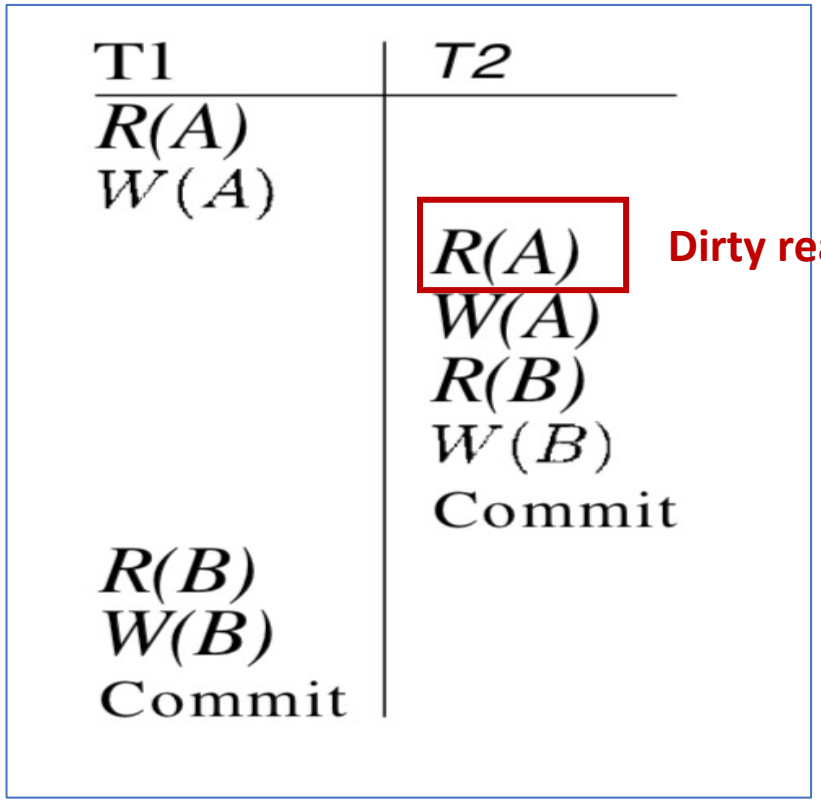
```
read_item (X);
X:=X+M;
write_item (X);
```

Birden fazla hareketin gerçekleştireceği işlemler (read, write, commit, abort) dizisine **schedule** (plan) denir

http://www.sakshieducation.com/Engineering/Story.aspx?cid=12&nid=101819

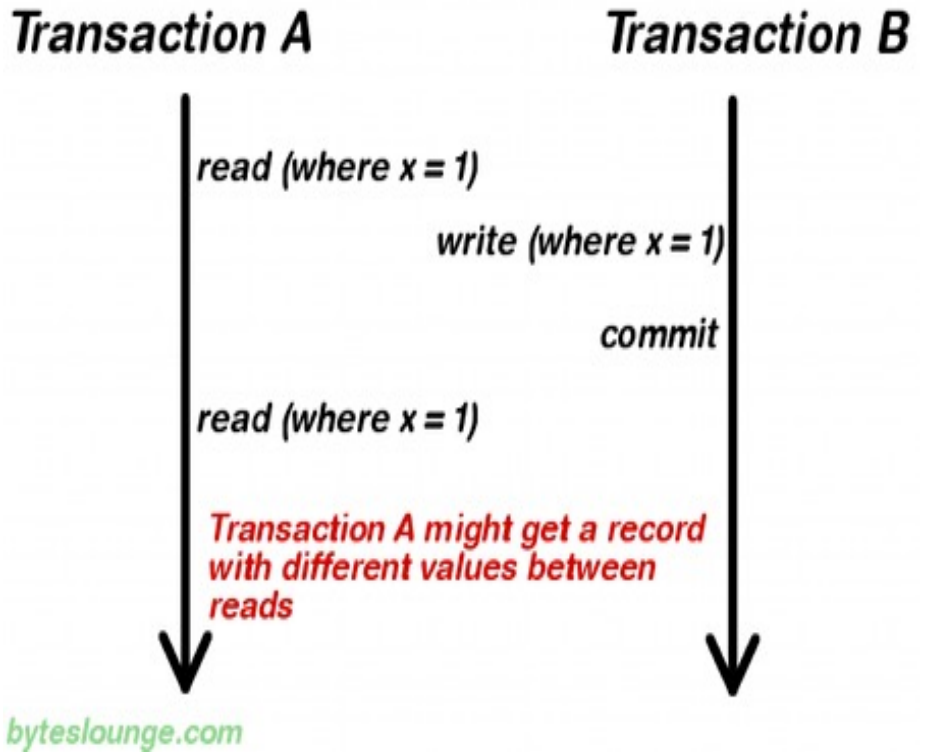
Reading uncommitted data (WR Conflict)

Bir hareket tarafından işlem yapılıp içeriği değiştirilen bir veriyi **commit** işlemi tamamlanmadan başka bir hareketin okuması

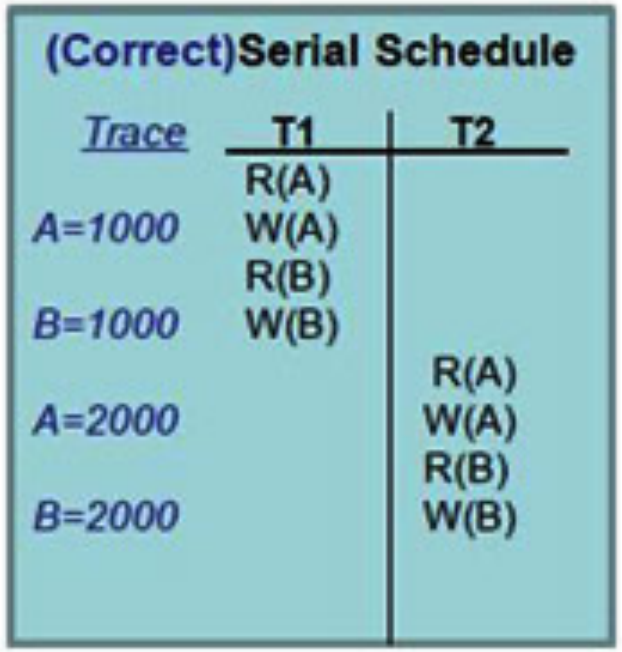


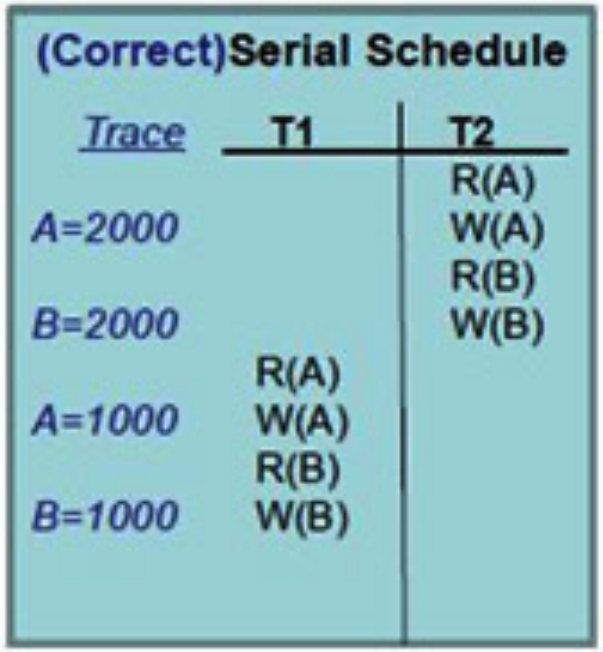
Unrepeatable Reads (RW Conflicts)

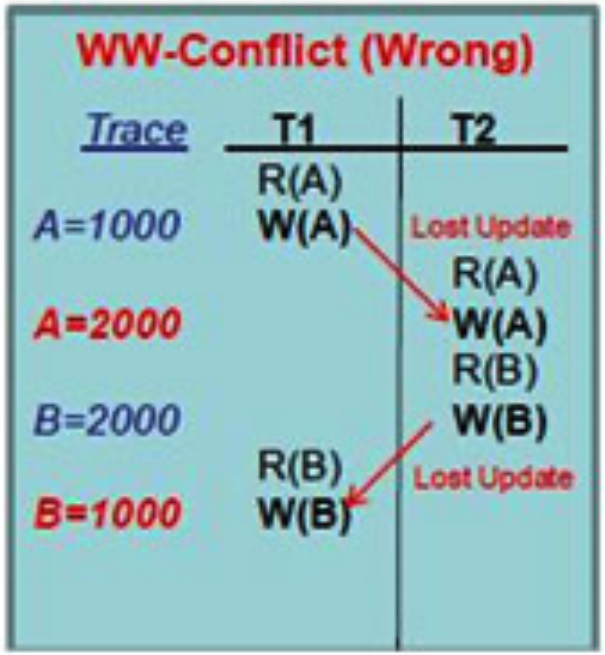
- TA'nın x değerini okuyup işlemi bitirmeden TB' nin x değerini değiştirmesi ve TA'nın tekrar x değerine erişim anında bir öncekinden faklı değer alması.
- Serial işlemlerde bu hata ile karşılaşılmaz.



Overwriting Uncommitted Data (WW Conflicts) : T1 tarafından değiştirilen ve commit işlemi gerçekleştirlemeyen A değerinin , T2 tarafından tekrar değiştirilmesi

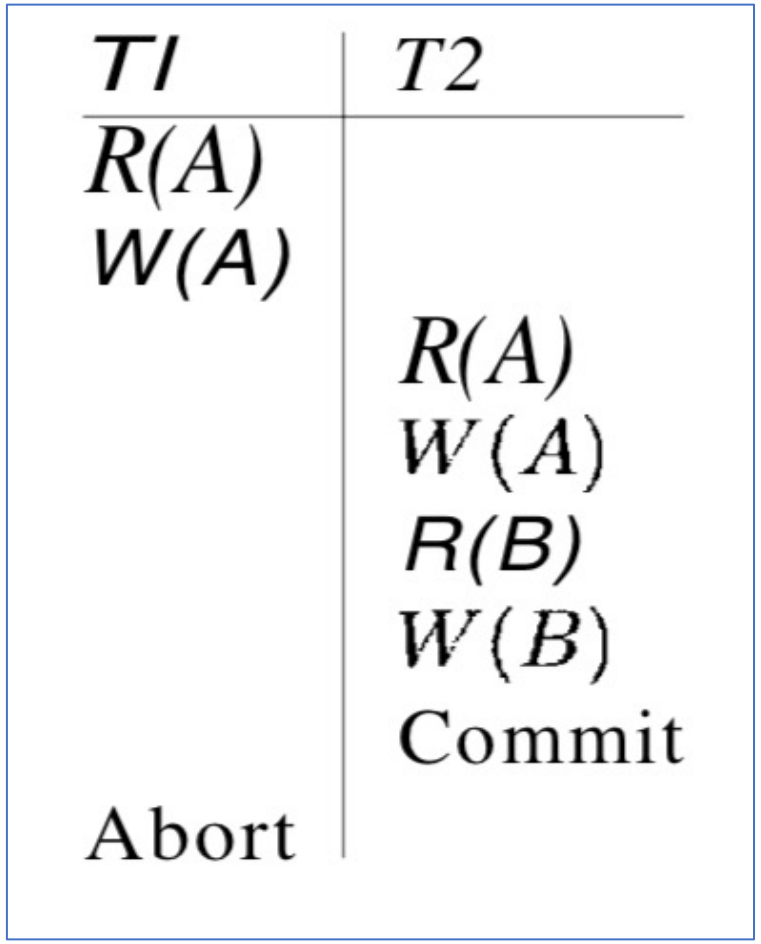






http://www.sakshieducation.com/Engineering/Story.aspx?cid=12&nid=101819

Eşzamanlılık (Concurrency) ve Problemler



Unrecoverable schedule
(telafi edilemeyen plan)

Anahtarlama Protokolleri (locking protocols)

- Bahsedilen problemlerin ortadan kaldırılması için Veritabanı Yönetim Sistemleri tarafından anahtarlama protokollerini (**locking protocol**) kullanılmaktadır.
- Locking protocol her bir transaction tarafından uyulması gereken kurallardır.
- Farklı anahtarlama protokolleri farklı anahtar kullanabilirler.

Anahtarlama Protokolleri (locking protocols)

Veritabanı yönetim sistemlerinde kullanılan anahtarlama protokolleri iki çeşit anahtar **(lock)** kullanılmaktadır:

1. **Paylaşılan anahtar** *(shared lock)*
2. **Dışlayıcı anahtar** (exclusive lock)

Anahtarlama Protokolleri (locking protocols)

1- Paylaşılan anahtar (S) *(shared lock)*

- Birden fazla uygulama bir veritabanı nesnesini aynı anda kullanabilir.
- Read (okuma)
- Bir hesap üzerinde okuma işlemi aynı anda gerçekleştirilebilir.
- Aynı hesap üzerinde yazma işlemi gerçekleştirmek için okuma işlemlerinin tamamlanması gerekmektedir.

Anahtarlama Protokolleri (locking protocols)

2- **Dışlayıcı anahtar (X)** (exclusive lock):

- Aynı anda sadece bir uygulamanın nesne üzerinde okuma ve yazma gerçekleştirmesine imkan sağlar
- Write- yazma
- Bir nesne üzerinde exclusive anhatar varsa bu anahtar sisteme teslim edilinceye kadar bu nesne uzerinde başka bir anahtar verilemez!

Anahtar Uyum Matrisi

	Shared	Exclusive
S	True	False
X	False	False

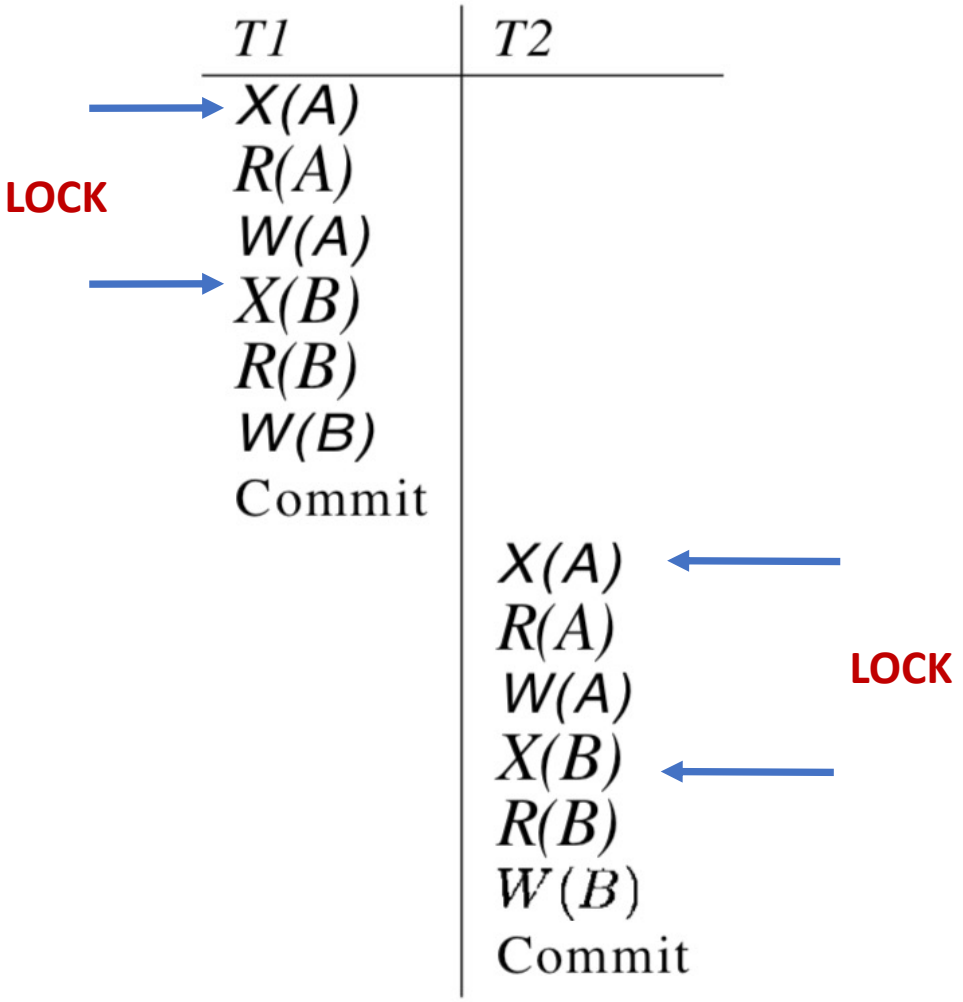
Bir nesne üzerinde aynı anda birden fazla sadece paylaşılan anahtar bulunabilir.

Anahtarlama Protokolleri (locking protocols)

Strict Two-Phase Locking (Strict 2PL)

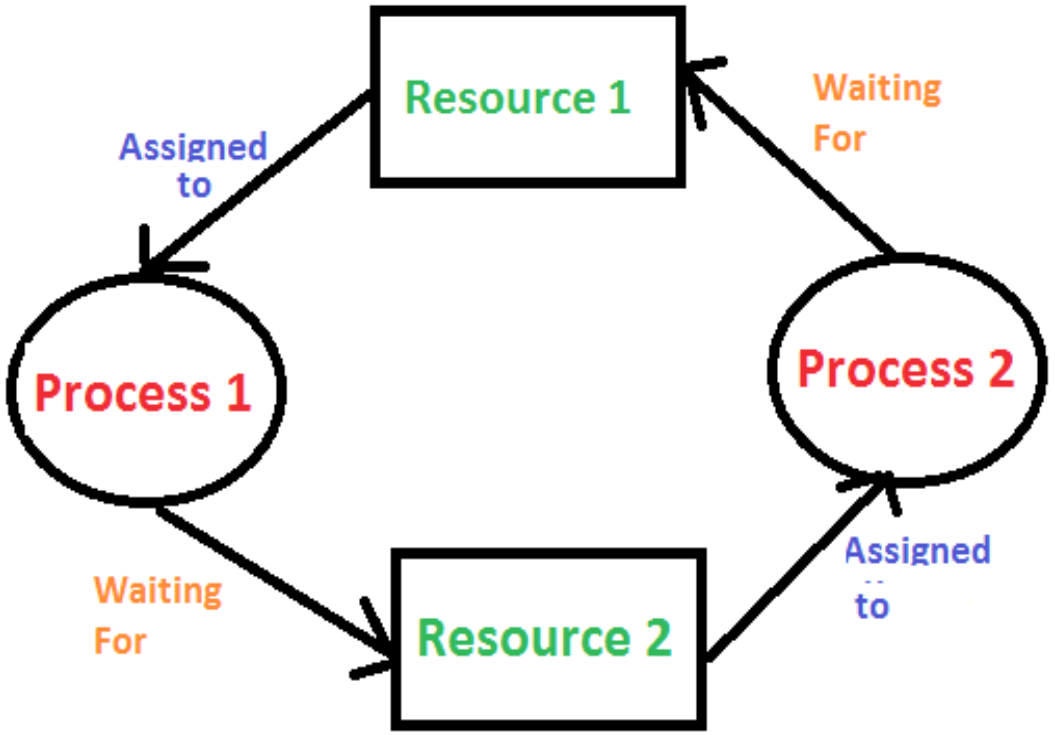
Bir nesne üzerinde

- okuma işlemi gerçekleştirecek olan bir hareket VTYS den bir **paylaşılan anahtar** (S:**shared lock**) istemektedir.
- yazma işlemi gerçekleştirecek olan bir hareket VTYS den bir **dışlayıcı anahtar** (X: exclusive lock) istemektedir.
- Nesne üzerinde işlem tamamlandığında bütün anahtarlar sisteme geri verilecektir.
- Bir nesne üzerindeki **exclusive lock** bırakılmadan o nesne üzerinde baska bir hareket işlem yapamaz.



Kilitlenme (Deadlock)

- Birden fazla hareketin birbirini beklemesiyle kilitlenme (deadlock) oluşur.
- Tamamlanması gereken işlemler bitmez ve işlem diğer kaynağa erişebilmek için bekler.
- Çizgi çizecek iki kişinin birinde kalem olup, diğerinde cetveli alıp birbirini beklemesidir.



Kilitlenme (Deadlock)

T_1	T_2
lock-X on A	
write (A)	
	lock-X on B
	write (B)
	wait for **lock-X** on A
wait for **lock-X** on B	

Deadlock durumunun ortadan kaldırılması için bir veya daha fazla işlemin iptal edilmesi (abort) gerekmektedir.

Kilitlenmenin (Deadlock) ortaya çıkmasının nedenleri

1. *Mutual Exclusion (Karşılıklı Dışlamak) :* bir kaynağın aynı anda birden fazla hareket tarafından kullanılamaması

2. *Hold and Wait (Bir kaynağı elde edip başka bir kaynağı beklemek):* Hareketlerin kullandıkları kaynaklar varken yeni kaynak tabelinde bulunması

Kilitlenmenin (Deadlock) ortaya çıkmasının nedenleri

3. *No Preemption (işlem üstünlüğü nün olmaması)* Hareketlerin kullandığı kaynakları başka bir hareketin zorla alamama durumu. Hareket istediği zaman kaynağı serbest bırakır.

4. *Circular Wait (Dairesel bekleme):* Birden fazla hareketin karşılıklı olarak sahip oldukları kaynakları beklemesidir.

Kilitlenme (Deadlock)

Deadlock'ın ortadan kaldırılması için :

1. *NO Mutual Exclusion (karşılıklı dışlamanın olmaması)*
2. *NO Hold and Wait Bir kaynağı sisteme geri iade etmeden başka kaynağın alınamaması*
3. *Preemption (işlem üstünlüğünün olması)*
4. *NO Circular Wait (Dairesel beklemenin olmaması)*

Banker Algoritması
Anahtarlama Yöntemleri
Bir veya birkaç hareketi durdur – rollback

Kilitlenme (Deadlock)

Veritabanı yönetim sistemlerinde oluşan hatalar

- WAL (Write Ahead Log) prensibi
- Checkpoint
- Rollback

kullanılarak ortadan kaldırılır.

Dinlediğiniz için

Teşekkürler...

İyi çalışmalar...