

# Veritabanı Yönetim Sistemleri

---

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Eşzamanlılık (concurrency)

- ☐ Hareket nedir?
- ☐ Hareketlerin yönetiminde karşılaşılabacak muhtemel problemler nelerdir?
- ☐ ACID properties ?
- ☐ WAL prensibi nedir ?
- ☐ Rollback ?
- ☐ Concurrency ?

- ❑ Veritabanı yönetim sistemleri aynı anda olan birden fazla hareketin (transaction) gerçekleştirilmesini yönetmektedirler.
- ❑ Birden fazla hareket eş zamanlı olarak (concurrent) çalıştırıldığında veritabanının tutarlılığına zarar verecek durumlar ortaya çıkabilir ve bu durumlar çelişkiler (conflict) olarak isimlendirilir.

Çelişkilerin etkilerinin ortadan kaldırılması, ve veritabanının tutarlı  
(consistency) durumunun ve verinin bütünlüğünün (data integrity)  
korunabilmesi için eşzamanlılık kullanılmaktadır.

Veritabanı nesnelerine eş zamanlı erişim problemlerini ortadan kaldırmak için kitleme-anahtarlama protokolleri (*locking protocols*) kullanılmaktadır.

Veritabanı yönetim sistemleri iki çeşit anahtar (*lock*) kullanılmaktadır:

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

## 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.

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

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

Shared Exclusive

			S		X	
	-----					
	S		True		False	
	-----					
	X		False		False	
	-----					

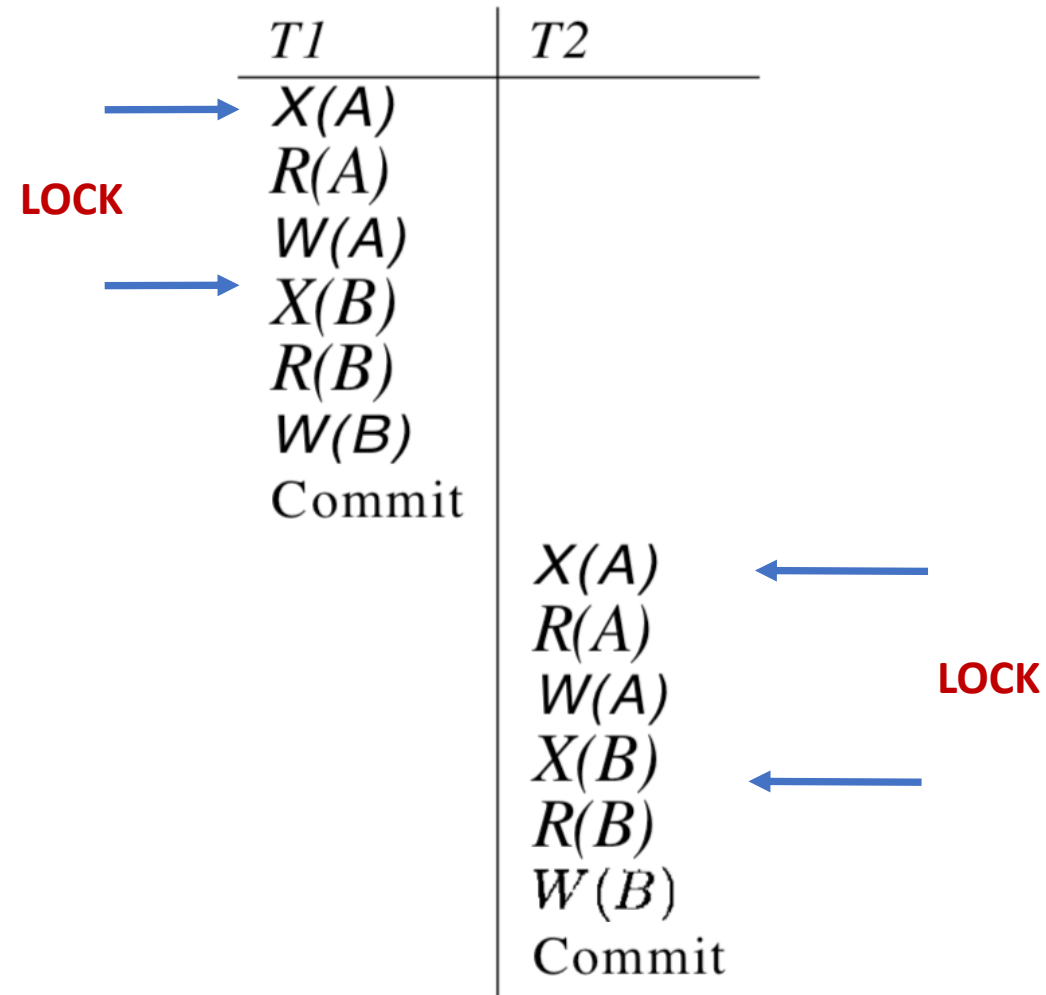
Bir nesne üzerinde  
aynı anda  
Sadece birden fazla  
paylaşılan anahtar bulunabilir.



Anahtarlama yöntemleri ile

- ❑ Serializability (Sıralanabilirlik)
- ❑ Recoverability (hataların geri alınabilmesi)

sağlanmaktadır



- Plan1: Hareket 1... Hareket n
- Plan2: Hareket 1... Hareket n

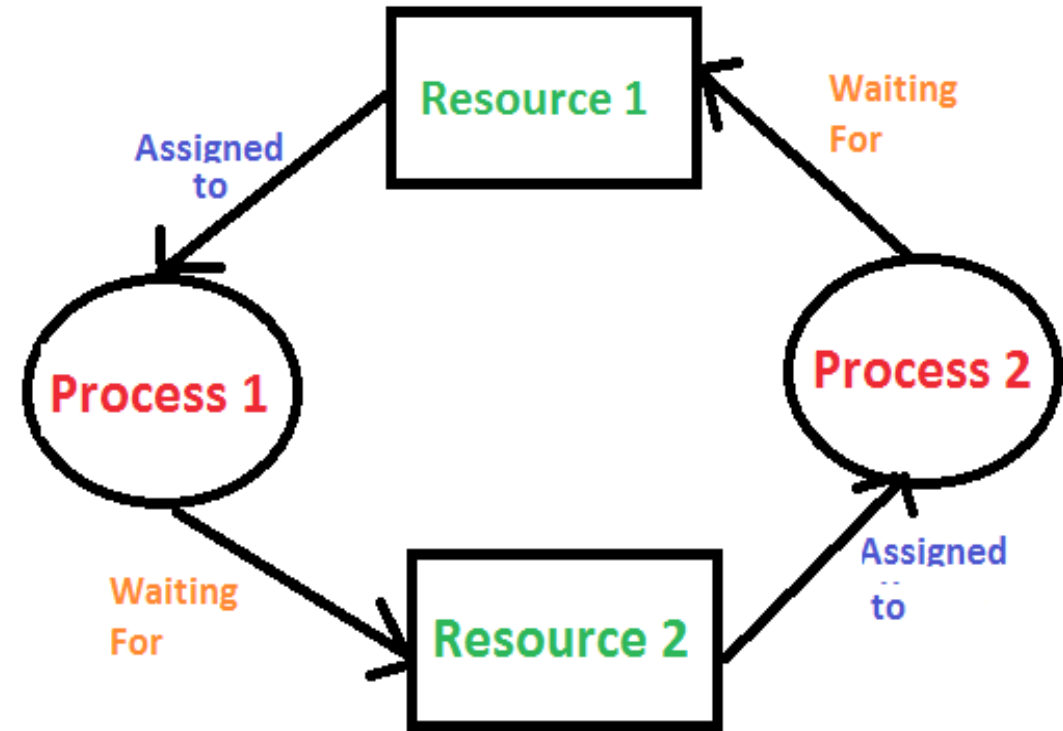
$$\text{Durum (Plan1)} = \text{Durum (Plan2)}$$

İki farklı hareket planının (schedule) sonucu farklı işlem sıralamasıyla ile birlikte aynı ise bu durum **conflict equivalent** olarak adlandırılır.

## Strict Two-Phase Locking (Strict 2PL)

- ❑ Bir nesne üzerinde yazma işlemi gerçekleştirecek olan bir hareket VTYS den bir **exclusive lock** almak zorundadır.
- ❑ Bir nesne üzerinde okuma işlemi için **ortak anahtar** (shared lock) gerekmektedir.
- ❑ Nesne üzerinde işlem tamamlandığında bütün anahtarlar sisteme geri verilecektir.
- ❑ Bir nesne üzerindeki exclusive lock bırakılmadan o nesne üzerinde başka bir hareket işlem yapamaz.

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



$T_1$	$T_2$
<b>lock-X</b> on A write (A)	<b>lock-X</b> on B write (B)
wait for <b>lock-X</b> on B	wait for <b>lock-X</b> on A

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

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

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.



Deadlock'in 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

Veritabanı yönetim sistemlerinde oluşan hatalar

- ❑ WAL (Write Ahead Log) prensibi
  - ❑ Checkpoint
  - ❑ Rollback
- kullanılarak ortadan kaldırılır.