

Veritabanı Yönetim Sistemleri (335)

Yrd.Doç.Dr. Ahmet Arif AYDIN



Transaction Management
Hareket Yönetimi

Transaction Management

Hareket (transaction)

- Kullanıcı programın veritabanı içerisinde gerçekleştirmiş olduğu **CRUD işlemlerinin (execution) her biri hareket** olarak isimlendirilir.

1. Banka Hesabını control
2. istenen miktar yeterli ise hesaptan düş
3. Banka hesabını güncelle yeni miktar ile
4. Arkadaşının hesabını kontrol et
5. Arkadaşının hesabına istenilen miktarı gönder
6. Arkadaşının hesabını güncelle

Para Transferi

Transaction Management

Hareket (transaction)

- **Concurrency** (eşzamanlı çalışma) ve veritabanının **sistem hatalarından korunmasının** (recovery from failure) temelini oluşturmaktadır.

Transaction Management

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

Raw data stream

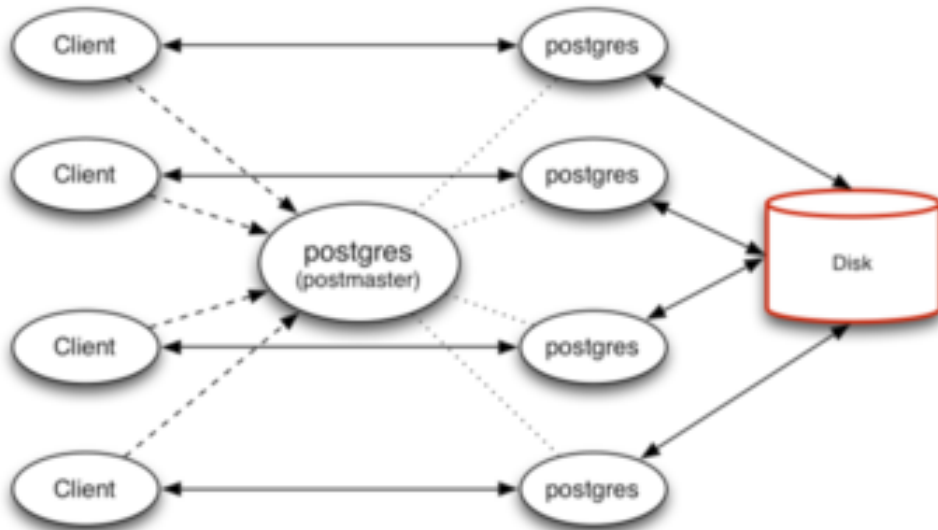


Interleaved data stream



Transaction Management

Veritabanı Yönetim Sistemleri eş zamanlı olarak gerçekleştirilen hareketlerin yönetimini ve kontrolünü sağlamaktadır. (Concurrency control)



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

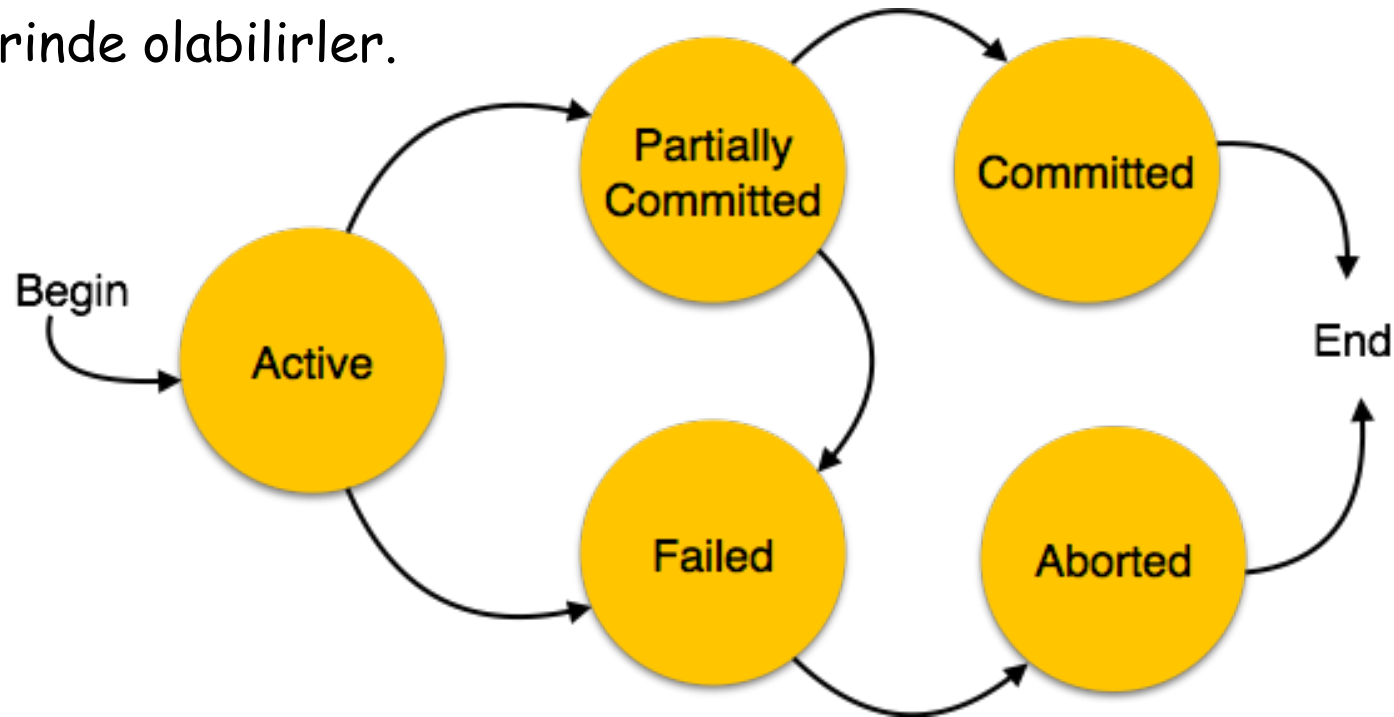


Transaction Management

Tamamlanamayan hareketler (Partial- incomplete transactions) VTYS tarafından idare edilmektedir.

Transaction Management

VTYS'lerinin yönettiği hareketlerin aşağıda belirtilen durumlardan (states of transactions) birinde olabilirler.



Transaction Management

Hareketlerin dört temel özelliği bulunmaktadır.

Atomicity (bölünemezlik)

Consistency (tutarlılık)

Isolation (yalıtım)

Durability (dayanıklılık)

ACID

Properties

ACID

Atomicity (bölünemezlik)

- Bir hareketin tamamlanabilmesi için kümelenmiş işlemlerin bölünmezliği.
- İşlemin bir kısmı gerçekleştirilmez hepsi başlar biter veya işlem gerçekleştirilmez
- Tamamlanamayan veya yarıda kalan hareketlerin işlemleri veritabanı tarafından geri alınır.

ACID

Atomicity (bölünmezlik)

K hesabında 400tl bulunmaktadır

T hesabında 700tl bulunmaktadır.

- K dan T ye 200 tl gönderilmesi gerekmektedir.
- Bu transfer iki aşamada gerçekleşmektedir.
 1. K hesabından 200tl düşülecek
 2. T hesabına 200 tl eklenecek

1. Başarılı 2. başarısız olursa

K (200tl) ve T (700tl) olarak kalacak

ACID

Consistency (tutarlılık)

- Veritabanı üzerinde gerçekleştirilen hareketlerde veritabanında bulunan verinin tutarlılığının korunmasına **consistency** denir.
- VTYS yapılan her işlemde tutarlılığın sağlandığını varsaymaktadır
- İlişkisel bütünlük (relational integrity) korunmaktadır.

Hesaplar arasında transfer yapılırken toplam miktar aynı kalmalıdır

ACID

isolation (yalıtım)

VTYS'lerinde birden fazla hareket yönetilirken bir hareketin diğer bir hareketten izole edilmesine yalıtım denir.

Bir hesaptan aynı anda (T1 ve T2) para çekildiğinde önce başlanılan işlem (T1) tamamlanmadan diğerinin (T2) tamamlanması beklenir.

ACID

Durability (dayanıklılık)

- Sistem hataları ortaya çıktığında veya hareketlerde problemler ile karşılaşıldığında veritabanının hareket bilgilerini kaybetmeden ve veritabanında kayıtlı olan veriyi kaybetmeden işleme devam edebilmesi **durability** olarak tanımlanır.

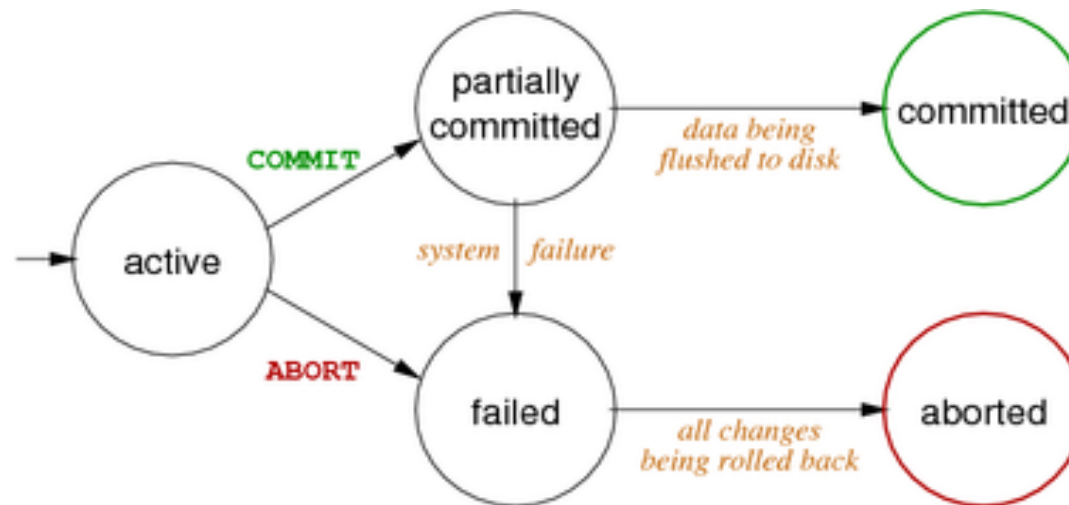
Transaction Management

Bir banka hesabından alınan parayı başka bir hesaba para göndermeden ortaya çıkan sistem hatasını **VTYS sistem hatası giderildikten sonra düzeltmek zorundadır.**

- VTYS gerçekleştirilen her işlemi log dosyalarına (hard diske)
WAL (Write-Ahead Log) prensibini kullanarak yazmaktadır.
- Hataların düzeltilmesi işlemi VTYS tarafından periyodik olarak disk üzerindeki log dosyaları okunarak yapılır. Bu işlem **kontrol noktası (checkpoint)** olarak adlandırılır

Transaction Management

- Bir hareket içerisindeki bütün işlemler başarıyla tamamlanmışsa bu işleme committed (teslim etmek) denir. Yapılan bütün değişiklikler diske kaydedilir.
- Bir hareket içerisinde herhangi bir hata oluştuğunda yapılan bütün değişiklikler geri alınarak hareket başlamadan önceki duruma geri dönülmesine rollback (geri dönüş) denir.



Transaction Management

Hareket planı (transaction schedule)

- Birden fazla hareketin gerçekleştireceği işlemler (read, write, commit, abort) listesine **schedule** (plan) denir.
- Schedule VTYS tarafından yönetilir.

$T1$	$T2$
$R(A)$	
$W(A)$	
	$R(B)$
	$W(B)$
$R(C)$	
$W(C)$	

Bu hareket planında transaction management hangi özellikleri görebiliyoruz?

Transaction Management

Hareket planı (transaction schedule)

- Birden fazla hareketin gerçekleştireceği işlemler (read, write, commit, abort) listesine **schedule** (plan) denir.
- Schedule VTYS tarafından yönetilir.

$T1$	$T2$
$R(A)$	
$W(A)$	
	$R(B)$
	$W(B)$
$R(C)$	
$W(C)$	

Bu hareket planında transaction management hangi özellikleri görebiliyoruz?

- Interleaving
- Isolation
- Atomicity

Transaction Management

- Birden fazla işlemin aynı zaman dilimi içerisinde gerçekleştirilmesine **concurrent execution (eşzamanlı çalışma)** denir.
- **Throughput** (üretilen iş) : Verilen zaman dilimi içerisinde gerçekleştirilen ortalama işlem sayısı
- Aynı zaman dilimi içerisinde işlem performansını arttırmak için **concurrency** kullanılır.
- Concurrent olmayan işlemlerde kısa süreli bir transaction uzun zaman alan bir transaction ın ardında ise ne zaman biteceği tam olarak kestirilemeyebilir.

Transaction Management

$T1$	$T2$
$R(A)$ $W(A)$	
	$R(A)$ $W(A)$
$R(B)$ $W(B)$	
	$R(B)$ $W(B)$ Commit
Commit	

**Serializable
Schedule**

$T1$	$T2$
	$R(A)$ $W(A)$
$R(A)$	
	$R(B)$ $W(B)$
$W(A)$ $R(B)$ $W(B)$	
	Commit
Commit	

**sıra ile işlem
Yapılabilen plan**

Transaction Management Problems

Reading uncommitted data (WR Conflict)

Bir hareket tarafından işlem yapıp içeriği değiştirilen bir veriyi **commit** işlemi tamamlanmadan okumak.

T1	T2
$R(A)$ $W(A)$	$R(A)$ $W(A)$ $R(B)$ $W(B)$ Commit
$R(B)$ $W(B)$ Commit	

Transaction Management Problems

Unrepeatable Reads (RW Conflicts)

- T1 A değerini okuyup üzerindeki işlemi bitirmeden T2 nin A değerini değiştirmesi ve T1 in tekrar A değerine erişim anında bir öncekinden farklı değer alması.
- Serial işlemlerde bu hata ile karşılaşılmaz.

Transaction Management Problems

Overwriting Uncommitted Data (WW Conflicts)

- T1 tarafından değiştirilen ve commit işlemi gerçekleştirilmeyen A değerinin , T2 tarafından tekrar değiştirilmesi

Transaction Management Problems

$T1$	$T2$
$R(A)$ $W(A)$	$R(A)$ $W(A)$ $R(B)$ $W(B)$ Commit
Abort	

Unrecoverable schedule

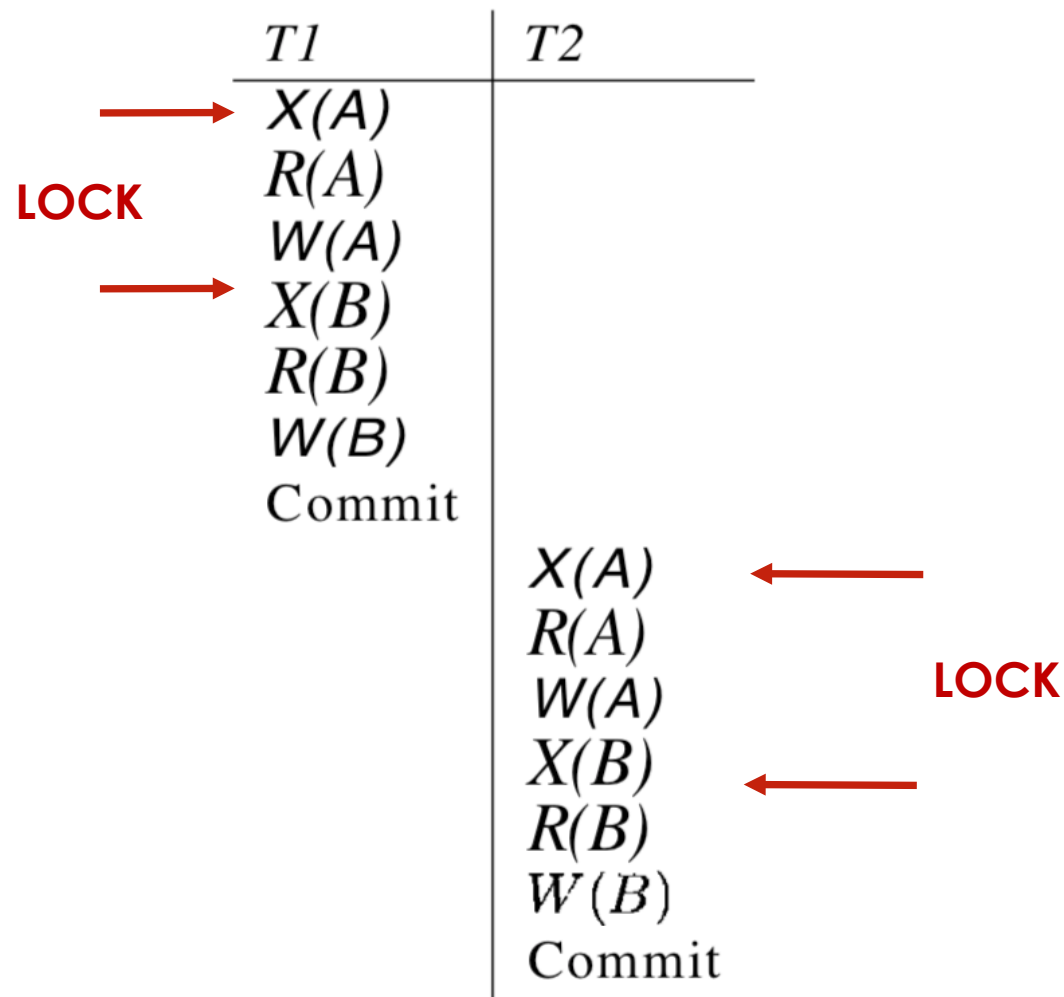
Lock-Based Concurrency Control

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

Lock-Based Concurrency Control

Strict Two-Phase Locking (Strict 2PL)

1. Bir nesne üzerinde okuma veya yazma işlemi gerçekleştirecek olan bir hareket VTYS den bir ortak anahtar (shared lock) istemektedir.
2. Nesne üzerinde işlem tamamlandığında bütün anahtarlar sisteme geri verilecektir.



Postgres Transaction

BEGIN;

UPDATE accounts SET balance = balance - 100.00 WHERE name = 'Alice';

SAVEPOINT my_savepoint;

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Bob';
-- oops ... forget that and use Wally's account

ROLLBACK TO my_savepoint;

UPDATE accounts SET balance = balance + 100.00 WHERE name = 'Wally';

COMMIT;