

Veri Tabanı Sistemleri

Eşzamanlı Çalışma

H. Turgut Uyar Şule Öğüdücü

2002-2012

1 / 45

License



©2002-2012 T. Uyar, Ş. Öğüdücü

You are free:

- ▶ to Share – to copy, distribute and transmit the work
- ▶ to Remix – to adapt the work

Under the following conditions:

- ▶ Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- ▶ Noncommercial – You may not use this work for commercial purposes.
- ▶ Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Legal code (the full license):

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2 / 45

Konular

Hareket Yönetimi

Giriş
Sistemin Düzeltilmesi
İki Aşamalı Sonlandırma

Eşzamanlı Çalışma

Giriş
Kilitleme
Yalıtım Düzeyleri
Niyet Kilitleri

3 / 45

Hareket Yönetimi

- ▶ birden fazla işlemin topluca yapılması gerekebilir
 - ▶ bir işlemin yapılıp diğerlerinin yapılmaması tutarsızlık yaratabilir
- ▶ birden fazla işlemin topluca yapılması garanti edilemez
 - ▶ en azından değişikliklerden önceki duruma dönülebilmeli

Tanım

hareket: bir işin mantıksal bir birimi

4 / 45

Hareket Örneği

Örnek (bir banka hesabından diğerine para aktarma)

```
UPDATE ACCOUNTS SET BALANCE = BALANCE - 100
WHERE ACCOUNTID = 123
```

```
UPDATE ACCOUNTS SET BALANCE = BALANCE + 100
WHERE ACCOUNTID = 456
```

5 / 45

Hareket Özellikleri

- ▶ bölünmezlik
 - ▶ ya tam yapılır, ya hiç yapılmaz
- ▶ tutarlılık
 - ▶ bir tutarlı durumdan diğer bir tutarlı duruma geçiş
- ▶ yalıtım
 - ▶ sona ermemiş bir hareketin işlemlerinin diğer hareketleri etkileyip etkilemediği
- ▶ kalıcılık
 - ▶ bir hareket sonlandırıldıktan sonra sistem çökse de verilerin zarar görmemesi

6 / 45

Hareket İşlemleri

başlatma

```
BEGIN [ WORK | TRANSACTION ]
```

sonlandırma

```
COMMIT [ WORK | TRANSACTION ]
```

vazgeçme

```
ROLLBACK [ WORK | TRANSACTION ]
```

7 / 45

Hareket Örneği

Örnek

```
BEGIN TRANSACTION
ON ERROR GOTO UNDO
UPDATE ACCOUNTS SET BALANCE = BALANCE - 100
  WHERE (ACCOUNTID = 123)
UPDATE ACCOUNTS SET BALANCE = BALANCE + 100
  WHERE (ACCOUNTID = 456)
COMMIT
...
```

```
UNDO :
ROLLBACK
```

8 / 45

Sistemin Düzeltilmesi

- ▶ bir hareket sürerken sistemin çöktüğünü düşünün
 - ▶ bellek tamponlarındaki veriler diske yazılmamış durumda
- ▶ kalıcılık nasıl sağlanacak?
 - ▶ veri, sistemde başka yerde yazılı verilerden türetilmeli
 - ▶ iç düzeyde

9 / 45

Günlük

- ▶ **günlük** her işlemten etkilenen her çoklunun işlemten önceki ve sonraki değerlerini tutar
- ▶ **günlüğe önceden yazma kuralı:** hareket sonlanmadan önce günlük fiziksel ortama yazılmalı
- ▶ günlük kayıtlarına erişim işlemin doğası gereği ardışıl

10 / 45

Denetim Noktaları

- ▶ belli aralıklarla günlükte **denetim noktaları** oluşturulur
- ▶ bellek tamponlarındaki veriler fiziksel ortama yazılır
- ▶ denetim noktası günlüğe not edilir
- ▶ o an sürmekte olan hareketler not edilir

11 / 45

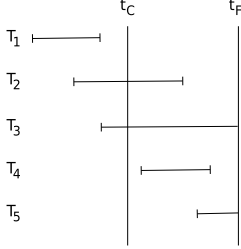
Düzeltilme Listeleri

- ▶ aksaklıktan sonra hangi hareketler geri alınacak, hangileri sonlandırılacak?
 - ▶ iki liste oluştur: *geri alınacaklar* (G), *yineleneceler* (Y)
- ▶ t_C : günlükte kayıtlı son denetim noktası
 - ▶ t_C anında etkin olan hareketleri G'ye ekle
- ▶ t_C 'den başlayarak kayıtları ileri doğru tara
 - ▶ başlayan bir hareketle karşılaşsan G'ye ekle
 - ▶ biten bir hareketle karşılaşsan Y'ye geçir

12 / 45

Düzeltilme Örneği

Örnek



- t_C :
 $G = \{T_2, T_3\}$ $Y = \emptyset$
- T_4 başladı:
 $G = \{T_2, T_3, T_4\}$ $Y = \emptyset$
- T_2 bitti:
 $G = \{T_3, T_4\}$ $Y = \{T_2\}$
- T_5 başladı:
 $G = \{T_3, T_4, T_5\}$ $Y = \{T_2\}$
- T_4 bitti:
 $G = \{T_3, T_5\}$ $Y = \{T_2, T_4\}$

13 / 45

Düzeltilme Süreci

- kayıtları günlük sonundan geriye doğru tara
 - G'deki hareketlerin yaptıkları değişiklikleri geri al
- kayıtları ileriye doğru tara
 - Y'deki hareketlerin yaptıkları değişiklikleri yinele

14 / 45

İki Aşamalı Sonlandırma

- farklı kaynak yöneticileri var
 - geri alma - sonlandırma sistemleri ayrı
- etkilenecek veriler farklı kaynak yöneticilerinde
 - ya hepsinde birden sonlandırılacak
 - ya da hepsinde birden geri alınacak
- eşgüdüm sağlayıcı

15 / 45

Protokol

- eşgüdüm sağlayıcı, bütün katılımcılara hareketle ilgili bütün verilerin kayıtlarını kalıcı ortama yazmalarını söyler
- eşgüdüm sağlayıcı, bütün katılımcılardan hareketi başlatmalarını ve sonucu kendisine bildirmelerini ister
 - bütün katılımcılardan "başarılı" yanıtı alırsa hareketin sonlandırılmasına karar verir
 - bir tane bile "başarısız" yanıtı gelirse hareketin geri alınmasına karar verir
- eşgüdüm sağlayıcı, bütün katılımcılara kararı bildirir

16 / 45

Kaynaklar

Okunacak: Date

- Chapter 15: Recovery

17 / 45

Eşzamanlı Çalışma

- eşzamanlı çalışan hareketler nedeniyle çıkabilecek sorunlar:
- yitirilen güncelleme
- kesinleşmemiş veriye bağımlılık
- tutarsız çözümleme

18 / 45

Yitirilen Güncelleme

Örnek

Hareket A	Hareket B
...	...
RETRIEVE p	...
...	...
...	RETRIEVE p
...	...
UPDATE p	...
...	...
...	UPDATE p
...	...

19 / 45

Kesinleşmemiş Veriye Bağımlılık

Örnek

Hareket A	Hareket B
...	...
...	UPDATE p
...	...
RETRIEVE p	...
...	...
...	ROLLBACK
...	...

20 / 45

Tutarsız Çözümleme

Örnek (hesap toplamı: acc1=40, acc2=50, acc3=30)

Hareket A	Hareket B
...	...
RETRIEVE acc1 (40)	...
RETRIEVE acc2 (90)	...
...	...
...	UPDATE acc3 (30 → 20)
...	UPDATE acc1 (40 → 50)
...	COMMIT
...	...
RETRIEVE acc3 (110)	...
...	...

21 / 45

Çakışmalar

- ▶ A okuyor, B okuyor
 - ▶ sorun yok
- ▶ A okuyor, B yazıyor
 - ▶ yinelenemez okuma (tutarsız çözümleme)
- ▶ A yazıyor, B okuyor
 - ▶ kirli okuma (kesinleşmemiş veriye bağımlılık)
- ▶ A yazıyor, B yazıyor
 - ▶ kirli yazma (yitirilen güncelleme)

22 / 45

Kilitleme

- ▶ hareketler üzerinde işlem yapacakları çokluları kilitlesinler
 - ▶ okuma kilidi (S)
 - ▶ yazma kilidi (X)
- ▶ işleri bitince kilitleri bıraksınlar

23 / 45

Kilit İstekleri

kilit tipi uyumluluk matrisi

	X	S	-
X	H	H	E
S	H	E	E

- ▶ yazma kilidi varsa başka hareketlerin her türlü isteği reddedilir
- ▶ okuma kilidi varsa:
 - ▶ başka hareketlerin yazma kilidi istekleri reddedilir
 - ▶ başka hareketlerin okuma kilidi istekleri kabul edilir

24 / 45

Kilitleme Protokolü

- ▶ hareket, yapmak istediği işleme göre kilit isteğinde bulunur
 - ▶ okuma kilidi varsa yazma kilidine çevrilmesi
- ▶ istek yerine getirilemiyorsa beklemeye başlar
 - ▶ diğer hareket kilidi bırakınca devam eder
 - ▶ **sonsuz bekleme**

25 / 45

Yitirilen Güncelleme

Örnek

Hareket A	Hareket B
...	...
RETRIEVE p (S+)	...
...	...
...	RETRIEVE p (S+)
...	...
UPDATE p (X-)	...
bekle	...
bekle	UPDATE p (X-)
bekle	bekle

26 / 45

Kesinleşmemiş Veriye Bağımlılık

Örnek

Hareket A	Hareket B
...	...
...	UPDATE p (X+)
...	...
RETRIEVE p (S-)	...
bekle	...
bekle	ROLLBACK
RETRIEVE p (S+)	...
...	...

27 / 45

Tutarsız Çözümleme

Örnek (hesap toplamı: acc1=40, acc2=50, acc3=30)

Hareket A	Hareket B
...	...
RETRIEVE acc1 (S+)	...
RETRIEVE acc2 (S+)	...
...	...
...	UPDATE acc3 (X+)
...	UPDATE acc1 (X-)
...	bekle
RETRIEVE acc3 (S-)	bekle
bekle	bekle

28 / 45

Ölümçül Kilitlenme

Tanım

ölümçül kilitlenme:

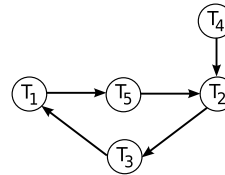
hareketlerin birbirlerinin kilitleri bırakmalarını beklemesi

- ▶ neredeyse her zaman iki hareket arasında
- ▶ yapılabilecekler:
 - ▶ farketmek ve çözmek
 - ▶ önlemek

29 / 45

Ölümçül Kilitlenmenin Çözülmesi

Örnek



- ▶ bekleme grafi
- ▶ bir **kurban** seç ve öldür

30 / 45

Ölümcül Kilitlenmenin Önlenmesi

- ▶ her hareketin başlama zamanı mührü var
- ▶ A hareketinin kilit isteği
B hareketinin tuttuğu bir kilitte çelişiyorsa:
 - ▶ **bekle-öl**: A, B'den yaşlıysa bekler, gençse ölür
A geri alınıp yeniden başlatılır
 - ▶ **yarala-bekle**: A, B'den gençse bekler, yaşlıysa B'yi yaralar
B geri alınıp yeniden başlatılır
- ▶ yeniden başlatılan hareketin zaman mührü değiştirilmez

31 / 45

Kilit Komutları

okuma kilidi

`SELECT query FOR SHARE`

yazma kilidi

`SELECT query FOR UPDATE`

32 / 45

Yalıtım Düzeyleri

- ▶ yalıtım azaltılırsa eşzamanlılık artırılabilir
- ▶ değişik yalıtım düzeyleri:
- ▶ serileştirilebilir
- ▶ yinelenebilir okuma
- ▶ sonlandırılanları okuyabilme
- ▶ sonlandırılmayanları okuyabilme

33 / 45

Serileştirilebilirlik

- ▶ **seri çalıştırma**: hareketlerin biri bitmeden diğeri başlamıyor
- ▶ **serileştirilebilir**: eşzamanlı çalışmanın sonucu her zaman seri çalıştırmalardan birinin sonucu ile aynı

Örnek

- ▶ $x = 10$
- ▶ A hareketi: $x = x + 1$
- ▶ B hareketi: $x = 2 * x$
- ▶ önce A, sonra B: $x = 22$
- ▶ önce B, sonra A: $x = 21$

34 / 45

İki Aşamalı Kitleme

- ▶ **iki aşamalı kitleme**: herhangi bir kilit bırakıldıktan sonra yeni kilit isteğinde bulunulmaz
 - ▶ genişleme aşaması: alınan kilit sayısı artıyor
 - ▶ daralma aşaması: alınan kilit sayısı azalıyor
- ▶ **iki aşamalı sıkı kitleme**: bütün kilitler hareketin sonunda bırakılır
- ▶ *Bütün hareketler iki aşamalı kitleme protokolüne uysa bütün eşzamanlı çalıştırmalar serileştirilebilir.*

35 / 45

Sonlandırılanları Okuyabilme

- ▶ yalnızca yazma kilitleri hareket sonuna kadar tutulur

Örnek

Hareket A	Hareket B
...	...
RETRIEVE p (S+)	...
...	...
kilidi bırak	...
...	...
...	UPDATE p (X+)
...	COMMIT
RETRIEVE p (S+)	

36 / 45

Hayaletler

Tanım

hayalet: sorgu yeniden çalıştırıldığında yeni çoklular ortaya çıkıyor

Örnek

- ▶ A hareketi bir müşterinin hesaplarının ortalamasını hesaplıyor:
 $\frac{100+100+100}{3} = 100$
- ▶ B hareketi aynı müşteriye 200 birimlik yeni bir hesap yaratıyor
- ▶ A hareketi hesabı yeniden yapıyor:
 $\frac{100+100+100+200}{4} = 125$

37 / 45

Yalıtım Düzeyi Belirleme

Komut

```
SET TRANSACTION ISOLATION LEVEL  
[ SERIALIZABLE | REPEATABLE READ |  
  READ COMMITTED | READ UNCOMMITTED ]
```

38 / 45

Yalıtım Düzeyi Sorunları

yalıtım düzeyi	kirli okuma	yinelemeyen okuma	hayalet
READ UNCOMMITTED	E	E	E
READ COMMITTED	H	E	E
REPEATABLE READ	H	H	E
SERIALIZABLE	H	H	H

39 / 45

Kilitleme Birimi

- ▶ kilitleme çoklu değil bağıntı değişkeni biriminde yapılabilir
 - ▶ hatta veri tabanı biriminde
- ▶ birim genişledikçe eşzamanlılık azalır
- ▶ çoklular üzerinde alınmış kilitlerin bulunması zor
→ önce bağıntı değişkeni düzeyinde **niyet kilitleri** alınsın

40 / 45

Niyet Kilitleri

- ▶ Parçayı Okuma (IS):
hareket bazı çokluları okumaya niyetleniyor
- ▶ Parçaya Yazma (IX):
IS + hareket bazı çoklulara yazmaya niyetleniyor
- ▶ Bütünü Okuma (S):
bağıntıda eşzamanlı okuyucular olabilir ama yazıcılar olmamalı
- ▶ Bütünü Okuma + Parçaya Yazma (SIX):
S + IX
- ▶ Bütüne Yazma (X):
bağıntıda hiçbir eşzamanlı çalışma olmamalı

41 / 45

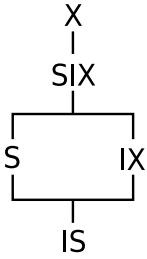
Kilit İstekleri

kilit uyumluluk matrisi

	X	SIX	IX	S	IS	-
X	H	H	H	H	H	E
SIX	H	H	H	H	E	E
IX	H	H	E	H	E	E
S	H	H	H	E	E	E
IS	H	E	E	E	E	E

42 / 45

Kilit Öncelikleri



- ▶ çoklu üzerinde okuma kilidi için bağıntı üzerinde en az IS
- ▶ çoklu üzerinde yazma kilidi için bağıntı üzerinde en az IX

43 / 45

Kilitleme Komutları

Komut

```
LOCK [ TABLE ] table_name  
[ IN lock_mode MODE ]
```

- ▶ kilit kipleri:
 - ▶ ACCESS SHARE
 - ▶ ROW SHARE
 - ▶ ROW EXCLUSIVE
 - ▶ SHARE UPDATE EXCLUSIVE
 - ▶ SHARE
 - ▶ SHARE ROW EXCLUSIVE
 - ▶ EXCLUSIVE
 - ▶ ACCESS EXCLUSIVE

44 / 45

Kaynaklar

Okunacak: Date

- ▶ Chapter 16: Concurrency

45 / 45