

IB-DB 6 - 2015

Dipl.-Ing. Reinhard Schlager

its
FH Salzburg

11.4.2015/ IB-Datenbanksysteme

Gliederung

- 1 Transaktionen
 - ACID - Eigenschaften von Transaktionen
 - Unkontrollierter Mehrbenutzerbetrieb
 - Transaktionen in Oracle
 - Isolation Level
 - Isolation Level - Die Konsequenzen
- 2 Locks
- 3 Übung 6

Gliederung

- 1 Transaktionen
 - ACID - Eigenschaften von Transaktionen
 - Unkontrollierter Mehrbenutzerbetrieb
 - Transaktionen in Oracle
 - Isolation Level
 - Isolation Level - Die Konsequenzen
- 2 Locks
- 3 Übung 6

ACID - Eigenschaften von Transaktionen

A-Atomarität

Alles oder Nichts

C-Consistency

Die Datenbank von einem konsistenten in einen wiederum konsistenten Zustand überführt

I-Isolation

Keine unerwünschten Nebenwirkungen gleichzeitig laufender Transaktionen

D-Durability Dauerhaftigkeit

ACID - Eigenschaften von Transaktionen

A-Atomarität

Alles oder Nichts

C-Consistency

Die Datenbank von einem konsistenten in einen wiederum konsistenten Zustand überführt

I-Isolation

Keine unerwünschten Nebenwirkungen gleichzeitig laufender Transaktionen

D-Durability Dauerhaftigkeit

ACID - Eigenschaften von Transaktionen

A-Atomarität

Alles oder Nichts

C-Consistency

Die Datenbank von einem konsistenten in einen wiederum konsistenten Zustand überführt

I-Isolation

Keine unerwünschten Nebenwirkungen gleichzeitig laufender Transaktionen

D-Durability Dauerhaftigkeit

ACID - Eigenschaften von Transaktionen

A-Atomarität

Alles oder Nichts

C-Consistency

Die Datenbank von einem konsistenten in einen wiederum konsistenten Zustand überführt

I-Isolation

Keine unerwünschten Nebenwirkungen gleichzeitig laufender Transaktionen

D-Durability Dauerhaftigkeit

Gliederung

- 1 **Transaktionen**
 - ACID - Eigenschaften von Transaktionen
 - **Unkontrollierter Mehrbenutzerbetrieb**
 - Transaktionen in Oracle
 - Isolation Level
 - Isolation Level - Die Konsequenzen
- 2 Locks
- 3 Übung 6

Unkontrollierter Mehrbenutzerbetrieb

Wo ist das Problem?

lost update

Transaktion 1 führt eine Überweisung von 10 von Konto K auf Konto X durch. Gleichzeitig führt eine Transaktion 2 die Zinsberechnung von Konto K durch.

lost update Beispiel

Zeitpunkt	Transa1 Überweisung	Transa2 Zinsberechnung
1	read K (100)	
2	$K = K - 10$	
3	read X (500)	read K (100)
4	$X = X + 10$	$K = K * 1.04$
5	write X(510)	write K(104)
6	write K(90)	

Unkontrollierter Mehrbenutzerbetrieb

Wo ist das Problem?

dirty read

Wie vorher, aber Transaktion 1 bricht ab.

dirty read Beispiel

Zeitpunkt	Transa1 Überweisung	Transa2 Zinsberechnung
1	read K (100)	
2	K=K-10	
3	write K (90)	
4		read K (90)
5		$K = K * 1.04$
6		write K(93.6)
7	...	
8	cancel	

Gliederung

- 1 **Transaktionen**
 - ACID - Eigenschaften von Transaktionen
 - Unkontrollierter Mehrbenutzerbetrieb
 - **Transaktionen in Oracle**
 - Isolation Level
 - Isolation Level - Die Konsequenzen
- 2 Locks
- 3 Übung 6

Transaktionen in Oracle

- Es gibt keinen besonderen Befehl zum Start einer Transaktion
- Soll ein SQL Kommando ausgeführt werden, wird automatisch von Oracle eine Transaktion gestartet, falls das noch nicht geschehen ist.
- Eine Transaktion wird mit *commit* oder *rollback* beendet.
- Wird die Transaktion mit *commit* abgeschlossen, werden alle durchgeführten Änderungen dauerhaft gespeichert. Ab diesem Zeitpunkt sind diese Änderungen frühestens für andere Transaktionen sichtbar. (siehe Isolation Level)
- Wird die Transaktion mit *rollback* abgeschlossen, werden alle Änderungen dieser Transaktion rückgängig gemacht.

Transaktionen in Oracle

- Es gibt keinen besonderen Befehl zum Start einer Transaktion
- Soll ein SQL Kommando ausgeführt werden, wird automatisch von Oracle eine Transaktion gestartet, falls das noch nicht geschehen ist.
- Eine Transaktion wird mit *commit* oder *rollback* beendet.
- Wird die Transaktion mit *commit* abgeschlossen, werden alle durchgeführten Änderungen dauerhaft gespeichert. Ab diesem Zeitpunkt sind diese Änderungen frühestens für andere Transaktionen sichtbar. (siehe Isolation Level)
- Wird die Transaktion mit *rollback* abgeschlossen, werden alle Änderungen dieser Transaktion rückgängig gemacht.

Transaktionen in Oracle

- Es gibt keinen besonderen Befehl zum Start einer Transaktion
- Soll ein SQL Kommando ausgeführt werden, wird automatisch von Oracle eine Transaktion gestartet, falls das noch nicht geschehen ist.
- Eine Transaktion wird mit *commit* oder *rollback* beendet.
- Wird die Transaktion mit *commit* abgeschlossen, werden alle durchgeführten Änderungen dauerhaft gespeichert. Ab diesem Zeitpunkt sind diese Änderungen frühestens für andere Transaktionen sichtbar. (siehe Isolation Level)
- Wird die Transaktion mit *rollback* abgeschlossen, werden alle Änderungen dieser Transaktion rückgängig gemacht.

Transaktionen in Oracle

- Es gibt keinen besonderen Befehl zum Start einer Transaktion
- Soll ein SQL Kommando ausgeführt werden, wird automatisch von Oracle eine Transaktion gestartet, falls das noch nicht geschehen ist.
- Eine Transaktion wird mit *commit* oder *rollback* beendet.
- Wird die Transaktion mit *commit* abgeschlossen, werden alle durchgeführten Änderungen dauerhaft gespeichert. Ab diesem Zeitpunkt sind diese Änderungen frühestens für andere Transaktionen sichtbar. (siehe Isolation Level)
- Wird die Transaktion mit *rollback* abgeschlossen, werden alle Änderungen dieser Transaktion rückgängig gemacht.

Transaktionen in Oracle

- Es gibt keinen besonderen Befehl zum Start einer Transaktion
- Soll ein SQL Kommando ausgeführt werden, wird automatisch von Oracle eine Transaktion gestartet, falls das noch nicht geschehen ist.
- Eine Transaktion wird mit *commit* oder *rollback* beendet.
- Wird die Transaktion mit *commit* abgeschlossen, werden alle durchgeführten Änderungen dauerhaft gespeichert. Ab diesem Zeitpunkt sind diese Änderungen frühestens für andere Transaktionen sichtbar. (siehe Isolation Level)
- Wird die Transaktion mit *rollback* abgeschlossen, werden alle Änderungen dieser Transaktion rückgängig gemacht.

Gliederung

- 1 **Transaktionen**
 - ACID - Eigenschaften von Transaktionen
 - Unkontrollierter Mehrbenutzerbetrieb
 - Transaktionen in Oracle
 - **Isolation Level**
 - Isolation Level - Die Konsequenzen
- 2 Locks
- 3 Übung 6

Isolation Level

Read Committed

Read Committed

Wird kein Isolation Level angegeben, wird dieser Level verwendet

Änderungen die von anderen Transaktionen bereits committed wurden, können gelesen werden

Isolation Level

Read Committed

Read Committed

Wird kein Isolation Level angegeben, wird dieser Level verwendet

Änderungen die von anderen Transaktionen bereits committed wurden, können gelesen werden

Isolation Level

Serializable

Serializable

Wird durch
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
eingeleitet

Die Transaktion “sieht” nur eigene Änderungen, die
Änderungen anderer Transaktionen sind “unsichtbar”

Isolation Level

Serializable

Serializable

Wird durch
`SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;`
eingeleitet

Die Transaktion “sieht” nur eigene Änderungen, die
Änderungen anderer Transaktionen sind “unsichtbar”

Isolation Level

READ ONLY

READ ONLY

Wird durch
SET TRANSACTION READ ONLY;
eingeleitet

Es darf nur lesend auf die Datenbank zugegriffen werden

Die Transaktion liest nur Daten, die bei Beginn der Transaktion committed waren

Isolation Level

READ ONLY

READ ONLY

Wird durch
`SET TRANSACTION READ ONLY;`
eingeleitet

Es darf nur lesend auf die Datenbank zugegriffen werden

Die Transaktion liest nur Daten, die bei Beginn der Transaktion
committed waren

Isolation Level

READ ONLY

READ ONLY

Wird durch
`SET TRANSACTION READ ONLY;`
eingeleitet

Es darf nur lesend auf die Datenbank zugegriffen werden

Die Transaktion liest nur Daten, die bei Beginn der Transaktion committed waren

Gliederung

- 1 **Transaktionen**
 - ACID - Eigenschaften von Transaktionen
 - Unkontrollierter Mehrbenutzerbetrieb
 - Transaktionen in Oracle
 - Isolation Level
 - **Isolation Level - Die Konsequenzen**
- 2 Locks
- 3 Übung 6

Isolation Level

Die Konsequenzen:

Das Ergebnis einer Abfrage liefert NICHT immer die aktuellsten Daten!

Transaktionen

So kurz wie möglich - so lang wie nötig

Isolation Level

Die Konsequenzen:

Das Ergebnis einer Abfrage liefert NICHT immer die aktuellsten Daten!

Transaktionen

So kurz wie möglich - so lang wie nötig

Nächste freie Nummer - lückenlos

```
CREATE TABLE SerialNr (SerialNumber NUMBER(9,0))  
INSERT INTO SerialNr VALUES (1)
```

Nächste freie Nummer - lückenlos

```
CREATE TABLE SerialNr (SerialNumber NUMBER(9,0))  
INSERT INTO SerialNr VALUES (1)
```

Nächste freie Nummer - lückenlos 2

```
LOCK TABLE SerialNr IN EXCLUSIVE MODE
UPDATE SerialNr SET SerialNumber = SerialNumber +1
SELECT SerialNr FROM SerialNumber
commit
```


Nächste freie Nummer - lückenlos 2

```
LOCK TABLE SerialNr IN EXCLUSIVE MODE
  UPDATE SerialNr SET SerialNumber = SerialNumber +1
SELECT SerialNr FROM SerialNumber
commit
```

Nächste freie Nummer - lückenlos 2

```
LOCK TABLE SerialNr IN EXCLUSIVE MODE  
  UPDATE SerialNr SET SerialNumber = SerialNumber +1  
SELECT SerialNr FROM SerialNumber  
commit
```

Nächste freie Nummer - lückenlos 2

```
LOCK TABLE SerialNr IN EXCLUSIVE MODE  
  UPDATE SerialNr SET SerialNumber = SerialNumber +1  
SELECT SerialNr FROM SerialNumber  
commit
```

Übung 6

ERM

Entwerfen Sie ein minimales ERM einer Bank und setzen Sie das ERM in eine ORACLE Datenbank um

stored function

Schreiben Sie eine stored function, die 3 Parameter erwartet (Konto1,Konto2,Betrag) und dann innerhalb *einer* Transaktion die Überweisung von Konto 1 auf Konto 2 vornimmt.

Rückgabewerte

Wenn die Transaktion erfolgreich beendet wurde, soll 0 zurückgegeben werden. Überlegen Sie Fehlerfälle und definieren Sie dafür Rückgabewerte.

Übung 6

ERM

Entwerfen Sie ein minimales ERM einer Bank und setzen Sie das ERM in eine ORACLE Datenbank um

stored function

Schreiben Sie eine stored function, die 3 Parameter erwartet (Konto1,Konto2,Betrag) und dann innerhalb *einer* Transaktion die Überweisung von Konto 1 auf Konto 2 vornimmt.

Rückgabewerte

Wenn die Transaktion erfolgreich beendet wurde, soll 0 zurückgegeben werden. Überlegen Sie Fehlerfälle und definieren Sie dafür Rückgabewerte.

Übung 6

ERM

Entwerfen Sie ein minimales ERM einer Bank und setzen Sie das ERM in eine ORACLE Datenbank um

stored function

Schreiben Sie eine stored function, die 3 Parameter erwartet (Konto1,Konto2,Betrag) und dann innerhalb *einer* Transaktion die Überweisung von Konto 1 auf Konto 2 vornimmt.

Rückgabewerte

Wenn die Transaktion erfolgreich beendet wurde, soll 0 zurückgegeben werden. Überlegen Sie Fehlerfälle und definieren Sie dafür Rückgabewerte.