

Transactiebeheer





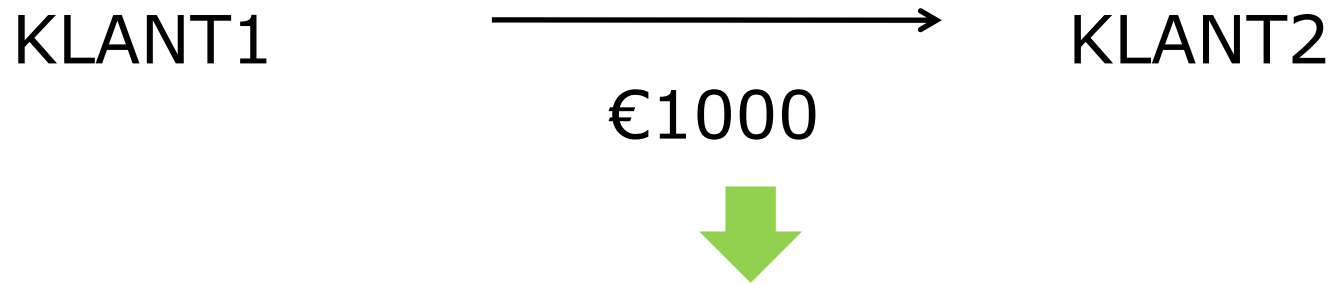
Cursus Boek

- Cursus 'Databanken 1' blz. 61-65
- Deze slides
- Aanbevolen: Oracle Database 11g: SQL Fundamentals I Exam Guide Blz. 427-436

Wat is een transactie?

Voorbeeld:

Binnen eenzelfde bank



Er moeten 2 UPDATE instructies uitgevoerd worden:

- 1. Update waarbij de rekening van klant1 met €1000 wordt verminderd*
- 2. Update waarbij de rekening van klant2 met €1000 wordt verhoogd*

Wat is een transactie?

Veronderstel volgende situatie:

Update 1 wordt uitgevoerd (de rekening van klant1 wordt met €1000 verminderd)

Systeem valt uit

- ➡ Update waarbij de rekening van klant2 met €1000 wordt verhoogd, wordt niet uitgevoerd
- ➡ De rekeningen in de databank vertonen inconsistenties

Wat is een transactie?

Oplossing:

Door de 2 instructies samen in een transactie te plaatsen voorkomt het DBMS voorgaande situatie.

Begin transactie ()*

Update1 (rekening klant1 wordt met €1000 verminderd)

Update2 (rekening klant2 wordt met €1000 verhoogd)

einde transactie ()*

(*) wanneer een transactie begint/eindigt komt verder aan bod

Wat is een transactie?

Alle instructies binnen een transactie slagen of falen steeds samen.

Begin transactie

Update1 (rekening klant1 wordt met €1000 verminderd)

Systeem valt uit

Update2 (rekening klant2 wordt met €1000 verhoogd)

Einde transactie

⇒ de 1e UPDATE wordt teruggedraaid dus consistentie blijft behouden

Wat is een transactie?

Begin transactie

Update1 (rekening klant1 wordt met €1000 verminderd)

Update2 (rekening klant2 wordt met €1000 verhoogd)

Systeem valt uit

⇒ de 2 UPDATES worden teruggedraaid dus consistentie blijft behouden

Wat is een transactie?

Begin transactie

Update1 (rekening klant1 wordt met €1000 verminderd)

Update2 (rekening klant2 wordt met €1000 verhoogd)

Commit; (=einde transactie)

Systeem valt uit

⇒ 2 UPDATES werden definitief doorgevoerd voor de panne dus consistentie blijft behouden

Wat is een transactie?



Een transactie bestaat uit een aantal bij elkaar horende instructies waarvoor het DBMS de garantie geeft dat ze samen slagen of samen falen.

- Het is een samenhangende groep van instructies (waaronder meestal DML-instructies!) die of
- door een COMMIT (expliciet of impliciet) definitief wordt gemaakt
- door ROLLBACK (expliciet of impliciet) terug ongedaan gemaakt wordt

Wanneer begint een transactie?

- 1^e SQL instructie in begin sessie
- na een COMMIT of ROLLBACK instructie
 - COMMIT maakt wijzigingen definitief
 - ROLLBACK maakt wijzigingen ongedaan tot aan de laatste COMMIT
- na een DDL instructie: CREATE, ALTER, DROP...
 - = impliciete commit
- na een DCL instructie (beveiliging: zie later)
 - = impliciete commit

Impliciete commit

- wil zeggen dat de instructie onmiddellijk wordt doorgevoerd naar de databank, de wijzigingen worden dus definitief weggeschreven!

Wanneer eindigt een transactie?

- bij een expliciete COMMIT of ROLLBACK
- bij een impliciete COMMIT ten gevolge van een DDL instructie (CREATE, ALTER, DROP, TRUNCATE)
- bij een impliciete COMMIT ten gevolge van een DCL instructie (zie later: beveiliging)
- bij eindigen van sessie (EXIT) = impliciete commit
- bij het foutief beëindigen van sessie (bv computer valt uit ..)
= impliciete rollback

Voorbeeld

SOFI_NR	ACHTERNAAM	SALARIS
999887777	Muiden	25000
999222222	Amelsvoort	25000
999888888	Pregers	25000

1. **UPDATE** medewerkers
SET salaris = 26000 **WHERE**
achternaam='Pregers';

SOFI_NR	ACHTERNAAM	SALARIS
999887777	Muiden	25000
999222222	Amelsvoort	25000
999888888	Pregers	26000

2. **ROLLBACK** ;

SOFI_NR	ACHTERNAAM	SALARIS
999887777	Muiden	25000
999222222	Amelsvoort	25000
999888888	Pregers	25000

3. **UPDATE** medewerkers
SET salaris = 27000 **WHERE**
achternaam='Pregers';
COMMIT;

SOFI_NR	ACHTERNAAM	SALARIS
999887777	Muiden	25000
999222222	Amelsvoort	27000
999888888	Pregers	27000

4. **ROLLBACK** ;

SOFI_NR	ACHTERNAAM	SALARIS
999887777	Muiden	25000
999222222	Amelsvoort	25000
999888888	Pregers	27000

ROLLBACK TO SAVEPOINT



ROLLBACK maakt DML instructies ongedaan tot aan de laatst gegeven COMMIT (expliciet of impliciet).

Je kan binnen een transactie savepoints plaatsen met de instructie:

```
SAVEPOINT naam_savepoint; (*)
```

Met de instructie

```
ROLLBACK TO SAVEPOINT naam_savepoint (*)
```

kan je dan DML instructies ongedaan maken tot aan het aangegeven savepoint (=een bepaald punt binnen de transactie)

(*) *naam_savepoint* moet voldoen aan de regels voor naamgeving

ROLLBACK TO SAVEPOINT



<= Begin transactie

```
UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Pregers';

SAVEPOINT sal_pregers;

UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Joosten';

SAVEPOINT sal_joosten;

SELECT SUM(salaris)
FROM medewerkers;

ROLLBACK TO SAVEPOINT sal_pregers;

UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Joosten';

ROLLBACK TO SAVEPOINT sal_joosten;
```

<= Error. Waarom?

ROLLBACK TO SAVEPOINT



-- *begin transactie*

```
UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Pregers';
```

```
SAVEPOINT sal_pregers;
```

```
UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Joosten';
```

```
SAVEPOINT sal_joosten;
```

```
SELECT SUM(salaris)
FROM medewerkers;
```

```
ROLLBACK TO SAVEPOINT sal_pregers;
```

```
UPDATE medewerkers
SET salaris = salaris * 1.1
WHERE INITCAP(achternaam)='Joosten';
```

```
ROLLBACK TO SAVEPOINT sal_joosten;
```

ACHTERNAAM	SALARIS
------------	---------

Joosten	38.000
---------	--------

Pregers	25.000
---------	--------

ACHTERNAAM	SALARIS
------------	---------

Joosten	38.000
---------	--------

Pregers	27.500
---------	--------

ACHTERNAAM	SALARIS
------------	---------

Joosten	41.800
---------	--------

Pregers	27.500
---------	--------

```
SUM(MDW_SALARIS)
290300
```

ACHTERNAAM	SALARIS
------------	---------

Joosten	38.000
---------	--------

Pregers	27.500
---------	--------

ACHTERNAAM	SALARIS
------------	---------

Joosten	41.800
---------	--------

Pregers	27.500
---------	--------

Error report -

SQL Error: ORA-01086: savepoint 'SAL_JOOSTEN' never
01086. 00000 - "savepoint '%s' never established"

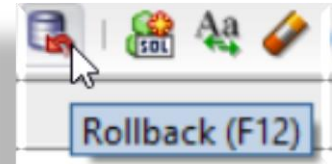
Wat is het resultaat van een COMMIT?

- ❑ Alle wijzigingen in de huidige transactie worden effectief op database geplaatst;
- ❑ Alle SAVEPOINTS worden uitwist;
- ❑ Alle transactie locks (zie verder) worden vrijgeven.
- ❑ Andere gebruikers zien de wijzigingen die in de afgesloten transactie werden doorgevoerd. (zie verder)



Wat is het resultaat van een ROLLBACK?

- ☐ De transactie wordt beëindigd.
- ☐ Alle wijzigingen in de huidige transactie worden vernietigd;
- ☐ Alle savepoints in de huidige transactie worden verwijderd;
- ☐ Alle locks van deze transactie worden vrijgegeven.



Wat is het resultaat van een ROLLBACK TO SAVEPOINT?

- ☐ Een deel van de transactie wordt ongedaan gemaakt
- ☐ Het vermelde savepoint blijft bestaan, maar alle savepoints die nog volgden worden verwijderd!

Hoe groot moet een transactie zijn?

COMMIT na elke instructie?

NEEN -> Gevaar voor inconsistenties

Update1 (rekening klant1 verminderd met €1000)
COMMIT;



systeem valt uit

Update2 (rekening klant2 verhoogd met €1000)
COMMIT;

Hoe groot moet een transactie zijn?

COMMIT enkel op einde van de sessie?

NEEN -> ROLLBACK maakt alle in de sessie gegeven
DML instructies opnieuw ongedaan!



begin sessie

update

update

delete

insert

insert

laatste insert opnieuw ongedaan maken



er wordt gerollbackt tot aan het begin van
de sessie!

Hoe groot moet een transactie zijn?

Hangt af van de bedrijfslogica. Sluit een transactie af na een groep logisch bij elkaar horende DML instructies (logical grouping)

(zie vb klant1 doet een overschrijving naar bankrekening klant2)

Eigenschappen van transacties (ACID)



Definitie

Een transactie moet steeds aan de volgende 4 eigenschappen voldoen:

Atomair:

de instructies binnen de transactie zijn één en ondeelbaar. Of zij worden allemaal uitgevoerd of er wordt geen enkele uitgevoerd.

Consistent:

de databank was consistent voor het uitvoeren van de transactie en moet dat ook zijn erna.

Eigenschappen van transacties (ACID)

- **Isolatie:**
 - een transactie wordt uitgevoerd alsof het de enige transactie is die wordt uitgevoerd. Zij ziet de veranderingen die in andere transacties die nog bezig zijn niet.
- **Duurzaam:**
 - Na uitvoering van de transactie worden de wijzigingen die in de transactie werden doorgevoerd definitief (COMMIT) . In geval van een systeem fout kunnen we steeds naar deze staat terugkeren. Bij ROLLBACK is het alsof de wijziging nooit plaatsvond.

autocommit

De variabele 'autocommit' bepaalt of na elke wijziging automatisch een COMMIT gebeurt. Je past de variabele aan met het SET commando:

```
SET autocommit OFF;
```

Expliciete COMMIT (of ROLLBACK) voor DML instructies door gebruiker (standaard waarde).

```
SET autocommit ON;
```

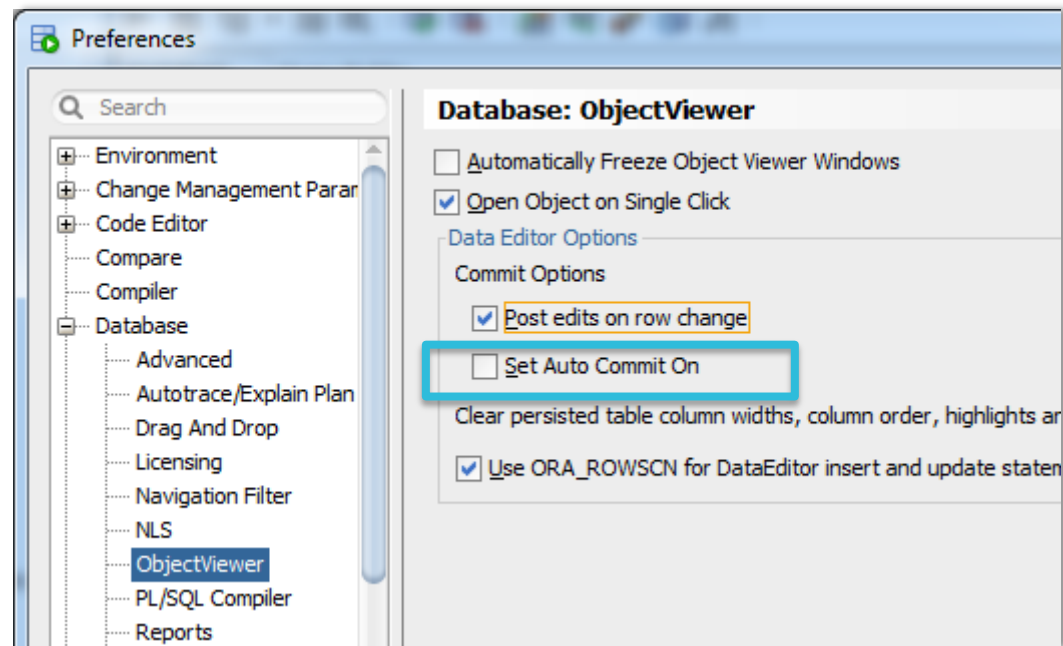
Elke wijziging wordt onmiddellijk naar de databank geschreven. Hier ben je dan niet meer in staat om iets ongedaan te maken, dus denk hier goed over na!

Om de waarde van de variabele te bekijken:

```
SHOW autocommit;
```

autocommit

SQLDeveloper: Aan of afzetten kan via
Tools > Preferences > Database > ObjectViewer



- p.26