

TP : Niveaux d'isolation dans Oracle 10g
Hala Skaf-Molli

L'objectif de ce Tp est de tester les niveaux d'isolation d'oracle.

Question1: Transaction-Level Read Consistency (set transaction isolation level):

Vous devez travailler en binôme. Les tables sont créés par une seule personne.
 Tester les scénarios ci-dessous et indiquer les problèmes lies à chaque scénario.

```
create table toto (i int);
insert into toto values(1);
insert into toto values(2);
grant all on toto to ?????;
```

Niveau d'isolation 1 (READ COMMITTED) Niveau d'isolation 3 (SERIALIZABLE)

Scénario1 :

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
select * from toto;	
	insert into toto values(3);
select * from toto;	
	commit;
select * from toto;	

Scénario 2

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
select * from toto;	
	update toto set i =4 ;
select * from toto;	
	commit;
select * from toto;	

Scénario 3

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
	insert into toto values(7);
insert into toto values (8);	
	select * from toto ;
select * from toto;	
	commit;
select * from toto ,	
	select * from toto;
commit;	
	select * from toto

Scénario 4

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
	update toto set i= 9
update toto set i =10 ;	
	commit;
select * toto;	

Re-exécuter les mêmes scénarios avec un niveau d'isolation 3 (set transaction isolation level serializable). (Pensez à prendre de valeurs qui montrent le changement ..)

Pour la suite, je modifie la définition de la table toto :

```
alter table toto add j int default 0;  
insert into toto (11,1);
```

Scénario 5

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
	update toto set i= 13 where j=1;
update toto set i =15 where j =0;	
	select * from toto ;
select * from toto ;	
	commit;
select * from toto;	
commit;	
select * from toto;	

Scénario 6

t1	t2
set transaction isolation level serializable	set transaction isolation level serializable
	update toto set i= 20 where j=1;
update toto set i =20 where j =0;	
	select * from toto
select * toto	
	commit;
select * from toto	

Cannot serialize access for this transaction

Oracle generates an error when a serializable transaction tries to update or delete data modified by a transaction that commits after the serializable transaction began:

ORA-08177: Cannot serialize access for this transaction

When a serializable transaction fails with the "Cannot serialize access" error, the application can take any of several actions:

- * Commit the work executed to that point
- * Execute additional (but different) statements (perhaps after rolling back to a savepoint established earlier in the transaction)
- * Roll back the entire transaction

Scénario 7

t1	t2
set transaction isolation level SERIALIZABLE	set transaction isolation level SERIALIZABLE
	update toto set i= 9
update toto set i =90	
	commit;
commit;	

```
create table titi(i int);
insert into titi values(11);
grant all on titi to ?????;
```

Scénario 8

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
insert into toto values (1,2);	
	insert into titi values(22);
	select * from toto ;
select * from titi;	
	commit;
select * from titi;	

Scénario 9

t1	t2
set transaction isolation level READ COMMITTED	set transaction isolation level READ COMMITTED
update toto set i =3 where j=2	
	update titi set i =33;
select * from titi;	
	select * from toto where j=2;

Question 2: Statement-Level Read Consistency

Oracle always enforces statement-level read consistency. This guarantees that all the data returned by a single query comes from a single point in time--the time that the query began. Therefore, a query never sees dirty data nor any of the changes made by transactions that commit during query execution. As query execution proceeds, only data committed before the query began is visible to the query. The query does not see changes committed after statement execution begins.

Scénario 10

t1	t2
	update toto set i= 30 where j=1
update toto set i =6 where j=2	
	update toto set i=300 where j=2
update toto set i=100 where j=1	

Conclusion:

Read Committed and Serializable Transactions (from Oracle Documentation)

READ COMMITTED	SERIALIZABLE
-----------------------	---------------------

Dirty write	Not possible	Not possible
Dirty read	Not possible	Not possible
Nonrepeatable read	Possible	Not possible
Phantoms	Possible	Not possible
Compliant with ANSI/ISO SQL 92	Yes	Yes
Read materialized view time	Statement	Transaction
Transaction set consistency	Statement level	Transaction level
Row-level locking	Yes	Yes
Readers block writers	No	No
Writers block readers	No	No
Different-row writers block writers	No	No
Same-row writers block writers	Yes	Yes
Waits for blocking transaction	Yes	Yes
Subject to "cannot serialize access"	No	Yes
Error after blocking transaction terminates	No	No
Error after blocking transaction commits	No	Yes

Question 3: Rollback and Save point :

- Exécuter les scénarios suivants :

Scénario 11:

```
select * from toto ;
insert into toto values(100,100);
rollback,
select * from toto;
```

Scénario 12 :

```
select * from toto ;
insert into toto values(100,100);
select * from toto;
savepoint sp_1;
select * from titi;
update titi set l=15;
select * from titi;
rollback to sp_1;
select * from toto;
select * from titi;
commit;
select * from toto;
```