

Snapshot nivoi izolacije

Snapshot – kopija konzistentnog stanja baze u jednom trenutku. Transakcija sa snapshot nivoom izolacije se obavlja nad tom kopijom. Minimizuje se blokiranje, poboljšava se konkurentnost. Međutim, opterećuje se **tempdb** baza.

tempdb – globalni resurs dostupan svim korisnicima povezanim na instance SQL Servera. Čuva privremene tabele, tabele sa međurezultatima, kursore i *snapshot*-ove.

U dosadašnjim nivoima izolacije (osim *Read Uncommitted*), modifikacija podataka (*X lock*) uvek blokira čitanje podataka (*S lock*) i obrnuto. Da bi se smanjila količina blokiranja, uvedeni su *snapshot* nivoi izolacije. Oni čuvaju jednu verziju prethodno komitovanih podataka u **tempdb**. Time dopuštaju čitanje starih konzistentnih podataka, iako su ti podaci možda ustvari blokirani i promenjeni u međuvremenu.

Postoje **Snapshot** i **Read Committed Snapshot** nivoi izolacije.

Snapshot Isolation (SI)

Prvo mora da se omogući *snapshot* na nivou baze.

```
ALTER DATABASE transakcije  
SET ALLOW_SNAPSHOT_ISOLATION ON
```

Sledeća naredba proverava da li su omogućeni *Read Committed Snapshot* i *Snapshot* nivoi izolacije.

```
SELECT name, is_read_committed_snapshot_on, snapshot_isolation_state  
FROM sys.databases  
WHERE name = 'transakcije'
```

U ovom slučaju, samo je SI omogućen.

Primeri će biti rađeni nad tabelom **racuni**.

	id	iznos
1	1	1100
2	2	100
3	3	500
4	4	1000

Primer 1 – READ COMMITTED piše, READ COMMITTED čita

Ovaj primer pokazuje kako modifikacija podataka blokira čitanje istih tih podatka.

T1, T2 – *read committed*, konkurentno

T1 piše nesto

T2 čita to isto – blokira se

```
-- 1. RC piše - RC čita

--1
SELECT iznos FROM racuni
WHERE id = 1
--1100

--2
SET TRANSACTION ISOLATION LEVEL read
committed
BEGIN TRAN t1
UPDATE racuni
SET iznos = 1200
WHERE id = 1

--3
SELECT iznos FROM racuni
WHERE id = 1
--1200

--5
COMMIT TRAN t1
```

```
-- 1. RC piše - RC čita

--1
SELECT iznos FROM racuni
WHERE id = 1
--1100

--4
SET TRANSACTION ISOLATION LEVEL read
committed
BEGIN TRAN t2
SELECT iznos FROM racuni
WHERE id = 1
-- blokira
-- 1200 nakon commit-a t1

--6
COMMIT TRAN t2
```

Primer 2 - READ COMMITTED piše, SNAPSHOT čita

Sa *Snapshot* nivoom izolacije, modifikacija podataka ne blokira čitanje tih istih podataka.

T1 – *read committed*, T2 – *snapshot*, konkurentno

T1 piše nešto

T2 čita to isto – čita stanje pre transakcije

T1 *commit*

T2 čita stanje pre transakcije

T2 *commit*

T2 čita stanje posle transakcije

-- 2. read committed piše - snapshot čita

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1200

--2
SET TRANSACTION ISOLATION LEVEL read
committed
BEGIN TRAN t1
UPDATE racuni
SET iznos = 1100
WHERE id = 1

--3
SELECT iznos FROM racuni
WHERE id = 1
--1100

--5
COMMIT TRAN t1

--8
SELECT iznos FROM racuni
WHERE id = 1
--1100
```

-- 2. read committed piše - snapshot čita

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1200

--4
SET TRANSACTION ISOLATION LEVEL snapshot
BEGIN TRAN t2
SELECT iznos FROM racuni
WHERE id = 1
--1200

--6
SELECT iznos FROM racuni
WHERE id = 1
--1200

--7
COMMIT TRAN t2

--8
SELECT iznos FROM racuni
WHERE id = 1
--1100
```

Primer 3 - *READ COMMITTED* piše, *SNAPSHOT* piše

Šta se dešava ako transakcija sa *snapshot* nivoom izolacije pokuša da modifikuje već modifikovane podatke?

T1 – *read committed*, T2 – *snapshot*, konkurentno

T1 piše nešto

T2 čita to isto – čita stanje pre transakcije

T2 modifikuje to isto – blokira se

T1 *commit*

T2 *aborted*, jer treba da modifikuje modifikovano od strane druge transakcije

-- 3. read committed piše - snapshot piše

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1100

--2
SET TRANSACTION ISOLATION LEVEL read
committed
BEGIN TRAN
UPDATE racuni
SET iznos = 1200
WHERE id = 1

--3
SELECT iznos FROM racuni
WHERE id = 1
--1200

--6
COMMIT TRAN

--7
SELECT iznos FROM racuni
WHERE id = 1
--1200
```

-- 3. read committed piše - snapshot piše

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1100

--4
SET TRANSACTION ISOLATION LEVEL snapshot
BEGIN TRAN
SELECT iznos FROM racuni
WHERE id = 1
--1100

--5
UPDATE racuni
SET iznos = 1000
WHERE id = 1
--blokira se
--abort posle t1 commit-a
--automatski rollback

--7
SELECT iznos FROM racuni
WHERE id = 1
--1200
```

Read Committed Snapshot Isolation (RCSI)

Da bi se omogućio ovaj nivo izolacije, potrebno je izvršiti naredbu:

```
ALTER DATABASE transakcije  
SET READ_COMMITTED_SNAPSHOT ON
```

Međutim, ova naredba neće raditi ako na instance postoji više od jedne konekcije. U tom slučaju je potrebno izvršiti:

```
ALTER DATABASE transakcije  
SET READ_COMMITTED_SNAPSHOT ON  
WITH ROLLBACK IMMEDIATE
```

Ova naredba prekida sve ostale konekcije i omogućava RCSI nivo.

Sada je RCSI podrazumevani nivo izolacije. To je i jedini način da koristimo ovaj nivo, ne postoji zasebna naredba za njegovo postavljanje.

RCSI uvek čita poslednje *commit*-ovane podatke.

Primer 4 - *READ COMMITTED* piše, *READ COMMITTED SNAPSHOT* čita i piše

T1 – *read committed*, T2 – *read committed snapshot*

T1 piše nešto

T2 čita to isto – čita stanje pre transakcije

T1 *commit*

T2 čita stanje posle *commit*-a

T2 može to da modifikuje

T2 *commit*

-- 4. RC piše - RCSI čita

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1200

--2
SET TRANSACTION ISOLATION LEVEL read
committed
BEGIN TRAN t1
UPDATE racuni
SET iznos = 1100
WHERE id = 1

--3
SELECT iznos FROM racuni
WHERE id = 1
--1100

--5
COMMIT TRAN t1

--10
SELECT iznos FROM racuni
WHERE id = 1
--1000
```

-- 4. RC piše - RCSI čita

```
--1
SELECT iznos FROM racuni
WHERE id = 1
--1200

--4
-- read committed snapshot je sada
podrazumevani nivo
BEGIN TRAN t2
SELECT iznos FROM racuni
WHERE id = 1
--1200

--6
SELECT iznos FROM racuni
WHERE id = 1
--1100!

--7
UPDATE racuni
SET iznos = 1000
WHERE id = 1

--8
SELECT iznos FROM racuni
WHERE id = 1
--1000!

--9
COMMIT TRAN t2

--10
SELECT iznos FROM racuni
WHERE id = 1
--1000
```