



Računarstvo i informatika

Katedra za računarstvo

Elektronski fakultet u Nišu

Sistemi baza podataka

Transakcije i mehanizam zaključavanja

Letnji semestar 2014/2015



Sadržaj

- Obrada transakcija
- COMMIT
- ROLLBACK
- SAVEPOINT
- Isolation levels
- Mehanizam zaključavanja



Obrada transakcija

- Svi poslovi koje Oracle obavlja nazivaju se sesijama.
- Korisnička sesija se startuje kada se korisnik korišćenjem nekog alata ili aplikacije konektuje na Oracle server.
- Oracle poseduje mehanizme za kontrolu konkurentnih korisničkih sesija čime je obezbeđeno da veći broj korisnika može simultano da radi sa istim skupom podataka u bazi podataka bez narušavanja integriteta.
- Oracle koristi mehanizme zaključavanja da bi pratio konkurentno pristupanje istom skupu podataka.
- Lock preuzima privremeno vlasništvo nad određenim resursom (tabela ili vrsta u tabeli) u bazi podataka dok se određena operacija ne završi.
- Oracle upravlja mehanizmom zaključavanja nezavisno od korisnika. Korisnici, međutim, poseduju mogućnost da u slučaju potrebe izmene funkcionisanje ovih mehanizama.



Obrada transakcija

- Za svaki upit koji se izvršava nad bazom podataka Oracle obezbeđuje read-consistent view podataka. Time je omogućeno da podaci koji su dobijeni izvršavanjem upita ne mogu biti promenjeni od strane drugih sesija.
- Za svaku naredbu ažuriranja koja se izvršava Oracle kreira rollback segment u koji snima trenutno stanje (snapshot) tabela koje se ažuriraju. Rollback segment se koristi za oporavak baze podataka u slučaju grešaka koje mogu da se jave.



Obrada transakcija

- ACID svojstva transakcije

- Atomičnost (**A**tomicity)

- Sve naredbe transakcije se izvršavaju kao jedinstvena celina ili se uopšte ne izvršavaju

- Konzistentnost (**C**onsistency)

- Transakcija prevodi bazu podataka iz jednog konzistentnog stanja u drugo konzistentno stanje.

- Izolacija (**I**solation)

- Efekti transakcije nisu vidljivi u drugim transakcija sve dok se transakcija ne komituje.

- Trajnost (**D**urability)

- Izmene nad podacima načinjene tokom transakcije su trajne.



Obrada transakcija

- Transakcija predstavlja sekvencu SQL DML (data manipulation naredbi) naredbi koja predstavlja jednu logičku celinu.
- Oracle transakciju tretira kao celinu sve dok sve izmene načinjene od strane SQL naredbi ne postanu validne (committed) ili dok ne budu poništene (rolled back).
- Ukoliko dođe do greške tokom transakcije Oracle poništava sve efekte transakcije i vraća bazu podataka u stanje pre početka transakcije.
- Prva SQL naredba nekog programa započinje novu transakciju.



Obrada transakcija

- Kada se jedna transakcija završi, prva naredna SQL naredba, započinje novu transakciju.
- Svaka SQL naredba je sastavni deo neke transakcije, čak i ukoliko je jedina naredba u toj transakciji.
- Naredbe COMMIT i ROLLBACK obezbeđuju da su svi efekti transakcije moraju postati validni ili se moraju potpuno poništiti.
- Transakciju čine sve SQL naredbe izvršene nakon poslednjeg poziva COMMIT ili ROLLBACK naredbe.
- Naredba SAVEPOINT označava i snima tekuću tačku u transakciji.



COMMIT

- Naredba COMMIT završava tekuću transakciju i potvrđuje i trajno snima sve modifikacije učinjene tokom transakcije.

COMMIT [WORK] [COMMENT <tekst komentara>];

- Sve dok se promene ne potvrde (commit-uju) one nisu vidljive ostalim korisnicima. Ostali korisnici vide podatke u onom obliku u kome su bili pre početka transakcije.
- Naredba COMMIT poništava sve lock-ove koji su tokom transakcije kreirani nad tabelama ili pojedinim vrstama u tabelama.



COMMIT

DECLARE

ime **VARCHAR**(20);

prezime **VARCHAR**(30);

sslovo **CHAR**(1);

BEGIN

INSERT INTO RADNIK **VALUES**(...);

INSERT INTO RADNIK **VALUES**(...);

COMMIT;

SELECT LIME, PREZIME, INIC

INTO ime, prezime, sslovo

FROM PREDUZECE.RADNIK

WHERE MATBR = 10;

COMMIT WORK;

UPDATE RADNIK **SET** ... **WHERE** MATBR = 10;

COMMIT COMMENT 'IZMENA';

END;

COMMIT nakon serije naredbi kojima se dodaje novi slog.

COMMIT se može naći nakon bilo koje SQL naredbe.

COMMIT WORK ima isti efekat kao i COMMIT.

Svakoj COMMIT naredbi se može pridružiti komentar.



ROLLBACK

- Naredba ROLLBACK završava tekuću transakciju i poništava sve modifikacije učinjene tokom transakcije.
- Transakcija se poništava u situacijama kada je izdata naredba koja je dovela do neželjenih efekata (obrisani ili ažurirani pogrešni podaci) ili kada naredba iz određenih razloga ne može da se izvrši ili se tokom izvršenja javi izuzetak.
- Naredba ROLLBACK bazu podataka vraća u prvobitno stanje odnosno stanje u kome se nalazila pre nego što je izvršena prva naredba transakcije.
- To omogućava korisniku da preduzme korake koji će ispraviti uočene greške i nedostatke i da zatim ponovo pokrene transakciju.



ROLLBACK

- Ukoliko prilikom izvršavanja neke SQL naredbe dođe do greške ili pojave izuzetka Oracle automatski poništava (automatic rollback) sve efekte te naredbe.
- Automatski rollback na nivou naredbe ne utiče na efekte prethodnih naredbi transakciji.
- Oraclu automatski može da poništi neku SQL naredbu ukoliko detektuje pojavu deadlock-a u sistemu.



ROLLBACK

DECLARE

ime **VARCHAR**(20);

prezime **VARCHAR**(30);

sslovo **CHAR**(1);

BEGIN

INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);

INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);

COMMIT;

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN ROLLBACK;

END;

Dva puta se u bazu podataka dodaju podaci o istom radniku.

Ukoliko sve naredbe prođu bez grešaka transakcija se commit-uje.

Desio se izuzetak tokom izvršavanja transakcije. Poništavaju se efekti kompletne transakcije..



SAVEPOINT

- Naredba **SAVEPOINT** omogućava da se označi (imenuje) i snimi tekuća tačka u transakciji.
- U kombinaciji sa naredbom **ROLLBACK**, naredba **SAVEPOINT**, omogućava da se u slučaju grešaka ponište efekti samo pojedinih delova transakcije a ne čitave transakcije.
- Nazivi **SAVEPOINT** tačaka se mogu ponavljati više puta.
- Aktivna **SAVEPOINT** tačka je poziv **SAVEPOINT** naredbe nakon poslednjeg poziva **COMMIT** ili **ROLLBACK** naredbe.
- Transakcija može da ima neograničeni broj aktivnih **SAVEPOINT** tačaka..



SAVEPOINT

BEGIN

```
UPDATE RADNIK SET PLATA = PLATA * 2;  
DELETE FROM RADNIK WHERE BROD = 10;
```

```
SAVEPOINT sledi_insert;
```

```
INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);  
INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);
```

```
COMMIT;
```

EXCEPTION

```
WHEN DUP_VAL_ON_INDEX THEN  
    ROLLBACK sledi_insert;
```

```
END;
```

Snima se tekuća tačka transakcije.

Transakcija se commit-uje ukoliko nema grešaka.

U slučaju pokušaja da se duplira radnik poništiće se efekti samo naredbi nakon SAVEPOINT tačke.

Izmene koje potiču od naredbi DELETE i UPDATE nisu poništene.



SAVEPOINT

BEGIN

INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);

SAVEPOINT sledi_update;

UPDATE RADNIK SET PLATA = PLATA * 2;

SAVEPOINT sledi_delete;

DELETE FROM RADNIK WHERE BROD = 10;

SAVEPOINT sledi_insert;

INSERT INTO RADNIK VALUES(111, 'Mirko', 'Peric', ...);

COMMIT;

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

ROLLBACK sledi_update;

END;

Poništavaju se efekti svih naredbi nakon specificirane SAVEPOINT tačke.



Isolation levels

- Nivo izolacije (isolation level) je sposobnost sistema da promene učinjene u jednoj operaciji učini nevidljivim za druge operacije.
- Oracle podržava tri nivoa izolacije:
 - READ COMMITTED
 - Podrazumevani nivo izolacije.
 - Ukoliko neka naredba transakcije zahteva lock koji već drži neka druga transakcija, naredba čeka da se taj lock oslobodi.
 - Transakcije se izvršavaju paralelno i izmene koje načini jedna transakcija nisu vidljive u drugim transakcijama sve dok se ne pozove COMMIT naredba.
 - Svaka naredba u transakciji vidi samo izmene koje su bile validne pre početka izvršavanja te naredbe.
 - U slučaju višestrukog izvršavanja iste naredbe u okviru jedne transakcije mogu se dobiti različiti rezultati.



Isolation levels

◦ SERIALIZABLE

- Transakcije se izvršavaju tako da izgleda kao da se izvršavaju jedna za drugom.
- Ukoliko transakcija sadrži naredbu koja zahteva lock koji već drži neka druga transakcija, onda se ta naredba završava greškom.
- Svaka naredba u transakciji vidi samo izmene koje su bile validne pre početka transakcije i izmene koje potiču od drugih naredbi u transakciji.
- Koriste se u slučaju sistema sa kraktim transakcijama koje ažuriraju mali broj vrsta, kada je mala verovatnoća da dve transakciju modifikuju istu vrstu i kod drugih transakcija koje ne menjaju podatke.

◦ READ ONLY

- Poseban slučaj SERIALIZABLE izolacije.
- Transakcija samo čita podatke iz baze podataka.
- Svaka naredba u transakciji vidi samo izmene koje su bile validne pre početka transakcije.
- Podržane su naredbe: podržavaju samo naredbe: SELECT INTO, OPEN, FETCH, CLOSE, LOCK TABLE, COMMIT, and ROLLBACK.
- FOR UPDATE u SELECT naredbi nije podržan.



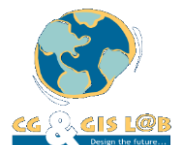
Isolation levels

- Fenomeni koji mogu da se jave kao posledica postojanja konkurentnih transakcija:
 - Dirty read – transakcija čita podatke koje je izmenila druga transakcija koja još uvek nije komitovana
 - Nonrepeatable (fuzzy reads) – Ukoliko transakcija pokuša da ponovo učitati podatke, koje je prethodno učitala, može se desiti da ti podaci budu modifikovani ili obrisani kao posledica neke druge komitovane transakcije.
 - Phantom reads – višestruko izvršavanje istog upita u okviru transakcije može da vraća različite rezultate.



Isolation levels

	Dirty reads	Fuzzy reads	Phantom reads
READ COMMITED	Ne	Da	Da
SERIALIZABLE	Ne	Ne	Ne
READ ONLY	Ne	Ne	Ne



Isolation levels

SET TRANSACTION

[ISOALTION LEVEL {READ COMMITED | SERIALIZABLE | READ ONLY}]

[USE ROLLBACK SEGMENT <rollback segment>]

[NAME <ime transakcije>];

- Naredba SET TRANSACTION se koristi da definiše svojstva nove transakcije odnosno najčešće da promeni isolation level transakcije.
- Transakcija ne mora da ima naredbu SET TRANSACTION na početku.
- Ukoliko postoji mora biti prva naredba transakcije odnosno naredba kojom transakcija započinje.
- USE ROLLBACK SEGMENT – pridružuje transakciju određenom rollback segmentu.
- NAME – omogućuje da se transakciji pridruži ime.



Isolation levels

DECLARE

```
daily_sales REAL;  
weekly_sales REAL;  
monthly_sales REAL;
```

BEGIN

```
COMMIT;
```

```
SET TRANSACTION ISOLATION LEVEL READ  
ONLY;
```

```
SELECT SUM(amt) INTO daily_sales  
FROM sales  
WHERE dte = SYSDATE;
```

```
SELECT SUM(amt) INTO weekly_sales  
FROM sales  
WHERE dte > SYSDATE - 7;
```

```
SELECT SUM(amt) INTO monthly_sales  
FROM sales  
WHERE dte > SYSDATE - 30;
```

```
COMMIT;
```

```
END:
```

Kraj prethodne transakcije.

Početak nove transakcije u READ ONLY modu.

READ ONLY transakcija ne ažurira podatke u bazi podataka.



Isolation levels

- Nivo izolacije se može promeniti na nivou čitave sesije korišćenjem naredbe ALTER SESSION.

```
ALTER SESSION SET ISOLATION_LEVEL READ COMMITTED;  
ALTER SESSION SET ISOLATION_LEVEL SERIALIZABLE;  
ALTER SESSION SET ISOLATION_LEVEL READ ONLY;
```

- Na ovaj način se definiše podrazumevani nivo izolacije za sve transakcije u okviru sesije.



Mehanizam zaključavanja

- Mehanizam zaključavanja omogućava korišćenje baze podataka u višekorisničkom okruženju bez opasnosti da će doći do narušavanja konzistentnosti podataka.
- Mehanizam zaključavanja sprečava mogućnost destruktivne interakcije između transakcija koje koriste isti resurs.
- U većini slučajeva Oracle automatski vodi računa o mehanizmu zaključavanja bez potrebe za ikakvom intervencijom od strane korisnika.
- Mehanizam automatskog zaključavanja se može zaobići na nivou transakcije.



Mehanizam zaključavanja

- Svi lock-ovi koje naredbe pribave u okviru jedne transakcije ostaju validni sve do kraja transakcije odnosno dok se transakcija ne potvrdi (commit-uje) ili u potpunosti ne poništi (rollback-uje).
- Po završetku transakcije Oracle oslobađa sve lock-ove koje su kreirale naredbe u okviru te transakcije.
- U slučaju rollback operacije do određene savepoint tačke Oracle oslobađa sve lock-ove koji su kreirani nakon te tačke.



Mehanizam zaključavanja

- Oracle podržava dva moda zaključavanja:
 - Ekskluzivni mod (Exclusive lock mode) – Onemogućava deljenje resursa između transakcija. Koristi se prilikom ažuriranja resursa. Prva transakcija koja pribavi ovakav lock je jedina transakcija u sistemu koja ima mogućnost izmene resursa sve dok lock ne bude oslobođen (odnosno dok se transakcija ne završi).
 - Deljivi mod (Shared lock mode) – Omogućava deljenje resursa u zavisnosti od operacije koja se nad tim resursom izvršava. Veći broj transakcija mogu da dele podatke prilikom čitanja. Međutim ove operacije drže lock nad podacima koje čitaju kako bi sprečile neku drugu transakciju da vrši izmene nad tim podacima.



Mehanizam zaključavanja

- Oracle podržava nekoliko tipova zaključavanja:
 - DML locks (data locks)
 - Row Level Locks
 - Table Level Locks
 - DDL locks (dictionary locks) – štiti definiciju objekta tokom izvršavanja DDL operacija (Npr. Prilikom kreiranja procedura ili funkcija štiti sve referencirane objekte)
 - Oracle Internal Locks/Latches – potpuno automatizovani mehanizam zaključavanja za Oracle-ove interne strukture podataka.
 - Oracle Distributed locks -
 - Oracle Parallel Cache Management Locks



Mehanizam zaključavanja

- Row level lock (TX) omogućava zaključavanje sloga ili skupa slogova za potrebe jedne transakcije.
- Row level lock je zaključavanje najnižeg nivoa koje obezbeđuje najbolju konkurentnos transakcijama koje se izvršavaju paralelno.
- Transakcije koje samo čitaju zaključane podatke ne moraju da čekaju transakciju koja modifikuje te podatke. Transakcije koje čitaju podatke rade sa originalnim kopijama podataka (kopijama podataka pre nego što je započela njihova izmena).



Mehanizam zaključavanja

- Vrsta koja se modifikuje je uvek zaključana u ekskluzivnom row level lock modu, tako da druge transakcije ne mogu da ažuriraju tu vrstu dok se započeta operacija ne završi.
- Transakcija koja pribavi row level lock nad nekom vrstom istovremeno pribavlja i table level lock nad tom tabelom kako bi bile sprečene DDL operacije.
- Svi DML lock-ovi koje Oracle automatski kreira su row level lock-ovi.
- Svaka transakcija koja sadrži naredbe INSERT, UPDATE, DELETE ili SELECT ... FOR UPDATE automatski pribavlja row level lock-ove.
- Ne postoji ograničenje što se tiče row level lock-ova koje jedna transakcija može da pribavi.



Mehanizam zaključavanja

```
DECLARE  
CURSOR radnici IS SELECT *  
           FROM PREDUZECE.RADNIK;  
FOR UPDATE OF PLATA NOWAIT;  
BEGIN  
  FOR radnik IN radnici LOOP  
    UPDATE RADNIK  
      SET PLATA = PLATA * 1.15  
      WHERE CURRENT OF radnici;  
END LOOP;  
  CLOSE radnici;  
  COMMIT;  
END;
```

FOR UPDATE identifikuje vrstu ili vrste koje će biti ažurirane i nad kojima treba pribaviti ekskluzivni lock.

NOWAIT specificira da komanda ne treba da čeka ukoliko su vrste već zaključane od strane neke druge transakcije.

Otvaranjem kursora transakcija pribavlja sve neophodne lock-ove.

Pribavljeni lock-ovi se ne oslobađaju zatvaranjem kursora već završetkom transakcije.



Mehanizam zaključavanja

```
SELECT *  
FROM PREDUZECE.RADNIK  
WHERE MATBR = 100  
FOR UPDATE OF PLATA;
```

```
UPDATE RADNIK  
SET PLATA = PLATA + 1014  
WHERE MATBR = 100;
```

Ukoliko se ove dve naredbe izvršavaju u različitim sesijama druga naredba će biti blokirana sve dok se u prvoj sesiji ne pozove naredba COMMIT ili ROLLBACK.



Mehanizam zaključavanja

- Transakcija pribavlja pribavlja Table level lock (TM) u sledećim situacijama:
 - kada se izvršavaju operacije ažuriranja podataka INSERT/UPDATE/DELETE
 - kada se vrši priprema za ažuriranje podataka SELECT ... FOR UPDATE
 - Eksplicitno korišćenjem naredbe LOCK TABLE.
- Table level lock sprečava DDL operacije nad tabelom. Table level lock sprečava sve ALTER TABEL ili DROP TABLE operaciju dokle god postoji aktivna transakcija koja drži lock nad tom tabelom.