

## 18 Database Transactions

- Control Transaction banke osigurava konzistentnost podataka
- Kako se proces menjanja podataka rukovodi i kako to utiče na db
- COMMIT, ROLLBACK i SAVEPOINT nisu podržani od Oracle APEX
- Svaki put kada se pošalje transakcija preko APEX, implicitan COMMIT se desi. Pošto je tako, ne postoji način za ROLLBACK da se desi

## Transactions

- Transakcije su osnovni koncept za sve db sisteme; one omogućavaju korisnicima da prave promene na podacima i onda odlučuju da li da sačuvaju ili odbace posao
- Db transakcije čini više koraka u jednoj logičkoj jedinici posla
- Transakcija se sastoji od jednog od sledećeg:
  - DML iskazi koji čine jednu konzistentnu promenu podataka
  - DML iskazi uključuju INSERT, UPDATE, DELETE i MERGE
  - Jedan DDL iskaz kao što je CREATE, ALTER, DROP, RENAME ili TRUNCATE
  - Jedan DCL iskaz kao što su GRANT ili REVOKE

## Transaction Analogy

- Db banke sadrži sume mušterijskih računa, kao i totalan balans depozita za druge ogranke
- Ako mušterija hoće da povuče i uradi transfer novca od njegovog računa i ostavi ga na drugi račun različitog ogranka
- Postoji nekoliko odvojenih koraka za ostvarivanje ove operacije
- Oba ogranka banke žele da se osiguraju da svi koraci u transakciji se dese ili da se nijedan od njih ne desi a ako sistem padne, transakcija ne sme da ostane delimično izvršena
- Grupisanje povlačenje i depozit u jednu transakciju omogućava ovu garanciju
- Transakcija se mora ili cela desiti ili da se uopšte ne desi

## Controlling Transactions

- Transakcije su kontrolisane korišćenjem iskaza:
  - COMMIT: Predstavlja tačku u vremenu gde korisnik je napravio sve promene koje želi da budu logički grupisane zajedno i pošto se nikakva greška nije desila, korisnik je spreman da sačuva posao (kada se COMMIT desi, trenutna transakcija se završava čineći sve promene permanentnim)
  - ROLLBACK: Omogućava korisniku da odbaci promene napravljene u db (kada se desi iskaz ROLLBACK, sve promene su odbačene)
  - SAVEPOINT: Kreira marker u transakciji, koji deli transakcije na manje delove
  - ROLLBACK TO SAVEPOINT: Omogućava korisniku da vrati trenutnu transakciju na specificiranu tačku stanja (ako se desila greška, korisnik može dati ROLLBACK TO SAVEPOINT iskaz čime se odbacuju samo one promene koje su napravljene posle SAVEPOINT)

## Transaction Example

- U sledećem primeru, korisnik je izdao UPDATE iskaz i odmah kreirao SAVEPOINT one

```
UPDATE copy_departments
SET manager_id= 101
WHERE department_id = 60;
```

```
SAVEPOINT one;
```

- Posle INSERT iskaza i UPDATE iskaza (sledeći primer), korisnik je shvatio da WHERE iskaz nije uključen u poslednji UPDATE
- Da bi ispravio grešku, korisnik je izdao ROLLBACK TO SAVEPOINT one
- The data is now restored to its state at SAVEPOINT one.

```
INSERT INTO copy_departments(department_id, department_name,
manager_id, location_id)
VALUES(130, 'Estate Management', 102, 1500);
```

```
UPDATE copy_departments
SET department_id = 140; ← WHERE clause omitted
```

```
ROLLBACK TO SAVEPOINT one;
```

```
COMMIT;
```

## When Does a Transaction Start or End ?

- Transakcija počinje sa prvim DML (INSERT, UPDATE, DELETE ili MERGE) iskazom
- Transakcija se završava kada se desi jedno od sledećih:
  - COMMIT ili ROLLBACK se dese
  - DDL (CREATE, ALTER, DROP, RENAME ili TRUNCATE) iskaz se desi
  - DCL (GRANT ili REVOKE) se desi
  - korisnik izađe normalno iz Oracle Db utility, izazivajući trenutnu transakciju da se implicitno izvrši
- Posle završetka jedne transakcije, sledeći izvršni SQL iskaz automatski startuje sledeću transakciju
- DDL iskaz ili DCL iskaz je automatski izvršen i zato implicitno se završava transakcija
- Svaki dan promena izvršena tokom transakcije je privremena sve dok se transakcija ne završi

## Data Consistency

- Zamislite nekoliko provedenih sati praveći promene u employee podacima samo da bi videli da neko drugi je uneo informaciju koja se sukobljava sa vašim promenama
- Da bi se sprečilo takav konflikt i da bi se omogućilo više korisnika da priđu db istovremeno, db sistem izdaje automatsku implementaciju nazvanu "read consistency"

## Read Consistency

- Read consistency (RC) garantuje konsistentnost pogleda na podatke za sve korisnike u svakom momentu
- Čitaoci ne vide podatke koji su u toku procesa menjanja
- Pisci su sigurni da promene u db su urađene na odgovarajući način

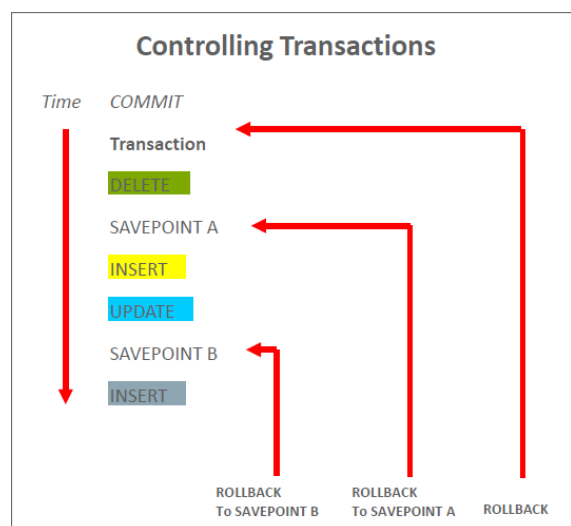
- Promene napravljene od jednog pisca neuništavaju ili imaju konflikt sa promenama drugog pisca
- RC je automatska implementacija
- Delimična kopija db se čuva u undo segmentima. Kada Korisnik A da jedan unos, updejt ili delete operaciju na db, Oracle Server uzima snapshot (kopiju) podataka pre nego je promenjena i ispisuje je na undo (rollback) segment
- Korisnik B još uvek vidi db kao što je bila pre promena; on vidi undo segment snapshot podataka
- Pre nego se promene primene u db, samo korisnik koji je promenio podatke vidi promene; svi drugi vide snapshot u undo segmentu
- Ovo garantuje da čitaoci podataka vide konsistentne podatke koje neće potkopati tekuće promene

### Changes Visible

- Kada DML iskaz je izvršen, promena napravljena u db postaje vidljiva svakome ko izvrši SELECT iskaz
- Ako transakcija je roll back, promene su poništene:
  - originalna, starija verzija podataka u undo segmentu je ponovo ispisana na tabeli
  - svi korisnici vide db kao što je postojala pre početka transakcije

### COMMIT, ROLLBACK and SAVEPOINT

- COMMIT i ROLLBACK obezbeđuju konzistentnost podataka, čineći mogućim i pregled promena podataka pre nego promene postanu trajne i grupisanje logičkih relacionih operacija
- SAVEPOINT kreira tačku u transakciji na koju se može vratiti bez moranja da undo celu transakciju
- COMMIT, ROLLBACK i SAVEPOINT su poznate kao Transaction Control Language ili TCL
- U transakciji na grafici, DELETE iskaz je izdat a onda SAVEPOINT A je ustanovljen
- Ovaj SAVEPOINT radi kao marker koji će dozvoliti korisniku da vrati bilo koju promenu napravljenu u podacima nazad na stanje podataka pošto ona postoji u tom trenutku
- U primeru, posle SAVEPOINT A, korisnik izdaje INSERT i UPDATE iskaze, a zatim daje novi rollback marker na SAVEPOINT B
- Ako iz bilo kog razloga korisnik ne želi da INSERT i/ili UPDATE iskazi da se dese, korisnik može dati ROLLBACK TO SAVEPOINT A iskaz
- Ovo će vratiti na stanje podataka kao što su bili na SAVEPOINT A markeru
- Dodavanje drugih SAVEPOINTS pravi dodatne markere za rollback tačke



- Ako korisnik izda ROLLBACK bez ROLLBACK TO SAVEPOINT iskaza, cela transakcija se završava i sve promene nad podacima se odbacuju

### Implicit Transaction Processing

- Automatsko izvršavanje promena podataka se dešava pod sledećim okolnostima:
  - DDL iskaz je izdat
  - DCL iskaz je izdat
  - korisnik izlazi normalno iz Oracle db utility, izazivajući trenutnu transakciju da bude implicitno izvršena
  - eksplicitno izdaje COMMIT ili ROLLBACK iskaze
- Automatski rollback se dešava pod nenormalnim završetkom Oracle db utility, ili kada sistemski greška se desi
- Ovo sprečava bilo koju grešku u u podacima da izazove neželjene promene u tabeli ispod
- Integritet podataka je stoga zaštićen

### Locking

- Važno je sprečiti podatke od promene sa više od jednog korisnika istovremeno
- Oracle koristi locks koji sprečavaju destruktivnu interakciju između transakcija prilazeći istim resursima, ili korisničkim objektima (tabele ili redovi) ili sistemskim objektima koji nisu vidljivi korisniku (deljene strukture podataka i DD redovi)
- Oracle locking se izvodi automatski i ne traži nikakvu akciju korisnika
- Implicitno zaključavanje se dešava za SQL iskaze ako je neophodno, u zavisnosti od tražene akcije
- Implicitno zaključavanje se dešava za sve SQL iskaze osim SELECT
- Korisnici mogu lock podatke ručno, što se naziva eksplicitno zaključavanje
- Kada COMMIT ili ROLLBACK iskazi se dese, lock na redove koji su u pitanju se desi