

Administracija baza podataka

Zaključavanje objekata baze podataka

Sadržaj

- ☐ Uvod
 - ☐ Nivoi izolacije transakcija
 - ☐ Granulacija i hijerarhija zaključavanja
 - ☐ Režimi zaključavanja
 - ☐ Kompatibilnost zaključavanja
 - ☐ Dinamičko zaključavanje
 - ☐ Prikazivanje informacija o zaključavanju
 - ☐ Deadlocking
-

Uvod

- Baza podataka je zajednički resurs za više programa istovremeno, što može izazvati probleme
 - Kao rešenje, koriste se *transakcije*
 - Pojam transakcije
 - Transakcija je jedno izvršenje neke “logičke jedinice posla”
 - Osobine transakcije: **A**tomicity, **C**onsistency, **I**solation, **D**urability (ACID)
-

Zaključavanje baze podataka (locking in)

- ❑ **Zaključavanje** je mehanizam koji se koristi od strane SUBP-a da sinhronizuje pristup više korisnika istim podacima u isto vreme.
 - ❑ Pre nego što transakcija počne da menja stanje podataka, prilikom čitanja ili ažuriranja, mora da se zaštiti od efekata druge transakcije koja pokušava da menja podatke istovremeno.
 - ❑ Transakcija ovo realizuje slanjem zahteva za zaključavanjem tog dela podataka.
 - ❑ Zaključavanje ima različite režime, kao što su **shared** (deljen) ili **exclusive** (isključivo za sebe).
 - ❑ Zaključavanje definiše nivo zavisnosti koji transakcija ima nad određenim podacima.
-

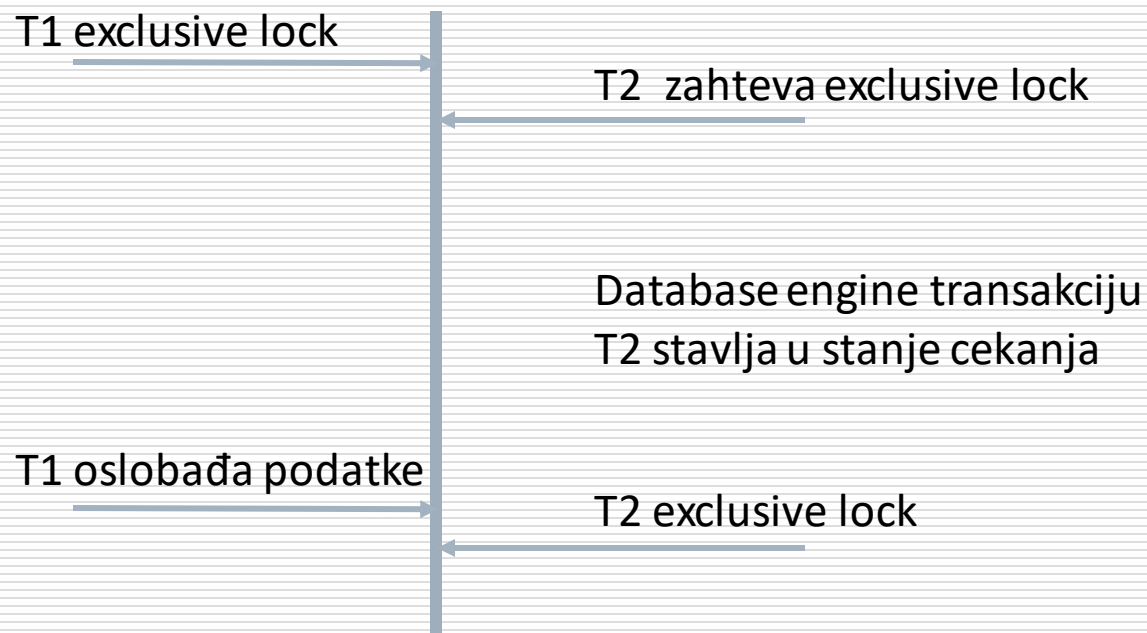
Zaključavanje baze podataka (locking in)

- ❑ **Zaključavanje** je mehanizam koji se koristi od strane SUBP-a da sinhronizuje pristup više korisnika istim podacima u isto vreme.
 - ❑ Pre nego što transakcija počne da menja stanje podataka, prilikom čitanja ili ažuriranja, mora da se zaštiti od efekata druge transakcije koja pokušava da menja podatke istovremeno.
 - ❑ Transakcija ovo realizuje slanjem zahteva za zaključavanjem tog dela podataka.
 - ❑ Zaključavanje ima različite režime, kao što su **shared** (deljen) ili **exclusive** (isključivo za sebe).
 - ❑ Zaključavanje definiše nivo zavisnosti koji transakcija ima nad određenim podacima.
-

Zaključavanje baze podataka

- ❑ Transakciji se ne može dodeliti pravo zaključavanja koje bi prouzrokovalo konflikt sa modom zaključavanja koji je druga transakcija već dobila.
 - ❑ Ako transakcija zahteva mod zaključavanja koji će prouzrokovati konflikt sa zaključavanjem koje je već dodeljeno nad istim podacima, SUBP će zahtevanu transakciju staviti u status cekanja sve dok se ne oslobode podaci.
 - ❑ Primer: T1 ima **exclusive** pravo nad podacima. T2 traži **exclusive** pravo nad istim podacima. Database engine transakciju T2 stavlja u stanje cekanja. T1 oslobađa podatke. Database engine dodeljuje **exclusive** pravo nad podacima za transakciju T2.
-

Zaključavanje baze podataka



Zaključavanje baze podataka

- ❑ Kada transakcija ažurira podatke, ona ih drži zaključanim štiteći ih od promena sve dok se transakcija ne završi.
 - ❑ Vreme koliko transakcija drži podatke zaključanim zavisi od podešavanja kojima je definisan nivo izolacije transakcije.
 - ❑ Svako zaključavanje se oslobađa prilikom završetka transakcije (bilo u slučaju commit-a ili rollback-a).
-

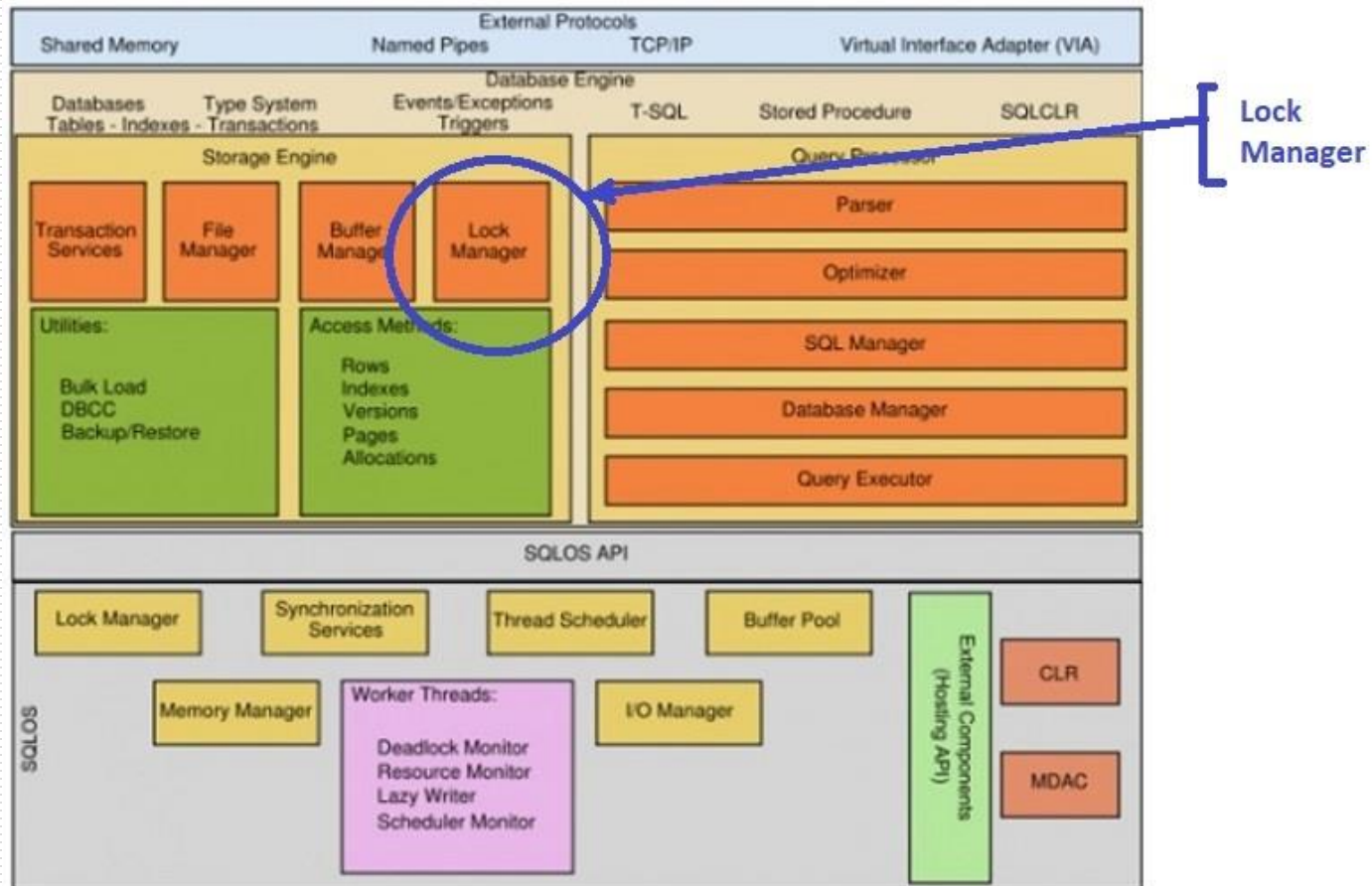
Zaključavanje baze podataka

- ❑ Aplikacije obično ne zahtevaju zaključavnje podataka. Zaključavanjima se upravlja interno, komponentom Database Engine koji se zove **lock manager**.
 - ❑ Kada istanca Database Engine procesira SQL upit, Database Engine Query procesor određuje kojim reursima može da se pristupa.
-

Zaključavanje baze podataka

- ❑ Query procesor odlučuje koji tipovi zaključavanja su neophodni da bi se zaštitio svaki resurs.
 - ❑ Odlučivanje o tipu zaključavanja je zasnovano na tipu pristupa i nivou izolacije.
 - ❑ Query procesor dalje zahteva određeno zaključavanje od ***lock manager-a***.
 - ❑ Lock manager odobrava zaključavanje, samo ako ne postoji mogućnost da dodje do konflikta zbog neke druge transakcije.
-

Arhitektura SUBP – Lock Manager



Optimističko i pesimističko zaključavanja (1/2)

❑ Database locking, Optimistic and Pessimistic locking

- **Locking** (zaključavanje, blokiranje) je mehanizam koji koristi SUBP da sinhronizuje pristup više korisnika istim podacima u istom vremenskom intervalu. Kada transakcija modifikuje deo podataka, ona postavlja “*katanac*” nad tim podacima, štiteći izmenu do kraja transakcije.
 - Savremeni SUBP sistemi uobičajeno obezbeđuju minimalno restriktivan nivo koji garantuje očuvanje konzistentnosti baze podataka u višekorisničkom režimu rada, tkzv. “**optimističko zaključavanje**” (engl. Optimistic locking).
-

Optimističko i pesimističko zaključavanja (2/2)

- ❑ Međutim, korisnik SUBP može sprovoditi eksplicitno zaključavanje resursa i pri tome samo može pooštriti restriktivnost zaključavanja koju nameće SUBP, tkzv. ***“pesimističko zaključavanje”*** (engl. Pessimistic locking), koje se postiže narednom LOCK TABLE.
-

Paralelnoizvršavanje transakcija

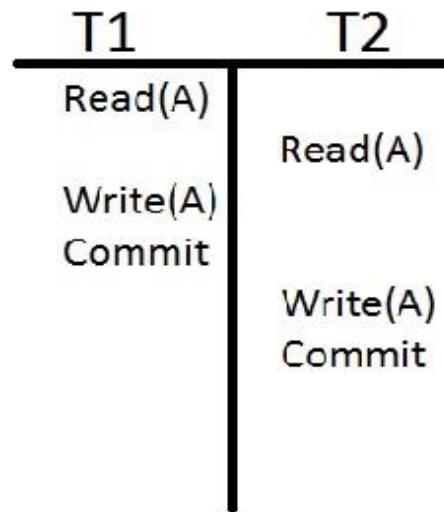
- ❑ SUBP treba da obezbedi adekvatno upravljanje paralelnim izvršavanjem transakcija kako one nepoželjno mešale svoja dejstva jedna na drugu.
 - ❑ U osnovi postoje tri slučaja kada je transakcija samostalno korektna, ali zbog mešanja transakcija može nastati pogrešan rezultat:
 1. Izgubljena ažuriranja
 2. Zavisnost od privremenih ažuriranja
 3. Narušavanje serijabilnosti
-

Problem izgubljenog ažuriranja

❑ Konkurentno izvršavanje transakcija i problemi

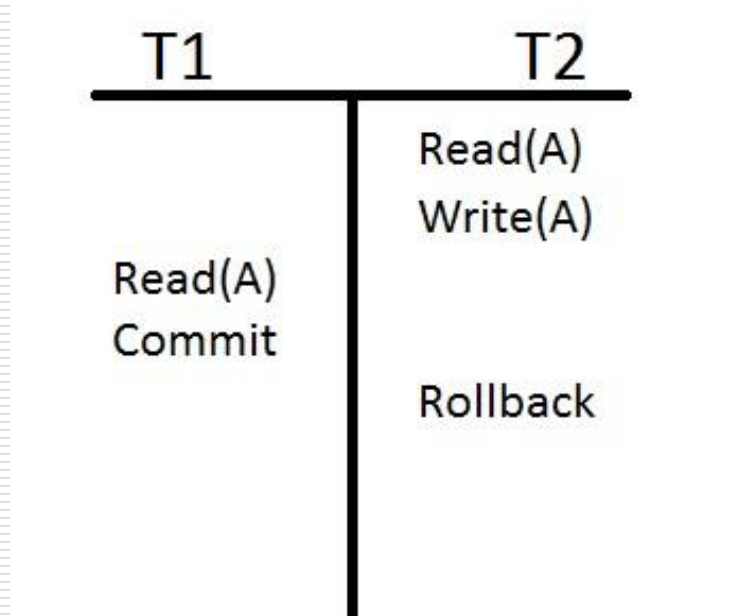
Konkurentan rad znači da više transakcija ima pristup istim podacima, istovremeno. Problemi:

- **problem izgubljenih ažuriranja**



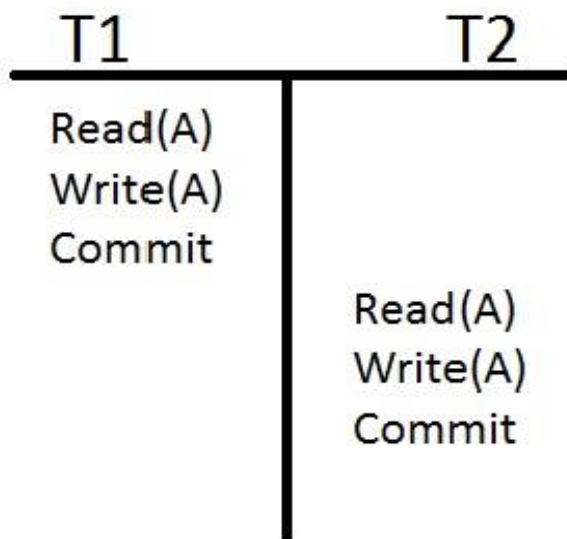
Zavisnost od privremenih ažuriranja

- Problem zavisnosti od poništenog ažuriranja

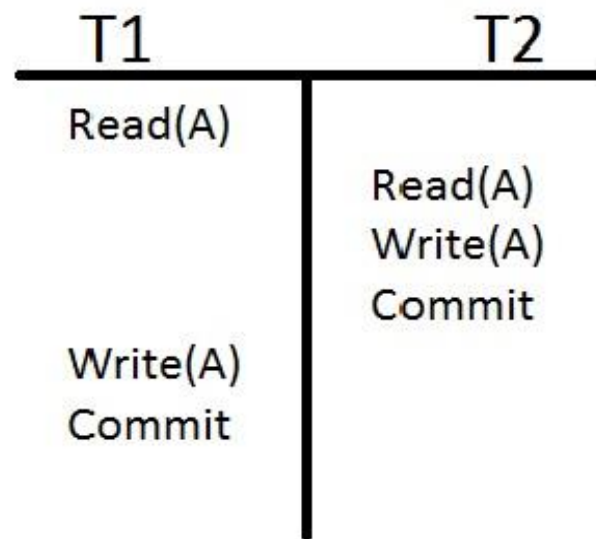


Narušavanje serijabilnosti

- Problem nekonzistentne analize



Baza je konzistentna!



Baza nije konzistentna!

Ovi problemi se rešavaju sistemom zaključavanja (Database locking)

Nivoi izolacije transakcija

- ❑ Pomenuti konflikti se u okviru SUBP i njegove administracije kontrolišu tako što se definiše **Nivo izolacije transakcija** (*Isolation levels*)
 - ❑ Definiše se nivo vidljivosti promena koje su napravljene u jednom procesu drugim procesima.
 - ❑ Nivo izolacije (*Isolation levels*) u SUBP definiše koliko će promene koje se dešavaju u jednom procesu biti vidljive drugim procesima koji pristupaju istoj bazi podataka.
 - ❑ Postoji nekoliko nivoa izolacije u DBMS-u, a oni definišu kako se vrši kontrola konflikata u višekorisničkom okruženju.
-

Nivo izolacije transakcije

- ❑ ***Isolation levels*** su nivoi izolacije koji se koriste kako bi se regulisalo ponašanje transakcija u odnosu na ostale transakcije koje se izvršavaju u isto vreme.
 - ❑ Omogućavaju definisanje nivoa vidljivosti podataka između različitih transakcija i utiču na to kako će se podaci prikazati kada se izvrši upit, u zavisnosti od toga koje su druge transakcije bile aktivne u isto vreme.
 - ❑ Postoje četiri standardna nivoa izolacije transakcija, koji su definisani u ANSI/ISO SQL standardu. Oni su:
 1. Read Uncommitted (nivo izolacije čitanja nekomitovanih podataka)
 2. Read Committed (nivo izolacije čitanja komitovanih podataka)
 3. Repeatable Read (nivo izolacije ponovljivog čitanja)
 4. Serializable (nivo izolacije serijalizabilnosti)
-

Nivoi izolacije transakcije

1. Read Uncommitted (čitanje nekomitovanih podataka)
 - Najmanje restriktivan nivo izolacije.
 - Transakcija može da čita i prikaže podatke koji nisu još uvek komitovani u bazi podataka.
 - Može da dovede do prikaza nekonzistentnih podataka, jer se druga transakcija može izvršiti između dva čitanja istih podataka.
 2. Read Committed (čitanje komitovanih podataka)
 - Najčešće koristi u bazama podataka.
 - transakcija može da čita samo one podatke koji su već komitovani u bazi podataka.
 - Obezbeđuje da će druga transakcija videti samo komitovane podatke.
-

Nivoi izolacije transakcije

3. Repeatable Read (ponovljivo čitanje)

- Transakcija može da čita podatke koji su bili prisutni u bazi podataka na početku transakcije.
- Bilo kakve promene koje se dešavaju u drugim transakcijama biti ignorisane,
- Transakcija ponašati kao da se ništa nije promenilo.

4. Serializable (serijalizabilnost)

- Najrestriktivniji nivo izolacije.
 - Transakcije se izvršavaju jedna po jedna.
 - Sve druge transakcije će biti blokirane dok se jedna transakcija ne završi. Ovo obezbeđuje najviši stepen konzistentnosti podataka
 - Skup u smislu performansi
-

Granularnost i hijerarhija zaključavanja

- ❑ Database Engine ima multigranularno zaključavanje koje omogućava različitim tipovima resursa da budu zaključani određenom transakcijom.
 - ❑ Da bi smanjili troškove zaključavanja, Database Engine zaključava resurse automatski na nivou koji je prikladan tom zadatku.
 - ❑ Zaključavajući manju granularnost resursa, kao što su redovi u tabeli, povećava se konkurentnost ali se troškovi povećavaju takođe jer više zaključavanja moramo obraditi istovremeno.
 - ❑ Zaključavanje većih granularnosti, kao što su tabele, smanjićemo opšte troškove ali ćemo zaključati celu bazu, odnosno niko neće moći da pristupi bilo kojem delu te baze, čime ćemo smanjiti konkurentnost.
-

Nivo izolacije transakcije

- ❑ *Isolation levels* su nivoi izolacije koji se koriste kako bi se regulisalo ponašanje transakcija u odnosu na ostale transakcije koje se izvršavaju u isto vreme.
 - ❑ Omogućavaju definisanje nivoa vidljivosti podataka između različitih transakcija i utiču na to kako će se podaci prikazati kada se izvrši upit, u zavisnosti od toga koje su druge transakcije bile aktivne u isto vreme.
 - ❑ Postoje četiri standardna nivoa izolacije transakcija, koji su definisani u ANSI/ISO SQL standardu. Oni su:
 1. Read Uncommitted (nivo izolacije čitanja nekomitovanih podataka)
 2. Read Committed (nivo izolacije čitanja komitovanih podataka)
 3. Repeatable Read (nivo izolacije ponovljivog čitanja)
 4. Serializable (nivo izolacije serijalizabilnosti)
-

Nivoi izolacije transakcije

1. Read Uncommitted (čitanje nekomitovanih podataka)

- Najmanje restriktivan nivo izolacije.
- Transakcija može da čita i prikaže podatke koji nisu još uvek komitovani u bazi podataka.
- Može da dovede do prikaza nekonzistentnih podataka, jer se druga transakcija može izvršiti između dva čitanja istih podataka.

2. Read Committed (čitanje komitovanih podataka)

- Najčešće koristi u bazama podataka.
 - transakcija može da čita samo one podatke koji su već komitovani u bazi podataka.
 - Obezbeđuje da će druga transakcija videti samo komitovane podatke.
-

Nivoi izolacije transakcije

3. Repeatable Read (ponovljivo čitanje)

- Transakcija može da čita podatke koji su bili prisutni u bazi podataka na početku transakcije.
- Bilo kakve promene koje se dešavaju u drugim transakcijama biti ignorisane,
- Transakcija ponašati kao da se ništa nije promenilo.

4. Serializable (serijalizabilnost)

- Najrestriktivniji nivo izolacije.
 - Transakcije se izvršavaju jedna po jedna.
 - Sve druge transakcije će biti blokirane dok se jedna transakcija ne završi. Ovo obezbeđuje najviši stepen konzistentnosti podataka
 - Skup u smislu performansi
-

Granularnost i hijerarhija zaključavanja

❑ Šta je granulacija i hijerarhija zaključavanja?

- Zaključavanje se može zahtevati za različite vrste resursa, kao što su redovi, stranice, indeksi, tabele, ili baze podataka. Neke operacije zahtevaju postavljanje “katanaca” na više nivoa granularnosti, formirajući *hijerarhiju zaključavanja*.
 - Na primer, da bi se potpuno zaštitilo čitanje nekog indeksa, instanca Database Engine-a mora da zaključa određene redove, i da zaključa određene stranice i tabele.
 - Microsoft SQL Server Database Engine ima multigranularno zaključavanje što omogućava da transakcija zaključa različite vrste resursa.
-

Granularnost i hijerarhija zaključavanja

- Koje resurse SUBP može zaključati?

Granularnost i hijerarhija zaključavanja

Resurs	Opis
RID	Identifikator reda - zaključa samo jedan red
KEY	Katanac nad redom unutar indeksa – štiti ključne opsege
PAGE	8-kilobajtna strana - podaci ili indeksne stranice.
EXTENT	Susedna grupa od 8 stranica - podaci ili indeksne stranice.
HoBT	Katanac štiti B-tree (indeks) ili skup stranica podataka u tabeli
TABLE	Cela tabela uključujući sve podatke I indekse.
FILE	Fajl baze podataka
APPLICATION	Resurs određen od strane aplikacije
METADATA	Zaključavanje metapodataka
ALLOCATION_UNIT	Alokacija jedinica
DATABASE	Cela baza podataka

Režimi zaključavanja

- ❑ Čemu služe režimi zaključavanja?
 - Zaključavanje ima različite režime koji specificiraju koji nivo pristupa imaju druge transakcije zaključanim resursima. Microsoft SQL Server Engine zaključava resurse koristeći različite režime zaključavanja koji određuju kako resursima mogu pristupiti transakcije u istom vremenskom intervalu.
-

Koje režime zaključavanja koristi Database Engine?

Režim	Opis
Shared(S)	Koristi se za operacije čitanja koje ne menjaju i ne ažuriraju podatke, kao što je SELECT naredba.
Update(U)	Koristi se nad resursima prilikom ažuriranja. Sprečava čest oblik zastoja (engl. deadlock) do kojeg dolazi kada više sesija čita, zaključava, i potencijalno ažurira resurse.
Exclusive(X)	Služi za operacije modifikovanja podataka, kao što su INSERT, UPDATE ili DELETE. Obezbeđuje da se više ispravki ne mogu desiti u isto vreme nad istim resursom.
Intent	Koristi se za uspostavljanje hijerarhije zaključavanja. Vrste intentova su: Intent shared (IS), Intent exclusive(IX) i shared with intent exclusive(SIX).
Schema	Koristi se kada operacija zavisi od šeme tabele koja se izvršava. Vrste schema zaključavanja su: schema modification (Sch-M), i schema stability (Sch-S).
Bulk Update	Koristi se kada se najveći deo podataka kopira u tabelu a TABLELOCK hint (nagoveštaj) je specificiran.
Key Range	Štiti opseg redova pročitanih upitom. Osigurava da druge transakcije ne mogu umetnuti redove koji bi se mogli kvalifikovati za upit serijalizovane transakcije ukoliko bi se taj upit izvršio ponovo.

Kompatibilnost zaključavanja

- ❑ *Kompatibilnost zaključavanja* kontroliše da li više transakcija mogu steći “katance” nad istim podacima u isto vreme.
 - ❑ Ukoliko je resurs već zaključan od strane druge transakcije, zahtev za novim zaključavanjem se može odobriti samo ukoliko je zahtevano zaključavanje kompatibilno sa postojećim režimom.
-

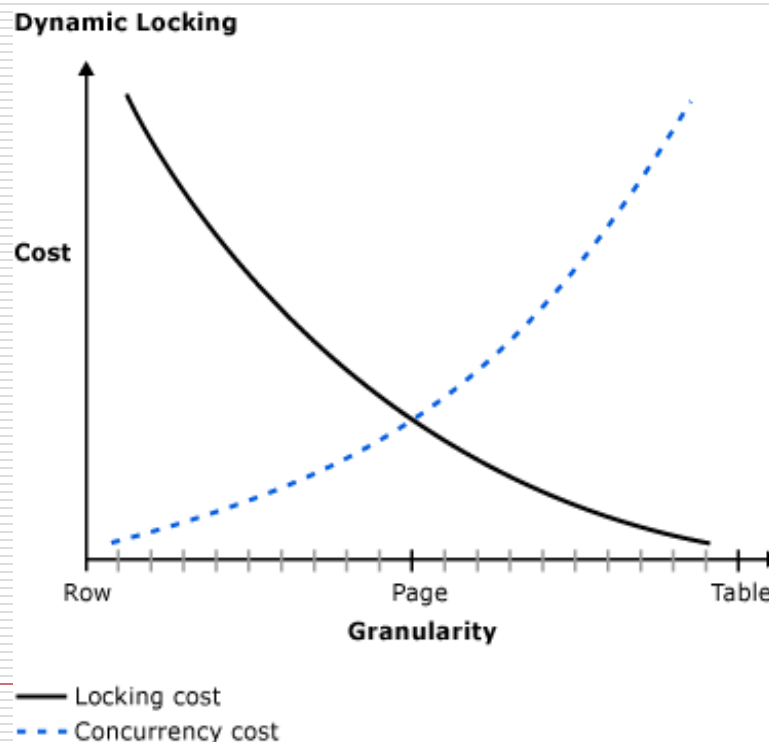
Kompatibilnost zaključavanja

- Matrica kompatibilnosti zaključavanja

režim	Postojeći odobreni					
	IS	S	U	IX	SIX	X
Zahtevani režim						
Intent Shared (IS)	Yes	Yes	Yes	Yes	Yes	No
Shared (S)	Yes	Yes	Yes	No	No	No
Update (U)	Yes	Yes	No	No	No	No
Intent Exclusive (IX)	Yes	No	No	Yes	No	No
Shared with Intent Exclusive (SIX)	Yes	No	No	No	No	No
Exclusive (X)	No	No	No	No	No	No

Dinamičko zaključavanje

- ❑ Šta je dinamičko zaključavanje?
 - Microsoft SQL Server Database Engine koristi strategiju dinamičkog zaključavanja za utvrđivanje najisplativijih “katanaca”.



Dinamičko zaključavanje

Prednosti dinamičkog zaključavanja

- Pojednostavljena administracija baze podataka. Administratori bazi podataka ne moraju da prilagode “katance” eskalaciji pragova.
 - Povećane performanse. Database Engine minimizira sistemske troškove koristeći zaključavanje koje odgovara zadatku.
 - Developeri aplikacija se mogu koncentrisati na razvoj. Database Engine automatski prilagođava zaključavanje.
-

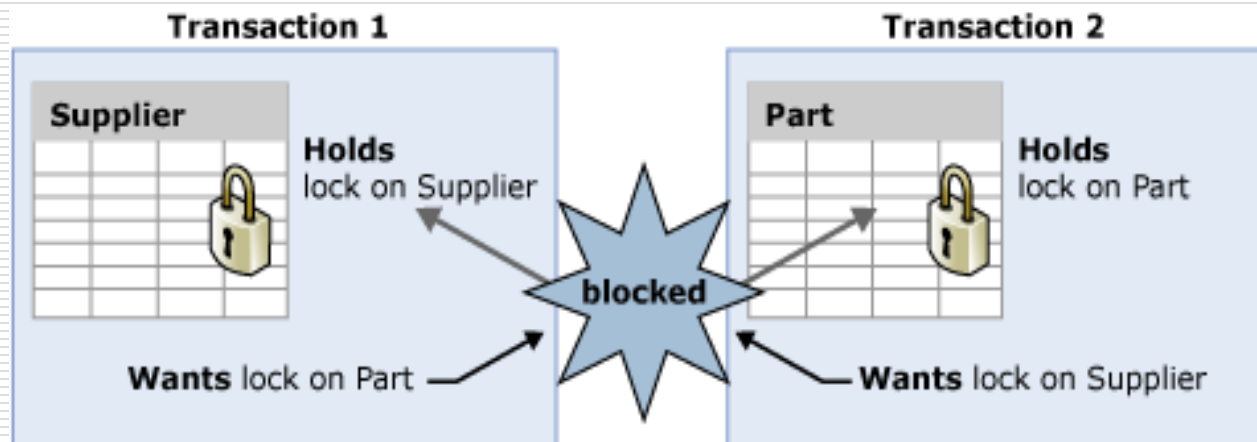
Prikaz informacija o zaključavanju

Tema	Opis
Lock Event Category	Koristeći SQL Server Profiler, možete odrediti kategoriju događaja da bi se prikupile informacije o događajima zaključavanja.
SQL Server, Lock Object	U System Monitor možete odrediti brojače iz objekta zaključavanja da prate nivo zaključavanja u instanci Database Engine-a.
sys.dm_tran_locks (Transact-SQL)	Možete upitom sys.dm_tran_locks dinamično upravljati pogledom da dobijete informacije o trenutnom stanju zaključavanja u instanci Database Engine-a.
EnumLocks	Aplikacija koja koristi SQL Server Management Objects API može dobiti listu aktivnih "katanaca" u instanci Database Engine-a koristeći EnumLocks metode na Server klasi.
EnumLocks	Aplikacija koja koristi SMO API može dobiti listu aktivnih "katanaca" u određenoj bazi podataka koristeći EnumLocks metodu klase Database.

Deadlock

❑ Šta je deadlock?

- *Deadlock* nastaje kada dve ili više transakcija trajno blokiraju jedna drugu tako što jedna transakcija drži “katanac” nad resursom kojeg ostale transakcije pokušavaju da zaključaju.



Deadlock

- ❑ Microsoft SQL Server Database Engine deadlock kontrolor periodično proverava zadatke koji su u zastoju odnosno na mrtvoj tački.
 - ❑ Ukoliko kontrolor detektuje cikličnu zavisnost, on bira jedan od zadataka kao žrtvu, i prekida tu transakcija sa greškom.
 - ❑ Ovo omogućava da drugi zadatak završi svoju transakciju.
 - ❑ Aplikacija sa transakcijom koja se završila sa greškom može ponovo pokušati izvršiti transakciju, što se obično desi kada se druga transakcija na mrtvoj tački, završi.
-

Deadlock

- ❑ Rukovanje sa deadlock-ovima
 - Kada instanca Microsoft Sql Server Database Engine-a izabere transakciju kao žrtvu deadlock-a, ona uništava trenutne serije, izvršava rollback i vraća aplikaciji poruku o greški 1205.
 - Aplikacija treba da ima error handler koji hvata poruku o grešci 1205.
 - Implementacija error handler-a koji hvata poruku o grešci 1205 omogućava aplikaciji da upravlja sa deadlock situacijom, i da preuzme mere za otklanjanje (npr, automatski ponovo poslati upit koji je bio uključen u deadlock).
 - Aplikacija treba kratko pauzirati do ponovnog slanja upita.
-