

Конкурентност

Ненад Митић

Математички факултет
nenad.mitic@matf.bg.ac.rs

5. decembar 2023.

Проблем зависности од непотврђених података

Први случај

Трансакција А	време	Трансакција В
-----	--	-----
-----	--	-----
-----	t1	UPDATE R
-----	--	-----
FETCH R	t2	-----
-----	--	-----
-----	t3	ROLLBACK
-----	--	-----

Трансакција А у тренутку t2 зависи од непотврђених промена

Проблем зависности од непотврђених података

Други случај

Трансакција А	време	Трансакција В
----	--	----
----	--	----
----	t1	UPDATE R
----	--	----
UPDATE R	t2	----
----	--	----
----	t3	ROLLBACK
----	--	----

Трансакција А у тренутку t2 ажурира податке чија промена није потврђена, и учињене промене се губе у тренутку t3

Проблем неузастопних читања

Трансакција А	време	Трансакција В
-----	---	-----
-----	---	-----
FETCH R	t1	-----
-----	---	-----
-----	t2	UPDATE R (DELETE R)
-----	---	-----
-----	t3	COMMIT
-----	---	-----
FETCH R	t4	-----
-----	---	-----

Трансакција А чита слог пре других захтева. У међувремену, В мења/брише слог и потврђује промене. Ако А касније пожели да прочита оригинални слог он ће бити промењен или неће постојати

Проблем појављивања фантома

Трансакција А	време	Трансакција В
-----	-----	-----
FETCH R where x=10	t1	-----
-----	-----	-----
---	t2	INSERT INTO R ... x=10
-----	--	(UPDATE R SET X=10 WHERE....)
-----	-----	-----
FETCH R where x=10	t3	-----
-----	-----	-----
-----	-----	-----

Трансакција А чита табелу по критеријуму и добија скуп слогова. Апликација В уноси нове слокове или ажурира постојеће тако да задовољавају критеријум задат од А. Апликација А поново чита табелу у оквиру исте трансакције и добија додатне ("фантомске") слокове

Проблем неконзистентне анализе

Трансакција А	време	Трансакција В	
-----	--	-----	
FETCH Rac1(40)	t1	-----	
sum=40	--		
-----	--	-----	
FETCH Rac2(50)	t2	-----	
sum=90	--		
-----	--	-----	
-----	t3	FETCH Rac3(30)	
-----	--	-----	
-----	t4	UPDATE Rac3	(30-->20)
-----	--	-----	
-----	t5	FETCH Rac1(40)	
-----	--	-----	
-----	t6	UPDATE Rac1	(40-->50)
-----	--	-----	
-----	t7	COMMIT	
-----	--	-----	
FETCH Rac3(20)	t8	-----	
sum=110	--		
-----	--	-----	

Проблем неконзистентне анализе

Три рачуна са стањем

Рачун 1: 40 Рачун 2: 50 Рачун 3: 30

Трансакција А сабира стање на сва три рачуна, док трансакција В преноси 10 са рачуна 3 на рачун 1.

Трансакција А примењује неконзистентну анализу: последица је добијање погрешне вредности за збир (110 уместо 120)

Проблеми у конкурентном раду

Проблеми који се јављају у претходним примерима су

Термин	Опис
Прљаво писање	Трансакција мења непотврђене податке које је променила друга трансакција која још није дала COMMIT или ROLLBACK
Прљаво читање	Трансакција чита податке модификоване од стране друге трансакције која још није дала COMMIT или ROLLBACK
Расплинуто читање / Не-узастопно читање	Трансакција која чита податке не види исте податке као у претходном читању (у оквиру те исте трансакције)
Читање фантома	Трансакција која чита податке види нове податке у односу на претходно читање

Закључавање

Претходни проблеми могу да буду решени преко механизма који се назива **закључавање** (постављање катанаца, енг. *locking*)

Основна идеја:

- када трансакција жели да ради са неким ресурсом она захтева закључавање тог ресурса (поставља катанац над тим ресурсом)
- када се трансакција заврши ресурс се ослобађа



Претпоставке

Елементарни приступ

- Систем подржава бар две врсте катанаца: приватни (ексклузивни, X) и дељиви (S)
- Ако је трансакција A поставила X катанац над ресурсом t тада захтев било које друге трансакције B за постављање било ког катанца над t бива одбијен
- Ако је трансакција A поставила дељиви катанац над торком t тада захтев било које друге трансакције B за постављање
 - S катанца над t може бити испуњен
 - X катанца над t бива одбијен

Претпоставке - наставак

Матрица компатибилности начина закључавања

		Transakcija A		
		X	S	-
Transakcija B	X	N	N	Y
	S	N	Y	Y
	-	Y	Y	Y

Катанац: X - приватни, S - дељиви, - не постоји

N означава конфликт

Y означава компатибилност

Протокол приступа подацима

Стриктни двофазни протокол закључавања гарантује да се проблеми конкурентности не јављају

- 1) Трансакција која жели да прочита торку мора прво над њом да постави S катанац
- 2) Трансакција која жели да ажурира торку мора прво над њом да постави X катанац. Алтернативно, ако већ држи S катанац над том торком, трансакција мора да захтева унапређење S катанца у X катанац

Протокол приступа подацима - наставак

- 3) Ако је захтев за постављањем катанца од стране трансакције В одбијен јер је у конфликту са већ постављеним катанцем од стране трансакције А, трансакција В иде у стање чекања. В чека док А не ослободи кључ (систем мора да гарантује да В неће заувек да остане у стању чекања!)
- 4) X катанац се задржава до краја трансакције (COMMIT или ROLLBACK). S катанац се, уобичајено, такође задржава (најдуже) до краја трансакције

Проблем изгубљених ажурирања - решење

Трансакција А	време	Трансакција В
-----	--	-----
-----	--	-----
FETCH R	t1	----- (захтева S катанац над R)
-----	--	-----
-----	t2	FETCH R (захтева S катанац над R)
-----	--	-----
UPDATE R	t3	----- (захтева X катанац над R)
чека	--	-----
чека	t4	UPDATE R (захтева X катанац над R)
чека	--	чека
чека	--	чека

Проблем изгубљених ажурирања - решење

- У тренутку T1 трансакција А поставља S катанац
- У тренутку T2 трансакција Б поставља S катанац
- У тренутку T3 трансакција А не може да постави X катанац и иде у стање чекања
- У тренутку T4 трансакција Б не може да постави X катанац и иде у стање чекања
- Ни једна трансакција не може да настави рад, обе чекају једна на другу - мртва петља (енг. *deadlock*)

Проблем зависности од непотврђених података - решење

Први случај

Трансакција А	време	Трансакција В
----	--	----
----	--	----
----	t1	UPDATE R (захтева X катанац над R)
----	--	----
FETCH R	t2	---- (захтева S катанац над R)
чека	--	----
чека	--	----
чека	t3	ROLLBACK (тачка синхронизације ослобађа се X катанац са R-a)
чека	--	----
FETCH R	t4	---- (поново се извршава и поставља S катанац над R)

Трансакција А је спречена да прочита непотврђене промене у тренутку t2

Проблем зависности од непотврђених података - решење

Drugi slucaj

Трансакција А	време	Трансакција В
----	--	----
----	--	----
----	t1	UPDATE R (захтева X катанац над R)
----	--	----
UPDATE R	t2	---- (захтева X катанац над R)
чека	--	----
чека	t3	ROLLBACK (тачка синхронизације ослобађа се X катанац са R-a)
чека	--	----
UPDATE R	t4	---- (поново се извршава и поставља X катанац над R)

Трансакција А је спречена да ажурира непотврђене промене у тренутку t2

Проблем неузастопних читања - решење

Трансакција А	време	Трансакција В
-----	---	-----
-----	---	-----
FETCH R	t1	----- (захтева S катанац над R)
-----	---	-----
-----	t2	UPDATE R (DELETE R)
-----	--	чека
-----	t3	чека
-----	--	чека
FETCH R	t4	чека
-----	--	чека

Трансакција А чита слог и поставља S катанац на R. Трансакција В не може да добије X катанац и чека, док А у каснијем читању добија исту вредност слога

Проблем појављивања фантома - решење

Transakcija A	vreme	Transakcija B
-----	-----	-----
FETCH R where x=10	t1	----- (захтева S катанац над R)
-----	--	-----
---	t2	INSERT INTO R ... x=10
-----	--	(UPDATE R SET X=10 WHERE....)
-----	--	чека
FETCH R where x=10	t3	чека
-----	--	чека
-----	--	чека

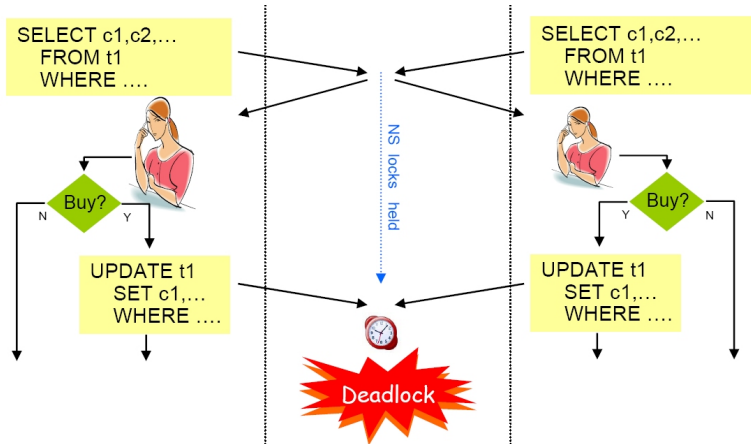
Трансакција А чита слог и поставља S катанац на R. Трансакција В не може да добије X катанац и чека. А у поновљеном читању добија исти скуп слогова (без "фантома")

Проблем неконзистентне анализе - решење

Transakcija A	време	Transakcija B
-----	-----	-----
----	--	----
----	--	----
FETCH Rac1(40)	t1	---- (поставља S катанац над Rac1)
sum=40	--	
----	--	----
FETCH Rac2(50)	t2	---- (поставља S катанац над Rac2)
sum=90	--	
----	--	----
----	t3	FETCH Rac3(30) (поставља S катанац над Rac3)
----	--	----
----	t4	UPDATE Rac3(30-->20) (поставља X катанац над Rac3)
----	--	----
----	t5	FETCH Rac1(40) (поставља S катанац над Rac1)
----	--	----
----	t6	UPDATE Rac1(40-->50) (захтева X катанац над Rac1)
----	--	чека
----	--	чека
FETCH Rac3(20)	t7	чека (захтева S катанац над Rac3)
чека	--	чека
чека	--	чека

Неконзистентна анализа је спречена али се у тренутку t7
јавила мртва петља

Мртва петља



Ко ће од корисника успети да обави куповину?

Мртва петља

- Мртва петља (енг. *deadlock*) је ситуација када две или више трансакција имају постављен катанац над ресурсом који је потребан оној другој тако да ни једна не може да настави са радом.
- Мртва петља може да укључи две или више трансакција
- Открива се графом чекања
- Разрешава се тако што се изабере жртва цији се ефекти пониште и тако ослободе ресурси

Серијализабилност

- Критеријум коректности извршавања датог скупа трансакција
- Извршавање датог скупа трансакција је коректно ако је серијализабилно, тј. ако производи исти резултат као и серијско извршавање истог скупа трансакција
- Ефекат закључавања је да исфорсира серијализабилност трансакција
- Серијализабилност скупа трансакција не гарантује редослед трансакција при извршавању!

Двофазни протокол закључавања

Важи следећа теорема

Ако све трансакције примењују двофазни протокол закључавања тада су сви могући распореди извршавања трансакција серијализабилни

Начин рада:

- Пре рада са било којим објектом трансакција мора да захтева катанац над тим објектом
- По ослобађању катанца трансакција не сме више да захтева било који катанац

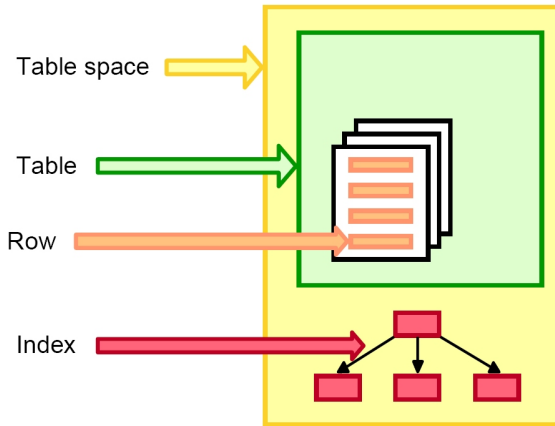
Двофазни протокол закључавања - наставак

Трансакције које се понашају у складу са овим протоколом имају две фазе

- фаза захтева за катанцима
- фаза укидања катанаца тј. ослобађања ресурса

Друга фаза се често сажима у једну операцију (COMMIT ili ROLLBACK)

Објекти закључавања



Закључавање и перформансе

Због перформанси и повећања нивоа конкурентности (могућности за истовремено извршавање више програма) проширује се елементарни концепт закључавања ресурса:

- подржава три опште категорије катанаца: S, U (ажурирање са намером), X и више начина за њихово коришћење
- уводи се концепт *нивоа изолације*
- уводе се додатне технике за повећање нивоа конкурентности

Ниво изолације

- Термин **ниво изолације** се користи за опис степена ометања које текућа трансакција може да поднесе при конкурентном извршавању
- Ако су трансакције серијализабилне степен ометања не постоји, тј. ниво изолације је максималан
- Реални системи због различитих разлога (нпр. перформансе) допуштају рад са нивоом изолације који је мањи од максималног
- Што је већи ниво изолације мање су допуштене сметње и обратно

Нивои изолације у DB2

Могући нивои изолације у DB2 су

- **RR** - Repeatable Read - закључава све слогево који су реферисани у оквиру трансакције
- **RS** - Read Stability - закључава слогево који су прочитани у оквиру трансакције
- **CS** - Cursor Stability (predefinisan) - закључава се сваки слог коме се приступило док се курсор налази на њему.
Катанци се држе док се не приступи наредном слогу
 - **CC** - currently committed - уколико неко чита слог, не чека да се ослободи кључ већ се чита претходно потврђена верзија података
- **UR** - Uncommitted Read - допушта читање непотврђених података друге трансакције

Нивои изолације у DB2

Карактеристике нивоа изолације се могу видети у
DB2 SQL Reference

SET CURRENT ISOLATION - промена нивоа
изолације за динамичке SQL наредбе

SQL код који се враћа у случају мртве петље је
-911

Начини закључавања

- Под начином закључавања подразумева се врста приступа закључаном објекту која је доступна конкурентној трансакцији.
- Протокол закључавање са намером (енг. *intent locking protocol*) - трансакцији није дозвољено да захтева катанац над торком пре првог захтева за постављањем катанца уопште
- Конфликт између захтева се открива на нивоу табела а не на нивоу торки

Начини закључавања - наставак

Начин закључавања	Објекат	Опис
IN (Intent None)	Простор за табеле, блокови, табеле, партиције	Власник катанца може да чита све (и непотврђене) податке у објекту, али не може да их мења. Остале апликације могу да читају или ажурирају табелу. Не поставља ни један катанац.
IS (Intent Share)	Простор за табеле, блокови, табеле, партиције	Власник катанца може да чита податке у закључаној табели али не може да их мења. Поставља S катанац. Остале апликације могу да читају или ажурирају табелу све док не ажурирају слог над којим је постављен S катанац
IX (Intent Exclusive)	Простор за табеле, блокови, табеле, партиције	Власник катанца може да чита (ако добије S или U катанац) и мења податке (ако добије X катанац) над слогом. Остале апликације могу да читају и ажурирају табелу све док не ажурирају слог над којим је постављен X катанац

Начини закључавања - наставак

Начин закључавања	Објекат	Опис
NS (Next key Share)	Слог	Власник катанца може да чита али не и да ажурира слог. Катанац се поставља над појединачним слоговима табеле уместо S катанца, када је ниво изолације RS или CS. Остале апликације могу да читају и ажурирају табелу све док не ажурирају слог над којим је постављен NS катанац
NW (Next Key Weak Exclusive:)	Слог	Катанац се поставља на следећи ред када је ред унет у индекс не-каталог табеле. Када се слог унет у индекс, на наредни слог се поставља HW катанац, ако је следећи слог закључан са RR претраживањем. Власник катанца може да чита али не и да ажурира закључани слог. Катанац је сличан X катанцу, али је компатибилан са NS катанцем
S (Share)	Слогови, блокови, табеле, партиције	Власника катанца и све конкурентен апликације могу да читају податке али не и да их ажурирају нити да добију катанац над слоговима. Поставља се C катанац над табелом
SIX (Share with Intent Exclusive)	Табеле, блокови, партиције	Власник катанца може да чита и ажурира податке ако добије X катанац. Код читања се не постављају катанци над слоговима. Остале апликације могу да читају податке али не и да их мењају. SIX катанац се добија конверзијом IX када се захтева C, и обратно

Начини закључавања - наставак

Начин закључавања	Објекат	Опис
U (Update)	Слогови, блокови, табеле, партиције	Власник катанца може да чита закључани објекат и може да унапреди U катанац у X катанац. Док се промена не оствари, остале апликације могу да читају али не и да ажурирају објекат
X (Exclusive)	Слогови, блокови, табеле, пул бафера, партиције	Власник катанца може да чита и ажурира објекат. Остале апликације могу да читају податке једино са UR нивоом изолације
Z (Super Exclusive)	Простори табела, табеле, партиције, блокови	Катанац се поставља над табелом у случају посебних догађаја (табела је промењена или избрисана, индекс над табелом је формиран или избрисан, или је табела у реорганизацији. Остале апликације не могу да приђу табели

Начини закључавања - наставак

Начини закључавања - наставак

Row Lock		Minimum* Supporting Table Lock
S	Share	IS
U	Update	IX
X	eXclusive	IX
W	Weak exclusive	IX
NS	Next key Share	IS
NW	Next key Weak exclusive	IX

An application does not acquire

Row locks

if it is using Table Locks of

S, U, X, or Z

Начини закључавања - наставак

Начини закључавања - наставак

IN	Intent None
IS	Intent Share
IX	Intent eXclusive
SIX	Share with Intent eXclusive
S	Share
U	Update
X	eXclusive
Z	superexclusive

Row Locking also used

Strict Table Locking

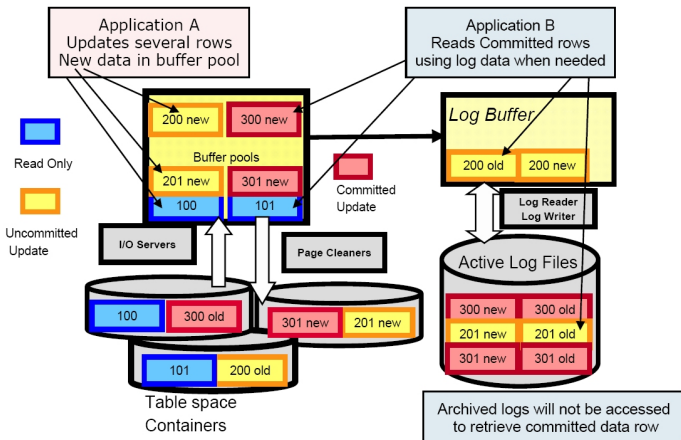
(See next page)

Компатибилност начина закључавања

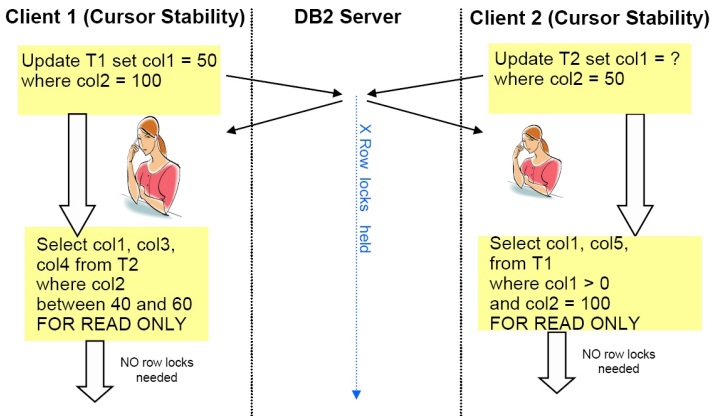
Матрица компатибилности начина закључавања

	IN	IS	NS	S	IX	SIX	U	NW	X	Z
IN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
IS	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
NS	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗
S	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗
IX	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗
SIX	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
U	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
NW	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
X	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Z	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

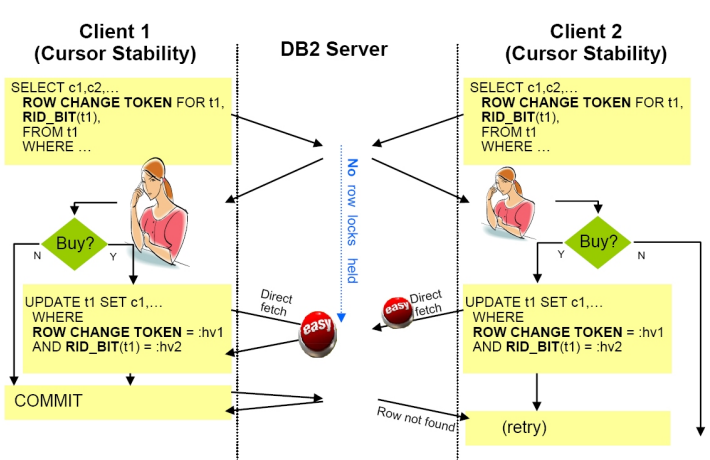
Тренутно потврђени (енг. Currently Committed)



Тренутно потврђени - елиминација мртве петље

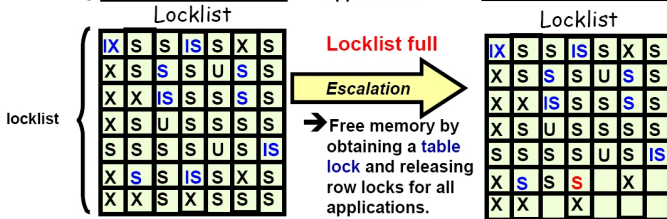
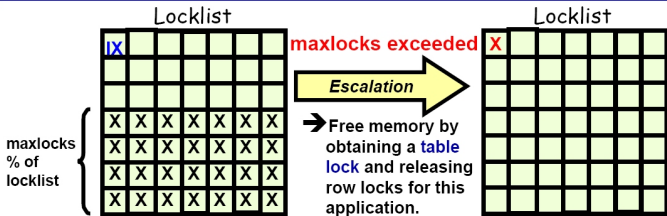


Оптимистичко закључавање

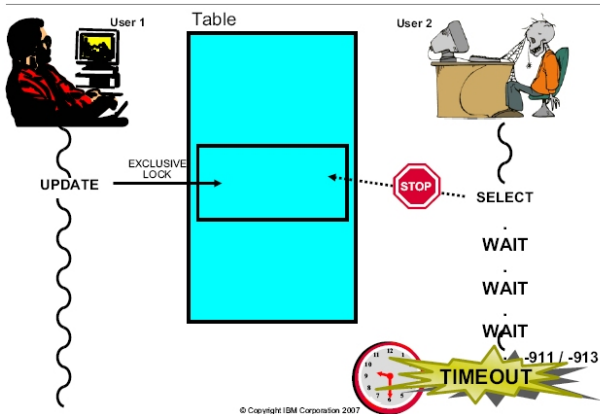


Ескалација катанаца

Legend: Row locks (black) Table locks (blue) Escalated table locks (red)



Timeout



Параметри базе који се односе на закључавање:
LOCKTIMEOUT, DLCHKTIME, MAXLOCKS,

SQL подршка - наставак

Стандард дефинише следеће нивое изолације

- READ UNCOMMITTED
- READ COMMITED
- REPEATABLE READ
- SERIALIZABLE (предефинисан)

SQL подршка - наставак

Ниво изолације		Прљаво писање	Прљаво читање	Неузастопно читање	Фантоми
DB2	ANSI				
UR	READ UNCOMMITTED	N	Y	Y	Y
CS	READ COMMITED	N	N	Y	Y
RS	REPEATABLE READ	N	N	N	Y
RR	SERIALIZABLE	N	N	N	N

SQL подршка - Db2

Закључавање табеле

- Корисник
 - LOCK TABLE <ime tabele> IN SHARE MODE
 - LOCK TABLE <ime tabele> IN EXCLUSIVE MODE
- Администратор
 - ALTER TABLE <ime tabele> LOCKSIZE ROW
 - ALTER TABLE <ime tabele> LOCKSIZE TABLE
 - ALTER TABLE <ime tabele> LOCKSIZE BLOCKINSERT
(samo za MDC)

Примери закључавања

Примери закључавања - 15.primeri.sql

- Чекање на ресурс
- Мртва петаља
- Експлицитне наредбе за закључавање
- SQL наредбе за анализу
- ...