Paweł Rajba
pawel@cs.uni.wroc.pl
http://pawel.ii.uni.wroc.pl/

SQL Server
Transakcje i blokady

Agenda

- Transakcje i blokady
  - Wprowadzenie do transakcji, rodzaje transakcji
  - Punkty zapisu, błędy w transakcjach
  - Zagnieżdżanie, zabronione instrukcje
  - Używanie transakcji
  - Problemy z równoległym dostępem
  - Kontrola dostępu równoległego
  - Poziomy odseparowania, blokady i zakleszczenia
  - Zasoby do blokowania, tryby blokad
  - Hierarchia blokad, eskalacje, wskazówki
  - Kompatybilność blokad, czas oczekiwania
  - Informacje o blokadach

Wprowadzenie

- Jedna transakcja może zawierać wiele wsadów, jeden wsad może zawierać wiele transakcji
- Transakcje spełniają własność ACID:
  - Niepodzielność (**A**tomicity)
  - Spójność (**C**onsistency)
  - Odseparowanie (**I**solation)
  - Wytrzymałość (**D**urability)
- W ramach transakcji zwykle umieszcza się instrukcje modyfikujące dane

Rodzaje transakcji

- Explicit (na żądanie)
  - Domyślny sposób zarządzania transakcjami
  - Transakcję rozpoczynamy poleceniem BEGIN TRANSACTION [WITH MARK ...]
  - Transakcję kończymy poleceniami COMMIT lub ROLLBACK

Rodzaje transakcji

- Implicit (domniemane)
  - Dowolna instrukcja rozpoczyna transakcję
    - Po zakończeniu transakcji kolejna instrukcja tworzy nową tran.
  - Nie są dozwolone transakcje zagnieżdżone
  - Transakcja musi być jawnie zakończona poleceniem COMMIT lub ROLLBACK
  - Domyślnie opcja ta jest wyłączona. Składnia:
    - SET IMPLICIT_TRANSACTIONS {ON | OFF}
  - Polecenia, które przy włączeniu opcji implicit transactions rozpoczną transakcję:
    - ALTER TABLE, INSERT, CREATE, OPEN, DELETE, REVOKE, DROP, SELECT, FETCH, TRUNCATE TABLE, GRANT, UPDATE

Rodzaje transakcji

- Autocommit
  - Domyślny tryb
  - Każda instrukcja jest osobną transakcją
    - Dopóki nie zostanie uruchomiona transakcja Explicit lub Implicit

Rodzaje transakcji

- Distributed transactions
  - Transakcja obejmująca np. kilka serwerów
  - Jest realizowana poprzez proces 2-phase commit
    - Faza przygotowania
      - Koordynator odpytuje wszystkie węzły, czy operacji transakcji udało się przeprowadzić (ale jeszcze nie ma ostatecznego zatwierdzenia)
    - Faza zatwierdzenia lub wycofania
      - Jeśli wszyscy potwierdzą → zatwierdzenie
      - Jeśli choć jeden nie potwierdzi → wycofanie

Punkty zapisu

- Są po to, żeby można było wycofać fragment transakcji, a nie wszystko
  - Tworzenie punktu zapisu odbywa się poprzez polecenie
    - SAVE TRANSACTION nazwa
  - Wycofanie do podane punktu odbywa się poprzez polecenie
    - ROLLBACK TRANSACTION nazwa_punktu_zapisu

Błędy w transakcjach

- Błędy umożliwiające wykonanie operacji powodują automatyczne wycofanie transakcji
- Opcja XACT_ABORT
  - SET XACT_ABORT { ON | OFF }

    ON → każdy runtime error wycofuje transakcje

    OFF → niektóre błędy nie wycofują całej transakcji, tylko pojedynczą instrukcję
- Zalecanych mechanizmem zarządzania błędami/wyjątkami jest instrukcja THROW i blok TRY .. CATCH

Zagnieżdżanie transakcji

- SQLServer pozwala na zagnieżdżanie instrukcji BEGIN TRANSACTION/COMMIT(ROLLBACK)
- Stopień zagnieżdżenia jest określony przez zmienną @@trancount (jeśli brak aktywnej transakcji – zmienna ma wartość 0)
- Uwagi:
  - Jeśli stopień zagnieżdżenia jest równy 1, wtedy zatwierdzane są transakcji ze wszystkich poziomów
  - Operację wycofania możemy wykonać na dowolnym poziomie zagnieżdżenia, jednak wycofywane są wszystkie transakcje

Zagnieżdżanie transakcji

- Inaczej mówiąc:
  - BEGIN TRAN – zwiększa @@trancount o 1
  - COMMIT – zmniejsza @@trancount o 1
  - ROLLBACK – ustawia @@trancount na 0
- Jeszcze jedna uwaga
  - Wywołanie COMMIT lub ROLLBACK w przypadku, gdy nie ma aktywnej transakcji, skończy się błędem

Zagnieżdżanie transakcji

- Nazewnictwo transakcji
  - Warto nazywać transakcję, ponieważ możemy lepiej panować nad kodem, jednak należy uważać:
    - jeśli wykonujemy ROLLBACK, to podana nazwa musi być nazwą transakcji z poziomu 1
    - wykonując COMMIT, podana nazwa nie ma znaczenia, ponieważ wykonywany tak naprawdę jest i tak COMMIT na poziomie 1
- UWAGA:
  - zagnieżdżonych transakcji należy w miarę możliwości unikać (mogą się pojawić kłopoty z blokadami)

Zabronione instrukcje

- Poniższych instrukcji nie można umieszczać w transakcjach:
  - ALTER DATABASE
  - BACKUP LOG
  - CREATE DATABASE
  - DROP DATABASE
  - RECONFIGURE
  - RESTORE DATABASE
  - RESTORE LOG
  - UPDATE STATISTICS

Używanie transakcji

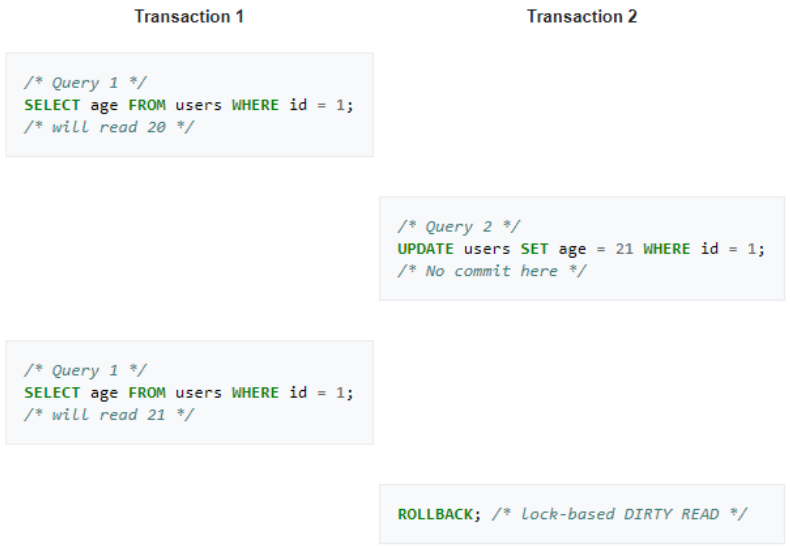
- Czas wykonania powinien być jak najkrótszy
  - należy przemyśleć użycie np. pętli while
  - czas blokad jest krótszy
- Nie należy w ramach transakcji czekać na dane od użytkownika
- Wszystkie niezbędne analizy danych należy wykonać przed rozpoczęciem transakcji
- Transakcja powinna uzyskiwać dostęp do minimalnego zbioru wierszy

Przykład

- 01-podstawowe-uzycie

Problemy z równoległym dostępem

- Dirty reads (uncommitted dependency)
  - Pierwsza osoba edytuje dane
  - W międzyczasie druga pobiera te częściowo zmienione
  - Następnie pierwsza osoba decyduje o wycofaniu zmian



https://en.wikipedia.org/wiki/Isolation_%28database_systems%29#Non-repeatable_reads

Problemy z równoległym dostępem

- Niepowtarzalność odczytu (nonrepeatable read)
  - Aka. *Inconsistent analysis*. Podobne do „brudne dane”
  - Czytamy dane, ktoś je modyfikuje, czytamy dane powtórnie
  - Wyniki obliczeń wykonane za pierwszym i drugim razem mogą być inne co prowadzi do niespójności

Transaction 1

```
/* Query 1 */
SELECT * FROM users WHERE id = 1;
```

Transaction 2

```
/* Query 2 */
UPDATE users SET age = 21 WHERE id = 1;
COMMIT; /* in multiversion concurrency
   control, or lock-based READ COMMITTED */
```

```
/* Query 1 */
SELECT * FROM users WHERE id = 1;
COMMIT; /* lock-based REPEATABLE READ */
```

https://en.wikipedia.org/wiki/Isolation_%28database_systems%29#Non-repeatable_reads

Problemy z równoległym dostępem

- Odczyty fantomów (phantom reads)
  - Dwa kolejne wykonania zapytania dają różne zbiory wierszy:

Transaction 1

```
/* Query 1 */
SELECT * FROM users
WHERE age BETWEEN 10 AND 30;
```

Transaction 2

```
/* Query 2 */
INSERT INTO users(id, name, age) VALUES (3, 'Bob', 27);
COMMIT;
```

```
/* Query 1 */
SELECT * FROM users
WHERE age BETWEEN 10 AND 30;
COMMIT;
```

  - Groźniejsze zapytanie: select avg(age) from users;

Kontrola dostępu równoległego

- Pesymistyczna
  - Polega na blokowaniu zasobów (tabel, wierszy, itd.)
  - Częsty w środowiskach dużej walki o zasoby
  - Stosowany, gdy koszt blokad jest niższy od wycofania transakcji, gdy pojawi się konflikt
- Optymistyczna
  - Polega na weryfikacji, czy podczas przetwarzania danych ktoś ich w międzyczasie nie zmienił
    - Zwykle realizowane przez wersjonowanie wierszy
  - Częsty w środowiskach małej walki o zasoby
  - Stosowany, gdy koszt (okazjonalny) wycofania transakcji jest niższy niż koszt blokowania zasobów przy czytaniu

Kontrola dostępu równoległego

- SQL Server wspiera różne sposoby
- Decyduje o tym
  - poziom izolacji transakcji lub
  - opcje współbieżności dla kursorów
- Ustawić to możemy
  - Na poziomie T-SQL-a
  - Atrybuty lub opcje API (ODBC, ADO.NET, itp.)

Poziomy odseparowania

OPARTE O BLOKADY

- READ UNCOMMITED
  - nie są nakładane żadne blokady
  - mogą się pojawić ,,brudne" odczyty
- READ COMMITED
  - nakładane są dzielone blokady przy czytaniu
  - ,,brudne" odczyty już nie pojawią
- REPEATABLE READ
  - gwarantowany jest brak ,,brudnych" odczytów i niepowtarzalnych odczytów
- SERIALIZABLE
  - nie pojawią fantomy

OPARTE O WERSJE WIERSZY

- READ COMMITED SNAPSHOT
  - To samo co READ COMMITED
  - Realizowany, jeśli włączona opcja bazy danych READ_COMMITTED_SNAPSHOT
- SNAPSHOT
  - Realizuje poziom analogiczny do SERIALIZABLE
  - Dostępna, gdy włączona opcja ALLOW_SNAPSHOT_ISOLATION

Poziomy odseparowania

- Do ustawienia poziomu odseparowania służy polecenie
  - SET TRANSACTION ISOLATION LEVEL
    { READ COMMITTED | READ UNCOMMITTED |
    REPEATABLE READ | SERIALIZABLE }
- Poziom odseparowania można sprawdzić przez polecenie DBCC USEROPTIONS

Poziomy odseparowania

- Podsumowanie

Isolation level	Dirty read	Nonrepeatable read	Phantom
Read uncommitted	Yes	Yes	Yes
Read committed	No	Yes	Yes
Repeatable read	No	No	Yes
Snapshot	No	No	No
Serializable	No	No	No

Blokady

- Blokują dostęp do danych, aby zsynchronizować dostęp wielu transakcji
- W przypadku konfliktu transakcja jest wstrzymywana i oczekuje na zwolnienie blokady innej transakcji
- Blokady są zwalniane po commit lub rollback
- Blokady są zarządzane przez lock managera automatycznie w oparciu o poziom izolacji
  - Czyli zwykle nie nakłada się blokad „ręcznie”, ale…
  - Można tym sterować poprzez tzw. „wskazówki”

Blokady

- Blokady mogą być nakładane na różnym poziomie granulacji
  - Większa granulacja →
    - Bardziej efektywny dostęp równoległy, ale
    - … większy koszty utrzymania blokad
  - Niższa granulacja →
    - Niski koszt utrzymania blokad, ale
    - … mniej efektywny dostęp równoległy
  - SQL Server stara się dobierać blokady odpowiednio na różnych poziomach
    - Tworzona jest tzw. lock hierarchy (np. na wierszu i tabeli)

Zakleszczenia

- Pojawiają się, gdy dwie transakcje wzajemnie czekają na blokowane przez siebie zasoby
- SQL Server kończy zakleszczenie poprzez wyznaczenie ,,ofiary zakleszczenia", czyli proces, który po prostu ubija; zwykle jest to ten młodszy proces
  - pojawia się wtedy komunikat o numerze 1205

Zakleszczenia

- Porady na zmniejszenie ryzyka powstania zakleszczenia
  - używać zasobów w transakcjach w tej samej kolejności
  - transakcje powinny być jak najkrótsze
    - czasowo
    - pod względem ilości kroków

Zasoby do blokowania

Resource	Description
RID	A row identifier used to lock a single row within a heap.
KEY	A row lock within an index used to protect key ranges in serializable transactions.
PAGE	An 8-kilobyte (KB) page in a database, such as data or index pages.
EXTENT	A contiguous group of eight pages, such as data or index pages.
HoBT	A heap or B-tree. A lock protecting a B-tree (index) or the heap data pages in a table that does not have a clustered index.
TABLE	The entire table, including all data and indexes.
FILE	A database file.
APPLICATION	An application-specified resource.
METADATA	Metadata locks.
ALLOCATION_UNIT	An allocation unit.
DATABASE	The entire database.

Tryby blokad

- Shared (S)
  - Blokada nakładana przy odczycie
    - Uniemożliwia modyfikacje przez inne transakcje
  - Czas trwania zależy od poziomu izolacji
    - Niskie poziomy → zwalniana od razu po odczycie wiersza
    - Wysokie poziomy (od repeatable read) → zwalniana po zakończeniu transakcji

Tryby blokad

- Exclusive (X)
  - Dają wyłączność dostępu do zasobu
    - Niezbędne dla operacji INSERT, UPDATE, DELETE
  - Tylko jedna transakcja może nałożyć taką blokadę
  - Problem: łatwo doprowadzić do zakleszczenia w repeatable read i serializable
    - T1: odczyt danych d, blokada S
    - T2: odczyt danych d, blokada S
    - T1: chęć modyfikacji d i próba nałożenia X
      - ale trzeba czekać, aż zwolni się T2.S
    - T2: chęć modyfikacji d i próba nałożenia X
      - ale trzeba czekać, aż zwolni się T1.S
    - … no i mamy dealock
  - Dlatego wprowadzony jest jeszcze jeden rodzaj blokady: U

Tryby blokad

- Update (U)
  - Wyraża chęć modyfikacji danych
  - Tylko jedna transakcja może nałożyć taką blokadę
  - Jeśli w poprzednim scenariuszu zostanie zastosowana U zamiast S:
    - Od razu jest wyrażona intencja czytania danych
    - W zależności od poziomu izolacji, druga transakcja będzie wstrzymana w odpowiednim momencie
  - Stosowana np. przy operacji UPDATE, gdzie częścią tej operacji jest SELECT

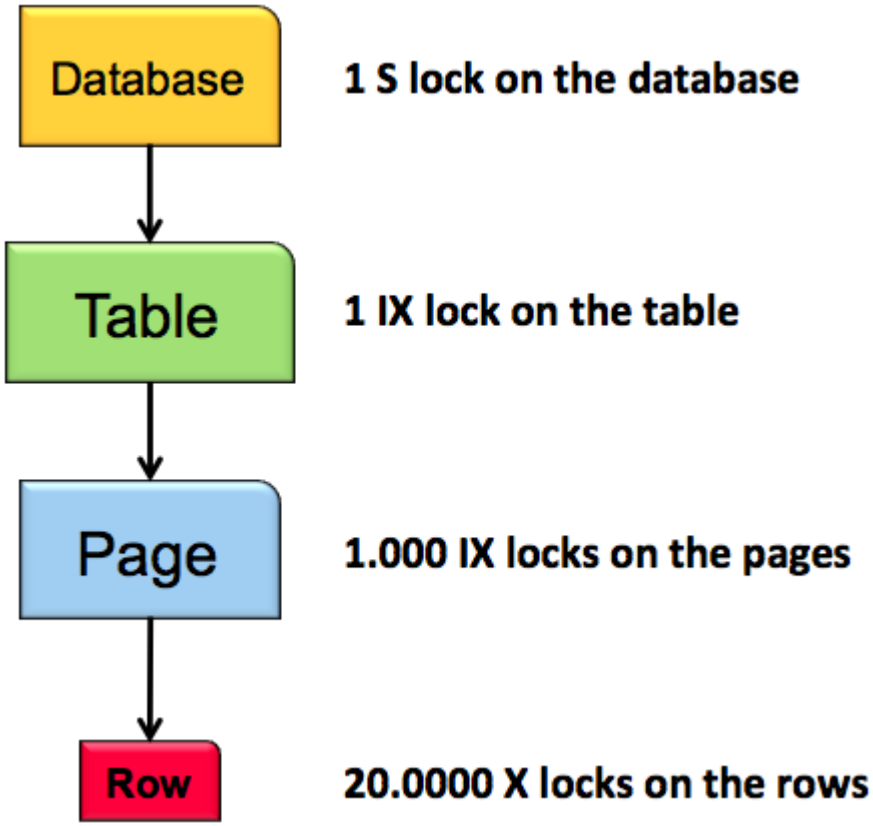
Tryby blokad

- Intent locks
  - Blokady zakładane na poziomie PAGE lub TABLE
  - Lock Manager może dzięki nim lepiej zarządzać blokadami
  - Wykorzystywane w eskalacji blokad
  - Mamy różne warianty
    - IS – transakcja zamierza czytać część zasobów przez nałożenie na nie blokady S
    - IX – transakcja zamierza modyfikować część zasobów poprzez nałożenie na nie blokady X
    - SIX – transakcja zamierza czytać wszystkie wiersze i niektóre z nich modyfikować (poprzez nakładanie odpowiednich blokad)
    - …

Tryby blokad

- Schema locks
  - Schema stability (Sch-S)
    - Używany podczas kompilacji zapytań
    - Uniemożliwia wykonanie poleceń, które wymagają Sch-M
  - Schema modification (Sch-M)
    - Używany podczas table DDL
    - Blokuje dostęp na czas modyfikacji tabel
- Bulk update locks
  - Używany podczas operacji masowych

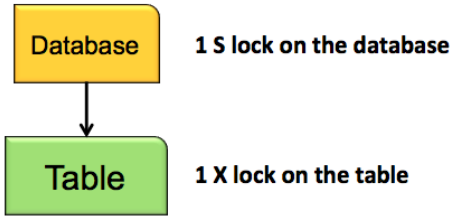
Hierarchia blokad



http://www.sqlpassion.at/archive/2014/02/25/lock-escalations/

Eskalacja blokad

- Chcemy usunąć 1000 wierszy
- Dwa podejścia
  - Blokada na każdym z 1000 wierszy
  - Blokada na tabeli (po eskalacji)
- Wady i zalety obu podejść?
- SQL Server stosuje oba podejścia
  - Granica to 5000 wierszy
  - Można to zmienić w ramach sesji
    - SET LOCK_ESCALATION 1000;
- Sterowanie eskalacją
  - ALTER TABLE table SET (LOCK_ESCALATION = { AUTO | TABLE | DISABLE }



Wskazówki dla blokowania

- Nadpisuje zachowanie zdefiniowane przez bieżący poziom izolacji
- Kilka wybranych
  - NOLOCK – nie blokuje przy operacjie SELECT
    - SELECT Title FROM Employee WITH (NOLOCK);
  - TABLOCK – blokuje całą tabelę
    - UPDATE Product WITH (TABLOCK) SET ListPrice = ListPrice * 1.10 WHERE ProductNumber LIKE 'BK-%';
  - UPDLOCK – nakłada blokadę UPDATE

Kompatybilność blokad

- Jeśli na zasób jest nałożona blokada, poniższa tabelka określa, która inna blokada może być założona dodatkowo:

	Existing granted mode					
Requested mode	**IS**	**S**	**U**	**IX**	**SIX**	**X**
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

Czas oczekiwania na zasób

- Opcja lock timeout
  - określa ile milisekund transakcja będzie czekała na dostęp do zasobu, zanim zostanie zwrócony błąd locking error
    - powyższy błąd nie powoduje przerwania transakcji
  - domyślna wartość to -1, czyli transakcja będzie czekać w nieskończoność
  - ustawienie opcji
    - SET LOCK_TIMEOUT milisekundy
  - ustalenie aktualnej wartości
    - SELECT @@lock_timeout

Informacje o blokadach

- Listę blokad możemy zobaczyć
  - sp_lock (SQL Server)
  - SELECT * FROM sys.dm_tran_locks (SQL Azure)
  - przydatne też sp_who2
- Znaczenie niektórych pól wyniku
  - Type – określa rodzaj blokowanego zasobu
    - DB – baza danych, EXT – obszar (extent)
    - TAB – tabela (table),
    - KEY – klucz w indeksie (key),
    - PAG – stronę (page),
    - RID – wiersz (row identifier).

Informacje o blokadach

- Znaczenie niektórych pól wyniku
  - Resource – adres blokowanego zasobu
    - Przykładowo: 1:528:0 – wiersz 0, na stronie 528 i pliku 1
  - Mode – tryb blokowania, który składa się z
    - shared (S), exclusive (X), intent (I),
    - update (U), or schema (Sch).
  - Status
    - GRANT – blokada uzyskana
    - WAIT – oczekiwana
    - CNVRT, CONVERT – w trakcie zmiany

Przykład

- 02-poziomy-izolacji-blokady
- 03-zakleszczenie

Literatura

- Transakcje
  http://msdn.microsoft.com/en-us/library/ms190612(v=sql.105).aspx
- Poziomy izolacji (+ćwiczenia)
  https://msdn.microsoft.com/en-us/library/cc546518.aspx
- Blokady
  http://msdn.microsoft.com/en-us/library/ms187101(v=sql.105).aspx
  https://technet.microsoft.com/en-us/library/ms190615(v=sql.105).aspx
  https://technet.microsoft.com/en-us/library/ms172429(v=sql.110).aspx
  http://www.sqlnotes.info/2012/10/10/update-with-updlock/
  http://www.sqlpassion.at/archive/2014/07/28/why-do-we-need-update-locks-in-sql-server/
  https://www.sqlshack.com/understanding-impact-clr-strict-security-configuration-setting-sql-server-2017/
  https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-locks-server-configuration-option?view=sql-server-ver15
  https://medium.com/swlh/sql-server-lock-escalation-dbf4779a76e8
  https://pl.seequality.net/sql-server-pare-slow-tablock/
  https://www.sqlshack.com/locking-sql-server/
  https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-tran-locks-transact-sql?view=sql-server-ver15
  https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-transaction-locking-and-row-versioning-guide?view=sql-server-ver15
  https://www.informit.com/articles/article.aspx?p=686168&seqNum=5
- Wskazówki
  https://technet.microsoft.com/en-us/library/ms187713(v=sql.105).aspx
- Artykuł przeglądowy
  https://technet.microsoft.com/pl-pl/library/jj856598(v=sql.110).aspx