TUGUN VAZIYATLARNI ANIQLASH. TUGUN VAZIYATLARNI HAL QILISH

Tojimamatov Israiljon Nurmamatovich
Farg`ona davlat universiteti
israiltojimamatov@gmail.com
Ne`matova Shohsanam Nodirbek qizi
Farg‘ona davlat universiteti talabasi
nematovashohsanam5@gmail.com

Anotatsiya: Bu mavzu ma'lumotlar bazasi tizimlarida, ayniqsa, tarqatilgan ma'lumotlar bazalari, replikatsiya stsenariylari va bir vaqtda kirish kontekstida **tugun vaziyatlarni (konfliktlarni) aniqlash va hal qilish**ning muhim jarayoniga bag'ishlangan. U vaziyatlarning keng tarqalgan sabablarini, ko'pincha bir vaqtda amalga oshirilgan yangilash operatsiyalari (yozish-yozish vaziyatlari), replikalar bo'ylab ma'lumotlar nomuvofiqligi yoki tarmoq bo'linishi hodisalaridan kelib chiqadigan sabablarni o'rganadi. Annotatsiya bu vaziyatlarni aniqlashning turli usullarini, shu jumladan bir vaqtda boshqarish mexanizmlaridan (masalan, bloklash, ko'p versiyali bir vaqtda boshqarish - MVCC), vaziyatni aniqlash algoritmlaridan va replikatsiya vaziyatini hal qiluvchilardan foydalanishni qamrab oladi. Bundan tashqari, u vaziyatlarni hal qilish strategiyalari va vositalarini, masalan, "oxirgi yozuvchi g'olib" siyosatlari, maxsus vaziyatni hal qilish mantig'i, sxema yoki ma'lumotlar farqlari uchun uch tomonlama birlashtirish va qo'lda aralashuvni ko'rib chiqadi. Ma'lumotlar bazalaridagi tugun vaziyatlarni tushunish va samarali boshqarish ma'lumotlar yaxlitligini saqlash, izchillikni ta'minlash va murakkab, tarqatilgan ma'lumotlar bazasi muhitida yuqori ish faoliyatini va samaradorlikni kafolatlash uchun nihoyatda muhimdir.

Kalit so‘zlar: Ma'lumotlar Bazasi Vaziyatlari, Tugun Vaziyatlari, Tarqatilgan Ma'lumotlar Bazalari, Replikatsiya Vaziyatlari, Bir Vaqtda Boshqarish, Vaziyatni Hal Qilish, Ma'lumotlar Izchilligi, Ma'lumotlar Yaxlitligi, Yozish-Yozish Vaziyatlari, MVCC (Ko'p Versiyali Bir Vaqtda Boshqarish), Bloklash, Vaziyatni Aniqdash, Birlashtirish Algoritmlari, Yuqori Ish Faoliyati, Tarqatilgan Tizimlar.

Аннотация: Эта тема посвящена критически важному процессу **выявления и разрешения конфликтов узлов** в системах баз данных, особенно в контексте распределенных баз данных, сценариев репликации и одновременного доступа. В ней рассматриваются распространенные причины возникновения конфликтов, которые часто связаны с одновременными операциями обновления (конфликты записи-записи), несогласованностью данных между репликами или событиями разделения сети. Аннотация

охватывает различные методологии обнаружения этих конфликтов, включая использование механизмов управления параллелизмом (например, блокировка, многоверсионный контроль параллелизма - MVCC), алгоритмы обнаружения конфликтов и средства разрешения конфликтов репликации. Кроме того, она исследует стратегии и инструменты для разрешения конфликтов, такие как политики "последний записывающий выигрывает", пользовательская логика разрешения конфликтов, трехстороннее слияние для различий в схеме или данных, а также ручное вмешательство. Понимание и эффективное управление конфликтами узлов в базах данных имеют первостепенное значение для поддержания целостности данных, обеспечения согласованности и гарантии высокой доступности и производительности в сложных, распределенных средах баз данных.

Ключевые слова: Конфликты Баз Данных, Конфликты Узлов, Распределенные Базы Данных, Конфликты Репликации, Управление Параллелизмом, Разрешение Конфликтов, Согласованность Данных, Целостность Данных, Конфликты Записи-Записи, MVCC (Многоверсионный Контроль Параллелизма), Блокировка, Обнаружение Конфликтов, Алгоритмы Слияния, Высокая Доступность, Распределенные Системы.

Annotation: This topic focuses on the critical process of **identifying and resolving node conflicts** within database systems, particularly in the context of distributed databases, replication scenarios, and concurrent access. It explores common causes of conflicts, which often arise from simultaneous update operations (write-write conflicts), data inconsistencies across replicas, or network partitioning events. The annotation covers various methodologies for detecting these conflicts, including the use of concurrency control mechanisms (e.g., locking, multi-version concurrency control - MVCC), conflict detection algorithms, and replication conflict resolvers. Furthermore, it examines strategies and tools for conflict resolution, such as "last-writer-wins" policies, custom conflict resolution logic, three-way merging for schema or data differences, and manual intervention. Understanding and effectively managing node conflicts in databases are paramount for maintaining data integrity, ensuring consistency, and guaranteeing high availability and performance in complex, distributed database environments.

Keywords: Database Conflicts, Node Conflicts, Distributed Databases, Replication Conflicts, Concurrency Control, Conflict Resolution, Data Consistency, Data Integrity, Write-Write Conflicts, MVCC (Multi-Version Concurrency Control), Locking, Conflict Detection, Merging Algorithms, High Availability, Distributed Systems.

Kirish: Tugun (Deadlock) — bu ikki yoki undan ortiq transaksiyalar (operatsiyalar) bir-biri tomonidan band qilingan resurslarni kutib, bir-birini bloklab

qo‘ygan holat. Natijada ikkala operatsiya ham bajarilmay qoladi, chunki har biri ikkinchisini kutishga majbur bo‘ladi.

Tugun (Deadlock) Sodir Bo‘lishining Oddiy Misoli

1. Transaksiya A:
   - Jadval1 dan ma’lumotni o‘qiydi va uni o‘zgartirish uchun qulflaydi (lock).
   - Keyin Jadval2 ga ma’lumot qo‘shishga urinadi, lekin u allaqachon Transaksiya B tomonidan qulflangan.
2. Transaksiya B:
   - Jadval2 dan ma’lumotni o‘qiydi va uni o‘zgartirish uchun qulflaydi (lock).
   - Keyin Jadval1 ga ma’lumot qo‘shishga urinadi, lekin u allaqachon Transaksiya A tomonidan qulflangan.

Natija:

- Transaksiya A Jadval2 ni kutadi.
- Transaksiya B Jadval1 ni kutadi.
- Ikkalasi ham bloklangan, hech biri oldinga siljiy olmaydi!

Tugun (Deadlock) Qachon Kelib Chiqadi?

Agar 2+ transaksiya bir-birining tugagan resurslarini kutsa.
Agar ma’lumotlarga murojaat tartibi bir xil bo‘lmasa (masalan, bir transaksiya A→B tartibida, ikkinchisi B→A tartibida ishlayotgan bo‘lsa).
Agar transaksiyalar uzoq vaqt davom etsa va qo‘shimcha resurslarni bloklab qo‘ysa.

Tugun (Deadlock) ni Qanday Oldini Olish Mumkin?

Transaksiyalarni qisqa tutish (tezroq yakunlash).
Ma’lumotlarga har doim bir xil tartibda murojaat qilish (masalan, A→B→C tartibiga rioya qilish).
Indekslardan to‘g‘ri foydalanish (qidiruv tezligini oshirish).
Deadlock detektorlardan foydalanish (DBMS avtomatik aniqlab, birini bekor qiladi).
Lock timeout sozlamalari (agar resurs band bo‘lsa, kutish vaqtini cheklash).

Tugun (Deadlock) Qanday Hal Qilinadi?

1. DBMS (MySQL, SQL Server, Oracle) avtomatik aniqlaydi va bir transaksiyani "qurbon" (victim) qilib, uni bekor qiladi.
2. Dasturiy tomondan qayta ishlash (retry mechanism) — agar transaksiya deadlock tufayli bekor qilinsa, uni qayta urinish.

3. KILL buyrug‘i (agar kerak bo‘lsa, ma’lum bir seansni majburan to‘xtatish).

```
-- SQL Serverda deadlockni hal qilish
KILL 55;  -- 55 - bloklangan seans IDsi
```

Tugun (Deadlock) — bu ikki yoki undan ortiq jarayonlar bir-birini kutib, tizimni bloklab qo‘ygan holat.
Buning oldini olish uchun transaksiyalarni optimallashtirish, bir xil tartibda resurslardan foydalanish va DBMS sozlamalarini to‘g‘ri o‘rnatish kerak.

Agar tizimingizda deadlocklar tez-tez uchrasa, monitoring qurish va qayta ishlash mexanizmlarini joriy etish yaxshi yechim bo‘ladi.

Tugun (deadlock) vaziyatlari ma'lumotlar bazasida avtomatik yoki qo'lda hal qilinadi. Quyida tugunni qanday hal qilish va qaysi tizimlar bunga javob beradi haqida tushuntiraman:

1. Ma'lumotlar Bazasi Tizimining Avtomatik Hal Qilishi

Ko'pgina zamonaviy DBMS (MySQL, SQL Server, Oracle, PostgreSQL) deadlocklarni o'zi aniqlab, bir transaksiyani "qurbon" (victim) qilib bekor qiladi.

Qanday Ishlaydi?

- DBMS deadlock detector yordamida aylanma bog‘liqlikni (circular wait) aniqlaydi.
- Bir transaksiyani tanlab (qoida bo‘yicha) uni bekor qiladi (masalan, kamroq ish qilganini yoki kechroq boshlanganini).
- Qolgan transaksiya bajarilishini davom ettiradi.

Misol (SQL Server):

```
-- Avtomatik xabar: "Transaction (Process ID 62) was deadlocked on lock resources..."
-- DBMS o'zi hal qilib, ERROR qaytaradi
```

2. Dasturchi Tomonidan Hal Qilish Usullari

Agar tizim o‘zi hal qilolmasa, dasturiy tomondan yechimlar qo‘llaniladi:

a) Transaksiyalarni Qayta Urinish (Retry Logic):

```
import time
from django.db import transaction, OperationalError

def run_transaction():
    max_retries = 3
```

```
    for attempt in range(max_retries):
        try:
            with transaction.atomic():
                # Transaksiya kodlari
                return True  # Muvaffaqiyatli yakunlandi
        except OperationalError as e:
            if "deadlock" in str(e).lower() and attempt < max_retries - 1:
                time.sleep(0.1)  # Qisqa kutish
                continue
            raise  # Qayta ishlab bo‘lmasa, xatolikni ko‘tarish
```

b) Seansni Majburan To‘xtatish (KILL):

```
-- Bloklangan seansni topish
SELECT blocking_session_id FROM sys.dm_exec_requests WHERE session_id = @@SPID;

-- Seansni o‘chirish (SQL Server)
KILL 55;  /* 55 - blocking process ID */
```

c) Lock Timeout Sozlash:

```
-- Transaksiya N sekunddan keyin timeoutga uchrasa, bekor qilinadi
SET LOCK_TIMEOUT 5000;  /* 5 soniya */
```

3. Tugunni Oldini Olish – Eng Yaxshi Amaliyotlar

Tugunlarni hal qilishdan ko‘ra, ularni oldini olish afzal:

a) Transaksiyalarni Optimallashtirish

- Qisqa vaqtli transaksiyalardan foydalaning.
- Faqat kerakli ma'lumotlarni lock qiling.

b) Bir Xil Jadval Tartibi

Har doim ma'lumotlarga bir xil ketma-ketlikda murojaat qiling:
To‘g‘ri: A → B → C
Noto‘g‘ri: A → B va B → A (deadlock riski!)

c) Indexlardan Foydalanish

Tezroq so‘rovlar = qisqaroq lock vaqtlari.

d) Isolation Darajasini To‘g‘ri Tanlash

```
-- SQL Serverda READ COMMITTED (deadlock kamroq)
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

Usul	Kim Bajaradi?	Izoh
DBMS avtomatik hal qilish	MySQL/SQL Server/Oracle	Transaksiyalardan birini bekor qiladi
Retry mexanizmi	Dasturchi	Xatolikni qayta ishlash
KILL buyrug‘i	Admin/DBA	Bloklovchi seansni majburan to‘xtatish
Lock timeout	DBMS/Dastur	Kutish vaqtini cheklash

Eng yaxshi yechim — deadlocklarni oldini olish!

Agar tugun yuzaga kelsa, DBMS ko‘pincha o‘zi hal qiladi, lekin dasturingizda retry mexanizmi bo‘lishi mustahkamroq.

Agar sizning ma'lumotlar bazangizda tez-tez deadlocklar bo‘layotgan bo‘lsa, jadval dizaynini, so‘rovlarni va transaksiya tartibini tekshiring.

Tugun vaziyati - bu ikki yoki undan ortiq transaksiyalar bir-birini kutib turgan holda, hech biri oldinga siljimay qolgan holat. Bu ma'lumotlar bazasida jiddiy muammolarga olib kelishi mumkin.

Tugun Vaziyatlarni Aniqlash

1. SQL Serverda Tugunlarni Aniqlash

```
-- Faol tugun vaziyatlarni ko'rish
SELECT
    tl.request_session_id,
    wt.blocking_session_id,
    OBJECT_NAME(p.OBJECT_ID) AS ObjectName,
    tl.resource_type,
    tl.resource_description,
    tl.request_mode,
    tl.request_status
FROM sys.dm_tran_locks tl
JOIN sys.dm_os_waiting_tasks wt ON tl.lock_owner_address = wt.resource_address
JOIN sys.partitions p ON p.hobt_id = tl.resource_associated_entity_id
WHERE tl.resource_database_id = DB_ID()
```

2. MySQLda Tugunlarni Aniqlash

```
-- Joriy transaksiyalar va ularni bloklovchilarni ko'rish
SHOW ENGINE INNODB STATUS;
```

```
-- Yoki
SELECT * FROM information_schema.INNODB_TRX;
SELECT * FROM information_schema.INNODB_LOCKS;
SELECT * FROM information_schema.INNODB_LOCK_WAITS;
```

3. Oracleda Tugunlarni Aniqlash

```
   l.session_id,
   o.owner,
   o.object_name,
   l.oracle_username,
   l.os_user_name
FROM
   v$locked_object l,
   dba_objects o
WHERE
   o.object_id = l.object_id;
```

Tugun Vaziyatlarni Oldini Olish Usullari

1. Transaksiyalarni Qisqa Tutish

- Transaksiyalarni iloji boricha qisqa vaqt ichida yakunlang
- Faqat zarur bo'lgan operatsiyalarni transaksiyaga oling

2. Ma'lumotlarga Bir Xil Tartibda Murojaat Qilish

- Jadval va satrlarga doimo bir xil tartibda murojaat qiling
- Masalan: A jadvali, keyin B jadvali deb ketma-ketlikka rioya qiling

3. Optimallashtirilgan So'rovlar

- Indekslardan to'g'ri foydalaning
- Uzun vaqt oladigan so'rovlarni optimallashtiring

4. Isolation Darajalarini To'g'ri Tanlash

- READ COMMITTED kabi pastroq isolation darajalaridan foydalaning
- Serializable faqat zarur bo'lganda ishlating

Tugun Vaziyatlarni Hal Qilish

1. SQL Serverda Tugunni Hal Qilish

```
-- Bloklangan seansni o'chirish
KILL <session_id>;

-- Tugunni avtomatik hal qilish uchun sozlash
ALTER DATABASE YourDatabase SET DEADLOCK_PRIORITY LOW;
```

2. MySQLda Tugunni Hal Qilish

```
-- Faol transaksiyalarni ko'rish
SHOW PROCESSLIST;
```

```
-- Kerakli processni to'xtatish
KILL <process_id>;
```

3. Oracleda Tugunni Hal Qilish

```
-- Tugun holatdagi seanslarni ko'rish
SELECT * FROM v$session WHERE blocking_session IS NOT NULL;

-- Seansni tugatish
ALTER SYSTEM KILL SESSION '<sid,serial#>';
```

Tugunlarni Monitoring Qilish

SQL Serverda Monitoring

```
-- Tugunlarni kuzatish uchun trace yaratish
CREATE EVENT SESSION [Deadlock_Monitor] ON SERVER
ADD EVENT sqlserver.xml_deadlock_report
ADD TARGET package0.event_file(SET filename=N'Deadlock_Monitor.xel')
GO

-- Trace ni boshlash
ALTER EVENT SESSION [Deadlock_Monitor] ON SERVER STATE = START;
GO
```

MySQLda Monitoring

```
-- Tugunlarni logga yozish
SET GLOBAL innodb_print_all_deadlocks = ON;
```

Eng Yaxshi Amaliyotlar

1. **Dasturiy ta'minotda xatoliklarni qayta ishlash**: Transaksiyalar uchun qayta urinish mexanizmini amalga oshiring
2. **Connection Poolingdan foydalaning**: Bog'lanishlarni qayta ishlatish
3. **Timeoutlarni o'rnating**: Transaksiyalar uchun maksimum vaqt belgilang
4. **Monitoring tizimini yarating**: Tugunlarni vaqtida aniqlash uchun
5. **Testlash**: Yuqori yuk ostida sistemani test qilish

Tugun vaziyatlarni tushunish va ularni samarali boshqarish har qanday ma'lumotlar bazasi dasturchisi uchun muhim ko'nikmadir. Yuqoridagi usullar va so'rovlar yordamida siz o'z tizimlaringizdagi tugun muammolarini aniqlashingiz, oldini olishingiz va hal qilishingiz mumkin.

Foydalanilgan adabiyotlar:

1. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006).

2. Muchnick, S. S. (1997).
3. Cooper, K., & Torczon, L. (2011).
4. Grune, D., van Reeuwijk, K., Bal, H. E., Jacobs, C. J. H., & Langendoen, K. (2012).
5. Karimberdiyevich, O. M., & Abdulaziz o‘g’li, Y. M. (2024). SUN’IY INTELLEKTNING AFZALLIKLARI VA KAMCHILIKLARI. IZLANUVCHI, 1(1), 75-85.
6. Karimberdiyevich, O. M., & Abdulaziz o‘g‘li, Y. M. (2024). NEYRO KOMPYUTERLAR. YANGI O ‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(5), 19-27.
7. Karimberdiyevich, O. M., & Abdulaziz o‘g‘li, Y. M. (2024). K-YAQIN QO'SHNI ALGORITMI. IZLANUVCHI, 1(1), 122-124.
8. Abdulaziz o'g'li, Y. M. (2025). WPFDA ANIMATSIYA YARATISHNI QO’LLANISHI. MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS, 1(4), 172-175.
9. Abdulaziz o’g’li, Y. M. (2025). MOLIYA VA HISOB–KITOB ILOVALARIDA WPF BILAN ISHLASH. MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS, 1(4), 189-193.
10. Karimberdiyevich, O. M. (2024). NEYROEMULYATORLAR VA ULARNING QO'LLANILISHI. YANGI O ‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(5), 82-89.
11. Abdulaziz ogli, Y. M. (2025). WPF DA IKKI O’LCHOVLI VA UCH O’LCHOVLI GRAFIKALAR BILAN ISHLASHNING HAYOTGA TATBIQLARI. MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS, 1(4), 176-179.
12. Karimberdiyevich, O. M., & Abdulaziz o’g’li, Y. M. (2024). MASHINANI O ‘RGANISHDA TASNIFLASH VA REGRESSIYA. IZLANUVCHI, 1(1), 114-121.
13. Karimberdiyevich, O. M., & Abdulaziz o‘g‘li, Y. M. (2024). PIVOT JADVALI YARATISH VA TAHRIRLASH. YANGI O ‘ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(5), 28-30.
14. Karimberdiyevich, O. M., & Abdulaziz o’g’li, Y. M. (2024). MASHINANI O ‘RGANISHDA TASNIFLASH VA REGRESSIYA. IZLANUVCHI, 1(1), 114-121.
15. Karimberdiyevich, O. M. (2024). FORMAL GRAMMATIKA VA SEMANTIK TO’R. IZLANUVCHI, 1(1), 94-99.
16. Karimberdiyevich, O. M., & Abdulaziz o‘g’li, Y. M. (2024). PROGNOZLASH VA VIZUALIZATSIYA. TA’LIM, TARBIYA VA INNOVATSIYALAR JURNALI, 1(1), 124-132.

17. Karimberdiyevich, O. M. (2024). O'ZBEKISTON RESPUBLIKASIDA BIG DATA NI RIVOJLANISH TENDENSIYALARI. TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI, 1(1), 147-151.
18. Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). ICHKI MUAMMOLARNI TUSHUNISH. TA'LIM, TARBIYA VA INNOVATSIYALAR JURNALI, 1(1), 98-104.
19. Karimberdiyevich, O. M., Abdulaziz o'g'li, Y. M., & Hokimjon o'g, I. M. R. (2024). EVALUTSION DASTURLASH. GENETIK ALGORITM. YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(4), 519-522.
20. "Optimizing Compilers for Modern Architectures"
21. **"High Performance Browser Networking"**
22. "Performance Optimization Techniques in C# and .NET"
23. "Computer Systems: A Programmer's Perspective"
24. "Modern Compiler Implementation in C / Java / ML"
25. **"Software Optimization Guide"** (Intel Developer Manual)