

**BLOKLASH SHAKLLARI VA MEXANIZMLARINING BANK
TIZIMLARIDA QO‘LLANILISHI****Quvvatali Rahimov***Farg‘ona Davlat Universiteti, (PhD),**quvvatali.rahimov@gmail.com,***Tojimatov Isroil Nurmamatovich***Farg‘ona davlat universiteti katta o‘qituvchisi**israiltojimatov@gmail.com***Muhammaddiyorova Madina Bahodirjon qizi***Farg‘ona davlat universiteti 2-kurs talabasi**mmuhammaddiyorova0110@gmail.com*

Annotatsiya: Maqola bank tizimlarida bloklash mexanizmlarining qo‘llanilishiga bag‘ishlangan. Umumiy, eksklyuziv bloklash va 2PL prinsiplari tahlil qilinadi. Pul o‘tkazmalari va hisob balansini sinxronlashtirish misollari orqali bloklash ahamiyati ko‘rsatiladi. Deadlock holatlari va ularni bartaraf etish usullari yoritiladi. Microsoft SQL Server‘da bloklash qisqacha ko‘rib chiqiladi. Maqola mutaxassislar va talabalar uchun foydali manba hisoblanadi.

Kalit so‘zlar: Bloklash, bank tizimlari, berilganlar bazasi, tranzaksiya, ma‘lumotlar integralligi, deadlock, ikki fazali bloklash, Microsoft SQL Server, sinxronlashtirish, pul o‘tkazmalari.

Аннотация: Статья посвящена применению механизмов блокировки в банковских системах. Рассматриваются общие, эксклюзивные блокировки и принципы 2PL. Показана значимость блокировок на примерах переводов средств и синхронизации баланса счетов. Освещены тупиковые ситуации и методы их устранения. Кратко описана реализация блокировок в Microsoft SQL Server. Статья полезна для специалистов и студентов.

Ключевые слова: Блокировка, банковские системы, база данных, транзакция, целостность данных, тупик (deadlock), двухфазная блокировка, Microsoft SQL Server, синхронизация, денежные переводы.

Abstract: The article explores the application of locking mechanisms in banking systems. It analyzes shared, exclusive locks, and 2PL principles. The importance of locks is demonstrated through money transfers and account balance synchronization examples. Deadlock situations and their resolution methods are discussed. A brief overview of lock implementation in Microsoft SQL Server is provided. The article is a valuable resource for professionals and students.

Keywords: Locking, banking systems, database, transaction, data integrity, deadlock, two-phase locking, Microsoft SQL Server, synchronization, money transfers.

Kirish

Zamonaviy axborot tizimlarida berilganlar bazalari ma'lumotlar integralligi va tranzaksiyalarni sinxronlashtirishni ta'minlashda muhim rol o'ynaydi. Ushbu jarayonda bloklash mexanizmlari tranzaksiyalar o'rtasidagi konfliktlarni oldini olish va ma'lumotlarning izchil holatda saqlanishini kafolatlash uchun asosiy vosita sifatida xizmat qiladi. Umumiy (shared), eksklyuziv (exclusive) va boshqa bloklash turlari, shuningdek, ikki fazali bloklash (2PL) kabi mexanizmlar berilganlar bazasi tizimlarining samarali ishlashini ta'minlaydi. Ayniqsa, bank tizimlari kabi yuqori darajada ishonchli va sinxronlashtirilgan operatsiyalarni talab qiluvchi sohalarda bloklash mexanizmlari alohida ahamiyatga ega. Masalan, pul o'tkazmalari yoki hisob balansini yangilash jarayonida noto'g'ri sinxronlashtirish jiddiy moliyaviy xatolarga olib kelishi mumkin. Bank tizimlari doirasida bloklash mexanizmlari nafaqat ma'lumotlar integralligini ta'minlaydi, balki deadlock kabi muammolarni bartaraf etishda ham muhim rol o'ynaydi. Ushbu maqola blokirovka shakllarini tahlil qilish va ularning bank tizimlaridagi amaliy qo'llanilishini o'rganishga bag'ishlangan. Maqolada umumiy va eksklyuziv bloklash, 2PL prinsiplari va deadlock holatlarini hal qilish usullari ko'rib chiqiladi. Shuningdek, Microsoft SQL Server misolida bloklash mexanizmlarining zamonaviy berilganlar bazasi tizimlarida qanday amalga oshirilishi qisqacha yoritiladi. Maqola bank tizimlari, berilganlar bazasi mutaxassisleri va ushbu sohada bilim olmoqchi bo'lgan talabalar uchun foydali manba bo'lib xizmat qilishga qaratilgan.

Berilganlar bazalarida tranzaksiyalarni sinxronlashtirish va ma'lumotlar integralligini ta'minlash uchun bloklash mexanizmlari muhim ahamiyatga ega. Bloklash tranzaksiyalarni parallel ravishda bajarishda yuzaga kelishi mumkin bo'lgan konfliktlarni oldini oladi va ma'lumotlarning izchil holatda saqlanishini kafolatlaydi. Ushbu bo'limda blokirovka shakllari, ularning turlari va ikki fazali bloklash (2PL) prinsiplari ko'rib chiqiladi.

Bloklash mexanizmlari turli shakllarda qo'llaniladi, ularning har biri ma'lum bir vazifani bajaradi. Eng keng tarqalgan blokirovka turlari quyidagilardan iborat:

- ✚ Umumiy bloklash (Shared Lock): Ushbu turdagi bloklash bir nechta tranzaksiyalarga ma'lumotni faqat o'qish imkonini beradi. Masalan, bir tranzaksiya ma'lumotni o'qiyotgan bo'lsa, boshqa tranzaksiyalar ham uni o'qishi mumkin, lekin uni o'zgartirish (yozish) taqiqlanadi. Bu ma'lumotni o'qishda izchillikni ta'minlaydi.

- ✚ Eksklyuziv bloklash (Exclusive Lock): Eksklyuziv bloklash ma'lumotni faqat bitta tranzaksiya tomonidan o'qish va yozish uchun bloklaydi. Bu turdagi bloklash ma'lumotni o'zgartirishda boshqa tranzaksiyalarning aralashuvini oldini oladi. Masalan, agar tranzaksiya ma'lumotni yangilayotgan bo'lsa, boshqa tranzaksiyalar bu ma'lumotni na o'qiy oladi, na o'zgartira oladi.
- ✚ Yangilash bloklash (Update Lock): Bu bloklash ma'lumotni o'qishga ruxsat beradi, lekin uni yangilashga tayyorlanayotgan tranzaksiyalar uchun maxsus ishlatiladi. U deadlock holatlarini kamaytirishga yordam beradi. Misol uchun, agar ikkita tranzaksiya bir xil hisob balansini yangilamoqchi bo'lsa, update lock birinchi tranzaksiyaga ma'lumotni o'qish va yangilashga ruxsat beradi, ikkinchi tranzaksiyani esa kutishga majbur qiladi, bu esa eksklyuziv bloklashga o'tishdan oldin konfliktlarni kamaytiradi.

Bloklashlar turli darajalarda qo'llanilishi mumkin: qator darajasida (row-level), sahifa darajasida (page-level) yoki butun jadval darajasida (table-level). Qator darajasidagi bloklash eng keng tarqalgan bo'lib, u faqat ma'lum bir qatorni bloklaydi va boshqa qatorlar bilan ishlashga imkon beradi, bu esa tizimning samaradorligini oshiradi. Ikki fazali blokirovka (2PL) prinsiplari.

Bloklash mexanizmlarining samarali ishlashi uchun ikki fazali bloklash (Two-Phase Locking, 2PL) protokoli keng qo'llaniladi. 2PL protokoli tranzaksiyalarni ikki bosqichda boshqaradi: Kengayish fazasi (Growing Phase): Bu bosqichda tranzaksiya kerakli ma'lumotlarni bloklaydi, ya'ni bloklashlar qo'yiladi, lekin hech qanday bloklash olib tashlanmaydi. Qisqarish fazasi (Shrinking Phase): Bu bosqichda tranzaksiya barcha operatsiyalarni bajarib bo'lgach, bloklashlar olib tashlanadi, ya'ni ma'lumotlar boshqa tranzaksiyalar uchun ochiladi. 2PL protokoli tranzaksiyalarning seriyalashuvini (serializability) ta'minlaydi, ya'ni tranzaksiyalar parallel ravishda bajarilsa ham, ularning natijasi ketma-ket bajarilgandek bo'ladi. Bu ma'lumotlar integralligini saqlashda muhim ahamiyatga ega. Biroq, 2PL protokoli deadlock holatlarini to'liq bartaraf eta olmaydi. Masalan, agar ikkita tranzaksiya bir-birining bloklagan ma'lumotlarini kutsa, deadlock yuzaga keladi. Bunday holatlarni bartaraf etish uchun deadlock aniqlash (deadlock detection) va oldini olish (deadlock prevention) usullari qo'llaniladi.

Bloklash va uning muammolari.

Bloklash mexanizmlari foydali bo'lsa-da, ular ba'zi muammolarni keltirib chiqarishi mumkin. Eng keng tarqalgan muammo – deadlock holati. Deadlock ikkita yoki undan ko'p tranzaksiya bir-birining bloklagan resurslarini kutganda yuzaga keladi. Masalan, tranzaksiya A hisob X ni eksklyuziv blokirovka bilan bloklaydi va hisob Y ni yangilashni kutadi. Shu bilan birga, tranzaksiya B hisob Y ni bloklaydi va

hisob X ni kutadi. Natijada, ikkala tranzaksiya ham davom eta olmaydi, ya'ni deadlock yuzaga keladi. SQL Server'da deadlock aniqlash algoritmi (deadlock detection) ushbu holatni aniqlaydi va odatda bir tranzaksiyani to'xtatib, boshqasiga davom etish imkonini beradi. Buni oldini olish uchun tranzaksiyalar resurslarni doim bir xil tartibda bloklashi tavsiya etiladi. Bloklash mexanizmlari tranzaksiyalarni sinxronlashtirishda muhim vosita bo'lib, ayniqsa bank tizimlari kabi yuqori darajada ishonch talab qiluvchi sohalarda keng qo'llaniladi.

Bank tizimlari yuqori darajada ishonch va sinxronlashtirish talab qiluvchi sohalardan biridir. Pul o'tkazmalari, hisob balansini yangilash va boshqa moliyaviy operatsiyalar paytida ma'lumotlar integralligini ta'minlash uchun blokirovka mexanizmlari muhim ahamiyatga ega. Ushbu bo'limda blokirovka mexanizmlarining bank tizimlaridagi amaliy qo'llanilishi misollar orqali tahlil qilinadi.

Bank tizimlarida eng keng tarqalgan operatsiyalardan biri pul o'tkazmalaridir. Faraz qilaylik, foydalanuvchi A dan foydalanuvchi B ga 1000 so'm pul o'tkazmoqchi. Bu jarayonda ikkita tranzaksiya amalga oshiriladi: A ning hisobidan 1000 so'm ayiriladi va B ning hisobiga 1000 so'm qo'shiladi. Agar ushbu operatsiyalar sinxronlashtirilmasa, ma'lumotlar integralligi buzilishi mumkin. Masalan, agar A ning hisobidan pul ayirilgan bo'lsa, lekin tizim nosozligi tufayli B ning hisobiga pul qo'shilmagan bo'lsa, moliyaviy xato yuzaga keladi. Bunday holatlarni oldini olish uchun eksklyuziv bloklash qo'llaniladi. Tranzaksiya boshlanganda A va B ning hisob ma'lumotlari eksklyuziv blokirovka bilan bloklanadi, ya'ni boshqa tranzaksiyalar bu ma'lumotlarni o'qiy olmaydi yoki o'zgartira olmaydi. Quyidagi tarzda jarayon amalga oshiriladi:

1. Tranzaksiya A ning hisobini bloklaydi va balansidan 1000 so'm ayiradi.
2. Tranzaksiya B ning hisobini bloklaydi va balansiga 1000 so'm qo'shadi.
3. Operatsiya muvaffaqiyatli bajarilgach, bloklashlar olib tashlanadi va tranzaksiya tasdiqlanadi (commit).

Bu jarayon ACID (Atomicity, Consistency, Isolation, Durability) xususiyatlarini ta'minlaydi, chunki tranzaksiya to'liq bajariladi yoki umuman bajarilmaydi.

Faraz qilaylik, bank tizimidan foydalanuvchi X dan foydalanuvchi Y ga 1500 so'm pul o'tkazishi kerak. SQL Serverda bu jarayon quyidagi tarzda amalga oshirilishi mumkin:

Dastlab baza yaratildi va yangi jadval ham hosil qilinib, ma'lumotlar kiritildi va tranzaksiya boshlanadi:

```
CREATE DATABASE BANK_TIZIMI;
```

```
USE BANK_TIZIMI;  
GO
```

```

DROP TABLE IF EXISTS HISOBLAR;
GO
CREATE TABLE HISOBLAR (
    HISOB_ID VARCHAR(10) PRIMARY KEY,
    BALANS DECIMAL(18, 2)
);
GO
INSERT INTO HISOBLAR (HISOB_ID, BALANS)
VALUES ('X', 5000), ('Y', 3000);
GO
-- TRANZAKSIYANI BOSHLASH
BEGIN TRANSACTION;
UPDATE HISOBLAR WITH (ROWLOCK)
SET BALANS = BALANS - 1500
WHERE HISOB_ID = 'X';
UPDATE HISOBLAR WITH (ROWLOCK)
SET BALANS = BALANS + 1500
WHERE HISOB_ID = 'Y';
COMMIT TRANSACTION;
GO
SELECT * FROM HISOBLAR
GO

```

Natija:

	HISOB_ID	BALANS
1	X	3500.00
2	Y	4500.00

Ushbu kod faqat bitta tranzaksiya amalga oshirilganligini ko'rsatadi. Quyida esa parallel tranzaksiyalar o'rtasidagi konfliktlarni ko'rsatish uchun SQL kod keltirilgan:

```

USE BANK_TIZIMI
-- Tranzaksiya 1: X dan Y ga 1500 so'm o'tkazish
BEGIN TRANSACTION;
UPDATE HISOBLAR WITH (ROWLOCK)
SET BALANS = BALANS - 1500
WHERE HISOB_ID = 'X';
WAITFOR DELAY '00:00:05'; -- 5 soniya kechikish

```

UPDATE HISOBLAR WITH (ROWLOCK)

SET BALANS = BALANS + 1500

WHERE HISOB_ID = 'Y';

COMMIT TRANSACTION;

-- Tranzaksiya 2: Y dan X ga 1000 so'm o'tkazish (parallel ishlaydi)

BEGIN TRANSACTION;

UPDATE HISOBLAR WITH (ROWLOCK)

SET BALANS = BALANS - 1000

WHERE HISOB_ID = 'Y';

WAITFOR DELAY '00:00:05'; -- 5 soniya kechikish

UPDATE HISOBLAR WITH (ROWLOCK)

SET BALANS = BALANS + 1000

WHERE HISOB_ID = 'X';

COMMIT TRANSACTION;

Tranzaksiya 1 da tranzaksiya X hisobidan 1500 so'm ayiradi, 5 soniya kutadi (parallel tranzaksiya bilan sinxronlash uchun), keyin Y hisobiga 1500 so'm qo'shadi. ROWLOCK qatorni bloklaydi, boshqa tranzaksiyalarning aralashuvining oldini oladi. Muvaffaqiyatli tugagach, o'zgarishlar tasdiqlanadi (COMMIT).

Tranzaksiya 2 da esa tranzaksiya Y hisobidan 1000 so'm ayiradi, 5 soniya kutadi, keyin X hisobiga 1000 so'm qo'shadi. Parallel ishlaydi va ROWLOCK bilan sinxronlashtiriladi. Tasdiqlash (COMMIT) bilan o'zgarishlar saqlanadi.

Ikkala tranzaksiya bir vaqtning o'zida ishlaydi. Tranzaksiya 1 X ni bloklaydi va Y ni kutadi, Tranzaksiya 2 Y ni bloklaydi va X ni kutadi. 5 soniyalik kechikish (WAITFOR DELAY) tufayli deadlock holati yuzaga kelishi mumkin, chunki har bir tranzaksiya boshqasining bloklagan resursini kutadi. SQL Server deadlockni aniqlab, bir tranzaksiyani to'xtatadi (masalan, "Transaction was deadlocked" xatosi chiqadi) va boshqasi davom etadi.

SQL Serverda blokirovkalar samaradorligini oshirish uchun "lock escalation" mexanizmi qo'llaniladi. Agar tranzaksiya ko'p qatorlarni bloklasa, SQL Server resurslarni tejash uchun qator darajasidagi bloklashdan sahifa yoki jadval darajasiga o'tadi. Masalan, minglab qatorlarni yangilashda jadval darajasidagi bloklash avtomatik qo'llanilishi mumkin. Shuningdek, tranzaksiya izolyatsiya darajalari (isolation levels) muhim rol o'ynaydi. Masalan, READ COMMITTED bank operatsiyalarida keng tarqalgan bo'lsa, yuqori izchillik talab qilinadigan holatlarda SERIALIZABLE ishlatiladi. Misol uchun, quyida agar "HISOBLAR" jadvalida ko'p qatorlar bo'lsa (masalan, 5000 dan ortiq), SQL Server qator darajasidagi bloklashdan jadval

darajasidagi bloklashga o'tadi (lock escalation). Buni tekshirish uchun SQL Server Management Studio'da "Activity Monitor" dan locklarni kuzatish mumkin.

```
USE BANK_TIZIMI;  
GO
```

```
-- Katta miqdorda qatorlarni yangilash
```

```
BEGIN TRANSACTION;
```

```
UPDATE HISOBLAR WITH (ROWLOCK)
```

```
SET BALANS = BALANS + 100
```

```
WHERE BALANS > 0; -- Barcha hisoblarni yangilash
```

```
COMMIT TRANSACTION;
```

```
-- Natijani ko'rish
```

```
SELECT * FROM HISOBLAR;
```

Xulosa

Maqolada bloklash mexanizmlarining bank tizimlarida qo'llanilishi va ularning tranzaksiyalarni sinxronlashtirishdagi ahamiyati o'rganildi. Umumiy, eksklyuziv va yangilash bloklashlar hamda ikki fazali bloklash (2PL) prinsiplari tahlil qilindi. Bank tizimlarida pul o'tkazmalari va hisob balansini sinxronlashtirish misollari orqali bloklashlar ahamiyati ko'rsatiladi. Deadlock holatlari va ularni bartaraf etish usullari bank tizimlarida xavfsiz operatsiyalarni ta'minlashda muhimligi ta'kidlandi. Microsoft SQL Server misolida qator darajasidagi bloklash va tranzaksiya boshqaruvi bank operatsiyalari uchun samarali yechim sifatida ko'rib chiqildi. Ushbu maqola bank tizimlari va berilganlar bazasi sohasidagi mutaxassislar uchun foydali manba bo'lib xizmat qiladi.

Adabiyotlar va resurslar ro'yxati:

1. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель(2015). *SQL Полное руководство Третье издание*.
2. Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems (7th ed.)*. Pearson Education.
3. S. K. Ganiyev, M. M. Karimov, K. A. Tashev. *Axborot xavfsizligi*.
4. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts (7th ed.)*. McGraw-Hill Education.
5. Дмитрий Короткевич. *SQL Server. Наладка и оптимизация для профессионалов*.
6. Connolly T., Begg C. *Database Systems: A Practical Approach to Design, Implementation, and Management*. – 6th ed. – Harlow: Pearson, 2014. – 1440 p.

7. Tojimamatov, I., & Mirsiddiqova, M. (2025). BERILGANLAR BAZASIDA HAYOTIY SIKL. Модели и методы в современной науке, 4(6), 66-70.
8. Tojimamatov, I., & Siddiqova, G. (2025). TRANZAKSIYALARNI TAQSIMLANGAN TARZDA QAYTA ISHLASH MODELLARI. Современные подходы и новые исследования в современной науке, 4(6), 30-35.
9. Нурмаматович, Т. И., & Рахила, А. (2025). НА ОСНОВЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ПОВЫШЕНИЕ УСТОЙЧИВОСТИ К ПОЛОМКАМ И АВАРИЯМ. YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 2(8), 197-204.
10. Tojimamatov, I., & Ahmataliyeva, S. (2025). BERILGANLARNI MARKAZLASHGAN TARZDA BOSHQARISH TAMOYILLARI. Академические исследования в современной науке, 4(21), 59-64.
11. Tojimamatov, I., & Adxamova, C. (2025). AMALIY TIZIMLARDA BERILGANLAR BAZASINI BOSHQARISH TIZIMLARI O 'RNI. Академические исследования в современной науке, 4(21), 77-82.
12. Tojimamatov, I., & Fazliddinov, X. (2025). BERILGANLAR BAZASI ADMINISTRATORI VA UNING XUSUSYATLAR. Академические исследования в современной науке, 4(21), 90-95.
13. Karimberdiyevich, O. M., Nurmamatovich, T. I., & Abdulaziz o'g'li, Y. M. (2024). BIG DATA SOHASIDAGI XALQARO LOYIHALAR. IZLANUVCHI, 1(1), 39-45.
14. Karimberdiyevich, O. M., Abdulaziz o'g'li, Y. M., & Zarifjon o'g'li, X. N. (2024). DATA MINING METODLARI VA BOSQICHLARI. YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(4), 303-311.
15. Nurmamatovich, T. I. (2024). BERILGANLARNING TARMOQ MODELLARI: ODDIY VA MURAKKAB TARMOQ TUZILISHLARI.
16. Tojimamatov, I., Soliyeva, X., & Israilova, R. (2025). FAYL NOMLARINI QISQARTIRISH ALGORITMLARI. Академические исследования в современной науке, 4(26), 45-52.
17. Nurmamatovich, T. I. (2025). MONGODB DA BIG DATA BILAN ISHLASH USULLARI. YANGI O 'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 2(8), 792-798.
18. Nurmamatovich, T. I. (2025). MOBIL OPERATSION SISTEMALARNING KELAJAGI. Лучшие интеллектуальные исследования, 44(5), 133-139.
19. Nurmamatovich, T. I., & Umidjon o'g, M. Z. S. (2025). BERILGANLAR BAZASIDA HAYOTIY SIKL. Лучшие интеллектуальные исследования, 44(5), 169-178.

20. Nurmatovich, T. I., & Umidjon o'g, M. Z. S. (2025). MASHINA KODLARI BILAN ISHLASH. Лучшие интеллектуальные исследования, 44(5), 159-168.
21. Nurmatovich, T. I. (2025). BERILGANLAR BAZASI ADMINISTRATORI. Лучшие интеллектуальные исследования, 44(5), 276-282.
22. Tojimamatov, I. (2025). ADO-NET TEXNOLOGIYASI YORDAMIDA HISOBOTLAR VA FORMALARNI SHAKLLANTIRISH. Академические исследования в современной науке, 4(25), 122-126.
23. Nurmatovich, T. I. (2025). STATISTIKA SOHASIDA AXBOROT TIZIMLARI VA TEXNOLOGIYALARINI SINTAKSIS TAXLIL QILISH. Лучшие интеллектуальные исследования, 44(4), 157-166.
24. Nurmatovich, T. I. (2025). AXBOROTLARNI TAQDIM ETISH VA ULAR BILAN ISHLASH. Лучшие интеллектуальные исследования, 44(4), 135-140.
25. Tojimamatov, I., & Abduvaliyev, X. (2025). KO 'P FOYDALANUVCHILI BBOT ARHITEKTURASI. Инновационные исследования в науке, 4(5), 16-22.
26. Tojimamatov, I., & Xolmurod o'g, A. O. H. (2025, May). SQL SERVERDA CHEKLASHLAR. In CONFERENCE OF MODERN SCIENCE & PEDAGOGY (Vol. 1, No. 1, pp. 409-413).
27. Tojimamatov, I., & Abdulhafizov, I. (2025). OBYEKTAR VA ATRIBUTLAR. BRIDGING THE GAP: EDUCATION AND SCIENCE FOR A SUSTAINABLE FUTURE, 1(1), 107-112.
28. Tojimamatov, I. N., & Iminova, G. I. (2025). SEMANTIK OBEKT MODELI VA KATTA MA'LUMOTLAR (BIG DATA). ОБРАЗОВАНИЕ И НАУКА В XXI BEKE, (58-3).
29. <https://www.wikipedia.org>
30. www.geeksforgeeks.org