

Бази от данни

Транзакции

доц. д-р Димитър Димитров

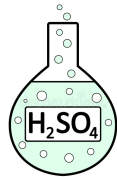
- Настоящата лекция е допълнителен материал, т.е. не е необходимо да се изучава за изпитите

Мотивация

- Поредица от една или повече операции върху БД, които се третират като едно цяло
 - Не е задължително да са няколко отделни заявки
- Пример: покупка на имот
 - Две UPDATE операции
 - Слив между двете

Свойства на транзакциите

- Гарантират надеждността на транзакциите
- Съкращение ACID
- **Atomicity**
 - Всичко или нищо
- **Consistency**
 - Транзакция води БД от едно консистентно състояние в друго
 - Ограниченията са спазени преди и след изпълнението ѝ
- **Isolation**
 - Конкурентно изпълнение на транзакции води до същите резултати, както и последователното
- **Durability**
 - След като е утвърдена (committed), транзакцията остава дори и след срыв

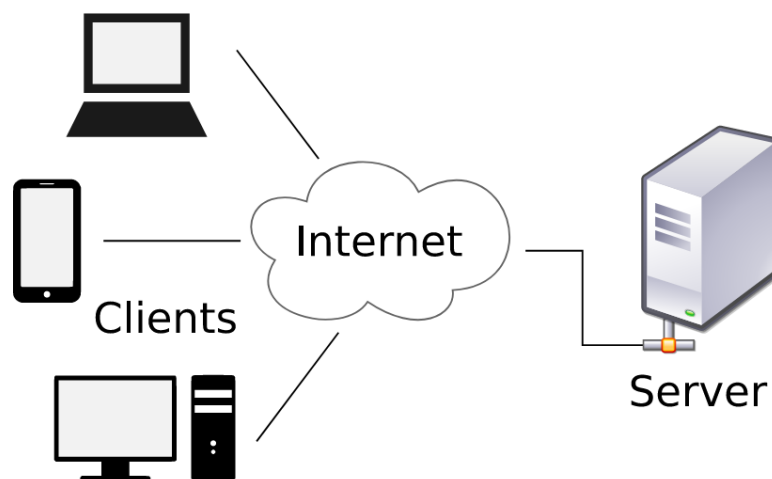


SQL

- **BEGIN TRANSACTION**
- **COMMIT**
- **ROLLBACK**
- **SET TRANSACTION ISOLATION LEVEL**
 - MS SQL, важи за текущата връзка с БД
- И други
- Имплицитно създаване на транзакции в някои случаи

Конкурентен достъп

- Потенциално много заявки се изпълняват едновременно
- Можем ли да подредим заявките в опашка и да ги изпълняваме последователно?



Проблеми при конкурентен достъп

- Транзакция 1 чете
- Транзакция 2 прави промени
- Транзакция 1 отново чете същите данни
- Възможни проблеми (аномалии) при второто четене

Dirty reads

- Прочитане на неутвърдени данни (които впоследствие могат да бъдат отхвърлени или отново променени)
- Пример (от Wikipedia):

Transaction 1

```
BEGIN;  
SELECT age FROM users WHERE id = 1;  
-- retrieves 20
```

```
SELECT age FROM users WHERE id = 1;  
-- retrieves 21 (dirty read)  
COMMIT;
```

Transaction 2

```
BEGIN;  
UPDATE users SET age = 21 WHERE id = 1;  
-- no commit here
```

```
ROLLBACK;
```


Non-repeatable reads

- При второто прочитане данните са други, понеже транзакция 2 е утвърдена

Transaction 1

```
BEGIN;  
SELECT age FROM users WHERE id = 1;  
-- retrieves 20
```

```
SELECT age FROM users WHERE id = 1;  
-- retrieves 21 (non-repeatable read)  
COMMIT;
```

Transaction 2

```
BEGIN;  
UPDATE users SET age = 21 WHERE id = 1;  
COMMIT;
```

Phantom reads

- Данните, получени при второто прочитане, съдържат нови или изтрити редове спрямо първото четене

Transaction 1

```
BEGIN;  
SELECT name FROM users WHERE age > 17;  
-- retrieves Alice and Bob
```

```
SELECT name FROM users WHERE age > 17;  
-- retrieves Alice, Bob and Carol (phantom read)  
COMMIT;
```

Transaction 2

```
BEGIN;  
INSERT INTO users VALUES (3, 'Carol', 26);  
COMMIT;
```

Нива на изолация

- READ UNCOMMITTED
 - Най-ниско ниво
 - Транзакция може да вижда данни от други транзакции, преди да бъдат утвърдени
 - Съвременни СУБД рядко го реализират
- READ COMMITTED
 - Може да вижда данни само от утвърдени транзакции
- REPEATABLE READ
 - Гарантира, че транзакция ще вижда едни и същи данни през целия си живот, дори и да са променени от друга утвърдена транзакция
- SERIALIZABLE
 - Най-високо ниво
 - Същият резултат, както при последователно изпълнение на транзакциите (а не паралелно)

Нива на изолация – сравнение

Database anomalies

Transaction Isolation Levels

	Dirty Read	Non Repeatable Read	Phantom Read
Read Uncommitted	Possible	Possible	Possible
Read Committed <small>is used by most of the applications</small>	Solved	Possible	Possible
Repeatable Read	Solved	Solved	Possible
Serializable <small>Very poor performance</small>	Solved	Solved	Solved

Как да изберем ниво на изолация?

- Избираме най-ниското ниво, което покрива изискванията ни, например:
- READ UNCOMMITTED – само ако сме сигурни, че данните няма да се променят
- Наличност на продукт в магазин - READ COMMITTED обикновено е достатъчно
- REPEATABLE READ – подходящ при серия от операции само за четене
- SERIALIZABLE – когато точността е абсолютно задължителна, напр. при банкови операции

Ако не укажем нищо?

- Ниво на изолация по подразбиране
 - READ COMMITTED за SQL Server, PostgreSQL, Oracle
 - REPEATABLE READ за MySQL
 - SERIALIZABLE за CockroachDB



Транзакции при микрослуги

- Какво е Microservices architecture?
- Всяка микрослуга може да има отделна БД
- Осигуряването на транзакции звучи като да е цяла сага, затова
- Шаблон “Saga”

Въпроси?