



Spring Transactions



Presented by Hosein Zare
Twitter: @zare88r





تراکنش ها

- مدیریت تراکنش ها بخش مهمی از سیستم مدیریت یکپارچه پایگاه های داده DBMS بوده که درستی و ثبات ثبت و انتقال داده را کنترل می کند
- تراکنش ها همانطور که قبلاً اشاره شد خاصیت ACID دارند

Atomicity –

Consistency –

Isolation –

Durability –



تراکنش ها

- تکمیل یک تراکنش به وسیله SQL در پایگاه داده به صورت زیر است :

- آغاز اجرای تراکنش به وسیله دستور - `begin transaction`
- انجام عملیات مختلف حذف ، ویرایش ، وارد نمودن اطلاعات
- اگر کلیه عملیات های مورد نظر با موفقیت انجام شد بایستی دستور `Commit` ثبت نهایی انجام شده یا کل عملیات `rollback` شود .



تراکنش ها

- برنامه نویسان مجبورند همیشه تراکنش های خود را دستی مدیریت کنند
- با روی کار آمدن JTA مدیریت آنها به Container سپرده شد با این حال نیاز به کدنویسی برای دسترسی به آن وجود دارد
- چهارچوب کاری Spring یک لایه مجزا را بر روی API های مرتبط با مدیریت تراکنش ها ، جهت کنترل بهتر برنامه فراهم می کند



Isolation

- Isolation، در سیستم‌های پایگاه داده، مشخص می‌کند که چگونه تراکنشها برای دیگر سیستم‌ها قابل مشاهده باشد. ایزولیشن در سیستم‌هایی که تراکنش‌های بالا و همزمان دارند بسیار مهم است و ایجاد ایزولیشن بالا باعث کم شدن مشکلات همزمانی می‌شود.



Isolation

- دو اصطلاح :

– Dirty Read : فرض کنید تراکنشی باعث تغییر اطلاعات در بدنه خود شده است ولی هنوز آن تراکنش Commit نشده است. اگر با استفاده از دستور Select داده های تغییر یافته معلق را استخراج نماییم اصطلاحاً به آن Dirty Read گفته می شود.

– Phantom Read : اگر حین انجام Select داخل یک تراکنش، اطلاعات جدیدی در دیتابیس درج شود و این درج اطلاعات روی نتیجه Select ما تاثیر بگذارد اصطلاحاً گفته می شود که Phantom Read اتفاق افتاده است



Isolation Levels

• Read UnCommitted

– این سطح ارزیابی به نوعی پایین ترین سطح ارزیابی می باشد به طوری که هیچ قفلی بر روی رکوردها اعمال نمی شود. مثلا اگر اطلاعاتی را داخل یک تراکنش به روزرسانی می نمایید یک تراکنش دیگر می تواند این اطلاعات را تغییر دهد. یا اگر در حال اجرای تراکنش، یک دستور Select بنویسید اطلاعاتی را دریافت کنید که هنوز تایید و نهایی نشده است.



Isolation Levels

• Read Committed

– در این حالت اطلاعاتی که داخل یک تراکنش باشد تا لحظه اتمام آن تراکنش به حالت قفل در خواهد آمد. یعنی نه می توان آن ها را Select کرد و نه می توان آن ها را تغییر داد یا حذف کرد

• Repeatable Read

– این سطح ارزیابی شبیه Read Committed می باشد ولی با این تفاوت که Repeatable Read این تضمین را به ما می دهد که تمامی دستورات update، که شما آن ها را Select کرده اید منتظر خواهد ماند تا Select شما به اتمام برسد.



Isolation Levels

• Serializable

– این سطح ارزیابی شبیه Repeatable Read می باشد منتها Serializable یک تضمین اضافی را نیز به ما می دهد که اجازه درج رکورد جدید در اطلاعات مربوطه را تا اتمام تراکنش نخواهد داد. بنابر این جلوی Phantom Read نیز گرفته خواهد شد.



Programmatic Transaction

```
@Service
public class PersonService {

    private JdbcTemplate jdbcTemplate;
    private TransactionTemplate transactionTemplate;

    @Autowired
    public PersonService(JdbcTemplate jdbcTemplate
        , PlatformTransactionManager transactionTemplate) {
        this.jdbcTemplate = jdbcTemplate;
        this.transactionTemplate = new TransactionTemplate(transactionTemplate);
        this.transactionTemplate.setTimeout(30);
        this.transactionTemplate.setIsolationLevel(
            TransactionDefinition.ISOLATION_READ_COMMITTED);
    }

    public void updatePersonName(long id, String name) {
        transactionTemplate.execute(status ->
            jdbcTemplate.update("update Persons set first_name = ? " +
                "where person_id = ?", name, id));
    }
}
```



Declarative Transaction Management

```
@Service
public class PersonService2 {

    private JdbcTemplate jdbcTemplate;

    @Transactional(isolation = Isolation.READ_COMMITTED
        , propagation = Propagation.REQUIRED,
        rollbackFor = Throwable.class, timeout = 40)
    public void updatePersonName(long id, String name) {
        jdbcTemplate.update("update Persons set first_name = ? " +
            "where person_id = ?", name, id);
    }
}
```



Propagation

- PROPAGATION_MANDATORY
- PROPAGATION_NESTED
- PROPAGATION_NEVER
- PROPAGATION_SUPPORTS
- PROPAGATION_NOT_SUPPORTED
- PROPAGATION_REQUIRED
- PROPAGATION_REQUIRES_NEW

Declarative Transaction Management

