

一、MySQL事务的概念

- 事务是一种机制、一个操作序列，包含了一组数据库操作命令，并且把所有的命令作为一个整体一起向系统提交或撤销操作请求，即这一组数据库命令要么都执行，要么都不执行。
- 事务是一个不可分割的工作逻辑单元，在数据库系统上执行并发操作时，事务是最小的控制单元。
- 事务适用于多用户同时操作的数据库系统的场景，如银行、保险公司及证券交易系统等等。
- 事务通过事务的整体性以保证数据的一致性。

说白了，所谓事务，它是一个操作序列，这些操作要么都执行，要么都不执行，它是一个不可分割的工作单位。

二、事务的ACID特点

ACID，是指在可靠数据库管理系统（DBMS）中，事务(transaction)应该具有四个特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。这是可靠数据库所应具备的几个特性。

1、原子性

指事务是一个不可再分割的工作单位，事务中的操作要么都发生，要么都不发生。

事务是一个完整的操作，事务的各元素是不可分的。

事务中的所有元素必须作为一个整体提交或回滚。

如果事务中的任何元素失败，则整个事务将失败。

案例：

A给B转帐100元钱的时候只执行了扣款语句，就提交了，此时如果突然断电，A账号已经发生了扣款，B账号却没收到加款，在生活中就会引起纠纷。这种情况就需要事务的原子性来保证事务要么都执行，要么就都不执行。

2、一致性

指在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏

当事务完成时，数据必须处于一致状态。

在事务开始前，数据库中存储的数据处于一致状态。

在正在进行的事务中，数据可能处于不一致的状态。

当事务成功完成时，数据必须再次回到已知的一致状态。

案例：

对银行转帐事务，不管事务成功还是失败，应该保证事务结束后表中A和B的存款总额跟事务执行前一致。

3、隔离性

指在并发环境中，当不同的事务同时操纵相同的数据时，每个事务都有各自的完整数据空间。

对数据进行修改的所有并发事务是彼此隔离的，表明事务必须是独立的，它不应以任何方式依赖于或影响其他事务。

修改数据的事务可在另一个使用相同数据的事务开始之前访问

这些数据，或者在另一个使用相同数据的事务结束之后访问这些数据

4、持久性

在事务完成以后，该事务对数据库所作的更改便持久的保存在数据库之中，并不会被回滚。

指不管系统是否发生故障，事务处理的结果都是永久的。

一旦事务被提交，事务的效果会被永久地保留在数据库中。

三、事务之间的相互影响

1、脏读：一个事务读取了另一个事务未提交的数据，而这个数据是有可能回滚的。

2、不可重复读：一个事务内两个相同的查询却返回了不同数据。这是由于查询时系统中其他事务修改的提交而引起的。

3、幻读：一个事务对一个表中的数据进行了修改，这种修改涉及到表中的全部数据行。同时，另一个事务也修改这个表中的数据，这种修改是向表中插入一行新数据。那么，操作前一个事务的用户会发现表中还有没有修改的数据行，就好象发生了幻觉一样。

4、丢失更新：两个事务同时读取同一条记录，A先修改记录，B也修改记录（B不知道A修改过），B提交数据后B的修改结果覆盖了A的修改结果。

四、MySQL及事务隔离级别

（1）read uncommitted：读取尚未提交的数据：不解决脏读

（2）read committed：读取已经提交的数据：可以解决脏读

（3）repeatable read：重读读取：可以解决脏读和不可重复读—mysql默认的

（4）serializable：串行化：可以解决脏读不可重复读和虚读—相当于锁表

mysql默认的事务处理级别是 repeatable read，而Oracle和SQL Server是 read committed。

1、查询全局事务隔离级别

```
show global variables like '%isolation%';
```

```
SELECT @@global.tx_isolation;
```

1

2

2、查询会话事务隔离级别

```
show session variables like '%isolation%';
```

```
SELECT @@session.tx_isolation;
```

```
SELECT @@tx_isolation;
```

1

2

3

3、设置全局事务隔离级别

```
set global transaction isolation level read committed;
```

1

4、设置会话事务隔离级别

```
set session transaction isolation level read committed;
```

1

五、事务控制语句

BEGIN 或 START TRANSACTION：显式地开启一个事务。

COMMIT 或 COMMIT WORK：提交事务，并使已对数据库进行的所有修改变为永久性的。

ROLLBACK 或 ROLLBACK WORK：回滚会结束用户的事务，并撤销正在进行的所有未提交的修改。

SAVEPOINT S1：使用 SAVEPOINT 允许在事务中创建一个回滚点，一个事务中可以有多个 SAVEPOINT；“S1”代表回滚点名称。

ROLLBACK TO [SAVEPOINT] S1：把事务回滚到标记点。

案例：

```
create database SCHOOL;
use SCHOOL;
create table CLASS(
id int(10) primary key not null,
name varchar(40),
money double
);
insert into CLASS values(1,'A',1000);
insert into CLASS values(2,'B',1000);
select * from CLASS;
```

1、测试提交事务

```
begin;
update CLASS set money= money - 100 where name='A';
select * from CLASS;
commit;
quit
mysql -u root -p
use SCHOOL;
select * from CLASS;
```

2、测试回滚事务

```
begin;
update CLASS set money= money + 100 where name='A';
select * from CLASS;
rollback;
quit
mysql -u root -p
use SCHOOL;
select * from CLASS;
```

3、测试多点回滚

```
begin;
update CLASS set money= money + 100 where name='A';
select * from CLASS;
SAVEPOINT S1;
update CLASS set money= money + 100 where name='B';
select * from CLASS;
SAVEPOINT S2;
insert into CLASS values(3,'C',1000);
select * from CLASS;
ROLLBACK TO S1;
select * from CLASS;
```

4、使用 set 设置控制事务

SET AUTOCOMMIT=0; #禁止自动提交

SET AUTOCOMMIT=1; #开启自动提交，Mysql默认为1

SHOW VARIABLES LIKE 'AUTOCOMMIT'; #查看Mysql中的AUTOCOMMIT值

如果没有开启自动提交，当前会话连接的mysql的所有操作都会当成一个事务直到你输入rollback|commit;当前事务才算结束。当前事务结束前新的mysql连接时无法读取到任何当前会话的操作结果。

如果开起了自动提交，mysql会把每个sql语句当成一个事务，然后自动的commit。

当然无论开启与否，begin; commit|rollback; 都是独立的事务。

```
use SCHOOL;
select * from CLASS;
SET AUTOCOMMIT=0;
SHOW VARIABLES LIKE 'AUTOCOMMIT';
update CLASS set money= money + 100 where name='B';
select * from CLASS;
quit
mysql -u root -p
use SCHOOL;
select * from CLASS;
```
