

数据库事务

• 简介

- 什么是事务：**事务(Transaction)**是访问并可能更新数据库中各种数据项的一个程序执行单元(**unit**)。，事务由事务开始(begin transaction)和事务结束(end transaction)之间执行的全体操作组成。
- 事务是一个不可分割的数据库操作序列，也是数据库并发控制的基本单位，其执行的结果必须使数据库从一种一致性状态变到另一种一致性状态。
- 事务结束有两种，事务中的步骤全部成功执行时，提交事务，如果其中一个失败，那么将发生回滚操作，并且撤销之前的所有操作。
- 在MySQL中只有InnoDB或BDB类型的数据表支持事务，可以通过**show engines**查看
- 在nosql数据库中，事务要求很低，
- 使用SQL事务的原因：保证数据安全有效
- 事务的特性：事务是恢复和并发控制的基本单位。

事务操作：

- 提交：commit
- 回滚：rollback

四个特征（ACID）

- 原子性（Atomicity）：
 - 操作要么全部成功，否则回滚，什么也不做（“要么不做，要么全做！”）
- 一致性（Consistency）：
 - 事务操作之后,数据库所处的状态和业务规则是一致的;比如a,b账户相互转账之后，总金额不变！
 - 如果数据库系统运行中发生故障，有些事务尚未完成就被迫中断，这些未完成事务对数据库所做的修改有一部分已写入物理数据库，这时数据库就处于一种不正确的状态，或者说的不一致的状态
- 隔离性（Isolation）：
 - 事务的隔离级别有4级，一个事务的执行不能有其他事务的干扰，事务的内部操作和使用数据对其他的并发事务是隔离的，互不干扰。
- 持久性（Durability）：
 - 事务一旦提交后，他对数据的改变应该永久性的，不能回滚。接下来其他的操作或者故障不会对已经提交了的事务产生影响。

事务隔离级别

隔离级别决定了一个session中的事务可能对另一个session中的事务的影响。ANSI标准定义了4个隔离级别，MySQL的InnoDB和Oracle都支持，分别是：

- Oracle共支持3种事务隔离级别：Oracle默认的隔离级别是read committed。
 - serializable
 - read committed
 - read only （Oracle自己独有的事务隔离级别）

- [隔离级别](#)
- READ UNCOMMITTED(读取未提交): 幻读, 不可重复读和脏读均允许; 通常称为dirty read.
 - 则其他线程可以看到未提交的数据, 因此就出现脏读;
- READ COMMITTED (读取已经提交): 允许幻读和不可重复读, 但不允许脏读;
 - 即没提交的数据别人是看不见的, 就避免了脏读
 - 正在读取的数据只获得了读取锁, 读完之后就解锁, 不管当前事务有没有结束, 这样就容许其他事务修改本事务正在读取的数据。导致不可重复读。
- REPEATABLE READ (可重复读): 允许幻读, 但不允许不可重复读和脏读;
 - 对正在操作的数据加锁, 并且只有等到事务结束才放开锁, 则可以避免不可重复读;
 - 只能保证正在被本事务操作的数据不被其他事务修改, 却无法保证有其他事务提交新的数据
 - 比如: 线程1在操作表T1的时候(特别是统计性的事务), 其他线程仍然可以提交新数据到表T1, 这样会导致线程1两次统计的结果不一致, 就像发生幻觉一样(幻读)。
- SERIALIZABLE (序列化): 幻读, 不可重复读和脏读都不允许; 最高级别的隔离, 只允许事务串行执行。
 - 因为获得范围锁, 且事务是一个接着一个串行执行, 则保证了不会发生幻读。
 - 由此可见, 隔离级别越高, 受其他事物干扰越少, 并发性能越差。
- ORACLE默认的是 READ COMMITTED。
- MYSQL默认的是 REPEATABLE READ。

查看事务隔离级别的方法

```
1. SELECT * FROM dual FOR UPDATE;
2. SELECT s.sid, s.serial#,
   CASE BITAND(t.flag, POWER(2, 28))
     WHEN 0 THEN 'READ COMMITTED'
     ELSE 'SERIALIZABLE'
   END AS isolation_level
   FROM v$transaction t
  JOIN v$session s ON t.addr = s.taddr AND s.sid =
sys_context('USERENV', 'SID');
设置隔离级别使用
SET TRANSACTION ISOLATION LEVEL [READ UNCOMMITTED|READ
COMMITTED|REPEATABLE > > > READ|SERIALIZABLE]
```

修改事务隔离级别

- 在MySQL中默认事务隔离级别是可重复读(Repeatable read). 可通过SQL语句查询: mysql> SELECT @@global.tx_isolation
- 查看InnoDB系统级别的事务隔离级别: mysql> SELECT @@tx_isolation
- 修改mysql: mysql> set global transaction isolation level read committed;
- 修改InnoDB: mysql> set session transaction isolation level read committed;

事务并发带来的问题

- 幻读: 幻读的重点在于新增或者删除, 同样条件下两次读出来的记录数不一样。

- 一个事务重新执行一个查询，返回一套符合查询条件的行，发现这些行因为其他最近提交的事务而发生了改变。
- 事务T1读取一条指定的WHERE子句所返回的结果集。然后事务T2新插入一行记录，这行记录恰好可以满足T1所使用的查询条件中的WHERE子句的条件。然后T1又使用相同的查询再次对表进行检索，但是此时却看到了事务T2刚才插入的新行。这个新行就称为“幻像”，因为对T1来说这一行就像突然出现的一样。
- **举例：**目前分数为90分以上的学生有15人，事务A读取所有分数为90分以上的学生人数有15人。此时，事务B插入一条分数为99的学生记录。这时，事务A再次读取90分以上的学生，记录为16人。此时产生了幻读。
- 大部分数据库缺省的事物隔离级别都会出现这种状况，此种事物隔离级别将带来表级锁
- **不可重复读：**不可重复读的重点是修改，同样条件下两次读取结果不同，也就是说，被读取的数据可以被其它事务修改；
 - 一个事务重新读取前面读取过的数据，发现该数据已经被另一个已提交的事务修改过。
 - 事务T1读取一行记录，紧接着事务T2修改了T1刚才读取的那一行记录。然后T1又再次读取这行记录，发现与刚才读取的结果不同。这就称为“不可重复”读，因为T1原来读取的那行记录已经发生了变化。
 - **举例：**在事务A中，读取到小明的分数为89，操作没有完成，事务还没提交。与此同时，事务B把小明的分数改为98，并提交了事务。随后，在事务A中，再次读取小明的分数，此时分数变为98。在一个事务中前后两次读取的结果并不致，导致了**不可重复**。
- **脏读：**
 - 一个事务读取了其另一个未提交的并行事务写的数据。
 - 事务T1更新了一行记录的内容，但是并没有提交所做的修改。事务T2读取更新后的行，然后T1执行回滚操作，取消了刚才所做的修改。现在T2所读取的行就无效了，即脏数据。
 - **举例：**小明的分数为89,事务A中把他的分数改为98,但事务A尚未提交。与此同时，事务B正在读取小明的分数，读取到小明的分数为98。随后，事务A发生异常，而回滚了事务。小明的分数又回滚为89。最后，事务B读取到的小明的分数为98的数据即为脏数据事务B做了一次脏读。

MYSQL的事务处理主要有两种方法

- 任何SQL语句如果仅仅是运行，是不会对数据库做持久化修改的，必须提交事务才能完成持久化保持。MySQL数据库默认自动提交事务。
- 关闭MySQL自动提交：set autocommit = 0;
- 开启MySQL自动提交：set autocommit = 1;
- 查看当前自动提交状态：show variables like "autocommit"; // on/off
- 手动提交：update user set money = 5000 where id = 2;
- 用BEGIN,ROLLBACK,COMMIT来实现
 - 开始：START TRANSACTION或BEGIN语句可以开始一项新的事务
 - 提交：COMMIT可以提交当前事务，是变更成为永久变更
 - 回滚：ROLLBACK可以回滚当前事务，取消其变更
- 直接用set来改变mysql的自动提交模式
 - MYSQL默认是自动提交的，也就是你提交一个QUERY，它就直接执行！
 - 我们可以通过set autocommit=0/1 禁止自动提交/开启自动提交
 - 但注意当你用 set autocommit=0 的时候，你以后所有的SQL都将做为事务处理，直到

你用commit确认或rollback结束，并且只用于当前连接。

事务分为显性事务和隐性事务

- 显性事务：使用BEGIN TRANSACTION明确指定的事务
- 隐性事务：select、insert、update、delete语句都是隐性事务的一部分。
- 自动提交事务：每一条单独的SQL语句都是一个事务，如果成功执行，就提交，否则回滚。

mysql的事务支持

MySQL的事务支持不是绑定在MySQL服务器本身，而是与存储引擎相关

- MyISAM：不支持事务，用于只读程序提高性能；
- InnoDB：支持ACID事务、行级锁、并发；
- Berkeley DB：支持事务。

锁：排它锁、共享锁、乐观锁、悲观锁。

问题：二个或以上事务在操作同一个共享记录集时，可能会出现的问题：

- (A) 脏读 (B) 不可重复读 (C) 幻读
- 隔离级别：(1) read-uncommit, (2) read-commit, (3) read-repeatable, (4) read-serializable
- 都是用来阻止上面的问题的，其中：
 - (1) 什么都阻止不了。
 - (2) 阻止 (A)
 - (3) 阻止 (A) (B)
 - (4) 阻止 (A) (B) (C)
 - (1) --->(4)隔离级别越高，性能损失越大。