

## 1.事务的实现->mysql锁机制->innodb->b-tree/b+tree

## 2.自动提交

```
show variables like "%autocommit%";
```

--全局修改

```
set global autocommit=0;
```

--当前会话

```
set session autocommit=0;
```

--终极修改

修改配置文件

```
autocommit=0
```

## 3.事务

### 3.1事务的四大特性

原子性:在操作数据时候要么成功,要么失败

隔离性:一个事务的影响在该事务提交前对其它的事务都不可见

持久性:事务一旦提交,结果保存磁盘

一致性:事务从一种状态转换成另一种状态

### 3.2事务的生命周期

开启事务(begin) -> 创建日志文件 -> 执行业务逻辑进行提交(根据实际的SQL写入redo和undo但没有执行SQL) -> 刷新日志 -> 数据写入磁盘也是SQL最终的执行流程的环节 -> 写入ckpt日志文件数据,保持数据的一致性

两个日志都在 提交前 写入, 然后 【提交事务】才会刷新日志

### 3.3

3.3.1原子性:事务的日志:redo log(重做日志) 和 undo log(回滚日志) => data目录下

redo => ib\_logfile

undo => ibdata

mysql中讲究日志先行

3.3.2一致性:undo中存储的是与执行SQL相反的语句,一般在运行事务回滚命令的时候应用或者在mysql服务器挂了,再次开机执行redo log的时候执行undo log(保持事务的一致性)

3.3.3持久性:执行事务提交后,写入了redo log 但是mysql服务器挂了,再次开机执行redo log

### 3.3.4隔离性(不同的隔离级别):

1.未提交读/以提交读/可重复度/序列化

2.锁类型:

2.1表级锁:mysqlam:锁住一张表,开销小,速度快,并发量最小,锁冲突的概略高

2.2页级锁:BDB(了解):锁住一页的数据,一页大小16k

2.3行级锁:innodb:锁住一行的数据,开销大,加锁慢,并发量高,所冲突低

2.3.1排他锁:for update(能读不能写,不能与任何锁一起使用)

2.3.2共享锁:lock in share mode(能读不能写,不能与排他锁一起使用,可以与共享锁一起使用)

普通的查询不受锁的影响,悲观锁(加了锁就算),乐观锁(MVCC)

## 4.innodb与mysqlam主键索引与普通索引的区别

innodb普通索引指向的是主键索引的位置,而主键索引的指向的是数据在磁盘中的准确的地址,这样设计与innodb的行锁有关系  
b+tree 与 b-tree 索引结构区别



