======mysql======

分布式 id 雪花算法原理 数据库步长原理

todo gjw 订单分表场景怎么按用户关联多个分表查

8.一亿数据的 a 表 十亿数据的 b 表，按都有的 tid 关联数据查出第 10001 到 10200 的数据

```
1、如果A表TID是自增长,并且是连续的,B表的ID为索引

select * from a,b where a.tid = b.id and a.tid>500000 limit 200;

2、如果A表的TID不是连续的,那么就需要使用覆盖索引.TID要么是主键,要么是辅助索引,B表ID也需要有索引。

select * from b , (select tid from a limit 50000,200) a where b.id = a .tid;
```

7.Select count（1） count（*） count（clonum）区别

```
效果：两者的返回结果是一样的。
意义：当count的参数是具体值时（如count(1)，count('a')），
count的参数已没有实际意义了。

范围：在统计范围，count(*)和count(1) 一样，都包括对NULL的统计；
            count(column) 是不包括NULL的统计。

速度：表没有主键(Primary key)，count(1)比count(*)快；
            否则，主键作为count的参数时，count(主键)比count(1)和count(*)都快；
            表只有一个字段，count(*)，count(1)和count(主键)速度一样。
```

15.持久性怎么保证

```
1 人赞同了该文章
一、binlog日志
binlog其实在日常的运维开发中是听得很多的，因为很多时候数据库数据的更新就是依赖binlog日志的。

比如我们的数据保存在数据库中，现在我们有新的数据写入，或者有数据被更改致使数据库发生变更，而用户检索出来数据是走搜索引擎的，为了用户能搜到最新的数据我们需要把引擎的数据也改掉。

总之就是：数据库变更，搜索引擎也需要变更，于是我们就会监听binlog的变更，如果binlog变更了。那么我们就需要将变更写入到对应的数据源中

什么是binlog?

binlog记录了数据库表结构和表数据变更,比如update/delete/insert/truncate/create。它不会
```

记录select，因为这个没有对表进行变更。

binlog到底是什么样的？binlog我们可以理解位存储着每条变更的SQL语句。

binlog一般用来干什么？

主要的两个作用：复制和恢复数据

。MySQL在公司使用的时候往往都是一主多从结构的，从服务器需要与主服务器的数据保持一致，这就是通过binlog来是实现的，

。数据库的数据被干掉了，我们可以通过binlog来对数据进行恢复。

因为binlog记录了数据库表的变更，所以我们可以用binlog进行复制和恢复数据

二、什么是redo log
我们来看一条sql语句

```
update user_table set name='java3y' where id = '3';
```

MySQL执行这条语句，肯定先把id=3这条语句查出来，然后将name字段给改掉。这没问题吧？

实际上MySQL的基本存储结构是页，所以MySQL是先把这条记录所在的页找到，然后把该页加载到内存中，将对应记录进行修改。

现在就可能出现一个问题；如果在内存中把数据改了还没来得及写入磁盘，而此时的数据库挂了怎么办，显然这次修改就丢失了。

如果每个请求都需要将数据立马写入磁盘之后，那速度会很慢，MySQL可能也顶不住，所以MySQL引入了redo log，内存写完了，然后会写一份redo log，这份redo log 记载着这次在某个页做了什么修改。

其实写redo log的时候，也会有buffer，是先写buffer，在真正写入到磁盘中的，至于从buffer什么时候写入磁盘，会有配置供我们配置。

写redo log也是需要写磁盘的，但它的好处就是顺序IO，所以，redo log 的存在为了当我们修改的时候，写完内存了，但数据还没真正写到磁盘的时候，此时数据库挂了，我们可以对数据进行恢复。因为redo log 是顺序IO，所以写入速度很快，并且redo log 记载的是物理变化，文件的体积小，恢复速度很快。

redo log的作用是为了持久化而生的。写完内存，如果数据库挂了，那我们可以通过redo log来恢复内存还没来得及写入磁盘的数据，将redo log加载到内存里面，那内存就能恢复到挂掉之前的数据了

三、什么是undo log
undo log主要有两个作用：回滚和多版本控制(MVCC)

在数据修改的时候，不仅记录了redo log，还记录undo log，如果因为某些原因导致事务失败或回滚了，可以用undo log进行回滚

undo log主要存储的也是逻辑日志，比如我们要insert一条数据了，那undo log会记录的一条对应的delete日志。我们要update一条记录时，它会记录一条对应相反的update记录。

这也应该容易理解，毕竟回滚嘛，跟需要修改的操作相反就好，这样就能达到回滚的目的。因为支持回滚操作，所以我们就能保证：“一个事务包含多个操作，这些操作要么全部执行，要么全都不执行”。

【原子性】

因为undo log存储着修改之前的数据，相当于一个前版本，MVCC实现的是读写不阻塞，读的时候只要返回前一个版本的数据就行了。

Mysql有哪些日志 16.Redolog binlog的区别

3. redo log与binlog的区别
第一:redo log是在InnoDB存储引擎层产生，而binlog是MySQL数据库的上层产生的，并且二进制日志不仅仅针对INNODB存储引擎，MySQL数据库中的任何存储引擎对于数据库的更改都会产生二进制日志。

第二：两种日志记录的内容形式不同。MySQL的binlog是逻辑日志，其记录是对应的SQL语句。而innodb存储引擎层面的重做日志是物理日志。

第三：两种日志与记录写入磁盘的时间点不同，二进制日志只在事务提交完成后进行一次写入。而innodb存储引擎的重做日志在事务进行中不断地被写入，并日志不是随事务提交的顺序进行写入的。

二进制日志仅在事务提交时记录，并且对于每一个事务，仅在事务提交时记录，并且对于每一个事务，仅包含对应事务的一个日志。而对于innodb存储引擎的重做日志，由于其记录是物理操作日志，因此每个事务对应多个日志条目，并且事务的重做日志写入是并发的，并非在事务提交时写入，其在文件中记录的顺序并非是事务开始的顺序。

第四：binlog不是循环使用，在写满或者重启之后，会生成新的binlog文件，redo log是循环使用。

第五：binlog可以作为恢复数据使用，主从复制搭建，redo log作为异常宕机或者介质故障后的数据恢复使用。

10.数据库 in用的了索引吗

结论：IN肯定会走索引，但是当IN的取值范围较大时会导致索引失效，走全表扫描

3.两张表用户表user和帖子表thread，查找发帖数量前10的用户名和发帖数量

常见实现

常见实现是在 thread 表的 uid 上建索引，并通过下面这条 SQL 查出前10名发帖最多的 uid 及帖子数量

```
select uid,count(*) as total from thread group by uid order by total desc limit 10;
```
然后通过查到的 uid 获取用户名(uid有索引)

```
select uid, usernmae from user where uid in (%s, %s, %s);
查出数据后缓存 1 到 2 分钟，性能也还可以。如果要写成一条 SQL 就加个联表查询。
```

9.死锁 答了死锁条件、死锁避免 银行家算法 8.锁是锁聚集索引还是非聚集索引 11.事务的特性 16.慢查询怎么排查 Explain 9.事物的隔离级别 3.数据库 abc ab% %bc “” null 哪些可以用索引 15.最左匹配原则 4.数据库所有类型字段都能建索引吗 7.Mysql有那些锁 10.Mysql索引数据结构 11.Rc怎么实现的 mvcc 12.索引结构了解吗 13.为什么2000万数据是三层 9.Mysql 索引数据结构 聚集非聚集 14.数据库b+树结构 13.乐观锁的实现方式 15.联合索引 abc 为什么bc不行 16.什么时候建立索引 17.城市字段适合建索引吗