mysql笔记：

1：mysam和innodb的区别

mysam在内存中运行

底层的是非聚簇索引 B+树的节点存储的是地址而不像innodb 存储的是数据本身；

innodb 支持acid事务

2：

读未提

读已提交 解决了脏读

重复读 解决了不可重复读

串行化 解决了幻读

3：

varchar 存放大小 和数据本身

char存放的是固定长度数据

如果数据不是很大 用char 数据对齐 速度更快

4：

prikey是数据的标识

候选键是 可以唯一标识一条记录的属性集

5：

BLOB 区分大小写；

TEXT 是不区分大小写

6

select \* from ab limit 0,50 限制50个；

7

联合索引就是 从左往右开始匹配 最多16个

8

mysql执行语句的：

请求=>链接 授予权限=> 查询缓存 如果缓存里有就直接返回=》 分析语法=》进行方案的优化=》 执行器=》 获得结果

请求=》获得连接=》查询缓存=》 分析语法树=》进行优化

然后where筛选，然后join连接 然后做映射；

9：

使用索引可以极大的提高查询的效率；

10:

覆盖索引

使用辅助索引查询的时候 光索引就满足了所有的查询内容 不需要回表

11：

对于 username 和age建立了索引的事情

没有索引下推 会回表每次比较，有了之后就不会了；

会判断age是否等于99；

MySQL 的 server 层首先调用存储引擎定位到第一个以 j 开头的 username。

找到记录后，存储引擎并不急着回表，而是继续判断这条记录的 age 是否等于 99，如果 age=99，再去回表，如果 age 不等于 99，就不去回表了，直接继续读取下一条记录。

存储引擎将读取到的数据行返回给 server 层，此时如果还有其他非索引的查询条件，server 层再去继续过滤，在我们上面的案例中，此时没有其他查询条件了。假设 server 层还有其他的过滤条件，并且这个过滤条件把刚刚查到的记录过滤掉了，那么就会通过记录的 next\_record 属性读取下一条记录，然后重复第二步。

12:

索引失效

表达式

or

like %开头

13：

字符串 我们往往选一个部分， 测试一下识别度， 然后建立前缀索引

语句怎么写？

14: changebuffer

是mysql发生修改时候 的一个缓冲区， 数据会先落地到缓冲区里面，再挑选合适的时机同步到本地磁盘里面；

15：

redolog 用于崩溃的时候回复

binlog 用于主从同步数据；

16：发生崩溃的时候 redolog 和binlog如何运行；

17：慢查询：

没有走索引

网络延迟

18：主从复制

iothread从master复制binlog

存到本地变成中转日志relaylog；

sql线程执行

SQL

19 mysql的锁

表锁

X S IS Ix意向锁

行级锁：

记录锁

间隙锁

next-keylock

意向插入锁 多个插入锁是不冲突的；

行锁