

初级

- 1 MySQL 5.5 5.6 5.7 8.0 各个版本的特性
- 2 主键索引和非主键索引的区别
- 3 MySQL 各个分支的特点
- 4 什么时候会用到全局读锁
- 5 MyISAM 和 InnoDB 存储引擎的区别
- 6 MySQL 有哪些 SQL 分析方法
- 7 explain 需要注意的项
- 8 什么是元数据锁?
- 9 主从复制原理
- 10 mysqldump 的备份原理
- 11 MySQL 监控项
- 12 MySQL 事务隔离级别有哪些, 特点是什么?
- 13 InnoDB 是如何解决幻读的?
- 14 innodb_flush_log_at_trx_commit参数0、1和2分别代表什么?
- 15 MySQL binlog的几种日志录入格式的涵义和在复制中的优劣
- 16 mysqldump 的参数--master-data 有那几个值, 各自代表什么意思?
- 17 B+树索引和 HASH 索引有哪些区别?
- 18 解释一下两阶段锁协议
- 19 解释一下 MRR 和 ICP
- 20 Redis 适用于哪些场景?
- 21 Redis 有哪些数据结构?
- 22 Redis6 新特性
- 23 MySQL 8.0 新特性
- 24 说几本你认为比较好的 MySQL、Redis 相关书籍或者资料

欢迎关注公众号“悦专栏”，专注分享数据库相关内容，包括 MySQL、Redis、ClickHouse、MongoDB 等。关注后，回复“合集”，可获取“悦专栏”历史原创干货文章。



微信搜一搜



悦专栏

初级

1 MySQL 5.5 5.6 5.7 8.0 各个版本的特性

MySQL 5.6 开始支持 online DDL、基于 database 的多线程复制；

MySQL 5.7 开始支持 基于组提交的多线程复制，支持 json；

MySQL 8.0 Redo log 禁用、快速加列、降序索引、默认字符集 utf8mb4、新的数据字典、json 加强。

快速加列：只需要在表的 metadata 中记录新增列的基本信息即可。

降序索引: <https://dev.mysql.com/doc/refman/8.0/en/descending-indexes.html>

在 8.0降序索引中, 数据被倒序组织, 正向查找。而在之前的版本中, 虽然支持创建降序排列的索引, 但其实现方式是通过创建常见的正序索引, 然后进行反向查找来实现的。

2 主键索引和非主键索引的区别

非主键索引的叶子节点存放的是主键的值

而主键索引的叶子节点存放的是整行数据

3 MySQL 各个分支的特点

Percona server

基于 InnoDB 增加提升性能及易管理性补丁, 形成 xtraDB 引擎, 支持线程池;

extra port (如果连接数满了, 可以额外开个连接进去, 如果使用了thread pool就会有该特性)

慢查询时间最早支持毫秒

基于 Galera 形成 PXC

MariaDB 分支

MySQL 创始人发起

强大开源社区支持

目标是取代 MySQL

SQL 语法完全兼容

最早实现基于组提交的并行复制

4 什么时候会用到全局读锁

全库逻辑备份的时候

5 MyISAM 和 InnoDB 存储引擎的区别

a、InnoDB 支持事务;

b、InnoDB 行锁, MyISAM 表锁

c、InnoDB 的性能远高于 MyISAM 的。在一定线程数之内, 线程数越多, MyISAM 性能与 InnoDB 性能差距越大

d、InnoDB 有 crash-safe 能力

e、InnoDB 只会再备份开始时短暂获取全局读锁, MyISAM 需要在备份整个过程中加全局读锁。

6 MySQL 有哪些 SQL 分析方法

explain: 查看执行计划;

show profile: 确定 SQL 执行过程具体在哪个过程耗时比较久;

trace: 查看优化器如何选择执行计划。

7 explain 需要注意的项

explain 主要项: select_type、type、key、key_len、rows、Extra

select_type: 查询类型: 显示是简单还是复杂查询

type: 本次查询的表连接类型 (const: 基于主键或唯一索引查询一行数据, range: 利用索引范围查询, index: 全索引扫描, ALL: 全表扫描)

key: 实际选择的索引

rows: 预计扫描的行数

Extra: 附加信息 (Using filesort, Using index, Using where)

8 什么是元数据锁?

在 MySQL 5.5 版本中引入了 MDL, 当对一个表做增删改查操作的时候, 加 MDL 读锁; 当要对表做结构变更操作的时候, 加 MDL 写锁。

读锁之间不互斥, 因此你可以有多个线程同时对一张表增删改查。读写锁之间、写锁之间是互斥的, 用来保证变更表结构操作的安全性。因此, 如果有两个线程要同时给一个表加字段, 其中一个要等另一个执行完才能开始执行。

<http://mysql.taobao.org/monthly/2015/11/04/>

9 主从复制原理

1.主库必须开启二进制日志

2.当有增删改的语句时, 会记录到主库的binlog中

3.主库通过IO线程把binlog里面的内容传给从库的relay binlog (中继日志) (这是mysql复制是异步复制的原因)

4.从库的sql线程负责读取它的relay log里的信息并应用到数据库中

10 mysqldump 的备份原理

flush tables (避免元数据锁)

执行 FTWRL(flush tables with read lock);

设置当前会话隔离级别为 RR;

开启事务并创建快照;

获取当前 binlog 位置;

解锁所有表；

对指定的库和表进行 dump；

备份完成。

11 MySQL 监控项

系统相关

系统cpu_iowait

系统cpu负载

系统内存交换

系统文件描述符

系统网络流量（进出）

磁盘利用率

MySQL 状态相关

MySQL 进程状态

QPS

TPS

慢查询数量

锁相关

表锁

行锁

死锁

锁等待

连接相关

最大连接数

活跃连接

主从

io sql线程

从库read_only

主从延迟

主从切换

业务相关

主键自增值

表数据量

12 MySQL 事务隔离级别有哪些，特点是什么？

Read uncommitted: 在该隔离级别，所有事务都可以看到其他未提交的事务的执行结果。读取未提交的事务，称之为“脏读”。

Read Committed: 一个事务只能看见已经提交事务所做的改变。因为同一事务的其他实例在该实例处理期间可能会有新的commit，所以同一select可能返回不同结果。

Repeatable Read: 这是MySQL的默认事务隔离级别，它确保同一事务的多个实例在并发读取数据时，会看到同样的数据行。这种级别会出现幻读。

Serializable: 这是最高的隔离级别，它通过强制事务排序，使之不可能相互冲突，从而解决幻读问题

13 InnoDB 是如何解决幻读的？

通过间隙锁解决幻读的

14 innodb_flush_log_at_trx_commit参数0、1和2分别代表什么？

0

每秒一次地日志缓存区的数据被写到日志文件，并且对日志文件做到磁盘操作的刷新。但是在一个事物提交不做任何操作

性能最好，安全性最差，宕机会丢失一秒的数据

1

在每次事物提交时，日志缓冲被写到日志文件，对日志文件做的磁盘操作的刷新

2

在每个提交，日志缓冲被写到文件，但不对日志文件做到磁盘操作的刷新。会每秒执行一次磁盘操作的刷新。

15 MySQL binlog的几种日志录入格式的涵义和在复制中的优劣

(1) STATEMENT格式:(基于SQL语句)每一条会修改数据的sql都会记录在binlog中。

不需要记录每一行的变化，但是某些特定语句不能复制

(2) ROW: (基于行) 会非常清楚的记录下每一行数据被修改的细节

优点: 复制基本会正常，但是会产生大量的日志。

(3) MIXED: (混合模式) mysql默认采用的STATEMENT格式进行二进制日志

文件的记录，但是在一些情况下会使用ROW格式

16 mysqldump 的参数--master-data 有那几个值，各自代表什么意思？

0: 备份文件不记录二进制信息

1: 备份文件记录二进制日志，并且不会有注释

2: 备份文件记录二进制日志，会有注释

17 B+树索引和 HASH 索引有哪些区别？

B+树索引，可能需要多次运用折半查找来找到对应的数据块（对比跟节点-子树-叶子节点-数据块）

hash索引，只需要通过hash函数，计算出hash值，在表中找出对应的数据。

大量不同数据等值精确查询，HASH索引效率通常比B+TREE高

HASH索引不支持模糊查询、排序、范围查询和联合索引中的最左匹配规则。

18 解释一下两阶段锁协议

两阶段锁：锁操作分为两个阶段，加锁阶段和解锁阶段，并且保证加锁阶段和解锁阶段不相交。

19 解释一下 MRR 和 ICP

MRR：Multi-Range Read 优化是指：在查询辅助索引时，首先更具得到的查询结果，按照主键进行排序，并按照主键排序的顺序进行书签查找。

的目的是为了减少磁盘的随机访问，并且将随机访问转化为较为顺序的数据访问，减少缓冲池中页被替换的次数。

ICP：Index Condition Pushdown

在MySQL 5.6 之前的版本，当进行索引查询时，首先根据索引来查找记录，然后再根据where条件来过滤记录。在MySQL 5.6 支持ICP 后，MySQL 数据库会在取出索引的同时，判断是否可以进行 where 条件的过滤，也就是将where 的部分过滤操作放在存储引擎层。某些查询下，可以大大减少上层 SQL 层对记录的索取。从而提高数据库的整体性能。

20 Redis 适用于哪些场景？

缓存

排行榜

计数器

社交网络

消息队列

等等

21 Redis 有哪些数据结构？

string(字符串)：缓存、计数、共享 session、限速

Hash(哈希)：

list(列表：队列和栈)

Set(集合)

zset(sorted set:有序集合)

22 Redis6 新特性

面向网络处理的多 IO 线程

客户端缓存

细粒度的权限控制

RESP 3 协议的使用

支持 SSL

提供了众多的新模块 API

23 MySQL 8.0 新特性

降序排序

JSON增强

Redo 和 Undo Log 加密

直方图

资源组

角色管理

DDL原子化

快速加列

等等

24 说几本你认为比较好的 MySQL、Redis 相关书籍或者资料

根据自己实际情况讲，最好讲自己看过的。