



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

第三版：MySQL 18 道

存储引擎分类有哪些以及使用场景？

存储引擎主要有

- 1、 MyIsam
- 2、 InnoDB
- 3、 Memory
- 4、 Archive
- 5、 Federated

默认为 InnoDB 引擎。InnoDB 底层存储结构为 B+ 树，B 树的每个节点对应 innodb 的一个 page，page 大小是固定的，一般设为 16k。

使用场景

- 1、 经常更新的表，适合处理多重并发的更新请求
- 2、 支持事务。
- 3、 可以从灾难中恢复(通过 bin-log 日志等)
- 4、 外键约束。只有他支持外键。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

5、支持自动增加列属性 auto_increment

创建索引的原则？

- 1、选择唯一性索引：唯一性索引的值是唯一的。可以更快的通过该索引来确定某条记录。
- 2、为经常需要排序、分组、以及联合查询的列创建索引。
- 3、为经常作为查询条件的列创建索引。
- 4、限制索引的数目：越多的索引越导致表的查询效率变低，因为索引表在每次更新表数据的时候都会重新创建这个表的索引，表的数据越多，索引列越多，那么创建索引的时间消耗就越大。
- 5、如果索引的值很长，那么查询的速度会受到影响。
- 6、如果索引字段的值很长，最好使用值得前缀来进行索引。
- 7、删除不再使用或者很少使用的索引。
- 8、最左前缀匹配原则，非常重要的原则。
- 9、尽量选择区分度高的列作为索引。
- 10、索引列不能参与计算，保持列“干净”：带函数的查询不参与索引。
- 11、尽量扩展索引，不要新建索引。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

索引失效情况? ==校验 SQL 语句是否使用了索引方式为:

在 SQL 语句前面使用 explain 关键字==

- 1、 like 以%开头索引无效, 当 like 以&结尾, 索引有效。
- 2、 or 语句前后没有同事使用索引, 当且仅当 or 语句查询条件的前后列均为索引时, 索引生效。
- 3、 组合索引, 使用的不是第一列索引时候, 索引失效, 即最左匹配规则。
- 4、 数据类型出现隐式转换, 如 varchar 不加单引号的时候可能会自动转换为 int 类型, 这个时候索引失效。
- 5、 在索引列上使用 IS NULL 或者 IS NOT NULL 时候, 索引失效, 因为索引是不索引空值得。
- 6、 在索引字段上使用, NOT, <>, !=, 时候是不会使用索引的, 对于这样的处理只会进行全表扫描。
- 7、 对索引字段进行计算操作, 函数操作时不会使用索引。
- 8、 当全表扫描速度比索引速度快的时候不会使用索引。

索引分类?

单列索引

关注公众号: 磊哥聊编程, 回复: 面试题, 获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

1、普通索引：MySQL 中基本索引类型，没有什么限制，允许在定义索引的列中插入重复值和空值，纯粹为了查询数据更快一点。

2、唯一索引：索引列中的值必须是唯一的，但是允许为空值，

3、主键索引：是一种特殊的唯一索引，不允许有空值。

组合索引：

多个字段组合上创建的索引，只有在查询条件中使用了这些字段的左边字段时，索引才会被使用，使用组合索引时遵循最左前缀集合。

全文索引：

只有在 MyISAM 引擎上才能使用，只能在 CHAR, VARCHAR, TEXT 类型字段上使用全文索引，介绍了要求，说说什么是全文索引，就是在一堆文字中，通过其中的某个关键字等，就能找到该字段所属的记录行，比如有“你是个靓仔，靓女 ...” 通过靓仔，可能就可以找到该条记录

空间索引：

空间索引是对空间数据类型的字段建立的索引，MySQL 中的空间数据类型有四种，GEOMETRY、POINT、LINESTRING、POLYGON。在创建空间索引时，使用 SPATIAL 关键字。要求，引擎为 MyISAM，创建空间索引的列，必须将其声明为 NOT NULL。

linux 添加索引

1、主键索引：ALTER TABLE table_name ADD PRIMARY KEY (column)

2、唯一索引：ALTER TABLE table_name ADD UNIQUE (column)

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、普通索引：ALTER TABLE table_name ADD INDEX index_name (column)

4、全文索引：ALTER TABLE table_name ADD FULLTEXT (column)

5、多列索引：ALTER TABLE table_name ADD INDEX index_name (column1, column2, column3)

什么是索引？

索引其实就是一种数据结构，能够帮助我们快速的检索数据库中的数据。

索引具体采用那种数据结构呢？

常见的 MySQL 主要有两种结构：hash 索引和 B+Tree 索引，我们使用的是 innodb 引擎，默认的是 B+ 树。

既然提到了 InnoDB 使用户的 B+ 树的索引模型

那么你知道为什么采用 B+ 树吗？这和 Hash 索引比较起来有什么缺点吗？

因为 hash 索引底层是哈希表，哈希表是一种以 key-value 存储数据的结构，所以多个数据在存储关系上是完全没有任何顺序关系的，所以，对于区间查询是无法直接通过索引查询的，就需要全表扫描。所以，哈希索引只适用于等值查询的场景。而 B+ 树是一种多路平衡查询树，所以他的节点是天然有序的（左子节点小于于父节点，父节点小于于右子节点），所以对于范围查询的时候不需要做全表扫描。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

B+ Tree 索引和 Hash 索引区别?

- 1、 hash 索引适合等值查询，但是无法进行范围查询。
- 2、 hash 索引没办法利用索引完成排序。
- 3、 hash 索引不支持多列联合索引的最左匹配规则。
- 4、 如果有大量重复键值得情况下，hash 索引的效率会很低，因为哈希碰撞问题。

B+Tree 的叶子节点都可以存放哪些东西?

innoDB 的 B+Tree 可能存储的是整行数据，也有可能是主键的值。

innoDB 的 B+Tree 存储整行数据和主键的值得区别?

- 1、 整行数据：innoDB 的 B+Tree 存储了整行数据的是主键索引，也被成为聚簇索引。
- 2、 存储主键的值：成为非主键索引，也被称为非聚簇索引

聚簇索引和非聚簇索引，在查询数据的时候有区别吗？为什么？

聚簇索引查询会更加快些。因为主键索引树的叶子节点存储的是整行数据。也就是我们需要得到的数据。而非主键索引的叶子节点是主键的值，查询的主键之后，我们还需要通过主键的值再次进行查询数据。（这个过程被称之为回表）。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

非主键索引一定会查询多次吗？

不一定？因为通过覆盖索引也可以只查询一次。

覆盖索引是什么？

1、覆盖索引指的是一个查询语句的执行只用从索引中就能获取到。不必从数据表中读取。也可以被称之为索引覆盖。当一条查询语句符合覆盖索引条件时候，MySQL只需要通过索引就可以返回查询所需要的数据。这样就可以避免回表操作，减少 I/O 提高效率。

2、表 `covering_index_sample` 中有一个普通索引 `idx_key1_key2(key1,key2)`。当我们通过 SQL 语句：

3、`select key2 from covering_index_sample where key1 = 'keytest'`；的时候，就可以通过覆盖索引查询，无需回表。

最左匹配原则？

在创建联合索引时候，一般需要遵循最左匹配原则。即联合索引中的属性识别度最高的放在查询语句的最前面。

MySQL5.6 和 MySQL5.7 对索引做了哪些优化？

1、MySQL5.6 引入了索引下推优化，默认是开启的。

2、例子：user 表中 (a,b,c) 构成一个索引。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题



微信搜一搜

磊哥聊编程

扫码关注



回复：面试题 获取最新版面试题

3、 `select * from user where a='23' and b like '%eqw%' and c like 'dasd'.`

4、 解释：如果没有索引下推原则，则 MySQL 会通过 `a='23'` 先查询出一个对应的数据。然后返回到 MySQL 服务端。MySQL 服务端再基于两个 `like` 模糊查询来校验 `and` 查询出的数据是否符合条件。这个过程就设计到回表操作。

5、 如果使用了索引下推技术，则 MySQL 会首先返回返回条件 `a='23'` 的数据的索引，然后根据模糊查询的条件来校验索引行数据是否符合条件，如果符合条件，则直接根据索引来定位对应的数据，如果不符合直接 `reject` 掉。因此，有了索引下推优化，可以在有 `like` 条件的情况下，减少回表的次数。

怎么查询 SQL 语句是否使用了索引查询？

使用 `explain` 查询 SQL 语句的执行计划，通过执行计划来分析索引的使用情况。

优化器的执行过程？

- 1、 根据搜索条件，找出可能使用的索引。
- 2、 计算全表扫描的代价。
- 3、 计算使用不同索引执行查询的代价。
- 4、 对比各种执行方案的代价，找出成本最低的一个。

关注公众号：磊哥聊编程，回复：面试题，获取最新版面试题