

大家好，很高兴我们可以继续学习交流Java高频面试题。第五章主要是对MySQL数据库进行浅析，我们主要针对面试中常见的MySQL知识点进行交流分析。在对高频面试题分析的过程中，加入实际工作中经常需要使用到的MySQL优化等知识，希望大家可以有效的理解与掌握。

面试中针对MySQL数据库的考察主要集中在以下的知识点：

- MySQL索引
- 常用存储引擎
- MySQL日志模块
- 事务与隔离级别
- MySQL的锁机制
- SQL优化，线上故障排查

在本小节中，我们主要介绍MySQL数据库索引以及存储引擎相关知识点。让我们一起来学习MySQL数据库相关知识点吧~

（1）MySQL的索引有了解吗？（重点掌握）

答：索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可**快速访问数据库表**中的特定信息，就像一本书的目录一样，可以加快查询速度。InnoDB 存储引擎的索引模型底层实现**数据结构为B+树**，所有数据都是存储在B+ 树中的。

解析：

MySQL的索引是面试必考的一个知识点，希望大家可以理解掌握。不同的存储引擎中存在不同的索引结构，本小节中，我们以最常见的InnoDB为例来阐述。在面试中，还可以接着考察下边的索引知识点。

面试官：“为什么底层数据结构使用B+树，而不是B树？”

- B+树是B树的变种，B+树的非叶子节点只用来保存索引，不存储数据，所有的数据都保存在叶子节点；而B树的非叶子节点也会保存数据。这样就使得B+树的查询效率更加稳定，均为从根节点到叶子节点的路径。
- B+树的内部结点并没有指向关键字具体信息的指针，因此其内部结点相对B 树更小，同样空间可以读入更多的节点，所以B+树的磁盘读写代价更低。

面试官：“聚簇索引和非聚簇索引有了解吗？”

聚簇索引也称为**主键索引**，其索引树的叶子节点中存的是**整行数据**，表中行的物理顺序与键值的逻辑（索引）顺序相同。一个表只能包含一个聚集索引。因为索引（目录）只能按照一种方法进行排序。

非聚簇索引（普通索引）的叶子节点内容是**主键的值**。在 InnoDB 里，非主键索引也被称为二级索引（secondary index）。

案例解析：

为了让大家更好的理解主键索引和普通索引的区别，我们给出一个具体的案例如下：

我们先来创建一张表User，主键为id，并且拥有字段uid和字段name，uid字段上有索引，建表语句如下所示：

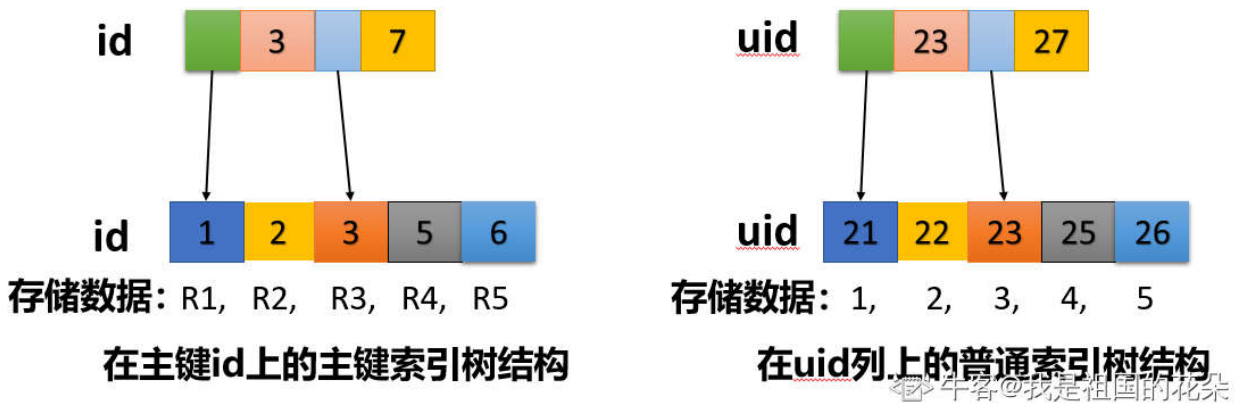
```
1 create table User(  
2   id int primary key,  
3   uid int not null,  
4   name varchar(16),  
5   index (uid))engine=InnoDB;
```

8

接着我们插入如下几条数据：

```
1 insert into User values(1,21,'zhangsan'),(2,22,lisi),(3,23,'wangwu'),(5,25,'ywq'),(6,26,'dym')
```

我们知道，主键上自动创建了主键索引，并且我们手动在uid字段上创建的普通索引。接下来，我们一起看下主键索引树和普通索引树的形状吧~



由上图可以看出，左边主键索引树的叶子节点存储的是完整的记录，而普通索引树上存储的是其对应的主键的值。那么主键索引和普通索引在查询方面的差距也就显而易见了吧~

MySQL回表：

- 如果语句是 `select * from User where id=3`，即主键查询方式，则只需要搜索 主键索引树。
- 如果语句是 `select * from User where uid=23`，即普通索引查询方式，则需要先搜索 普通索引树，得到其对应的主键值为 3，再到主键索引树搜索一次。这个过程称为**回表**。

覆盖索引：

如果在普通索引树上的查询已经直接提供了结果，不需要回表操作，这样的普通索引叫做覆盖索引。覆盖索引的使用可以显著提高查询效率，是常见的MySQL性能优化手段。

索引的最左前缀原则：

在联合索引的情况下，不需要索引的全部定义，只要满足最左前缀，就可以利用索引来加快查询速度。这个最左前缀可以是联合索引的最左 N 个字段，也可以是字符串索引的最左 M 个字符。最左前缀原则的利用也可以显著提高查询效率，是常见的MySQL性能优化手段。

索引下推：

在MySQL5.6之前，只能从根据最左前缀查询到ID开始一个个回表。到主键索引上找出数据行，再对比字段值。MySQL5.6引入的索引下推优化，（联合索引前提）可以在索引遍历过程中，对索引中包含的其余字段先做判断，直接过滤掉不满足条件的记录，减少回表次数，提升查询效率。

8

面试官：“哪些列上适合创建索引？创建索引有哪些开销？”

经常需要作为条件查询的列上适合创建索引，并且该列上也必须有一定的区分度。创建索引需要维护，在插入数据的时候会重新维护各个索引树（数据页的分裂与合并），对性能造成影响。

自增主键：

为了在插入数据的时候不需要调整主键索引树的结构，强烈建议在建立表的时候使用自增主键。主键的顺序按照数据记录的插入顺序排列，自动有序。

(2) MySQL常见的存储引擎有哪些（重点掌握）？

答：MySQL中最常见的存储引擎有InnoDB和MyISAM，它们的主要区别如下：

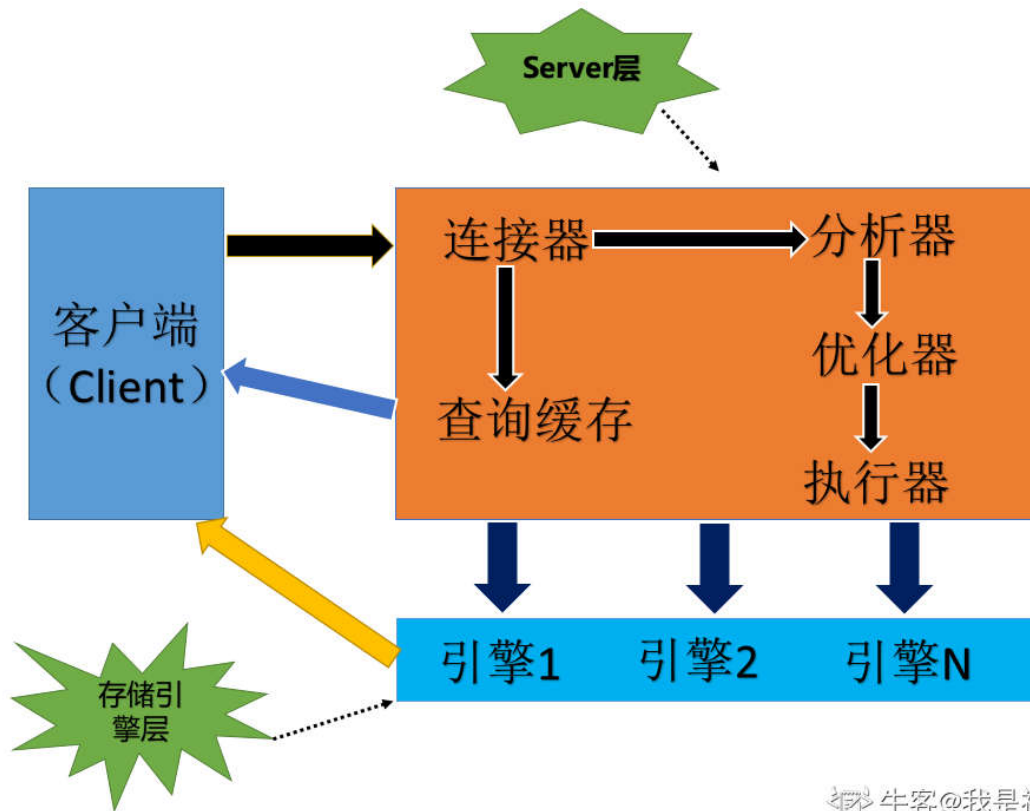
- MyISAM不支持事务；InnoDB是事务类型的存储引擎。
- MyISAM只支持表级锁；InnoDB支持行级锁和表级锁，默认为行级锁。
- MyISAM引擎不支持外键；InnoDB支持外键。
- 对于count(*)查询来说MyISAM更有优势，因为其保存了行数。
- InnoDB是为处理巨大数据量时的最大性能设计的存储引擎。
- MyISAM支持全文索引（FULLTEXT）；InnoDB不支持。

总结：

最主要的区别就是MyISAM表不支持事务、不支持行级锁、不支持外键。InnoDB表支持事务、支持行级锁、支持外键。

解析：

对MySQL的存储引擎的考察也是一个几乎必考的知识点，为了让大家对存储引擎有一个更好的了解，我们先来看下MySQL的基本逻辑架构图，从整体上了解下MySQL包括哪些逻辑架构。



由MySQL的逻辑架构图我们可以看出，逻辑架构包括**Server层**和**存储引擎层**。其中Server层包括连接器，分析器，优化器以及执行器；存储引擎层包括多种支持的存储引擎。各个逻辑部件的作用如下：

- **连接器**：验证客户端权限，建立和断开MySQL连接
- **分析器**：进行SQL语句的语法分析
- **优化器**：选择索引，生成具体的SQL语句执行计划
- **执行器**：操作存储引擎，执行SQL，返回执行结果
- **存储引擎层**：各个不同的存储引擎都提供了一些读写接口来操作数据库

好了，我们接着说存储引擎的知识点，在MySQL5.5.5版本之后，**InnoDB已经成为了其默认的存储引擎**，也是大部分公司的不二选择，毕竟谁家公司会不要求数据库支持事务呢？谁家公司又可以忍受表级锁导致的读写冲突呢？

除了InnoDB以及MyISAM存储引擎外，常见的考察存储引擎还有**Memory**，使用**Memory**作为存储引擎的表也可以叫做**内存表**，将数据存储在了内存中，所以适合做临时表来使用，在索引结构上支持B+树索引和Hash索引。

(3) MySQL中where、group by、having关键字？

答：这三个MySQL关键字作用可以总结如下：

- where子句用来筛选from子句中指定的操作所产生的的行
- group by 子句用来分组where子句的输出
- having子句用来从分组的结果中筛选行

having和where的区别：

- 语法类似，where搜索条件在进行分组操作之前应用；having搜索条件在进行分组操作之后应用
- having可以包含聚合函数sum、avg、max等
- having子句限制的是组，而不是行

当同时含有where子句、group by 子句、having子句及聚集函数时，执行顺序如下：

- 执行where子句查找符合条件的数据
- 使用group by 子句对数据进行分组
- 对group by 子句形成的组运行聚集函数计算每一组的值
- 最后用having 子句去掉不符合条件的组

上节习题参考答案：

习题回顾：

从浏览器中输入一个网址，比如：<https://www.nowcoder.com/> 之后都发生了啥？请尽可能详细的阐述。

参考答案：

在浏览器中输入一个网址，首先要做的就是**域名的解析**，也就是域名->ip的解析，依次查看了浏览器缓存，系统缓存，hosts文件，路由器缓存，递归搜索根域名服务器，直到**找到最终的ip地址**。

然后就是TCP协议的**三次握手**建立连接，过程我们已经阐述了。由浏览器发送一个HTTP请求；经过路由器的转发，通过服务器的防火墙，该HTTP请求到达了服务器；服务器处理该HTTP请求，返回一个**HTML文件**；浏览器解析该HTML文件，并且**渲染显示**在浏览器端。

这里，我们可以来具体阐述下HTTP协议和TCP协议的关系与区别。

HTTP协议和TCP协议的关系与区别：

HTTP协议是一种基于TCP/IP的应用层协议，进行HTTP数据请求必须先建立TCP/IP连接。HTTP协议提供了封装或者显示数据的具体形式；TCP协议提供了网络通信的能力。两个计算机之间的交流其实就是两个端口之间的数据通信，具体的数据会以什么样的形式展现是以不同的应用层协议来定义的。

总结：

本小节中我们主要针对面试中常见的MySQL知识点进行了交流与学习，包括索引和数据存储引擎几乎是MySQL相关面试中得必考点，希望大家可以有效理解与掌握。下一小节，我们将继续阐述MySQL的日志模块，锁机制和事务相关的知识点。

限于作者水平，文章中难免会有不妥之处。大家在学习过程中遇到我没有表达清楚或者表述有误的地方，欢迎随时在文章下边指出，我会及时关注，随时改正。另外，大家有任何话题都可以在下边留言，我们一起交流探讨。

讨论

评论



刘畅201904211606816

1#

催更

发表于 2019-12-29 08:22:43

赞(1) 回复(1)

我是祖国的花朵 [作者]： 已经更新，加油~

2019-12-29 09:48:41

赞(0) 回复(0)

请输入你的观点

回复

8



牧水s N

2#

打卡

发表于 2020-01-20 10:48:58

赞(0) 回复(0)



Deer77

3#

那个主键索引和普通索引第一行的3、7；23,27啥意思呐？

发表于 2020-03-04 22:22:28

赞(1) 回复(1)

我是祖国的花朵 N 作者：索引树结构的简单示意图，可以学习下B+树的结构。

2020-03-05 17:59:38

赞(0) 回复(0)

请输入你的观点

回复



人见人爱的OFFER收割机

4#

“B+树的内部结点并没有指向关键字具体信息的指针”你好，这句话有点不太理解，B和B+树内部节点不都有关键字吗？关键字具体信息又是指什么呢？望解答，谢谢

发表于 2020-03-28 00:27:01

赞(0) 回复(1)

中都：就是那一行的数据信息，B+Tree树的非叶子节点只储存主键（或者索引键）信息，而B树的所有节点不但储存索引键信息，还储存了该索引对应的行数据信息，那么同样的空间，对于非叶子节点来说，B+Tree能储存的指针就更多（因为它不储存索引键的行数据，只储存索引），导致B+Tree树的整体高度低于B树，进而磁盘IO次数少于B树，因而效率高于B树

2020-04-21 11:39:57

赞(0) 回复(0)

请输入你的观点

回复



转不了身

5#

联合索引树的非叶子节点存储内容是什么样子的？会是联合索引列的数据都在一个节点里吗？

发表于 2020-04-25 20:34:13

赞(0) 回复(0)



Shawn_Liu

6#

05/11 Mark

发表于 2020-05-12 06:25:11

赞(0) 回复(0)



赚多多

7#

你好，请问一下主键索引和联合索引的区别，还有是不是指定了主键，主键就自动创建了索引，而不需要我们自己手动创建

发表于 2020-05-14 16:50:35

赞(0) 回复(1)

8

我是祖国的花朵 (作者)： 指定了主键，主键索引就不需要手动创建了

2020-05-15 09:21:55

赞(0) 回复(0)

回复



牛客83076051号

8#

MySQL中最常见的存储引擎有InnoDB和MyISAM，主要区别

MyISAM支持全文索引（FULLTEXT）；InnoDB不支持。（严谨点是不是要结合版本来说啊）mysql 5.6以后 InnoDB 引擎也支持全文索引吧。

发表于 2020-05-21 03:15:59

赞(0) 回复(1)

我是祖国的花朵 (作者)： 怎么说的，如果面试的是DBA岗位，必然要严谨回答。一般情况下，我们也不需要区分版本了，说白了，大部分面试官可能也不期望你按照版本区分区别（因为他也不清楚>_<）

2020-05-22 13:18:58

赞(0) 回复(0)

回复