

有点东西稍微有点忘，赶紧复习面经学习一下。

对于这个面试，还是准备一下算是比较好的。

一、数据库基础知识

1.索引

索引：对数据库表中的一列或者多列的值进行排序的结构，使用索引可以快速访问数据库表中的特定信息。

DB在执行SQL语句时，默认条件是根据搜索条件全表搜索，遇到匹配结果，就会加入结果集合。

如果创建索引，查询时会一次定位到特定值的行数，大大减少遍历匹配的行数，所以能够明显增加查询的速度。

索引的实现通常是B树以及变种的B+树。

索引的好处：提高查询系统性能，唯一索引可以保证数据唯一性，减少分组和排序的时间。

索引的坏处：耗费时间、占用物理空间、数据增加要耗费较多时间，因为索引也要动态的维护哦。

主键索引、最基本的索引、唯一索引（可以是空值）、复合索引

一些专用语句：create index idx_name_age on student(name, age)

show index from student

可以查看建立好的索引

最佳左前缀选择，很有趣的。

索引失效原因：

条件中使用or。

对于多列、联合索引，不是第一部分，不会使用索引，需要左前缀匹配，才能使用对应的索引。

like模糊查询以%开头，索引会失效。

列类型是字符串，吧数据用引号引起来，否则不会用索引

!=时，也不使用索引。

对索引列运算，此时索引也不会起作用。

存在null时，不起作用。

SQL语句的优化：能用索引尽量用索引，不要用select * from t,

尽量使用where，使用多表查询，where后面 and or 左右执行的顺序是从右到左。

这个是一些小的知识点。

知乎专栏的一些见解：

联合索引：INDEX idx_test(col_a, col_b)

这种包含多个字段的索引就叫做联合索引。

按照顺序一个一个找，需要满足最左前缀匹配，就可以使用该索引。

聚簇索引

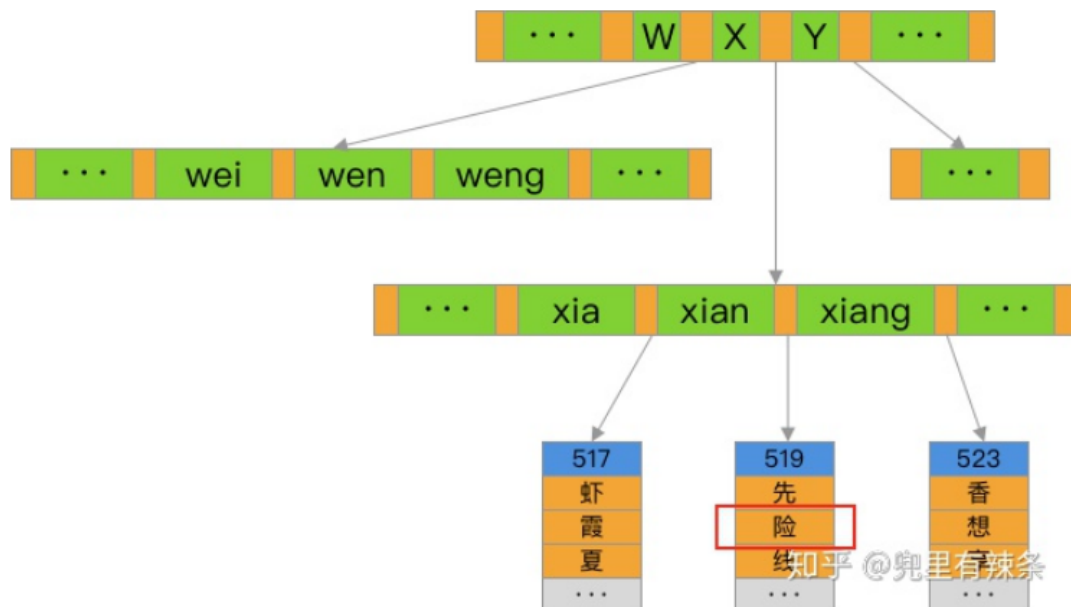
聚集索引：类似于拼音目录，数据会根据索引中的顺序进行排列和组织，叫聚集索引。

非聚集索引：其他的一般索引。

在MySQL的InnoDB存储引擎中，主键索引就是聚集索引，所有数据的主键索引进行组织。

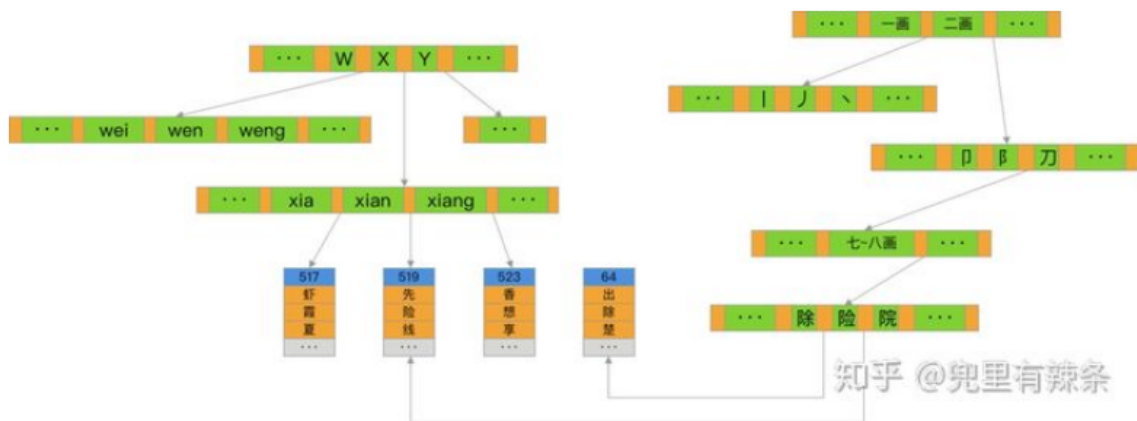
聚集索引使用B+树实现，是一种高阶版本的字典树

数据结构。



非聚集索引：

从非聚集索引的叶子结点连接到聚集索引的叶子结点查询实际数据，这个过程叫 回表

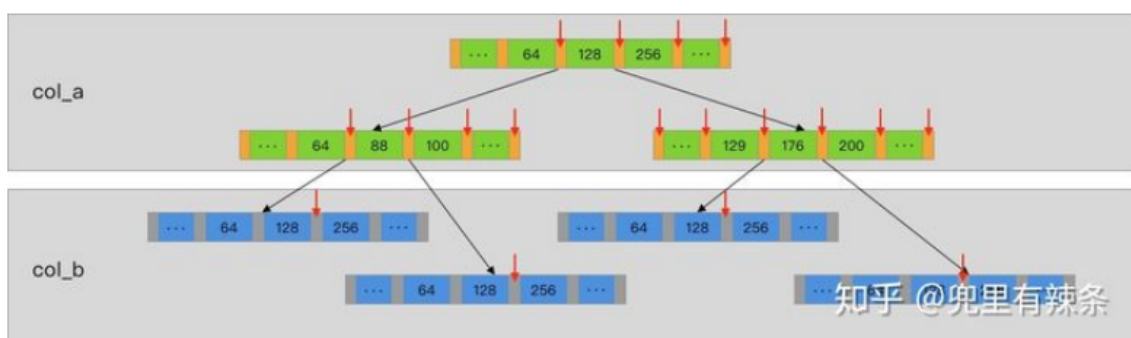


全覆盖索引：在索引中获取SQL语句所需要的字段，就是全覆盖索引。

范围查询条件：

倘若使用范围查询，此时倘若使用联合索引，会遍历B+树过多地节点，得不偿失。

所以这总情况下，数据库会直接比那里所有col_a>63的记录。



索引设计的原则：

分析每个字段的可选择性，手机号明显要比性别的选择性大很多。

一般来说，我们会把可选择性高的字段放在前面，低的放在后面。太低的不会放在索引里面。

索引是根据数据库层的语句来对应设计，并且进行优化的。这些是非常容易理解的。

将索引优化为全覆盖索引的手段：

创建一个同时包含所有字段的索引，以此来提高查询的效率efficiency

减少SQL语句中使用的字段。

绝大多数索引都是用B+树来实现的。

使用场景：当修改性能要求大于检索性能时，不应该创建索引。text、image、bit数据类型不应当增加索引。

二、数据库事务

数据库事务（Transaction）一系列对系统中的数据进行访问和更新操作所组成的一个程序执行逻辑单元，事务是DBSM中的最基础的单位，事务不可分割。

具有最基本的四个特征：ACID 原子性、一致性、隔离性、持久性

并发操作应当具有隔离的特点，实现隔离性的特点？

不同的隔离级别：

Read Uncommitted 一个事务可以读取到另一个事务未提交的结果

Read Committed 可以解决脏读问题

Repeated Read 可重复读，在一个事务中，对一份数据的读取结果总是相同的，无论其他情况。解决脏读、不可重复读问题。

Serialization 可串行化 事务串行化执行，牺牲并发，可以解决并发的所有问题。

三、三大范式

第一范式：关系模式R中的所有数据都不能被分解为更基本的数据单位。

第二范式：满足第一范式，R的所有非主属性都依赖于R的每一个候选关键主属性

第三范式：X非传递依赖于R的任意一个候选关键字，非主属性不传递依赖于猪娃建？

二、MySQL

四种隔离状态

MVCC机制：多版本并发控制机制，是MySQL的InnoDB存储引擎实现隔离级别的一种具体范式。

实现了可重复读隔离级别。

MVCC通过保存数据在某个时间点的快照来实现这个机制，分别会存储这个行的创建版本号 and 删除版本号。

InnoDB使用的也是B+ Tree实现聚集索引，指向真实存储的所有数据，这里的logic还是比较明显的。

外键：用于和另一张表关联，是可以重复的。

简单的级联操作：

```
SELECT * FROM teacher WHERE project = (SELECT label FROM classes WHERE id = 1)
```

复杂的级联操作：

这就是关联数据表的作用，提供了复杂的查询机制。

```
SELECT
```

```
t.NAME AS username,
```

```
c.label AS project,
```

```
c.description AS ds
```

```
FROM
```

```
teacher AS t,
```

```
classes AS c
```

```
WHERE
```

```
t.project = c.label;
```