

公告

wiki和教程：www.5xclass.cn

免费教学视频：[B站](#)：[凸头统治地球](#)

高级专题教程：[网易云课堂](#)：[武沛齐](#)

聊技术，加武Sir微信

 武沛齐



扫一扫上面的二维码图案，加我微信

昵称： 武沛齐
园龄： 11年9个月
粉丝： 12988
关注： 44
[+加关注](#)

我的标签

- Python(17)
- ASP.NET MVC(15)
- python之路(7)
- Tornado源码分析(5)
- 每天一道Python面试题(5)
- crm项目(4)
- 面试都在问什么？(2)
- Python企业面试题讲解(1)
- Python面试315题(1)
- Python开源组件 - Tyrion(1)

积分与排名

积分 - 492822
排名 - 1343

随笔分类

- JavaScript(1)
- MVC(15)
- Python(17)
- 面试都在问什么系列？【图】(2)
- 其他(37)

随笔 - 140 文章 - 164 评论 - 1087 阅读 - 269万

索引补充

1、索引

索引是表的目录，在查找内容之前可以先在目录中查找索引位置，以此快速定位查询数据。对于索引，会保存在额外的文件中。

2、索引种类

- 普通索引：仅加速查询
- 唯一索引：加速查询 + 列值唯一（可以有null）
- 主键索引：加速查询 + 列值唯一 + 表中只有一个（不可以有null）
- 组合索引：多列值组成一个索引，
专门用于组合搜索，其效率大于索引合并
- 全文索引：对文本的内容进行分词，进行搜索

索引合并，使用多个单列索引组合搜索
覆盖索引，select的数据列只用从索引中就能够取得，不必读取数据行，换句话说查询列要被所建的索引覆盖

3、相关命令

```
1 - 查看表结构
2     desc 表名
3
4 - 查看生成表的SQL
5     show create table 表名
6
7 - 查看索引
8     show index from 表名
9
10 - 查看执行时间
11     set profiling = 1;
12     SQL...
13     show profiles;
```

4、使用索引和不使用索引

```
1 由于索引是专门用于加速搜索而生，所以加上索引之后，查询效率会快到飞起来。
2
3 # 有索引
4 mysql> select * from tb1 where name = 'wupeiqi-888';
5 +-----+-----+-----+-----+
6 | nid | name          | email                  | radom |
7 +-----+-----+-----+-----+
8 | 889 | wupeiqi-888  | wupeiqi888@live.com   | 5312269e76a16a90b8a8301d5314;
9 +-----+-----+-----+-----+
10 1 row in set (0.00 sec)
11
12 # 无索引
13 mysql> select * from tb1 where email = 'wupeiqi888@live.com';
14 +-----+-----+-----+-----+
15 | nid | name          | email                  | radom |
16 +-----+-----+-----+-----+
17 | 889 | wupeiqi-888  | wupeiqi888@live.com   | 5312269e76a16a90b8a8301d5314;
```

企业面试题及答案(1)
请求响应(6)
设计模式(9)
微软C#(34)

随机档案

2020年6月(1)
2020年5月(1)
2019年11月(1)
2019年10月(1)
2019年9月(4)
2018年12月(1)
2018年8月(1)
2018年5月(2)
2018年4月(1)
2017年8月(1)
2017年5月(1)
2017年3月(1)
2016年10月(1)
2016年7月(1)
2015年10月(1)
更多

相册

git(14)

最新评论

1. Re:django channels
2
--长街旧人,...
2. Re:Python生成随机验证码
多亏楼上评论 半天找不到这个文章 谢谢武老师 谢谢楼上
--sugaryy
3. Re:Python生成随机验证码
来下字体文件的
--Hinata-
4. Re:Python之路【第十七篇】： Django
【进阶篇】
<script>
alert(123);
</script>
--柠檬の夏天
5. Re:【第4题】 什么是https
最好理解的解说, 赞
--华丽丽的肉虫子

```
18 +-----+-----+-----+-----+
19 1 row in set (1.23 sec)
```

5、正确使用索引

数据库表中添加索引后确实会让查询速度起飞，但前提必须是正确的使用索引来查询，如果以错误的方式使用，则即使建立索引也会不奏效。
即使建立索引，索引也不会生效：

```
1 - like '%xx'
2     select * from tb1 where name like '%cn';
3 - 使用函数
4     select * from tb1 where reverse(name) = 'wupeiqi';
5 - or
6     select * from tb1 where nid = 1 or email = 'seven@live.com';
7     特别的：当or条件中有未建立索引的列才失效，以下会走索引
8         select * from tb1 where nid = 1 or name = 'seven';
9         select * from tb1 where nid = 1 or email = 'seven@live.com' ;
10 - 类型不一致
11     如果列是字符串类型，传入条件是必须用引号引起来，不然...
12     select * from tb1 where name = 999;
13 - !=
14     select * from tb1 where name != 'alex'
15     特别的：如果是主键，则还是会走索引
16     select * from tb1 where nid != 123
17 - >
18     select * from tb1 where name > 'alex'
19     特别的：如果是主键或索引是整数类型，则还是会走索引
20     select * from tb1 where nid > 123
21     select * from tb1 where num > 123
22 - order by
23     select email from tb1 order by name desc;
24     当根据索引排序时候，选择的映射如果不是索引，则不走索引
25     特别的：如果对主键排序，则还是走索引：
26     select * from tb1 order by nid desc;
27
28 - 组合索引最左前缀
29     如果组合索引为：(name,email)
30     name and email      -- 使用索引
31     name                 -- 使用索引
32     email                -- 不使用索引
```

6、其他注意事项

- 1 - 避免使用select *
- 2 - count(1)或count(列) 代替 count(*)
- 3 - 创建表时尽量时 char 代替 varchar
- 4 - 表的字段顺序固定长度的字段优先
- 5 - 组合索引代替多个单列索引（经常使用多个条件查询时）
- 6 - 尽量使用短索引
- 7 - 使用连接（JOIN）来代替子查询(Sub-Queries)
- 8 - 连表时注意条件类型需一致
- 9 - 索引散列值（重复少）不适合建索引，例：性别不适合

7、limit分页

无论是否有索引，limit分页是一个值得关注的问题



每页显示10条：
当前 118 120, 125

倒序：

大

小

980 970 7 6 6 5 54 43 32

21 19 98

下一页：


```
select
  *
from
  tbl
where
  nid < (select nid from (select nid from tbl where nid < 当前页最
order by
  nid desc
limit 10;
```

```
select
  *
from
  tbl
where
  nid < (select nid from (select nid from tbl where nid < 970 o
order by
  nid desc
limit 10;
```

上一页：

```
select
  *
from
  tbl
where
  nid < (select nid from (select nid from tbl where nid > 当前页最
order by
  nid desc
limit 10;
```

```
select
  *
from
  tbl
where
  nid < (select nid from (select nid from tbl where nid > 980 or
order by
  nid desc
limit 10;
```



8、执行计划

explain + 查询SQL - 用于显示SQL执行信息参数，根据参考信息可以进行SQL优化

```
1 mysql> explain select * from tb2;
2 +-----+-----+-----+-----+-----+-----+-----+
3 | id | select_type | table | type | possible_keys | key | key_len | ref
4 +-----+-----+-----+-----+-----+-----+-----+
5 | 1 | SIMPLE      | tb2   | ALL  | NULL          | NULL | NULL    | NULL
6 +-----+-----+-----+-----+-----+-----+-----+
7 1 row in set (0.00 sec)
```

id

查询顺序标识

如: mysql> explain select * from (select nid,name from tbl1

+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | <derived2> | ALL | NULL |
| 2 | DERIVED | tbl1 | range | PRIMARY |
+-----+-----+-----+-----+-----+-----+-----+

特别的：如果使用union连接气值可能为null

select_type

查询类型

SIMPLE

简单查询

PRIMARY

最外层查询

SUBQUERY

映射为子查询

DERIVED

子查询

UNION

联合

UNION RESULT

使用联合的结果

...

table

正在访问的表名

type

查询时的访问方式，性能：all < index < range < index_merge < ref_or

ALL

全表扫描，对于数据表从头到尾找一遍

select * from tbl1;

特别的：如果有limit限制，则找到之后就不在继续向下找

select * from tbl1 where email = 'se'

select * from tbl1 where email = 'se'

虽然上述两个语句都会进行全表扫描，第二句使

INDEX

全索引扫描，对索引从头到尾找一遍

select nid from tbl1;

RANGE

对索引列进行范围查找

select * from tbl1 where name < 'alex';

PS:

between and

in

> >= < <= 操作

注意：!= 和 > 符号

INDEX_MERGE

合并索引，使用多个单列索引搜索

select * from tbl1 where name = 'alex' or :

REF

根据索引查找一个或多个值

select * from tbl1 where name = 'seven';

EQ_REF	连接时使用primary key 或 unique类型 <code>select tb2.nid,tb1.name from tb2 left join</code>
CONST	常量 表最多有一个匹配行, 因为仅有一行, 在这行的列值可被优化 <code>select nid from tb1 where nid = 2 ;</code>
SYSTEM	系统 表仅有一行 (=系统表)。这是const联接类型的一个特例。 <code>select * from (select nid from tb1 where n</code>
possible_keys	可能使用的索引
key	真实使用的
key_len	MySQL中使用索引字节长度
rows	mysql估计为了找到所需的行而要读取的行数 ----- 只是预估值
extra	该列包含MySQL解决查询的详细信息 "Using index" 此值表示mysql将使用覆盖索引, 以避免访问表。不要把覆盖索引和index访问 "Using where" 这意味着mysql服务器将在存储引擎检索行后再进行过滤, 许多where条件里涉 "Using temporary" 这意味着mysql在对查询结果排序时会使用一个临时表。 "Using filesort" 这意味着mysql会对结果使用一个外部索引排序, 而不是按索引次序从表里读取 "Range checked for each record(index map: N)" 这个意味着没有好用的索引, 新的索引将在联接的每一行上重新估算, N是显示

更多参见:

<http://www.cnblogs.com/xiaoboluo768/p/5400990.html>

http://dev.mysql.com/doc/refman/5.7/en/explain-output.html#jointype_system

9、慢日志查询

a、配置MySQL自动记录慢日志

slow_query_log = OFF	是否开启慢日志记录
long_query_time = 2	时间限制, 超过此时间, 则记录
slow_query_log_file = /usr/slow.log	日志文件
log_queries_not_using_indexes = OFF	为使用索引的搜索是否记录

注: 查看当前配置信息:

```
show variables like '%query%'
```

修改当前配置:

```
set global 变量名 = 值
```

b、查看MySQL慢日志

```
mysqldumpslow -s at -a /usr/local/var/mysql/MacBook-Pro-3-slow.log
```

```
1  ""
2  --verbose  版本
3  --debug    调试
4  --help     帮助
5
6  -v         版本
7  -d         调试模式
8  -s ORDER   排序方式
9
10             what to sort by (al, at, ar, c, l, r, t), 'at' is default
11             al: average lock time
12             ar: average rows sent
13             at: average query time
14             c: count
15             l: lock time
16             r: rows sent
17             t: query time
18 -r          反转顺序, 默认文件倒序拍。reverse the sort order (largest last
19 -t NUM      显示前N条just show the top n queries
20 -a          不要将SQL中数字转换成N, 字符串转换成S。don't abstract all numbe
21 -n NUM      abstract numbers with at least n digits within names
22 -g PATTERN  正则匹配; grep: only consider stmts that include this string
23 -h HOSTNAME mysql机器名或者IP; hostname of db server for *-slow.log filer
24             default is '*', i.e. match all
25 -i NAME     name of server instance (if using mysql.server startup scrip
26 -l          总时间中不减去锁定时间; don't subtract lock time from total ti
27 ""
```



作者: [武沛齐](#)
出处: <http://www.cnblogs.com/wupeiqi/>
本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接。

好文要顶

关注我

收藏该文

微信分享



武沛齐
粉丝 - 12988 关注 - 44

6

0

[+加关注](#)

posted @ 2016-07-29 07:50 武沛齐 阅读(11436) 评论(1) 编辑 收藏 举报

会员力量，点亮园子希望

[刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】博客园商业化之路-开篇：开源的脚步，商业化的出路

【推荐】园子周边第二季：更大的鼠标垫，没有logo的鼠标垫

【推荐】阿里云云市场联合博客园推出开发者商店，欢迎关注

【推荐】会员力量，点亮园子希望，期待您升级成为园子会员



编辑推荐:

- [async/await 贴脸输出，这次你总该明白了](#)
- [WPF 随笔收录-实时绘制心率曲线](#)
- [「布局进阶」巧用 :has & drop-shadow 实现复杂布局效果](#)
- [\[Nano Framework ESP32篇\] WS2812 彩色灯带实验](#)
- [经过腾讯云这波故障，我想表扬的点和学到的职场保命法则](#)

 **阿里云**

开发者商店
由阿里云云市场和博客园联合提供

AI、API、基础软件及服务，
产品折扣、先试后买！

[立即查看 →](#)

阅读排行:

- [在Windows电脑上快速运行AI大语言模型-Llama3](#)
- [.NET开源免费的跨平台框架 - MAUI（附学习资料）](#)
- [使用纯c#在本地部署多模态模型，让本地模型也可以理解图像](#)
- [本地部署Llama3-8B/70B 并进行逻辑推理测试](#)
- [如何将 ASP.NET Core MVC 项目的视图分离到另一个项目](#)