


祥龙之子

祥龙之子

随笔 - 1124, 文章 - 0, 评论 - 503, 引用 - 9

导航

博客园
首页
新随笔
联系
订阅 
管理

<	2008年10月						>
日	一	二	三	四	五	六	
28	29	30	1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31	1	
2	3	4	5	6	7	8	

公告

昵称：cy163
园龄：13年1个月
粉丝：370
关注：1
[+加关注](#)

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

Tomcat+Axis+Eclipse(3)
ASP comparision
operators(1)
AUSU (A3v) 华硕笔记本资
料(2)
big little endian
Bolero(1)
C# programming(4)
C# Window project(1)
CLucene, Solr, Lucene(43)
CygWin(1)
DB2(6)
DEV-C++(1)

为mysql数据库建立索引

前些时候，一位颇高级的程序员居然问我什么叫做索引，令我感到十分的惊奇，我想这绝不会是沧海一粟，因为有成千上万的开发者（可能大部分是使用MySQL的）都没有受过有关数据库的正规培训，尽管他们都为客户做过一些开发，但却对如何为数据库建立适当的索引所知较少，因此我起了写一篇相关[文章](#)的念头。

最普通的情况，是为出现在where子句的字段建一个索引。为方便讲述，我们先建立一个如下的表。

Code代码如下：

```
CREATE TABLE mytable (  
  id serial primary key,  
  category_id int not null default 0,  
  user_id int not null default 0,  
  adddate int not null default 0  
);
```

很简单吧，不过对于要说明这个问题，已经足够了。如果你在查询时常用类似以下的语句：

```
SELECT * FROM mytable WHERE category_id=1;
```

最直接的应对之道，是为category_id建立一个简单的索引：

```
CREATE INDEX mytable_categoryid  
ON mytable (category_id);
```

OK，搞定？先别高兴，如果你有不止一个[选择](#)条件呢？例如：

```
SELECT * FROM mytable WHERE category_id=1 AND  
user_id=2;
```

你的第一反应可能是，再给user_id建立一个索引。不好，这

Discuz(1)
 Drupal(14)
 Dvbbs(43)
 Flash(12)
 FusionChart Prefuse Carrot
 Flare 类聚可视化(25)
 Git(1)
 GoJS、GoDiagram(1)
 gSOAP(7)
 GTK(9)
 HTML CSS(2)
 IA - 32、IA - 64、x86 -
 32、x86 - 64(3)
 Icon下载(1)
 ICTCLAS(6)
 Impala(1)
 Java(46)
 Javascript tips(9)
 Java调用C++的DLL(4)
 JAVA中文乱码(9)
 JMail44(1)
 JNI Jace(1)
 journal-center.com(自己的
 网站)(2)
 JSP(14)
 Linux(27)
 MFC, C++, Visual C(195)
 MongoDB(5)
 MPICH2(2)
 MyEclipses Eclipse(5)
 MySQL(103)
 Nutch(9)
 OP710机Java Tomcat
 MySQL的安装软件(2)
 Openlayers, Geoserver(1)
 OpenPBS(12)
 OSC's mpiexec(1)
 Ping
 RIA FLEX(19)
 SCI EI(4)
 SOCKET(8)
 SSH(6)
 STL(3)
 Struts(1)
 Tomcat(10)
 UML(8)
 VS 2005(1)
 Web 2.0 网站投资(1)
 Web Searching(4)
 Web Service(52)
 Windows(6)
 Xfire(Web Service)(3)
 饼图 pie chart jsp(5)
 并行计算(18)
 病毒(3)
 从Word PDF抽取文本(1)
 电子书资源(4)

不是一个最佳的方法。你可以建立多重的索引。

```
CREATE INDEX mytable_categoryid_userid ON mytable (category_id,user_id);
```

注意到我在命名时的习惯了吗？我使用"表名_字段1名_字段2名"的方式。你很快就会知道我为什么这样做了。

现在你已经为适当的字段建立了索引，不过，还是有点不放心吧，你可能会问，数据库会真正用到这些索引吗？测试一下就OK，对于大多数的数据库来说，这是很容易的，只要使用EXPLAIN命令：

EXPLAIN

```
SELECT * FROM mytable
WHERE category_id=1 AND user_id=2;
```

This is what Postgres 7.1 returns (exactly as I expected)

NOTICE: QUERY PLAN:

```
Index Scan using mytable_categoryid_userid on
mytable (cost=0.00..2.02 rows=1 width=16)
```

EXPLAIN

以上是postgres的数据，可以看到该数据库在查询的时候使用了一个索引（一个好开始），而且它使用的是我创建的第二索引。看到我上面命名的好处了吧，你马上知道它使用适当的索引了。

接着，来个稍微复杂一点的，如果有个ORDER BY字句呢？不管你信不信，大多数的数据库在使用order by的时候，都会从索引中受益。

```
SELECT * FROM mytable
WHERE category_id=1 AND user_id=2
ORDER BY adddate DESC;
```

动态连接库(6)
 对联式广告(1)
 多文件上传, 进度条(8)
 多线程(17)
 飞机 编队 形状 战术动作(3)
 服务器安全设置(2)
 负载均衡(3)
 概率与统计(1)
 高校教师(1)
 个人投资 文章 网站(7)
 工作相关(3)
 公开情报(6)
 规则表达式(9)
 好的博客(1)
 好的专业技术论坛地址(2)
 好文共赏(164)
 化境ASP无组件上传
 家中电脑互连, 共享网线上网
 (12)
 建站素材(6)
 健康医疗信息(2)
 军事指挥(1)
 可视化(1)
 马连洼(1)
 普园(2)
 企业异构数据源集成 (863课
 题申请) (1)
 清远通业(1)
 软件产品线(1)
 商铺投资(1)
 设计模式(2)
 省交通厅
 什么是数据驱动
 时空挖掘_Spatial_Temporal
 DM(1)
 视频挖掘(1)
 数据库访问速度 内存访问速度
 (2)
 数据融合
 数据挖掘为本体建设服务(1)
 图片无缝滚动(1)
 网络爬虫(2)
 网络视频标准(1)
 文本地理信息系统(3)
 文本分类(7)
 文本分析(21)
 物流 logistics(5)
 小学英语教学(1)
 信号量 进程(6)
 休闲(4)
 硬盘坏道修复(2)
 语义搜索 Ontology(2)
 预先项目出现问题(2)
 云计算、多核(1)
 植入式广告(6)

有点迷惑了吧?很简单,就象为where字句中的字段建立一个索引一样,也为ORDER BY的字句中的字段建立一个索引:

```
CREATE INDEX mytable_categoryid_userid_adddate
ON mytable (category_id,user_id,adddate);
```

注意: "mytable_categoryid_userid_adddate" 将会被截短为

"mytable_categoryid_userid_addda"

CREATE

```
EXPLAIN SELECT * FROM mytable
WHERE category_id=1 AND user_id=2
ORDER BY adddate DESC;
```

NOTICE: QUERY PLAN:

Sort (cost=2.03..2.03 rows=1 width=16)

-

> Index Scan using mytable_categoryid_userid_addda

on mytable (cost=0.00..2.02 rows=1 width=16)

EXPLAIN

看看EXPLAIN的输出,好象有点恐怖啊,数据库多做了一个我们没有要求的排序,这下知道性能如何受损了吧,看来我们对于数据库的自身运作是有点过于乐观了,那么,给数据库多一点提示吧。

为了跳过排序这一步,我们并不需要其它另外的索引,只要将查询语句稍微改一下。这里用的是postgres,我们将给该数据库一个额外的提示--在ORDER BY语句中,加入where语句中的字段。这只是一个技术上的处理,并不是必须的,因为实际上在另外两个字段上,并不会有任何的排序操作,不过如果加入,postgres将会知道哪些是它应该做的。

EXPLAIN SELECT * FROM mytable

随笔档案

2017年8月 (1)
2017年6月 (2)
2016年4月 (2)
2016年3月 (2)
2016年2月 (1)
2016年1月 (3)
2015年12月 (1)
2015年11月 (1)
2015年6月 (2)
2015年5月 (2)
2015年4月 (6)
2015年3月 (3)
2015年2月 (1)
2015年1月 (1)
2014年12月 (1)
2014年10月 (4)
2014年9月 (1)
2014年8月 (1)
2014年7月 (1)
2014年4月 (1)
2013年12月 (3)
2013年11月 (1)
2013年6月 (1)
2013年5月 (2)
2013年4月 (1)
2013年3月 (3)
2013年2月 (15)
2013年1月 (3)
2012年8月 (1)
2012年2月 (1)
2012年1月 (1)
2011年6月 (1)
2011年4月 (2)
2011年3月 (2)
2011年2月 (2)
2011年1月 (38)
2010年12月 (12)
2010年11月 (8)
2010年10月 (11)
2010年9月 (3)
2010年8月 (3)
2010年7月 (11)
2010年6月 (3)
2010年5月 (11)
2010年4月 (12)
2010年3月 (16)
2010年2月 (16)
2010年1月 (14)
2009年12月 (27)
2009年11月 (15)
2009年10月 (24)
2009年9月 (50)
2009年8月 (32)

```
WHERE category_id=1 AND user_id=2
```

```
ORDER BY category_id DESC,user_id DESC,adddate D  
ESC;
```

NOTICE: QUERY PLAN:

```
Index Scan Backward using  
mytable_categoryid_userid_addda on mytable  
(cost=0.00..2.02 rows=1 width=16)
```

EXPLAIN

现在使用我们料想的索引了，而且它还挺聪明，知道可以从索引后面开始读，从而避免了任何的排序。

以上说得细了一点，不过如果你的数据库非常巨大，并且每日的页面请求达上百万算，我想你会获益良多的。不过，如果你要做更为复杂的查询呢，例如将多张表结合起来查询，特别是where限制字句中的字段是来自不止一个表格时，应该怎样处理呢？我通常都尽量避免这种做法，因为这样数据库要将各个表中的东西都结合起来，然后再排除那些不合适的行，搞不好开销会很大。

如果不能避免，你应该查看每张要结合起来的表，并且使用以上的策略来建立索引，然后再用EXPLAIN命令验证一下是否使用了你料想中的索引。如果是的话，就OK。不是的话，你可能要建立临时的表来将他们结合在一起，并且使用适当的索引。

要注意的是，建立太多的索引将会影响更新和插入的速度，因为它需要同样更新每个索引文件。对于一个经常需要更新和插入的表格，就没有必要为一个很少使用的where字句单独建立索引了，对于比较小的表，排序的开销不会很大，也没有必要建立另外的索引。

以上介绍的只是一些十分基本的东西，其实里面的学问也不少，单凭EXPLAIN我们是不能判定该方法是否就是最优化的，每个数据库都有自己的一些优化器，虽然可能还不太完善，但是它们都会在查询时对比过哪种方式较快，在某些情况下，建立索引的话也未必会快，例如索引放在一个不连续的存储空间

- 2009年7月 (25)
- 2009年6月 (13)
- 2009年5月 (18)
- 2009年4月 (18)
- 2009年3月 (17)
- 2009年2月 (7)
- 2009年1月 (8)
- 2008年12月 (15)
- 2008年11月 (17)
- 2008年10月 (25)
- 2008年9月 (16)
- 2008年8月 (26)
- 2008年7月 (2)
- 2008年6月 (22)
- 2008年5月 (28)
- 2008年4月 (35)
- 2008年3月 (3)
- 2008年2月 (12)
- 2008年1月 (20)
- 2007年12月 (1)
- 2007年10月 (1)
- 2007年9月 (4)
- 2007年8月 (1)
- 2007年7月 (2)
- 2007年6月 (15)
- 2007年5月 (28)
- 2007年4月 (15)
- 2007年3月 (13)
- 2007年2月 (24)
- 2007年1月 (6)
- 2006年12月 (3)
- 2006年11月 (35)
- 2006年10月 (37)
- 2006年9月 (19)
- 2006年8月 (14)
- 2006年7月 (13)
- 2006年6月 (11)
- 2006年5月 (17)
- 2006年4月 (9)
- 2006年3月 (5)
- 2006年2月 (20)
- 2006年1月 (9)
- 2005年12月 (23)
- 2005年11月 (35)
- 2005年10月 (13)
- 2005年9月 (10)
- 2005年8月 (5)
- 2005年7月 (18)
- 2004年7月 (9)

文章分类

GoJS、GoDiagram

BoryCHAN

时，这会增加读磁盘的负担，因此，哪个是最优，应该通过实际的使用环境来检验。

在刚开始的时候，如果表不大，没有必要作索引，我的意见是在需要的时候才作索引，也可用一些命令来优化表，例如MySQL可用"OPTIMIZE TABLE"。

综上所述，在如何为数据库建立恰当的索引方面，你应该有一些基本的概念了。

分类: MySQL

好文要顶

关注我

收藏该文

cy163

关注 - 1

粉丝 - 370

+加关注

71

« 上一篇 : [Migrate legacy system into SOA, a wrapping approach](#)
» 下一篇 : [全国高校 名单](#)
<http://www.edu.cn/20040203/3098271.shtml>

posted on 2008-10-27 22:34 cy163 阅读(84383) 评论(3) 编辑 收藏

评论

#1楼

先收藏了

支持(0) 反对(0)

2013-11-17 15:24 | 互联网Fans

#2楼

呵呵，也许你说的就是我吧。我只知道索引原理，就是不会用。

支持(0) 反对(0)

BoryChan-Lucene,Solr,中文分词

2016-11-10 20:55 | AndrewYUE

最新评论

1. Re: left join 和 left outer join 的区别
我是来看评论的

--Godlovezk

2. Re:为mysql数据库建立索引

博主习惯很好，工作了还在坚持写博客，而且很认真，给我们这些初学者帮助很大，谢谢

--xushukui

3. Re:人物关系网络图可视化

1.邀请码在线购买 购买注册邀请码(5.00分) 已有 196 个站点评分2.卡密购买 自动发卡(5.00分) 已有 95 个站点评分3.LcardY积分充值系统 蓝讯积分充值(5.00分) 已有 1.....

--支付接口

4. Re:清远通业(谁有 清远通业 请在跟帖中：(1) 留名 (2)留个电子邮件地址 (3)回答你分了几次红利)

我也有5000股，请问现在还有用吗？有消息告知一下，邮箱810126346@qq.com

--曾生

5. Re: left join 和 left outer join 的区别

这写的鸟不对铨啊

--tangtsl

阅读排行榜

1. left join 和 left outer join 的区别(246540)
2. JDBC连接MySQL (191296)
3. 为mysql数据库建立索引 (84378)
4. UNICODE,GBK,UTF-8 区别(66695)
5. 83款 网络爬虫开源软件 (60558)

评论排行榜

1. 清远通业(谁有 清远通业 请在跟帖中：(1) 留名 (2)留个电子邮件地址 (3)回答你分了几次红利)(83)
2. 测试网络连接速度
<http://www.phoenixtv.com>

#3楼

博主习惯很好，工作了还在坚持写博客，而且很认真，给我们这些初学者帮助很大，谢谢

支持(0) 反对(0)

2017-08-03 18:32 | xushukui

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻：

- 马斯克：AI是人类史上最好的事情，也可能是最坏的
 - 乐视否认电视停产 正在洽谈新代工厂或增至7家
 - 苹果罕见邀请开发者媒体前往总部 推广增强现实平台
 - 谷歌服软了？已向欧盟递交终止垄断行为计划细节
 - ICO乱象调查：定价随意，暴富诱惑已揽入社区大妈
- » [更多新闻...](#)

最新知识库文章：

- 做到这一点，你也可以成为优秀的程序员
 - 写给立志做码农的大学生
 - 架构腐化之谜
 - 学会思考，而不只是编程
 - 编写Shell脚本的最佳实践
- » [更多知识库文章...](#)

3. left join 和 left outer join 的区别(26)
4. 从年轻大学教师待遇说到大学教学质量(18)
5. UNICODE,GBK,UTF-8 区别(17)

推荐排行榜

1. left join 和 left outer join 的区别(20)
2. UNICODE,GBK,UTF-8 区别(16)
3. JDBC连接MySQL (12)
4. 为mysql数据库建立索引(7)
5. 如何调用DLL (基于Visual C++6.0的DLL编程实现)(7)

Powered by:

博客园

Copyright © cy163