

Pvi 推荐序二

第 2 行

原文：“当年加入淘宝的毕业生成了淘宝开发 DBA 的主管，他就是本书的作者”

修改：“当年加入淘宝的毕业生成了淘宝开发 DBA 的主管，而当年加入阿里巴巴 B2B 的毕业生，就是本书的作者”

Pvi 推荐序二

倒数第 3 行

原文：“《搭建 Oracle 高可用环境》一书的作者”

修改：“《构建 Oracle 高可用环境》一书的作者”

P90

第 3 行

原文：“通过调用存储引擎借口来获取”

修改：“通过调用存储引擎接口来获取”

P90

倒数第 3 行

原文：“(假设 id 为 100)”

修改：“(假设 id 为 1)”

P90

倒数第 1 行

原文：“取出前 20 个”

修改：“取出第 100 至 120 个”

P91

第 1 行

原文：

```
and user_group.user_id = user.id  
LIMIT 100,20;
```

修改：

```
and user_group.user_id = user.id  
ORDER BY gmt_create desc  
LIMIT 100,20;
```

也就是在 P91 的第 2 行和第 3 行代码之间插入 1 行：“ORDER BY gmt_create DESC”

P112

最后一行

原文：“query”

修改：“query”

P117

第 13 行

原文：“尽两减少大的复杂 Query”

修改：“尽量减少大的复杂 Query”

P152

倒数第 3 行

原文：“不仅 user_group 表的访问从 ref 变成了 ALL”

修改：“不仅 group_message_content 表的访问从 ref 变成了 ALL”

P167

代码 8-31 替换成如下：

```
sky@localhost : example 03:12:45> EXPLAIN SELECT DISTINCT MAX(user_id)
```

```
-> FROM group_message
```

```
-> WHERE group_id > 1 AND group_id < 10
```

```
-> GROUP BY group_id\G
```

```
***** 1. row *****
```

```
id: 1
```

```
select_type: SIMPLE
```

```
table: group_message
```

```
type: range
```

```
possible_keys: idx_gid_uid_gc
```

```
key: idx_gid_uid_gc
```

```
key_len: 4
```

```
ref: NULL
```

```
rows: 3563
```

```
Extra: Using where; Using index; Using temporary; Using filesort
```

```
1 row in set (0.00 sec)
```

P167

代码段之后

原文：“最后再看一下这个和 GROUP BY 一起使用的带有聚合函数的示例，与上面第三个示例相比，可以看到已经多了 filesort 排序操作了，因为我们使用了 MAX 函数。”

修改：“最后再看一下这个和 GROUP BY 一起使用的带有聚合函数的示例，与上面第三个示例相比，可以看到已经多了 filesort 排序操作了，因为我们使用了 MAX 函数。要取得分组后的 MAX 值，又无法使用索引完成操作，只能通过排序才行了。”

P178

倒数第 3 行

原文：“但是从 5.0.3 开始，VARCHAR 的最大存储限制已经更改为字节数限制了，扩展到可

以存放 65535 bytes 的数据，不同的字符集可能存放的字符数并不一样。也就是说，在 MySQL 5.0.3 之前的版本，M 所代表的是字符数，而从 5.0.3 版本开始，M 代表的是字节数了。”

修改：“但是从 5.0.3 开始，VARCHAR 的最大存储限制已经改为字节数，而且不再有单个字段的限制，而是受单条记录除 TEXT 和 BLOB 类型字段外最大不超过 65536 Bytes 的限制。不过，字段定义中的 M 仍然表示字符数，所以定义后的 VARCHAR 类型字段实际最大可存放数据长度与字符集相关的。”

P199

倒数第 9 行

原文：“最多将缓存 32 个连接线程”

修改：“最多将缓存 64 个连接线程”

P200

代码 10-7 中

原文：“ $(127 - 12) / 127 * 100\%$ ”

修改：“ $(127 - 11) / 127 * 100\%$ ”

P202

正文第 4 行

原文：“平台上可以超出 4BG 的限制”

修改：“平台上可以超出 4GB 的限制”

P207

代码第 1 行

原文：“ $\text{Key_buffer_UsageRatio} = (1 - \text{Key_blocks_used} / (\text{Key_blocks_used} + \text{Key_blocks_unused})) * 100\%$ ”

修改：“ $\text{Key_buffer_UsageRatio} = (\text{Key_blocks_used} / (\text{Key_blocks_used} + \text{Key_blocks_unused})) * 100\%$ ”

P286

正文第 10 行

原文：“Lucene 具肖高效的全文索引和分词算法”

修改：“Lucene 具有高效的全文索引和分词算法”