

## 索引重建的必要性与影响 (Doc ID 1525787.1)

### 文档内容

- [用途](#)
- [适用范围](#)
- [详细信息](#)
- [参考](#)

### 适用于:

Oracle Database Cloud Schema Service - 版本 N/A 和更高版本  
Oracle Database Exadata Express Cloud Service - 版本 N/A 和更高版本  
Oracle Database Exadata Cloud Machine - 版本 N/A 和更高版本  
Oracle Cloud Infrastructure - Database Service - 版本 N/A 和更高版本  
Oracle Database Backup Service - 版本 N/A 和更高版本  
本文档所含信息适用于所有平台

### 用途

本文章将重点概述重建索引的各种影响。我们通常会定期重建索引，但事实上，判断索引重建是否有用一般并不以统计数字为基础，而且很少保留索引的重建历史记录。

### 适用范围

本文章的目标受众是数据库管理员。

### 详细信息

关于重建索引有用与否的讨论有很多。一般而言，极少需要重建 B 树索引，基本原因是 B 树索引很大程度上可以自我管理或自我平衡。

认为需要重建索引的最常见理由有：

- 索引碎片在不断增加
- 索引不断增加，删除的空间没有重复使用
- 索引 clustering factor（群集因子）不同步

事实上，大多数索引都能保持平衡和完整，因为空闲的叶条目可以重复使用。插入/更新和删除操作确实会导致索引块周围的可用空间形成碎片，但是一般来说这些碎片都会被正确的重用。

Clustering factor 群集因子可以反映给定的索引键值所对应的表中的数据排序情况。重建索引不会对群集因子产生影响，要改变集群因子只能通过重组表的数据。

另外，重建索引的影响非常明显，请仔细阅读以下说明：

1. 大多数脚本都依赖 index\_stats 动态表。此表使用以下命令填充：

```
analyze index ... validate structure;
```

尽管这是一种有效的索引检查方法，但是它在分析索引时会获取独占表锁。特别对于大型索引，它的影响会是巨大的，因为在此期间不允许对表执行 DML 操作。虽然该方法可以在不锁表的情况下在线运行，但是可能要消耗额外的时间。

2. 重建索引的直接结果是 REDO 活动可能会增加，总体的系统性能可能会受到影响。

插入/更新/删除操作会导致索引随着索引的分割和增长不断发展。重建索引后，它将连接的更为紧凑；但是，随着对表不断执行 DML 操作，必须再次分割索引，直到索引达到平衡为止。结果，重做活动增加，且索引分割更有可能对性能产生直接影响，因为我们需要将更多的 I/O、CPU 等用于索引重建。经过一段时间后，索引可能会再次遇到“问题”，因此可能会再被标记为重建，从而陷入恶性循环。因此，通常最好是让索引处于自然平衡和（或）至少要防止定期重建索引。

3. 通常是优先考虑 index coalesce（索引合并），而不是重建索引。索引合并有如下优点：

- 不需要占用近磁盘存储空间 2 倍的空间
- 可以在线操作
- 无需重建索引结构，而是尽快地合并索引叶块，这样可避免系统开销过大，请见第 2 点中的解释。

注意：例如，如要将索引转移到其他表空间，则需要重建索引。

综上所述，强烈建议不要定期重建索引，而应使用合适的诊断工具。

请参阅以下文章，其中列出了可用于分析索引结构的脚本。它不使用“analyze index validate structure”命令，但将基于当前表和索引统计信息来估计索引大小。

[Note 989186.1](#) - Script to investigate a b-tree index structure

## 参考

---

[NOTE:989186.1](#) - Script to investigate a b-tree index structure

Didn't find what you are looking for?