

索引的管理和维护

1. 引入索引的目的

引入索引目的就是为了加快查询的速度。oracle 索引是一个独立于表的对象，它可以存放在与表不同的表空间中。

即使索引崩溃，甚至索引被删除都不会影响真正存在的数据的表。

一个索引一旦被建立就由 oracle 系统自动维护，而且由 oracle 系统决定什么时候使用该索引，不用在查询语句中指定使用哪个索引。

但是当个表被删除时所有基于该表的索引都被自动地删除掉。

索引是一种允许直接访问表中某一数据行的树形结构。

索引既可以按索引的逻辑设计分类，也可以按它们的物理实现分类。

索引的数据是按顺利排列的，并且每个索引中包含了一个 ROWID 号码。这个 ROWID 号码记录了数据所在的数据块的位置。

表中每一行也都有 ROWID 号码。18 位号码。

查看一个表中的 ROWID 号码。

```
SELECT A1,ROWID FROM T1;
```

2. 索引的分类

按逻辑分类：

单列索引：基于一列的索引，如在 emp 表中的 ename 列上的索引。

多列索引：也叫组合索引，是基于多列的索引，如在 emp 表中的 job 和 sal 两列上所创建的索引。组合索引的列不一定与表中列的顺序相同，这些列也没有必要相邻。组合索引中的列数最多为 32 列。

唯一索引：保证表中任何数据行的索引列的值都不相同。

非唯一索引：表中不同数据行的索引列的值可以相同。

基于函数的索引：利用表中的一列或者多列使用函数或表达式所创建的索引。基于函数的索引预先计算函数或表达式的值并存在该索引中。基于函数的索引既可以是 B 树索引也可以是位图索引。

物理分类：

分区或非分区索引：非分区索引既可以是 b 树索引，也可以是位图索引。

B 树：包括正常或反转关键字索引。

位图索引：

3. B 树索引

oracle 中所有的索引都是 B 树索引。

当对表进行 DML 操作时，oracle 服务器将自动维护于该表的全部索引。维护方法如下：

当对表进行插入操作时，在对应的索引数据块中插入一行索引项。

当对表进行删除操作时，oracle 服务器仅对索引项进行逻辑删除操作，即仅在所删除的索引项上加一个标记，并不真正地删除该项，而只有等该块中所有的项都被删除后才真正地删除它们。

当对表进行修改操作时，oracle 服务器实际上对索引进行的是两个操作，一个是逻辑删除操作而另一个是插入操作。

4. 位图索引

位图索引也是一种 B 树结构，但是位图索引的叶子节点存的不是 ROWID 而是没一个键值的位图。

5. 创建索引

5.1 创建索引一般遵循如下原则：

平衡查询和 DML 的需要。在 DML 操作频繁的表上尽量减少索引的数量，因为索引虽然加快了查询的速度却降低了 DML 操作的速度。

将索引放入单独的表空间，不要与表、临时段或还原（回滚）段放在一个表空间。因为索引会与这些段竞争输入/输出。

使用统一的 EXTENT 尺寸：数据块尺寸的 5 倍，或表空间的 MINIMUM EXTENT 的尺寸。这样做的目的是为了减少系统的转换时间。

对大索引可以考虑使用 NOLOGGING。

索引的 INITRANS 参数通常应该比相对应表的高。

```
CREATE INDEX IDX_TL ON SCOTT.EMP(ENAME) TABLESPACE TBS3;
```

```
CREATE UNIQUE INDEX IDX2_TL ON SCOTT.EMP(ENAME) TABLESPACE TBS3;
CREATE BITMAP INDEX IDX3_TL ON SCOTT.EMP(JOB) TABLESPACE TBS3;
```

5.2 查询索引的内容

```
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, INDEX_TYPE, UNIQUENESS, STATUS FROM
USER_INDEXES;
SELECT INDEX_NAME, TABLE_NAME, COLUMN_NAME, INDEX_OWNER, TABLE_OWNER FROM
DBA_IND_COLUMNS WHERE TABLE_OWNER='SCOTT';
```

5.3 重建和维护索引

在长时间运行后索引的效率可能会变得越来越差，此时就需要重建正常索引和位图索引。

重建索引，在重建的同时将它的 P C T F R E E 改为 40%，NEXT EXTENT 改为 300k

```
ALTER INDEX IDX_TL REBUILD PCTFREE 40 STORAGE (NEXT 300K);
SELECT INDEX_NAME, PCT_FREE, PCT_INCREASE, INITIAL_EXTENT, NEXT_EXTENT FROM
DBA_INDEXES WHERE OWNER = 'SCOTT';
```

手工为索引增加一个 EXTENT 的磁盘空间

```
ALTER INDEX IDX_TL ALLOCATE EXTENT;
收回索引段中没有用的磁盘空间
ALTER INDEX IDX_TL DEALLOCATE UNUSED;
```

合并索引段中的碎片

```
ALTER INDEX IDX_TL COALESCE;
```

另外阶段表命令也会释放基于该表的索引段所用的全部磁盘空间。

6. 标识索引的使用

查看索引是否使用过：

打开监控索引

```
ALTER INDEX IDX_TL MONITORING USAGE;
```

使用一次索引

```
SELECT * FROM SCOTT.EMP;
```

查看索引的使用情况

```
SQL> SELECT * FROM V$OBJECT_USAGE;
```

INDEX_NAME	TABLE_NAME	MON USE	START_MONITORING
END_MONITORING			
-----	-----	-----	-----
IDX_TL	EMP	YES NO	11/10/2012 06:30:29

取消索引的监控

```
ALTER INDEX IDX_TL NOMONITORING USAGE;
```

7. 删除索引

```
DROP INDEX IDX_TL;
```