**[oracle索引总结](http://www.cnblogs.com/liangyihui/p/5886619.html)**

**简介**

1.说明

　　1）索引是数据库对象之一，用于加快数据的检索，类似于书籍的目录。在数据库中索引可以减少数据库程序查询结果时需要读取的数据量，类似于在书籍中我们利用索引可以不用翻阅整本书即可找到想要的信息。

　　2）索引是建立在表上的可选对象；索引的关键在于通过一组排序后的索引键来取代默认的全表扫描检索方式，从而提高检索效率

　　3）索引在逻辑上和物理上都与相关的表和数据无关，当创建或者删除一个索引时，不会影响基本的表；

　　4）索引一旦建立，在表上进行DML操作时（例如在执行插入、修改或者删除相关操作时），oracle会自动管理索引，索引删除，不会对表产生影响,降低DML操作效率

　　5）索引对用户是透明的，无论表上是否有索引，sql语句的用法不变

　　6）oracle创建主键时会自动在该列上创建索引 主键=唯一索引

**索引原理**

1.  若没有索引，搜索某个记录时（例如查找name='wish'）需要搜索所有的记录，因为不能保证只有一个wish，必须全部搜索一遍

2. 若在name上建立索引，oracle会对全表进行一次搜索，将每条记录的name值升序排列，然后构建索引条目（name和rowid），存储到索引段中，查询name为wish时即可直接查找对应地方

3.创建了索引并不一定就会使用，oracle自动统计表的信息后，决定是否使用索引，表中数据很少时使用全表扫描速度已经很快，没有必要使用索引

**索引使用（创建、修改、删除、查看）**

**1.创建索引语法**

[复制代码](javascript:void(0);)

复制代码

CREATE [UNIQUE] | [BITMAP] INDEX index\_name --unique表示唯一索引

ON table\_name([column1 [ASC|DESC],column2 --bitmap，创建位图索引

[ASC|DESC],…] | [express])

[TABLESPACE tablespace\_name]

[PCTFREE n1] --指定索引在数据块中空闲空间

[STORAGE (INITIAL n2)]

[NOLOGGING] --表示创建和重建索引时允许对表做DML操作，默认情况下不应该使用

[NOLINE]

[NOSORT]; --表示创建索引时不进行排序，默认不适用，如果数据已经是按照该索引顺序排列的可以使用

复制代码

[复制代码](javascript:void(0);)

**2.修改索引**

1）重命名索引

alter index index\_sno rename to bitmap\_index;

2) 合并索引（表使用一段时间后在索引中会产生碎片，此时索引效率会降低，可以选择重建索引或者合并索引,合并索引方式更好些，无需额外存储空间，代价较低）

alter index index\_sno coalesce;

3)重建索引

　　方式一：删除原来的索引，重新建立索引

　　方式二：

alter index index\_sno rebuild;

**3.删除索引**

drop index index\_sno;

**4.查看索引**

[复制代码](javascript:void(0);)

复制代码

select index\_name,index-type, tablespace\_name, uniqueness from all\_indexes where table\_name ='tablename';

-- eg:

create index index\_sno on student('name');

select \* from all\_indexes where table\_name='student';

复制代码

[复制代码](javascript:void(0);)

**索引分类**

**1. B树索引**（默认索引，保存讲过排序过的索引列和对应的rowid值）

1）说明：

　　1.oracle中最常用的索引；B树索引就是一颗二叉树；叶子节点（双向链表）包含索引列和指向表中每个匹配行的ROWID值

　　2.所有叶子节点具有相同的深度，因而不管查询条件怎样，查询速度基本相同

　　3.能够适应精确查询、模糊查询和比较查询

2）分类：

　　 UNIQUE,NON-UNIQUE(默认),REVERSE KEY（数据列中的数据是反向存储的）

3）创建例子

craete index index\_sno on student('sno');

4）适合使用场景：

　　列基数（列不重复值的个数）大时适合使用B数索引

**2. 位图索引**

1）说明：

　1.创建位图索引时，oracle会扫描整张表，并为索引列的每个取值建立一个位图（位图中，对表中每一行 使用一位（bit，0或者1）来标识该行是否包含该位图的索引列的取值，如果为1，表示对应的rowid所在的记录包含该位图索引列值），最后通过位图索 引中的映射函数完成位到行的ROWID的转换

2)创建例子

create bitmap index index\_sno on student(sno);

3) 适合场景：

对于列基数小的列适合简历位图索引（例如性别等）

**3.单列索引和复合索引**（基于多个列创建）

1) 注意：

　　即如果索引建立在多个列上，只有它的第一个列被where子句引用时，优化器才会使用该索引，即至少要包含组合索引的第一列

**4. 函数索引**

1)说明：

　　1. 当经常要访问一些函数或者表达式时，可以将其存储在索引中，这样下次访问时，该值已经计算出来了，可以加快查询速度

　　2. 函数索引既可以使用B数索引，也可以使用位图索引；当函数结果不确定时采用B树索引，结果是固定的某几个值时使用位图索引

　　3. 函数索引中可以水泥用len、trim、substr、upper（每行返回独立结果），不能使用如sum、max、min、avg等

2）例子：

create index fbi on student (upper(name));

select \* from student where upper(name) ='WISH';

**索引建立原则总结**

　　1. 如果有两个或者以上的索引，其中有一个唯一性索引，而其他是非唯一，这种情况下oracle将使用唯一性索引而完全忽略非唯一性索引

　　2. 至少要包含组合索引的第一列（即如果索引建立在多个列上，只有它的第一个列被where子句引用时，优化器才会使用该索引）

　　3. 小表不要建立索引 维度表

　　4. 对于基数大的列适合建立B树索引，对于基数小的列适合建立位图索引

　　5. 列中有很多空值，但经常查询该列上非空记录时应该建立索引

　　6. 经常进行连接查询的列应该创建索引，where条件查询

　　7. 使用create index时要将最常查询的列放在最前面（复合索引）

　　8. LONG（可变长字符串数据，最长2G）和LONG RAW（可变长二进制数据，最长2G）列不能创建索引

　　9.限制表中索引的数量（创建索引耗费时间？，并且随数据量的增大而增大；索引会占用物理空间；当对表中的数据进行增加、删除和修改的时候，索引也要动态的维护，降低了数据的维护速度）

**注意事项**

1. 通配符在搜索词首出现时，oracle不能使用索引，eg：

[复制代码](javascript:void(0);)

复制代码

--我们在name上创建索引；

create index index\_name on student('name');

--下面的方式oracle不适用name索引

select \* from student where name like '%wish%';

--如果通配符出现在字符串的其他位置时，优化器能够利用索引；如下：

select \* from student where name like 'wish%';

复制代码

[复制代码](javascript:void(0);)

 2. 不要在索引列上使用not，可以采用其他方式代替如下：（oracle碰到not会停止使用索引，而采用全表扫描）

[复制代码](javascript:void(0);)

复制代码

select \* from student where not (score=**100);**

**select \* from student where score <> 100;**

**--替换为**

**select \* from student where score>100 or score <100**

复制代码

[复制代码](javascript:void(0);)

3. 索引上使用空值比较将停止使用索引， eg：

select \* from student where score is not null;

4.索引列不能使用函数

5.索引列不能计算