# Oralce索引

1. 索引是数据库对象之一，用于加快数据的检索，类似于书籍的索引。

在数据库中索引可以减少查询时需要读取的数据量

1. 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的方式，从而提高检索效率

1. 索引在逻辑上和物理上都与相关的表和数据无关。当创建或删除一个索引时，不会影响基本的表
2. 索引一旦建立，在表上进行DML操作时（插入，修改，删除），Oracle会自动管理索引，索引删除，不会对表产生影响
3. 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
4. Oracle在创建主键时，会自动在主键列上创建索引

## 索引原理

1. 若没有索引，搜索某个记录时（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值按照升序排序，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的地方
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，使用全表扫描已经很快，没必要使用索引

## 索引分类

##### B树索引

###### 说明

1. 默认索引，存的是排序过的索引列和对应的rowid
2. B树索引就是一颗二叉树。叶子节点（C:\Users\ADMINI~1\AppData\Local\Temp\SGPicFaceTpBq\9688\066C4B34.png双向链表）包含 索引列 和 指向表中每个匹配行的rowid值
3. 所有叶子节点具有相同的深度，所以不管查询条件怎样，查询速度基本相同
4. 能够适应精确查询，模糊查询和比较查询

###### 分类

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

###### 创建例子

Create index index\_sno on student(‘sno’);

**适用场景**

列基数（列不重复值的个数）大时适合使用B树索引

##### 位图索引

###### 说明

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1）来标识该行是否包含该位图的索引列的取值，若为1，表示对应的rowid所在的记录包含该位图索引列值）

###### 创建例子

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

###### 适合场景

对于基数小的列适合建立位图索引（如：性别）

位图索引不适合并发环境，在并发环境下可能会造成大量事务的阻塞

##### 单列索引和符合索引

# 索引使用

## 创建索引语法

Ctreat [UNIQUE | BITMAP] INDEX index\_name on table\_name([column1 [ASC | DESC],column2 [ASC | DESC],…] | [express]) [TABLESPACE tablespace\_name] [pctfree n1] [storage (INITIAL n2)] [NOLOGGING] [NOLINE] [NOSORT]

## 修改索引

#### 重命名索引

Alter index\_sno rename to bitmap\_index;

#### 合并索引

表使用一段时间后在索引中会产生碎片，这时索引效率会降低，可以选择重建索引 或者 合并索引，合并方式会更好些，无需额外存储空间，代价较低

Alert index index\_sno coalesce;

#### 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;

#### 删除索引

Drop index index\_sno;

#### 查看索引

Select index\_name,index\_type,tablespace\_name,uniqueness from all\_indexes where table\_name=’tablename’;