

使用排序哈希簇来加提高查询速度 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/474/2021_2022__E4_BD_BF_E7_94_A8_E6_8E_92_E5_c67_474174.htm 当数据存储在一个普通表中的时候，这些记录将以插入到数据库时的顺序物理地保存到分配的块中。例如，如果有一个用于存储员工信息的表，那么员工姓名将会按照插入到表的顺序存储在表中。如果员工记录非常多的话，那么数据表的响应速度就会逐渐变慢。你可以通过选择值相对等分布的一列（如员工的部门编号）并建立一个簇表来提高查询员工的速度。在簇表中，如果员工属于同一个部门，那么它们的记录将物理地存储在同一个系列的块中。这样就可以提高查找员工信息的速度，这是因为在检索某个特定部门的员工时，需要读取数据库块的数量减少了。而在非簇表中查找员工，就可能需要对每个数据库块进行访问。当表中存在大量键值的时候，你就会开始发现由于存在许多簇块而导致的性能问题。避免这个问题的一个方法就是使用一个哈希函数来约束簇块的数量。哈希函数将会给定一个数值用来限定簇块数量的预计范围，但它得到的值是相对等分布的。例如你可以创建一个哈希函数，只比较部门编号的最后两位。哈希函数中存在的一个问题就是函数值会打乱记录原本的顺序。你可以通过 ORDER BY来解决这个问题；但是，在很多情况下，记录数量是非常庞大的。在Oracle 10g中，你可以将一个数据定义为"natural order"，那么就可以不用经过排序而以你所希望的顺序来检索哈希簇的数据，从而解决了上面的提出问题。例如，假设你有一个信用卡业务的数据库。你决定以信用卡号作为簇主键将有利于

数据的存储分布。但是，由于存在大量的信用卡号，所以可以使用一个哈希函数来约束簇块的数量。而且你希望在你的大部分报表中数据是按照时间顺序排列的，那么在进行每个查询操作时使用排序哈希簇，而不要使用ORDER BY.下面给出了相关语句：

```
create cluster credit_cluster(card_no
varchar2(16),transdate date sort)hashkeys 10000 hash is
ora_hash(card_no)size 256.create table credit_orders(card_no
varchar2(16),transdate date,amount number)cluster
credit_cluster(card_no,transdate).alter session set nls_date_format =
"YYYYMMDDHH24MISS".insert into credit_orders
(card_no,transdate,amount)values
(4111111111111111,20050131000123,57.99).insert into
credit_orders (card_no,transdate,amount)values
(4111111111111111,20050130071216,16.59).insert into
credit_orders (card_no,transdate,amount)values
(4111111111111111,20050131111111,39.00).insert into
credit_orders (card_no,transdate,amount)values
(4111111111111111,20050130081001,25.16).
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com