

综合的考量

- 数据量增加对性能的预估
 - 隐藏在查询背后对数据量的高敏感性
 - 比如max () 对高数据量的敏感，而直接引起子查询性能缓慢降低，必须使用非关联子查询。
- 排序的影响
 - 字节数量而不是记录数量
 - 也就是被排序的总数据量
 - Join应该延后到查询的最后阶段



Join延迟到查询的最后阶段

例子：查询一年内的10大客户的名称和地址

目标，对尽量少的数据进行排序

```
select *
  from (select c.customer_name,
               c.customer_address,
               c.customer_postal_code,
               c.customer_state,
               c.customer_country
          sum(d.amount)
        from customers c,
             orders_o,
             order_detail d
       where c.customer_id = o.customer_id
            and o.order_date >= some date expression
            and o.order_id = d.order_id
      group by c.customer_name,
               c.customer_address,
               c.customer_postal_code,
               c.customer_state,
               c.customer_country
      order by 6 desc) as A
 limit 10
```



```
select c.customer_name,  
       c.customer_address,  
       c.customer_postal_code,  
       c.customer_state,  
       c.customer_country  
       b.amount  
from (select a.customer_id,  
            a.amount  
      from (select o.customer_id,  
                  sum(d.amount) as amount  
            from orders_o,  
                  order_detail d  
            where o.order_date >= some date expression  
                  and o.order_id = d.order_id  
            group by o.customer_id  
            order by 2 desc) as a  
      limit 10) as b,  
      customers c  
where c.customer_id = b.customer_id  
order by b.amount desc
```



消除关联子查询

例子：每小时以批处理的形式更新安全管理表

```
insert /*+ append */ into fast_scrty
( emplid,
  rowsecclass,
  access_cd,
  empl_rcd,
  name,
  last_name_srch,
  setid_dept,
  deptid,
  name_ac,
  per_status,
  scrty_ovrd_type)
select distinct
    emplid,
    rowsecclass,
    access_cd,
    empl_rcd,
    name,
    last_name_srch,
    setid_dept,
    deptid,
    name_ac,
    per_status,
    'N'
from pers_search_fast
```



```

1 select a.emplid,
2         sec.rowsecclass,
3         sec.access_cd,
4         job.empl_rcd,
5         b.name,
6         b.last_name_srch,
7         job.setid_dept,
8         job.deptid,
9         b.name_ac,
10        a.per_status
11 from person a,
12        person_name b,
13        job,
14        scrty_tbl_dept sec
15 where a.emplid = b.emplid
16        and b.emplid = job.emplid
17        and (job. effdt=
18              ( select max(job2. effdt)
19                from job job2
20                where job.emplid = job2.emplid
21                      and job.empl_rcd = job2.empl_rcd
22                      and job2. effdt <=
23                        to_date(to_char(sysdate,
24                                'YYYY-MM-DD'), 'YYYY-MM-DD'))

```

```

25        and job. effseq =
26          ( select max(job3. effseq)
27            from job job3
28            where job.emplid = job3.emplid
29                  and job.empl_rcd = job3.empl_rcd
30                  and job. effdt = job3. effdt ) )
31 and sec.access_cd = 'Y'
32 and exists
33      ( select 'X'
34        from treenode tn
35        where tn.setid = sec.setid
36              and tn.setid = job.setid_dept
37              and tn.tree_name = 'DEPT_SECURITY'
38              and tn. effdt = sec.tree_ effdt
39              and tn.tree_node = job.deptid
40              and tn.tree_node_num between sec.tree_node_num
41                                     and sec.tree_node_num_end
42        and not exists
43          ( select 'X'
44            from scrty_tbl_dept sec2
45            where sec.rowsecclass = sec2.rowsecclass
46                  and sec.setid = sec2.setid
47                  and sec.tree_node_num <> sec2.tree_node_num
48                  and tn.tree_node_num
49                      between sec2.tree_node_num
50                      and sec2.tree_node_num_end
51                  and sec2.tree_node_num
52                      between sec.tree_node_num
53                      and sec.tree_node_num_end ))

```



通过分区提高性能

- 记住，单边范围条件不能充分利用索引和分区
- 所有的更新操作中，删除（delete）最优可能造成麻烦
- 当数据量大到一定程度，不得不进入“读写分离”的数据仓库领域



End

第八个部分就结束了，后面我们讲讲一些关键原则

