2019
05
08-10
北京新云南皇冠假日酒店

数据风云十年变迁

DTCC 第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019

IT168 ChinaUnix ITPUB

SQL改写优化妙手集锦

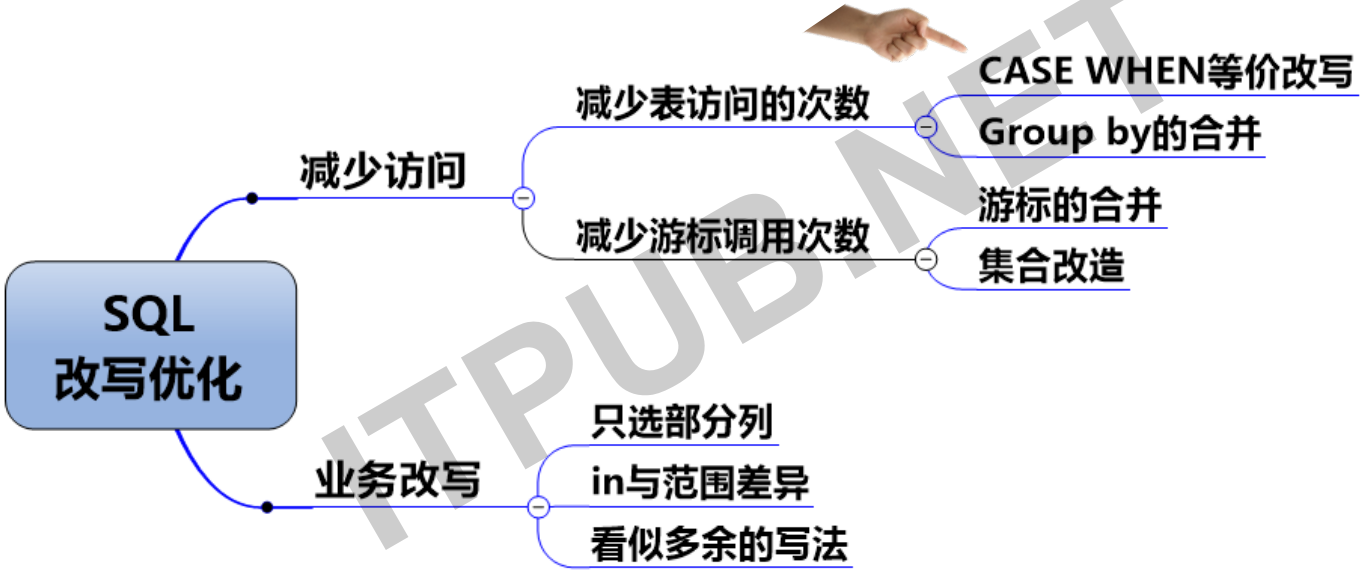
梁敬彬

梁敬彬，福富研究院副理事长、公司四星级内训师、公司特级专家。著有《收获，不止Oracle》、《收获，不止SQL优化》、《收获，不止SQL优化》（第2版）等多本畅销技术书籍。

能听到最后一场的，都是真爱粉，谢谢你们！

最后一讲不占用大家太多时间

结束后，给你们惊喜！


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

```
set autotrace traceonly
set linesize 1000
select t1.object_name,
       t1.object_id,
       (select count(*)
          from t2
         where temporary = 'Y'
           and t2.object_id = t1.object_id) CNT_TEMPORARY_Y,
       (select count(*)
          from t2
         where created >=sysdate-365
           and t2.object_id = t1.object_id) CNT_CREATED_NEW,
       (select sum(object_id)
          from t2
         where status <> 'VALUD'
           and t2.object_id = t1.object_id) SUM_OBJID_STATUS_V,
       (select sum(object_id)
          from t2
         where generated = 'Y'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_Y,
       (select sum(object_id)
          from t2
         where generated = 'M'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_M,
       (select sum(object_id)
          from t2
         where generated = 'Q'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_Q
  from t1
 where t1.object_id <= 50;
```

优化前

```
Execution Plan
----------------------------------------------------------
Plan hash value: 2340670826

-------------------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      |    12 |   948 |   291   (1)| 00:00:04 |
|   1 |  SORT AGGREGATE     |      |     1 |    15 |            |          |
| * 2 |   TABLE ACCESS FULL | T2   |     1 |    15 |   292   (2)| 00:00:04 |
|   3 |  SORT AGGREGATE     |      |     1 |    22 |            |          |
| * 4 |   TABLE ACCESS FULL | T2   |    10 |   220 |   291   (1)| 00:00:04 |
|   5 |  SORT AGGREGATE     |      |     1 |    18 |            |          |
| * 6 |   TABLE ACCESS FULL | T2   |   761 | 13698 |   292   (2)| 00:00:04 |
|   7 |  SORT AGGREGATE     |      |     1 |    15 |            |          |
| * 8 |   TABLE ACCESS FULL | T2   |   125 |  1875 |   292   (2)| 00:00:04 |
|   9 |  SORT AGGREGATE     |      |     1 |    15 |            |          |
| * 10|   TABLE ACCESS FULL | T2   |     1 |    15 |   292   (2)| 00:00:04 |
|  11 |  SORT AGGREGATE     |      |     1 |    15 |            |          |
| * 12|   TABLE ACCESS FULL | T2   |     1 |    15 |   292   (2)| 00:00:04 |
| * 13|  TABLE ACCESS FULL  | T1   |    12 |   948 |   291   (1)| 00:00:04 |
-------------------------------------------------------------------------------------
```

优化后

```
with w_t2 as
(select
       t2.object_id,
       count(case when t2.temporary='Y' then 1 end ) CNT_TEMPORARY_Y,
       count(case when created >=sysdate-365  then 1 end )
CNT_CREATED_NEW,
       sum(case when t2.status<>'VALID' then t2.object_id end )
SUM_OBJID_STATUS_V,
       sum(case when t2.generated = 'Y' then t2.object_id end )
SUM_OBJID_GENERATED_Y,
       sum(case when t2.generated = 'M' then t2.object_id end )
SUM_OBJID_GENERATED_M,
       sum(case when t2.generated = 'Q' then t2.object_id end )
SUM_OBJID_GENERATED_Q
 from  t2
 group by t2.object_id)
 select * from w_t2,t1
where t1.object_id=w_t2.object_id
and t1.object_id<=50;
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 3226881135

-------------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      |    11 |  2750 |    583   (1)| 00:00:07 |
|   1 |  HASH GROUP BY      |      |    11 |  2750 |    583   (1)| 00:00:07 |
| * 2 |   HASH JOIN         |      |    11 |  2750 |    582   (1)| 00:00:07 |
| * 3 |    TABLE ACCESS FULL| T1   |    12 |  2628 |    291   (1)| 00:00:04 |
| * 4 |    TABLE ACCESS FULL| T2   |    12 |   372 |    291   (1)| 00:00:04 |
-------------------------------------------------------------------------------
```

```
set autotrace traceonly
set linesize 1000
select t1.object_name,
       t1.object_id,
       (select count(*)
          from t2
         where temporary = 'Y'
           and t2.object_id = t1.object_id) CNT_TEMPORARY_Y,
       (select count(*)
          from t2
         where created >=sysdate-365
           and t2.object_id = t1.object_id) CNT_CREATED_NEW,
       (select sum(object_id)
          from t2
         where status <> 'VALUD'
           and t2.object_id = t1.object_id) SUM_OBJID_STATUS_V,
       (select sum(object_id)
          from t2
         where generated = 'Y'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_Y,
       (select sum(object_id)
          from t2
         where generated = 'M'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_M,
       (select sum(object_id)
          from t2
         where generated = 'Q'
           and t2.object_id = t1.object_id)  SUM_OBJID_GENERATED_Q
  from t1
 where t1.object_id <= 50;
```

```
with w_t2 as
(select
       t2.object_id,
       count(case when t2.temporary='Y' then 1 end ) CNT_TEMPORARY_Y,
       count(case when created >=sysdate-365  then 1 end )
CNT_CREATED_NEW,
       sum(case when t2.status<>'VALID' then t2.object_id end )
SUM_OBJID_STATUS_V,
       sum(case when t2.generated = 'Y' then t2.object_id end )
SUM_OBJID_GENERATED_Y,
       sum(case when t2.generated = 'M' then t2.object_id end )
SUM_OBJID_GENERATED_M,
       sum(case when t2.generated = 'Q' then t2.object_id end )
SUM_OBJID_GENERATED_Q
 from  t2
 group by t2.object_id)
 select * from w_t2,t1
where t1.object_id=w_t2.object_id
and t1.object_id<=50;
```

代码差
异比对

优化效果比对

Execution Plan

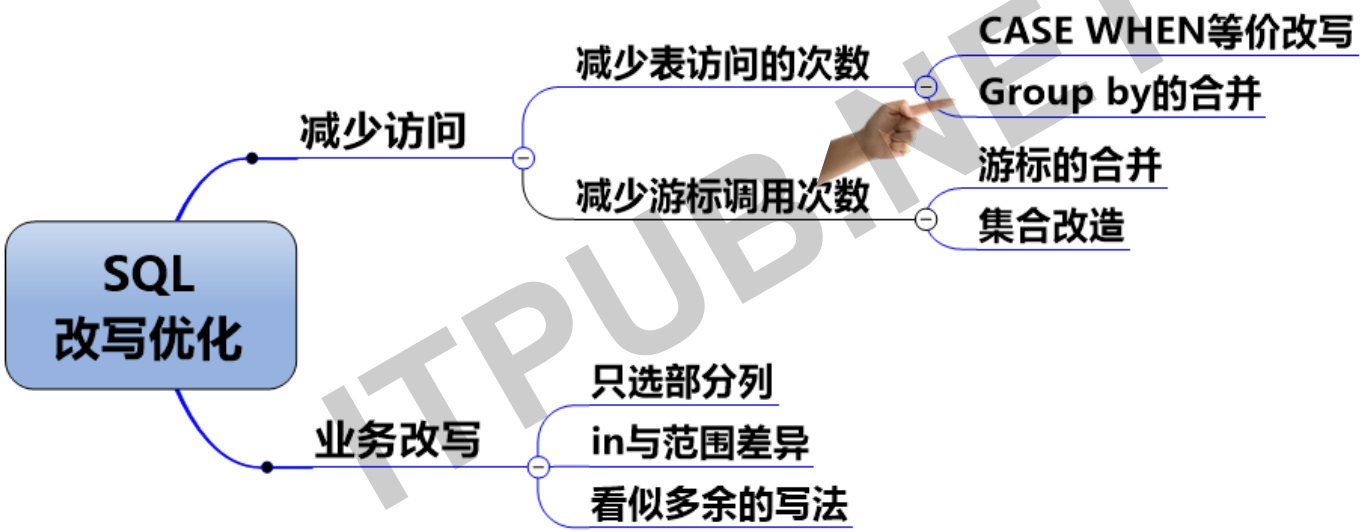
Plan hash value: 3226881135

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		11	2750	583 (1)	00:00:07
1	HASH GROUP BY		11	2750	583 (1)	00:00:07
* 2	HASH JOIN		11	2750	582 (1)	00:00:07
* 3	TABLE ACCESS FULL	T1	12	2628	291 (1)	00:00:04
* 4	TABLE ACCESS FULL	T2	12	372	291 (1)	00:00:04

Execution Plan

Plan hash value: 2340670826

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		12	948	291 (1)	00:00:04
1	SORT AGGREGATE		1	15		
* 2	TABLE ACCESS FULL	T2	1	15	292 (2)	00:00:04
3	SORT AGGREGATE		1	22		
* 4	TABLE ACCESS FULL	T2	10	220	291 (1)	00:00:04
5	SORT AGGREGATE		1	18		
* 6	TABLE ACCESS FULL	T2	761	13698	292 (2)	00:00:04
7	SORT AGGREGATE		1	15		
* 8	TABLE ACCESS FULL	T2	125	1875	292 (2)	00:00:04
9	SORT AGGREGATE		1	15		
* 10	TABLE ACCESS FULL	T2	1	15	292 (2)	00:00:04
11	SORT AGGREGATE		1	15		
* 12	TABLE ACCESS FULL	T2	1	15	292 (2)	00:00:04
* 13	TABLE ACCESS FULL	T1	12	948	291 (1)	00:00:04


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

优化前

```
1 select  substr(object_id,1,2),count(*)  from t where
  object_type='INDEX'  group by substr(object_id,1,2)
2 union
3 select  substr(object_id,1,3),count(*)  from t  where
  object_type<>'INDEX' group by substr(object_id,1,3) ;
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 2983795838

-------------------------------------------------------------------------------------
| Id  | Operation           | Name | Rows  | Bytes | TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |      | 74786 |  1752K|        |  1098   (74)| 00:00:14 |
|   1 |  SORT UNIQUE        |      | 74786 |  1752K|  2264K |  1098   (74)| 00:00:14 |
|   2 |   UNION-ALL         |      |       |       |        |             |          |
|   3 |    HASH GROUP BY    |      |  2908 | 69792 |        |   293    (2)| 00:00:04 |
| * 4 |     TABLE ACCESS FULL| T    |  2908 | 69792 |        |   291    (1)| 00:00:04 |
|   5 |    HASH GROUP BY    |      | 71878 |  1684K|  2264K |   805    (2)| 00:00:10 |
| * 6 |     TABLE ACCESS FULL| T    | 71878 |  1684K|        |   291    (1)| 00:00:04 |
-------------------------------------------------------------------------------------
```

优化后

```
1 select substr(object_id,1,CASE WHEN object_type = 'INDEX' then 2 ELSE
  3  end),count(*) from t
2  group by substr(object_id,1,CASE WHEN object_type = 'INDEX' then 2
  ELSE 3  end)  ;
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 47235625

-------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      | 74786 |  1752K|   296   (3)| 00:00:04 |
|   1 |  HASH GROUP BY     |      | 74786 |  1752K|   296   (3)| 00:00:04 |
|   2 |   TABLE ACCESS FULL| T    | 74786 |  1752K|   291   (1)| 00:00:04 |
-------------------------------------------------------------------------------
```

```
select substr(object_id,1,CASE WHEN object_type = 'INDEX' then 2 ELSE
3  end),count(*) from t
 group by substr(object_id,1,CASE WHEN object_type = 'INDEX' then 2
ELSE 3  end)  ;
```

```
select  substr(object_id,1,2),count(*)  from t where
object_type='INDEX'  group by substr(object_id,1,2)
union
select  substr(object_id,1,3),count(*)  from t  where
object_type<>'INDEX' group by substr(object_id,1,3) ;
```

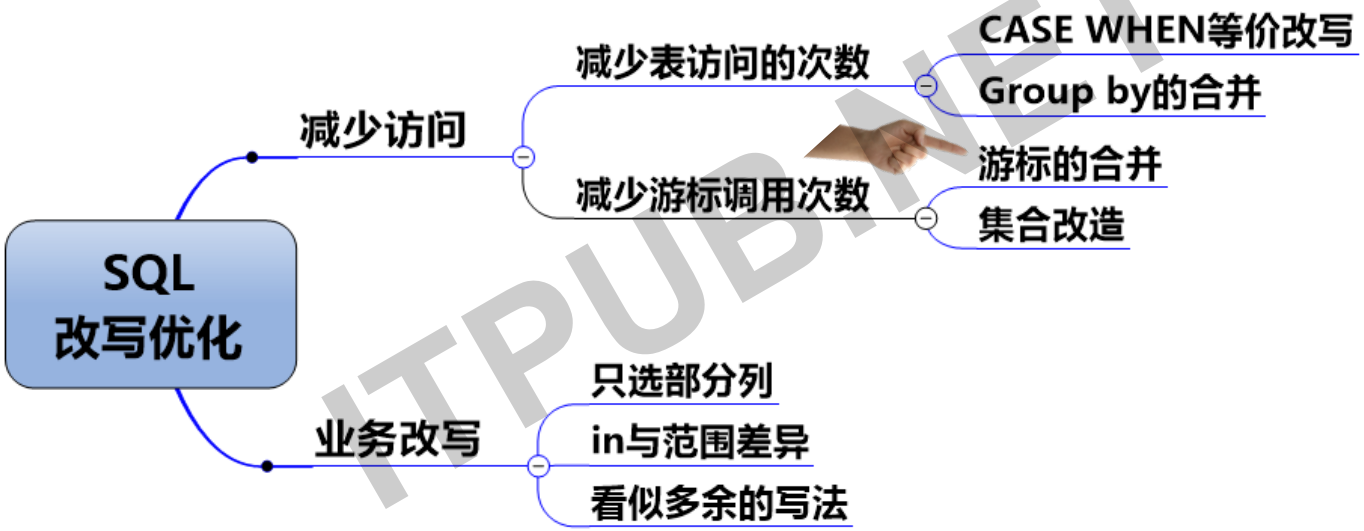

优化效果比对

```
Execution Plan
----------------------------------------------------------
Plan hash value: 47235625

-------------------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      | 74786 | 1752K |   296   (3)| 00:00:04 |
|   1 |  HASH GROUP BY     |      | 74786 | 1752K |   296   (3)| 00:00:04 |
|   2 |   TABLE ACCESS FULL| T    | 74786 | 1752K |   291   (1)| 00:00:04 |
-------------------------------------------------------------------------------------------
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 2983795838

-------------------------------------------------------------------------------------------------
| Id  | Operation            | Name | Rows  | Bytes | TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT     |      | 74786 |  1752K|        |  1098  (74)| 00:00:14 |
|   1 |  SORT UNIQUE         |      | 74786 |  1752K|  2264K |  1098  (74)| 00:00:14 |
|   2 |   UNION-ALL          |      |       |       |        |            |          |
|   3 |    HASH GROUP BY     |      |  2908 | 69792 |        |   293   (2)| 00:00:04 |
| * 4 |     TABLE ACCESS FULL| T    |  2908 | 69792 |        |   291   (1)| 00:00:04 |
|   5 |    HASH GROUP BY     |      | 71878 |  1684K|  2264K |   805   (2)| 00:00:10 |
| * 6 |     TABLE ACCESS FULL| T    | 71878 |  1684K|        |   291   (1)| 00:00:04 |
-------------------------------------------------------------------------------------------------
```

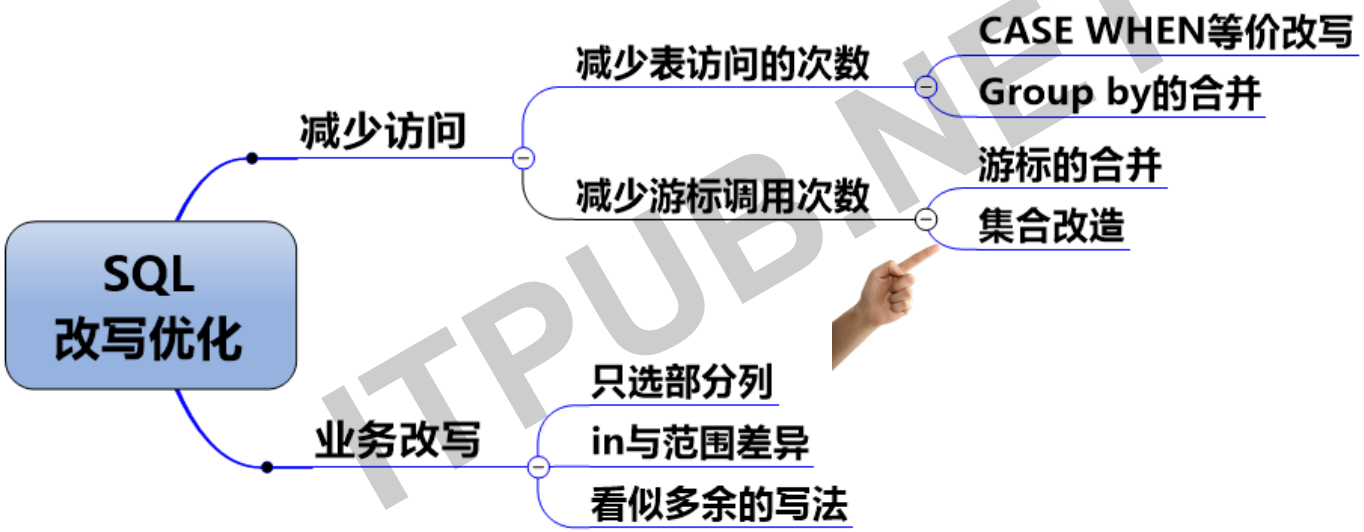

SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

```
set timing on
begin
    for i in 1 .. 100000
    loop
        for a in ( select t1.a, t1.y
                     from t1 where t1.a = i )
        loop
            for b in ( select t2.b, t2.a, t2.y
                         from t2 where t2.a = a.a )
            loop
                for c in ( select t3.c, t3.b, t3.y
                             from t3 where t3.b = b.b )
                loop
                    null;
                end loop;
            end loop;
        end loop;
    end loop;
end;
/
```

优化前

```
set timing on
begin
    for i in 1 .. 100000
    loop
        for x in ( select t1.a t1a, t1.y t1y,
                          t2.b t2b, t2.a t2a, t2.y t2y,
                          t3.c t3c, t3.b t3b, t3.y t3y
                     from t1, t2, t3
                    where t1.a = i
                      and t2.a (+) = t1.a
                      and t3.b (+) = t2.b )
        loop
            null;
        end loop;
    end loop;
end;
/
```

优化后


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

2.减少游标调用

案例二

```
/*--需求描述
--1 确定部门里的员工弄的平均工资
--2. 如果员工的工资与平均工资差别在20%以上，在
EMP_SAL_LOG表里新增一行，并且含偏差值
--3. 如果工资在部门里最低，则在EMP_SAL_LOG表里标志出来。
*/
```

2.减少游标调用 案例二

```
create or replace
 procedure report_sal_adjustment1 is
 v_avg_dept_sal emp.sal%type;
 v_min_dept_sal emp.sal%type;
 v_dname dept.dname%type;
 cursor c_emp_list is
 select empno, ename, deptno, sal, hiredate
 from emp;
begin
 for each_emp in c_emp_list loop
 select avg(sal)
 into v_avg_dept_sal
 from emp
 where deptno = each_emp.deptno;
 if abs(each_emp.sal - v_avg_dept_sal ) / v_avg_dept_sal > 0.20 then
 select dept.dname, min(emp.sal)
 into v_dname, v_min_dept_sal
 from dept, emp
 where dept.deptno = each_emp.deptno
 and emp.deptno = dept.deptno
 group by dname;
 if v_min_dept_sal = each_emp.sal then
 insert into emp_sal_log
 values ( each_emp.ename, each_emp.hiredate,
 each_emp.sal, v_dname, 'Y');
 else
 insert into emp_sal_log
 values ( each_emp.ename, each_emp.hiredate,
 each_emp.sal, v_dname, 'N');
 end if;
 end if;
end loop;
end report_sal_adjustment1;
```

优化前

```
create or replace
 procedure report_sal_adjustment4 is
begin
 insert into emp_sal_log
 select e.empno, e.hiredate, e.sal, dept.dname,
 case when sal > min_sal then 'Y'
 else 'N'
 end case
 from (
 select empno, hiredate, sal, deptno,
 avg(sal) over ( partition by deptno ) as avg_sal,
 min(sal) over ( partition by deptno ) as min_sal
 from emp ) e, dept
 where e.deptno = dept.deptno
 and abs(e.sal - e.avg_sal)/e.avg_sal > 0.20;
 end report_sal_adjustment4;
/
```

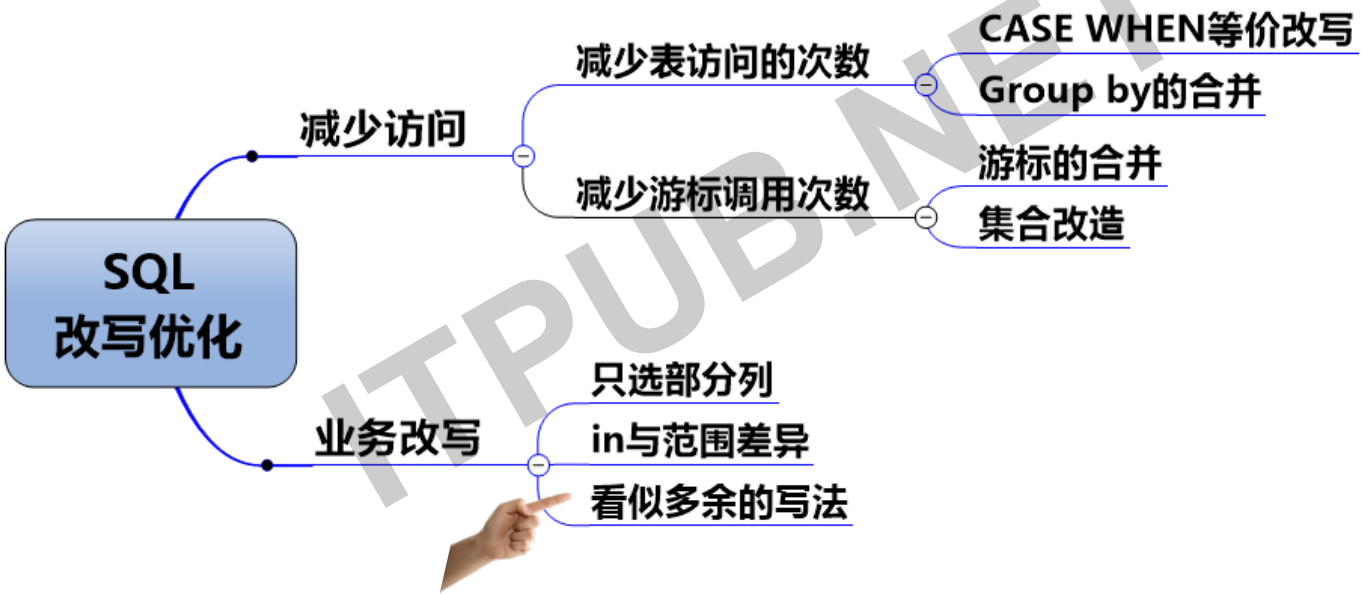
优化后


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

```
SQL> select * from v_t1_join_t2;
-----------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |  8870 |  1039K|   332   (1)| 00:00:04 |
|*  1 |  HASH JOIN         |      |  8870 |  1039K|   332   (1)| 00:00:04 |
|   2 |   TABLE ACCESS FULL| T2   |  8870 |   684K|    40   (0)| 00:00:01 |
|   3 |   TABLE ACCESS FULL| T1   | 69551 |  2784K|   291   (1)| 00:00:04 |
-----------------------------------------------------------------------------
SQL> select object_id,object_name from v_t1_join_t2;
-----------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |  8870 |   684K|    40   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | T2   |  8870 |   684K|    40   (0)| 00:00:01 |
-----------------------------------------------------------------------------
```


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

```
select  /*+index(t,idx_object_id)*/ * from t  where object_TYPE='TABLE'  AND OBJECT_ID >= 20 AND OBJECT_ID<= 21;
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name          | Starts | E-Rows | A-Rows |   A-Time    | Buffers |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |               |      1 |        |   2925 |00:00:00.03 |    1103 |
|   1 |  TABLE ACCESS BY INDEX ROWID| T             |      1 |   2126 |   2925 |00:00:00.03 |    1103 |
|*  2 |   INDEX RANGE SCAN           | IDX_OBJECT_ID |      1 |    320 |   2925 |00:00:00.02 |     730 |
-------------------------------------------------------------------------------------------------------------
select  /*+index(t,idx_object_id)*/ * from t t where object_TYPE='TABLE'  AND  OBJECT_ID IN (20,21);
-------------------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name          | Starts | E-Rows | A-Rows |   A-Time    | Buffers |
-------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |               |      1 |        |   2925 |00:00:00.01 |     588 |
|   1 |  INLIST ITERATOR             |               |      1 |        |   2925 |00:00:00.01 |     588 |
|   2 |   TABLE ACCESS BY INDEX ROWID| T             |      2 |   2126 |   2925 |00:00:00.01 |     588 |
|*  3 |    INDEX RANGE SCAN          | IDX_OBJECT_ID |      2 |      1 |   2925 |00:00:00.01 |     215 |
-------------------------------------------------------------------------------------------------------------
```


SQL
改写优化
减少访问
减少表访问的次数
CASE WHEN等价改写
Group by的合并
减少游标调用次数
游标的合并
集合改造
业务改写
只选部分列
in与范围差异
看似多余的写法

梁敬彬
福建 福州

SQL优化的知识非常通用，而且远不止着一些，惊喜在此。

仅限当场扫我微信的人获取==》

扫一扫上面的二维码图案，加我微信