第二章

第2章 SQL 执行

2.5 SGA-缓冲区缓存

eg:

```
alter system set events 'immediate trace name flush_cache';
alter system flush shared_pool;
set autotrace traceonly statistics
select * from employees where department_id = 60;
set autotrace off
alter system set events 'immediate trace name flush_cache';
set autotrace traceonly statistics
select * from employees where department_id = 60;
set autotrace off
```

2.6 查询转换

eg：

```
select * from employees where department_id in (select department_id from departments)
```

or

```
select e.* from employees e, departments d where e.department_id = d.department_id
```

2.7 试图合并

eg:

```
select * from orders o, (select sales_rep_id from orders) o_view where o.sales_rep_id = o_view.sales_rep_id(+) and o.order_total > 100000;
```

or

```
select * from orders o, (select /*+ no_merge */ sales_rep_id from orders) o_view where o.sales_rep_id = o_view.sales_rep_id(+) and o.order_total > 100000;
```

Note: 除了加 no_merge hint, 还有一些情况也会组织视图合并：

视图查询块包含解析函数，聚合函数，集合运算，order by 子句，或者使用了rownum。

不过此时你可以通过使用 merge hint 来强制执行视图合并。

eg:

```
select e1.last_name, e1.salary, v.avg_salary from employees e1, (select department_id, avg(salary) avg_salary from employees e2 group by department_id) v where e1.department_id = v.department_id and e1.salary > v.avg_salary;
```

or

```
select e1.last_name, e1.salary, v.avg_salary from employees e1, (select /*+ merge */ department_id, avg(salary) avg_salary from employees e2 group by department_id) v where e1.department_id = v.department_id and e1.salary > v.avg_salary;
```

视图合并行为由一个隐藏参数 _complex_view_merging 控制。

2.8 子查询解嵌套

当子查询 位于 where 子句时，可能发生子查询解嵌套。

eg 见 2.6

使用 no_unnest hint

```
select employee_id, last_name, salary, department_id from employees where department_id in (select /*+ no_unnest */ department_id from departments where location_id > 1700);
```

eg: 联结子查询的解嵌套转换

```
select outer.employee_id, outer.last_name, outer.salary, outer.department_id
from employees outer where outer.salary >
(select avg(inner.salary) from employees inner where inner.department_id = outer.department_id);
```

or

```
select outer.employee_id, outer.last_name, outer.salary, outer.department_id from employees outer,
(select department_id, avg(salary) avg_sal  from employees group by department_id) inner
where outer.department_id = inner.department_id
```

2.9 谓语前推

eg:

```
set autotrace traceonly explain
select e1.last_name, e1.salary, v.avg_salary from employees e1,
(select department_id, avg(salary) avg_salary from employees e2 group by department_id) v
where e1.department_id = v.department_id
and e1.salary > v.avg_salary and e1.department_id = 60;
or
select e1.last_name, e1.salary, v.avg_salary from employees e1,
(select department_id, avg(salary) avg_salary from employees e2
where rownum > 1      -- rownum prohibits predicate pushing!
group by department_id) v
where e1.department_id = v.department_id
and e1.salary > v.avg_salary and e1.department_id = 60;
```

Note: 使用rownum 不仅会禁止谓语前推，也会禁止视图合并。

2.10 使用物化视图进行查询重写

eg:

```
set autotrace traceonly explain
select p.prod_id, p.prod_name, t.time_id, t.week_ending_day, s.channel_id, s.promp_id, s.cust_id, s.amount_sold from sales s, products p, times t
where s.time_id = t.time_id and s.prod_id = p.prod_id;
set autotrace off
create materialized view sales_time_product_mv
enable query rewrite as
select p.prod_id, p.prod_name, t.time_id, t.week_ending_day, s.channel_id, s.promp_id, s.cust_id, s.amount_sold from sales s, products p, times t
where s.time_id = t.time_id and s.prod_id = p.prod_id;
set autotrace traceonly explain
select p.prod_id, p.prod_name, t.time_id, t.week_ending_day, s.channel_id, s.promp_id, s.cust_id, s.amount_sold from sales s, products p, times t
where s.time_id = t.time_id and s.prod_id = p.prod_id;
select /*+ rewrite(sales_time_product_mv) */
p.prod_id, p.prod_name, t.time_id, t.week_ending_day, s.channel_id, s.promp_id, s.cust_id, s.amount_sold from sales s, products p, times t
where s.time_id = t.time_id and s.prod_id = p.prod_id;
```

2.11 确定执行计划

表的统计信息可以从 dba_tables 中得到

列的统计信息可以从 dba_tab_cols 中得到

索引的统计信息可以从 dba_indexes 中得到

2.12 执行计划并取得数据行

SQL 执行步骤

解析 --> 绑定 --> 执行 --> 提取 --> 提取

eg: 列大小设置是如何影响逻辑读取的

```
set arraysize 15
set autotrace traceonly statistics
select * from order_items;
set arraysize 45
/
```

Note: 可见 arraysize 可以影响逻辑读的次数和网络往返的次数。

关于 arraysize 和 fetch size 对性能优化的影响可参考博客 :

https://blog.csdn.net/tianlesoftware/article/details/6579913

https://blog.csdn.net/swordmanwk/article/details/6263097