第十一章

半联结和反联结

11.1 半联结

eg: in 半联结的例子

```
select /*+ using in */ department_name
from hr.departments dept
where department_id in (select department_id from hr.employees emp);
```

eg: exists 半联结

```
select /*+ using exists */ department_name
from hr.departments dept
where exists (select null from hr.employees emp where emp.department_id = dept.department_id);
```

eg: exists 和 in 的可替换语法 —— inner join

```
select /*+ inner join */ department_name
from hr.departments dept, hr.employees emp
where dept.department_id = emp.department_id;
```

eg: exists 和 In 的可替换语法 —— 具有 distinct 的 inner join

```
select /*+ inner join with distinct */ distinct department_name
from hr.departments dept, hr.employees emp
where dept.department_id = emp.department_id;
```

eg: exists 和 in 的可替换语法 —— 丑陋的交集

```
select /*+ uply intersect */ department_name
from hr.departments dept,
    (select department_id from hr.departments
    intersect
    select department_id from hr.employees) b
where b.department_id = dept.department_id;
```

eg: exists 和 in 的可替换语法 —— =any 子查询

```
select /*+ any subquery */ department_name
from hr.departments dept
where department_id = any (select department_id from hr.employees emp);
```

eg: semi-join 和 distinct 是不同的

```
select /* semi using in */ department_id
from hr.employees
where department_id in (select department_id from hr.departments);
select /* inner join with distinct */ distinct emp.department_id
from hr.departments dept, hr.employees emp
where dept.department_id = emp.department_id;
```

Note: 使用 exists 语法要确定子查询与外层查询是相关的。

eg: 使用exists 的常见错误 —— 相关与不相关子查询

```
select /* correlated */ department_id
from hr.departments dept
where exists (select department_id from hr.employees emp
            where emp.department_id = dept.department_id);
select /* not correlated */ department_id
from hr.departments dept
where exists (select department_id from hr.employees emp);
select /* not correlated no nulls */ department_id
from hr.departments dept
where exists (select department_id from hr.employees emp were department_id is not null);
select /* non-correlated totally unrelated */ department_id
from hr.departments dept
where exists (select null from dual);
select /* non-correleated empty subquery */ department_id
from hr.departments dept
```

```
where exists (select 'anything' from dual where 1=2);
```

11.2 半联结执行计划

eg: 半联结执行计划

```
select /* in */ department_name
from hr.departments dept
where department_id in (select department_id from hr.employees emp);
select /* exists */ department_name
from hr.departments dept
where exists (select null from hr.employees emp where emp.department_id = dept.department_id);
```

Note:

```
alter session set events '10053 trace name context forever, level 1';
```

设定这个事件将会在迎接西时贼USER_DUMP_DEST文件夹中建立一个追踪文件。

11.3 控制半联结执行计划

11.3.1 使用提示控制半联结执行计划

semijoin
no_semijoin
nl_sj
hash_sj
merge_sj

eg: 使用 no_semijoin 提示的 exists 语句

```
set autotrace trace
select /* exists no_semijoin */ department_name
from hr.departments dept
where exists (select /*+ no_semijoin */ null from hr.employees emp
            where emp.department_id = dept.department_id);
```

Note: explain plan 语句有时候得出的计划于优化器不同，故不推荐。但有时候它提动力一些dbms_xplan无法获得的附加信息。

eg: 使用 no_semijoin 提示的 exists 语句

```
set echo on
select sql_id, sql_text from v$sqlarea
where sql_text like 'select /* exists no_semijoin */ %';
@plan
set lines 150
select * from table(dbms_xplan.display_cursor('SQL_ID',null,'typical'));
```

11.3.2 在实例级控制半联结执行计划

eg: _always_semi_join 的有效值

```
select name_kspvld_values name, value_kspvld_values value
from x$kspvld_values
where name_kspvld_values like nvl('&name',name_ksplvd_values);
```

eg: 使用 _always_semi_join 将执行计划改变为Merge半联结

```
select /* using in */ department_name
from hr.departments dept
where department_id in (select department_id from hr.employees emp);
alter session set "_always_semi_join"=MERGE'
select /* using in */ department_name
from hr.departments dept
where department_id in (select department_id from hr.employees emp);
```

11.5 半联结必要条件

语句必须使用关键字 in(=any) 或 exists
语句必须在 in 或 exists 子句中有子查询
如果使用exists语法，则必须使用相关子查询
in 和 exists 子句不能包含在 or 分支中

11.6 反联结

eg: not in 和 not exists 例子

```
select /* not in */ department_name
from hr.departments dept
where department_id not in
(select department_id from hr.employees emp);
select /* not exists */ department_name
from hr.departments dept
where not exists (select null from hr.employees emp where emp.department_id = dept.department_id);
```

Note: 在上面两个例子中 not in 和 Not exists 并没有返回同样的数据，功能上不是等价的。

原因在于查询是如何处理子查询返回的空值的。

eg: 避免 not in 中的空值

eg: 法1 nvl函数

```
select /* in with nvl */ department_name
from hr.departments dept
where department_id not in
(select nvl(department_id,-10) from hr.employees emp);
```

eg: 法2 is not null

```
select /* in with not null */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp
                            where department_id is not null);
```

eg: not in 和 not exists 的替代语法

```
select /* minus */ department_name
from hr.departments
where department_id in
    (select department_id from hr.departments
    minus
    select department_id from hr.employees);
select /* left outer */ department_name
from hr.departments dept left outer join
hr.employees emp on dept.department_id = emp.department_id
where emp.department_id is null;
select /* left outer old (+) */ department_name
from hr.departments dept, hr.employees emp
where dept.department_id = emp.department_id(+)
and emp.department_id is null;
```

11.7 反联结执行计划

```
select /* not in */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp);
```

Note: not in 语句生成了合并反联结（merge join anti na）执行计划

```
select /* not exists */ department_name
from hr.departments dept
where not exists (select null from hr.employees emp where emp.department_id = dept.department_id);
```

Note: not exists 语句生成了嵌套循环反联结（nested loops anti）执行计划

eg: 反联结执行计划

```
set echo on
@flush_pool
alter system flush shared_pool;
@anti_ex2
set echo on
```

eg:

```
select /* in */ department_name
from hr.departments dept
where department_id not in
(select department_id from hr.employees emp);
```

eg:

```
select /*in with nvl */ department_name
from hr.departments dept
where department_id not in
(select nvl(department_id,-10) from hr.employees emp);
```

eg:

```
select /* in with not null */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp
                          where department_id is not null);
```

eg:

```
select /* exists */ department_name
from hr.departments dept
where not exists (select null from hr.employees emp
                wher emp.department_id = dept.department_id;
set echo off
set echo on
@fsp
```

eg:

```
select distinct s.sql_id, s.child_number, s.plan_hash_value plan_hash, sql_text,
case when options like '%SEMI%’ or options like '%ANTI%’ then
operation||' '||options end join
from v$sql s, v$sql_plan p
where s.sql_id = p.sql_id
and s.child_number = p.child_number
and upper(sql_text) like upper(nvl('&sql_text','%department%'))
and sql_text not like '%from v$sql where sql_text like nvl(%'
and s.sql_id like nvl('&sql_id', s.sql_id)
order by 1,2,3
/
```

eg: 可替代的反联结语法执行计划

```
set echo on
@flush_pool
alter system flush shared_pool;
@anti_ex3
set echo on
select /* not exists */ department_name
from hr.departments dept
where not exists (select null from hr.employees emp
                where emp.department_id = dept.department_id);
```

eg:

```
select /* not in not null */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp
                        where department_id is not null);
```

eg:

```
select /* left outer */ department_name
from hr.departments dept left outer join
    hr.employees emp on dept.department_id = emp.department_id
where emp.department_id is null;
```

eg:

```
select /* left outer old (+) */ department_name
from hr.departments dept, hr.employees emp
where dept.department_id = emp.department_id(+)
and emp.department_id is null;
```

eg:

```
select /* minus */ department_name
```

```
from hr.departments
where department_id in
    (select department_id from hr.departments
    minus
    select department_id from hr.employees);
```

eg:

```
set echo off
@fsp
```

11.8 控制反联结执行计划

11.8.1 使用提示控制反联结执行计划

antijoin

use_anti

nl_aj

hash_aj

merge_aj

eg:

```
set autotrace traceonly exp
@anti)ex4
select /* in */ department_name
from hr.departments dept
where department_id not in (select /*+ nl_aj */ department_id fron hr.employees emp);
```

eg:

```
select /* exists */ department_name
from hr.departments dept
where not exists (select /*+ nl_aj*/ null from hr.employees emp where emp.department_id = dept.department_id);
```

11.8.2 在实例级控制反联结执行计划

一些隐藏参数会印象优化器对反联结的选择

_always_anti_

_gs_anti_semi_join_allowed

_optimizer_null_aware_antijoin

_optimizer_outer_to_anti_enabled

eg: 通过参数控制反联结执行计划

```
select /* exists */ department_name
from hr.departments dept
where not exists (select null from hr.employees emp where emp.department_id = dept.department_id);
```

eg:

```
select /* exists with hint */ department_name
from hr.departments dept
where not exists (select /*+ hash_aj */ null from hr.employees emp where emp.department_id = dept.department_id);
```

eg:

```
select /* in */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp);
```

eg:

```
alter session set "_optimizer_nul_aware_antijon"=false;
select /* in with aaj=off */ department_name
from hr.departments dept
where department_id not in (select department_id from hr.employees emp);
```

eg:

```
alter session set "_optimizer_null_aware_antijoin"=true;
set echo off
@fsp
```

11.9 反联结限制条件

11.10 反联结必要条件

语句必须使用 not in (!= all) 或 not exists

语句必须在 not in 或 not exists 子句中有一个子查询

not in 或 not exists 子句不能包含在 or 分支中

not exists 子句中的子查询必须与外层查询相关

10g 需要 not in 子查询的代码中确定不会返回空值