1 执行计划

执行计划显示SQL优化器的查询执行路径。
优化器：Oracle数据库内置的软件，决定了执行SQL语句最有效的方式，查询优化器由查询转换器、估计器和计划生成器组成。

2 启用执行计划

2.1 创建plan_table

[oracle@strong ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Thu Aug 2 17:49:16 2018

SQL> conn / as sysdba
Connected.
SQL> @$ORACLE_HOME/rdbms/admin/utlxplan.sql

Table created.

2.2 创建Plustrace角色

SQL> @$ORACLE_HOME/sqlplus/admin/plustrce.sql

2.3 授权Plustrace角色

SQL> grant plustrace to public;

Grant succeeded.

SQL> grant all on plan_table to public;

Grant succeeded.

3 控制Autotrace报告

Autotrace Setting	Result
SET AUTOTRACE OFF	No AUTOTRACE report is generated. This is the default.
SET AUTOTRACE ON EXPLAIN	The AUTOTRACE report shows only the optimizer execution path.
SET AUTOTRACE ON STATISTICS	The AUTOTRACE report shows only the SQL statement execution statistics.
SET AUTOTRACE ON	The AUTOTRACE report includes both the optimizer execution path and the SQL statement execution statistics.
SET AUTOTRACE TRACEONLY	Like SET AUTOTRACE ON, but suppresses the printing of the user's query output, if any. If STATISTICS is enabled, query data is still fetched, but not printed.

4 统计信息（Statistics）说明

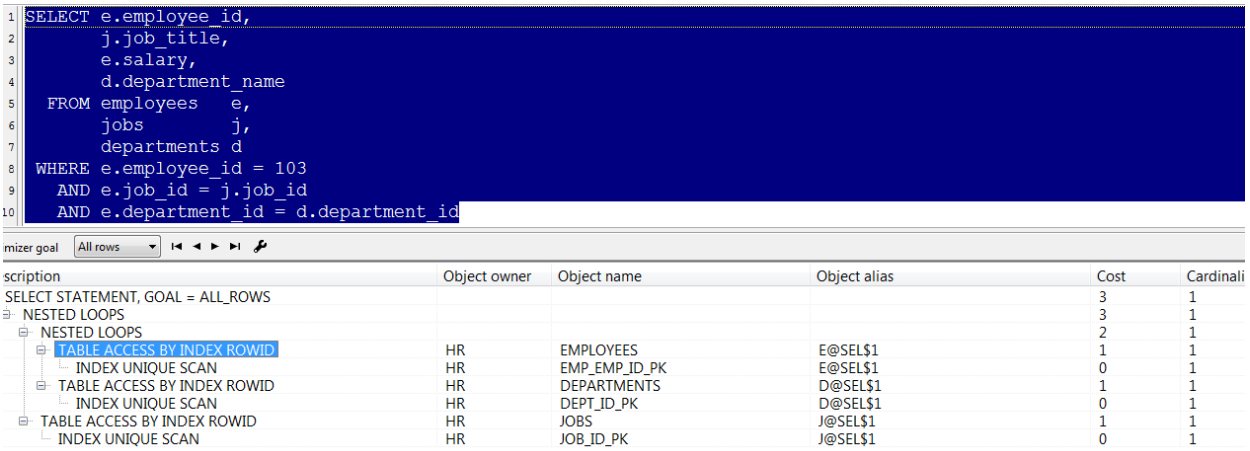
Database Statistic Name	Description
recursive calls	Number of recursive calls generated at both the user and system level. Oracle Database maintains tables used for internal processing. When Oracle Database needs to make a change to these tables, it internally generates an internal SQL statement, which in turn generates a recursive call.
db block gets	Number of times a CURRENT block was requested.
consistent gets	Number of times a consistent read was requested for a block
physical reads	Total number of data blocks read from disk. This number equals the value of "physical reads direct" plus all reads into buffer cache.
redo size	Total amount of redo generated in bytes
bytes sent through SQL*Net to client	Total number of bytes sent to the client from the foreground processes.
bytes received through SQL*Net from client	Total number of bytes received from the client over Oracle Net.
SQL*Net round-trips to/from client	Total number of Oracle Net messages sent to and received from the client
sorts (memory)	Number of sort operations that were performed completely in memory and did not require any disk writes
sorts (disk)	Number of sort operations that required at least one disk write
rows processed	Number of rows processed during the operation

5 查看执行计划方法

5.1 GUI工具

PL/SQL Developer （F5键）

```
SELECT e.employee_id,
       j.job_title,
       e.salary,
       d.department_name
  FROM employees   e,
       jobs        j,
       departments d
 WHERE e.employee_id = 103
   AND e.job_id = j.job_id
   AND e.department_id = d.department_id;
```



scription	Object owner	Object name	Object alias	Cost	Cardinali
SELECT STATEMENT, GOAL = ALL_ROWS				3	1
NESTED LOOPS				3	1
NESTED LOOPS				2	1
TABLE ACCESS BY INDEX ROWID	HR	EMPLOYEES	E@SEL$1	1	1
INDEX UNIQUE SCAN	HR	EMP_EMP_ID_PK	E@SEL$1	0	1
TABLE ACCESS BY INDEX ROWID	HR	DEPARTMENTS	D@SEL$1	1	1
INDEX UNIQUE SCAN	HR	DEPT_ID_PK	D@SEL$1	0	1
TABLE ACCESS BY INDEX ROWID	HR	JOBS	J@SEL$1	1	1
INDEX UNIQUE SCAN	HR	JOB_ID_PK	J@SEL$1	0	1

5.2 Autotrace功能

阅读和分析执行计划的方法：**从右到左，从上到下，由内及外，细致观察，通盘分析**。

```
SQL> set autotrace on
SQL> SELECT e.employee_id,
       j.job_title,
       e.salary,
       d.department_name
  FROM employees   e,
       jobs        j,
```

```
      departments d
 WHERE e.employee_id = 103
   AND e.job_id = j.job_id
   AND e.department_id = d.department_id;

EMPLOYEE_ID JOB_TITLE                         SALARY DEPARTMENT_NAME
----------- ---------------------------------- ---------- ------------------------------
        103 Programmer                           9000 IT


Execution Plan
----------------------------------------------------------
Plan hash value: 4029615031

-------------------------------------------------------------------------------------------
---
| Id  | Operation                    | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
---
|   0 | SELECT STATEMENT             |               |     1 |    63 |     3   (0)| 00:00:01 |
|   1 |  NESTED LOOPS                |               |     1 |    63 |     3   (0)| 00:00:01 |
|   2 |   NESTED LOOPS               |               |     1 |    36 |     2   (0)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID| EMPLOYEES     |     1 |    20 |     1  (0)|
00:00:01 |
|*  4 |     INDEX UNIQUE SCAN        | EMP_EMP_ID_PK |     1 |       |     0   (0)|
00:00:01 |
|   5 |    TABLE ACCESS BY INDEX ROWID| DEPARTMENTS   |     1 |    16 |     1   (0)|
00:00:01 |
|*  6 |     INDEX UNIQUE SCAN        | DEPT_ID_PK    |     1 |       |     0   (0)|
00:00:01 |
|   7 |   TABLE ACCESS BY INDEX ROWID | JOBS          |     1 |    27 |     1   (0)|
00:00:01 |
|*  8 |    INDEX UNIQUE SCAN         | JOB_ID_PK     |     1 |       |     0  (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
---
```

```
Predicate Information (identified by operation id):
---------------------------------------------------

   4 - access("E"."EMPLOYEE_ID"=103)
   6 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
   8 - access("E"."JOB_ID"="J"."JOB_ID")
```

紧接上面的输出，输出统计信息以及谓词信息：

```
Statistics
----------------------------------------------------------
          8  recursive calls
          0  db block gets
          8  consistent gets
          0  physical reads
          0  redo size
        779  bytes sent via SQL*Net to client
        607  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed

SQL> set autotrace off
```

说明：

执行顺序为：4-3-6-5-2-8-7-1-0

清空缓冲区：

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;
```

System altered.

5.3 DBMS_XPLAN包

1）执行SQL语句

```
SQL> explain plan for
  2  SELECT e.employee_id,
       j.job_title,
       e.salary,
       d.department_name
  FROM employees    e,
       jobs         j,
       departments d
 WHERE e.employee_id = 103
   AND e.job_id = j.job_id
   AND e.department_id = d.department_id;

Explained.
```

2）查看执行计划（方法一）

```
SQL> select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
--------------------------------
Plan hash value: 4029615031

--------------------------------------------------------------------------------
---
| Id  | Operation                    | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
---
```

```
|   0 | SELECT STATEMENT            |             |     1 |    63 |     3       (0)| 00:00:01 |
|   1 |  NESTED LOOPS               |             |     1 |    63 |     3       (0)| 00:00:01 |
|   2 |   NESTED LOOPS              |             |     1 |    36 |     2       (0)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID| EMPLOYEES     |     1 |    20 |     1  (0)|
00:00:01 |
|*  4 |     INDEX UNIQUE SCAN             | EMP_EMP_ID_PK |     1 |       |     0   (0)|
00:00:01 |
|   5 |    TABLE ACCESS BY INDEX ROWID| DEPARTMENTS   |     1 |    16 |     1       (0)|
00:00:01 |

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------
------------------------------
|*  6 |     INDEX UNIQUE SCAN             | DEPT_ID_PK    |     1 |       |     0       (0)|
00:00:01 |
|   7 |   TABLE ACCESS BY INDEX ROWID | JOBS          |     1 |    27 |     1       (0)|
00:00:01 |
|*  8 |    INDEX UNIQUE SCAN       | JOB_ID_PK     |     1 |       |     0  (0)| 00:00:01 |
--------------------------------------------------------------------------------------------
---

Predicate Information (identified by operation id):
---------------------------------------------------

   4 - access("E"."EMPLOYEE_ID"=103)
   6 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
   8 - access("E"."JOB_ID"="J"."JOB_ID")

22 rows selected.
```

3）查看执行计划（方法二）

```
SQL> select to_char(dbms_xplan.display_plan()) from dual;

TO_CHAR(DBMS_XPLAN.DISPLAY_PLAN())
```

```
-------------------------------------------------------------------------------------------
------------------------------
 Plan Hash Value  : 4029615031

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name            | Rows | Bytes | Cost | Time      |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                 |    1 |    63 |    3 | 00:00:01 |
|   1 |   NESTED LOOPS               |                 |    1 |    63 |    3 | 00:00:01 |
|   2 |    NESTED LOOPS              |                 |    1 |    36 |    2 | 00:00:01 |
|   3 |     TABLE ACCESS BY INDEX ROWID | EMPLOYEES       |    1 |    20 |    1 |
00:00:01 |
| * 4 |      INDEX UNIQUE SCAN       | EMP_EMP_ID_PK |    1 |       |    0 | 00:00:01 |
|   5 |     TABLE ACCESS BY INDEX ROWID | DEPARTMENTS   |    1 |    16 |    1 |
00:00:01 |

TO_CHAR(DBMS_XPLAN.DISPLAY_PLAN())
-------------------------------------------------------------------------------------------
------------------------------
| * 6 |      INDEX UNIQUE SCAN       | DEPT_ID_PK    |    1 |       |    0 | 00:00:01 |
|   7 |    TABLE ACCESS BY INDEX ROWID | JOBS          |    1 |    27 |    1 |
00:00:01 |
| * 8 |     INDEX UNIQUE SCAN        | JOB_ID_PK     |    1 |       |    0 | 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------
* 4 - access("E"."EMPLOYEE_ID"=103)
* 6 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
* 8 - access("E"."JOB_ID"="J"."JOB_ID")
```

5.4 SQL跟踪方法

1）10046事件

```
SQL> alter session set tracefile_identifier='tt';

Session altered.
SQL> alter session set events'10046 trace name context forever,level 12';

Session altered.
SQL> set linesize 200
SQL> SELECT e.employee_id,
       j.job_title,
       e.salary,
       d.department_name
  FROM hr.employees    e,
       hr.jobs         j,
       hr.departments d
 WHERE e.employee_id = 103
   AND e.job_id = j.job_id
   AND e.department_id = d.department_id;

EMPLOYEE_ID JOB_TITLE                           SALARY DEPARTMENT_NAME
----------- ---------------------------------- ---------- ------------------------------
        103 Programmer                               9000 IT

SQL> alter session set events '10046 trace name context off';

Session altered.

SQL> SELECT DISTINCT t3.SPID FROM v$mystat t1 JOIN v$session t2 ON
t1.SID=t2.SID JOIN v$process t3 ON t2.PADDR=t3.ADDR;

SPID
------------------------
28280
```

说明：10046事件比SQL_TRACE跟踪获取的信息更多，有四个级别，分别为1、4、8、12，获取的信息逐级增多，级别1相当于SQL_TRACE。

2）Optimizer_Trace（10053追踪事件）

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

Session altered.

SQL> set linesize 200
SQL> SELECT e.employee_id,
       j.job_title,
       e.salary,
       d.department_name
  FROM hr.employees    e,
       hr.jobs         j,
       hr.departments d
 WHERE e.employee_id = 103
   AND e.job_id = j.job_id
   AND e.department_id = d.department_id;

EMPLOYEE_ID JOB_TITLE                          SALARY DEPARTMENT_NAME
----------- ---------------------------------- ---------- ------------------------------
        103 Programmer                               9000 IT

SQL> alter session set events '10053 trace name context off';

Session altered.

SQL> SELECT DISTINCT t3.SPID FROM v$mystat t1 JOIN v$session t2 ON
t1.SID=t2.SID JOIN v$process t3 ON t2.PADDR=t3.ADDR;

SPID
------------------------
28401
```

说明：10053事件有2个级别，分别为1和2，获取信息量逐级减少，级别2是级别1的子集，该方法会在跟踪文件里记录优化器分析、选择执行计划过程，10053生效有两个条件：SQL语句发生硬解析；优化器是CBO模式。