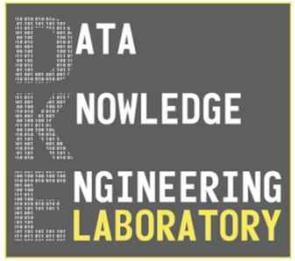
COMP322003 Introduction to Database (Fall 2018)



LAB 11: EXPLAIN PLAN

- For accessing the information the execution plan of a given SQL query

Prof. Young-Kyoon Suh

How to View a Query Plan on Oracle?

- Use the `EXPLAIN PLAN` statement.
  - The information of an explained plan by the statement is stored into `PLAN_TABLE`, which is provided by Oracle by default.

```
SQL> desc PLAN_TABLE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 STATEMENT_ID                                       VARCHAR2(30)
 PLAN_ID                                            NUMBER
 TIMESTAMP                                          DATE
 REMARKS                                            VARCHAR2(4000)
 OPERATION                                          VARCHAR2(30)
 OPTIONS                                            VARCHAR2(255)
 OBJECT_NODE                                        VARCHAR2(128)
 OBJECT_OWNER                                       VARCHAR2(128)
 OBJECT_NAME                                        VARCHAR2(128)
 OBJECT_ALIAS                                       VARCHAR2(261)
 OBJECT_INSTANCE                                    NUMBER(38)
 OBJECT_TYPE                                        VARCHAR2(30)
 OPTIMIZER                                          VARCHAR2(255)
```

How to View a Query Plan on Oracle? (Cont'd)

```
SEARCH_COLUMNS                   NUMBER
ID                               NUMBER(38)
PARENT_ID                        NUMBER(38)
DEPTH                            NUMBER(38)
POSITION                         NUMBER(38)
COST                             NUMBER(38)
CARDINALITY                      NUMBER(38)
BYTES                            NUMBER(38)
OTHER_TAG                        VARCHAR2(255)
PARTITION_START                  VARCHAR2(255)
PARTITION_STOP                   VARCHAR2(255)
PARTITION_ID                     NUMBER(38)
OTHER                            LONG
OTHER_XML                        CLOB
DISTRIBUTION                     VARCHAR2(30)
CPU_COST                         NUMBER(38)
IO_COST                          NUMBER(38)
TEMP_SPACE                       NUMBER(38)
ACCESS_PREDICATES                VARCHAR2(4000)
FILTER_PREDICATES                VARCHAR2(4000)
PROJECTION                       VARCHAR2(4000)
TIME                             NUMBER(38)
QBLOCK_NAME                      VARCHAR2(128)

SQL> _
```

`EXPLAIN PLAN` on Oracle

- Basic syntax below

Syntax
`EXPLAIN PLAN` **`FOR`** *query_statement*

- Query statement: an `SELECT` statement

EXPLAIN PLAN FOR Practice

```
SELECT P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate
FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
WHERE P.Dnum=D.Dnumber AND D.Mgr_ssn=E.Ssn AND P.Plocation='Stafford'
```

```
SQL> EXPLAIN PLAN FOR
  2    SELECT P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate
  3    FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
  4    WHERE P.Dnum=D.Dnumber AND D.Mgr_ssn=E.Ssn AND P.Plocation='Stafford';

Explained.

SQL>
```

How to Display an Explained Plan?

- DISPLAY
  - Declared in the DBMS_XPLAN system package (installed by default)
  - Function to show a query execution plan for a single SQL statement

```
FUNCTION DISPLAY(TABLE_NAME   VARCHAR2 DEFAULT 'PLAN_TABLE',
                 STATEMENT_ID VARCHAR2 DEFAULT NULL,
                 FORMAT       VARCHAR2 DEFAULT 'TYPICAL',
                 FILTER_PREDS VARCHAR2 DEFAULT NULL)
```

파라미터	설명
TABLE_NAME	Execution Plan이 저장되는 테이블을 테이블을 지정하며, Defalut는 'PLAN_TABLE'이다.
STATEMENT_ID	Execution Plan시 SET STATEMENT_ID를 지정한 경우 이를 불러올 수 있다. 값이 NULL일 경우 마지막에 실행된 문장을 불러온다.
FORMAT(BASIC)	가장 기본적인 정보만 보여줌
FORMAT(TYPICAL)	Format의 Default값인 Typical은 SQL 튜닝에 필요한 Normal한 정보를 보여줌 SQL 튜닝에 가장 유용하게 사용되는 Predicate Information이 제공된다
FORMAT(ALL)	Typical Format에 Query Block Name과 Column Projection Information이 추가로 제공된다
FORMAT(OUTLINE)	Typical Format에 추가적으로 Hidden Hint인 Outline Global Hint를 제공한다
FORMAT(ADVANCED)	ALL Format에 OUTLINE Format를 합친 정보를 제공한다
FILTER_PREDS	저장된 PLAN에서 일부 Row 또는 Row Set을 제한하여 출력할 수 있다.

How to Call the Display Function?

- For basic information,
  - `SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE', NULL, 'BASIC', NULL));`

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY('PLAN_TABLE', NULL, 'BASIC', NULL));

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
Plan hash value: 3383296466

-------------------------------------------------------------
| Id  | Operation                     | Name         |
-------------------------------------------------------------
|   0 | SELECT STATEMENT              |              |
|   1 |  NESTED LOOPS                 |              |
|   2 |   NESTED LOOPS                |              |
|   3 |    NESTED LOOPS               |              |
|   4 |     TABLE ACCESS FULL         | PROJECT      |
|   5 |     TABLE ACCESS BY INDEX ROWID| DEPARTMENT   |

PLAN_TABLE_OUTPUT
-------------------------------------------------------------
|   6 |      INDEX UNIQUE SCAN        | SYS_C0016378 |
|   7 |    INDEX UNIQUE SCAN          | SYS_C0016383 |
|   8 |   TABLE ACCESS BY INDEX ROWID | EMPLOYEE     |
-------------------------------------------------------------

15 rows selected.
```

- `Id` indicates a (tree) *node identifier*.
- Each `Operation` represents a *query operator* (node) (in the query tree).
- `Name` indicates which object is associated with the corresponding operator node.

How to Call the Display Function? (Cont'd)

- `SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);`

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------
Plan hash value: 3383296466

------------------------------------------------------------------------------
---------------

| Id  | Operation                      | Name         | Rows  | Bytes | Cost (%C
PU)| Time     |

------------------------------------------------------------------------------
---------------


PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |              |     2 |   154 |     7
(0)| 00:00:01 |

|   1 |  NESTED LOOPS                  |              |     2 |   154 |     7
(0)| 00:00:01 |

|   2 |   NESTED LOOPS                 |              |     2 |   154 |     7
(0)| 00:00:01 |

|   3 |    NESTED LOOPS                |              |     2 |    56 |     5
(0)| 00:00:01 |
```

- Shows for query tuning proper information of the explained plan

How to Call the Display Function? (Cont'd)

- `SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);`

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------

|*  4 |     TABLE ACCESS FULL          | PROJECT      |     2 |    30 |     3
(0)| 00:00:01 |

|   5 |     TABLE ACCESS BY INDEX ROWID| DEPARTMENT   |     1 |    13 |     1
(0)| 00:00:01 |

|*  6 |      INDEX UNIQUE SCAN         | SYS_C0016378 |     1 |       |     0
(0)| 00:00:01 |

|*  7 |    INDEX UNIQUE SCAN           | SYS_C0016383 |     1 |       |     0

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------
(0)| 00:00:01 |

|   8 |   TABLE ACCESS BY INDEX ROWID  | EMPLOYEE     |     1 |    49 |     1
(0)| 00:00:01 |

------------------------------------------------------------------------------
---------------
```

How to Call the Display Function? (Cont'd)

- `SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);`

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

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------

   4 - filter("P"."PLOCATION"='Stafford')
   6 - access("P"."DNUM"="D"."DNUMBER")
   7 - access("D"."MGR_SSN"="E"."SSN")

Note
-----
   - this is an adaptive plan

26 rows selected.
```

// predicate information

- "Access predicate": a predicate used to determine access type
- "Filter predicate": a predicate to filter out tuples not satisfying a selection condition

Lab #11: Playing with `EXPLAIN PLAN`

1) Display the output (in the `TYPICAL` format by default) of `EXPLAIN PLAN` for the following query:

```
SELECT e1.Fname, e1.Lname
FROM   EMPLOYEE e1
WHERE  e1.Salary = (SELECT MAX(Salary)
                    FROM  EMPLOYEE e2)
```

- Q: How many *bytes* will be needed for the final output?

Lab #11: Playing with `EXPLAIN PLAN` (Cont'd)

2) Display the output in the `BASIC` format by default of `EXPLAIN PLAN` for the following query:

```
SELECT COUNT(*)
FROM   DEPARTMENT d
WHERE  d.Dnumber IN (SELECT E.Dno
                     FROM   EMPLOYEE E
                     WHERE  E.Salary > 20000);
```

- Q: *Which join* algorithm is used for this query?

Lab #11: Playing with `EXPLAIN PLAN` (Cont'd)

3) Display the output in the `BASIC` format by default of `EXPLAIN PLAN` for the following query:

```
SELECT /*+ USE_MERGE(EMPLOYEE PROJECT WORKS_ON) */ Fname, Lname
FROM   EMPLOYEE, PROJECT, WORKS_ON
WHERE  Ssn = Essn AND Pno = Pnumber AND Pname = 'ProductX'
ORDER BY Lname desc;
```

- ***Hints***: specified by application developer
  - Syntax: `/*+ ... */`
    - Embedded between `SELECT` and a list of projection columns
  - *Comments* in a SQL statement that pass instructions to the Oracle optimizer.
  - The optimizer uses *these hints* to choose an execution plan for the statement, unless some condition exists that prevents the optimizer from doing so.
  - Types: access path, join order, ***join method***, enabling/disabling a transformation

- Q1: *What join algorithms* are used for this plan?
- Q2: *How many operators* are used for this plan?
- Q3: *Which table* is fully scanned in outer loop?

Lab #11: Playing with `EXPLAIN PLAN` (Cont'd)

[Submission]

- Deadline: **Sunday midnight** (11/25/2018)
- Name your file like:
  - '`lab11-`Your_Student_ID`.sql`'
  - Put each answer into this `sql` file. If your answer should be written in plain text, then just comment that line.
    - "`-- 11-1) …`"
  - Save as image file captured output.
- Zip the sql and any screenshot image files.
- Upload the zip file into LMS.

References

- http://wiki.gurubee.net/display/DBSTUDY/EXPLAIN+PLAN
- https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements006.htm#SQLRF51101