

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO BÀI TẬP LỚN**  
**Học phần: Thiết kế quản trị cơ sở dữ liệu**

**Đề tài: Đọc Hiểu Diễn Giải Kế Hoạch Thực Thi**

**Nhóm sinh viên thực hiện:**

- |                           |                 |
|---------------------------|-----------------|
| <b>1. Lê Trung Kiên</b>   | <b>20111717</b> |
| <b>2. Đỗ Văn Tuấn</b>     | <b>20112412</b> |
| <b>3. Phùng Thanh Huy</b> | <b>20111663</b> |

**Giảng viên hướng dẫn: TS. Trần Việt Trung**

**Hà Nội 5 – 2015**

## Nội Dung

1. Mục tiêu .....	4
2. Kế hoạch thực thi là gì? .....	4
3. Nơi tìm kế hoạch thực thi? .....	4
4. Xem kế hoạch thực thi .....	5
5. Giải thích kế hoạch .....	5
6. Ví dụ:.....	6
7. Bảng kế hoạch .....	7
8. Hiển thị từ bảng kế hoạch: Đặc trưng.....	7
9. Hiển thị từ bảng kế hoạch: Tất cả .....	8
10. Hiện thị bảng kế hoạch : nâng cao.....	9
11. Kế hoạch giải thích sử dụng SQL Developer .....	11
12. Autotrace.....	11
13. Cú pháp của lệnh 'AUTOTRACE'.....	12
14. Một số ví dụ về 'AUTOTRACE'.....	12
15. AUTOTRACE: Thống kê .....	13
16. AUTOTRACE: Sử dụng công cụ SQL Developer .....	14
17. Sử dụng khung nhìn 'V\$SQL_PLAN' .....	14
18. Một số cột trong 'V\$SQL_PLAN'.....	15
19. Kết nối giữa các khung nhìn thực thi động quan trọng .....	16
20. Truy vấn 'V\$SQL_PLAN' .....	17
21. Kho chứa lịch sử công việc tự động (AWR) .....	18
22. Quản lý AWR với PL/SQL .....	19
23. Các khung nhìn quan trọng của AWR.....	20
24. Truy vấn AWR.....	20
25. Sản sinh báo cáo SQL từ dữ liệu AWR.....	21
26. SQL Monitoring: Tổng quan.....	22
27. Ví Dụ SQL Monitoring Report: .....	23
28. Interpreting an Execution Plan(Giải thích một Kế hoạch thực thi) .....	24
29. Execution Plan Interpretation: Example 1 .....	26
30. Execution Plan Interpretation: Example 2.....	28
31. Execution Plan Interpretation: Example 3.....	30

<b>32.</b>	<b>Reading More Complex Execution Plans .....</b>	<b>31</b>
<b>33.</b>	<b>Reviewing the Execution Plan .....</b>	<b>31</b>
<b>34.</b>	<b>Looking Beyond Execution Plans .....</b>	<b>32</b>
	<b>Phân Công Công Việc.....</b>	<b>33</b>

## 1. Mục tiêu

- Tổng hợp kế hoạch thực thi
- Trình bày được kế hoạch thực thi
- Giải thích kế hoạch

## 2. Kế hoạch thực thi là gì?

- Kế hoạch thực thi một câu lệnh SQL được bao gồm xây dựng các khối nhỏ gọi là các nguồn hàng cho một loạt kế hoạch thực thi.
- Sự kết hợp của các nguồn hàng cho một câu lệnh được gọi là kế hoạch thực thi.
- Bằng cách sử dụng các mối quan hệ cha-con, kế hoạch thực thi có thể được hiển thị trong một cấu trúc giống như cây

Một kế hoạch thực thi là đầu ra của bộ tối ưu hóa và thể hiện các bước thực hiện cho bộ thực thi. Kế hoạch chỉ thị các hoạt động của bộ thực thi phải thực hiện để truy vấn lấy dữ liệu một cách hiệu quả nhất.

Tối ưu hóa các kế hoạch thực thi cho các thao tác SELECT, UPDATE, INSERT, và DELETE. Các bước thực hiện kế hoạch không nhất thiết phải thực hiện theo thứ tự mà chúng được đánh số. Có các quan hệ cha con giữa các bước.

Kế hoạch thực thi cho thấy các thông tin sau:

- Thứ tự của các bảng được gọi trong câu lệnh.
- Phương pháp truy vấn cho mỗi bảng trong câu lệnh.
- Các phép toán xử lý dữ liệu như bộ lọc, sắp xếp hoặc tập hợp.
- Phương pháp kết nối các bảng tác động bởi phép toán kết nối trong câu lệnh.

Ngoài các luồng dữ liệu cây cho các phép toán song song, bảng kế hoạch còn cho biết các thông tin :

- Tối ưu hóa, chẳng hạn như chi phí của mỗi hoạt động
- Phân vùng, chẳng hạn như tập hợp các phân vùng đã được truy cập
- Thực hiện song song, chẳng hạn như các phương pháp phân phối đầu vào của các phép toán kết nối.

## 3. Nơi tìm kế hoạch thực thi?

Có rất nhiều cách để lấy kế hoạch thực thi bên trong cơ sở dữ liệu.

- Lệnh EXPLAIN PLAN cho phép bạn xem kế hoạch thực thi mà bộ tối ưu hóa có thể sử dụng để thực thi câu lệnh SQL. Lệnh này rất hữu ích vì nó đưa ra kế hoạch sơ bộ mà bộ tối ưu có thể sử dụng và chèn nó vào một bảng gọi là PLAN\_TABLE mà không cần thực hiện câu lệnh SQL. Lệnh này có sẵn từ SQL \* Plus hoặc SQL Developer
- V\$SQL\_PLAN cung cấp một cách để kiểm tra kế hoạch thực thi của các con trỏ được thực thi gần đây. Thông tin trong V\$SQL\_PLAN rất giống như đầu ra của một câu lệnh

EXPLAIN\_PLAN. Tuy nhiên, trong khi EXPLAIN\_PLAN cho thấy một kế hoạch lý thuyết có thể được sử dụng nếu câu lệnh này được thực hiện, V\$SQL\_PLAN chứa các kế hoạch thực tế sử dụng.

- V\$SQL\_PLAN\_MONITOR hiển thị các thông kê theo dõi mức độ kết hoạch thực thi cho mỗi câu lệnh SQL được tìm thấy trong V\$SQL\_MONITOR. Mỗi hàng trong V\$SQL\_PLAN\_MONITOR tương ứng với một phép toán trong kế hoạch thực thi được theo dõi.
- The Automatic Workload Repository (AWR) là cơ sở hạ tầng và kho lưu trữ Statspack cho kế hoạch thực thi của các câu lệnh SQL được thực hiện thường xuyên nhất. Các kế hoạch là các bản ghi trong DBA\_HIST\_SQL\_PLAN hoặc STAT\$SQL\_PLAN.
- Kế hoạch và các phép toán trên dòng gốc được đổ vào các tập tin lưu vết được tạo ra bởi DBMS\_MONITOR
- Các cơ sở quản lý SQL (SMB) là một phần của từ điển dữ liệu nằm bên trong các không gian bảng SYSAUX. Nó lưu trữ các kết quả, lịch sử kế hoạch, kế hoạch cơ sở của SQL, cũng như cấu hình SQL.
- Các sự kiện 10053, được sử dụng để tối ưu hóa chi phí dựa trên (CBO) tính toán có thể bao gồm một kế hoạch.

#### 4. Xem kế hoạch thực thi

Nếu bạn thực hiện lệnh **EXPLAIN\_PLAN** trên **SQL \* Plus**, thì sau đó có thể dùng lệnh **SELECT from PLAN\_TABLE** để xem kế hoạch thực thi. Có một số kịch bản **SQL \* Plus** có sẵn để định dạng đầu ra bảng kế hoạch. Cách dễ nhất để xem một kế hoạch thực thi là sử dụng các gói phần mềm DBMS\_XPLAN. Gói phần mềm DBMS\_XPLAN có chức năng:

- DISPLAY: Để định dạng và hiển thị nội dung của một bảng kế hoạch
  - DISPLAY\_AWR: Để định dạng và hiển thị nội dung kế hoạch thực thi của câu lệnh SQL được lưu trữ trong AWR
  - DISPLAY\_CURSOR: Để định dạng và hiển thị nội dung kế hoạch thực thi của bất kỳ con trỏ đã được thực thi.
  - DISPLAY\_SQL\_PLAN\_BASELINE: Để hiển thị một hoặc nhiều kế hoạch cho các câu lệnh SQL được xác định bởi bộ điều khiển SQL.
  - DISPLAY\_SQLSET: Để định dạng và hiển thị nội dung kế hoạch thực thi của câu lệnh được lưu trữ trong tập tin chỉnh SQL.
- ⇒ Một lợi thế của việc sử dụng các chức năng trong gói DBMS\_XPLAN với đầu ra là định dạng nhất quán mà không quan tâm đến nguồn.

#### 5. Giải thích kế hoạch

Lệnh EXPLAIN\_PLAN được sử dụng để sản sinh các kế hoạch thực thi mà bộ tối ưu hóa sử dụng để thực thi một câu lệnh SQL. Nó không thực thi câu lệnh, chỉ đơn giản là tạo ra các kế hoạch có thể được sử dụng, và chèn kế hoạch này vào một bảng. Nếu bạn kiểm tra kế hoạch, bạn có thể xem Oracle Server thực hiện các câu lệnh như thế nào.

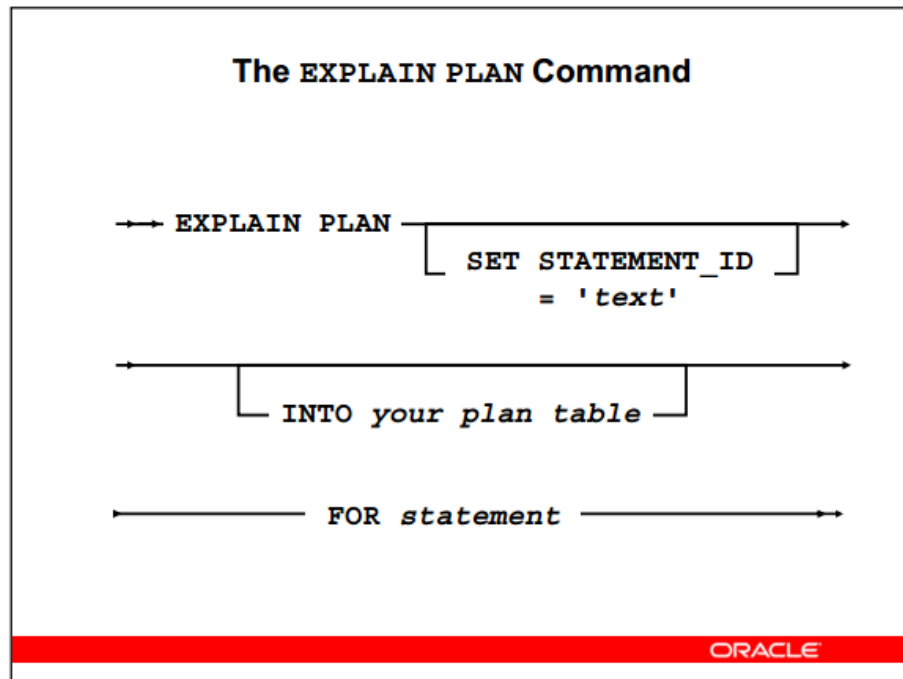
Sử dụng EXPLAIN\_PLAN:

- Đầu tiên sử dụng lệnh EXPLAIN\_PLAN để giải thích một câu lệnh SQL.

- Sau đó lấy từng bước trong kế hoạch bằng cách truy vấn PLAN\_TABLE

Lưu ý: bạn có thể tạo riêng PLAN\_TABLE bằng cách sử dụng kịch bản \$ORACLE\_HOME / rdbms / admin / utlxplan.sql nếu bạn muốn giữ lại các thông tin kế hoạch thực thi cho một thời gian dài.

PLAN\_TABLE được tự động tạo ra như là một bảng tạm thời toàn cục để lưu giữ kết quả của các câu lệnh EXPLAIN\_PLAN cho tất cả người dùng. PLAN\_TABLE là bảng mẫu đầu ra mặc định nơi câu lệnh EXPLAIN\_PLAN chèn vào các dòng mô tả kế hoạch thực thi.



## 6. Ví dụ:

```

SQL> EXPLAIN PLAN
  2  SET STATEMENT_ID = 'demo01' FOR
  3  SELECT e.last_name, d.department_name
  4  FROM hr.employees e, hr.departments d
  5  WHERE e.department_id = d.department_id;

Explained.

SQL>

```

Lệnh này sẽ chèn kế hoạch thực thi của câu lệnh SQL vào trong bảng kế hoạch và thêm một nhãn tên tùy chọn "demo01" để tham khảo trong tương lai. Bạn cũng có thể sử dụng cú pháp sau đây:

*EXPLAIN PLAN FOR*

*SELECT e.last\_name, d.department\_name*

```
FROM hr.employees e, hr.departments d  
WHERE e.department_id = d.department_id;
```

## 7. Bảng kế hoạch

- Được tự động tạo ra để chứa kết quả của câu lệnh EXPLAIN PLAN.
- Bạn có thể tạo ra bằng cách sử dụng kịch bản utlxplan.sql.
- Ưu điểm: Không cần thực thi câu lệnh SQL.
- Nhược điểm: Có thể phải là kế hoạch thực thi thực tế.
- Bảng kế hoạch là bảng có thứ bậc.
- Hệ thống cấp bậc được thiết lập thông qua cột ID và PARENT\_ID.

Có nhiều phương pháp khác nhau để thu thập các kế hoạch thực thi. Chúng ta mới giới thiệu về câu lệnh EXPLAIN\_PLAN. Câu lệnh này tập hợp các kế hoạch thực thi của câu lệnh SQL mà không thực hiện nó, đồng thời đẩy kết quả vào trong bảng PLAN\_TABLE. Các phương pháp để thu thập và hiển thị các kế hoạch giải thích, các định dạng cơ bản và mục tiêu đều giống nhau. Tuy nhiên, PLAN\_TABLE chỉ cho bạn thấy một kế hoạch mà có thể không phải là kế hoạch được chọn bởi bộ tối ưu hóa. PLAN\_TABLE được tự động tạo ra như là một bảng tạm thời toàn cục và hiển thị với tất cả người dùng. PLAN\_TABLE là bảng mẫu đầu ra mặc định chứa các dòng mô tả kế hoạch thực thi của EXPLAIN\_PLAN. PLAN\_TABLE được tổ chức trong một cấu trúc cây và bạn có thể lấy về cấu trúc đó bằng cách sử dụng cả hai cột ID và PARENT\_ID với mệnh đề CONNECT BY trong một câu lệnh SELECT. Trong khi bảng PLAN\_TABLE được tự động thiết lập cho mỗi người dùng, bạn có thể sử dụng kịch bản SQL utlxplan.sql để tự tạo một PLAN\_TABLE cục bộ trong lược đồ của bạn và sử dụng nó để lưu trữ các kết quả của EXPLAIN PLAN. Tên chính xác và vị trí của kịch bản này phụ thuộc vào hệ điều hành của bạn. Oracle khuyến cáo bạn nên xóa và xây dựng lại bảng PLAN\_TABLE địa phương của bạn sau khi nâng cấp phiên bản của cơ sở dữ liệu vì các cột có thể thay đổi. Điều này có thể gây ra các kịch bản thất bại hoặc gây TKPROF thất bại, nếu như bạn dùng tên bảng tuyệt đối.

## 8. Hiển thị từ bảng kế hoạch: Đặc trưng

```

SQL> EXPLAIN PLAN SET STATEMENT_ID = 'demo01' FOR SELECT * FROM emp
2 WHERE ename = 'KING';

Explained.

SQL> SET LINESIZE 130
SQL> SET PAGESIZE 0
SQL> select * from table(DBMS_XPLAN.DISPLAY());

Plan hash value: 3956160932

-----
| Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time     |
-----+-----+-----+-----+-----+-----+-----+
|  0 | SELECT STATEMENT    |      |    1 |    37 |    3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP  |    1 |    37 |    3   (0)| 00:00:01 |
-----+-----+-----+-----+-----+

Predicate Information (identified by operation id):
-----
1 - filter("ENAME"='KING')

```

Trong ví dụ trong slide, lệnh EXPLAIN\_PLAN chèn các kế hoạch thực thi của câu lệnh SQL vào trong PLAN\_TABLE và thêm nhãn tên demo01 để tham chiếu trong tương lai. Các chức năng của gói DISPLAY DBMS\_XPLAN có thể được sử dụng để định dạng và hiển thị câu lệnh cuối cùng được lưu trữ trong PLAN\_TABLE. Bạn cũng có thể sử dụng cú pháp sau đây để lấy kết quả tương tự:

```
SELECT * FROM TABLE (dbms_xplan.display('plan_table','demo01','typical',null));
```

Trong ví dụ này, bạn có thể thay thế tên bảng kế hoạch thay vì PLAN\_TABLE và demo01 để biểu diễn câu lệnh ID.

Khung nhìn TYPICAL(tiêu biểu) hiển thị các thông tin quan trọng nhất trong bảng kế hoạch. phép toán ID,tên và tùy chọn, số hàng, kích thước theo byte và chi phí tối ưu. Tham số cuối cùng trong chức năng DISPLAY là filter\_preds. Thông số này đại diện cho một bộ lọc tiền xử lý để hạn chế tập các dòng được chọn từ bảng mà kế hoạch này được lưu trữ. Khi giá trị là null (mặc định), kế hoạch được hiển thị tương ứng với kế hoạch của câu lệnh cuối cùng. Tham số này có thể tham chiếu bất kỳ cột của bảng mà kế hoạch này được lưu trữ và có thể chứa bất kỳ cấu trúc SQL nào ví dụ, truy vấn phụ hoặc các lời gọi hàm.

Lưu ý: Ngoài ra, bạn có thể chạy các kịch bản utlxpls.sql (hoặc utlxplp.sql trong truy vấn song song, nằm trong ORACLE\_HOME/rdbms/admin/directory) để hiển thị các kế hoạch thực thi lưu trữ trong PLAN\_TABLE giải thích cho câu truy vấn mới nhất. Kịch bản này sử dụng bảng chức năng DISPLAY từ gói DBMS\_XPLAN.

## 9. Hiển thị từ bảng kế hoạch: Tất cả



```
SQL> select * from table(DBMS_XPLAN.DISPLAY(null,null,'ALL'));

Plan hash value: 3956160932

-----
| Id | Operation          | Name | Rows | Bytes | Cost (%CPU)| Time     |
-----
|  0 | SELECT STATEMENT    |      |    1 |    37 |    3   (0)| 00:00:01 |
|*  1 | TABLE ACCESS FULL| EMP  |    1 |    37 |    3   (0)| 00:00:01 |
-----

Query Block Name / Object Alias (identified by operation id):

-----
1 - SEL$1 / EMP@SEL$1

Predicate Information (identified by operation id):

-----
1 - filter("ENAME"='KING')

Column Projection Information (identified by operation id):

-----
1 - "EMP"."EMPNO"[NUMBER,22], "EMP"."ENAME"[VARCHAR2,10], "EMP"."JOB"[VARCHAR2,9],
   "EMP"."MGR"[NUMBER,22], "EMP"."HIREDATE"[DATE,7], "EMP"."SAL"[NUMBER,22],
   "EMP"."COMM"[NUMBER,22], "EMP"."DEPTNO"[NUMBER,22]
```

Tùy chọn ALL được sử dụng với chức năng DISPLAY cho phép bạn xuất ra thông tin của người dùng có cấp cao nhất. Nó bao gồm các thông tin hiển thị với mức chuẩn, với thông tin bổ sung như PROJECTION, ALIAS, và thông tin về REMOTE SQL, nếu các phép toán là phân tán.

Để kiểm soát tốt hơn về thông tin hiển thị trên màn hình, các từ khóa sau đây có thể được thêm vào các tham số định dạng tùy chỉnh hành vi mặc định của nó. Mỗi từ khóa hoặc là đại diện cho một nhóm logic các cột trong bảng kế hoạch (như PARTITION) hoặc bổ sung hợp lý vào đầu ra của bảng kế hoạch cơ sở. Định dạng từ khóa này phải được phân cách bằng dấu phẩy hoặc một khoảng trắng:

- ROWS: hiển thị số lượng hàng ước tính bởi bộ tối ưu hóa.
- BYTE: hiển thị số lượng các byte ước tính bởi bộ tối ưu hóa.
- COST: hiển thị thông tin chi phí tối ưu
- PARTITION: hiển thị thông tin phân vùng cắt tia
- PARALLEL: hiển thị thông tin PX (phương pháp phân phối và bảng hàng đợi thông tin)
- PREDICATE: hiển thị phần trước đó.
- PROJECTION: hiển thị phần chiếu.
- ALIAS : hiển thị phần “Query Block Name/Object Alias”
- REMOTE: hiển thị các thông tin cho phân phối truy vấn.
- NOTE: hiển thị phần ghi chú của kế hoạch thực thi.

Nếu bảng kế hoạch mục tiêu cũng lưu trữ kế hoạch của các cột thống kê, bổ sung định dạng từ khóa có thể được sử dụng để xác định các lớp thống kê để hiển thị chức năng DISPLAY khi sử dụng. Những từ khóa định dạng bổ sung là IOSTATS, MEMSTATS, ALLSTATS và LAST

## 10. Hiện thị bảng kế hoạch : nâng cao

```

select plan_table_output from table(DBMS_XPLAN.DISPLAY(null,null,'ADVANCED
-PROJECTION -PREDICATE -ALIAS'));
Plan hash value: 3956160932

-----
| Id | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-----
| 0  | SELECT STATEMENT   |      |    1  |    37 |    3   (0)| 00:00:01 |
| 1  | TABLE ACCESS FULL| EMP  |    1  |    37 |    3   (0)| 00:00:01 |
-----

Outline Data
-----
/*+
  BEGIN_OUTLINE_DATA
  FULL(@"SEL$1" "EMP"@"SEL$1")
  OUTLINE_LEAF(@"SEL$1")
  ALL_ROWS
  DB_VERSION('11.1.0.6')
  OPTIMIZER_FEATURES_ENABLE('11.1.0.6')
  IGNORE_OPTIM_EMBEDDED_HINTS
  END_OUTLINE_DATA
*/

```

Định dạng ADVANCED(cao cấp) chỉ có từ Oracle Database 10g, phiên bản 2 và các phiên bản sau.

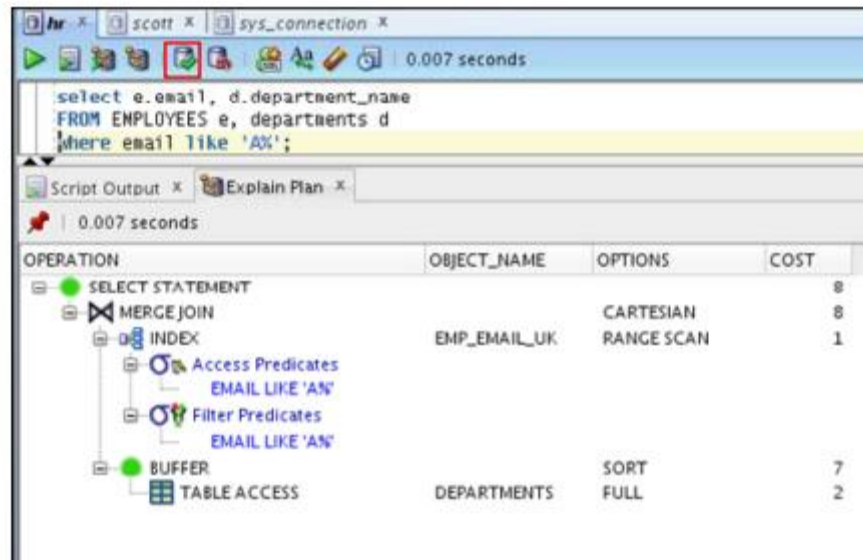
Định dạng đầu ra này bao gồm tất cả các tùy chọn từ định dạng ALL cộng với dữ liệu phác thảo chứa một tập hợp các gợi ý để sản xuất lại các kế hoạch nhất định.

Phần này có thể hữu ích nếu bạn muốn tạo một kế hoạch thực thi cụ thể ở một môi trường khác nhau.

Phần này cũng được hiển thị trong các tập tin lưu vết cho sự kiện 10053.

## 11. Kế hoạch giải thích sử dụng SQL Developer

### Explain Plan Using SQL Developer



Nháy biểu tượng Kế hoạch thực thi để tạo ra các kế hoạch thực thi, bạn có thể thấy trong thanh Giải thích. Một kế hoạch thực thi biểu diễn một cây thực thi với hệ thống phân cấp của các phép toán hình thành nên câu lệnh. Đối với mỗi phép toán, nó cho thấy thứ tự tham chiếu các bảng của câu lệnh SQL, cũng như phương pháp truy cập vào mỗi bảng, phương pháp kết nối các bảng chịu tác động bởi phép toán kết nối trong câu truy vấn, và phép toán xử lý dữ liệu như bộ lọc, sắp xếp, hoặc phép toán tập hợp. Ngoài cây thực thi, bảng kế hoạch đưa ra thông tin về tối ưu hóa (như chi phí của mỗi phép toán), phân vùng (chẳng hạn như tập các vùng đã truy cập), và thực hiện song song (như phương pháp phân phối đầu vào của phép kết nối).

## 12. Autotrace

- Khi chạy câu lệnh SQL bằng công cụ SQL\*Plus hoặc SQL Developer, có thể tự động lấy báo cáo về kế hoạch thực thi (execution plan) và các thông kê về câu lệnh đang thực thi. Thông báo là được sản sinh sau khi thực thi thành công các câu lệnh SQL DML( đó là SELECT, DELETE, UPDATE, INSERT). Nó là công dụng trong việc đánh giá và nâng cao hiệu năng thực thi của các câu lệnh SQL.

- Để sử dụng chức năng này, cần phải để 'PLAN\_TABLE' ở chế độ sẵn sàng trong lược đồ hiện tại, đồng thời trong ca làm việc hiện tại người dùng phải được cấp quyền 'PLUSTRACE'. Cần có quyền của người quản trị cơ sở dữ liệu (DBA) để có thể xin quyền 'PLUSTRACE'. Quyền 'PLUSTRACE' được tạo và gán cho DBA bằng cách chạy mã thực thi \$ORACLE\_HOME/sqlplus/admin/plustrace.sql .

- Trong nhiều phiên bản và nền tảng, mã thực thi trên là được chạy tại thời điểm tạo cơ sở dữ liệu. Nếu nó là không được thực thi trên nền tảng của bạn thì bạn có thể kết nối tới hệ quản trị cơ sở dữ liệu dưới tài khoản quản trị (SYSDBA) và chạy tệp thực thi plustrace.sql .

- Đặc quyền 'PLUSTRACE' chứa các quyền lựa chọn trên ba khung nhìn 'V\$'. Các quyền này là cần thiết được sản sinh khi thực hiện thống kê bằng 'AUTOTRACE'.

- 'AUTOTRACE' là công cụ vô cùng hữu ích trong việc đánh giá và nâng cao hiệu năng các câu lệnh SQL. Do sự rõ ràng trong việc khai báo nên nó dễ dàng sử dụng hơn so với 'EXPLAIN PLAN'.

Chú ý: Hệ quản trị không hỗ trợ thực hiện chức năng 'EXPLAIN PLAN' cho các câu lệnh thực hiện chuyển đổi kiểu dữ liệu ngầm cho các biến mờ. Nên đối với các biến mờ nói chung thì đầu ra của 'EXPLAIN PLAN' không biểu diễn kế hoạch thực thi thực tế được sử dụng.

### 13. Cú pháp của lệnh 'AUTOTRACE'

- Có thể bật chế độ 'AUTOTRACE' bằng nhiều cách tùy theo mục đích sử dụng với mẫu chung như sau:       SET AUTOTRACE 'option'

- Một số tùy chọn như:

+ 'OFF' : tắt chế độ tự động lưu vết câu lệnh SQL.

+ 'ON' : bật chế độ tự động lưu vết câu lệnh SQL.

+ 'TRACE' hoặc 'TRACE [ONLY]' : bật chế độ tự động lưu vết câu lệnh SQL và ngăn không hiển thị kết quả của câu lệnh.

+ 'EXPLAIN': Hiển thị kế hoạch thực thi nhưng không hiển thị các thống kê về câu lệnh.

+ 'STATISTICS': Hiển thị thống kê chỉ số nhưng không hiển thị kế hoạch thực thi.

Chú ý: Nếu cả hai tùy chọn 'EXPLAIN' và 'STATISTICS' bị bỏ qua, thì cả thống kê và kế hoạch thực thi được hiển thị theo mặc định.

### 14. Một số ví dụ về 'AUTOTRACE'

- Bạn có thể điều khiển thông báo bằng cách thay đổi một số thông số của biến hệ thống 'AUTOTRACE'. Ví dụ:

+ SET AUTOTRACE ON : Thông báo đưa ra bởi 'AUTOTRACE' bao gồm cả kế hoạch thực thi tối ưu và các thống kê về việc thực thi câu lệnh SQL.

+ SET AUTOTRACE TRACEONLY EXPLAIN : Thông báo đưa ra bởi 'AUTOTRACE' đưa ra chiến lược thực thi tối ưu nhưng không thực thi câu lệnh.

+ SET AUTOTRACE ON STATISTICS : Thông báo đưa ra bởi 'AUTOTRACE' hiển thị các thống kê cũng như các dòng chịu tác động trong việc thực thi câu lệnh SQL.

+ SET AUTOTRACE TRACEONLY : câu lệnh có tác dụng tương tự như SET AUTOTRACE ON, nhưng nó bỏ qua kết quả truy vấn nếu có. Nếu trường 'STATISTICS' là được mở thì dữ liệu truy vấn vẫn được lấy nhưng không hiển thị ra.

+ SET AUTOTRACE OFF: Thông báo bởi 'AUTOTRACE' không được sinh ra, đây là cài đặt mặc định.

## 15. AUTOTRACE: Thống kê

- Các thống kê đưa ra bởi 'AUTOTRACE' là các bản ghi của máy chủ khi câu truy vấn của bạn được thực thi, nó đưa ra chi tiết các loại tài nguyên trên máy chủ cũng như số lượng của chúng cần thiết để thực thi câu truy vấn. Kết quả đưa ra bao gồm một số thông số sau:

+ 'recursive calls' : là số lượng lời gọi đệ qui được sản sinh ra cả mức người dùng và hệ thống. Cơ sở dữ liệu Oracle lưu trữ các bảng tạm được sử dụng trong các phép xử lý nội bộ. Khi Cơ sở dữ liệu Oracle cần thay thế các bảng này, nó nội sinh một lệnh SQL ngầm, và lệnh ngầm này sẽ sinh ra một lời gọi đệ qui.

+ 'db block gets' thời gian cần thiết để một khối dữ liệu kết quả được lấy về.

+ 'consistent gets' thời gian cần thiết để một lệnh đọc thích hợp cho một khối được hoàn thành.

+ 'physical reads' tổng số lượng khối dữ liệu đọc từ đĩa. Số lượng này bằng với giá trị của các khối phải đọc trực tiếp trên đĩa cộng với các khối đã nằm trong bộ nhớ đệm.

+ 'redo size' tổng số lượng byte được sản sinh để chuyển về trạng thái cũ từ trạng thái hiện tại.

+ 'bytes sent via SQL\*Net to client' : tổng số byte nhận được từ người dùng thông qua Oracle Net.

+ 'SQL\*Net roundtrips to/from client': tổng số lượng thông điệp Oracle Net gửi và nhận từ người dùng.

Chú ý: Các thống kê đưa ra bởi 'AUTOTRACE' lấy từ 'V\$SESSTAT'.

+ 'sort (memory)': số lượng phép toán sắp xếp được thực hiện thành hiện trong bộ nhớ và không đòi hỏi thêm các phép ghi đĩa.

+ 'sort (disk)': số lượng các phép toán sắp xếp đòi hỏi tối thiểu một phép ghi trên đĩa.

+ 'rows processed': số lượng hàng cần xử lý trong quá trình thực thi truy vấn.

Người dùng được phép xem các thống kê trên công cụ SQL\*PLUS. Oracle Net lưu trữ quá trình xử lý và trao đổi tổng thể giữa công cụ SQL\*PLUS và máy chủ, bất kể là Oracle Net có được cài đặt trên máy người dùng hay không. Bạn không thể thay đổi các khuôn dạng mặc định của báo cáo thống kê.

Chú ý: 'db block gets' là xác định cho các khối dữ liệu được đọc từ cơ sở dữ liệu tại thời điểm hiện tại. 'consistent gets' là xác định cho việc đọc các khối dữ liệu thỏa mãn một hệ thống đối số đặc biệt. 'physical reads' chỉ tới các khối dữ liệu được đọc từ đĩa. 'db block gets' và 'consistent gets' là hai chỉ số thống kê thường được đánh giá. Các chỉ số này nên có giá trị thấp so với với số lượng dòng lấy về. Các phép sắp xếp nên được thực hiện trong bộ nhớ hơn trên đĩa.

## 16. AUTOTRACE: Sử dụng công cụ SQL Developer

**AUTOTRACE Using SQL Developer**

OPERATION	OBJECT_NAME	COST	LAST_CR_BUFFER_GETS
SELECT STATEMENT		8	
MERGE JOIN CARTESIAN		8	8
INDEX RANGE SCAN	EMP_EMAIL_UK	1	1
Access Predicates			
EMAIL LIKE 'A%'			
Filter Predicates			

V\$STATNAME Name	V\$MYSTAT Value
recursive calls	1
db block gets	0
consistent gets	8
physical reads	0
redo size	0
bytes sent via SQL*Net to client	1872
bytes received via SQL*Net from client	658
SQL*Net roundtrips to/from client	2
sorts (memory)	2
sorts (disk)	0

- Khung nhìn Autotrace hiển thị các thông tin được lưu vết khi bạn thực hiện câu truy vấn SQL bằng cách chọn biểu tượng Autotrace. Các thông tin này có thể giúp bạn xác định các câu truy vấn SQL có thể cải thiện hiệu năng.

## 17. Sử dụng khung nhìn 'V\$SQL\_PLAN'

- Đây là khung nhìn hiển thị các kế hoạch thực thi cho các con trỏ vẫn nằm trong thư viện đệm. Các thông tin trong khung nhìn này rất giống với thông tin trong 'PLAN\_TABLE'. Nhưng 'V\$SQL\_PLAN' chứa các kế hoạch thực thi đã thực sự được thực hiện. Các kế hoạch thực thi lấy được từ 'EXPLAIN\_PLAN' có thể khác so với các kế hoạch thực thi được thực hiện trên con trỏ. Bởi vì con trỏ có thể đã được biên dịch với một giá trị biến phiên làm việc khác hoặc có chức các biến mù.

- 'V\$SQL\_PLAN' hiển thị kế hoạch cho một con trỏ thay vì cho tất cả các con trỏ liên hệ với câu lệnh SQL. Một câu lệnh SQL có thể có nhiều hơn một con trỏ liên hệ với nó, từng con trỏ có xác định bởi thông số 'CHILD\_NUMBER'. Ví dụ, cùng một câu truy vấn được thực thi bởi những người dùng khác nhau có thể có các con trỏ khác nhau liên hệ với nó nếu đối tượng được tham chiếu nằm trong các lược đồ khác nhau. Tương tự, sự khác biệt về các thông số khác có thể dẫn đến số lượng con trỏ khác nhau cho cùng một câu truy vấn. Bảng 'V\$SQL\_PLAN' có thể dùng để quan sát các kế hoạch khác nhau cho các con trỏ khác nhau của cùng một câu truy vấn.

Chú ý: Một khung nhìn hữu ích khác là 'V\$SQL\_PLAN\_STATISTICS', nó cung cấp các thống kê cho từng phép toán trong kế hoạch thực thi của từng con trỏ được lưu trong bộ đệm. Hoặc 'V\$SQL\_PLAN\_STATISTICS\_ALL' chứa thông tin kết nối từ 'V\$SQL\_PLAN' cùng với thông tin thống kê từ 'V\$SQL\_PLAN\_STATISTICS' và 'V\$SQL\_WORKAREA'.

## 18. Một số cột trong 'V\$SQL\_PLAN'

- HASH\_VALUE: giá trị băm của câu lệnh cha trong thư viện đệm.
- ADDRESS: địa chỉ của câu lệnh cha của con trỏ hiện tại.
- CHILD\_NUMBER: Chỉ số của con trỏ hiện tại trong kế hoạch thực thi.
- POSITION: Số thứ tự gọi thực thi cho tất cả các phép toán có cùng giá trị PARENT\_ID.
- PARENT\_ID: mã ID của bước thực thi tiếp theo có phép toán trên đầu ra của bước hiện tại.
- ID: Số hiệu gán cho từng bước thực thi trong kế hoạch thực thi.
- PLAN\_HASH\_VALUE: Số hiệu của kế hoạch SQL chứa con trỏ hiện tại.

Các cột trong PLAN\_TABLE có cùng giá trị là ADDRESS và HASH\_VALUE. Hai cột này có thể sử dụng để kết nối với 'V\$SQLAREA' để lấy thêm các thông tin quan trọng khác liên quan với con trỏ. Hoặc hai cột trên cùng với cột CHILD\_NUMBER có thể dùng để kết nối với 'V\$SQL' để lấy thêm các thông tin quan trọng của các con trỏ con cháu. Hoặc bằng cách kết hợp cột 'PLAN\_HASH\_VALUE' với các cột hoặc bảng khác, bạn có thể dễ dàng xác định hai con trỏ có cùng kế hoạch hay không.

Chú ý: Từ phiên bản Oracle Database 10g, SQL\_HASH\_VALUE trong 'V\$SESSION' đã được thay thế với SQL\_ID, giá trị bạn có thể thu được từ nhiều khung nhìn V\$ khác. SQL\_HASH\_VALUE là một giá trị 32-bit không đủ lớn cho các dữ liệu AWR. SQL\_ID là một giá trị băm 64-bit với 32-bit cuối là SQL\_HASH\_VALUE. Nó được biểu diễn như một chuỗi các ký tự khiến cho việc quản lý nó dễ dàng hơn.

Khung nhìn 'V\$SQL\_PLAN\_STATISTICS': cung cấp các thông tin thống kê thực thi thực tế cho tất cả các phép toán trong kế hoạch, như số lượng dòng đưa ra, thời gian thực thi. Các thống kê này được tính cộng dồn ngoại trừ số lượng dòng đưa ra. Ví dụ, các thống kê cho phép toán kết hợp đã bao gồm thống kê cho hai đầu vào. Các thống kê trong 'V\$SQL\_PLAN\_STATISTICS' có

thể lấy về cho các con trỏ đã được biên dịch với tham số 'STATISTICS\_LEVEL' được khởi tạo giá trị ALL hoặc sử dụng gợi ý 'GATHER\_PLAN\_STATISTICS'.

Khung nhìn 'V\$SQL\_PLAN\_STATISTICS\_ALL' chứa các thống kê về sử dụng bộ nhớ cho các phép toán có sử dụng bộ nhớ chính của SQL như sắp xếp và kết nối băm. Các thông tin trong khung nhìn này là kết nối với 'V\$SQL\_PLAN' và các thống kê thực thi từ 'V\$SQL\_PLAN\_STATISTICS' và 'V\$SQL\_WORKAREA'

## 19. Kết nối giữa các khung nhìn thực thi động quan trọng

- 'V\$SQLAREA' hiển thị các thông tin thống kê trong vùng chia sẻ SQL và chứa một dòng mỗi xâu SQL. Nó cung cấp các thống kê cho các câu lệnh SQL trong bộ nhớ, đã được phân tích và đang sẵn sàng thực thi:

- + SQL\_ID là chỉ số SQL của con trỏ cha nằm trong thư viện đệm.

- + VERSION\_COUNT là số lượng con trỏ con đang nằm trong bộ đệm của con trỏ cha này.

- 'V\$SQL' chứa danh sách các thống kê trong vùng chia sẻ SQL và chứa một dòng cho từng con của văn bản SQL gốc đã nhập vào:

- + ADDRESS chứa địa chỉ con trỏ cha của con trỏ hiện tại.

- + HASH\_VALUE là giá trị của câu lệnh cha trong thư viện đệm.

- + SQL\_ID là chỉ số SQL của con trỏ cha nằm trong thư viện đệm.

- + PLAN\_HASH\_VALUE: Số hiệu của kế hoạch SQL chứa con trỏ hiện tại. Bằng cách so sánh giá trị PLAN\_HASH\_VALUE với một con trỏ khác, bạn có thể dễ dàng xác định hai kế hoạch có giống nhau hay không.

- + CHILD\_NUMBER là chỉ số của con trỏ con hiện tại.

Thống kê hiển thị trong 'V\$SQL' thông thường được cập nhật tại thời điểm kết thúc thực thi câu truy vấn. Nhưng trong trường hợp những câu truy vấn tốn nhiều thời gian để chạy thì các giá trị sẽ được cập nhật sau 5 giây. Điều này cho phép dễ dàng quan sát các tác động trên những câu truy vấn tốn nhiều thời gian trong khi chúng vẫn trong quá trình xử lý.

- 'V\$SQL\_PLAN' chứa thông tin kế hoạch thực thi cho từng con trỏ con được tải vào trong thư viện đệm. Các cột ADDRESS, HASH\_VALUE, và CHILD\_NUMBER có thể sử dụng để kết nối với 'V\$SQL' để lấy thêm các thông tin xác định về con trỏ.

- 'V\$SQL\_PLAN\_STATISTICS' cung cấp các thống kê thực thi ở mức dòng cho từng con trỏ con. Các cột ADDRESS và HASH\_VALUE có thể dùng để kết nối với 'V\$SQLAREA' để lấy ra vị trí của con trỏ cha. Các cột ADDRESS, HASH\_VALUE, và CHILD\_NUMBER có thể sử dụng để kết nối với 'V\$SQL' để lấy ra vị trí của con trỏ con dùng trong khu vực này.



- 'V\$SQL\_PLAN\_STATISTICS\_ALL' chứa các thống kê về sử dụng bộ nhớ cho các tài nguyên dòng cần sử dụng bộ nhớ SQL (như sắp xếp hoặc kết nối băm). Các thông tin trong khung nhìn này là kết nối với các thống kê thực thi từ 'V\$SQL\_PLAN\_STATISTICS' và 'V\$SQL\_WORKAREA'.

- 'V\$SQL\_WORKAREA' hiển thị thông tin về các vùng làm việc sử dụng bởi các con trỏ SQL. Mỗi câu lệnh SQL trong vùng chia sẽ có một hoặc nhiều con trỏ con được liệt kê trong khung nhìn 'V\$SQL'. 'V\$SQL\_WORKAREA' liệt kê tất cả các vùng làm việc cần thiết bởi các con trỏ con này. 'V\$SQL\_WORKAREA' có thể kết nối với 'V\$SQLAREA' thông qua các cột (ADDRESS, HASH\_VALUE) và kết nối với 'V\$SQL' thông qua các cột (ADDRESS, HASH\_VALUE, CHILD\_NUMBER). Bạn có thể sử dụng khung nhìn này để trả lời cho các câu hỏi:

+ Tập 10 vùng làm việc nào được yêu cầu nhiều nhất trong các vùng đệm.

+ Với các vùng làm việc được cấp phát trong chế độ 'AUTO' thì bao nhiêu phần trăm các vùng làm việc sử dụng tối đa bộ nhớ.

- 'V\$SQLSTATS' hiển thị các thống kê hiệu năng cơ bản cho các con trỏ SQL, với từng dòng biểu diễn dữ liệu cho một kết hợp xác định của văn bản SQL và kế hoạch tối ưu (đó là kết hợp xác định SQL\_ID và PLAN\_HASH\_VALUE). Các cột định nghĩa trong 'V\$SQLSTATS' là tương đồng với các cột trong V\$SQL và V\$SQLAREA. Nhưng V\$SQLSTATS khác với V\$SQL và V\$SQLAREA ở điểm là nó nhanh hơn, có khả năng mở rộng cao hơn và duy trì một lượng dữ liệu lớn hơn (các thống kê có thể vẫn nằm trong khung nhìn mặc dù các con trỏ đã ra khỏi vùng chia sẽ). Chú ý là 'V\$SQLSTATS' chứa một tập con các cột xuất hiện trong V\$SQL và V\$SQLAREA.

## 20. Truy vấn 'V\$SQL\_PLAN'

- Bạn có thể truy vấn 'V\$SQL\_PLAN' sử dụng hàm DBMS\_XPLAN.DISPLAY\_CURSOR() để hiển thị các truy vấn hiện tại hoặc trước đó. Bạn có thể truyền giá trị SQL\_ID cho câu truy vấn như một tham số để thu lại kế hoạch thực thi cho câu truy vấn đã đưa. SQL\_ID là chỉ số của câu truy vấn SQL trong con trỏ bộ đệm. Bạn có thể lấy giá trị chính xác bằng cách truy vấn cột SQL\_ID trong 'V\$SQL' hoặc 'V\$SQLAREA'. Tùy chọn, bạn có thể chọn cột PREV\_SQL\_ID cho một phiên làm việc xác định trong 'V\$SESSION'. Tham số này mặc định là rỗng trong trường hợp kế hoạch của con trỏ thực thi cuối cùng trong phiên đã được hiển thị.

- CHILD\_NUMBER là chỉ số con của con trỏ cần hiển thị. Nếu không xác định thì tất cả các kế hoạch thực thi của những con trỏ có tham số SQL\_ID trùng sẽ được hiển thị. CHILD\_NUMBER chỉ có thể xác định khi SQL\_ID là xác định.

- Tham số FORMAT điều khiển mức độ chi tiết của kế hoạch. Thêm vào các giá trị chuẩn (BASIC, TYPICAL, SERIAL, ALL, and ADVANCED), có một số giá trị hỗ trợ thêm vào để hiển thị các thống kê thời gian chạy cho con trỏ.

+ IOSTATS: Giả sử các thống kê kế hoạch cơ bản là được tập hợp sau khi các câu truy vấn đã được thực thi (bằng cách hoặc dùng gợi ý 'GATHER\_PLAN\_STATISTICS' hoặc chỉnh sửa tham số 'statistics\_level' thành giá trị ALL), khuôn dạng này hiển thị các thống kê vào ra cho tất cả (hoặc cuối cùng) các thực thi của con trỏ.

+ MEMSTATS: Giả sử chương trình quản lý bộ nhớ toàn cục đang chạy (kiểm tra bằng cách xét tham số 'pga\_aggregate\_target' có giá trị khác 0 hay không), khuôn dạng này cho phép hiển thị các thống kê về quản lý bộ nhớ (ví dụ: chế độ thực thi của các phép toán, bao nhiêu phần trăm bộ nhớ đã được sử dụng, số lượng byte thực hiện truy vấn trên đĩa, .v.v). Các thống kê này chỉ áp dụng cho các toán tử có tác động lên bộ nhớ mạnh như phép sắp xếp, phép băm kết hợp.

+ ALLSTATS: bao gồm cả 'IOSTATS' và 'MEMSTATS'

+ LAST: Mặc định các thống kê kế hoạch là hiển thị cho tất cả các thực thi của con trỏ. Từ khóa LAST có thể xác định chỉ xem duy nhất các thống kê của phép thực thi cuối cùng.

## 21. Kho chứa lịch sử công việc tự động (AWR)

- AWR là một phần của nền tảng thông minh được giới thiệu trong hệ quản trị cơ sở dữ liệu Oracle 10g. Nền tảng này được sử dụng bởi nhiều thành phần khác nhau, như hệ thống tự động kiểm soát lỗi (ADDM). AWR tự động tập hợp, xử lý và duy trì các thống kê thao tác thực thi của hệ thống cho việc phát hiện vấn đề và mục đích tự nâng cao hiệu năng truy vấn, nó lưu trữ các thông tin này dài hạn trong cơ sở dữ liệu.

- Các thông tin thống kê được tập hợp và xử lý bởi AWR bao gồm:

+ Các đối tượng có truy xuất hoặc sử dụng các thông tin thống kê của các phân đoạn cơ sở dữ liệu.

+ Mô hình thời gian dựa trên thời gian sử dụng cho các hoạt động, biểu diễn trong các khung nhìn 'V\$SYS\_TIME\_MODEL' và 'V\$SESS\_TIME\_MODEL'.

+ Một số thống kê hệ thống và phiên làm việc trong các khung nhìn 'V\$SYSSTAT' và 'V\$SESSTAT'.

+ Các câu truy vấn SQL sử dụng nhiều tài nguyên của hệ thống được đánh giá dựa trên các tiêu chí như thời gian thực thi, thời gian sử dụng CPU, dung lượng bộ nhớ đệm sử dụng, .v.v.

+ Các thống kê ASH, biểu diễn lịch sử của các phiên làm việc gần đây.

- Cơ sở dữ liệu tự động sản sinh các bản sao trạng thái sử dụng của cơ sở dữ liệu mỗi giờ và tập hợp các thống kê vào trong Kho chứa lịch sử công việc. Dữ liệu trong các bản sao sẽ được phân tích bởi ADDM. ADDM so sánh sự khác biệt giữa các bản sao để xác định sự ảnh hưởng của các câu truy vấn tới hiệu năng tải của hệ thống. Điều này sẽ cho phép giảm số lượng câu truy vấn SQL cần phải phân tích theo thời gian.

Chú ý: Bằng cách sử dụng các gói PL/SQL, như 'DBMS\_WORKLOAD\_REPOSITORY' hoặc Oracle Enterprise Manager, bạn có thể quản lý tần suất và thời gian duy trì các câu truy vấn SQL được lưu trong AWR.

## 22. Quản lý AWR với PL/SQL

+ Mặc dù giao diện nguyên thủy cho quản lý AWR là Enterprise Manager, nhưng các chức năng giám sát cũng có thể quản lý với các thủ tục trong gói DBMS\_WORKLOAD\_REPOSITORY.

+ Các bản sao là tự động sản sinh cho một cơ sở dữ liệu Oracle, nhưng bạn có thể sử dụng các thủ tục của DBMS\_WORKLOAD\_REPOSITORY để tự thao tác việc tạo, lưu, chỉnh sửa các bản sao và thông tin cơ sở được sử dụng bởi ADDM. Các bản sao và thông tin cơ sở là một tập các lịch sử trạng thái dữ liệu trong một khoảng thời gian xác định được sử dụng để so sánh hiệu năng truy suất. Để sử dụng các thủ tục này người dùng cần có quyền gọi DBA.

+ Tạo bản sao: bạn có thể tự tạo các bản sao với thủ tục CREATE\_SNAPSHOT nếu bạn muốn bắt các thông kê tại một thời khác so với các bản sao sản sinh tự động theo chu kỳ. Ví dụ:

```
Exec DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT ('ALL');
```

Trong ví dụ trên, một bản sao cho trạng thái hiện tại được tạo ngay lập tức với một mức độ quét xác định được để mặc định là 'tiêu chuẩn'. Bạn có thể xem bản sao trong khung nhìn 'DBA\_HIST\_SNAPSHOT'.

+ Lưu bản sao: bạn có thể lưu một khoảng bản sao sử dụng thủ tục DROP\_SNAPSHOT\_RANGE. Để quan sát một danh sách mã ID của các bản sao cùng với mã ID của cơ sở dữ liệu, kiểm tra khung nhìn 'DBA\_HIST\_SNAPSHOT'. Ví dụ:

```
Exec DBMS_WORKLOAD_REPOSITORY.DROP_SNAPSHOT_RANGE -  
(low_snap_id => 22, high_snap_id => 32, dbid => 3310949047);
```

Trong ví dụ trên, khoảng bản sao có mã ID xác định từ 22 đến 32 được lưu lại. Mã ID xác định của cơ sở dữ liệu là 3310949047. Nếu bạn không chỉ rõ giá trị cho dbid thì mã định danh của cơ sở dữ liệu cụ thể được sử dụng như giá trị mặc định. Dữ liệu ASH thuộc về một khoảng thời gian xác định bằng một dải các bản sao được lọc khi thủ tục 'DROP\_SNAPSHOT\_RANGE' được gọi.

+ Chỉnh sửa các cài đặt bản sao: Bạn có thể chỉnh sửa khoảng thời gian giữa hai lần tạo bản sao, cũng như thời gian duy trì bản sao cho một cơ sở dữ liệu xác định. Nhưng chú ý rằng điều này có thể ảnh hưởng tới độ chính xác của các công cụ chuẩn đoán Oracle. Ví dụ:

```
Exec DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS( -  
retention => 43200, interval => 30, dbid => 3310949047);
```

Trong ví dụ trên, thời gian lưu trữ bản sao được hiệu chỉnh thành 43200 phút (tương đương 30 ngày), và khoảng thời gian tự động sinh bản sao được chỉnh sửa thành 30 phút. Nếu giá trị NULL được xác định thì giá trị hiện tại của đại lượng tương ứng được giữ nguyên, các thay đổi trên tác động trên cơ sở dữ liệu có định danh là 3310949047. Nếu bạn không chỉ rõ giá trị cho dbid thì mã

định danh của cơ sở dữ liệu cụ bộ được sử dụng như giá trị mặc định. Bạn có thể kiểm tra các cài đặt mặc định cho cơ sở dữ liệu của bạn thông qua khung nhìn DBA\_HIST\_WR\_CONTROL.

### 23. Các khung nhìn quan trọng của AWR

+ Bạn có thể xem dữ liệu AWR trên màn hình Oracle Enterprise Manager hoặc trong các báo cáo của AWR. Nhưng bạn có thể xem các thông tin thống kê trực tiếp từ các khung nhìn sau:

- 'V\$ACTIVE\_SESSION\_HISTORY': Khung nhìn hiển thị hoạt động của các phiên làm việc trên cơ sở dữ liệu, các mẫu được lấy mỗi giây.

Các khung nhìn V\$ metric (Khung nhìn định lượng) cung cấp các thông tin định lượng để tính độ hiệu năng của hệ thống. Khung nhìn định lượng được tổ chức trong nhiều nhóm khác nhau như sự kiện, lớp sự kiện, hệ thống, phiên làm việc, dịch vụ, tệp tin và các bảng định lượng. Các nhóm này được xác định trong khung nhìn 'V\$METRICGROUP'.

Khung nhìn DBA\_HIST chứa lịch sử dữ liệu lưu trong cơ sở dữ liệu. Nhóm khung nhìn này bao gồm:

- 'DBA\_HIST\_ACTIVE\_SESS\_HISTORY' hiển thị lịch sử nội dung của các mẫu phiên lịch sử hoạt động trong bộ nhớ ghi lại các hoạt động gần đây của hệ thống.

- 'DBA\_HIST\_BASELINE' hiển thị các thông tin cơ sở đã bắt trong hệ thống.

- 'DBA\_HIST\_DATABASE\_INSTANCE' hiển thị các thông tin về môi trường cơ sở dữ liệu.

- 'DBA\_HIST\_SNAPSHOT' hiển thị thông tin về các bản sao trong hệ thống.

- 'DBA\_HIST\_SQL\_PLAN' hiển thị các kế hoạch thực thi SQL.

- 'DBA\_HIST\_WR\_CONTROL' hiển thị các cài đặt cho việc điều khiển AWR.

### 24. Truy vấn AWR

+ Bạn có thể sử dụng hàm DBMS\_XPLAN.DISPLAY\_AWR() để hiển thị tất cả các kế hoạch được lưu trong AWR. Ví dụ:

1. Thực hiện truy vấn SQL:

```
SQL> select /* example */ * from hr.employees natural
      join hr.departments;
```

2. Truy vấn khung nhìn 'V\$SQL\_TEXT' để lấy SQL\_ID:

```
SQL> select sql_id, sql_text from v$SQL
      where sql_text
```

```
like '%example%';
```

3. Sử dụng SQL\_ID, kiểm tra xem câu lệnh SQL trên đã được bắt lại trong khung nhìn DBA\_HIST\_SQLTEXT hay chưa. Nếu câu truy vấn không trả lại dòng nào tức là câu truy vấn SQL trước chưa được nạp vào trong AWR.

```
SQL> SELECT SQL_ID, SQL_TEXT FROM dba_hist_sqltext WHERE  
SQL_ID =' 454rug2yva18w';
```

```
no rows selected
```

Bạn có thể tự tạo bản sao AWR thay vì chờ đến lần tạo bản sao tự động sau. Sau đó kiểm tra lại xem câu truy vấn có trong DBA\_HIST\_SQLTEXT hay chưa.

```
SQL> exec dbms_workload_repository.create_snapshot;
```

```
PL/SQL procedure successfully completed.
```

```
SQL> SELECT SQL_ID, SQL_TEXT FROM dba_hist_sqltext WHERE  
SQL_ID =' 454rug2yva18w';
```

```
SQL_ID
```

```
SQL_TEXT
```

```
-----  
454rug2yva18w select /* example */ * from ...
```

4. Sử dụng hàm DBMS\_XPLAN.DISPLAY\_AWR () để nhận về kế hoạch thực thi

```
SQL>SELECT PLAN_TABLE_OUTPUT FROM  
TABLE(DBMS_XPLAN.DISPLAY_AWR('454rug2yva18w'));
```

Trong ví dụ trên, bạn đã truyền SQL\_ID như một tham số. SQL\_ID là mã định danh của câu truy vấn SQL trong con trỏ bộ đệm. Hàm DISPLAY\_AWR() sử dụng các thông số PLAN\_HASH\_VALUE, DB\_ID, và FORMAT.

## 25. Sản sinh báo cáo SQL từ dữ liệu AWR

+ Từ hệ quản trị cơ sở dữ liệu Oracle 10g, phiên bản 2, đã có khả năng tự sinh báo cáo SQL từ dữ liệu AWR, cơ bản là tương đương với sqrepsql.sql với Statspack. Báo cáo AWR SQL có thể tự sinh bằng cách gọi tệp mã nguồn \$ORACLE\_HOME/rdbms/admin/awrsqrpt.sql.

+ Bạn có thể hiển thị các thông tin kế hoạch trong AWR bằng cách sử dụng bảng chức năng display\_awr trong gói dbms\_xplan PL/SQL. Ví dụ, để hiển thị thông tin cho một SQL\_ID trong AWR:

```
select * from table(dbms_xplan.display_awr('dvza55c7zu0yv'));
```

Bạn có thể lấy giá trị chính xác cho một câu lệnh SQL mong muốn bằng cách truy vấn SQL\_ID trong cột DBA\_HIST\_SQLTEXT.

## 26. SQL Monitoring: Tổng quan

Các tính năng giám sát SQL được kích hoạt theo mặc định khi khởi tạo STATISTICS\_LEVEL. Thông số này được thiết lập là ALL hoặc TYPICAL (giá trị mặc định).

Ngoài ra, các tham số CONTROL\_MANAGEMENT\_PACK\_ACCESS phải được thiết lập là DIAGNOSTIC+TUNING (giá trị mặc định), vì giám sát SQL là một tính năng của gói điều chỉnh cơ sở dữ liệu của Oracle.

Theo mặc định, giám sát SQL sẽ tự động bắt đầu khi một câu lệnh SQL chạy song song, hoặc khi nó đã chạy ít nhất năm giây trên CPU hoặc I/O lần trong một thực thi duy nhất.

Như đã đề cập, giám sát SQL được kích hoạt theo mặc định. Tuy nhiên, vẫn tồn tại hai cấp gọi ý có sẵn để thực hiện hoặc ngăn chặn một câu lệnh SQL đang được theo dõi. Để thực hiện giám sát SQL, sử dụng lệnh MONITOR. Để ngăn chặn các câu lệnh SQL, sử dụng lệnh NO\_MONITOR. Bạn có thể theo dõi các số liệu thống kê để thực hiện câu lệnh SQL bằng cách sử dụng thông số V\$SQL\_MONITOR và V\$SQL\_PLAN\_MONITOR.

Sau khi theo dõi được bắt đầu, một mục được thêm vào hiệu suất động V\$SQL\_MONITOR. Mục này theo dõi số liệu hiệu suất thu được để thực hiện, bao gồm cả thời gian trôi qua, thời gian CPU, số lần đọc và viết, I/O thời gian chờ đợi, và nhiều thời gian chờ đợi khác. Các thống kê này được làm mới gần như thời gian thực tế, thường là một giây.

Sau khi kết thúc, thông tin giám sát không bị xóa ngay lập tức, nó được lưu vào V\$SQL\_MONITOR ít nhất một phút. Thông tin được xóa để bộ nhớ có thể được khôi phục khi báo cáo mới được theo dõi.

Các biến V\$SQL\_PLAN\_MONITOR và V\$SQL\_MONITOR có thể được sử dụng kết hợp với các biến sau đây: V\$SQL, V\$SQL\_PLAN, V\$ACTIVE\_SESSION\_HISTORY, V\$session\_longops, và V\$SESSION. Hoặc bạn có thể sử dụng các báo cáo giám sát SQL để xem dữ liệu giám sát SQL. Các báo cáo giám sát SQL cũng có sẵn trong một phiên bản GUI qua Enterprise Manager và SQL Developer.

## 27. Ví Dụ SQL Monitoring Report:

```
SQL> set long 10000000
SQL> set longchunksiz 10000000
SQL> set linesize 200
SQL> select dbms_sqltune.report_sql_monitor from dual;

SQL Monitoring Report

SQL Text
-----
select count(*) from sales

Global Information
Status          : EXECUTING
Instance ID     : 1
Session ID      : 125
SQL ID          : fazrk33ng71km
SQL Execution ID : 16777216
Plan Hash Value  : 1047182207
Execution Started : 02/19/2008 21:01:18
First Refresh Time : 02/19/2008 21:01:22
Last Refresh Time : 02/19/2008 21:01:42

-----
| Elapsed | Cpu | IO | Other | Buffer | Reads |
| Time(s) | Time(s) | Waits(s) | Waits(s) | Gets |      |
-----
| 22 | 3.36 | 0.01 | 19 | 259K | 199K |
-----
```

In a different session

```
SQL> select count(*) from sales;
```

Giả sử ta SELECT từ SALES từ một phiên khác so với phiên bản sử dụng để in các báo cáo giám sát SQL. Hàm DBMS\_SQLTUNE.REPORT\_SQL\_MONITOR chấp nhận một số tham số đầu vào để xác định việc thực thi, mức độ chi tiết trong báo cáo, và các loại báo cáo (TEXT, HTML, XML ). Một báo cáo được tạo ra nếu không có các tham số như ví dụ trong hình. Trong khi SELECT, ta in các báo cáo giám sát SQL từ một phiên thứ hai. Từ báo cáo, bạn có thể thấy rằng các câu lệnh SELECT hiện thực.

Phần thông tin toàn cục cung cấp cho bạn một số thông tin quan trọng:

Để xác định cách thực thi duy nhất cho câu lệnh SQL. Key thực thi này bao gồm ba thuộc tính, mỗi thuộc tính tương ứng với một cột trong V\$SQL\_MONITOR:

- Id SQL để xác định các câu lệnh SQL (SQL\_ID).
- Nhận diện được tạo ra trong nội bộ để đảm bảo rằng khoá chính này là duy nhất (SQL\_EXEC\_ID).
- Mốc thời gian bắt đầu thực hiện (SQL\_EXEC\_START)

Báo cáo cũng cho bạn thấy một số thống kê quan trọng.

Báo cáo sẽ hiển thị các cách thực hiện đang được sử dụng bởi cấp phép của bạn. Giám sát SQL cho phép bạn hiển thị các hoạt động đang thực hiện. Điều này cho phép bạn phát hiện các hoạt động tốn thời gian nhất, để bạn có thể tập trung phân tích của bạn vào bộ phận đấy. Các hoạt động chạy được đánh dấu bằng một mũi tên trong cột Id của báo cáo.

Cột thời gian hoạt động(s) cho thấy các hoạt động đã được kích hoạt bao lâu. Cột bắt đầu hoạt động cho thấy, trong vài giây, khi các hoạt động trong kế hoạch thực hiện bắt đầu liên quan đến các câu lệnh SQL thời gian thực hiện bắt đầu. Trong báo cáo này, các hoạt động truy cập đầy đủ vào bảng Id 2 được bắt đầu (+ 1s bắt đầu hoạt động) và chạy 23 giây cho lần đầu tiên.

Cột bắt đầu cho thấy số lần hoạt động trong kế hoạch thực hiện đã được thực thi. Cột các hàng (Actual) chỉ ra số hàng tổ chức, và cột các hàng (Estim) cho thấy ước tính từ các tối ưu. Các cột hoạt động (percent) và Hoạt động chi tiết (sample #) có nguồn gốc bằng cách tham gia các thông số V\$ACTIVE\_SESSION\_HISTORY và V\$SQL\_PLAN\_MONITOR. Hoạt động (percent) cho thấy tỷ lệ phần trăm của thời gian cơ sở dữ liệu sử dụng của từng hoạt động của kế hoạch thực hiện. Hoạt động chi tiết (sample #) cho thấy bản chất của hoạt động đó (chẳng hạn như CPU hoặc chờ đợi sự kiện).

Trong báo cáo này, các cột hoạt động chi tiết (sample #) cho thấy rằng hầu hết các cơ sở dữ liệu thời gian, 100% được tiêu thụ bởi các hoạt động Id 2 (TABLE ACCESS PULL of SALES). Cho đến nay, hoạt động này bao gồm 4 mẫu đặc trưng cho CPU. Cột cuối cùng, Progress cho thấy sự tiến bộ theo dõi thông tin cho các hoạt động từ thông số V\$session\_longops. Trong báo cáo này, nó cho thấy rằng, cho đến nay, các TABLE ACCESS PULL hoạt động là 74% hoàn tất. Cột này chỉ xuất hiện trong các báo cáo sau một thời gian nhất định.

Lưu ý: Không được hiển thị trong báo cáo này, các cột Memory và Temp chỉ ra số lượng bộ nhớ và không gian tạm thời tiêu thụ bởi hoạt động của kế hoạch thực hiện tương ứng.

## 28. Interpreting an Execution Plan(Giải thích một Kế hoạch thực thi)

Giải thích sản lượng kế hoạch là một đại diện của một cây nguồn hàng. Mỗi bước (đòng trong kế hoạch thực hiện hoặc nút trong cây) đại diện cho một nguồn hàng. Các kế hoạch giải thích các nút thụt để cho biết rằng chúng là con cái của cha mẹ ở trên nó. Thứ tự của các nút theo cha mẹ chỉ ra thứ tự thực hiện của các nút trong cấp đó. Nếu hai bước thụt vào cùng cấp, một trong những bước đầu tiên được thực hiện đầu tiên. Trong các định dạng cây, các lá ở bên trái mỗi mức của cây là nơi thực thi bắt đầu. Các bước của kế hoạch thực hiện không được thực hiện theo thứ tự mà chúng được đánh số. Có một mối quan hệ cha-con giữa các bước. Trong PLAN\_TABLE và V\$SQL\_PLAN, các yếu tố quan trọng để lấy lại cơ cấu cây là ID, PARENT\_ID, và các cột Vị trí. Trong một dấu vết tập tin, các cột tương ứng với id, pid, và các lĩnh vực pos, tương ứng. Một cách để đọc một kế hoạch thực hiện là chuyển đổi nó thành một đồ thị có một cấu trúc cây. Bạn có thể bắt đầu từ đầu, với id = 1, đó là nút gốc của cây. Tiếp theo, bạn phải tìm ra các hoạt động mà ăn nút gốc này. Điều đó được hoàn thành bởi các hoạt động, trong đó có parent\_id hoặc pid với giá trị 1.

Lưu ý: Khóa học tập trung vào kế hoạch tuần tự và không thảo luận về kế hoạch thực hiện song song.

Để vẽ kế hoạch như một cây, làm như sau:

1. Lấy ID với số lượng thấp nhất và đặt nó ở đầu trang.
2. Hãy tìm hàng có một PID (cha-mẹ) bằng với giá trị này.
3. Những chỗ trong cây dưới cha-mẹ theo giá trị POS của chúng từ mức thấp nhất lên cao nhất, thực hiện từ trái sang phải.
4. Sau khi tất cả các ID cho một cha-mẹ được tìm thấy, di chuyển xuống ID tiếp theo và lặp lại quá trình, việc tìm kiếm các hàng mới với các PID tương tự.

Điều đầu tiên xác định trong một kế hoạch giải thích là có nút được thực hiện đầu tiên. Phương pháp trong slide giải thích điều này, nhưng đôi khi có kế hoạch phức tạp rất khó để làm điều này



và cũng khó khăn để thực hiện theo các bước từ đầu đến cuối. Các quy tắc cơ bản áp dụng tương tự. Bạn luôn luôn có thể thay đổi kế hoạch, giấu một nhánh của cây mà không tiêu tốn nhiều tài nguyên.

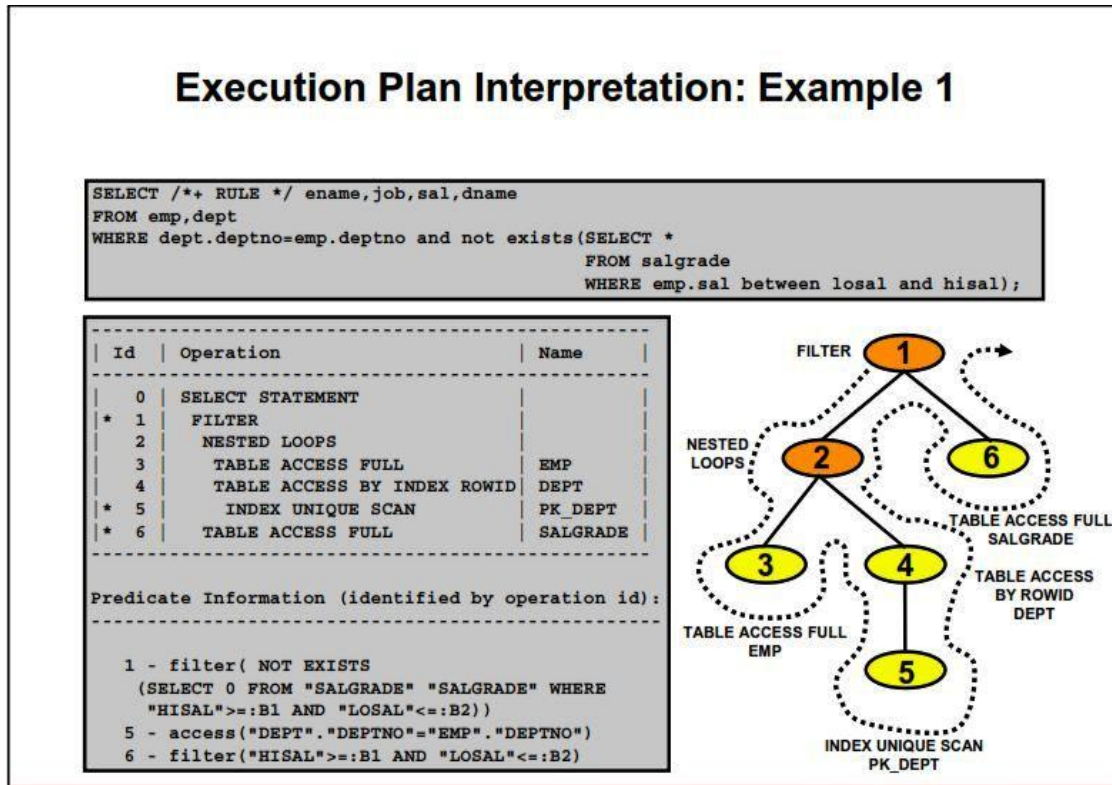
Chuẩn giải thích kế hoạch diễn giải:

1. Bắt đầu ở đầu trang.
2. Di chuyển xuống các hàng cho đến khi bạn đến được với một trong các dữ liệu sản xuất, nhưng không tiêu thụ bất kỳ thứ gì. Đây là hàng bắt đầu.
3. Tìm các anh chị em của hàng này. Chúng được thực hiện tiếp theo.
4. Sau khi các em được thực thi, cha-mẹ được thực hiện tiếp theo.
5. Bây giờ cha mẹ và con của nó được hoàn thành, làm việc trở lại cây, và tìm các anh chị em của hàng cha mẹ của nó. Thực hiện như trước.
6. Di chuyển trở lại kế hoạch cho đến khi tất cả các hàng hết.

Chuẩn cây diễn giải:

1. Bắt đầu ở đầu trang.
2. Di chuyển xuống cây bên trái cho đến khi bạn đạt đến nút bên trái. Điều này được thực hiện đầu tiên.
3. Tìm các anh chị em của hàng này. Chúng được thực hiện tiếp theo.
4. Sau khi các em được thực thi, cha-mẹ được thực hiện tiếp theo.
5. Bây giờ cha mẹ và con của nó được hoàn thành, làm việc trở lại cây, và tìm các anh chị em của hàng cha mẹ. Thực hiện như trước.
6. Di chuyển trở lại lên cây cho đến khi tất cả các hàng hết.

## 29. Execution Plan Interpretation: Example 1



Bạn bắt đầu với một ví dụ truy vấn để minh họa làm thế nào để giải thích một kế hoạch thực hiện. Slide cho thấy một truy vấn với kế hoạch thực hiện liên quan và các kế hoạch tương tự trong các định dạng cây.

Các truy vấn sẽ cố gắng để tìm thấy những nhân viên có mức lương bên ngoài phạm vi của tiền lương trong lương bảng lớp. Các truy vấn là một câu lệnh SELECT từ hai bảng với một subquery dựa trên bảng khác để kiểm tra việc xếp lương.

Xem thứ tự thực hiện cho truy vấn này. Dựa vào các ví dụ trong slide, và từ slide trước đó, thứ tự thực hiện là 3 - 5 - 4 - 2 - 6 - 1:

- 3: Kế hoạch bắt đầu với một bảng quét toàn bộ EMP (ID = 3).
- 5: Các hàng được chuyển lại cho các vòng lặp lồng nhau tham gia kiểm soát bước (ID = 2), trong đó sử dụng chúng để thực hiện việc tra cứu của các hàng trong các chỉ số PK\_DEPT trong ID = 5.
- 4: Các ROWIDs từ các chỉ số được sử dụng để tra cứu các thông tin khác từ DEPT bảng trong ID = 4.
- 2: ID = 2, các vòng lặp lồng nhau tham gia bước, được thực hiện cho đến khi hoàn thành.
- 6: Sau ID = 2 đã cạn kiệt nguồn hàng của nó, một bảng quét toàn bộ SALGRADE trong ID = 6 (tại cùng cấp trong cây như ID = 2, do đó, người anh em của nó) được thực thi.
- 1: Điều này được sử dụng để lọc các hàng từ ID2 và ID6.

Lưu ý rằng trẻ em được thực hiện trước khi các bậc cha mẹ, vì vậy mặc dù cấu trúc cho khác được nhập phải được thiết lập trước khi thực hiện trẻ em, trẻ em được ký hiệu như tiên thực hiện

trước. Có lẽ, cách dễ nhất là xem xét nó như là thứ tự thực hiện hoàn tất, vì vậy đối với các vòng lặp lồng nhau tham gia vào ID = 2, hai trẻ em {ID = 3 và ID = 4 (cùng với con của nó)} phải đã hoàn thành của họ thực hiện trước khi ID = 2 có thể được hoàn thành.

Các ví dụ trong slide là một bài chứa kế hoạch từ V\$SQL\_PLAN với STATISTICS\_LEVEL đặt cho ALL. Báo cáo này cho bạn thấy một số thông tin bổ sung quan trọng so với sản lượng của các giải thích PLAN lệnh:

- A-Rows tương ứng với số hàng được sản xuất bởi nguồn hàng tương ứng.
- Bộ đệm tương ứng với số lượng phù hợp lần đọc thực hiện bởi các nguồn hàng.
- Bắt đầu cho biết số lần các hoạt động tương ứng đã được xử lý.

Đối với mỗi hàng từ bảng EMP, hệ thống được ENAME của nó, SAL, JOB, và DEPTNO. Sau đó hệ thống truy cập bảng DEPT bởi chỉ số duy nhất của nó (PK\_DEPT) để có được sử dụng dname DEPTNO từ tập kết quả trước đó. Nếu bạn quan sát các số liệu thống kê chặt chẽ, các TABLE ACCESS hoạt động đầy trên bảng EMP (ID = 3) được bắt đầu một lần. Tuy nhiên, các hoạt động từ 5 ID và 4 được bắt đầu 14 lần; một lần cho mỗi EMP hàng. Tại bước này (ID = 2), hệ thống được tất cả ENAME, SAL, JOB, và dname.

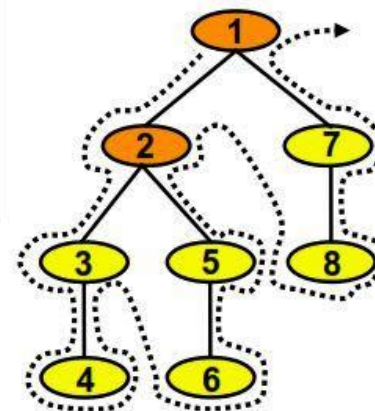
Hệ thống này bây giờ phải lọc ra những nhân viên có mức lương bên ngoài phạm vi của tiền lương trong các bảng xếp lương. Để làm điều đó, cho mỗi hàng từ ID = 2, hệ thống truy cập bảng SALGRADE sử dụng một hoạt động TABLE SCAN PULL để kiểm tra xem tiền lương của người lao động là bên ngoài phạm vi lương. Thao tác này chỉ cần thực hiện 12 lần trong trường hợp này bởi vì tại thời gian chạy hệ thống sẽ thực hiện kiểm tra đối với từng mức lương khác nhau, và có 12 mức lương khác nhau trong bảng EMP.

### 30. Execution Plan Interpretation: Example 2

## Execution Plan Interpretation: Example 2

```
SQL> select /*+ USE_NL(d) use_nl(m) */ m.last_name as dept_manager
2      ,      d.department_name
3      ,      l.street_address
4 from    hr.employees m    join
5          hr.departments d on (d.manager_id = m.employee_id)
6          natural join
7          hr.locations l
8 where   l.city = 'Seattle';
```

```
0  SELECT STATEMENT
1 0  NESTED LOOPS
2 1   NESTED LOOPS
3 2    TABLE ACCESS BY INDEX ROWID LOCATIONS
4 3     INDEX RANGE SCAN          LOC_CITY_IX
5 5    TABLE ACCESS BY INDEX ROWID DEPARTMENTS
6 5     INDEX RANGE SCAN          DEPT_LOCATION_IX
7 1   TABLE ACCESS BY INDEX ROWID EMPLOYEES
8 7    INDEX UNIQUE SCAN          EMP_EMP_ID_PK
```



ORACLE

Truy vấn này lấy tên, tên bộ phận, và địa chỉ cho nhân viên các phòng ban mà được đặt tại Seattle và những người có người quản lý.

Vì lý do định dạng, kế hoạch giải thích có ID trong cột đầu tiên, và PID trong cột thứ hai. Các vị trí được phản ánh bởi sự thụt vào. Kế hoạch thực hiện cho thấy hai vòng lặp lồng nhau tham gia các hoạt động.

Bạn làm theo các bước từ ví dụ trước:

1. Bắt đầu ở đầu trang. ID = 0
2. Di chuyển xuống các nguồn hàng cho đến khi bạn nhận được vào một, trong đó sản xuất dữ liệu, nhưng không tiêu thụ bất kỳ. Trong trường hợp này, ID 0, 1, 2, 3 và tiêu thụ dữ liệu. ID = 4 là nguồn hàng đầu tiên mà không tiêu thụ bất kỳ. Đây là nguồn gốc đầu hàng. ID = 4 được thực hiện đầu tiên. Phạm vi chỉ số quét sản xuất ROWIDs, được sử dụng để tra cứu trong bảng ĐỊA ĐIỂM ID = 3.
3. Nhìn vào các anh chị em của nguồn hàng này. Những nguồn hàng được thực hiện tiếp theo. Các anh chị em ở mức độ giống như ID = 3 là ID = 5. Node ID = 5 có một đứa con ID = 6, mà được thực hiện trước đó. Đây là một chỉ số quét phạm vi sản xuất ROWIDs, được sử dụng để tra cứu trong bảng KHOA trong ID = 5.

4. Sau khi các hoạt động cho trẻ em, các hoạt động phụ huynh là tới. Các vòng LOOPS tham gia tại ID = 2 được thực hiện tiếp theo cùng nhau đưa dữ liệu cơ bản.
5. Bây giờ cha mẹ và con của nó được hoàn thành, đi lại lên cây, và nhìn vào các anh chị em của nguồn hàng mẹ và cha mẹ của nó. Thực hiện như trước. Các anh chị em của ID = 2 cùng cấp trong kế hoạch là ID = 7. Điều này có một ID child = 8, mà được thực hiện đầu tiên. Các chỉ số quét độc đáo sản xuất ROWIDs, được sử dụng để tra cứu trong bảng NHÂN VIÊN ID = 7.
6. Di chuyển trở lại lên kế hoạch cho đến khi tất cả các nguồn hàng đang cạn kiệt. Cuối cùng này được đem đến cùng với các vòng lặp lồng nhau tại ID = 1, mà vượt qua các kết quả lại cho ID = 0.
7. Trình tự thực hiện là: 4 - 3 - 6 - 5 - 2-8 - 7 - 1-0

Dưới đây là mô tả đầy đủ về kế hoạch này:

- Các vòng lặp lồng nhau bên trong được thực hiện đầu tiên sử dụng ĐỊA ĐIỂM như bảng điều khiển, sử dụng một truy cập chỉ mục trên cột CITY. Điều này là bởi vì bạn tìm kiếm cho các phòng ban trong chỉ Seattle.
- Kết quả này được gia nhập với bảng KHOA, sử dụng các chỉ số trên cột LOCATION\_ID tham gia; là kết quả của hoạt động đầu tiên tham gia này là nguồn hàng lái xe cho các vòng lặp lồng nhau thứ hai tham gia.
- Việc tham gia thăm dò thứ hai chỉ số trên cột Employee\_ID của bảng NHÂN VIÊN. Hệ thống có thể làm điều đó bởi vì nó biết (từ lần đầu tham gia) ID nhân viên của tất cả các nhà quản lý của các sở ở Seattle. Lưu ý rằng đây là một quét duy nhất bởi vì nó được dựa trên khóa chính.

Cuối cùng, các bảng EMPLOYEES được truy cập để lấy lại cái tên cuối cùng.

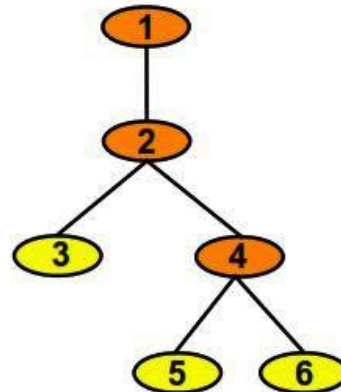
### 31. Execution Plan Interpretation: Example 3

## Execution Plan Interpretation: Example 3

```
select /*+ ORDERED USE_HASH(b) SWAP_JOIN_INPUTS(c) */ max(a.i)
from t1 a, t2 b, t3 c
where a.i = b.i and a.i = c.i;
```

0	SELECT STATEMENT
1	SORT AGGREGATE
2 1	HASH JOIN
3 2	TABLE ACCESS FULL T3
4 2	HASH JOIN
5 4	TABLE ACCESS FULL T1
6 4	TABLE ACCESS FULL T2

<a href="#">Expand All</a>	<a href="#">Collapse All</a>		
Operation	Object	Order	
▼ SELECT STATEMENT		7	
▼ SORT AGGREGATE		6	
▼ HASH JOIN		5	
TABLE ACCESS FULL	T3	1	
▼ HASH JOIN		4	
TABLE ACCESS FULL	T1	2	
TABLE ACCESS FULL	T2	3	



**Join order is: T1 - T2 - T3**

ORACLE

Xem các kế hoạch thực hiện trong slide. Hãy thử để tìm thứ tự trong đó kế hoạch được thực hiện và suy ra để tham gia (thứ tự trong đó hệ thống tham gia bảng) là gì. Một lần nữa, ID là trong lần đầu tiên cột và PID trong cột thứ hai. Các vị trí được phản ánh bởi sự thụt vào. Nó là quan trọng để nhận ra những thứ tự của một kế hoạch thực hiện, để có thể thấy kế hoạch của mình tham gia trong một dấu vết tập tin 10053 sự kiện.

Dưới đây là giải thích của kế hoạch này:

- Hệ thống đầu tiên của bảng băm T3 (Operation ID = 3) vào bộ nhớ.
- Sau đó, nó băm bảng T1 (Operation ID = 5) vào bộ nhớ.
- Sau đó, quá trình quét các bảng T2 bắt đầu (Operation ID = 6).
- Hệ thống chọn một hàng từ T2 T1 và đầu dò (T1.i = T2.i).
- Nếu hàng tồn, các tàu thăm dò hệ thống T3 (T1.i = T3.i).
- Nếu hàng tồn, hệ thống sẽ gửi nó đến hoạt động tiếp theo.
- Hệ thống xuất ra giá trị tối đa từ các kết quả thiết lập trước đó.

Trong kết luận, thứ tự thực hiện là: 3 - 5 - 6 - 4 - 2 - 1.

Việc tham gia theo thứ tự là: T1 - T2 - T3.

Bạn cũng có thể sử dụng Enterprise Manager để hiểu kế hoạch thực hiện, đặc biệt là bởi vì nó hiển thị các cột theo thứ tự.

Lưu ý: Một gợi ý đặc biệt được sử dụng để đảm bảo T3 sẽ là lần đầu tiên trong kế hoạch.

### 32. Reading More Complex Execution Plans

Các kế hoạch ở bàn tay trái, các truy vấn (trong slide) trên các dữ liệu từ điển. Đã quá lâu rằng nó là rất khó khăn để áp dụng các phương pháp trước đó để giải thích nó và xác định vị trí các hoạt động đầu tiên. Bạn luôn có thể sắp xếp một kế hoạch để làm cho nó có thể đọc được. Điều này được minh họa ở bên phải, nơi bạn có thể nhìn thấy cùng một kế hoạch sắp xếp. Như đã trình bày, điều này là dễ dàng để làm khi sử dụng Enterprise Manager hoặc SQL Developer giao diện đồ họa. Bạn có thể thấy rõ rằng kế hoạch này là một UNION ALL của hai nhánh. Kiến thức của bạn về các dữ liệu từ điển cho phép bạn hiểu rằng hai ngành tương ứng với bảng vào từ điển quản lý và những người địa phương quản lý. Kiến thức của bạn về cơ sở dữ liệu của bạn cho phép bạn biết rằng không có bảng dictionarymanaged. Vì vậy, nếu có một vấn đề, nó phải được trên các chi nhánh thứ hai. Để có được xác nhận, bạn phải nhìn vào các thông tin và số liệu thống kê thực hiện kế hoạch của từng nguồn hàng để xác định vị trí các phần của kế hoạch mà tiêu thụ nhiều nhất tài nguyên. Sau đó, bạn chỉ cần mở rộng các chi nhánh bạn muốn điều tra (nơi mà thời gian đang được chi tiêu). Để sử dụng phương pháp này, bạn phải nhìn vào số liệu thống kê thực hiện mà thường được tìm thấy trong V \$ SQL\_PLAN\_STATISTICS hoặc trong các báo cáo tkprof tạo ra từ các tập tin dấu vết. Ví dụ, tkprof cumulates cho mỗi hoạt động cha mẹ thời gian cần để thực thi tự cộng với tổng của tất cả các thời gian hoạt động của trẻ em.

### 33. Reviewing the Execution Plan

Khi bạn điều chỉnh một câu lệnh SQL trong một môi trường xử lý giao dịch trực tuyến (OLTP), mục đích là để điều khiển từ các bảng có các bộ lọc chọn lọc nhất. Điều này có nghĩa rằng có rất ít hàng thông qua bước tiếp theo.

Hãy kiểm tra để xem liệu các đường dẫn truy cập có phải là tối ưu. Khi kiểm tra kế hoạch thực hiện tối ưu, ta tìm kiếm:

- Kế hoạch điều khiển bảng có bộ lọc tốt nhất.
- Số lượng hàng ít nhất được trả lại cho bước tiếp theo.
- Các phương pháp tham gia là thích hợp cho số hàng bị trả lại. Ví dụ, vòng lặp lồng nhau tham gia thông qua các chỉ số có thể không được tối ưu khi có nhiều hàng được trả về.
- Các thông số được sử dụng một cách hiệu quả. Nhìn vào danh sách SELECT để thấy liệu truy cập thông số có cần thiết không.
- Có bất kỳ sản phẩm Descartes không chủ ý (thậm chí với các bảng nhỏ).
- Mỗi bảng đang được truy cập một cách hiệu quả: Hãy xem xét các vị từ trong các câu lệnh SQL và số lượng hàng trong bảng. Tìm hoạt động đáng ngờ, chẳng hạn như một bảng đầy đủ quét trên bảng có số lượng hàng lớn, trong đó có các vị từ trong mệnh đề WHERE.

Ngoài ra, một bảng quét đầy đủ có thể là hiệu quả hơn trên một chiếc bảng nhỏ, hoặc tận dụng một phương pháp tham gia tốt hơn (ví dụ, tham gia băm) cho số hàng trả lại.

Nếu bất cứ điều kiện là không tối ưu, hãy xem xét cơ cấu lại các câu lệnh SQL hoặc các chỉ số có sẵn trên bảng.

### **34. Looking Beyond Execution Plans**

Kế hoạch thực hiện một mình không thể phân biệt giữa lệnh điều chỉnh tốt với lệnh hoạt động kém. Ví dụ, một kết quả giải thích kế hoạch cho thấy rằng một lệnh sử dụng chỉ số không cần thiết, có nghĩa rằng lệnh chạy hiệu quả. Đôi khi chỉ số có thể cực kỳ không hiệu quả. Nó là tốt nhất để sử dụng giải thích kế hoạch để xác định một kế hoạch truy cập, và sau đó chứng minh rằng nó là phương án tối ưu thông qua thử nghiệm. Khi đánh giá một kế hoạch, bạn nên kiểm tra mức tiêu thụ tài nguyên thực tế của lệnh.



## **Phân Công Công Việc**

Đỗ Văn Tuấn đọc hiểu: Objectives -> Explain Plan Using SQL Developer

Lê Trung Kiên đọc hiểu: AUTOTRACE -> Generating SQL Reports from AWR Data.

Phùng Văn Huy đọc hiểu: SQL Monitoring: Overview -> Looking Beyond Execution Plans