**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**-----------------\*\*\*-----------------**



**BÁO CÁO MÔN THIẾT KẾ VÀ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**Đề tài: APPLICATION TRACING**

Sinh viên tham gia:

Nguyễn Thái Sơn – 20112076

Lê Hữu Diện – 20111328

Nguyễn Minh Vương – 20112485

Trần Khánh Duy – 20111351

Giảng viên hướng dẫn: TS.Trần Việt Trung

Hà Nội, ngày 12 tháng 05 năm 2015

PHỤ LỤC

[1. Mục đích 2](#_Toc418580595)

[2. End to End application tracing challenge 2](#_Toc418580596)

[3. Vị trí cho dự đoán tìm kiếm 4](#_Toc418580597)

[4. Service 5](#_Toc418580598)

[5. Sử dụng service với ứng dụng client. 6](#_Toc418580599)

[6. Tìm kiếm Service. 6](#_Toc418580600)

[7. Sử dụng Enterprise Manager để tìm kiếm serivce. 8](#_Toc418580601)

[8. Tìm kiếm Service: ví dụ 9](#_Toc418580602)

[9. Tìm kiếm cấp độ phiên làm việc: ví dụ 10](#_Toc418580603)

[11. Tìm kiếm phiên làm việc của riêng bạn. 11](#_Toc418580604)

[12. Trcsess Utility 12](#_Toc418580605)

[13. Cách sử dụng trcsess Utility 13](#_Toc418580606)

[ Trcess Utility : ví dụ 14](#_Toc418580607)

[14. Nội dung tệp tìm kiếm SQL 15](#_Toc418580608)

[a. Nội dung tệp tìm kiếm SQL : ví dụ 16](#_Toc418580609)

[15. Định dạng tệp SQL Trace: đánh giá 17](#_Toc418580610)

[16. Cách sử dụng công cụ Tkprof Utility 18](#_Toc418580611)

[17. Tkprof sắp xếp tùy chọn 20](#_Toc418580612)

[18. Đầu ra lệnh tkprof 21](#_Toc418580613)

[ Ví dụ 26](#_Toc418580614)

[a. Không đánh chỉ mục: ví dụ 26](#_Toc418580615)

[b. Đánh chỉ mục : ví dụ 27](#_Toc418580616)

## Mục đích

* Sau khi hoàn thành bài học này , chúng ta sẽ biết cách làm thế nào để:
  + - Cấu hình công cụ tìm kiếm Sql để thống kê phiên làm việc.
    - Sử dụng trcsess utility để tổng hợp các file SQL trace
    - Định dạng các tệp tìm kiếm sử dụng tkprof utility.
    - Giải thích được đầu ra của các câu lệnh tkprof

## End to End application tracing challenge

Machine generated alternative text: End-to-End Application Tracing Challenge 
End-to-End Application Tracing Challenge 
Client 
Client OE 
Client 
Client 
Clients 
Client JF/Søssion 6 
I want to retrieve traces from CRM service. 
I want to retrieve traces from client C4. 
I want to retrieve traces from session 6. 
OR ACI 

Oracle Database thực hiện tìm kiếm bằng cách sinh ra một file tìm kiếm cho mỗi xử lý của máy chủ khi sử dụng cơ chế tìm kiếm.

Tìm kiếm một client cụ thể thường không phải là một vấn đề trong mô hình máy chủ chuyên dụng giống như quá trình đơn chuyên phục vụ cho một phiên làm việc trong quá trình hoạt động. Tuy nhiên, trong một cấu hình máy chủ chia sẻ, một client sẽ được xử lý bởi các tiến trình khác nhau sau mỗi lần hoạt động. Dấu vết liên quan tới phiên làm việc của người sử dụng tồn tại rải rác trên file tìm kiếm khác nhau thuộc các quá trình khác nhau .Điều này gây khó khăn cho bạn lấy hình ảnh toàn cục của vòng đời của phiên làm việc.

Ngoài ra, nếu bạn muốn củng cố việc tìm kiếm thông tin cho một service cụ thể, cho hiệu năng hoặc cho mục đích gỡ lỗi? Đây cũng là khó khăn bởi vì bạn có nhiều client đang sử dụng các dịch vụ tương tự và từng tạo tệp tin tìm kiếm thuộc tiến trình máy chủ xử lý nó.

* Ứng dụng tìm kiếm End to end

Đơn giản hóa tiến trình của việc chuẩn đoán các vấn đề về hiệu năng trong môi trường nhiều tầng cho phép ứng dụng thấy được khối lượng công việc:

* + - * Service
      * Module
      * Action
      * Session
      * Client
* End to end application tracing tools
  + - * Enterprise Manager
      * DBMS\_APPLICATION\_INFO, DBMS\_SERVICE,DBMS\_MONITOR,DBMS\_SESSION
      * SQL Trace vaf trcsess utility
* End-to-end application tracing

Làm đơn giản hóa việc dự đoán các vấn đề về hiệu năng trong môi trường nhiều tầng. Trong môi trường nhiều tầng, một yêu cầu từ người dùng cuối được định tuyến tới các phiên làm việc cơ sở dữ liệu khác nhau bởi tầng trung gian,làm nó gặp khó khăn cho việc tìm kiếm một client cụ thể. Mỗi client xác định được sử dụng để tìm kiếm một client cuối duy nhất cụ thể thông qua tất cả các tầng tới máy chủ cơ sở dữ liệu.

Sử dụng để xác định nguồn của một khối lượng công việc lớn như một số lượng lớn cấu lệnh sql. Ngoài ra, bạn có thể xác định phiên làm việc của người dụng hiện tại đã làm những gì ở cấp cơ sở dữ liệu để giải quyết vấn đề hiệu suất của người dùng đó.

Làm đơn giản hóa quản lý khối lượng công việc ứng dụng bởi tìm kiếm các module cụ thể và hoạt động trong một service. Vấn đề khối lượng công việc có thể xác định bởi:

* + Client identifier: xác định người dùng cuối dựa trên login ID (HR)
  + Service : xác định một nhóm của ứng dụng với thuộc tính chung, ngưỡng cấp độ của service, và ưu tiên; hoặc một ứng dụng đơn giản.
  + Module :xác định một khối chức năng bên trong một ứng dụng.
  + Action: xác định hoạt động, giống như INSERT hoặc UPDATE trong một module.
  + Session: xác định một phiên làm việc dựa trên định dạng cơ sở dữ liệu (SID)

## Vị trí cho dự đoán tìm kiếm

Machine generated alternative text: Location for Diagnostic Traces 
Location for Diagnostic Traces 
DIAGNOSTIC DEST 
Diagnostic Previous Location 
$ PDR 
$ADR 
$ADR 
$ADR 
$ADR 
$ADR 
Foreground process 
Backg round process 
Aert log data 
Core dumps 
Incident dumps 
USER DUMP DEST 
BACKGROUND DUMP 
BACKGROUND DUMP 
CORE DUMP DEST 
USER DUMP DEST 
BACKGROUND DUMP 
DEST 
DEST 
DEST 
HOME/ e race 
HOME/ trace 
HOME/ alert 
HOME/ trace 
HOME/ edump 
HOME/ n 
V$DIÄG INFO 
critical error trace 
ACI 
R_HOME/trace Oracle Database 1 lg trace 

Từ phiên bản Oracle Database 11g, ADR là một nơi lưu trữ file cơ sở cho cơ sở dữ liệu lỗi như là tìm kiếm, lỗi nguy hiểm, các gói, ghi cảnh báo, báo cáo tình trạng thiết bị, lỗi nhân và những lỗi tương tự. Tham số khởi tạo …\_DUMP\_DEST được bỏ qua. Thư mục gốc ADR được biết như là ADR cơ bản. Vị trí đó được thiết lập bởi tham số khởi tạo DIAGNOSTIC\_DEST. Trong slide , vị trí này được bao hàm bởi $ADR\_HOME. Tuy nhiên, không có biến môi trường chính nào được gọi ADR\_HOME. Bảng trên đưa ra được mô tả các lớp khác nhau của việc tìm kiếm dữ liệu và các lỗi thuộc cả hai phiên bản trong Oracle Database 10g và 11g. Từ phiên bản Oracle Database 11g không có sự phân biệt giữa tệp tìm kiếm foreground và background. Hai kiểu của tệp đi từ thư mục $ADR\_HOME/trace. Bạn có thể sử dụng V$DIAG\_INFO từ danh sách vị trí ADR quan trọng.

Tất cả hiện tượng bình thường được lưu trữ bên trong thư mục con TRACE. Từ Oracle Database 11g, thông tin lỗi nghiêm trọng được đưa tới tệp tìm kiếm tiến trình tương ứng thay cho các lỗi bất thường. Lỗi bất thường là vị trí trong tệp tách rời nhau từ tệp tìm kiếm tiến trình xử lý bình thường.

* *Note : Sự khác biệt chính giữa một dấu vết và một lỗi:*
  + - *Dấu vết : đầu ra liến tục như: từ khi sql tracing được bật.*
    - *Một lỗi: một lần nào đó mà đầu ra trả về từ một sự kiện như một việc bất ngờ xảy ra. Ngoài ra, một lỗi nhân như là lỗi nhớ nhị phân hoặc như là một cổng cụ thể.*

## Service

* Các nhóm phiên làm việc thực hiện các công việc tương tự nhau.
* Cung cấp một hình ảnh hệ thống đơn thay vì một hình ảnh của đa đối tượng.
* Là một phần của các tác vụ quản trị thường xuyên cung cấp dịch vụ động theo yêu cầu được chỉ định.
* Là cơ sở cho tính sẵn sàng cao của các kết nối
* Cung cấp một chiều hướng tinh chỉnh hiệu năng.
* Thiết lập cho lưu giữ thông tin tìm kiếm.
* Service

Khái niệm về service được đưa ra lần đầu tiền trong Oracle như là một phương tiện cho người nghe để thực hiện kết nối cân bằng tải giữa các nút và các cụm. Tuy nhiên, khái niệm, định nghĩa và thực thi của service đã được mở rộng đáng kể. Một service tổ chức việc thực thi công việc trong phạm vi cơ sở dữ liệu để khiến nó trở nên dễ quản lý hơn, đo lường được, tinh chỉnh được, và phục hồi được. Một service là một nhóm các tác vụ liên quan đến nhau trong phạm vi cơ sở dữ liệu với chức năng phổ biến, chất lượng mong đợi, và quan hệ ưu tiên tới dịch vụ khác. Service cung cấp một hệ thống đơn giản cho việc quản lý ứng dụng tính toán mà service có thể cấu hình, quản trị, cấp quyền, vô hiệu hóa và được coi như là một thực thể duy nhất sử dụng giao diện chuẩn, Enterprise Manager, và SRVCTL…

Service có tính sẵn dùng: sau khi gặp sự cố ngừng hoạt động, một service sẽ phục hồi một cách tự động và nhanh chóng.

Service cung cấp thêm một chiều hướng để tinh chỉnh hiệu năng. Cùng với service, khối lượng công việc được ẩn đi và đo lường được. Tinh chỉnh bởi "service và sql" thay thế tinh chỉnh bởi "session và sql" trong phần lớn của hệ thống nơi mà phiên làm việc chưa xác định và chia sẻ.

Service cung cấp xử lý đơn giản cho phép giữ các thông tin tìm kiếm bởi tên service bất kì của phiên làm việc.

## Sử dụng service với ứng dụng client.

Machine generated alternative text: Using Services with Client Applications 
Using Services with Client Applications 
ERP- (DESCRIPTION- 
(ADDRESS- (PROTOCOL-TCP) (Hosrzmynode) (PORT-1521) ) 
(CONNECT (SERVICE NAME-ERP) ) ) 
j dbc : oracle : Oci 
url- "jdbc : oracle : thin: e (DESCRIPTION- 
(ADDRESS- (PROTOCOL-TCP) (HOST-mynode) (PORT-1521) ) 
(CONNECT DATA- (SERVICE NAME-ERP) ) ) " 
ACI 

Mỗi service được sử dụng bởi bất kì client nào kết nối tới máy chủ cơ sở dữ liệu, service đó tự động áp dụng cho hoạt động của client. Ứng dụng có thể nhóm lại theo các service bởi chỉ đơn giản là sử dụng tên service khác nhau cho mỗi ứng dụng để kết nối.

Ứng dụng và tầng trung gian kết nối lựa chọn chung một service bằng cách sử dụng mô tả kết nối TNS(transperent network substrate).

Serivce đã lựa chọn cần phải đối chiếu với dịch vụ mà đã được tạo ra.

* Ví dụ đầu tiên trong slide đưa ra về TNS mô tả kết nối mà có thể được sử dụng để truy cập dịch vụ ERP
* Ví dụ thứ hai đưa ra : JDBC mô tả kết nối sử dụng định nghĩa TNS trước khi mô tả kết nối.
* Ví dụ thứ 3 : JDBC mô tả kết nối sử dụng TNS mô tả kết nối cũng sử dụng TNS để mô tả kết nối.

## Tìm kiếm Service.

* Các ứng dụng sử dụng các service cần phải có đủ điều kiện như:
  + - * Module
      * Action
      * Client\_identifier
* Thiết lập sử dụng theo sau gói PL/SQL
  + - * DBMS\_APPICATION\_INFO
      * DBMS\_SESSION
* Tím kiếm có thể dừng lại ở tất cả các cấp độ.
  + - * CLIENT\_IDENTIFIER
      * SESSION\_ID
      * SERVICE\_NAMES
      * MODULE
      * ACTION
      * Kết hợp cả 3 : service\_name,module,action.

Mỗi ứng dụng có thể được cho là một service bởi tên MODULE và tên ACTION để xác định hoạt động quan trọng bên trong service. Điều này cho phép bạn xác định các vị trí thực hiện các giao dịch còn thiếu cho khối lượng công việc đã được phân loại. Đây là điều quan trọng khi bạn theo dõi hiệu năng trong hệ thống sử dụng các kết nối chung hoặc theo dõi tiến trình giao dịch. Đối với các hệ thống này, phiên làm việc được chia sẻ, điều này làm việc giải thích trở nên khó khăn.

SERVICE\_NAME ,MODULE ,ACTION ,CLIENT\_IDENTIFIER và SESSION\_ID thực tế là các cột trong V$SESSION.

* SERVICENAME được thiết lập tự động trong thời gian đăng nhập dựa trên mô tả kết nối.
* SESSION\_ID tự động thiết lập bởi cơ sở dữ liệu khi phiên làm việc được khởi tạo.
* MODULE và ACTION được thiết lập cho ứng dụng sử dụng gói DBMS\_APPLICATION\_INFO PL/SQL . Hoặc gọi giao diện đặc biệt Orracle Call Interface(OCI).
* MODULE nên được thiết lập tên mà được nhận ra bởi người sử dụng cho chương trình hiện tại thực hiện.
* ACTION nên được thiết lập một hoạt động đặc biệt hoặc nhiệm vụ mà người sử dụng thực hiện trong một module (ví dụ: truy cập một khác hàng mới).
* CLIENT\_IDENTIFIER có thể thiết lập sử dụng thủ tục DBMS\_SESSION.SET\_IDENTIFIER.

Thêm vào đó cách thức tìm kiếm mỗi file tìm kiếm mỗi phiên làm việc tạo ra các tệp tìm kiếm với câu lệnh sql mà có thể chứa thông tin tìm kiếm cho nhiều người dùng cuối hoặc các ứng dụng cuối. Trừ tất cả phiên làm việc cơ sở dữ liệu đang được tìm kiếm, một số thông tin từ phiên làm việc của người dùng cuối có thể bỏ qua.

Với tiêu chí mà bạn cung cấp, (SERVICE\_NAME, MODULE\_NAME, ACTION), thông tin tìm kiếm cụ thể được giữ trong việc thiết lập các tệp tìm kiếm và kết hợp vào file tìm kiếm đầu ra đơn giản. Điều này cho phép bạn tạo ra các tệp tìm kiếm chứa SQL, tệp này liên quan tới khối lượng công việc cụ thể. Nó cũng có thể làm giống như cho CLIENT\_IDS và SESSION\_IDS.

* *Chú ý: DBA\_ENABLED\_TRACES hiển thị thông tin nhiều hơn tìm kiếm cho phép.*

## Sử dụng Enterprise Manager để tìm kiếm serivce.

Machine generated alternative text: Use Enterprise Manager to Trace Services 
Use Enterprise Manager to Trace Services 
Tap 
Top Consumers 
Tops.rvic.s I.gp-CenE In-Senga; 
EnablesoL Trace Disable 
Sele'l_hll I 
a sy_SWSEES 
um ulativ. 
Earned Delta cpu 
500 FALSE 
250 FALSE 
290 FALSE 
OR ACI 

Trên trang hiệu năng, bạn có thể click vào link Top Consumers. Trang Top Consumers sẽ được hiển thị.

Trang Top Consumers có một số tab cho việc hiển thị cơ sở dữ liệu của bạn như một hệ thống đơn. Trang chứa 4 biểu đồ: Top clients, top services, top modules, top actions. Mỗi biểu đồ cung cấp một quan điểm khác nhau về nguồn client sử dụng hàng đầu trong cơ sở dữ liệu của bạn.

Trang tab Top Services hiển thị thông tin liên quan tới hiệu suất cho dịch vụ được định nghĩa trong cơ sở dữ liệu của bạn. Trong trang này, bạn có thể cho phép hoặc làm vô hiệu hóa tìm kiếm ở cấp độ service.

## Tìm kiếm Service: ví dụ

Machine generated alternative text: Service Tracing: Example 
Service Tracing: Example 
Trace on service, module, and action: 
exec DBMS MONITOR. SERV MOD ACT TRACE ENABLE( 'AP' ) ; 
exec DBMS MONITOR. SERV MOD ACT TRACE ENABLE(- 
' QUERY DELINQUENT • ) ; 
Trace a particular client identifier: 
exec DBMS MONITOR. CLIENT ID TRACE ENABLE 
(client id»'C4', waits TRUE, binds FALSE) ; 
ACI 

Ở dòng đầu tiên, tất cả phiên làm việc được ghi lại dưới service AP đã được truy suất. Một file tìm kiếm được tạo cho mỗi phiên làm việc sử dụng các service kể cả các module và các action. Bạn cũng có thể cho phép tìm kiếm cho các thao tác cụ thể với mỗi service. Đây là minh họa trong ví du thứ 2, nơi mà tất cả các phiên làm việc của AP service mà thực hiện hoạt động truy vấn QUERY\_DELINQUENT bên trong các module PAYMENTS đã được truy suất.

Tìm kiếm service, module và action cho phép bạn tập trung tinh chỉnh kết quả có được trên sql cụ thể. Thay vì chọn lọc thông qua tệp tìm kiếm với SQL từ các chương trình khác nhau. Chỉ các câu lệnh SQL đã được xác định với MODULE này và ACTION đã được ghi lại trong tệp tìm kiếm. Với chức năng này, liên hệ giữa các sự kiện chờ cho các hoạt động cụ thể có thể đã được xác định.

Bạn cũng có thể bắt đầu tìm kiếm cho một định danh client cụ thể như được đưa ra ở ví dụ thứ 3. Trong ví dụ này, C4 là định danh client cho SQL Tracing đã được bật. Tham số TRUE: các thông tin chờ đưa ra trong file trích xuất. Tham số FALSE : các thông tin rằng buộc ko được đưa ra trong file trích xuất.

Mặc dù không được đưa ra trong slide, bạn có thể sử dụng thủ tục CLIENT\_ID\_TRACE\_DISABLE để vô hiệu hóa tìm kiếm tổng thể đối với các cơ sở dữ liệu cho mỗi định dang client đã được xác định. Để vô hiệu hóa tìm kiếm, cho ví dụ trước, thực hiện lệnh sau:

EXECUTE

DBMS\_MONITOR.CLIENT\_ID\_TRACE\_DISABLE(client\_id => 'C4');

* *Chú ý: CLIENT\_IDENTIFIER có thể thiết lập để sử dụng thủ tục DBMS\_SESSION.SET\_IDENTIFIER.*

## Tìm kiếm cấp độ phiên làm việc: ví dụ

Machine generated alternative text: Session Level Tracing: Example 
Session Level Tracing: Example 
For all sessions in the database: 
EXEC dbms monitor. DATABASE TRACE ENABLE (TRUE, TRUE) ; 
EXEC dbms monitor .DATÄBÄSE TRACE DISABLE() ; 
For a particular session: 
EXEC dbms monitor. SESSION TRACE ENABLE (session 
27, serial num-»60, waits--»TRUE, binds--»FALSE) ; 
EXEC dbms monitor. SESSION TRACE DISABLE (session id 
serial ; 
ACI 

Bạn có thể sử dụng tìm kiếm để debug các vấn đề về hiệu năng. Thủ tục tìm kiếm cho phép thực hiện như một thành phần của gói DBMS\_MONITOR. Thủ tục này cho phép tìm kiếm toàn cục cho một cơ sở dữ liệu.

Bạn có thể sử dụng thủ tục DATABASE\_TRACE\_ENABLE để cho phép tìm kiếm phiên làm việc cấp SQL theo chiều rộng. Thủ tục có các tham số sau đây:

* WAITS: xác định có hay không chờ đợi các thông tin để tìm kiếm.
* BINDS: xác định có hay không thông tin rằng buộc để tìm kiếm.
* INSTANCE\_NAME: xác định trường hợp cho phép tìm kiếm.
* INSTANCE\_NAME có nghĩa là cho phép tìm kiếm cấp độ phiên làm việc trên toàn bộ cơ sở dữ liệu.

Sử dụng DATABASE\_TRACE\_DISABLE để vô hiệu hóa tìm kiếm SQL cho toàn bộ cơ sở dữ liệu hoặc một trường hợp cụ thể.

Tương tự như vậy, bạn có thể sử dụng SESSION\_TRACE\_ANABLE để cho phép tìm kiếm trên phiên làm việc xác định của cơ sở dữ liệu trên phạm vị địa phương. SID và SERIAL# có thể được tìm thấy từ V$SESSION.

Sử dụng SESSION\_TRACE\_DISABLE để vô hiệu hóa tìm kiếm cho phiên làm việc xác định của cơ sở dữ liệu và chuỗi số.

* *Chú ý: SQL Trace bao hàm một số chi phí. Vì vậy bạn thường không muốn bật SQL TRACE.*

## Tìm kiếm phiên làm việc của riêng bạn.

Machine generated alternative text: Trace Your Own Session 
Trace Your Own Session 
Enabling trace: 
EXEC DBMS SESSION. SESSION TRACE ENABLE (waits 
TRUE, binds FALSE) ; 
Disabling trace: 
EXEC DBMS SESSION. SESSION TRACE DISABLE ; 
Easily identifying your trace files: 
alter session set 
tracefile identifier: 'mytraceid' ; 
ACI 

Mặc dù gói DBMS\_MONITOR có thể được gọi chỉ bởi người sử dụng với quyền DBA, một số người sử dụng có thể bật SQL tìm kiếm cho họ hoặc những phiên làm việc của họ bởi sử dụng gói DBMS\_SESSION. Thủ tục SESSION\_TRACE\_ENABLE có thể được gọi bởi một số người dùng cho phép tìm kiếm SQL với cấp độ phiên làm việc cho họ hoặc phiên làm việc của họ. Một ví dụ được đưa ra ở trên slide.

TRACEFILE\_IDENTIFIER khởi tạo tham số cụ thể một định danh Client trở thành một thành phần của tên file tìm kiếm Oracle. Bạn có thể sử dụng giống như định danh Client để xác định một tệp tìm kiếm đơn giản từ tên của nó và cùng với việc mở nó hoặc xem nội dung của nó. Mỗi lần tham số này sẽ được thay đổi động ở cấp độ phiên làm việc. Tiếp theo tìm kiếm lỗi ghi lại từ một tệp tìm kiếm sẽ có giá trị tham số mới được nhúng trong tên của nó. Tham số này có thể chỉ được sử dụng để thay đổi tên của tên tìm kiếm xử lý trên forceground; xử lý background sẽ được tiếp tục để định dạng chính tên tệp tìm kiếm. Với tiến trình foreground , cột TRACEID của V$PROCESS chứa giá trị hiện tại của tham số này. Khi giá trị tham số này được thiết lập, tên tệp tìm kiếm có định dạng như sau: sid\_ora\_pid\_traceid.trc.

## Trcsess Utility

Machine generated alternative text: The trcsess Utility 
Client 
Ded i cated 
server 
The trcsess Utility 
Client 
Dedicated 
server 
Trace file 
for CRM service 
Client 
Dedicated 
Clients 
CRM ERP CRM 
Shared 
TRCSEss 
Trace file 
for one client 
server 
Trace 
ACI 
Shared 
Trace 
Report 

Trcess utility kết hợp để tìm kiếm đầu ra từ tệp tìm kiếm được lựa chọn trên cơ sở của nhiều tiêu chuẩn: ID phiên làm việc, định danh client, tên service, tên hoạt động, tên module. Sau khi trcsess kết hợp với thông tin tìm kiếm dưới một tệp tin đầu ra, tệp đầu ra có thể được xử lý bởi tkprof.

Khi sử dụng DBMS\_MONITOR.SERV\_MOD\_ACT\_TRACE\_ENABLE, thông tin tìm kiếm được đưa ra trong nhiều tệp tìm kiếm và bạn cần phải sử dụng công cụ trcsess để sưu tập nó thành một file.

Trcsess utility có lợi cho việc kết hợp để tìm kiếm thành phần của phiên làm việc hoặc service cho hiệu năng hoặc cho mục đích sửa lỗi.

Tìm kiếm một phiên làm việc cụ thể thường không là vấn đề trong quyết định mô hình máy chủ chuyên dụng bởi vì một tiến trình đơn chuyên dụng phục vụ một phiên làm việc trong thời gian sống của nó. Tất cả thông tin tìm kiếm cho phiên làm việc có thể được thấy từ tệp tìm kiếm thuộc về máy chủ chuyên dụng mà phục vụ nó. Tuy nhiên, tìm kiếm một service là một nhiệm vụ phục tạp thậm chí cả trong mô hình máy chủ chuyên dụng.

Ngoài ra, trong cấu hình máy chủ chia sẻ, một phiên làm việc của người sử dụng được phục vụ bởi các tiến trình khác từ lần này tới lần khác. Tìm kiếm liên quan tới phiên làm việc của người sử dụng rải rác trên các tệp tin tìm kiếm khác nhau thuộc tiến trình khác nhau. Điều này làm nó khó khăn để lấy toàn bộ thông tin của một chu kì phiên làm việc.

## Cách sử dụng trcsess Utility

Machine generated alternative text: Invoking the trcsess Utility 
trcsess 
Invoking the trcsess Utility 
(output—output file name) 
(session—session idl 
(clientid:client identifier) 
(service—service name) 
(action—action namel 
(module—module name) 
tetrace file name»J 
TRCSEss 
Consolidated 
trace file 
ORACLE 

* Cú pháp cho trcsess utility đã được đưa ra trên slide:
* **Output** xác định các tệp, nơi mà đầu ra đã được tạo ra. Nếu điều này không được xác định, tiêu chuẩn đầu ra sẽ được sử dụng cho đầu ra.
* **Session** tổng hợp các thông tin tìm kiếm cho một phiên làm việc cụ thể. Định danh phiên làm việc là một sự kết hợp của đánh chỉ mục phiên làm việc và số seri của phiên làm việc, giống như 21.2371. bạn có thể định vị trí giá trị này trong V$SESSION.
* **Clientid** tổng hợp các thông tin cho định danh client đã được xác định.
* **Service** tổng hợp các thông tin cho tên service đã được xác định.
* **Action** tổng hợp các thông tin cho tên hoạt động đã được xác định.
* **Module** tổng hợp các thông tin cho tên module đã được xác định.
* **<tên tệp tìm kiếm>** là một danh sách của tất cả các tên tệp tìm kiếm, tách ra bởi dấu cách, trong đó trcsess phải tìm kiếm thông tin tìm kiếm. Kí tự đại diện "\*" có thể được sử dụng để xác định tên tệp tìm kiếm. Nếu tệp tìm kiếm không xác định, tất cả tệp trong thư mục hiện tại được lấy ra làm đầu vào cho trcsess. Bạn có thể tìm tệp tìm kiếm trong ADR.
* *Chú ý : mỗi một session,clientId,service,action or module tùy chọn cần phải được xác định. Nếu có nhiều hơn một lựa chọn xác định, các tệp tìm kiếm, đáp ứng tất cả các tiêu chí quy định được hợp nhất thành các tập tin đầu ra.*

### Trcess Utility : ví dụ

Machine generated alternative text: The trcsess Utility: Example 
The trcsess Utility: Example 
exec dbmB session. set identifier ('HR session') ; 
First session 
Second session 
exec dbms session. set identifier ( 'HR session') ; 
exec DBMS MONITOR. CLIENT ID TRACE ENABLE( - 
client gesgion• 
waits FALSE, 
binds FALSE) ; 
select * from 
gel ect * from departments; 
exec DBMS MONITOR. CLIENT ID TRACE DISABLE( - 
client id 'HR session'); 
trcgegg Output—mytrace. trC clientid•'HR session' 
$ORACLE BASE/ diag/rdbms/orcl/orcl/trace/* . trc 
ORACLE 

Ví dụ trên giả sử rằng bạn có 3 phiên làm việc khác nhau. Hai phiên làm việc đã được tìm kiếm (bên trái và bên phải) và một phiên làm việc(ở giữa)đã được cho phép hoặc vô hiệu hóa tìm kiếm các chuối thông tin tìm kiếm từ 2 phiên làm việc trước.

Phiên làm việc đầu tiên và phiên làm việc thứ 2 thiết lập định danh cho client từ giá trị 'HR session'. Điều này sẽ được thực hiện bằng cách sử dụng gói DBMS\_SESSION. Phiên làm việc thứ 3 cho phép tìm kiếm cho 2 phiên làm việc sử dụng gói DBMS\_MONITOR.

Mỗi phiên làm việc được tìm kiếm ngay lập tức với câu lệnh SQL. Mỗi câu lệnh sẽ tạo ra thông tin tìm kiếm trong tệp tìm kiếm trong ADR(Automatic Diagnostic Repository).

Như vậy, phiên làm việc thứ 3 dừng tổ chức tìm kiếm sử dụng gói DBMS\_MONITOR, và tổng hợp các thông tin tìm kiếm cho 'HR session', định danh client trong tệp mytrace.trc. Ví dụ được giả sử rằng tệp tìm kiếm được tổ chức trong thư mục $ORACLE\_BASE/diag/rdbms/orcl/orcl/trace được mặc định trong hầu hết các trường hợp.

## Nội dung tệp tìm kiếm SQL

Như chúng ta đã biết, mỗi tệp SQL tìm kiếm cung cấp thông tin về hiệu năng trên mỗi câu lệnh SQL. Nó được tổ chức thành biểu đồ tìm kiếm cho mỗi câu lệnh.

* Phân tích cú pháp, thực thi, và lấy số lần.
* CPU và số lần đã thực hiện.
* Số lần đọc địa chỉ vật lý, số lần đọc địa chỉ logic.
* Số lượng hàng đã được xử lý.
* Nhớ vào thư viện bộ nhớ cache.
* Tên người dùng dưới mỗi lần thực hiện phân tích cú pháp
* Mỗi ghi chép và triệu hồi.
* Chờ dữ liệu sự kiện cho mỗi câu lệnh sql và sơ lược cho mỗi tệp tìm kiếm.

Nếu con trỏ cho câu lệnh SQL đóng, SQL trace cung cấp thông tin bao gồm:

* Hoạt động hàng đưa ra sơ đồ thực thi thực tế của mỗi câu lệnh SQL.
* Số lượng hàng , số lần đọc phù hợp, số lần đọc địa chỉ vật lý, số lần ghi địa chỉ vật lý, và số lần đã thực hiện cho mỗi hoạt động. Điều này chỉ xảy ra khi STATISTICS\_LEVEL khởi tạo tham số thiết lập là ALL.
* *Chú ý: sử dụng công cụ SQL Trace có thể có một tác động nghiêm trọng tới hiệu năng sử dụng và kết quả có thể làm gia tăng tải trọng cho hệ thống, sử dụng CPU quá mức và không gian đĩa không đủ.*

### Nội dung tệp tìm kiếm SQL : ví dụ

Machine generated alternative text: SQL Trace File Contents: Example 
SQL Trace File Contents: Example 
• I Unix pid 
15911 
••• 2010-07-29 
••• 2010-07-29 
••• 2010-07-29 
••• 2010-07-29 13143111.327 
PARSING IN CURSOR 
69246757 ad-•4edS7aeO• 
• —ploy eu 
END OF STMT 
PARSE '2 e-3000, 
EXEC 
FETCH 
FETCH 
FETCH 
dep-o lid—BS 
STAT '2 pid-o pos-I obi-73933 'TABLE ACCESS FULL EMPLOYEES 
pr. O eogt.3 
rlbk-o. rd 
OR ACI 

Có nhiều kiểu tệp tìm kiếm có thể được tổ chức bởi Oracle Database. Một trong những điều này được giới thiệu từ bài học này thường được gọi là một tệp tìm kiếm SQL. Bài học đưa ra cho bạn một ví dụ đầu ra từ mytrace.trc tìm kiếm tệp SQL được đưa ra từ ví dụ trước.

Trong mỗi loại tệp tìm kiếm, bạn có thể tìm các câu lệnh với một số con trỏ chi tiết tương ứng. Bạn có thể thấy chi tiết biểu đồ cho mỗi giai đoạn thực hiện câu lệnh: PARSE,EXEC,FETCH. Như bạn có thể thấy, bạn có nhiều FETCH cho mỗi một EXEC phụ thuộc trên số lượng hàng trả về bởi truy vấn.

Phần cuối của tìm kiếm là kế hoạch thực thi với một số biểu đồ tích lũy cho mỗi nguồn hàng.

Sự phụ thuộc trên cách bạn cho phép tìm kiếm. Bạn cũng có thể có được các thông tin nhờ chờ các sự kiện và giá trị rằng buộc trong tổ chức các tệp tìm kiếm.

Nói chung, bạn không cố gắng làm rõ tệp tìm kiếm của bạn. Đây là bởi vì bạn không lấy toàn bộ ý tưởng mà phiên làm việc trước đã làm. Ví dụ: mỗi phiên làm việc có thể thực thi câu lệnh giống nhau nhiều lần ở các thời điểm khác nhau. Thông tin tìm kiếm thích hợp sau đó nằm rải rác trên toàn bộ tệp tìm kiếm, điều này làm chúng khó tìm kiếm.

Thay vì bạn sử dụng một công cụ khác nào đó giống như tkprof để làm rõ các nội dung của thông tin tìm kiếm thô.

## Định dạng tệp SQL Trace: đánh giá

Machine generated alternative text: Formatting SQL Trace Files: Overview 
Formatting SQL Trace Files: Overview 
Use the tkprof utility to format your SQL trace files: 
Sort raw trace file to exhibit top SQL statements 
Filter dictionary statements 
Consolidated 
trace file 
Report 
Trace 
Concatenated 
trace file 
ACI 

Tkprof utility phân tích cú pháp tệp tìm kiếm SQL để đưa ra nhiều hơn đầu ra có thể đọc được. Nhớ rằng, tất cả các thông tin trong tkprof có sẵn từ thông tin tìm kiếm thô. Số lượng lớn thông tin này được sắp xếp tùy chọn mà bạn có thể gọi với tkprof ở dấu nhắc lệnh. Một điểm bắt đẩu tốt nhất là tùy chọn sắp xếp **fchela** đặt đầu ra bằng lấy số lần đã thực hiện. Tệp kết quả chứa hầu hết thời gian chi phối câu lệnh SQL ở thời điểm bắt đầu của tệp. Một tham số hữu ích khác là **SYS=NO**. Đây có thể được sử dụng để vô hiệu hóa câu lệnh SQL chạy như SYS user đã được hiển thị. Điều này làm cho đâu ra ngắn nhất và dễ dàng quản lý.

Sau một số tệp tìm kiếm SQL được tổ chức, bạn có thể thực hiện một số điều sau:

* Chạy tkprof trên mỗi tệp tìm kiếm riêng. Thực hiện định dạng một số tệp đầu ra đã định dạng, cho mỗi một phiên làm việc.
* Tệp tìm kiếm móc nối với nhau, và để chay tkprof trên kết quả để đưa ra tệp đầu ra đã được định dạng cho toàn bộ các trường hợp.
* Chạy công cụ dòng lệnh trcsess để kết hợp tìm kiếm thông tin từ nhiều tệp tìm kiếm để chạy rkprof lấy kết quả.
* Tkproft không báo cáo COMMITS và ROLLBACKS mà ghi trên tệp tìm kiếm.
* *Chú ý: thiết lập tham số TIMEd\_STATISTICS là TRUE khi tìm kiếm phiên làm việc bởi vì không so sánh dựa vào thời gian có thể thực hiện ngoài điều này. TRUE là giá trị mặc định với Oracle Database 11g.*

## Cách sử dụng công cụ Tkprof Utility

Machine generated alternative text: Invoking the tkprof Utility 
Invoking the tkprof Utility 
tkprof inputfile outputfile twai I no) 
tsort:option) 
tprin t:nJ 
taggregate:yes I no) 
t insert—sqlscriptfi 
tsys:yes I no) 
ttable:schema. table) 
t explain—user/ password) 
trecord:s tatementfi leJ 
twidth:nJ 

Khi bạn nhập dòng lệnh tkprof mà không cần bất kì một đối số nào, nó cũng tổ chức một thông điệp với một đặc tả của tất cả các lựa chọn tkprof. Các đối số được đưa ra trong slide:

* **Inputfile**: xác định tệp tìm kiếm đầu vào SQL.
* **Outputfile**: xác định tệp mà tkprof ghi để định dạng đầu ra cho nó.
* **Waits** : xác định có hoặc không có bản ghi chung cho bất kì sự kiện chờ được tìm thấy trong tệp tìm kiếm. Giá trị là **YES**,**NO**. mặc định là **YES**
* **Sorts**: sắp xếp câu lếnh SQL đã được tìm thấy theo thứ tự giảm dần của việc xác định sắp xếp tùy chọn trước khi trước khi liệt kê
* **Print**: Danh sách các số nguyên đầu tiên được sắp xếp lệnh SQL từ tệp đầu ra. Nếu bạn bỏ qua tham số này, **tkproft** liệt kê tất cả lệnh SQL đã được tìm kiếm. Tham số này không ảnh hưởng tới hoạt động tùy chọn SQL. Hoạt động SQL luôn tạo ra chèn dữ liệu cho tất cả lệnh SQL đã được tìm kiếm.
* **Aggregate**: nếu thiết lập là **NO**, tkprof không tập hợp nhiều người sử dụng văn bản SQL giống nhau.
* Insert: tạo một kịch bản SQL để lưu trữ biểu đồ tệp tin tìm kiếm trong cơ sở dữ liệu. Tkprof tạo kịch bản này với tên riêng của bạn cho **sqlscriptfile**. Kịch bản này tạo một bảng và chèn một hàng thông kê cho mỗi lệnh sql tìm kiếm vào bảng.
* **Sys**: cho phép và vô hiệu hóa danh sách lệnh sql được đưa ra bởi người sử dụng **SYS**, hoặc lệnh SQL đệ quy, vào tệp tin đầu ra. Giá trị mặc định là YES: **tkprof** liệt kê các lệnh. Giá trị là **NO** **tkprof** bỏ qua chúng. Tham số này không ảnh hưởng tới những kịch bản SQL tùy chọn. Kịch bản SQL luôn luôn kèm theo các thống kê cho tất cả lệnh SQL tìm kiếm, bao gồm lệnh SQL đệ quy.
* **Table** : xác định giản đồ và tên của bàng vào **tkproft** thực thi kế hoạch tạm thời trước khi ghi chúng vào tệp tin đầu ra. Nếu đã xác định bảng đã tồn tại, **tkproft** xóa rất cả hàng trong bảng, sử dụng nó cho lệnh **EXPLAIN** **PLAN** , và để xóa các hàng. Nếu bảng không tồn tại, tkproft tạo nó, sử dụng nó và xóa nó. Người sử dụng đã được xác định cần phải có khả năng **INSERT** **SELECT** và **DELETE** bảng. Nếu bảng không tồn tại, người sử dụng cũng phải có khả năng để **CREATE** **TABLE** và **DROP** **TABLE**. Tùy chọn này cho phép nhiều người chạy **tkprof** đồng thời với người sử dụng như nhau trong giá trị **EXPLAIN**. Người sử dụng có thể chỉ định giá trị **TABLE** khác nhau và tránh can thiệp xấu vào mỗi xử lý khác trên bảng kế hoạch tạm thời. Nếu bạn sử dụng tham số EXPAIN ngoài tham số **TABLE**. **Tkprof** sẽ sửu dụng bảng **PROF$PLAN\_TABLE** trong giản đồ của người sử dụng đã được xác định bởi tham số **EXPLAIN**. Nếu bạn sử dụng tham số TABLE ngoài tham số **EXPAIN** , **tkprof** bỏ qua tham số TABLE và sau đó xóa nó ở cuối.
* **Expain** : Quyết định việc thực hiện kế hoạch cho mỗi lệnh SQL trong tệp tìm kiếm và ghi kế hoạch thực thi vào tệp tin đầu ra. Tkprof quyết định kế hoạch thư thi bởi mật khẩu xác định trong tham số này. Người sử dụng xác định càn phải có quyền **CREATE SESSION** trong hệ thống. Tkprof mất nhiều thời gian để xử lsy tệp tìm kiếm lớn nếu tùy chọn **EXPAIN** đã được sử dụng.
* **Record**. Tạo một kịch bản SQL với tên tệp xác định tên **statementfile** với tất cả lệnh SQL không đệ quy trong tệp tìm kiếm. Điều này có thể được sử dụng để xem lại các sự kiện của người sử dụng từ tệp tìm kiếm.
* **Width** : một số nguyên mà điều khiển độ rộng của dòng đầu ra của một số đầu ra **tkprof**, giống như giải thích kế hoạch. Đây là tham số hữu ích cho sau khi xử lý của đầu ra **tkprof**.

Đầu vào và đầu ra là tham số bắt buộc.

## Tkprof sắp xếp tùy chọn

Machine generated alternative text: tkprof Sorting Options 
prscnt 
prscpu 
prsela 
prsdsk 
pr s qry 
prscu 
prsmzs 
execnt 
exec pu 
exeela 
exedsk 
exeqry 
execu 
tkprof Sorting Options 
Description 
Number of times parse was called 
CPU time parsing 
Elapsed time parsing 
Number of disk reads during parse 
Number of buffers for consistent read during parse 
Number of buffers for current read during parse 
Number of misses in the library cache during parse 
Number of executes that were called 
CPU time spent executing 
Elapsed time executing 
Number of disk reads during execute 
Number of buffers for consistent read during execute 
Number of buffers for current read during execute 
ACI 

Machine generated alternative text: tkprof Sorting Options 
exerow 
exeml s 
f chcn t 
f chcpu 
f che I a 
f chdsk 
f chq IN 
f chcu 
f chrow 
seri d 
tkprof Sorting Options 
Description 
Number of rows processed during execute 
Number of library cache misses during execute 
Number of times fetch was called 
CPU time spent fetching 
Elapsed time fetching 
Number of disk reads during fetch 
Number of buffers for consistent read during fetch 
Number of buffers for current read during fetch 
Number of rows fetched 
User ID of user that parsed the cursor 
ACI 

Bảng này liệt kê tất cả tùy chọn sắp xếp bạn có thể sử dụng với sắp xếp các đối số của tkprof.

## Đầu ra lệnh tkprof

Machine generated alternative text: Output of the tkprof Command 
Output of the tkprof Command 
Text of the SQL statement 
Trace statistics (for statement and recursive calls) 
separated into three SQL processing steps: 
PARSE 
EXECUTE 
FETCH 
Translates the SQL statement into an execution plan 
Executes the statement 
(This step modifies the data for the INSERT, UPDATE, 
and DELETE statements.) 
Retrieves the rows returned by a query 
(Fetches are performed only for the SELECT 
statements.) 
OR ACI 

* Tệp tin đầu ra rkprof liệt kê các số liệu thông kê cho lệnh sql bởi bước xử lý SQL.
* Bước cho mỗi hàng mà chứa thống kế được xác định bởi giá trị của cột gọi.
  + PARSE: đây là bước phân tích lệnh SQL vào kế hoạch thực thi và bao gồm việc kiểm tra cho việc xác thực bảo mật một cách chính xác và kiểm tra sự tồn tại của bảng, các cột, và đối lượng liên quan.
  + EXECUTE: đây là bước thực thi lệnh bởi máy chủ Oracle. Các lệnh Insert, update,, delete, đây là bước thay đổi dữ liệu(bao gồm sắp xếp khi cần thiết). Với lệnh select , bước này xác định các hàng được lựa chọn.
  + FETCH: đây là bước truy vấn các hàng được trả về bởi câu truy vấn và sắp xếp chúng khi cần thiết. Lấy chỉ được thực hiện cho lệnh select.
* *Chú ý: giá trị phân tích cú pháp bao gồm phân tích cứng và phân tích mềm. Phân tích cứng liên quan tới việc phát triển của kế hoạch thự thi(bao gồm cả sự tối ưu hóa). Nó được lưu lại ngay sau đó vào bộ nhớ cache. Phân tích cú pháp mềm có nghĩa là một lệnh SQL gửi cho bộ phận phân tích cú pháp tới cơ sở dữ liệu, nhưng cơ sở dữ liệu tìm kiếm nó trong bộ nhớ cache và chỉ cần để xác minh, giống như quyền hạn truy cập. Phân tích cú pháp cứng có thể tốn kém về chi phí thực hiện đặc biệt là do việc tối ưu hóa. Phân tích cú pháp mềm hầu hết tốn kém về chi phí trong giới hạn của hoat động bộ nhớ cache 25.*

Machine generated alternative text: Output of the tkprof Command 
Output of the tkprof Command 
There are seven categories of trace statistics: 
Count 
CPU 
El apsed 
Disk 
Query 
Current 
Rows 
Number of times the procedure was executed 
Number of seconds to process 
Total number of seconds to execute 
Number Of physical blocks read 
Number of logical buffers read for consistent read 
Number of logical buffers read in current mode 
Number of rows processed by the fetch or execute 
ACI 

Ví dụ:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Call | Count | Cpu | Alapsed | Disk | Query | Current | Rows |
| Parse | 1 | 0.03 | 0.06 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.06 | 0.30 | 1 | 3 | 0 | 0 |
| Fetch | 2 | 0.00 | 0.46 | 0 | 0 | 0 | 1 |
| Total | 4 | 0.09 | 0.83 | 1 | 3 | 0 | 1 |

Bên cạnh cột CALL, tkprof hiển thị thống kê cho mỗi câu lệnh:

* + Count : số lần câu lệnh được phân tích cú pháp , thực thi hoặc lấy
  + CPU: Tổng thời gian CPU bằng giây cho tất cả phân tích cú pháp, thực thi và lấy.
  + Elapsed: tổng thời gian đã thực hiện bằng giây cho tất cả phân tích cú pháp, thực thi và lấy.
  + Disk: tổng số khối dữ liệu vật lý được đọc từ tệp dữ liệu trên ổ đĩa cho tất cả phân tích cú pháp, thực thi và lấy.
  + Query: tổng số lần truy vấn bộ đệm trong chế độ phù hợp cho tất cả phân tích cú pháp, thực thi hoặc lấy
  + Current: tổng số truy vấn bộ đệm trong chế độ hiện tại.
  + Rows: Tổng số các hàng đã được xử lý bởi lệnh SQL. (không bao gồm hàng đã được xử lý bởi truy vấn con của lệnh SQL)
* Chú ý :
* *DISK tương đương với số lần đọc địa chỉ vật lý từ v$sysstart hoặc AUTOTRACE*
* *QUERY tương đương với lấy*

Machine generated alternative text: Output of the tkprof Command 
Output of the tkprof Command 
The tkprof output also includes the following: 
Recursive SQL statements 
Library cache misses 
Parsing user ID 
Execution plan 
Optimizer mode or hint 
Row source operation 
g gee library cache during 
optimizer mode: ALL Rows 
parsing id: SS 
ROV Source 
TABLE ACCESS BY INDEX 
INDEX RANGE SCAN EMP 
ROWID EMPLOYEES (cr-4 pr-1 tine-o us 
NAME IX (cr-2 pr.-I time-go ug cost-I 
OR ACI 

* Gọi đệ quy
  + Để thực thi lệnh SQL được đưa ra bởi người sử dụng, máy chủ Oracle phải đưa ra ngẫu nhiên thêm các lệnh. Như lệnh được gọi lệnh SQL đệ quy. Ví dụ, nếu bạn thêm một hàng trong bảng mà không có đủ không gian cho hàng đó, máy chủ Oracle thực hiện gọi đệ quy tới cấp phát không gian động. Gọi đệ quy cũng được tổ chức khi thông tin từ điển dữ liệu không có sẵn trong từ điển dữ liệu cache và phải được truy vấn từ ổ đĩa.
  + Nếu gọi đệ quy xảy ra trong khi các công cụ SQL tìm kiếm được bật, tkprof đánh dấu chúng một cách rõ ràng như lệnh đệ quy SQL trong tệp tin đầu ra. Bạn có thể ngăn chặn danh sách của gọi đệ quy trong tệp tin đầu ra bởi thiết lập tham số dòng lệnh SYS=NO. chú ý rằng các thống kê cho lệnh SQL đệ quy là luôn luôn bao gồm trong danh sách lệnh SQL gây ra gọi đệ quy.
* Bộ nhớ thư viện Cache
  + Tkprof cũng liệt kê các kết quả nhớ trên thư viện cache từ phân tích cú pháp và thực thi cho mỗi lệnh sql. Số liệu thống kê đưa ra trên dòng rời rạc dưới bảng số liệu thống kê.
* Row source Operations
  + Cung cấp số lượng hàng đã được xử lý cho mỗi hoạt động đã thực thi trên các hàng và thông tin nguồn hàng được thêm, như đọc địa chỉ vật lý và ghi; cr=đọc tương ứng, w= ghi địa chỉ vật lý, r=đọc địa chỉ vật lý, time=thời gian(tính bằng microseconds)
* Phân tích ID người dùng
  + Đây là Id của người sử dụng cuối từ phân tích cú pháp câu lệnh.
* Row Source Operation
  + Các hoạt động nguồn hàng đưa ra dữ liệu nguồn cho thực thi lệnh SQL. Điều này chỉ được bao gồm nếu con trỏ đã được đóng trong thời gian tìm kiếm. Nếu hoạt động nguồn hàng không được đưa trong trong tệp tin tìm kiếm. Bạn có thể muốn xem đầu ra sau đó của EXLAIN PLAN.
* Kế hoạch thực thi.
  + Nếu bạn chỉ định tham số EXPLAIN trên dòng lệnh tkprof , tkprof sử dụng lệnh EXPLAIN PLAN để tổ chức kế hoạch thực thi của mỗi lệnh tìm kiếm SQL. Tkprof cũng đưa ra số hàng đã được xử lý bởi mỗi bước trong kế hoạch thực thi.
* Tối ưu hóa hoặc gợi ý.
  + Điều này đưa ra gợi ý tối ưu hóa mà được sử dụng trong thời gian thực thi của lệnh, nếu không gợi ý, nó sẽ đưa ra chế độ tối ưu hóa được sử dụng.

## Ví dụ

### Không đánh chỉ mục: ví dụ

Machine generated alternative text: tkprof Output with No Index: Example 
tkprof Output with No Index: Example 
select credit I imit) from Where Cugt 
total 
1459 
1459 
0.00 
0.00 
0.02 
0.02 
0.00 
0.00 
0.10 
0.10 
Miggeg library cache during 
optimizer ALL Rows 
parsing id: BB 
ROW Source 
1 
1 SORT AGGREGATE (cr-14S9 pr-72 pv-o time-o us) 
-77 TABLE ACCESS FULL CUSTOMERS (cr-14S9 pr-72 tine-4104 us 
cogt:40S Size—1260 card: 90) 
ACI 

Ví dụ trên slide đưa ra rằng kết quả thực hiện lấy dữ liệu từ bảng CUSTOMERS, nếu cần tới 0.12s để CPU lấy dữ liệu, lệnh thực hiện quét toàn bộ bảng CUSTOMERS như bạn có thể thấy trong hoạt động nguồn hàng của đầu ra.

Lệnh cần phải được tối ưu hóa.

*Note : giá trị CPU hoặc elapsed là 0, timed\_statistics không được thiết lập*

### Đánh chỉ mục : ví dụ

Machine generated alternative text: tkprof Output with Index: Example 
tkprof Output with Index: Example 
select credit limit) from Where 
C Ol_mt 
Cpu 
0.00 
0.00 
0.00 
0.00 
o. 00 
0.00 
0.00 
0.00 
Exec u t e 
F e tCh 
t Ota I 
Mig g eg library cache parse: 
optimizer ALL Rows 
parsing id: 
ROW Sour C e 
1 
77 
SORT AGGREGATE (cr-77 pr-1 time-o us) 
TABLE ACCESS BY INDEX ROWID CUSTOMERS (cr-77 pr-l tine-760 uS 
gizez1260 
INDEX RANGE SCAN CUST CUST_CITY IDX (cr-2 pr-l time-IS2 us 
(Object id 78183) 
ACI 

Kết quả trên cho thấy rằng thời gian CPU giảm xuống 0.01s khi một chỉ mục được tạo trên cột CUST\_CITY. Kết quả này đạt được bởi vì lệnh sử dụng chỉ mục để truy vấn dữ liệu. Thêm vào đó, bởi vì ví dụ này thực thi lại lệnh như nhau, hầu hết khối dữ liệu có sẵn trong bộ nhớ. Bạn có thể đạt được những cải thiện đáng kể trong hiệu năng bởi đánh chỉ mục hợp lý. Xác định vùng cho khả năng sử dụng công cụ SQL Trace.

* *Chú ý: chỉ mục chỉ xây dựng khi cần thiết. Chỉ mục làm chậm xử lý của insert , delete và update bởi vì liên quan tới các hàng cần phải được thêm, thay đổi hoặc loại bỏ. Chỉ mục chưa được sử dụng nên được gỡ bỏ. Tuy nhiên, thay cho việc xử lý tất cả ứng dụng SQL thông qua EXPLAIN PLAN , bạn có thể sử dụng tìm kiếm chỉ mục để xác định và loại bỏ hoàn toàn chỉ mục mà không được sử dụng.*