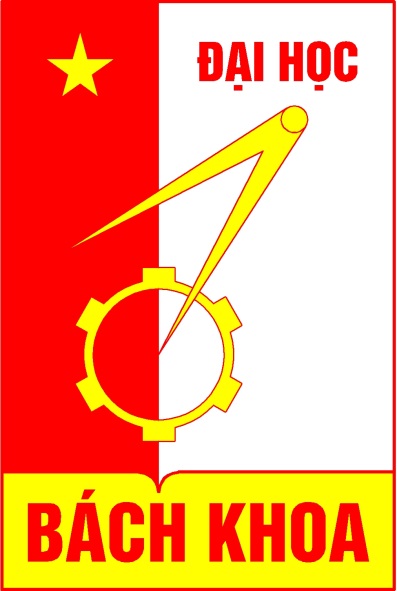
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG**

****

**BÁO CÁO**

Tìm hiểu Oracle Tuning

Chương 10: Tối ưu hóa thống kê

Sinh viên thực hiện: Trần Phúc Thành – 20112201

Nguyễn Văn Lam – 20112616

Hà Bình Xuyên – 20112524

Nguyễn Tiến Thành – 20112189

Hà Nội, tháng 4 năm 2015

Mục lục

[1. Mục tiêu đạt được. 4](#_Toc417999685)

[2. Tối ưu hóa thống kê. 4](#_Toc417999686)

[3. Các loại tối ưu hóa thống kê. 4](#_Toc417999687)

[4. Bảng thống kê. 5](#_Toc417999688)

[5. Chỉ số thống kê. 5](#_Toc417999689)

[6. Yếu tố cụm chỉ số. 6](#_Toc417999690)

[7. Thống kê cột (Colunm Statistics) DBA\_TAB\_COL\_STATISTICS. 8](#_Toc417999691)

[8. Biểu đồ (Histogram). 8](#_Toc417999692)

[9. Biểu đồ tần số (Frequency Histograms). 9](#_Toc417999693)

[10. Khung nhìn biểu đồ tần số. 10](#_Toc417999694)

[11. Biều đồ chiều cao cân bằng (Height-Balance Histogram). 11](#_Toc417999695)

[12. Khung nhìn biểu đồ chiều cao cân bằng. 12](#_Toc417999696)

[13. Biểu đồ đánh giá (Histogram Considerations). 12](#_Toc417999697)

[14. Thống kê nhiều cột: Tổng quan. 13](#_Toc417999698)

[15. Thống kê biểu thức: Tổng quan. 14](#_Toc417999699)

[16. Thu thập số liệu thống kê của hệ thống. 15](#_Toc417999700)

[17. Ví dụ hệ thống thu thập thống kê. 16](#_Toc417999701)

[18. Cơ chế thu thập số liệu thống kê. 18](#_Toc417999702)

[19. Các số liệu thống kê: Tổng quan. 18](#_Toc417999703)

[20. Khi nào tiến hành thu thập số liệu thống kê thủ công. 20](#_Toc417999705)

[21. Thu thập số liệu thống kê thủ công. 21](#_Toc417999706)

[22. Thu thập số liệu thống kê thủ công: Các yếu tố. 22](#_Toc417999707)

[23. Tối ưu hóa việc lấy mẫu động: Tổng quan. 23](#_Toc417999708)

[24. Tối ưu hóa lấy mẫu động trong công việc. 24](#_Toc417999709)

[25. OPTIMIZER\_SAMPLING\_GATHERING. 24](#_Toc417999710)

[26. Khóa các thống kê. 25](#_Toc417999711)

[27. Khôi phục các thống kê. 27](#_Toc417999712)

[28. Xuất khẩu và nhập khẩu các thống kê. 28](#_Toc417999713)

[29. QUIZ – Câu đố. 29](#_Toc417999714)

## Mục tiêu đạt được.

* Nắm được tối ưu thống kê dữ liệu.
* Hiểu được thống kê hệ thống.
* Thiết lập được một số tùy chọn thống kê.
* Sử dụng tập mẫu dữ liệu động.
* Thao tác thống kê tối ưu.

## Tối ưu hóa thống kê.

* Mô tả chi tiết các cơ sở dữ liệu và các đối tượng trong cơ sở dữ liệu.
* Thông tin được sử dụng bởi các truy vấn tối ưu để ước tính:
* Độ chọn lọc của các vị từ.
* Chi phí cho mỗi kế hoạch thực hiện.
* Phương pháp tiếp cận, thứ tự tham gia và phương pháp tham gia/
* CPU và chi phí đầu ra, đầu vào.
* Làm mới tối ưu hóa thống kê bất cứ khi nào nó được coi là cũ, và coi việc này cũng quan trọng như khi thu thập chúng:
* Tự động thu thập bởi hệ thống.
* Tự thu thập bởi người dùng với DBMS\_STATS.
* Những số liệu thống kê được sử dụng bằng các câu truy vấn tối ưu để chọn ra một phương pháp thực thi tốt nhất chô mỗi câu lệnh SQL.
* Bởi vì các đối tượng trong cơ sở dữ liệu thay đổi liên tục nên thống kê phải được cập nhật thường xuyên để có thể mô tả chính xác các đối tượng này. Thống kê này được duy trì tự động bởi cơ sở dữ liêu Oracle, hoặc có thể duy trì thống kê bằng cách sử dụng gói DBMS\_STATS.

## Các loại tối ưu hóa thống kê.

* Thống kê bảng:
* Tổng số hàng.
* Tổng số khối.
* Chiều dài hàng trung bình.
* Thống kê chỉ mục:
* Mức B\*-tree.
* Khóa riêng biệt.
* Số khối lá.
* Hệ số Clustering
* Thống kê hệ thống:
* Hiệu suất và sử dụng đầu vào / đầu ra.
* Hiệu suất và sử dụng CPU.
* Thống kê cột:
* Cơ bản: số giá trị riêng biệt, số giá trị null, giá trị trung bình, lớn nhất, bé nhất.
* Biểu đồ (histograms) (dữ liệu phân tán khi cột dữ liệu không đối xứng).
* Thống kê mở rộng.

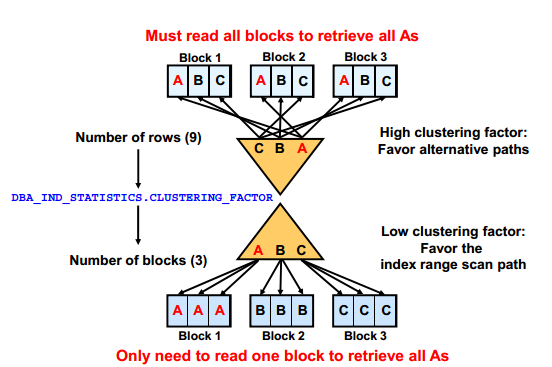
## Bảng thống kê.

* Bảng thống kê được sử dụng để xác định:
* Chi phí truy cập bảng.
* Kết hợp tính toán.
* Kết nối thứ tự
* Một số số liệu thống kê thu thập bảng:
* Đếm số hàng (NUM\_ROWS).
* Đếm số khối (BLOCKS) chính xác.
* Số trung bình hàng.
* Tình trạng (STALE\_STATS).
* NUM\_ROWS: Đây là cơ sở để tính toán các yếu tố. Số hàng đặc biệt quan trọng nếu bảng là bảng thuộc các vòng lặp lồng nhau, vì nó xác định có bao nhiêu lần các bảng được xét.
* BLOCKS: Đây là số lượng các dữ liệu khối được sử dụng. Số khối kết hợp với DB\_FILE\_MULTIBLOCK\_READ\_COUNT cung cấp chi phí truy cập bảng cơ sở.
* AVG\_ROW\_LEN: Đây là chiều dài trung bình của một hàng trong bảng tính bằng bytes.
* STALE\_STATS: Điều này cho bạn biết số liệu thống kê nếu có giá trị trên các bảng tương ứng.
* Lưu ý: có 3 số liệu thống kê khác: EMPTY\_BLOCKS, AVE\_ROW\_LEN, và CHAIN\_CNT không được sử dụng bởi tối ưu hóa, và không được thu thập bởi DBMS\_STATS nếu chúng được yêu cầu lệnh ANALYZE phải được sử dụng.

## Chỉ số thống kê.

* Được sử dụng để tránh việc tìm toàn bộ bảng.
* Thống kê thu thập được:
* Cấp B\*-tree (BLEVEL) chính xác.
* Số khối lá (LEAF\_BLOCKS).
* Hệ sộ clustering (CLUSTERRING\_FACTOR).
* Khóa phân biệt.
* Số trung bình của khối là trong mỗi giá trị riêng biệt trong chỉ số xuất hiện (AVG\_LEAF\_BLOCKS\_PER\_KEY).
* Số lượng trung bình của các khối dữ liệu trong bảng trỏ đến bởi một giá trị riêng biệt trong các chỉ số (AVG\_DATA\_BLOCKS\_PER\_KEY).
* Số hàng dữ liệu trong chỉ mục (NUM\_ROWS).
* Nói chung, để chọn một truy cập chỉ mục, các tối ưu hóa yêu cầu một vị từ về tiền tố của các cột chỉ số. Tuy nhiên, trong trường hợp không có các vị từ và tất các các cột tham chiếu trong câu truy vấn không có mặt trong một chỉ mục, việc tối ưu hóa xem xét sử dụng tìm kiếm toàn bộ chỉ mục hơn so với tìm kiếm toàn bảng.
* BLEVEL: Được sử dụng để tính toán chi phí các tra cứu khối lá. Nó chỉ ra độ sâu của chỉ số từ khối gốc đến khối nút lá. Một độ sâu của “0” chỉ ra rằng khối gốc và khối lá là một.
* LEAF\_BLOCKS: Được sử dụng để tính toán chi phí cho việc tìm kiếm tất các các chỉ mục.
* CLUSTERING FACTOR: Các biện pháp này đo thứ tự các hàng trong bảng dựa trên giá trị của chỉ mục. Nếu giá trị gần với số khối, thì bảng được sắp xếp rất tốt. Trong trường hợp này, chỉ mục trong một khối lá đơn có xu hướng trỏ đến hàng trong cùng khối dữ liệu. Nếu giá trị gần với số hàng thì bảng đã được sắp xếp một cách ngẫu nhiên. Trong trường hợp này, không chắc rằng chỉ mục nằm cùng với các hàng trong khối dữ liệu giống nhau.
* STALE\_STATS: Mục này sẽ cho biết các số liệu thống kê có giá trị trong các chỉ mục tương ứng.
* DISTINCT\_KEYS: Đây là số lượng giá trị các chỉ số riêng biệt. Đối với các chỉ số thực thi có lệnh UNIQUE và PRIMARY KEY hạn chế, giá trị này giống như số hàng trong bảng.
* AVG\_LEAF\_BLOCKS\_PER\_KEY: Đây là số trung bình của khối lá dữ liệu trong mỗi giá trị chỉ mục riêng biệt xuất hiện, làm tròn đến số nguyên gần nhất. Với mỗi chỉ mục mà bắt buộc chứa UNIQUE và PRIMARY KEY, giá trị này luôn là 1.
* AVG\_DATA\_BLOCKS\_PER\_KEY: Đây là một số trung bình của khối lá dữ liệu trong bảng được trỏ đến bởi một giá trị riêng biệt trong chỉ mục được làm tròn đến số nguyên gần nhất. Thống kế này là số trung bình của khối dữ liệu chứa chứa hàng có chứa một giá trị nhất định trong các cột chỉ mục.
* NUM\_ROWS: Đây là số hàng trong chỉ mục.

## Yếu tố cụm chỉ số.



Hệ thống thực hiện vào/ra bằng khối dữ liệu. Vì thế, việc tối ưu hóa quyết định để dùng cho việc tìm kiếm toàn bộ bảng dữ liệu bị ảnh hưởng bởi phần trăm các khối đã truy cập, không phải hàng. Khi một phạm vi quét chỉ mục được sử dụng, mỗi chỉ mục được lựa chọn đến một khối trong bảng. Nếu mỗi đầu vào điểm đến một khối khác, các hàng và khối được truy cập là như nhau.

Do đó, số lượng mong muốn của hàng có thể được nhóm lại với nhau trong một vài khối, hoặc chúng có thể được hiện ra trên một số lượng lớn các khối. Điều này được gọi các yếu tố chỉ mục.

Các công thức chi phí tìm kiếm của một loạt chỉ mục sử dụng cấp của B\*-tree, số lượng khối lá, các chỉ số chọn lọc, và các hệ số clustering. Một hệ sô clustering chỉ ra rằng các hàng riêng lẻ nằm tập ở một số khối trong bảng. Môt hệ số Clustering cao chỉ ra rằng các hàng riêng lẻ nằm rải rác ngẫu niên nhiều hơn trên các khối trong bảng.

Do đó, một hệ số clustering cao có nghĩa là chi phí sẽ nhiều hơn việc sử dụng một chỉ số động để quét lấy các hàng bởi ROWID vì nhiều khối trong bảng cần được phải ghé thăm để trả lại dữ liệu. Trong tình huống thực thế, nó xuất hiện mà các hệ số clustering đóng vai trò quan trọng trong việc xác định chi phí của một pham vi quét chỉ mục chi phí của một số phạm vi quét đơn giản bởi vì số khối lá và chiều cao của B\*-tree tương đối nhỏ so với hệ số clustering và tính chọn lọc của bảng

Lưu ý: nếu bạn có nhiều hơn một chỉ mục ở trên bảng, các hệ số clustering cho một chỉ mục có thể là nhỏ trong khi cùng thời điểm đó có thể rất lớn. Một cách tổ chức lại bảng để cải thiện các yếu tố cụm cho mỗi chỉ mục có thể gây ra sự giảm giá trị cho các yếu tố phân nhóm chỉ số khác.

Các yếu tố phân cụm được tính toán và lưu trữ trong cột CLUSTERING\_FACTOR của khung nhìn DBA\_INDEXS khi bạn thu thập số liệu về chỉ mục. Cách tính là tương đối dẽ dàng. Bạn đọc chỉ mục từ trái sang phải, và với mỗi chỉ mục được lập, bạn thêm một vào hệ số clustering nếu các hàng tương ứng nằm trong một khối dữ liệu khác nhau so với các dòng trước đó. Dựa trên thuật toán này, giá trị nhỏ nhất có thể của hệ số clustering là số lượng các khối, và giá trị cao nhất có thể là số hàng dữ liệu. Ví dụ trong slide cho thấy các các hệ số clustering có thể ảnh hưởng đến chi phí. Giả sử các tình huống sau:

Có một bảng gồm 9 hàng, có một chỉ mục không bắt buộc ở cột 1 của bảng, cột c1 hiện lưu trữ các giá trị A, B và C, và bảng chỉ có 3 dữ liệu khối.

* TH1: nếu các dòng trong cùng bảng được sắp xếp sao cho giá trị chỉ số nằm rải rác trên khối của bảng (khác với sắp xếp thành 1 chỗ), hệ số clustering là cao
* TH2: Hệ số clustering là thấp với các hàng mà được sắp xếp thành một chỗ như khối dữ liệu cho cùng giá trị.

Lưu ý: đối với chỉ mục bitmap, hệ số phân cụm không được áp dụng và không sử dụng được.

## Thống kê cột (Colunm Statistics) DBA\_TAB\_COL\_STATISTICS.

* Tính giá trị khác nhau của cột (NUM\_DISTINCT): được sử dụng trong tính toán chọn lọc, ví dụ, 1/số các giá trị khác biệt.
* Giá trị thấp (LOW\_VALUE) và Giá trị cao (HIGH\_VALUE): giá trị dựa trên chi phí giả định phân bố thống nhất của giá trị giữa giá trị thấp và cao cho tất cá các loại dữ liệu. Những giá trị này được sử dụng để xác định phạm vi chọn lọc.
* Số giá trị nulls (NUM\_NULLS): giúp chọn lọc trên các cột có thể null và null (xác định) và không null vị ngữ (thuộc tính).
* Ước tính Tùy biến các giá trị không phổ biến (DESITY) (mật độ): là chỉ liên quan đến biểu đồ. Nó được dùng như dự đoán chọn lọc cho các giá tri không phổ biến. Nó có thể được coi như là xác suất tìm thấy một giá trị cụ thể trong cột này. Việc tính toán phụ thuộc vào loại biểu đồ.
* Số nhóm biểu đồ (NUM\_BUCKETS): là số vùng chứa trong biểu đồ các cột.

## Biểu đồ (Histogram).

Các tối ưu giả định phân bố đồng đều; điều này có thể dễ dấn tới kế hoạch truy cập trong trường hợp dữ liệu lệch

Biểu đồ Histograms:

* Lưu trữ bổ sung thông tin phân bố cột
* Cung cấp cho các ước tính chọn lọc trong trường hợp phân bố không đồng đều.
* Với nguồn tài nguên không giới hạn bạn có thể lưu trữ mỗi giá trị khác nhau và số hàng cho giá trị đó
* Điều này trở nên không thể quản lý cho một số lượng lớn giá trị khác nhau và các tiếp cận khác được sử dụng
* Biểu đồ tần số (Frequency histogram): giá trị khác biệt <= #buckets (bộ chứa))
* Height-balanced histogram (biểu đồ lượng lưu trữ) (#buckets < giá trị khác biệt)

Chúng được lưu trữ trong DBA\_TAB\_HISTOGRAMS

Một biểu đồ ghi lại sư phân bố của các giá trị khác nhau trong một cột, vì vậy lợi ích mang lại tốt hơn ước lượng chọn lọc. Có biểu đồ trên các cột có chứa dữ liệu sai lệch hoặc giá trị với sự khác biệt lớn về số lượng các bản sao giúp tối ưu hóa truy vấn tạo ước tính chọn lọc tốt và đưa ra quyết định tốt hơn về cách sử dụng chỉ mục, nối thứ tự, và các phương pháp nối khác.

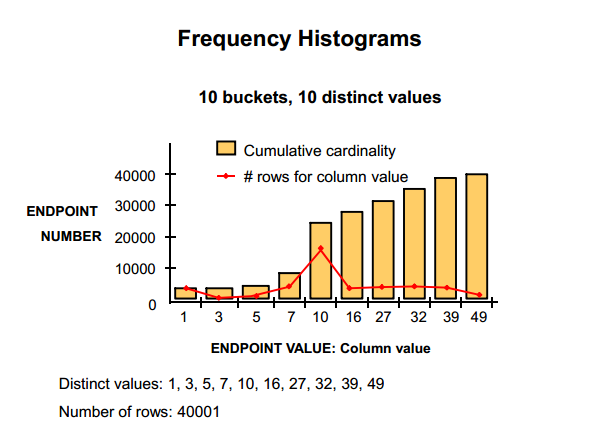
Ngoài histograms, một phân bố đồng đều được giả định. Nếu một biểu đồ có sẵn trên một cột, các ước lượng sử dụng nó thay vì số các giá trị khác biệt.

Khi tạo biểu đồ histogram, Oracle Database sử dụng 2 loại biểu đồ đại diện khác nhau tùy thuộc vào số giá trị khác biệt được tìm thấy trong các cột tương ứng. Khi bạn có một bộ dữ liệu với ít nhất 254 giá trị khác nhau, và số bộ chứa biểu đồ (histogram buckets) không được chỉ định, hệ thống tạo một biểu đồ tần số (frequency). Nếu số giá trị khác nhau là lớn hơn số lượng yêu cầu của vùng biểu đồ, thì hệ thống sẽ tạo một biểu đồ cân bằng (heigth-balanced histogram).

Bạn có thể tìm thông tin về biểu đồ trong các khung nhìn từ điển: DBA\_TAB\_HISTOGRAMS, DBA\_PART\_HISTOGRAMS, và DBA\_SUBPART\_HISTOGRAMS

Lưu ý: Việc thu thập số liệu biểu đồ là hoạt động tốn tài nguyên nhất trong việc thu thập thống kê.

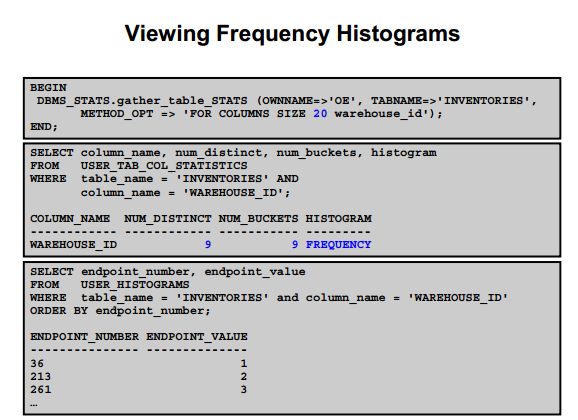
## Biểu đồ tần số (Frequency Histograms).



Đối với ví dụ trong slide (ví dụ trên), giả sử rằng bạn có một cột với số lượng giá trị là 40 001. Bạn chỉ có 10 giá trị khác nhau : 1, 3, 5, 7 ,10, 27, 32, 49. Giá trị 10 là giá trị phổ biến với 16 293 lần xuất hiện.

Khi số lượng yêu cầu của buckets bằng (hoặc cao hơn) số lượng giá trị khác nhau, bạn có thể lưu trữ cho mỗi giá trị khác nhau và ghi lại số liệu thống kê chính xác trong bảng số liệu. Trong trường hợp này, trong DBA\_TAB\_HISTOGRAMS, cột ENPOINT\_VALUE lưu trữ các giá trị cột và cột ENPOINT\_NUMBER lưu trữ số hàng tích lũy gồm giá trị cột, bởi vì điều này có thể tránh một vài tính toán cho phạm vi quét. Số hàng thực tế xuất phát từ các giá trị điểm cuối nếu cần thiết. Số hàng thực tế được thể hiện bởi các đường cong trên slide; chỉ cột ENDPOINT\_VALUE và ENDPOINT\_NUMBER được lưu trữ trong dữ liệu từ điển.

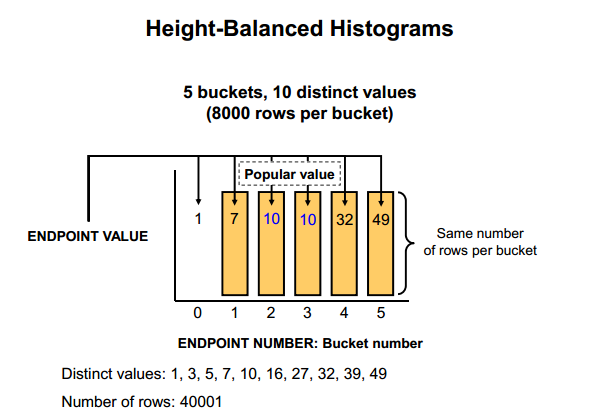
## Khung nhìn biểu đồ tần số.



Các ví dụ trong slide cho chúng ta thấy làm thế nào để xem một biểu đồ tần số. Vì số lượng giá trị khác nhau trong cột WAREHOUSE\_ID của bảng INVENTORIES là 9 và số các vùng yêu cầu là 20, hệ thống sẽ tự động tạo ra một biểu đồ tần số với 9 vùng. Bạn có thể thấy thông tin trong khung nhìn USER\_TAB\_COL\_STATISTICS. Để xem các biểu đồ của mình, bạn có thể truy vấn USER\_HISTOGRAMS (khung nhìn). Bạn có thể thấy cả hai ENPOINT\_NUMBER tương ứng với tần số tích lũy ENPOINT\_VALUE, đại diện trong trường hợp này, giá trị thực tế của cột dữ liệu.

Trong trường hợp này, warehouse\_id là 1 và có 36 hàng với warehouse\_id = 1. Có 177 hàng với warehouse\_id = 2 nên tổng của hàng là (36+177) khá xa so với tần số tích lũy của 213.

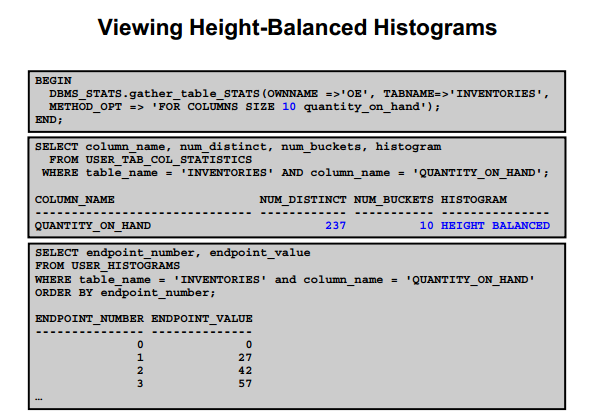
## Biều đồ chiều cao cân bằng (Height-Balance Histogram).



Trong một biểu đồ cân bằng, các giá trị cột được chia thành các dải, sao cho mỗi dải chứa xấp xỉ cùng một số hàng. Biểu đồ cho bạn biết gía trị của điểm cuối của mỗi dải. Trong ví dụ này, giả sử bạn có một cột số liệu với số lượng 40 001 giá trị. Sẽ có 8 000 giá trị cho mỗi dải. Bạn chỉ có 10 giá trị khác nhau: 1, 3, 5, 7, 10, 16, 27, 32, 39, và 49. Giá trị 10 là giá trị phổ biến với 16 293 lần xuất hiện. Khi số buckets ít hơn số lượng giá trị, ENDPOINT\_NUMBER ghi lại số vùng và ENPOINT\_VALUE ghi lại giá trị tương ứng với điểm cuối này. Trong ví dụ này, số hàng mỗi vùng là một phần năm tổng số hàng, đó là 8000. Dựa trên giả định này, giá trị 10 xuất hiện giữa 8 000 và 24 000 lần. Do đó bạn chắc chắn rằng giá trị 10 là một giá trị phổ biến. Đây là một loại biểu đồ tố cho sự cân bằng về giá trị phổ biến, và các phạm vi vị từ.

Số hàng trong một vùng thì không được ghi lại vì điều này có thể bắt nguồn từ tổng số giá trị và thực tế là tất cả các vùng chứa một số lượng tương đương các giá trị. Trong ví dụ này, số 10 là một giá trị phổ biến bởi vì nó mở rộng ra nhiều gía trị điểm cuối. Để tiết kiệm không gian, biểu đồ không thực sự lưu trữ trong vùng lặp. Trong ví dụ ở slide, vùng 2 (với giá trị điểm cuối là 10) sẽ không được ghi lại vào DBA\_TAB\_HISTOGRAMS vì lý do đó.

## Khung nhìn biểu đồ chiều cao cân bằng.



Các ví dụ trong slide cho bạn thấy làm thế nào để xem một biểu đồ cân bằng. Vì số các giá trị khác biệt trong cột QUANTITY\_ON\_HAND của bảng INVENTORIES là 237, và số yêu cầu vùng là 10, hệ thống sẽ tự động tạo một vùng cân bằng gồm 10 buckets. Bạn có thể xem thông tin trong khung nhìn USER\_TAB\_COL\_STATISTICS.

Để xem biểu đồ của riêng mình, bạn có thể truy cập đến khung nhìn USER\_HISTOGRAM. Bạn có thể thấy ENDPOINT\_NUMBER tương ứng với số vùng, và ENDPOINT\_VALUE tương ứng với giá trị điểm cuối kết thúc.

## Biểu đồ đánh giá (Histogram Considerations).

Biểu đồ rất hữu ích khi bạn có các cột dữ liệu phân bố không đồng đều.

Biểu đồ không tốt với:

* Cột mà không xuất hiện trong các câu lệnh WHERE hoặc JOIN.
* Cột phân bố đồng đều.
* Vị từ tương đương với các cột bắt buộc.

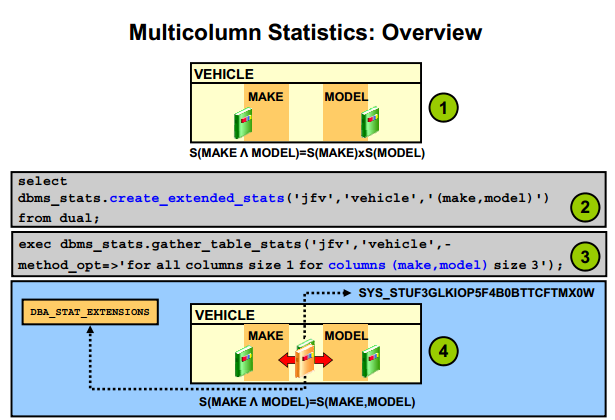
Số lượng tối đa các buckets ít nhất là (254, giá trị khác biệt)

Không sử dụng biểu đồ, trừ khi nó được cải thiện đáng kể hiệu suất.

Dữ liệu trong các cột có thể thay đổi miễn là phân phối vẫn không đổi. Nếu như dữ liệu phân phối của một cột thay đổi thường xuyên, bạn phải tính toán lại biểu đồ của nó thường xuyên. Biểu đồ tiện lợi khi bạn có một giá trị lệch (không đối xứng) cao trong cột mà bạn muốn tạo biểu đồ. Tuy nhiên, có một diều không cần thiết để tạo biểu đồ cho cột mà không xuất hiện khi câu lệnh SQL thực thi. Tương tự như vậy, không cần thiết để tạo biểu đồ cho cột phân bố đồng đều. Ngoài ra, đối với cột khai báo là bắt buộc, biểu đồ là vô ích bởi sự chọn lọc là hiển nhiên. tuy nhiên, giá trị lớn nhất của vùng là 254, có thể nhỏ hơn tùy thuộc vào số lượng thực tế của giá trị cột khác nhau. Biểu đồ có thể ảnh hưởng đến hiệu suất và nên chỉ được sử dụng khi cần cải thiện đáng kể các kế hoạch truy vấn. Đối với dữ liệu phân bố đồng đều, các tối ưu có thể làm cho dự đoán khá chính xác về chi phí thực hiện một xác nhận riêng không sử dụng histogram.

Lưu ý: cột ký tự có một số thành phần riêng biệt là dữ liệu biểu đồ được lưu trữ chỉ cho 32 byte đầu tiên của bất kỳ chuỗi nào

## Thống kê nhiều cột: Tổng quan.

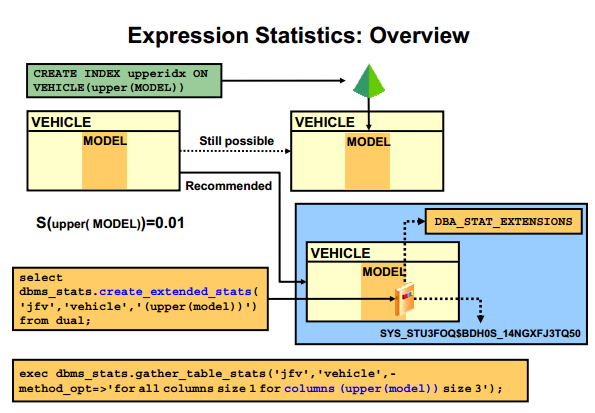


Với Oracle Database 10g, tối ưu truy vấn sẽ đưa vào tài khoản các mối liên kết giữa các cột khi tính chọn lọc của nhiều vị từ trong các trường hợp hạn chế sau:

* Nếu tất cả các cột của một vị ngữ nối tiếp khớp với tất cả cột của khóa chỉ mục nối tiếp, và các vị từ được sử dụng tương đương trong (equijoins) việc kết nối. Sau đó việc tối ưu dùng số lượng khóa khác nhau (NDK) trong chỉ mục để đánh gía và chọn lọc, như 1/NDK.
* Khi DYNAMIC\_SAMPLING được thiết lập đến cấp 4, truy vấn tối ưu dùng các mẫu dữ liệu động (dynamic sampling) để ước tính việc chọn lọc của các vị từ phức tạp liên quan đến nhiều cột trong cùng một bảng. Tuy nhiên, kích thước mẫu là rất nhỏ và làm tăng thời gian phân tích. Một kết quả, mẫu có khả năng không chính xác và có thể gây hại nhiều hơn.

Trong các trường hợp còn lại, việc tối ưu hóa giả định rằng giá trị của cột sử dụng trong một vị từ phức tạp là độc lập với nhau. Nó ước tính chọn lọc của mỗi vị từ nối tiếp bằng cách làm tăng chọn lọc của vị ngữ riêng biệt. Kết quả cách tiếp cận này trong sự đánh giá là khá thấp của sự chọn lọc nếu có một mối liên kết giữa các cột. Để khắc phục vấn đề này, Oracle Database 11g cho phép bạn thu thập, lưu trữ, và sử dụng số liệu thống kê sau đây để nắm lấy sự phụ thuộc hàm giữa 2 hoặc nhiều hơn cột (cũng được gọi là nhóm cột): số giá trị khác nhau, số giá trị null, tần số biểu đồ và mật độ.

## Thống kê biểu thức: Tổng quan.



Các vị từ liên quan đến các biểu thức trên các cột là một vấn đề quan trọng đối với các truy vấn tối ưu. Khi tính toán chọn lọc trên các vị từ của các hàm = hằng số, việc tối ưu giả thiết một giá trị chọn lọc tĩnh chiếm 1%. Phương pháp này gần như không bao giờ có sự lựa chọn chính xác và nó có thể tối ưu hóa để tạo ra các kế hoạch tối ưu.

Các truy vấn tối ưu đã được mở rộng để xử lý tốt hơn các vị từ trong các trường hợp hạn chế, nơi các hàm bảo vệ các đặc điểm phân bố dữ liệu của các cột và do đó cho phép tối ưu hóa để sử dụng các cột thống kê. Một ví dụ về hàm như là TO\_NUMBER.

Việc cải tiến hơn đã được thực hiện để đánh giá việc xây dựng các hàm trong quá trình tối ưu hóa tư vấn để đưa ra sự chọn lọc tốt hơn bằng cách sử dụng mẫu động. Cuối cùng, việc tối ưu thu thập các thống kê trên các cột ảo được tạo ra để hỗ trợ các chỉ mục dựa trên chức năng.

Tuy nhiên, các giải pháp này hoặc được giới hạn ở tầng nhất định nào đó của các hàm hoặc chỉ làm việc cho các biểu thức được sử dụng để tạo ra các chỉ mục dựa trên chức năng. Bằng cách sử dụng các biểu thức thống kê ở trong Oracle Database 11 g, bạn có thể sử dụng một giải pháp tổng quát hơn bao gồm các chức năng người dùng có thể định nghĩa tổng quát hơn và không phụ thuộc vào sự hiện diện của các chỉ mục dựa trên chức năng. Như trong ví dụ trong slide, tính năng này dựa trên cơ sở cột ảo để tạo ra số liệu thống kê hình thức thể hiên của các cột.

## Thu thập số liệu thống kê của hệ thống.

Hệ thống thống kê cho phép tối ưu để xem xét hệ thống I/O và hiệu suất của CPU và việc sử dụng. Đối với mỗi kế hoạch đề xuất, việc tối ưu hóa tính toán cho I/O và chi phí CPU. Điều này rất quan trọng để biết được đặc tính hệ thống để lựa chọn phương án hiệu quả nhất với tỉ lệ tối ưu giữa các I/O và chi phí CPU. Các đặc tính của hệ thống CPU và I/O phụ thuộc vào nhiều yếu tố và không ở lại liên tục trong tất cả thời gian. Sử dụng hệ thống quản lý thống kê một cách thường xuyên, bạn có thể nắm bắt số liệu thống kê trong khoảng thời gian khi hệ thống có khối lượng công việc phổ biến nhất.

Ví dụ, các ứng dụng cơ sở dữ liệu có thể xử lý giao dịch trực tuyến (OLTP) giao dịch trong và chạy OLAP vào ban đêm. Bạn có thể thu thập các thống kê giữa hai trạng thái và kích hoạt riêng biệt thống kê OLTP và OLAP trong trường hợp cần thiết. Điều này cho phép tối tư để tạo ra chi phí liên quan đối với các kế hoạch tài nguyên mà hệ thống sẵn có. Khi hệ thống tạo ra được hệ thống thống kê, nó sẽ phân tích hoạt động hệ thống trong một thời gian nhất định. Không giống như bảng thống kê, chỉ mục hay các cột thống kê, hệ thống không làm vô hiệu các câu lệnh SQL đã được phân tích cú pháp khi hệ thống thống kê được cập nhật. Tất cả các câu lệnh SQL mới được phân tích bằng cách sử dụng số liệu thống kê mới.

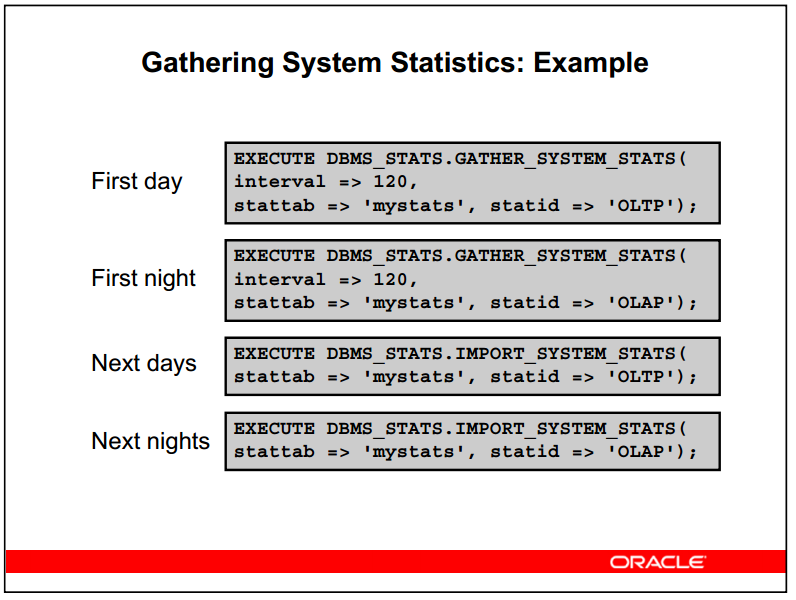
Điều này rất được khuyến khích khi bạn thu thập số liệu thống kê của hệ thống. Số liệu thống kê của hệ thống được thu thập trong một khung thời gian người dùng định nghĩa với DBMS\_STATS.GATHER\_SYSTEM\_STATS. Bạn cũng có thể thiết lập các giá trị hệ thống thống kê một cách rõ ràng bằng DBMS\_STATS.SET\_SYSTEM\_STATS. Sử dụng DBMS\_STATS.SET\_SYSTEM\_STATS để xác minh hệ thống thống kê.

Khi bạn sư dụng các thủ tục GATHER\_SYSTEM\_STATS, bạn nên xác minh tham số GATHERING\_MODE:

* NOWORKLOAD: đây là giá trị mặc định. Chế độ này ghi lại đặc điểm của hệ thống I/O. Việc thu thập có thể mất vài phú và phụ thuộc vào kích thước của cơ sở dữ liệu. Trong thời gian này, hệ thống ước tính thời gian tìm kiếm trung bình và tốc độ chuyển giao cho các hệ thống I/O. Chế độ này thích hợp cho tất cả các khối lượng công việc. Điều đó khuyến cáo bạn chạy GATHER\_SYSTEM\_STAT (noworkload) sau khi bạn tạo cơ sở dữ liệu và không gian bảng.
* INTERVAL: chụp lại hệ thống hoạt động trong một khoảng thời gian xác định. Việc này kết hợp với các thông số thời gian quy định cụ thể số lượng thời gian cho việc ghi lại. Bạn nên cung cấp một giá trị thời gian trong vài phút, sau đó hệ thống thống kê được tạo ra hay cập nhật trong từ điển hay một bảng dàn dựng. Bạn có thể sử dụng GATHERING\_SYSTEM\_STATS (gathering\_mode => ‘STOP’) để dừng thu thập sớm hơn kế hoạch trước đó.
* START|STOP: chụp lại hệ thống hoạt động trong thời gian bắt đầu và kết thúc và làm mới các từ điển hay một bảng dàn dựng với các thống kê cho giai đoạn đã trôi qua.

Chú ý: kể từ Oracle Databse 10g, release 2, hệ thống sẽ tự động thu thập các bộ phận cần thiết của hệ thống thống kê lúc khởi động.

## Ví dụ hệ thống thu thập thống kê.



Ví dụ trong slide cho thấy các ứng dụng cơ sở dữ liệu xử lý giao dịch OLTP trong suốt cả ngày và tiến hành báo cáo vào ban đêm.

Thứ nhất, số liệu thống kê của hệ thống phải được thu thập trong ngày. Trong ví dụ này, thu thập kết thúc sau 120 phút và được lưu trữ trong mảng mystats.

Nói chung, cú pháp trong slide được sử dụng để thu thập số liệu thống kê của hệ thống. trước khi khởi động thủ tục GATHERING\_SYSTEM\_STATS với tham số thời gian chỉ định INTERVAL, bạn phải kích hoạt quá trình làm việc bằng cách sử dụng một lệnh, chẳng hạn như SQL > thay đổi hệ thống thiết lập job\_queue\_processes = 1.

Chú ý: trong Oracle databse 11g Release 2, giá trị mặc định của job\_queue\_processes = 1000. Bạn cũng có thể gọi các thủ tục tương tự với lập luận khác nhau để cho phép thu thập nhãn thay vì sử dụng các công việc.

Nếu thích hợp, bạn có thể chuyển đổi số liệu thống kê giữa các số liệu thu thập được. lưu ý rằng nó có thể tự động hóa quá trình này bằng cách gửi một công việc để cập nhật từ điển với số liệu thống kê thích hợp.

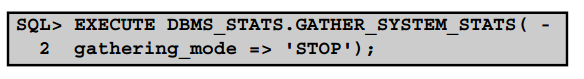
Trong ngày, một công việc có thể nhập số liệu thống kê OLTP cho các hoạt động ban ngày, và trong đêm, một công việc khác sẽ nhập các xử lý phân tích OLAP thống kê trực tuyến cho các hoạt động ban đêm.

Bắt đầu thu thập nhãn thống kê hệ thống trong các dữ liệu từ điển:



Tạo ra các khối lượng công việc.

Kết thúc thu thập số liệu thống kê của hệ thống



Ví dụ trong slide trước đó cho thấy làm thế nào để thu thập số liệu thống kê của hệ thống với công việc bằng cách sử dụng các tham số thủ tục nội bộ DBMS\_STATS.GATHER\_SYSTEM\_STATS. Để thu thập số liệu thống kê của hệ thống một cách thủ công, một tham số thủ tục có thể được sử dụng như trong slide. Đầu tiên, bạn cần phải bắt đầu thu thập số liệu thống kê của hệ thống và bạn có thể kết thúc quá trình thu thập bất cứ lúc nào sau khi bạn chắc chắn rằng một khối lượng công việc đại diện đã được tạo ra trên các ví dụ.

Ví dụ thu thập số liệu thống kê của hệ thống và lưu chúng trực tiếp trong các dữ liệu từ điển.

## Cơ chế thu thập số liệu thống kê.

Orcale Database cung cấp một số cơ chế để thu thập số liệu thống kê. Vấn đề này được thảo luận chi tiết hơn trong các slide tiếp theo. Chúng tôi đề nghị bạn sử dụng việc thu thập số liệu thống kê tự động cho các đối tượng.

Chú ý: khi hệ thống gặp phải một bảng thiếu số liệu thống kê, nó sẽ tự động thu thập các số liệu thống kê cần thiết cho việc tối ưu. Tuy nhiên, đối với một số loại bảng, nó không thực hiện việc lấy mẫu động. chúng bao gồm các bảng điều khiển từ xa và bảng điều khiển bên ngoài. Trong một vài trường hợp và đôi khi việc lấy dữ liệu động bị vô hiệu hóa, việc tối ưu sẽ sử dụng các giá trị mặc định cho các số liệu thống kê của nó.

## Các số liệu thống kê: Tổng quan.

## 

Các tính năng thu thập số liệu thống kê tự động đã được giới thiệu trong Oracle Databse 10g, Release 1 để giảm bớt gánh nặng của việc duy trì tối ưu hệ thống. tuy nhiên, có những trường hợp mà bạn phải vô hiệu hóa nó và chạy những kịch bản của riêng bạn để thay thế. Một lý do là sự thiếu kiểm soát mức đối tượng. Bất cứ khi nào bạn tìm thấy một tập hợp nhỏ các đối tượng mà việc tùy chọn thu thập số liệu thống kê không hoạt động tốt, bạn phải khóa các số liệu thống kê và phân tích chúng một cách riêng biệt bằng cách sử dụng các tùy chọn của riêng bạn. ví dụ, tính năng tự động cố gắng để xác định các mẫu đủ lớn (ESTIMATE\_PERCENT = AUTO\_SAMPLE\_SIZE) không làm việc tốt với các cột chứa dữ liệu với tần số lệch rất cao. Cách duy nhất giải quyết vấn đề này là tự xác định kích cỡ mẫu trong kịch bản của bạn.

Tính năng của các số liệu thống kê tùy chọn trong Oracle Database 11g giới thiệu một cách linh hoạt giúp bạn có thể dựa nhiều hơn vào các tính năng thu thập số liệu tự động để duy trì sự tối ưu trong thống kê khi một số đối tượng yêu cầu việc cài đặt khác nhau từ cơ sở dữ liệu mặc định.

Tính năng này cho phép bạn kết hợp các tùy chọn thu thập số liệu thống kê và ghi đè lên hành vi mặc định của thủ tục GATHER\_\*\_STATS và nhiệm vụ tự động tối ưu thu thập số liệu thống kê vào đối tượng hoặc mức lược đồ. Bạn có thể sử dụng gói DBMS\_STATS để quản lý các tùy chọn thu thập số liệu thống kê trong slide. Bạn có thể thiết lập, lấy, xóa, xuất, nhập các tùy chọn trong bảng, sơ đồ, cơ sở dữ liệu và mức độ toàn cục. các tủy chọn toàn cục được sử dụng cho các bảng không có tùy chọn, trong khi các tùy chọn cơ sở dữ liệu được sử dụng để thiết lập tùy chọn trên tất cả các bảng. các giá trị tùy chọn được xác định trong nhiều cách khác nhau được ưu tiên từ bên ngoài đến bên trong.

Trong hình ảnh ở slide, 3 tùy chọn cuối cùng là mới trong Oracle Database 11g, Release 1:

* CASCADE: thu thập số liệu thống kê với các chỉ số tốt. chỉ số của số liệu thống kê thu thập được không phải là song song.
* ESTIMATE\_PERCENT: là tỉ lệ ước tính hàng sử dụng để tính toán thống kê. (Null means all rows): phạm vi hợp lệ là [0.000001, 100] sử dụng liên tục DBSM.STATS\_AUTO.SAMPLE.SIZE để có hệ thống xác định phù hợp cỡ mẫu cho các thống kê tốt. đây là mặc định khuyến khích.
* NO\_INVALIDATE: kiểm soát việc hủy bỏ hiệu lực của con trỏ phụ thuộc của các bảng mà số liệu thống kê được thu thập. nó không làm mất hiệu lực của con trỏ phụ thuộc nếu đặt là TRUE.

Thủ tục làm mất hiệu lực ngay lập tức con trỏ phụ thuộc nếu đặt FALSE. Sử dụng DBMS.STATS\_AUTO.INVALIDATE để có hệ thống quyết định khi nào để làm mất hiệu lực của con trỏ phụ thuộc. đây là mặc định.

* PUBLISH: được sử dụng để quyết định xem có công bố số liệu thống kê vào từ điển hay để lưu trữ trong một khu vực được cấp phát trước.
* STALE\_PERCENT: được sử dụng để xác định mức ngưỡng của một đối tượng được coi là số liệu thống kê cũ. Giá trị bằng tỉ lệ phần trăm của hàng sửa đổi kể từ khi số liệu thu thập được thông qua. Ví dụ thay đổi mặc định 10% lên đến 13% của SH.SALES.
* DEGREE: xác định mức độ song song được sử dụng để tính toán thống kê. Mặc định của mức độ là null, có nghĩa là sử dụng các giá trị bảng mặc định quy định bởi DEGREE trong CREATE TABLE hay ALTER TABLE. Sử dụng các DBMS.STATS\_DEFAULT.DEGREE liên tục để xác định các giá trị mặc định dựa trên thông số khởi tạo. giá trị AUTO\_DEGREE xác định mức độ song song tự động. đây là hoặc 1 hoặc DEFAULT\_DEGREE, tùy thuộc vào kích thước của đối tượng.
* METHOD\_OPT: là một chuỗi SQL được sử dụng để thu thập số liệu thống kê biểu đồ. Giá trị mặc định là FOR ALL COLUMNS SIZE AUTO.

GRANULARITY: là mức độ chi tiết của các số liệu thống kê thu thập cho các bảng phân vùng.

INCREMENTAL được sử dụng để thu thập số liệu thống kê toàn cục về bảng phân vùng một cách gia tăng.

Điều quan trọng cần lưu ý rằng bạn có thể thay đổi giá trị mặc định cho các thông số trên bằng cách sử dụng thủ tục DBMS\_STATS.SET\_GLOBAL\_PREFS

Lưu ý: bạn có thể mô tả tất cả thiết lập ưu tiên thống kê hiệu quả cho tất cả các bảng có liên quan bằng cách sử dụng DBA\_TAB\_STAT\_PREFS.

## Khi nào tiến hành thu thập số liệu thống kê thủ công.

Dựa chủ yếu vào sự thu thập thống kê tự động.

Thay đổi tần suất của việc thu thập số liệu thống kê tự động tùy thuộc vào nhu cầu của bạn,

Hãy nhớ rằng STATISTICS\_LEVEL nên được thiết lập là TYPICAL hoặc ALL cho sự thu thập số liệu thống kê tự động để làm việc đúng cách.

Thu thập số liệu thống kê thủ công dành cho:

* Đối tượng không ổn định.
* Đối tượng bị thay đổi trong hoạt động thực thi. Thu thập số liệu thống kê như là một phần của các hoạt động hàng loạt.
* Bảng ngoài, hệ thống thống kê, các đối tượng cố định.
* Các đối tượng mới: thu thập số liệu thống kê ngay sau khi tạo đối tượng.

Cơ chế thu thập số liệu thống kê tự động là thu thập số liệu thống kê về đối tượng trong cơ sở dữ liệu thống kê vắng mặt hoặc đã cũ. Điều đó rất quan trọng để xác định khi nào và bao lâu để thu thập số liệu thống kê mới. khoảng thời gian mặc định là thu thập hàng đêm, nhưng bạn có thể thay đổi khoảng thời gian này cho phù hợp với nhu cầu kinh doanh của bạn. bạn có thể làm như vậy bằng cách thay đổi các đặc tính của cửa sổ bảo trì của bạn. một số trường hợp có thể yêu cầu nhãn hiệu thống kê tập hợp. ví dụ số liệu thống kê trên bảng được thay đổi đáng kể theo ngày có thể trở nên cũ. Thông thường có hai loại đối tượng như sau:

* Bảng không ổn định là loại được sửa đổi đáng kể trong ngày.
* Đối tượng là mục tiêu của việc tải dữ liệu lớn trong đó tăng thêm 10% hoặc nhiều hơn tổng kích thước của đối tượng giữa các khoảng thời gian thu thập số liệu thống kê.

Đối với các bảng bên ngoài, thống kê chỉ được thu thập bằng tay bằng cách dùng GATHER\_TABLE\_STATS. Lấy mẫu trên bảng ngoài không được hỗ trợ, vì vậy tùy chọn ESTIMATE\_PERCENT nên được thiết lập một cách rõ ràng bằng null. Bởi vì các thao tác dữ liệu không cho phép chống lại bảng bên ngoài, nó là đủ để phân tích các bảng bên ngoài khi các tập tin tương ứng thay đổi. các khu vực khác, trong đó thống kê cần phải thu thập thủ công là những hệ thống thống kê và các đối tượng cố định, chẳng hạn như các bảng hiệu suất năng động. các thống kê này không được tự động thu thập.

## Thu thập số liệu thống kê thủ công.

Bạn có thể sử dụng Entersprise Manager và gói DBMS\_STATS để:

* Tạo và quản lý số liệu thống kê cho việc tối ưu:

Thu thập/ sửa.

Xem/ tên.

Export/ import.

Xóa/ khóa.

* Thu thập số liệu thống kê về:

Chỉ số, bảng, cột, vách ngăn.

Đối tượng, lược đồ hoặc cơ sở dữ liệu.

* Thu thập số liệu thống kê hoặc nối tiếp hoặc song song.
* Thu thập, cài đặt hệ thống thống kê.

Cả Enterprise Manager và DBMS\_STATS cho phép bạn tự tạo và quản lý số liệu thống kê một cách tối ưu. Bạn có thể sử dụng gói DBMS\_STATS để thu thập, chỉnh sửa, xem, import, export, khóa, xóa và thống kê. Bạn cũng có thể sử dụng phần mềm này để nhận dạng hoặc thu thập tên của số liệu thống kê. Bạn có thể thu thập số liệu thống kê về chỉ số, bảng, cột và các phân vùng khác nhau ở mức độ chi tiết: đối tượng, lược đồ và các mức cơ sở dữ liệu.

DBMS\_STATS thu thập chỉ số thống kê cần thiết để tối ưu hóa, nó không thu thập số liệu thống kê khác. Ví dụ các bảng thông kê được thu thập bởi DBMS\_STATS bao gồm số lượng hàng, số lượng các khối đang chứa dữ liệu và chiều dài hàng trung bình, nhưng không phải số lượng hàng bị kiểm soát, không gian tự do trung bình, hoặc số lượng các khối dữ liệu không sử dụng.

Chú ý không sử dụng COMPUTE VÀ ESTIMATE của của câu lệnh ANALYZE để thu thập số liệu thống kê. Những điều khoản được hỗ trợ hoàn toàn tương thích và có thể được loại bỏ trong một phiên bản tương lai. Các gói DBMS\_STATS thu thập chính xác hơn và rộng hơn về số liệu thống kê, thu thập và thống kê hiệu quả hơn. Bạn có thể tiếp tục sử dụng báo cáo phân tích cho các mục đích khác không liên quan đến việc tối ưu thu thập số liệu thống kê:

* Để sử dụng VALIDATE hoặc LIST CHAINED ROWS.
* Để thu thập thông tin về khối danh sách miễn phí.

## Thu thập số liệu thống kê thủ công: Các yếu tố.

Giám sát các đối tượng cho DMLs.

* Xác định chính xác các kích thước mẫu.
* Xác định mức độ song song.
* Xác định các biểu đồ nên được sử dụng.
* Xác định các hiệu ứng tầng trên các chỉ số.
* Thủ tục để sử dụng trong DBMS\_STATS:

– GATHER\_INDEX\_STATS

– GATHER\_TABLE\_STATS

– GATHER\_SCHEMA\_STATS

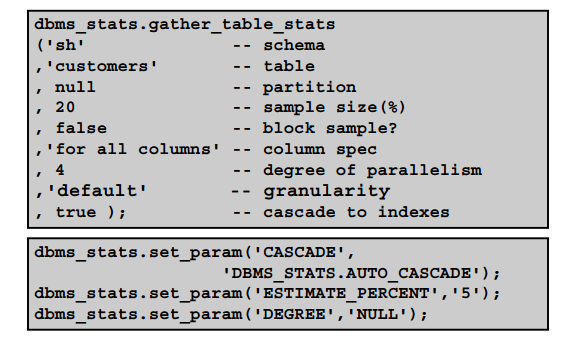
– GATHER\_DICTIONARY\_STATS

– GATHER\_DATABASE\_STATS

– GATHER\_SYSTEM\_STATS

Khi bạn tối ưu việc thu thập số liệu thống kê, bạn phải đặc biệt lưu ý đến các yếu tố sau:

* Giám sát các đối tượng cho hàng loạt ngôn ngữ thao tác dữ liệu (DML) hoạt động và thống kê tập hợp nếu cần thiết.
* Xác định các kích thước mẫu chính xác.
* Xác định mức độ song song để tăng tốc độ truy vấn trên các đối tượng lớn.
* Xác định các biểu đồ được tạo ra trên các cột có dữ liệu sai lệch.
* Xác định những thay đổi trên các đối tượng tầng của bất kì chỉ số phụ thuộc nào.



Ví dụ đầu tiên sử dụng gói DBMS\_STATS để thu thập số liệu thống kê trên bảng CUSTOMERS của giản đồ SH. Nó sử dụng một số các tùy chọn được thảo luận trong các slide trước đó.

Bạn có thể sử dụng các thủ tục SET\_PARAM trong DBMS\_STATS để thiết lập các giá trị mặc định cho các thông số của tất cả các thủ tục DBMS\_STATS. Ví dụ thứ hai trong slide cho thấy việc sử dụng này. Bạn cũng có thể sử dụng hàm GET\_PARAM để lấy các giá trị mặc định hiện tại của một tham số.

Lưu ý: mức độ chi tiết của các số liệu thống kê để thu thập là thích hợp chỉ khi các bảng được phân chia. Tham số này xác định mà tại đó thống kê mức độ cần được thu thập. điều này có thể ở phân vùng, subpartition hoặc cấp bảng.

## Tối ưu hóa việc lấy mẫu động: Tổng quan.

Lấy mẫu động có thể được thực hiện cho các bảng và lập chỉ mục:

* Nếu không có số liệu thống kê.
* Không thể tin tưởng người thống kê.

Được sử dụng để xác định các số liệu thống kê chính xác hơn khi ước tính.

* Bảng yếu tố.
* Vị từ chọn lọc.

Tính năng được kiểm soát bởi:

* Tham số OPTIMIZER\_DYNAMIC\_SAMPLING
* Tham số OPTIMIZER\_FEATURES\_ENABLE
* Gợi ý DYNAMIC\_SAMPLING
* Gợi ý DYNAMIC\_SAMPLING\_EST\_CDN

Lấy mẫu động cải thiện hiệu suất máy chủ bằng cách xác định các ước tính chọn lọc và yếu tố chính xác hơn, cho phép tối ưu để tạo ra các kế hoạch thực hiện tốt hơn. Ví dụ, mặc dù nó được khuyến khích để bạn thu thập số liệu thống kê trên tất cả các bảng của bạn được sử dụng bởi các CBO, bạn có thể không thu thập số liệu thống kê cho bảng tạm tời của bạn là bàng làm việc được sử dụng để thao tác với các dữ liệu trung gian. Bạn có thể sử dụng các mẫu động để:

* Ước tính vị từ chọn lọc của các bảng đơn khi số liệu thống kê đã thu thập có thể không được sử dụng hoặc có khả năng dẫn đến sai sót đáng kể trong dự toán.
* Ước tính bảng yếu tố cho các bảng và các chỉ số liên quan mà không cần số liệu thống kê hoặc bảng mà các thống kê quá lỗi thời để tin cậy vào.

Bạn kiểm soát mẫu động với các tham số khởi tạo OPTIMIZER\_DYNAMIC\_SAMPLING. DYNAMIC\_SAMPLING và DYNAMIC\_SAMPLING\_EST\_CDN có thể được sử dụng để kiểm soát việc lấy mẫu động.

Lưu ý: các tham số khởi tạo OPTIMIZER\_FEATURES\_ENABLE sẽ dừng lấy mẫu động nếu thiết lập một phiên bản trước 9.2.

## Tối ưu hóa lấy mẫu động trong công việc.

Lấy mẫu động được thực hiện tại thời điểm biên dịch

* Nếu câu truy vấn có ích từ việc lấy mẫu động:

Một câu lệnh SQL đệ quy được thực thi để lấy mẫu dữ liệu.

Số lượng các khối lấy mẫu phụ thuộc vào tham số khởi tạo OPTIMIZER\_DYNAMIC\_SAMPLING

* Trong quá trình lấy mẫu động, các vị từ được áp dụng để lấy mẫu được xác định có chọn lọc.
* Sử dụng mẫu động khi:

Thời gian lấy mẫu là một phần nhỏ trong thời gian thực hiện.

Câu truy vấn được thực hiện nhiều lần.

Bạn tin rằng có một kế hoạch tốt hơn có thể được tìm thấy.

Các thuộc tính hiệu năng sơ cấp là thời gian biên dịch. Hệ thống xác định tại thời điểm biên dịch cho dù một truy vấn sẽ được hưởng lợi từ việc lấy mẫu động. nếu vậy một câu lệnh SQL đệ quy được ban hành để quét một mẫu nhỏ ngẫu nhiên của các khối của bảng, và áp dụng các bảng duy nhất có liên quan vị từ để ước tính các vị tự chọn lọc.

Tùy thuộc vào các giá trị của các tham số khởi tạo OPTIMIZER\_DYNAMIC\_SAMPLING, một số lượng nhất định của các khối được đọc bởi các truy vấn lấy mẫu động.

Đối với một truy vấn thường được hoàn thanh một cách nhanh chóng (trong ít hơn một vài giây), bạn không muốn gánh chịu chi phí lấy mẫu động. tuy nhiên việc lấy mẫu động có thể mang lại những lợi ích sau:

* Kế hoạch tốt hơn có thể được tìm thấy bằng cách sử dụng mẫu động.
* Thời gian lấy mẫu là một phần nhỏ trong tổng thời gian thực hiện cho truy vấn.
* Các truy vấn được thực hiện nhiều lần.

Chú ý: việc lấy mẫu động có thể được áp dụng cho một tập con của của các bảng vị từ đơn và kết hợp với các ước tính chọn lọc tiêu chuẩn của các vị từ mà viêc lấy mẫu động không được thực hiện.

## OPTIMIZER\_SAMPLING\_GATHERING.

Phiên động hoặc tham số hệ thống:

* Có thể được đặt giá trị từ 0 đến 10.
* 0 sẽ dừng việc lấy mẫu động.
* 1 tất cả mẫu của các bảng sẽ không được phân tích, nếu một bảng không được phân tích.

Được liên kết đến các bảng khác hoặc xuất hiện trong một subquery hoặc nonmergeable view.

Không có chỉ số.

Có nhiều hơn 32 khối.

* 2 tất cả mẫu của các bảng không được phân tích.
* Các giá trị cao hơn các ứng dụng tích cực hơn của việc lấy mẫu.
* Lấy mẫu động được lặp lại nếu không có hoạt động cập nhật.

Bạn kiểm soát được việc lấy mẫu động với tham số OPTIMIZER\_DYNAMIC\_SAMPLING, có thể được đặt giá trị từ 0 đến 10. Một giá trị bằng 0 có nghĩa là việc lấy mẫu động không được thực hiện.

Giá trị 1 có nghĩa là việc lấy mẫu động được thực hiện trên tất cả các bảng sẽ không được phân tích nếu các tiêu chí sau đây không được đáp ứng:

* Có ít nhất một bảng không được phân tích trong truy vấn.
* Bảng không được phân tích này được kết hợp vào 1 bảng khác hoặc xuất hiện trong 1 subquery hay view nonmergeable.
* Bảng không được phân tích này không có chỉ mục.
* Bảng không được phân tích này có khối nhiều hơn số lượng mặc định của các khối được sử dụng để lấy số liệu động của bảng này. Giá trị mặc định là 32.

Giá trị mặc định là 2 nếu OPTIMIZER\_FEATURES\_ENABLE được thiết lập là 10.0.0 hoặc cao hơn. ở cấp độ này, hệ thống áp dụng lấy mẫu động cho tất cả các bảng không được phân tích. Số khối lấy được lấy mẫu là 2 lần giá trị mặc định của lấy mẫu động (32).

Tăng giá trị của kết quả tham số trọng ứng dụng của việc lấy mẫu động, cả các loại bảng mẫu (phân tích hoặc không được phân tích) và số lượng I/O dành cho việc lấy mẫu.

Lưu ý: việc lấy mẫu động được lặp lại nếu không có hàng đã được đưa vào, đã xóa hoặc đã cập nhật trong bảng được lấy mẫu từ các hoạt động mẫu trước đó.

## Khóa các thống kê.

Ngăn ngừa trước việc thu thập tự động.

Được sử dụng chủ yếu cho các bảng không ổn định.

- Khóa không thống kê bao hàm việc lấy mẫu động.

BEGIN

DBMS\_STATS.DELETE\_TABLE\_STATS('OE','ORDERS');

DBMS\_STATS.LOCK\_TABLE\_STATS('OE','ORDERS');

END;

- Khóa thống kê sử dụng cho các giá trị đại diện

BEGIN

DBMS\_STATS.GATHER\_TABLE\_STATS('OE','ORDERS');

DBMS\_STATS.LOCK\_TABLE\_STATS('OE','ORDERS');

END;

- Tham số FORCE ghi đè việc khóa các thống kê

SELECT stattype\_locked FROM dba\_tab\_statistics;

Bắt đầu với Oracle Database 10g, bạn có thể khóa các thống kê trong một bảng được chỉ định với phương thức LOCK\_TABLE\_STATS của gói DBMS\_STATS. Bạn có thể khóa các thống kê trong một bảng không có thống kê hoặc thiết lập chúng thành NULL sử dụng phương thức DELETE\_\*\_STATS để đón trước một cách tự động tập các thống kê từ đó bạn có thể sự dụng các mẫu động trong một bảng không ổn định không có thống kê. Bạn có thể khóa các thống kê trong một bảng không cố định tại một điểm khi nó được định cư một cách hoàn toàn do đó các bảng thống kê sẽ biểu thị rõ ràng hơn của các bảng định cư.

Bạn cũng có thể khóa các thống kê tại mức lược đồ bằng việc sử dụng phương thức LOCK\_SCHEMA\_STATS.

Bạn có thể truy vấn cột STATTYPE\_LOCKED trong khung nhìn {USER|ALL|DBA}\_TAB\_STATISTICS để xác định kể cả các thống kê trong bảng đã bị khóa.

Bạn có thể dùng phương thức UNLOCK\_TABLE\_STATS để mở khóa các thống kê trong bảng chỉ định.

Bạn có thể thiết lập giá trị của tham số FORCE thành TRUE để ghi đè các thống kê kể cả nếu chúng được khóa. Đối số FORCE được tìm thấy trong phương thức DBMS\_STATS dưới đấy:

DELETE\_\*\_STATS, IMPORT\_\*\_STATS, RESTORE\_\*\_STATS, và SET\_\*\_STATS.

Chú ý: Khi bạn khóa các thống kê trong một bảng, tất cả các thống kê phụ thuộc cũng được xem xét để khóa. Điều này bao gồm bảng thống kê, cột thống kê, biểu đồ tần số và các thống kê chỉ số phụ thuộc.

## Khôi phục các thống kê.

Các thống kê trước đó có thể được khôi phục với phương thức:

DBMS\_STATS.RESTORE\_\*\_STATS

BEGIN

DBMS\_STATS.RESTORE\_TABLE\_STATS(

OWNNAME=>'OE', TABNAME=>'INVENTORIES',

AS\_OF\_TIMESTAMP=>'15-JUL-10 09.28.01.597526000 AM -05:00');

END;

Các thống kê được lưu trữ tự động

- Với nhãn thời gian trong DBA\_TAB\_STATS\_HISTORY

- Khi thu thập với phương thức DBMS\_STATS

Các thống kê được thanh lọc

- Khi STATISTICS\_LEVEL được thiết lập thành TYPICAL hay ALL một cách tự động.

- Sau 31 ngày hay thời gian được định nghĩa bởi

DBMS\_STATS.ALTER\_STATS\_HISTORY\_RETENTION

Các phiên bản cũ của các thống kê được lưu một cách tự động khi các thống kê trong từ điển được điều chỉnh với phương thức DBMS\_STATS. Bạn có thể khôi phục các thống kê sử dụng phương thức RESTORE trong gói DBMS\_STATS. Những phương thức này sử dụng một nhãn thời gian như một tham số để khôi phục các thống kê theo các nhãn thời gian. Điều này hữu dụng khi các thống kê mới thu thập được dẫn đến các kế hoạch thực thi tối ưu và người quản trị muốn quay trở lại thiết lập trước của các thống kê. Chú ý: câu lệnh ANALYZE không khôi phục các thống kê cũ.

Có khung nhìn từ điển có thể sử dụng để ác định nhãn thời gian cho các lần khôi phục của các thống kê.

Khung nhìn \*\_TAB\_STATS\_HISTORY nhìn (ALL, DBA, hay USER) bao gồm lịch sử của các lần điều chỉnh thống kê của các bảng. Ví dụ trong slide nhãn thời gian được xác định bởi:

select stats\_update\_time from dba\_tab\_stats\_history

where table\_name = 'INVENTORIES'

Cơ sở dữ liệu thanh lọc các thống kê cũ một cách tự động một cách tự động dựa trên thiết lập duy trì lịch sử thống kê vf thời gian của các phân tích gần đây của hệ thống. Bạn có thể duy trì cấu hình sử dụng phương thức BMS\_STATS.ALTER\_STATS\_HISTORY\_RETENTION. Giá trị mặc định là 31 ngày, điều này có nghĩa là bạn có thể khôi phục các thống kê tối ưu bất kì lúc nào trong 31 ngày qua.

## Xuất khẩu và nhập khẩu các thống kê.

Sử dụng phương thức DBMS\_STATS:

- CREATE\_STAT\_TABLE để tạo bảng thống kê.

- EXPORT\_\*\_STATS di chuyển các thống kê vào bảng thống kê.

- Sử dụng Data Pump để di chuyển bảng thống kê.

- IMPORT\_\*\_STATS di chuyển thống kê vào từ điển dữ liệu.

Bạn có thể xuất khẩu và nhập khẩu thống kê từ từ điển dữ liệu vào các bảng của người dùng sở hữu (user-owned), cho phép bạn tạo nhiều phiên bản các thống kê của cùng một lược đồ. Bạn cũng có thể sao chép các thống kê từ một cơ sở dữ liệu vào một cơ sở dữ liệu khác. Bạn cũng có thể muốn làm điều này để sao chép các thống kê từ một cơ sở dữ liệu ban đầu vào một cơ sở dữ liệu nhở hơn để kiểm tra.

Trước khi xuất khẩu các dữ liệu, đầu tiên bạn cấn tạo một bảng để lưu giữ liệu thống kê. Phương thức DBMS\_STATS.CREATE\_STAT\_TABLE để tạo bảng thống kê. Sau khi bảng được tạo, bạn có thể xuất khẩu các thống kê từ một từ điển dữ liệu vào một bảng thống kê sử dụng phương thức DBMS\_STATS.EXPORT\_\*\_STATS.

Bạn có thể nhập khẩu thống kê sử dụng phương thức DBMS\_STATS.IMPORT\_\*\_STATS.

Bộ tối ưu không sử dụng các thống kê được lưu trữ trong bảng của người sử dụng. Thống kê duy nhất được sử dụng bởi bộ tối ưu là các dữ liệu được lưu trong từ điển dữ liệu. Để bộ tối ưu sử dụng được các thống kê trong các bảng của người dử dụng, bạn phải nhập khẩu những thống keek đó vào từ điển dữ liệu sử dụng các phương thức nhập khẩu dữ liệu.

Để di chuyển thống kê từ một bảng vào một bảng khác, đầu tiên bạn phải xuất khẩu các thống kê trong cơ sở dữ liệu đầu tiên, sau đó sao chép bảng dữ liệu thống kê vào bảng thứ hai, sử dụng tiện ích Data Pump Export và Import hay các kỹ thuật khác, và cuối cùng nhập khẩu các thống kê vào bảng thứ hai.

## QUIZ – Câu đố.

Khi không có các thống kê cho một đối tượng được sử dụng trong câu lệnh SQL, bộ tối ưu sử dụng:

a. Các tối ưu dựa trên luật

b. Các mẫu động

c. Các giá trị cố định

d. Các thống kê thu thập được trong giai đoạn phân tích

e. Giá tri ngẫu nhiên

Đáp án: b, c

Việc tối ưu hóa phụ thuộc vào các số liệu thống kê chính xác để tạo ra các kế hoạch thực thi tốt nhất. Việc thu thập các thống kê tự động (AGS) không thu thập thống kê cho tất cả mọi thứ. Những đối tượng nào yêu cầu bạn phải thu thập một cách thủ công?

a. Các bảng ngoài

b. Từ điển dữ liệu

c. Các đối tượng cố định

d. Các bảng biến động

e. Các thống kê hệ thống

Đáp án: a, c, e

Có một bảng rất biến động trong cơ sở dữ liệu. Kích thước của bảng thay đổi nhiều hơn 50% hàng ngày. Các bước nào là một phần của phương thức để lấy mẫu động một cách mạnh mẽ (force):

a. Xóa các thống kê

b. Khóa các thống kê

c. Thu thập các thống kê khi bảng ở trạng thái lớn nhất

d. Thiết lập DYNAMIC\_SAMPLING = 9

e. Thiết lập DYNAMIC\_SAMPLING = 0.

f. Cho phép tham số DYNAMIC\_SAMPLING thành mặc định.

Đáp án: a, b, f

**TỔNG KẾT:**

Trong bài học này, bạn cần phải học được:

- Thu thập các thống kê một cách tối ưu.

- Thu thập các thống kê hệ thống.

- Thiết lập các ưu tiên thống kê.

- Sử dụng thu thập mẫu động.

- Các thao tác thống kê tối ưu.

Phần luyện tập nằm trong các phần sau:

- Sử dụng thống kê hệ thống.

- Sử dụng thu thập thống kê tự động.