THIẾT KẾ VÀ ĐIỀU CHỈNH DỮ LIỆU MỨC VẬT LÝ

Lời khuyên cho người khách hàng mà phàn nàn về lỗ dột trên mái nhà dột ướt bàn ăn đó là: hãy chuyển bàn đi nơi khác”

Kiến trúc sư Frank Lloyd Wright

Câu lệnh của một DBMS trên các truy vấn thường gặp và các hoạt động cập nhật điển hình là thước đo cuối cùng của một thiết kế cơ sở dữ liệu. 1 DBA có thể cải tiến câu lệnh bằng cách điều chỉnh 1 vài thông số trong DBMS(ví dụ: kích thược của vùng nhớ đệm hoặc tầng số của thực thi của checkpoint) hay là tìm ra những giới hạn về khả năng đáp ứng của phẩn cứng(bottlenecks) sau đó ta sẽ thêm những phần cứng để đáp ứng cho vận hành hệ thống.tuy nhiên bước trên hết, để có thể thực hiện câu lệnh tốt thì thiết kế dữ liệu cần phải tốt.

sau khi thiết kế lược đồ ở mức quan niệm(conceptual and external) là tạo ra những quan hệ và những khung nhìn cùng với những ràng buộc toàn vẹn, ta phải xác định mục đích câu lệnh trong thiết kế mức vật lý(thiết kế lược đồ vật lý). Khi người dùng yêu cầu tăng cao, thiết kế lược đồ mức vật lý rất cần thiết để hoà hợp, điều chỉnh, tất cả các khía cạnh của 1 thiết kế cơ sở dữ liệu cho câu lệnh thực thi tốt.

mục đích của chương này sẽ được tóm gọn như sau, chúng tôi đưa ra 1 cái nhìn chung về thiết kế và tinh chỉnh ở mức vậy lý trong phần 16.1. điều quan trọng nhất của thiết kế mức vật lý là giải quyết vấn đề liên quan đến index. Chúng tôi đưa ra những nguyên tắc cho việc quyết định lựa chọn tạo indexs trong phần 16.2. những nguyên tắc này được minh hoạ trong 1 số ví dụ và những vấn đề chuyên sâu hơn trong phần 16.3 đến 16.6. trong phần 16.3 chúng tôi đưa ra những ví dụ làm nổi bật những cách lựa chọn index cơ bản. trong phần 16.4 chúng ta xem kỹ hơn những vấn đề gom nhóm (clustering); ta bàn luận làm thế nào để chọn index gom nhóm và có lưu các bộ từ các quan hệ khác nhau gần với chúng không. Trong phần 16.5 ta bàn về việc dùng index với hỗn hợp hoặc multiple-attribute search keys . trong phần 16.6 ta nhấn mạnh làm cách nào lựa chọn tốt index có thể cho phép nhiều câu queries được trả lởi mà không cần xem những dữ liệu có thật( actual data records).

Trong phần 16.7 ta xem tổng quát vấn đề chính của tinh chỉnh dữ liệu. hơn nữa việc tinh chỉnh index, ta có thể phải tinh chỉnh lược đồ mức quan niệm, cũng như thường xuyên dùng câu query và những định nghĩa khung nhìn( view definitions) . ta bàn làm thế nào để tinh lọc lại lược đồ mức quan niệm trong phần 16.8 và làm thế nào để tinh chỉnh lại câu queries và khung khùng nhìn trong phần 16.9. ta bàn luận ngắn gọn sự ảnh hưởng lớn về các câu lệnh thực thi xảy ra đồng thời trong 16.10. và kết thúc chương với bàn luận ngắn về những tiêu chuẩn(benchmarks) về DBMS trong 16.11; tiêu chuẩn lựa chọn DBMS để giúp thực thi tốt các câu lệnh.

***Công cụ thiết kế vật lý****: cho đến nay RDBMSs (relational Database Management System) đã cung cấp vài công cụ để hỗ trợ việc thiết kế và tinh chỉnh lược đồ mức vật lý. MS SQL server có công cụ tinh chỉnh mà hỗ trợ về tạo index; nó cũng hỗ trợ xoá bỏ 1 index. IBM DB2 V6 cũng có công cụ tinh chỉnh và Oracle Expert hỗ trợ về thêm, xoá index.*

**16. 1 giới thiệu về thiết kế mức vật lý.**

Giống như các khía cạnh của các mức thiết kế dữ liệu khác, thiết kế vật lý phải phụ thuộc phù hợp với những dữ liệu có ngoài đợi thực và những dụng ý sử dụng nó. Một cách riêng biệt, nó rất quan trọng để biết được những công việc điển hình cần phải làm(typical workload) mà cơ sở dữ liệu phải cung cấp cho người dùng. Wordload bao gồm các câu query hoặc update dữ liệu . điều tất nhiên là người dùng sẽ yêu cầu về tốc độ của câu query, udate phải nhanh bao nhiêu, hay là có bao nhiêu transaction phải thực thi mỗi giây. Miêu tả của workload và yêu cầu về thực thi của người dùng là nhưng cái yêu cầu cơ bản mà mà cần phải giải quyết trong thiết kế mức vật lý.

Để tạo 1 dữ liệu mức vật lý tốt và để tinh chỉnh hệ thống cho câu lệnh trong việc đáp trả lại yêu cầu của người dùng, người thiết kế cần phải hiểu công việc của 1 DBMS, đặc biệt là về chỉ mục và cách thực thi câu query của DBMS đó. Nếu dữ liệu mong đợi được truy cập đồng thời bởi nhiểu ngưởi dùng hoặc là 1 cơ sở dữ liệu phân tán(distributed database), nhiệm vụ trở nên phức tạp rất nhiều và các tính năng khác của DBMS được dùng đến. ta bàn luận ảnh hưởng lớn về truy cập đồng thơi trên dữ liệu được thiết kế trong 16.10.

**16.1.1 database workloads**

1 workload description bao gồm các thành phần sau:

* 1 danh sách các query và tần số sử dụng chúng, như 1 phần nhỏ của tất cả query và update.
* 1 danh sách các câu lệnh update và tần số sử dụng nó.
* Mục đích câu lệnh cho mỗi loại query và update.

Mỗi câu query trong workload, ta phải xác định:

* Những quan hệ nào được dùng đến.
* Những thuộc tính nào được xuất ra (trong lệnh SELECT).
* Những thuộc tính nào có điều kiện lựa chọn hoặc là có điều kiện kết trong lệnh WHERE và chọn lựa những điều kiện này như thế nào.

Giống như trên, mỗi lệnh update trong workload, ta phải xác định

* Những thuộc tính nào có điều kiện lựa chọn hoặc là điều kiện kết (trong lệnh where) và chọn lựa những điều kiện này như thế nào.
* Loại của câu lệnh update náy (insert , delete, hay update) và quan hệ cần update.
* Các trường mà bị điểu chỉnh lại trong lệnh update này.

Nên nhớ rằng các câu lệnh query và update thường có các đối số, ví dụ thao tác về sổ ghi nợ hay tín dụng thì sẽ gọi đến mã số tài khoản. giá trị của những đối số này xác định độ chọn lọc của điều kiện lựa chọn và điều kiện kết.

Lệnh update có 1 thành phần query mà được dùng đề tìm các bộ cần update. Thành phần này có thể giúp ích cho thiết kế mức vật lý tốt và sự có mặt của index. Mặt khác, lệnh update thường yêu cầu thêm việc đề duy trì index trên các thuộc tính mà chúng cần chỉnh. Vì vậy, trong khi các lệnh query chỉ đươc lợi từ sự có mặt của 1 index, thì 1 index có thể hoặc tăng tốc độ truy vấn hoặc giảm tốc độ cho lệnh update. Người thiết kế cần lưu ý vấn đề này khi tạo index.

**16.1.2 giải quyết việc thiết kế và tinh chỉnh**

Những giải quyết quan trong suốt quá trình thiết kế và tinh chỉnh dữ liệu bao gồm các bước sau:

1. Những index nào được tạo.
   * Những quan hệ nào được chỉ mục và những thuộc tính nào hoặc tập hợp thuộc tính nào được chọn làm khoá index để tìm.
   * Mỗi index, nó nên được gom nhóm hay không gom nhóm(unclustered), nó có nên được nhiều hay thưa thớt
2. Ta có nên thay đổi lược đồ mức quan niềm để cải thiện truy vấn không. Ví dụ, ta phải xét điều sau:
   * Thay đổi lược đồ bình thường hoá: ta luôn có nhiều cách để phân tích 1 lược đồ thành dạng chuẩn mong muốn(BCNF hoặc 3NF). Lựa chọn có thể được dựa vào tiêu chuẩn thực thi.
   * Sự không bình thường hoá (denormalization): ta có thể muốn xem xét lại phân rã lược đồ đã lấy đi sự bình thường hoá trong suốt quá trình thiết kế lược đồ mức quan niệm để cải tiến câu lệnh query mà gọi các thuộc tính từ các quan hệ bị phân rã trước đây.
   * Sự phân chia thẳng đứng: ở những tình huống dĩ nhiên ta có thể muốn phân rã các quan hệ thành nhiều quan hệ nhỏ hơn nữa để tăng tốc độ thực thi của query do chỉ thực hiện trên ít thuộc tính.
   * Khung nhìn: ta có thể tạo nhiều khung nhìn để che đi những thay đổi trong lược đồ quan hệ không cho người dùng biết.
3. Có nên thường xuyên thực thi query và transaction được ghi lại nhiều lần để tôc độ nhanh hơn không.

Trong dữ liệu song song hoặc phân tán có nhiều lựa chọn còn xét đến ví dụ như có nên chia quan hệ qua các vị trí khac nhua không hoặc có nên lưu bản sao chép quan hệ ở nhiều địa điểm khác nhau không.

**16.1.3 sự cần thiết của tinh chỉnh dữ liệu**

Một cach chính xác, thông tin chi tiết của workload có thể khó để nhận biết được trong khi bắt đầu thiết kế của hệ thống. do đó, điều chỉnh dữ liệu sau khi nó thiết kế và triển khai thì rất quan trọng- ta phải tinh chỉnh lại thiết kế ban đầu trong đúng với cách diễn tả của đời thực để thu được những câu lệnh thực thi tốt nhất có thể.

Sự khác biệt giữa thiết kết dữ liệu và tinh chỉnh dự liệu là 1 vài thứ không bị bó buộc. ta có thể xét quy trình thiết kế hơn 1 lần lược đồ ban đầu ở mức khái niệm được thiết kế và tập hợp các chỉ mục và giải quyết vấn đề gom nhóm được thực hiện. bất cứ sự thay đổi xảy ra sau khi đã tạo lược đồ mức quan niệm hoặc chỉ mục rồi thì ta có thể gọi đó là việc tinh chỉnh. Như một lựa chọn, ta có thể xét 1 vài cách tinh chỉnh lược đồ mức quan niệm là 1 phần của quá trình thiết kế vật lý.

Ranh giới giữa việc thiết kế và tinh chỉnh không cần thiết phải rõ ràng ( nó không quan trọng), và ta sẽ bàn luận 1 cách đơn giản vấn để lựa chọn chỉ mục và tinh chỉnh dữ liệu mà không cần quan tâm đến khi nào việc tinh chỉnh được thực hiện.

**16.2 hướng dẫn cho lựa chọn index**

Để quyết định chỉ mục nào được dùng thì trước tiên ta sẽ làm việc với danh sách query mà ban đầu ta đã tìm. Một cách rõ ràng, chỉ những quan hệ mà tham gia vào câu query thì mới được xét đến để xác định làm chỉ mục, và việc lựa chọn thuộc tính để tạo chỉ mục trên được hướng dẫn bởi các lệnh điều kiện xuất hiện trong lệnh WHERE của câu query, 1 index mà thích hợp với lược đồ thì sẽ cải tiến rất lớn cho tốc độ truy vấn.

1 cách lựa chọn chỉ mục là phải xét những câu query quan trọng, và mỗi câu query đó phải xác định chỉ mục thích hợp mà nằm trong danh sách chỉ mục hiện có.sau đó ta xét có có nên đi đến 1 kế hoạch tốt hơn với những chỉ mục khác không; nếu những chỉ mục thêm vào này mà là ứng cử bao hàm trong danh sách chỉ mục của ta. nhìn chung, phạm vi thu hồi sẽ được giúp bởi chỉ mục B+, và sự thu hổi chính xác sẽ được giúp bởi 1 chỉ mục kém. Việc gom nhóm sẽ giúp ích cho dãy câu query, và nó sẽ giúp các query chính xác nếu 1 vài thực thể có chứa cùng khoá.

Tuy nhiên Trước khi thêm 1 chỉ mục vào danh sách, t phải xét xem chỉ mục này có ảnh hưởng đến lệnh update trong danh sách ta liệt kê trong workload không. Nếu ta chú ý sớm, mặc dù 1 chỉ mục có thể tăng tốc thành phần truy vấn trong lệnh update nhưng tất cả chỉ mục trên 1 thuộc tính được update – trên tất cả thuộc tính, trong trường hợp insert, deletes phải được update bất cứ khi nào giá trị thuộc tính thay đổi. vì thế, thình thoảng ta phải xét sự hại của việc làm chậm lệnh update và có lợi cho truy vấn.

Rõ ràng hơn, việc chọn 1 bộ chỉ mục thích hợp đáp ứng yêu cầu trong workload yêu cầu phải hiểu kỹ thuật index có sẵn, và hiểu hoat động query. Những nguyên tắc dưới đây cho phần lựa chọn chỉ mục được tóm gọn như sau:

Guideline 1 (whether to index): quan điểm rõ ràng thường là điều quan trọng nhất. không xây dựng chỉ mục trừ khi nhiều câu truy vấn - bao gồm cả những thành phần query trong lệnh update- sẽ có ích cho chúng. Bất cứ khi nào có thể, chọn những chỉ mục mà làm tốc độ cho nhiều câu query.

Guideline 2 ( choice of search key): những thuộc tính được đề cập trong lệnh WHERE là những ứng cử viên lựa chọn làm chỉ mục.

* + 1 điều kiện lựa chọn đề nghị rằng ta nên xét 1 chỉ mục trên những thuộc tính đã được chọn.
  + 1 dãy điều kiện lựa chọn đề nghị rằng ta nên xét đến chỉ mục B+ tree (hoặc ISAM) trên những thuộc tính được lựa chọn. 1 chỉ mục B+ tree luôn thích hợp hơn 1 chỉ mục ISAM. 1 chỉ mục ISAM có thể có giá trị nếu quan hệ này thường xuyên phải update, nhưng ta sẽ thừa nhận rằng 1 chỉ mục B+ tree luôn luôn là lựa chọn tốt hơn chỉ mục ISAM.

Guideline 3 (multiple-attribute search keys): những chỉ mục với khoá tìm kiếm đa thuộc tính nên được xét đến tron 2 tình huống sau:

* + Lệnh WHERE chứa các điều kiện có hơn 1 thuộc tính trong quan hệ.
  + Chúng chỉ cho phép chiến lược định giá chỉ mục ( i.e.., việc truy cập quan hệ có thể được tránh) cho những câu query quan trọng. (tình huống này có thể dẫn đến những thuộc tính đang là khoá tìm kiếm thậm chí chúng không xuất hiện trong lệnh WHERE).

K khi tạo những chỉ mục trên các khoá tìm kiếm với nhiều thuộc tính, nếu dãy query được mmong đợi , hãy cẩn thận với những thuộc tính trong khoá tìm kiếm để thực hiện query.

Guideline 4 (whether to cluster): ở hầu hết 1 chỉ mục trong 1 quan hệ có thể được gom nhóm, và việc gom nhóm làm ảnh hưởng lớn đến lệnh thực thi; vì thế lựa chon gom nhóm chỉ mục rất quan trọng.

* + Như 1 luật thống trị, dãy query thì giống như làm lợi cho đa số việc gom nhóm. Nếu 1 vài dãy query được đặt ra trên quan hệ, việc bao gồm các tập hợp thuộc tính khác nhau, xét đến tính chọn lọc của những câu query và tần số liên kết của chúng trong workload khi quyết định chọn chỉ mục nào gom nhóm.
  + Nế u 1 chỉ mục cho phép chiến lược ước lượng chỉ 1 chỉ mục cho câu query thì được có dụng ý để tăng tốc độ, chỉ mục không cần gom nhóm. (chỉ mục chỉ có ý nghĩa quan trọng khi chỉ mục đó được dùng để lấy ra những bộ từ quan hê).

Guideline 5 (hast versus tree index): 1 chỉ mục B+ tree thì luôn được chuộng hơng bởi vì nó phục vụ cho dãy các query tốt như các query ngang bằng. 1 chỉ mục linh tinh thì tốt hơn trong trường hợp sau:

* + Chỉ mục