|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ tên** | **MSSV** | **Hình ảnh** | **Ghi chú** |
| 1 | Hà Thị Phương Thảo | 0912430 | Description: 0912430 |  |
| 2 | Nguyễn Thị Thanh Thảo | 0912431 | Description: 0912431 |  |
| 3 | Trương Nguyễn Thủy Tiên | 0912463 | Description: 0912463 |  |
| 4 | Nguyễn Văn Tiến | 0912469 | Description: 0912469 | Nhóm trưởng |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Nội dung công việc** | **Thành viên 1**  **0912430** | **Thành viên 2**  **0912431** | **Thành viên 3**  **0912463** | **Thành viên 4**  **0912469** |
| 1 | Dịch tài liệu |  |  |  |  |
| 2 | Tổng hợp kết quả, viết tài liệu |  |  |  |  |

**TỐI ƯU TRUY VẤN VÀ CÁC LỰA CHỌN**

Mục lục

[I. Quá trình truy vấn và tối ưu 4](#_Toc309861610)

[II. Lợi ích của tính năng tối ưu trong hệ thống cơ sở dữ liệu. 4](#_Toc309861611)

[1. Thay đổi ( hoặc viết lại) truy vấn. 4](#_Toc309861612)

[2. Quan sát sơ đồ thực thi truy vấn 4](#_Toc309861613)

[3. Tần số 5](#_Toc309861614)

[4. Những lời gợi ý sơ đồ thực hiện truy vấn 5](#_Toc309861615)

[5. Tối ưu sâu. 5](#_Toc309861616)

[III. Ước lượng chi phí truy vấn – ví dụ 5](#_Toc309861617)

[IV. Phát triển sơ đồ thực thi câu truy vấn. 10](#_Toc309861618)

[1. Nguyên tắc biến đổi sơ đồ truy vấn 11](#_Toc309861619)

[2. Thuật toán sắp xếp lại sơ đồ thực thi truy vấn. 11](#_Toc309861620)

[V. Nhân tố lựa chọn, kích thước bảng, và định lượng chi phí truy vấn. 12](#_Toc309861621)

[1. Việc ước lượng độ chọn lọc cho 1 phép chọn hoặc vị từ. 12](#_Toc309861622)

[2. Biểu đồ tần số. 13](#_Toc309861623)

[3. Ước lượng đồ chọn lọc cho phép kết. 13](#_Toc309861624)

1. **Quá trình truy vấn và tối ưu**

Các bước cơ bản của quá trình truy vấn:

* Duyệt tìm, phân tích ngữ pháp và phân rã của 1 câu truy vấn: bước này kiểm tra cú pháp của truy vấn, nếu cú pháp sai sẽ thông báo lỗi cho người dùng. Đầu ra của bước này là 1 dạng trung gian của truy vấn được biết dưới dạng cây truy vấn hoặc sơ đồ thực thi truy vấn.
* Tối ưu truy vấn:bước này bao gồm tối ưu cục bộ và toàn cục. tối ưu toàn cục xác định loại kết và loại lựa chọn và phép chiếu liên quan với những phép kết. tối ưu cục bộ xác định phương pháp chỉ mục cho phép chọn và phép kết.
* Tạo ra mã truy vẫn và xử lý: bước này dùng ngôn ngữ lập trình và trình biên dịch để tạo mã xử lý.

1. **Lợi ích của tính năng tối ưu trong hệ thống cơ sở dữ liệu.**
2. **Thay đổi ( hoặc viết lại) truy vấn.**

Các quản trị dữ liệu sẽ viết lại các truy vấn để nó trở thành dạng truy vấn có hiệu suất hơn trước khi qua tối ưu. Điều này giúp sự lựa chọn sơ đồ thực thi rất nhiều. Ví dụ các phép biến đổi thường dùng là: thay đổi các thành phần truy vấn thành các phép kết hoặc 1 phần là phép kết, việc đẩy các phép gom nhóm xuống dưới phép kết, việc loại trừ phép kết trên các khoá ngoại khi các bảng bao gồm các kết quả của phép kết mà không được dùng nhiều trong truy vấn, thay đổi kết ngoài thành kết trong khi chúng cho ra các kết quả như nhau, và thay thế view reference thành actual view.

1 phép biến đổi thường dùng nữa là viết lại khung nhìn cụ thể. Nếu 1 vài phần của truy vấn mà tương đương với 1 khung nhìn cụ thể thì truy vấn đó được thay thế bằng khung nhìn đó..

Biến đổi truy vấn rát quen thuộc trong kho dữ liệu sử dụng dạng lược đồ sao.

1. **Quan sát sơ đồ thực thi truy vấn**

Làm thế nào bạn biết được sơ đồ nào được chọn cho những câu truy vấn thường dùng nhất trong cơ sở dữ liệu của bạn.? hầu hết các hệ quản trị đều cung cấp 1 vài công cụ cho người dùng hiểu rõ được sơ đồ truy cập (access plan). Công cụ này miêu tả tất cả các bước trong sơ đồ, thứ tự các bảng nào được truy cập trong câu truy vấn, và có cần dùng chỉ mục cho bảng này hay không. Khi đó sẽ lựa chọn sơ đồ nào tốt nhất.

1. **Tần số**

Nhiều hệ thống cơ sở dữ liệu lưu tần số xuất hiện của dãy các giá trị thuộc tính từ dữ liệu thực để giúp ước lượng độ chọn lọc của phép chọn và kết.

1. **Những lời gợi ý sơ đồ thực hiện truy vấn**

Song song với giải thích, những gợi ý của sơ đồ trở thành có giá trị nhất định trong công nghiệp cơ sở dữ liệu, cho phép người lập trình ứng dụng lựa chọn sơ đồ nào thích hợp, loại bỏ những trường hợp không chăc chắn. những lời gợi ý là cẩm nang chỉ dẫn người lập trình trong câu truy vấn SQL mà có thể làm thay đổi sơ đồ thực thi của truy vấn đó. Chúng được cung cấp bởi tất cả các hệ thống cơ sở dữ liệu.

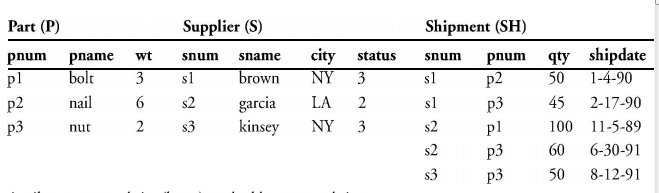
1. **Tối ưu sâu.**

Những quản trị dữ liệu khác nhau có tìm kiếm sâu khác nhau, đơn giản nhất là tìm kiếm kỹ.

1. **Ước lượng chi phí truy vấn – ví dụ**

Giả sử ta có 1 sơ đồ thực thi 1 câu truy vấn đơn giản, và ta tập trung vào định lượng chi phí truy vấn về thời gian truy xuất bé nhất để xử lý câu truy ấn mà sơ đồ của nó được để ra. Nó minh hoạ lợi ích của việc dùng chiến lược tối ưu truy vấn để giải quyết các vấn đề đơn giản ở ngoài đời thực. mặc dù vấn đề này còn cơ bản nhưng nó có lợi ích rất lớn trong thời gian truy vấn mà có thể được xem như là những nguyên tắc, và định nghĩa của những nguyên tắc này được minh hoạ rất rõ.

Ta giả sử có 3 bảng cơ sở dữ liệu được thể hiện cụ thể trong các bảng: part, supplier, intersection .



Tên và kích thước của thuộc tính , các bảng như sau:

* Supplier: snum(5), sname(20), city(10), status(2) 🡪 37 bytes trong 1 record của supplier.
* Part: pnum(8), pname(10), wt(5) --> 23 bytes trong 1 record của part
* Shipment: snum(5), pnum(8), qty(5), shipdate(8) 🡪 26 bytes trong 1 record trong shipment.

Chú ý: giả sử kích thước của block: 15000 bytes.

1. Ví dụ câu truy vấn cho cơ sở dữ liệu trê: những names nào của part được cung cấp bởi nhà cung cấp trong thành phố New York? Ta có truy vấn:

SELECT p.pname

FROM p, sh, s

WHERE P.PNUM = SH.PNUM

AND sh.snum = s.snum

AND s.city = ‘NY’;

* S kết SH kết P (1)
* SH kết S kết P (2)
* P kết SH kết S (3)
* SH kết P kết S (4)
* S X P kết SH (5)
* P X S kết SH (6)

Có 6 cách kết theo thứ tự kết khác nhau của: P, SH, S. phép kết ‘S kết SH kết P’ (1) và phép ‘ SH kết S kết P’ (2) thì tương đương nhau vì phép kết có tính chất giao hoán giữa 2 vế với nhau. Tương tự (3), (4) cũng tương đương nhau, (5) (6) tương đương nhau. Ta nên tránh lệnh (5) (6) nếu có thể vì chúng chứa các bộ cartesian của phép kết khi có sự chồng các cột. khi điều này xảy ra , kích thước của bảng kết quả sẽ tăng lên rất nhiều. và dữ liệu trong bảng kết quả cũng bị nối một cách tuỳ ý.

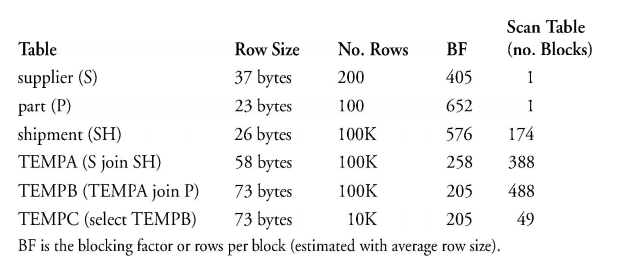
Xem lệnh (1) và (3) ta nên xét nên làm phép kết trước hay là pháp chọn trước. ta cũng xét nên tạo chỉ mục hay không. Giờ ta ước lượng chi phí cho 4 cách sau:

* Cách 1A: dùng lệnh 1 với kết trước, chọn sau và không dùng chỉ mục.
* Cách 1B: dùng lệnh 1 với chọn trước, kết sau, không dùng chỉ mục.
* Cách 3A: dùng lệnh 3 với kết trước, chọn sau, không có chỉ mục.
* Cách 3B: dùng lệnh 3 với chọn trước, kết sau, không có chỉ mục.

Bằng trực giác ta thấy cách 1B cải tiến dựa trên 1A vì phép kết là phép tốn chi phí nhiều nhất, nếu ta giảm kích thước (bằng cách dữ liệu trong bảng trước khi thực hiện phép kết thì sẽ làm tốc độ thức thi nhanh hơn.

Giờ ta sẽ dùng sequential block accesses (SBA) và random block accesses(RBA) như dụng cụ ước lượng thời gian I/O vì có 1 mối quan hệ tuyến tính giữa SBA hoặc RBA với thời gian I/O , và thời gian I/O thì thường là dạng được chấp nhận trong việc so sánh chi phí các lệnh trong cơ sở dữ liệu.

Ta tóm tắt kích thước bộ, số lượng dữ liệu, và số lượng block được thống kê dưới bảng sau.



* **Ước lượng chi phí của cách 1A**

Điều này được nói rõ hơn ở hình 3.1: dùng lệnh truy vấn. lệnh truy vấn này chỉ rõ phép kết được thực thi trước tiên, sau đó mới đến phép chọn. ta dùng phép kết merge, vì thế không có chỉ mục được dùng trong truy vấn này. Trong mỗi bước ta ước lượng số lượng record được xét đến trong khoang thời gian được xét, sau đó tính tổng SBA. Ta giả sử mỗi bảng được lưu trên đĩa sau mỗi thao tác lệnh và lệnh kế tiếp phải được đọc từ đĩa lên.

* Bước 1: kết bảng S và SH qua thuộc tính chung là snum tạo ra bảng TEMPA(snum, sname, city, status, qty, shipdate) với kích thước mỗi dòng là 58 byte. Nếu ta sắp xếp bảng trước khi thực thi phép kết thì chi phí cho cách Merge sort là 2 X nb X logM nb [O’Ncil 2001, Silbcrschatz 2006] với nb là số block (trang) trong 1 bảng. ví dụ này M = 3. Tuy nhiên trong trường hợp này, ta không cần sắp xếp SH từ bảng S rất nhỏ, nhỏ hơn cả 1 block, vì thế ta chỉ cần duyệt SH và so sánh trong bộ nhớ nhanh với 1 block của S.

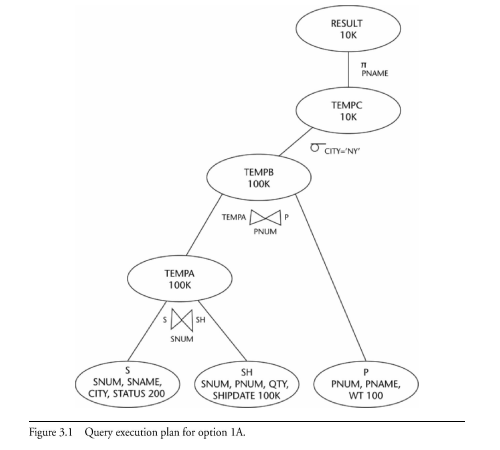
Số lượng block được truy cập:

= read S + read SH + write TEMPA

= ceiling(200/405) + ceiling(100K/576) + ceiling(100K/258).

= 1 + 174 + 388

= 563



* Bước 2: kết P và TEMPA bằng cột pnum thành TEMPB.

Số lượng block truy cập: 5091

* Bước 3: chọn TEMPB ở city = ‘NY’ thành TEMPC ở 73byte mỗi hàng. Giả sử NY có 10% là supplier.

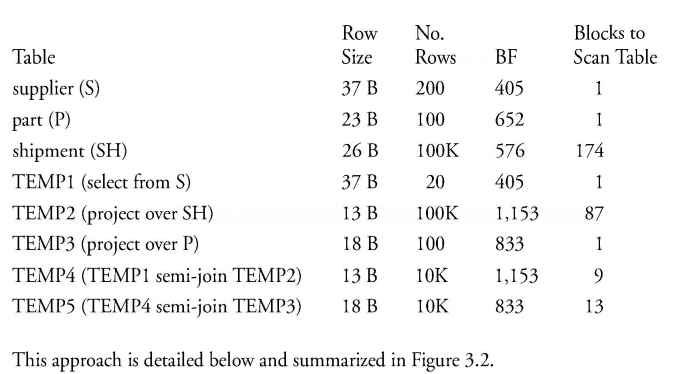
Số lượng block truy cập: 537

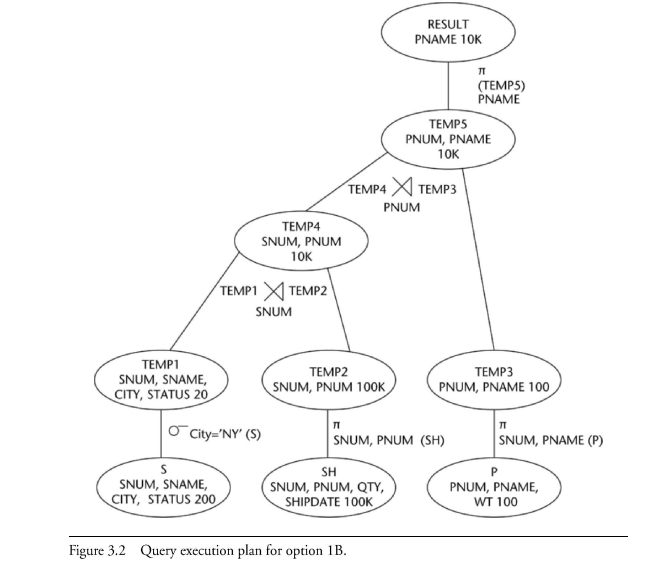
* Bước 4: chiếu TEMPC lấy thuộc tính pname ra RESULT có 10B mỗi hàng.

Số lượng hàng: 10K

Số lượng block: 49.

* **Tổng cộng ta có số lượng block truy cập trong truy vấn theo cách 1A: 6240**
* **Ước lượng chi phí trong cách 1B: làm phép chọn trước( bao gồm cả phép chiếu) phép kết,** **không có chỉ mục.**

****

****

**Cách tính theo hình 3.2 ta được số lượng block truy cập: 398.**

* **So sánh 2 cách ta có:**

**Cách 1A: 6240 xử lý joins trước**

**Cách 1B: 398 xử lý kết sau.**

Rất rõ ràng để thấy giảm số bộ trong bảng thì làm tăng tốc độ rất nhiều.

1. **Phát triển sơ đồ thực thi câu truy vấn.**

Một sơ đồ thực thi truy vấn là 1 cấu trúc dữ liệu mà trình bày các thao tác trên cở sở dữ liệu( phép chọn, phép chiếu, phép kết) như 1 điểm riêng biệt. thứ tự các thao tác trong sơ đồ thực thi truy vấn có thể được thể hiện bằng cách từ trên xuống hoặc từ dưới lên trên. Ta dùng cách tiếp cần từ dưới lên, và trong hình 3.1 và 3.2 những ví dụ kinh điển của việc dùng sơ đồ thực thi truy vấn để thể hiện thứ tự có thể xảy ra của các thao tác cần thực thi trong câu truy vấn SQL. 1 câu truy vấn SQL có thể có nhiều thứ tự thực thi truy vấn khác nhau, phụ thuộc vào đồ phức tạp của câu truy vấn đó, và mỗi thứ tự có thể được trình bày bởi 1 sơ đồ thực thi riêng biệt. mục đích của ta là tìm ra sơ đồ thực thi truy vấn nào mà nó đưa ra câu trả lời chính xác trong khoảng thời gian ngắn nhất.

Một thuật toán phổ biến để tối ưu truy vấn hiện nay đó là thuật toán nói ở phần III: đặt thao tác chọn và chiếu trước khi kết vì kết tốn nhiều chi phí. Chỉ nên làm việc với phép kết với các bảng kích thước càng bé càng tốt.

Để thuận tiện thay đổi 1 sơ đồ thực thi truy vấn từ việc sắp xếp ngẫu nhiên các thứ tự thao tác sang thứ tự có phương pháp mà phép chọn và chiếu được thực thi trước và phép kết sau thì ta tóm tắt ngắn gọn những nguyên tắc sau ( có thể xem chúng như một thuật toán):

1. **Nguyên tắc biến đổi sơ đồ truy vấn**

Dưới đây là những nguyên tắc hiển nhiên trong việc biến đổi sơ đồ thực thi truy vấn để đảo ngược thứ tự nhưng vẫn cho ra kết quả giống nhưa. Cho phép những cây truy vấn khác nhau thì tạo ra kết quả giống nhau.

* + - **nguyên tắc 1:** tính giao hoán của phép kết: R1 kết R2 = R2 kết R1.
    - **Nguyên tắc 2**: tính kết hợp của phép kết: R1 kết (R2 kết R3) = (R1 kết R2) kết R3.
    - **Nguyên tắc 3**: thứ tự các phép chọn trong 1 bảng không làm ảnh hưởng đến kết quả truy vấn.
    - **Nguyên tắc 4**: không cần quan tâm đến thứ tự phép kết và phép chiếu trên cùng 1 bảng. miễn là 1 phép chiếu không loại bỏ 1 thuộc tính được dùng trong phép chọn.
    - **Nguyên tắc 5**: phép chọn trên 1 bảng trước phép kết cho ra kết quả như nhau với cách chọn sau khi kết trên bảng đó.
    - **Nguyên tắc 6**: thứ tự phép chiếu và phép kết chứa các thuộc tính giống nhau thì không cần quan tâm đến thứ tự trong truy vấn miễn là các thuộc tính loại bỏ trong phép chiếu không được chứa trong phép kết.
    - **Nguyện tắc 7**: phép chọn hoặc phép chiếu và phép hội liên quan đến cùng bảng thì không cần phân biệt thứ tự thao tác.

1. **Thuật toán sắp xếp lại sơ đồ thực thi truy vấn.**

Dưới đây là 1 khám phá đơn giản đề sắp xếp lại sơ đồ truy vấn để nó tối ưu hoặc gần tối ưu.

* + - Một phép chọn riêng biệt với 1 vài mệnh đề AND thành những phép chọn có thứ tự.
    - Đặt các phép chọn xuống sâu trong sơ đồ nếu có thể, để thao tác chọn được thực thi trước.
    - Nhóm 1 thứ tự các phép chọn nếu có thể.
    - Đẩy phép chiếu xuống sâu trong sơ đồ nếu có thể.
    - Nhóm các phép chiếu trong cùng 1 bảng, loại bỏ phép chiếu dư thừa.

1. **Nhân tố lựa chọn, kích thước bảng, và định lượng chi phí truy vấn.**

1 lần chúng ta có được 1 sơ đồ thực thi truy vấn được đề xuất, ta cần phải ước lượng kích thước những bảng trung gian mà câu truy vấn tối ưu đó sẽ tạo ra trong suốt quá trình thực thi. 1 lần ta ước lượng tổng số kích thước những bảng đó , ta có thể tính toán được thời gian I/O cần dùng để thực thi truy vấn này.

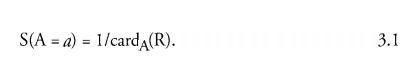
Đề dễ hiểu ta định nghĩa các đo lường sau trong bảng dữ liệu:

* + - Số lượng hàng của bảng R: card(R)
    - Số lượng các giá trị khác nhau của thuộc tính A trong bảng R: cardA(R )
    - Giá trị lớn nhất của A trong bảng R: maxA(R )
    - Giá trị nhỏ nhất của A trong bảng R: minA(R )

1. **Việc ước lượng độ chọn lọc cho 1 phép chọn hoặc vị từ.**

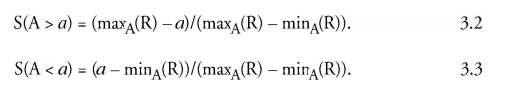
Mối quan hệ dưới đây cho biết làm thế nào tính độ chọn lọc của các phép chọn trong truy vấn SQL.

Độ chọn lọc của 1 thuộc tính A trong bảng R phải có 1 giá trị rõ ràng trong 1 record được chọn ra ở 2 tình huống. một là, nếu thuộc tính A là khoá chính, nơi mà mỗi giá trị chỉ xác định duy nhát, sau đó ta đo độ chọn lọc:



Ví dụ, 1 bảng có 50 record, khi đó độ chọn lọc là 1/50.

Mặt khác, nếu thuộc tính không phải là khoá và mỗi giá trị a xảy ra nhiều trong bảng thì phép tính như sau:



Độ chọn lọc của 2 phép chọn chéo nhau trên cùng 1 bảng thì đực tính:

S(P and Q) = S(P) X S(Q) 3.4

Với P và Q là vị tự.

1. **Biểu đồ tần số.**

Ta tính toán độ chọn lọc bằng cách tính độ trung bình dựa trên tính đúng đắn của dữ liệu, nhưng mỗi dữ liệu khác nhau có thể gây ra số lượng sai khác lớn, nếu tất cả cơ sở dữ liệu chỉ dùng cách tính xấp xỉ này thì ước lượng thời gian truy vấn có thể làm sai lệch nghiêm trọng. nhưng may mắn , nhiều hệ thống quản lý cơ sở dữ liệu hiện nay thường lưu lại các giá trị thuộc tính có thực dưới dạng sơ đồ tần số. trong sơ đồ tần số, các giá trị thuộc tính được chia thành các dãy, và trong mỗi dãy, sẽ thể hiện số lượng các hàng mà thuộc thuộc tính đó có. Sơ đồ tần số rất giúp ích cho việc ược lượng thời gian truy vấn tốt hơn, trung thực, khách quan hơn.

1. **Ước lượng đồ chọn lọc cho phép kết.**

Ước lượng độ chọn lọc cho 1 phép kết rất khó nếu nó dựa vào 1 thuộc tính không khoá, trong trường hợp xấu nhất có thể rơi vào trường hợp tạo ra tích Cartesian. Ta chỉ bàn ước lượng phép kết dựa trên thuộc tính khoá và khoá ngoại. ví dụ, phép kết giữa bảng R1(bảng chứa khoá) với bảng R2( chứa khoá ngoại):



S là độ chọn lọc cho thuộc tính chung dùng trong phép kết, tức là thuộc tính đó là 1 khoá.