SQL Việt blog

Thảo luận về lập trình và quản trị Microsoft SQL Server

Ứng Dụng của Join Bất Cân Bằng

Guess Post: Vũ Minh Tâm

Tiếp theo bài viết Cần thận với Join bất cân bằng, bài viết này mô tả một trường hợp thường gặp có thể xử lý bằng Join bất cân bằng (còn được gọi bằng Non Equi-Joins). Đó là những bài toán liên quan đến cách tính bậc thang.

Ví dụ giá điện dùng trong khoảng 0-100 thì áp mức giá 100 cho 1 số, từ 101 – 200 lại áp mức giá 200. Cần tính tiền cho từng khách hàng với số điện dùng trong tháng.

Hoặc trong viễn thống, ấp giá cho cuộc gọi từ 0-10 giây đầu là 1 đồng, từ giây thứ 11 là 2 đồng... Trong bưu chính, khối lượng hàng hóa từ 0-200g thì mức giá là 1000 đồng, từ 200-400g là 2000 đồng...

Để giải quyết bài toán này có nhiều cách, một trong số những cách thức hiệu quả là sử dụng Join bất cân bằng vì những ưu điểm sau:

- Xử lý hàng loạt, không sử dụng Cursor
- Có thể cài đặt bằng SQL, thuận tiên khi triển khai trên những hệ thống khác ngoài SQL Server như Oracle, MySQL...

MÔ TẢ BẢNG

SELECT * FROM PRICE

Bảng Price gồm có 3 cột, Lower Value là giá trị dưới, Upper Value là giá trị trên còn Price là giá trong khoảng Lower Value và Upper Value. Script tạo bảng và Insert dữ liệu mẫu như sau:

```
CREATE TABLE PRICE
LOWER VALUE INT,
UPPER_VALUE INT,
PRICE INT
G0
INSERT INTO PRICE (LOWER VALUE, UPPER VALUE, PRICE)
SELECT 0, 100, 5
UNION ALL
SELECT 100, 500, 10
UNION ALL
SELECT 500, 1000, 20
UNION ALL
SELECT 1000, NULL, 50
```

LOWER_VALUE	UPPER_VALUE	PRICE
0	100	5
100	500	10
500	1000	20
1000	NULL	50

Bảng thứ hai là bảng Customer. Bảng này gồm 3 trường Customer ID, Customer Name và Value. Script tạo bảng và dữ liêu mẫu như sau:

```
CREATE TABLE CUSTOMER
CUSTOMER ID INT,
CUSTOMER NAME VARCHAR(50),
VALUE INT
G0
INSERT INTO CUSTOMER (CUSTOMER ID, CUSTOMER NAME, VALUE)
SELECT 1, 'Mr Smith', 80
```

```
UNION ALL
SELECT 2, 'Mr Conrad', 240
UNION ALL
SELECT 3, 'Mrs Oreski', 800
UNION ALL
SELECT 4, 'Red Devilic', 1700
GO
SELECT *
FROM CUSTOMER
```

CUSTOMER_ID	CUSTOMER_NAME	VALUE
1	Mr Smith	80
2	Mr Conrad	240
3	Mrs Oreski	800
4	Red Devilic	1700

YÊU CÂU BÀI TOÁN

Cần viết truy vấn, đưa ra những dữ liệu sau:

- Customer ID
- Customer Name
- Price tương ứng với Value của khách hàng

Ví dụ với khách hàng có Customer ID = 2, Customer Name là Mr Conrad, Value = 240 sẽ được tính theo công thức sau:

100 đầu tiên nằm trong dải 0 – 100 nên có Price 100 x 5 = 500 140 sau đó nằm trong dải 101 – 500 nên có giá 140 x 10 = 1400

Vậy Price của khách hàng này là 500 + 1400 = 1900. Tương tự cho các khách hàng tiếp theo.

GIẢI QUYẾT BÀI TOÁN BẰNG JOIN BẮT CÂN BẰNG

Ý tưởng của giải pháp này được nhắc đến trong cuon sach SQL Hacks (Andrew Cumming, Gordon Russell – Publisher O'Reilly – Nov 2006). Đầu tiên ta nhận thấy trong bằng Price thì Value của khách hàng sẽ "bao" toàn bộ các bản ghi có Lower_Value thấp hơn. Như ví dụ ở trên, khách hàng có Value là 240 sẽ liên quan đến bản ghi thứ nhất và bản ghi thứ hai trong bảng Price (240 > 0 và 240 > 100).

SELECT *
FROM Price p JOIN Customer c
ON c.VALUE > p.lower_value;

LOWER_VALUE	UPPER_VALUE	PRICE	CUSTOMER_ID	CUSTOMER_NAME	VALUE
0	100	5	1	Mr Smith	80
0	100	5	2	Mr Conrad	240
100	500	10	2	Mr Conrad	240
0	100	5	3	Mrs Oreski	800
100	500	10	3	Mrs Oreski	800
500	1000	20	3	Mrs Oreski	800
0	100	5	4	Red Devilic	1700
100	500	10	4	Red Devilic	1700
500	1000	20	4	Red Devilic	1700
1000	NULL	50	4	Red Devilic	1700

Trong số những bản ghi liên quan, nếu giá trị value của khách hàng > Upper_Value, có nghĩa ta sẽ lấy trọn dải đó (Upper_Value – Lower_Value) đó nhân với Price. Ngược lại, sẽ lấy Value trừ đi Lower_Value rồi nhân với Price. Ví du trong trường hợp value của khách hàng là 240. Ta đã lấy tron 100 trong khoảng 0-100 nhân với Price là 5. Còn 140 (240 – 100) nhân với Price ứng trong khoảng 101-500 là 10.

Đến đây thì bài toán đã đơn giản hơn rất nhiều, ta lấy Extract_Value nhân với Price, sau đó SUM() toàn bộ với một phép Group By theo Customer_ID. Câu lênh cuối cùng sẽ như sau:

```
SELECT Customer_ID, CUSTOMER_NAME, SUM(Extract_Value * Price) RESULT FROM (

(SELECT c.Customer_ID, c.Customer_Name,c.VALUE,p.Lower_Value,

CASE WHEN c.VALUE > p.Upper_Value THEN p.Upper_Value - p.Lower_Value

ELSE c.VALUE - p.Lower_value END Extract_Value, p.Price
```

```
FROM Customer c JOIN Price p ON c.VALUE > p.Lower_Value
GROUP BY Customer_ID, CUSTOMER_NAME
ORDER BY Customer_ID
```

Customer_ID	CUSTOMER_NAME	Result
1	Mr Smith	400
2	Mr Conrad	1900
3	Mrs Oreski	10500
4	Red Devilic	49500

Tags: JOIN

14 Comments

Posted on 29/9/2014 | Categories: SQL Server Programming

Các bài viết tương tự

- Loại Bỏ Bản Ghi Trùng Trong Bảng
 Các Loại JOIN Trong SQL Server
- Lấy Một Số Bản Ghi Ngẫu Nhiên

Comments



Hay quá. cảm ơn Vũ Minh Tâm.

• Quan tran (06/10/2014 11:05 pm)

Quá hay! Thank you

Mình sẽ theo dõi thường xuyên.

• BacVT (09/10/2014 12:08 am)

sau một lúc ngồi ngắm code thì cuối cùng cũng đã ngấm . Ôi cảm ơn bác Tâm

• Dăng Khoa (09/10/2014 11:14 pm)

Rất hay, ngẫm code và chạy thử trên SQL 1 hồi đã hiểu ra, rất hữu ích khi ứng dụng cho thực tế! 😊 Cảm ơn bác Tâm! 😊

• vanhungbkcbg1 (15/10/2014 5:51 am)

bài viết này quá hay,ý tưởng rất thông minh,cảm ơn anh,hi vọng anh sẽ có nhiều bài viết hay hơn nữa cảm ơn anh

Reply

• *paguoihanoi* (21/11/2014 1:04 am)

Thấy chủ xi bảo giải pháp này là ý tưởng trong cuốn sách SQL Hacks của bọn để quốc Mỹ mà tự ái dân tộc nổi cuồn cuộn.

Đây là giải pháp trong cuốn sách Những luật vàng trong SQL (tác giả nguoihannoi) há há.

Người Việt chúng ta cần cù, chăm chỉ, thông minh, giỏi toán, có năng khiếu tin học, và nhiều thứ linh tinh khác (cụ Khải bảo thế – cấm cãi 🚇) nên không cần join "Cân Bằng" hay join "Bất Cân Bằng", chẳng where on gì ráo, chỉ răng với dế mà vẫn sinh con đẻ cái ngọn lành.

```
SELECT customer_id, customer_name, SUM(ket_qua) ket_qua FROM customer
outer apply (

SELECT
iif(VALUE between lower_value and isnull(upper_value,100000000), (value-lower_value)*price,0)
+
iif(VALUE > upper_value, (upper_value-lower_value)*price,0)
ket_qua FROM price
)oa
GROUP BY customer_id, customer_name
ORDER BY customer_id
```

Reply

Red Devilic (27/11/2014 12:06 pm)

Nice post,

Hồi 2k5 mình có đọc về cái Outer Apply này nhưng không để ý, cũng không xem lại cách dùng. Thanks bác đã cho mình xem lại 😁

Nhưng hình như format câu lệnh bị vỡ, bác gửi lại câu đầy đủ dc ko?

Reply



Vũ Huy Tâm (08/12/2014 2:50 pm)

Cái này đúng là độc chiều. Like mạnh

<u>Reply</u>

Nguyễn Hoàng Dũng (10/02/2015 10:42 pm)

Bác chủ xị đã nói một điểm mạnh của join bất cân bằng là "Có thể cài đặt bằng SQL, thuận tiện khi triển khai trên những hệ thống khác ngoài SQL Server như Oracle, MySQL...". Vì vây làm theo kiểu VN mình thì chi xài được trên sal server thôi. Nước ngoài họ đi trước mình rất lâu rồi, giờ họ đã lên tầm tổng quát, không còn cực bộ với một hệ thống nhất định nữa rồi.

Reply

Red Devilic (01/03/2015 10:57 pm)

Thực ra cái IFF kia cũng có thể viết lại bằng ANSI SQL mà 😊

Reply

• Learning (28/11/2014 3:34 am)

iif(VALUE > upper_value, (upper_value-lower_value)*price,0) => iif(VALUE > upper_value, (upper_value-lower_value)*price,0)

Reply



Red Devilic (01/12/2014 3:55 am)

Thanks bác.

Chưa cập nhật cái IIF mới này 😊

```
Reply

Thành Thiện (04/11/2015 5:59 am)
```

Câu 1:

```
SELECT hs.ma AS ma, phs.sobiennhan AS sobiennhan, hs.ten AS tenhs, lhs.ten loaihs, phs.veviec AS veviec, b.tennguoinop AS nguoinop, nv.ten AS nguoitiepnhan, h.ten AS nguoixily
,u.tinhtrang AS tinhtrang, (CASE ths.ketqua WHEN 1 THEN 'Duyêt' WHEN 2 THEN 'Không duyêt' ELSE 'Dang xử lý' END) AS ketqua FROM hoso hs join phieuhoso phs ON hs.ma=phs.mahoso FULL join trahoso ths ON
FULL join xulyhoso xhs ON phs.maql=xhs.maql FULL join (SELECT ten, ma FROM nhanvien ) h ON h.ma = xhs.manvxl
join loaihoso lhs ON hs.maloai=lhs.ma join nhanvien nv ON nv.ma = phs.manguoixl
join (SELECT phs.mahoso AS mahoso,nn.hoten AS tennguoinop FROM phieuhoso phs join nguoinop nn ON phs.manguoinop = nn.cmnd) b ON hs.ma = b.mahoso
join (SELECT g.maql,g.xuhs,g.trahs,g.phieuhs, (CASE (g.xuhs+g.trahs+g.phieuhs)
WHEN 1 THEN 'Chờ xử lý' WHEN 2 THEN 'Đạng xử lý' WHEN 3 THEN 'Đã xử lý' ELSE 'Không xử lý' END) AS Tinhtrang FROM (SELECT phs.maql AS maql,
(CASE WHEN xhs.xuhs IS null THEN 0 ELSE 1 END) xuhs, (CASE WHEN ths.trahs IS null THEN 0 ELSE 1 END) AS trahs, (CASE WHEN phs.phieuhs
IS null THEN 0 ELSE 1 END)AS phieuhs FROM ( SELECT COUNT(*) AS xuhs, maql FROM xulyhoso GROUP BY maql )xhs
FULL join (SELECT COUNT(*) AS phieuhs, maq1 FROM phieuhoso GROUP BY maq1) phs ON xhs.maq1=phs.maq1
FULL join (SELECT COUNT(*) AS trahs,maql FROM trahoso GROUP BY maql) ths ON phs.maql=ths.maql) g) u ON u.maql=phs.maql
WHERE to date(phs.ngay, 'dd-mm-yy') =to date('04-Nov-15', 'dd-mm-yy') or to date(ths.ngay, 'dd-mm-yy') =to date('04-Nov-15', 'dd-mm-yy')
or to_date(xhs.ngay,'dd-mm-yy') =to_date('04-Nov-15','dd-mm-yy')
Câu 2:
SELECT to_char(k.ngay,'dd/mm/yyyy') AS ngay ,( CASE WHEN k.phieuhs IS null THEN 0 ELSE k.phieuhs END)AS phieuhs,( CASE WHEN k.xuhs IS null THEN 0 ELSE k.xuhs END)AS xuhs,
(CASE WHEN k.trahs IS null THEN 0 ELSE k.trahs END) AS trahs ,(CASE WHEN k.tientu IS null THEN 0 ELSE k.tientu END)AS tientu,(CASE WHEN k.thanhtoan IS null THEN 0 ELSE k.thanhtoan END)AS phihs,
((CASE WHEN k.thanhtoan IS null THEN 0 ELSE k.thanhtoan END) - (CASE WHEN k.tientu IS null THEN 0 ELSE k.tientu END)) AS chiphi FROM
(SELECT (CASE WHEN h.ngay IS null THEN ttct.ngay ELSE h.ngay END) AS ngay, h.phieuhs AS phieuhs, h.xuhs AS xuhs, h.trahs AS trahs, h.tientu AS tientu, ttct.thanhtoan AS thanhtoan FROM
( SELECT SUM(sotien) AS thanhtoan,to_date( ngay,'dd-mm-yy') AS ngay FROM thanhtoanct GROUP BY to_date( ngay,'dd-mm-yy')) ttct FULL join
(SELECT (CASE WHEN n.ngay IS null THEN tu.ngay ELSE n.ngay END) AS ngay, tu.tientu AS tientu, n.trahs AS trahs, n.phieuhs AS phieuhs, n.xuhs AS xuhs FROM
(SELECT SUM(sotien) AS tientu,to_date( ngay, 'dd-mm-yy') AS ngay FROM tamung GROUP BY to_date( ngay, 'dd-mm-yy')) tu FULL join
(SELECT (CASE WHEN b.ngay IS null THEN ths.ngay ELSE b.ngay END) AS ngay, ths.trahs AS trahs, b.phieuhs AS phieuhs, b.xuhs AS xuhs FROM
(SELECT COUNT(ngay)AS trahs,to_date( ngay,'dd-mm-yy') AS ngay FROM trahoso GROUP BY to_date( ngay,'dd-mm-yy')) ths FULL join
(SELECT (CASE WHEN phs.ngay IS null THEN xhs.ngay ELSE phs.ngay END) AS ngay, phs.phieuhs AS phieuhs, xhs.xuhs AS xuhs
FROM (SELECT COUNT(ngay) AS phieuhs, to date( ngay, 'dd-mm-yy') AS ngay FROM phieuhoso GROUP BY to date( ngay, 'dd-mm-yy')) phs FULL join
(SELECT COUNT(ngay)AS xuhs, to_date(ngay,'dd-mm-yy')AS ngay FROM xulyhoso GROUP BY to_date( ngay,'dd-mm-yy'))
"xhs on (xhs.ngay=phs.ngay)) b on (b.ngay=ths.ngay)) n on (n.ngay=tu.ngay)) h on h.ngay=ttct.ngay) k where to_date(k.ngay,'dd-mm-yy') between "to_date('" + dt1 + "','dd-mm-yy') and to_date('" + dt2 + "','dd-mm-yy') ORDER BY ngay"
```

Ai có thể giúp mình rút gọn 2 câu SQL này lại với được không, mình mới tập viết nên không rành lắm, và không biết làm sao để lấy dữ liệu cho đúng và tối ưu hóa được câu lệnh cả. Cảm ơn trước

Reply

Thành Thiên (04/11/2015 6:02 am)

Bổ sung dữ lieu mẫu cho mọi người dễ hình dung nè...

Câu 1

151031081105788 ghj124 ở chi nhánh phong khám Hồ sơ nghiệp vụ y mở phòng khám trường giang Tô Thành Thiện Tô Thành Thiện ?ã x? lý Không duy?t 151031081105788 ghj124 ở chi nhánh phong khám Hồ sơ nghiệp vụ y mở phòng khám trường giang Tô Thành Thiện Hà Trọng Nghĩa ?ã x? lý Không duy?t 151031080927029 ghj123 hồ sơ xin cấp phép bán mở phòng khám Hồ sơ nghiệp vụ y mở phòng khám Lê Văn Khải Tô Thành Thiện Tô Thành Thiện ?ã x? lý Duy?t

<u>Reply</u>

Leave a Reply

	Name (required)
	Email (will not be published) (required)
	Website

Hướng dẫn: Để nhập mã T-SQL bạn dùng thẻ và .

Ví dụ: SELECT * FROM MyTable

Like 592 people like this. Be the first of your friends.

Đề nghị ghi rõ nguồn gốc khi trích dẫn hoặc đăng lại nội dung từ blog này

Giới Thiệu

SQL Việt Q&A

Hội thảo SQL Server – HCMC 2013 _{□ 68}

Seminar Presentation: Index trong SQL Server - Download

Search

PHẢN HỘI MỚI

- toru on Vì sao nên tránh viết SQL code trong ứng dụng
- toru on <u>DELETE và TRUNCATE</u>
- Lê Lam on Các Chế Đô Phục Hồi Của Database
- in áo theo yêu cầu on Cấu Hình SQL Server Để Gửi Mail
- john thornhill on Thêm Dữ Liệu Mới Vào Bảng Dùng LEFT JOIN
- Tho Nguyen on Tao SQL Job Dùng Để Backup Database Hàng Ngày
- Phat Dat on Tao SQL Job Dùng Để Backup Database Hàng Ngày
- Amazon Solar led light on Tån man về NULL
- Amazon Solar lights for garden on Tan man về NULL
- thiết kế phòng ngủ nhỏ 6m2 on <u>Cấu Hình SQL Server Để Gửi Mail</u>
- CườngMH on <u>Tạo SQL Job Dùng Để Backup Database Hàng Ngày</u>
- uniwersytet on Viết Resume Xin Việc Như Thế Nào
- Darren Slonski on Các Kiểu Backup Trong SQL Server
- Benton Pontillo on <u>Tao SQL Job Dùng Để Backup Database Hàng Ngày</u>
- HOAT on <u>Di chuyển Database File</u>
- NGUYỄN NHƯ HOAT on Sql Đông
- Nguyen Manh Cường on Bảng Tạm và Biến Kiểu Bảng
- Thắng on Làm Sao Giảm Bót Dung Lượng File LOG Của SQL Server
- Thắng on Làm Sao Giảm Bót Dung Lượng File LOG Của SQL Server
- Lý Mạc Sầu on Các Loại JOIN Trong SQL Server

BÀI MỚI

- Ghép Nối Nhiều Bản Ghi Vào Một Dòng

 45
- Log File Để Làm Gì ■7
- <u>Úng Dung của Join Bất Cân Bằng</u> ■15
- Kinh Nghiệm Phỏng Vấn □0
- Viết Resume Xin Việc Như Thế Nào ■37
- Cấu Hình SOL Server Để Gửi Mail

 32
- User Lac Me ■8
- Thiết Lập Cho xp cmdshell

 2
- [LINH TINH] Từ Vựng Lớp 3 □22
- LEFT JOIN Với Mênh Đề WHERE 15
- <u>Di chuyển Database File</u> <u>□2</u>
- Khai Trương Trang SQL Việt Q&A

 □ 26
- Tâm Sư Cuối Năm

 10

 10
- Partition Bảng Đã Có Sẵn ■16
- Bảng Tam và Biến Kiểu Bảng ➡39
- Tham Số Fill Factor ■4
- Hôi thảo SQL Server HCMC 2013 ■68
- Tản mạn về NULL

 53

Xem toàn bộ các bài post...

CHỦ ĐỀ

- <u>Bảo mật</u> (4)
- Bên trong SQL Server (3)
- Community (2)
- <u>Công đồng</u> (3)
- <u>Database Administration</u> (27)
- Download (2)
- Eventual Consistency (2)
- <u>Function</u> (2)
- IDENTITY (2)
- <u>Index</u> (10)
- <u>Linh tinh</u> (11)
- Performance tuning (11)
- <u>SQL Server Programming</u> (34)
- Sql đông (4)
- SSIS (1)
- Stored Procedure (3)
- Table partitioning (6)
- Thiết kế database (12)
- Tip & Trick (8)
- <u>Uncategorized</u> (1)
- <u>View</u> (1)

NHÃN

backup Bookmark Lookup CASCADE DELETE CASE CHECKIDENT clustered index Common Table Expression Convert Covering Index CROSS JOIN Data file DATE FIRE DATE TIME DBCC De-duplicate Default file location DELETE differential backup Dynamic SQL Eventual Consistency, EXECO Execution plan exists filegroup File location filtered index FOREIGN KEY full backup FULL OUTER JOIN GROUP BY HAVING IDENTITY INCREMENT IDENTITY SEED IDENTITY INSERT IF EXISTS Include Index partitioning Index Seek INFORMATION SCHEMA INNER JOIN INSERT ISNULL JOIN Key Lookup LEFT LEFT JOIN LIKE % Log file login MySpace NewIDIO, Nhát quán cuói công Non-clustered index NoSOL NOT IN NULL OBJECT ID OUTER JOIN OUTPUT OUTPUT INTO Parameter PARTITION Performance phân doan PRIMARY KEY RANDOM recovery model RESEED RESET IDENTITY restore restore log ROUTINES ROW_NUMBER Sarg Sargable SCOPE IDENTITY Search Condition SELECT sp. executesql sp. helpfile Sql động Stored Procedure SUBSTRING SYS.IDENTITY COLUMNS table-valued function table partitioning Tempdb Tham số Thủ tug transaction log backup Truncate Tim kiếm động UNION ALL unique constraint unique index UPDATE user

Blog Việt

- <u>ddth.com (SQL Server forum)</u>
- cione Đào tạo trực tuyến
- Buu Nguyen blog
- Trang CSDL trên facebook

Blog Anh

- <u>SQLBlog.com</u>
- SQL Server Central
- SQL Authority
- Database management and analytics
- <u>SQL Server Customer Advisory Team</u>
- SQL University
- SQL Server Pedia
- Brent Ozar's blog



© SQL Việt blog 2019 · Powered by Wordpress · Design by Thomas Klaiber