**[วิธีใช้เงื่อนไข IF , Sub-Query ฯลฯใน MySQL เพื่อให้ได้ผลลัพ](http://webboard.moph.go.th/board/viewtopic.php?f=28&t=2847&sid=80cfaa5834cb75956eb1b38ed71ad0b8" \l "p9227)**

[โพสต์](http://webboard.moph.go.th/board/viewtopic.php?p=9227&sid=03251200d657b3731d9014c3540b5b2f#p9227)โดย **[neowara](http://webboard.moph.go.th/board/memberlist.php?mode=viewprofile&u=80&sid=03251200d657b3731d9014c3540b5b2f)** » 21 ก.ค. 2011 15:16

หน้าแรกแท็กแฟ้มบทความโอเพนซอร์สประวัติโดมินิคติดต่อผมวิธีใช้เงื่อนไข IF , Sub-Query ฯลฯใน MySQL เพื่อให้ได้ผลลัพธ์ที่ต้องการ   
Blog > วิธีแก้ปัญหาเขียนโปรแกรม > วิธีใช้เงื่อนไข IF , Sub-Query ฯลฯใน MySQL เพื่อให้ได้ผลลัพธ์ที่ต้องการ  
Published by DominixZ on April 24, 2011 01:58 am under วิธีแก้ปัญหาเขียนโปรแกรม, เคล็ดลับ  
  
เนื่องจากผมจะต้องทำ Report พร้อมมีสมการบางอย่าง เพื่อเป็นการคำนวณ ซึ่งผมใช้ PHP กับ MySQL เป็นหลัก โดยปัญหาที่มักจะเกิดบ่อยๆก็คือ “ดึงข้อมูลจาก Database มาคำนวณที่ PHP แล้วก็เอามาเรียงใส่ Array ใหม่เป็นข้อมูลชุดใหม่” และยังมีเหตุการอื่นๆอีกเช่น ไม่ยอม Join Table แล้ว GET ข้อมูลจาก Table หนึ่งไปใส่อีก Table หนึ่งทำให้ Performance ช้าอีก  
  
วันนี้ผมเลยขอแนะนำตัวที่เราได้ใช้บ่อยๆกันใน MySQL กันนะครับผมก็พึงลองสดๆร้อนๆเลย  
  
CASE  
CASE value WHEN [compare\_value] THEN result [WHEN [compare\_value] THEN result ...] [ELSE result] END  
  
จริงๆผมสับสนระหว่าง CASE กับ IF แต่พอใช้แล้วก็เข้าใจขึ้น โดยผมมองว่า CASE คือการ FILTER ข้อมูล แต่ IF เป็นเงื่อนไข แต่ปกติเราเขียนโปรแกรมมักใช้ IF เป็นการ FILTER ข้อมูล แต่ใน Database มักแยกกันชัดเจน ยกตัวอย่าง เช่น if(number == 1) ใน php ที่จริงคือ CASE WHEN number = 1 ทำนองนั้นครับ ตัวอย่างที่ชัดเจนคือ  
  
SELECT page\_id , SUM(CASE WHEN score = 1 THEN 1 ELSE 0 END) AS good , SUM(CASE WHEN score = 0 THEN 1 ELSE 0 END) AS neutral , SUM(CASE WHEN score = -1 THEN 1 ELSE 0 END) AS bad  
FROM sentiments  
  
โดยใน Table มี Field หนึ่งที่เก็บ score ซึ่งส่วนใหญ่แบ่งเป็น 1 , 0 , -1 ซึ่งผมต้องการ 3 เลขนี้ขึ้นมาใน Row เดียวผมเลยต้องใช้วิธีนี้ เพราะมันจะเกี่ยวโยงกับ Subquery เพื่อนำไปใช้ต่อในการคำนวณ ดังนั้นมันเลยเป็นสิ่งจำเป็นที่ต้องได้ค่าเหล่านี้มาอยู่ใน Row เดียว  
  
IF  
IF(expr1,expr2,expr3)  
เป็นการใส่เงื่อนไขอย่างที่อธิบายก่อนหน้านี้ โดยวิธีใช้คือ IF(SUM(score) < 80,IF(SUM(score) < 50,’F',’C'),’A') มันก็คือ IF ELSE ใน PHP นั้นเอง แต่สิ่งเดียวที่มันทำไม่ได้คือ IF(score == 1,’YES’,'NO’) ทำนองนี้ เน้นอีกครั้งนะครับ ทำไม่ได้ ดังนั้นเราเลยต้องใช้ CASE มาผสมกัน  
  
อื่นๆ  
อาจจะมีบางคนไม่รู้ว่า จริงๆแล้ว SQL สามารถ + – \* / ( ) ได้ยกตัวอย่าง เช่น SELECT page\_id , score/100 FROM sentiments เป็นต้น  
  
รู้พื้นฐานแล้วแต่ก็คงยังไม่พอต้องผสมให้เป็น  
โดยผมจะลองยกตัวอย่าง Query ซ้อน Query มาให้ดูสักอันจากของจริงนะครับ  
  
SELECT page\_id,create\_date, IF(ABS((((good/devideval) + (neutral/devideval)) – ((bad/devideval) + (neutral/devideval)))) > 7,IF((((good/devideval) + (neutral/devideval)) – ((bad/devideval) + (neutral/devideval))) >0,1,-1),0) AS score  
FROM  
(  
SELECT page\_id , SUM(CASE WHEN score = 1 THEN 1 ELSE 0 END) AS good ,SUM(CASE WHEN score = 0 THEN 1 ELSE 0 END) AS neutral , SUM(CASE WHEN score = -1 THEN 1 ELSE 0 END) AS bad , COUNT(score) as total , (COUNT(score) / 100) as devideval , DATE(created\_at) AS create\_date  
FROM `sentiments`  
WHERE user\_id = 1 GROUP BY page\_id,DATE(created\_at)  
) AS sentiment\_result  
  
โดยแม้จะเป็น Query ที่ดูยากสักหน่อย แต่ผมจะอธิบาย SELECT ล่างก่อนนะครับโดยจะอ่านเป็นประมาณนี้ครับ  
  
“ให้ทำการเลือก page\_id , ผลรวมของ score โดย score มีค่าเป็น 1 ให้มีค่าเป็น 1 อย่างอื่นเป็น 0 , ผลรวมของ score โดย score มีค่าเป็น 0 ให้มีค่าเป็น 1 อย่างอื่นเป็น 0 เป็น good , ผลรวมของ score โดย score มีค่าเป็น 0 ให้มีค่าเป็น 1 อย่างอื่นเป็น 0 เป็น neutral , ผลรวมของ score โดย score มีค่าเป็น -1 ให้มีค่าเป็น 1 อย่างอื่นเป็น 0″  
  
อธิบายเพิ่มเติมตรงส่วนนี้คือ การที่ต้องทำให้ต้องมีค่าเป็น 1 เป็น 0 นั้นเพราะความต้องการของผมคือ “จำนวนนับของ score 1 , 0 และ -1″ เลยต้องทำแบบนั้น จะ COUNT ก็ได้เช่นกัน แต่ผมขี้เกียจแก้ของผมที่ทำได้แล้วเท่านั้นเอง  
  
ส่วนที่เหลือก็ Where ตามปกติ แต่ที่นี้ผมต้องอธิบายเพิ่มเพราะผมนำ SELECT อันล่างไปเป็นข้อมูลของ SELECT ข้างบนดังนั้นตอนนี้ผมเลยมี page\_id , good , neutral , bas , total , devideval , create\_date ที่สามารถเรียกใช้งานได้ ใน SELECt ด้านบนผมเลยมาทำสมการเลข สังเกตุว่าผมใส่ ( ) ค่อนข้างมาก เพื่อความมั่นใจว่า มันจะหารก่อนแล้วค่อยนำมา + – กันเสร็จแล้วผมก็ยังใช้ IF ELSE ฉบับ MySQL อีกด้วยคือ IF(express,IF(express,result1,result2),result3) ทำนองนั้นครับ  
  
ประโยชน์ที่ได้รับสำหรับการใช้การคำนวณอยู่ใน Database  
เร็วไม่เปลื้อง CPU , Memory ในการทำ Array ชุดใหญ่ใน PHP  
สำหรับผม Maintenance ทำได้สะดวกกว่าใช้ PHP SQL สลับไปสลับมา  
โดยหลักๆคือ 2 ข้อแต่ประโยชน์จะทวีคูณ หากนำไปใช้กับ PHP อย่างเหมาะสม โดยคำเตือนขอเดียวก็คือ “ให้รู้ Limit ของ SQL และไม่จำเป็นต้องตั้งเงื่อนไขว่า จะได้ข้อมูลเหล่านี้ด้วย SQL ล้วนๆ มีบางกรณีของผมซึ่งใช้ Query เดิมได้แต่ต้องนำ good\*1 , neutral\*0 , bad\*-1 แล้วนำมา + กันทั้งหมด ซึ่งตามจริงแล้วการจะได้เช่นนั้นผมจะต้อง SELECT อีกครั้งหนึ่งซึ่งมันสุดโต่งเกินไป ผมเลยหยุดที่ก่อน Step นั้นแล้วเวลานำไปใส่ใน Report ผมก็ใช้ PHP + – เอาแต่ทีนี้มันง่ายแล้วไงครับ เพราะใน Row ก็มี good , neutral , bad มาให้พร้อมแล้วแค่ใส่สมการ เอาไปแสดงผลเป็นอันเสร็จ วันนี้ผมลาไปก่อนหวังว่ามีประโยชน์ไม่น้อยสวัสดีครับ