SELECT FROM AUSTIN, TEXAS

Joe Celko over SQL en andere database-zaken

Log-tabel

Dit is een leuk probleempje dat eigenlijk meer een rapport is dan een query. Gegeven is een log-tabel die laat zien welke wijzigingen door de tijd heen op een machine zijn gemaakt op een aantal controls (a, b, c, d). Niet elke control wordt elke dag ge-reset. De tabel ziet er zo uit:

```
CREATE TABLE Machine_Log
(setting_date DATE NOT NULL,
 control CHAR(1) NOT NULL
       CHECK (control IN ('a', 'b', 'c', 'd')),
 setting INTEGER NOT NULL,
 PRIMARY KEY (setting_date, control));
```

Eerste opgave is de huidige status van de machine te laten zien door de settings van de vier controls te tonen.

```
CREATE VIEW Current_Machine_State
(control, setting)
 AS SELECT M1.control, M1.setting
     FROM Machine_Log AS M1
    WHERE setting_date
         = (SELECT MAX(M2.setting_date)
             FROM Machine_Log AS M2
            WHERE M1.control = M2.control);
```

Eigenlijk willen we een rapport hebben dat laat zien of we de controls op een bepaalde datum hebben bijgesteld naar boven, naar beneden of ongewijzigd hebben gelaten.

```
INSERT INTO Machine_Log VALUES ('2001-11-03', 'a', 10);
INSERT INTO Machine_Log VALUES ('2001-11-03', 'b', 10);
INSERT INTO Machine_Log VALUES ('2001-11-03', 'c', 10);
INSERT INTO Machine_Log VALUES ('2001-11-03', 'd', 10);
INSERT INTO Machine_Log VALUES ('2001-11-04', 'a', 10);
INSERT INTO Machine_Log VALUES ('2001-11-04', 'b', 9);
INSERT INTO Machine_Log VALUES ('2001-11-05', 'c', 10);
INSERT INTO Machine_Log VALUES ('2001-11-06', 'a', 10);
INSERT INTO Machine_Log VALUES ('2001-11-06', 'c', 15);
INSERT INTO Machine_Log VALUES ('2001-11-06', 'd', 11);
```

Wat denkt u van dit probleem? We hebben de andere kolom al, die ons de trend laat zien. In pseudo-code:

```
SELECT M1.control, M1.setting, <trend column>
  FROM Machine_Log AS M1
 WHERE setting_date
       = (SELECT MAX(M2.setting_date)
           FROM Machine_Log AS M2
          WHERE M1.control = M2.control);
```

We moeten een beslissing nemen over <trend>: moet dit een getal worden of een character string? Als we kiezen voor het laatste, moeten we een CASE-clausule gebruiken om de juiste string te krijgen. Maar als uw SQL een SIGN()-functie heeft, kunt u SIGN (huidige setting - vorige setting) gebruiken, waardoor u een -1, 0, +1 of NULL als resultaat krijgt. Vaak wordt vergeten dat ook NULL een mogelijk resultaat is! We verbeteren de pseudo-code een beetje:

```
SELECT M1.control, M1.setting,
       SIGN (M1.setting - <prior setting>) AS trend
  FROM Machine_Log AS M1
 WHERE setting_date
       = (SELECT MAX(M2.setting_date)
           FROM Machine_Log AS M2
          WHERE M1.control = M2.control);
```

Het is lastig de vorige stand te vinden, maar we kunnen dat zo doen:

```
SELECT M1.control, M1.setting,
       SIGN (M1.setting
        - (SELECT DISTINCT M3.setting
            FROM Machine_Log AS M3
           WHERE M1.control = M3.control
             AND M3.setting_date < M1.setting_date))
       AS trend
```

```
 FROM Machine_Log AS M1
WHERE setting_date
     = (SELECT MAX(M2.setting_date)
          FROM Machine_Log AS M2
         WHERE M1.control = M2.control)
```

Wat gebeurt er als slechts één stand op een control mogelijk is? Een goede vuistregel is dat u bij een ontbrekende waarde een COALESCE()-functie gebruikt.

```
SELECT M1.control, M1.setting,
       SIGN (M1.setting
        - COALESCE (
            (SELECT DISTINCT M3.setting
               FROM Machine_Log AS M3
              WHERE M1.control = M3.control),
           M1.setting)) AS trend
 FROM Machine_Log AS M1
WHERE setting_date
     = (SELECT MAX(M2.setting_date)
          FROM Machine_Log AS M2
         WHERE M1.control = M2.control)
```

Dit stukje code ziet er vreselijk uit. We hebben geneste query's en her en der zitten functies.

We proberen het nog eens. Een andere vuistregel is dat als een rij gemaakt moet worden van twee rijen uit dezelfde tabel, dit een self-join betekent. We gaan terug naar de pseudo-code en we benutten de kennis die we met de eerste poging hebben opgedaan.

```
SELECT M1.control, M1.setting,
       SIGN (M1.setting - COALESCE(M2.setting,
          M1.setting))
       AS trend
  FROM Machine_Log AS M1,
       Machine_Log AS M2  – the prior copy
 WHERE M2.setting_date < M1.setting_date
   AND M1.control = M2.control
   AND M1.setting_date
       = (SELECT MAX(M3.setting_date)
            FROM Machine_Log AS M3
           WHERE M1.control = M3.control)
   AND <other stuff>;
```

De eerste twee eigenschappen definiëren een gezamenlijke voorwaarde, waaraan voldaan moet zijn tussen de huidige en vorige standen in.

Nog een regel: is er sprake van ontbrekende gegevens, gebruik dan een OUTER JOIN. We repareren de pseudo-code opnieuw. We kunnen ook een ander trucje gebruiken voor de op één na hoogste waarde:

```
SELECT M1.control, M1.setting,
    SIGN (M1.setting - COALESCE(M2.setting, M1.setting))
       AS trend
  FROM Machine_Log AS M1
       LEFT OUTER JOIN
       Machine_Log AS M2  – the prior copy
       ON M2.setting_date < M1.setting_date
          AND M1.control = M2.control
          AND M1.setting_date
              = (SELECT MAX(M3.setting_date)
                   FROM Machine_Log AS M3
                  WHERE M1.control = M3.control)
          AND M2.setting_date
              = (SELECT MAX(M3.setting_date)
                   FROM Machine_Log AS M3
                  WHERE M1.control = M3.control
                    AND M3.setting_date < M1.setting_date);
```

Bij het gebruik van OUTER JOIN's en WHERE-clausules moeten we uiterst zorgvuldig de plaats van de eigenschappen bepalen.

```
SELECT M1.control, M1.setting_date, M1.setting,
       SIGN (M1.setting - COALESCE(M2.setting,
          M1.setting))
       AS trend
  FROM Machine_Log AS M1
       LEFT OUTER JOIN
       Machine_Log AS M2  – the prior copy
       ON M2.setting_date < M1.setting_date
          AND M1.control = M2.control
          AND M2.setting_date
              = (SELECT MAX(M3.setting_date)
                   FROM Machine_Log AS M3
                  WHERE M2.control = M3.control
                    AND M3.setting_date < M1.setting_date)
 WHERE M1.setting_date
       = (SELECT MAX(M3.setting_date)
            FROM Machine_Log AS M3
           WHERE M1.control = M3.control)
```

Nu nog de uitvoering controleren en kijken wat het beste werkt. ●

Joe Celko (www.celko.com) is onafhankelijk consultant en lid van het ANSI X3H2 Database Standards Committee. Hij is auteur van diverse boeken over SQL. Als SQL-specialist schijft hij behalve voor Database Magazine voor het blad *Intelligent Enterprise* (voorheen DBMS).