



SELECT WHERE nur ein Eintrag vorhanden

Select-Befehle sind in der Regel im SAP-Umfeld einigermaßen überschaubar. Meiner Meinung nach liegt es daran, dass in der Regel Programmierer am Werk sind, aber keine SQL-Spezialisten. Zudem bietet der Open-SQL-Standard von SAP auch nur einen eingeschränkten Funktionsumfang. Selbst wenn man etwas kompliziertere Selects durchführen möchte, ist das nicht unbedingt möglich.

Joins sind eine andere Geschichte. Hier toben sich Programmierer gerne einmal aus und joinen was das Zeug hält.

SELECT oder ABAP?

In beiden Fällen, der Select-Anweisung und der nachträglichen Verarbeitung, gibt es Situationen, wo beides irgendwie umständlich ist. Ein solcher Fall ist es zum Beispiel, wenn man nur die Daten selektieren möchte, von denen man erst nach einer Aggregatbildung weiß, ob man sie lesen möchte oder nicht. Also zum Beispiel: Gib mir alle Belege, zu denen nur ein Eintrag in der Positionstabelle vorhanden ist.

SUBQUERY

Der folgende Select ermittelt genau die Einträge, bei denen nur ein Eintrag in der VBAP vorhanden ist.

```
SELECT vbeln, matnr
  FROM vbap AS v
 WHERE vbeln IN @s_vbeln
        AND matnr IN @s_matnr
        AND 1 = ( SELECT COUNT( * ) FROM vbap
                  WHERE vbeln = v~vbeln )
```

```
GROUP BY vbeln, matnr
ORDER BY vbeln, matnr
INTO TABLE @DATA(list).
```

HAVING

durch Zufall bin ich bei der Nachstellung des Problems auf eine andere Lösung gestoßen, bei der die Anzahl der vorhandenen Sätze mittels HAVING eingeschränkt wird.

```
SELECT vbeln, sum( kwmeng ) as kwmeng, vrkme
FROM vbap AS v
WHERE vbeln IN @s_vbeln
AND matnr IN @s_matnr
GROUP BY vbeln, vrkme
HAVING COUNT(*) = 1
ORDER BY vbeln, vrkme
INTO TABLE @data(list).
```

Der Vorteil bei der HAVING-Variante ist, dass hier die Anzahl der Datensätze auch variabel abgefragt werden können. Das geht mit dem oben vorgestellten Subquery nicht.

Performancevergleich

Ich habe beide Varianten in zwei unterschiedlichen Systemen mit unterschiedlichen Tabellen ausprobiert und einen kleinen Performancevergleich gemacht. In der einen Version ist die Variante mit dem Subquery schneller, in der anderen Version ist HAVING schneller.

Im Gegensatz zu einer umständlichen Analyse per Programmcoding, welcher Satz nur einmal vorhanden ist, sind beide SELECT-Versionen meiner Meinung nach eleganter. Die Performance sollte man auf jeden Fall bei jeder Version im Blick haben.