Matrix Reports mit Apex

Thomas Hernando Gotthardt
syntegris information solutions GmbH
Neu-Isenburg

Schlüsselworte:
Matrix, Kreuztabellen, Reports, Apex, SQL, PL/SQL, Packages, JavaScript, HTML, Pivot-Klausel, Oracle 11g, Oracle Application Express

Einleitung

Mit Matrix Reports (oder auch Kreuztabellen genannt) lassen sich verschiedenste Daten kompakt und übersichtlich darstellen. Täglich benutzen wir sie bei öffentlichen Verkehrsmitteln in Form von Bus- und S-Bahnfahrplänen. In vielen Bereichen gibt es Anwendungsfälle für Kreuztabellen, z.B. bei der Darstellung von Verkaufszusammenhängen (siehe Tabelle 1). Allgemein stellen Kreuztabellen ein sehr hilfreiches Mittel dar, um dem Anwender schnell und effektiv Daten aufzubereiten.

Produkt \ Altersgruppe	**unter 45 Jahren**	**über 45 Jahren**
Produkt 1	450	475
Produkt 2	250	375

Tabelle 1 - Beispiel einer Kreuztabelle

Um in Oracle Application Express eine solche Darstellung von tabellarischen Daten zu erhalten gibt es verschiedene Ansätze. Zum einen kann mit reinem SQL-Select Befehlen eine matrixähnliche Darstellung erreicht werden. Ab Oracle 11g kann dafür die (statische) Pivot-Klausel genutzt werden um Zeilen „umzubrechen“. Zum anderen kann mithilfe von PL/SQL-Regions ein sehr hoher Individualisierungsgrad in Bezug auf die Darstellung von Kreuztabellen erzielt werden. Eine weitere Möglichkeit um Daten entsprechend aufzubereiten kann durch die Anpassung von Report Templates erreicht werden.

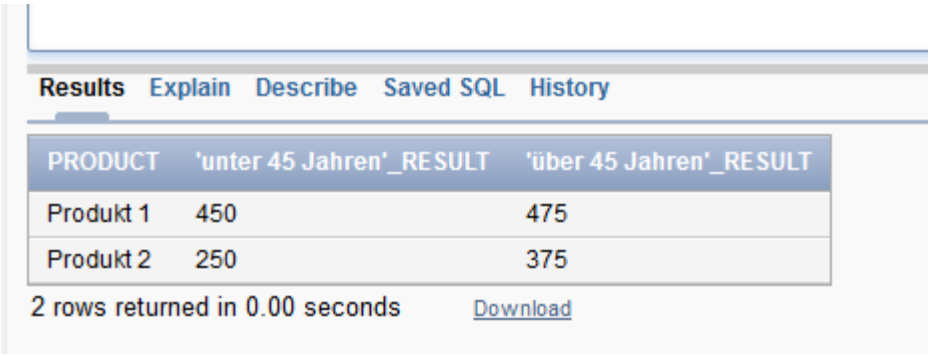
Pivot-Klausel

Die Pivot-Klausel ist standardmäßig seit Version 11g mit an Board, mit ihrer Hilfe kann die Datenbank tabellarische Daten als Kreuztabelle ausgeben. Das folgende Statement zeigt die Verwendung der Pivot-Klausel:

```
WITH tab (product, age, val) AS (
    SELECT 'Produkt 1', 'unter 45 Jahren', 450 FROM DUAL UNION ALL
    SELECT 'Produkt 1', 'über 45 Jahren',  475 FROM DUAL UNION ALL
    SELECT 'Produkt 2', 'unter 45 Jahren', 250 FROM DUAL UNION ALL
    SELECT 'Produkt 2', 'über 45 Jahren',  375 FROM DUAL
   )
SELECT * FROM tab
   PIVOT (SUM(val) Result
     FOR age IN ('unter 45 Jahren', 'über 45 Jahren'))
ORDER BY 1;
```

Listing 2 - Beispiel einer Pivot-Abfrage

Dieses Select liefert eine ähnliche Kreuztabelle wie in Tabelle 1 dargestellt (siehe Abbildung 1). Wie man sieht muss als Zellwert eine Aggregatsfunktion genutzt werden und jede Spalte die „umgebrochen“ wird, muss manuell angegeben werden. Vorteil der Methode ist es, dass die Daten direkt von der Datenbank zur Darstellung bereitgestellt werden. Allerdings wiederspricht dies dem Prinzip der Trennung von Daten und Layout. Außerdem muss das Statement um die Pivot-Klausel ergänzt werden, was die Nutzung des Statements in anderen Bereichen nicht mehr ermöglicht.


Results Explain Describe Saved SQL History

PRODUCT	'unter 45 Jahren'_RESULT	'über 45 Jahren'_RESULT
Produkt 1	450	475
Produkt 2	250	375

2 rows returned in 0.00 seconds Download

Abbildung 1 – Ergebnis der Pivot-Abfrage

PL/SQL

Ob mit Packages oder direkt innerhalb einer PL/SQL-Region, diese Möglichkeit bietet große Design- und Entwicklungsfreiheiten. Durch das htp-Package kann reines HTML ausgegeben werden.

```
BEGIN
  htp.p('<div id="report_MATRIX_DEMO_catch">');
  htp.p('<table id="report_MATRIX_DEMO" class="report-holder"
cellspacing="0" cellpadding="0" border="0">');
  htp.p('<tbody>');
  htp.p('<table class="report-standard-alternatingrowcolors"
border=0 cellpadding=0 cellspacing=0>');
  htp.p('<tbody>');
  htp.p('<tr>');
  -- Upper left corner
  htp.p('<th class="header">Matrix</th>');
  -- dim2
  htp.p('<th class="header">unter 45 Jahren</th>');
  htp.p('<th class="header">über 45 Jahren</th>');
  htp.p('</tr>');
  htp.p('<tr>');
  -- Dim1
  htp.p('<th class="header">Produkt 1</th>');
  -- Values
  htp.p('<td class="data">450</td>');
  htp.p('<td class="data">475</td>');
  htp.p('</tr>');
  htp.p('<tr>');
  -- Dim1
  htp.p('<th class="header">Produkt 2</th>');
  -- Values
```

```
  htp.p('<td class="dataalt">' ||apex_item.text(1,250) ||'</td>');
  htp.p('<td class="dataalt">' ||apex_item.text(2,375) ||'</td>');
  htp.p('</tr>');
  htp.p('</tbody></table>');
  htp.p('</tbody></table>');
  htp.p('</div>');
END;
```

Listing 2 - Beispiel des htp-Packages

Wie in Listing 2 zu sehen, ist es sehr einfach durch das htp-Package HTML-Code zu erzeugen. Außerdem ist es denkbar, den PL/SQL-Code durch Parameter steuerbar zu machen, so dass verschiedene Parameter genutzt werden können:

- Schriftstil (Farbe, Größe, Art, etc.)
- Ausrichtung der Überschriften
- Stil der Tabellenzellen

Ein großer Vorteil dieses Verfahrens ist es, dass es automatisch die Anzahl der Spalten erkennt und nicht „manuell“ (Pivot-Klausel) jede Spalte angelegt werden muss. Desweiteren ist die Modularisierung des Programmcodes ein großer Vorteil von PL/SQL-Code. Einmal erzeugt kann der Programmcode für (jedes) Statement genutzt werden. So kann durch einen Ref-Cursor auch das Statement als Parameter übergeben werden. Das Statement muss lediglich um die Zeilen- und Spalten-Nr. erweitert werden. Sobald die Struktur des Statements den Modulanforderungen entspricht, kann das Package automatisch einen Matrix-Report erzeugen. In Abbildung 2 ist der aus Listing 2 gezeigte Matrix Report dargestellt.

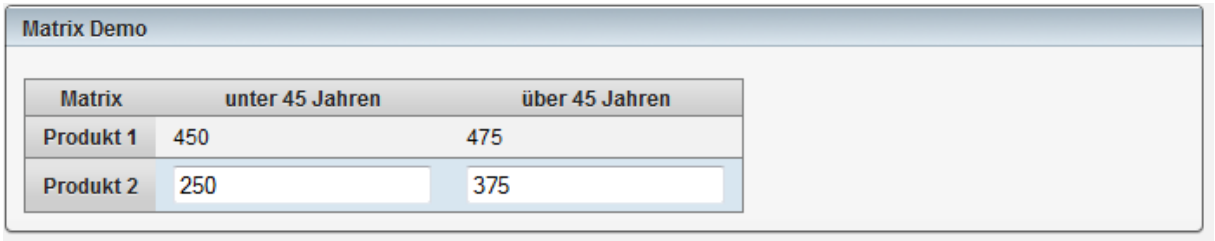


Abbildung 2 – Matrix Report aus PL/SQL-Region

Report Template(s)

Im Gegensatz zur PL/SQL-Variante benötigt die Template Variante keinen PL/SQL-Code. Lediglich das Select-Statement muss ähnlich dem Package um die Zeilen- und Spalten-Nummern erweitert werden. Die restliche Logik übernimmt die Apex eigene Template-Engine. Um einen Standard Report als Matrix darstellen zu können, muss zunächst ein neues Template erstellt werden. Dazu muss ein Report Template mit dem Template Type *Named Columns (row template)* ausgewählt werden. Bei diesem Template Typ können bis zu vier unterschiedliche Templates für die Zeilen einer Abfrage angegeben und mit eigenen Bedingungen versehen werden.

Die Auswertung der einzelnen Templates beginnt bei *Row Template 1* und endet bei *Row Template* 4. Sobald eine Bedingung zutrifft, wird für den aktuellen Datensatz das entsprechende Template benutzt.

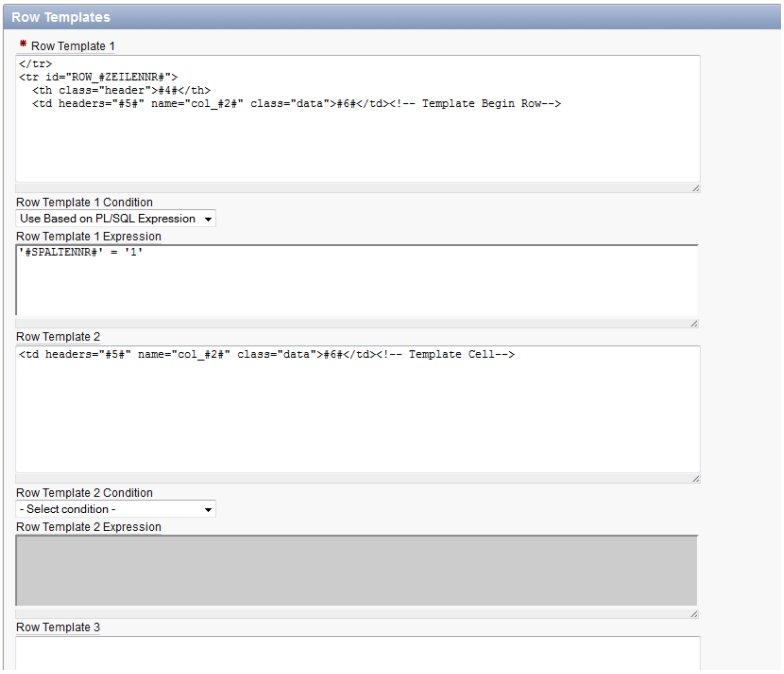


Abbildung 3 – Ansicht eines Row Template

Durch diesen Mechanismus ist es möglich zielgenau die Datensätze für ein bestimmtes Template auszuwählen. Für einen simplen Matrix-Report werden lediglich die ersten beiden *Row Templates* und ein wenig JavaScript benötigt (siehe Abbildung 3).

Row Template 1

Dieses Template sollte immer dann angewendet werden, wenn der erste Wert einer Datenreihe durch die Datenbank geliefert wird. Es gibt neben dem Wert auch die Zeilenüberschrift aus. Desweiteren können den HTML-Zeilen und –Spalten CSS-Klassen hinzugefügt werden um direkt den gewünschten Stil zu erhalten.

Row Template 2

Dieses Template liefert bedingungslos eine Tabellenzelle mit dem Zellwert als Inhalt zurück. Äquivalent zum *Row Template 1* können hier ebenfalls weitergehende Styles dem Template hinzugefügt werden.

JavaScript

Unter dem Punkt *After Rows* muss nun ein kleines JavaScript erstellt werden, welches die Überschriftenzeile erstellt. Um dies erreichen zu können, muss in den *Row Templates 1* und *2* zusätzlich ein weiteres Attribut der Tabellenzelle hinzugefügt werden: *headers*. Dieses Attribut wird benutzt, um temporär die Überschrift zu speichern. Das JavaScript nutzt diese Information um der HTML-Tabelle die Überschriftenzeile hinzuzufügen.
Durch dieses einfache Konstrukt ist es bereits möglich einen Matrix-Report aus einem „normalem“-Select heraus zu erzeugen (siehe Abbildung 4).

	Solobus	Gelenkbus	Gesamt
Einsatzwagen	1	0	1
Gesamt	5	9	4
Kurswagen	2	1	3
Reparatur	1	0	1
Reserve	0	1	1
Sonderfahrt	2	0	2
Verleih	0	1	1

Abbildung 4 – Ansicht eines Matrix-Reports

Fazit

Je nach Anwendungszweck passt eine andere Variante. So lassen sich mit der Oracle-eigenen Pivot-Klausel schnell und einfach Kreuztabellen erzeugen. Allerdings ist dieser Ansatz nicht sehr generisch und bedarf manuellen Eingreifens, sofern sich die Daten ändern. Außerdem kann kein großer Einfluss auf den Stil der Darstellung genommen werden. Dem gegenüber steht das PL/SQL-Package, welches durch seine Flexibilität bezogen auf Daten und Funktionalität besonders für stark generische Anwendungsgebiete geeignet ist (z.B. Nutzung in mehreren Applikationen). Die Template-Variante hingegen vereint Vorteile aus beiden vorgenannten Methoden. Sie bietet hinreichende Flexibilität bezogen auf die Funktionalität der Kreuztabelle (sofern mehrere Templates erstellt werden), ist allerdings nicht so schnell umgesetzt, wie ein Pivot-Select.
Daher sollten vor der Wahl der entsprechenden Methode folgende Punkte gegeneinander abgewogen werden: Flexibilität, Schnelligkeit, Einfachheit

Kontaktadresse:

Thomas Hernando Gotthardt
syntegris information solutions GmbH
Hermann-Straße, 54-56
D-63263 Neu-Isenburg

Telefon: +49 (0) 6102 29 86 68
Fax: +49 (0) 6102 55 88 06
E-Mail thomas.hernando@syntegris.de
Internet: www.syntegris.de