

Formatierte Berichte

Excel-Dateien mittels PL/SQL erzeugen

Robert Marz
its-people

Schlüsselworte:

Excel Dateien XLS, XLSX PL/SQL Formatiert

Einleitung

Mal eben schnell Daten aus der Datenbank als Excel-Arbeitsmappe zur Weiterverarbeitung durch die Fachabteilungen aufzubereiten, ist eine Herausforderung, der sich die meisten Datenbankentwickler und Administratoren schon einmal stellen mussten.

Wenn man dazu auf Krücken wie csv-Dateien oder Copy and Paste zurückgreifen muss, sind die Ergebnisse oft unbefriedigend oder erfordern eine Menge manuelle Nacharbeit.

Dabei ist es relativ einfach, mittels PL/SQL natives Excel 2007 bzw. 2010 (xlsx) oder sogar Excel 2003 (xls) mit Formatierungen oder gar Formeln zu erzeugen.

Microsoft Office XML Formate

Das Bedürfnis, Office Dokumente maschinell zu erzeugen, gibt es schon lange Zeit. Da Microsoft die Binär-Formate unter Verschluss hält, müssen andere Wege beschritten werden. Reverse Engineering dieser Formate, ist – obwohl mehrfach versucht – umständlich, fehleranfällig und niemals vollumfänglich.

Bereits Anfang dieses Jahrtausend hat Microsoft für Office 2003 und Office XP die „Microsoft Office XML“ Formate definiert. Dateien in diesen Formaten sind XML-Dokumente, die aber wie die Binär-Dokumente benannt werden, also statt emp.xml emp.xls heißen.

Diese Dateien öffnen sich klaglos im jeweiligen Office-Programm.

Die XML-Dokumente können mit wenigen Einschränkungen den kompletten Funktionsumfang der nativen binären Dokumentenformate haben, inklusive Formate, Formeln, oder Autofilter.

Die XML-Tag-Hierarchie einer Excel-Mappe, sieht wie folgt aus(stark gekürzt):

```
<ss:Workbook>
  <ss:Styles>
</ss:Styles>
  <ss:Worksheet>
    <ss:Table>
      <ss:Row>
        <ss:Cell>
          <ss:Data>
</ss:Data>
        </ss:Cell>
      </ss:Row>
    </ss:Table>
  </ss:Worksheet>
</ss:Workbook>
```

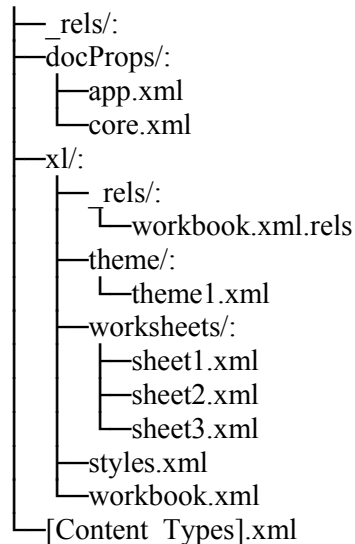
Mit der Microsoft Office Suite 2007 hat Microsoft dann „Office Open XML“ als offenen Dateiformat-Standard für alle Office Dokumente eingeführt. Dateien in diesem Format erkennt man an dem zusätzlichen „x“ an der Dateiendung, also z.B. „.docx“ oder „.xlsx“. Diese Formate gelten auch für die Office Suite 2010.

Diese neuen Formate sind unter den Bezeichnungen ECMA-376 1st edition (2006) und ISO/IEC 29500 (2008) standardisiert. Der Standard selbst ist sehr umfangreich, die Dokumentation hat mehrere tausend Seiten. Um einfache Dokumente zu erzeugen muss ja zum Glück nicht jedes Feature implementiert werden, sodass die letztendlichen Dateien ähnlich aufgebaut sind, wie die 2003 Varianten.

Ein wesentlicher Unterschied ist, dass Office 2007 Dokumente aus mehreren XML-Dateien bestehen, die nach den Open „Packaging Conventions“ organisiert und als ZIP-Archiv zu einer Datei zusammengefasst werden.

Wenn man ein solches Dokument entpackt (in diesem Fall eine leere Excel-Mappe), erhält man folgende Dateistruktur:

leer-extrakt.xlsx/:



Die meisten Dateiinhalte sind für einfache, generierte Berichte typischerweise statisch und müssen nicht verändert werden. Lediglich die einzelnen Blätter (Sheets) müssen in der `workbook.xml` eingetragen und im Verzeichnis `worksheets` angelegt werden.

Erzeugung mittels PL/SQL

Mit dem oben beschriebenen Vorwissen ausgestattet, läuft das Erzeugen von Excel-Dateien mit PL/SQL im wesentlichen auf das Arbeiten mit XML in der Datenbank hinaus. Mit `xmltype` bietet die Oracle Datenbank schon seit einiger Zeit einen mächtigen Datentyp, der sich mit SQL-eigenen Funktionen anlegen und bearbeiten lässt.

Die XML-Dokumente zerfallen in mehrere Einheiten, von denen die meisten Inhalte statisch sind und sich auf diese Weise hervorragend eignen, als Template in einer Tabelle vorgehalten und schnell miteinander verknüpft zu werden.

Wenn die Dokumente nicht nur einmalig erzeugt werden, sondern auch für länger in der Datenbank vorgehalten werden sollten, bietet es sich an, dafür ein einfaches Datenmodell zu verwenden, dessen Tabellen auch die einzelnen Schritte der Erstellung widerspiegeln:

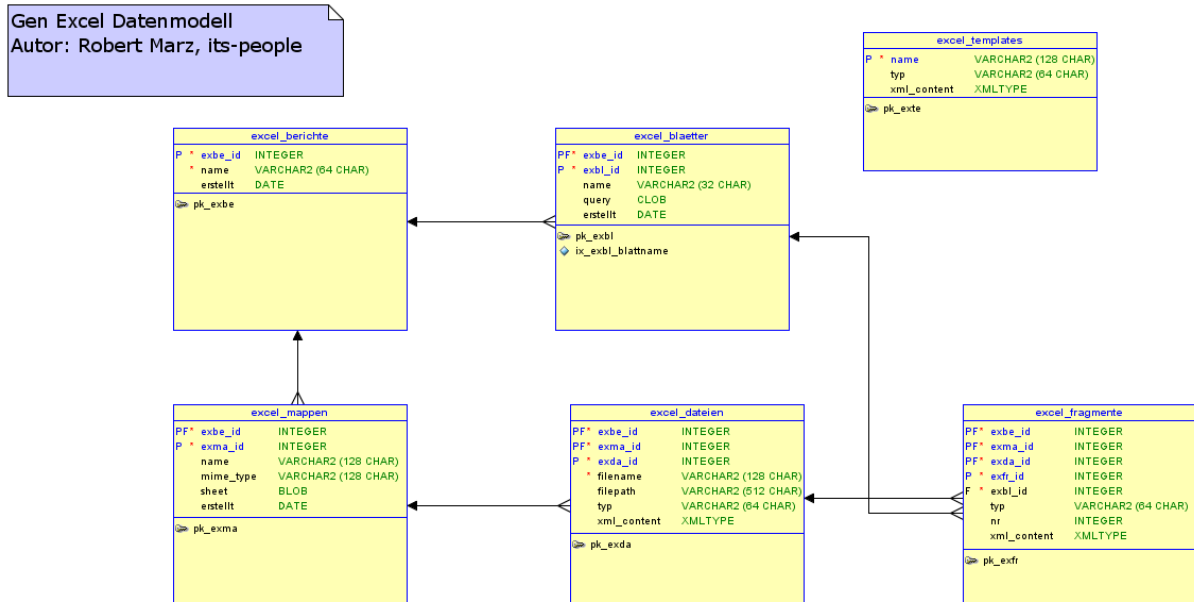


Abb. 1: Datenmodell für die Generierung

In diesem Beispiel zerfällt ein Excel Bericht logisch in einzelne Blätter, mit Namen und der SQL-Abfrage, die später den Inhalt des Blattes ergibt.

Physisch besteht ein Bericht aus einer Mappe. Da die Mappen in unterschiedlichen Formaten (z.B. als Office 2003 und als Office 2010) vorliegen oder Versionen zu verschiedenen Zeiten erzeugt werden können, kann es natürlich durchaus mehrere Mappen pro Bericht geben. Jede Mappe wird als Blob gespeichert.

Die Mappen bestehen aus XML-Dateien, die in der entsprechenden Tabelle als xmltype gespeichert sind. Für Office 2003 ergibt das ein XML-Dokument pro Mappe, Office 2007 Dateien setzen sich aus mehreren XML-Dokumenten zusammen.

Jede XML-Datei schließlich besteht aus mehreren Fragmenten, die einzeln erzeugt werden. Teils werden dazu Templates aus der der passenden Tabelle herangezogen. Kandidaten für Templates sind z.B. die eingesetzten Formate (Styles), aber auch die Rohstrukturen der Mappen (Workbooks) und Blätter (Worksheets).

Die Implementierung eines funktionsfähigen Packages birgt nur wenige Stolpersteine. Einige ausgewählte Herausforderungen sind:

- Die mehrfachen durch Microsoft vorgegebenen Namespaces in den XML-Dokumenten. Die Oracle-XML-Funktionen machen das Arbeiten mit solchen Dokumenten nicht immer einfach. Im Extremfall kann man durchaus von der reinen Lehre abweichen und sich

behelfen, indem der xmltype in ein clob serialisiert und nach dem Bearbeiten mit Funktionen wie regexp_replace() wieder in einen xmltype zurückverwandelt wird.

- Das dynamische Parsen einer beliebigen Abfrage mit Hilfe des Packages dbms_sql. Das Ermitteln der zurückgelieferten Spalten mit ihren Datentypen und das Aufnehmen der Werte ist zwar dokumentiert, das Vorgehen selbst birgt aber seine Tücken.
- Das Erstellen von ZIP-Archiven, die mehrere Dateien mit relativen Pfaden beinhalten können, ist in PL/SQL recht aufwendig, weil die zugehörigen Packages selbst entwickelt werden müssten. Einfacher geht es mit Java in der Datenbank, für das dann entsprechende Wrapper-Prozeduren für PL/SQL erstellt werden.

Der Quelltext für funktionsfähige Packages kann nach der Konferenz von unserem Blog auf der its-people Homepage heruntergeladen werden: <http://www.its-people.de/blog>

Kontaktadresse:

Robert Marz

its-people Hochtaunus GmbH
Lyoner Straße 44-48
60528 Frankfurt am Main

Telefon: +49 (69) 247521-00
Fax: +49 (69) 247521-021
E-Mail: robert.marz@its-people.de
Internet: www.its-people.de