

10. Übung "XQuery"

Hinweise zur Lösung der Aufgaben:

1. Benutzen Sie für die nachfolgenden XQuery-Anfragen im SQL-Developer die **Funktion ora:view()**:

```
SELECT XMLQuery('
    <result>{
        for $m in ora:view("companyXML")//Mitarbeiter
        return $m/Name
    }</result>
'RETURNING CONTENT ) AS Angestellte
FROM DUAL;
```

2. Setzen Sie zu Beginn jeder Session im SQL-Developer wieder die nachfolgenden **Systemvariable**, um vollständige Ausgaben zu erhalten:

```
set lines 10000
set long 10000
set pagesize 1000
```

3. Die **Zeilenumbrüche** bei den vorgegebenen Ausgaben einiger Aufgaben (1c, 1d, 1e ...) dienen nur der besseren Visualisierung des erwarteten Ergebnisses und müssen durch die XQuery-Anfrage nicht erzeugt werden.

In einigen Fällen kann der Einbau eines Zeilenumbruchs (siehe Skript zur Übung) aber eine Kontrolle der Lösung vereinfachen.

Aufgabe 1 – Einfache FLWOR-Anfragen

Starten Sie auf der Tabelle companyXML aus Übung9 die nachfolgenden XQuery-Anfragen mit der XMLQuery()-Funktion:

- a) Alle Daten der Mitarbeiter.
- b) Die Namen aller Mitarbeiter aufsteigend sortiert.
- c) Alle Angestellten in der Form:

```
<Angestellter>
  <ID>7839</ID>
  <EDat>1981-11-17</EDat>
  <Name>King</Name>
  <Beruf>President</Beruf>
  <Gehalt>5000</Gehalt>
</Angestellter>
<Angestellter>
  ...
</Angestellter>
...
```

d) Alle Abteilungen in der Form:

```
<Abteilung>10</Abteilung>
<Abteilung>20</Abteilung>
<Abteilung>30</Abteilung>
```

e) Name und Gehalt aller Angestellten, die mindestens 2500\$ verdienen - absteigend sortiert nach Gehalt - in der Form:

```
<topdog>
  <Name>King</Name>
  <Gehalt>5000</Gehalt>
</topdog>
...
```

Aufgabe 2 – Verbundoperationen

Erzeugen Sie aus den Daten der Tabelle SALGRADE eine XML-Instanz

```
<Salgrade>
  <Stufe>1</Stufe>
  <Min>700</Min>
  <Max>1200</Max>
</Salgrade>
<Salgrade>
  <Stufe>2</Stufe>
  ...
</Salgrade>
...
```

und speichern Sie diese direkt in einer Tabelle salgradesXML (siehe Übung 9).

Erzeugen Sie anschließend die nachfolgenden Anfragen:

- a)** Alle Angestellten (Name) mit Gehalt und Gehaltsstufe, aufsteigend sortiert nach Gehalt.
- b)** Die Obergrenze der Gehaltsstufe von SCOTT.
- c)** Verändern Sie Anfrage 2b) so, dass der Name des Angestellten zur Laufzeit eingegeben werden kann.
- d)** Gehaltsstufen und Angestellte in der Form

```
<Gehaltsstufe stufe="1">
  <Angestellter>
    <Name>Smith</Name>
    <Gehalt>800</Gehalt>
  </Angestellter>
  <Angestellter>
    <Name>Adams</Name>
    <Gehalt>1122</Gehalt>
  </Angestellter>
  ...
</Gehaltsstufe>
<Gehaltsstufe stufe="2">
  <Angestellter>
    <Name>Meier</Name>
    <Gehalt>1300</Gehalt>
  </Angestellter>
  ...
</Gehaltsstufe>
...
```

Aufgabe 3 – Gruppierungen

Erzeugen Sie die nachfolgenden Anfragen:

- a)** Die Anzahl der Angestellten zu jeder Abteilung in der Form

```
<Abteilung>
  <AbtNr>10</AbtNr>
  <AnzAngest>3</AnzAngest>
</Abteilung>
...
```

- b)** Die Abteilung mit den meisten Angestellten.

Hinweis:

Speichern Sie das Ergebnis der for-Klausel aus Aufgabe 3a mit einer LET-Anweisung in einer Variable. Iterieren Sie anschließend mit einer FOR-Anweisung durch diese Sequenz. Mit Hilfe der `max()`-Funktion kann in der WHERE-Klausel der größten Wert im Element `AnzAngest` festgestellt werden. Im RETURN ist der dazugehörige komplette Knoten auszugeben.

Zusatzaufgabe 4 - HTML-Ausgabe

- a)** Erzeugung Sie eine tabellarische HTMLAusgabe für alle Mitarbeiter (Spalten: Id, Name, Beruf, Gehalt, Einstellungsdatum). Überprüfen Sie den HTML-Output in einem HTML-Online-Tester, z.B. unter http://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro
- b)** Erweiterung Sie a) um eine zusätzliche Tabellenspalte mit der Abteilungsnummer zu jedem Angestellten.