

РАБОТА С XML

Сичная Мария, начальник бюро платежных приложений

XMLElement принимает имя элемента в качестве идентификатора, необязательный набор атрибутов для элемента и аргументы, составляющие содержимое элемента.

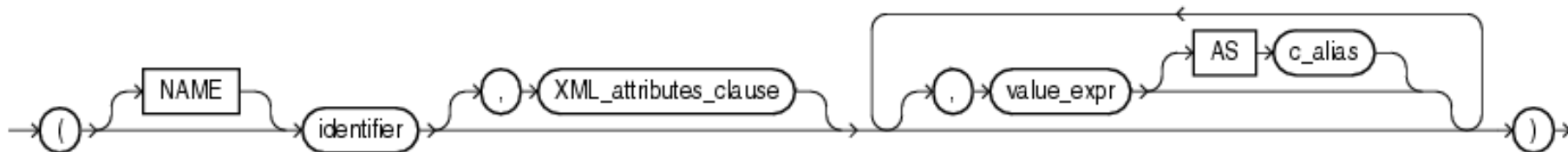
Он возвращает экземпляр типа XMLType.

Функция XMLElement обычно является вложенной для создания XML-документа с вложенной структурой.

Необходимо указать значение для идентификатора, который Oracle Database использует в качестве включающего тега. Идентификатор может содержать до 4000 символов и не обязательно должен быть именем столбца или ссылкой на столбец.

Это не может быть выражение или значение NULL.

→ XMLEMENT →



```

SELECT XMLEMENT("Department",
               XMLEMENT("Deptno", DEPTNO),
               XMLEMENT("Title", DNAME),
               XMLEMENT("Location", LOC))
FROM DEPT
WHERE DEPTNO = 40;
  
```

```

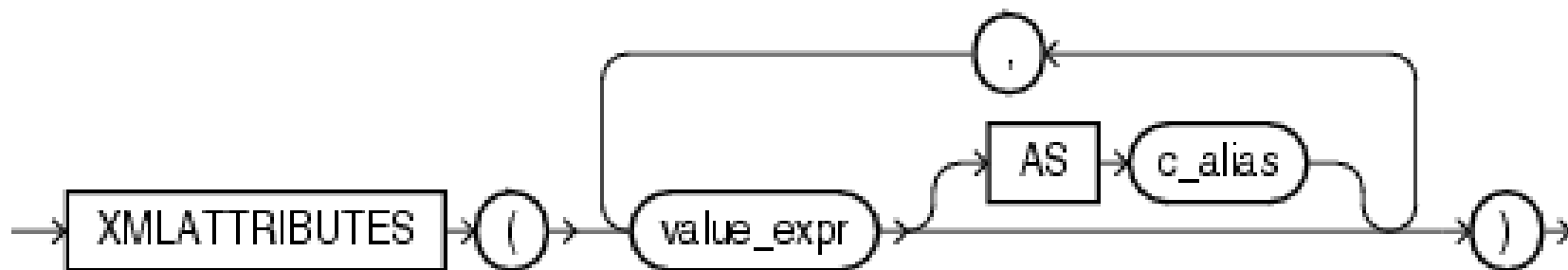
<Department>
  <Deptno>40</Deptno>
  <Title>OPERATIONS</Title>
  <Location>BOSTON</Location>
</Department>
  
```

```
SELECT
  XMLELEMENT(null,
    XMLELEMENT("Deptno", DEPTNO),
    XMLELEMENT("Title", DNAME),
    XMLELEMENT("Location", LOC))
FROM DEPT
WHERE DEPTNO = 40;
```

```
SELECT
  XMLELEMENT(5+3,
    XMLELEMENT("Deptno", DEPTNO),
    XMLELEMENT("Title", DNAME),
    XMLELEMENT("Location", LOC))
FROM DEPT
WHERE DEPTNO = 40;
```

ORA-00931: отсутствует идентификатор
00931. 00000 - "missing identifier"

В предложении XML_attributes_clause, если value_expr имеет значение null, тогда для этого выражения значения не создается никакой атрибут. Если вы укажете псевдоним для value_expr с помощью предложения AS, c_alias может иметь длину до 4000 символов.



Для необязательного выражения `value_expr`, которое следует за предложением `XML_attributes_` на диаграмме:

- Если `value_expr` является скалярным выражением, то вы можете опустить предложение `AS`, и Oracle использует имя столбца в качестве имени элемента.
- Если `value_expr` является типом объекта или коллекцией, тогда предложение `AS` является обязательным, и Oracle использует указанный `s_alias` в качестве включающего тега.
- Если `value_expr` имеет значение `null`, то для этого выражения значения не создается никаких элементов. Обратите внимание, что предложение идентификатора `AS` не было указано для столбца `last_name`.

```
SELECT
  XMLELEMENT("Department",
    XMLATTRIBUTES(DEPTNO AS "Dept_number", DNAME),
    XMLELEMENT("Location", LOC))
FROM DEPT
WHERE DEPTNO = 20;
```

```
<Department Dept_number="20" DNAME="RESEARCH">
  <Location>DALLAS</Location>
</Department>
```

```
SELECT
  XMLELEMENT("Emp",
    XMLATTRIBUTES(e.EMPNO, e.ENAME),
    XMLELEMENT("Dept",
      XMLATTRIBUTES(e.DEPTNO, (SELECT d.DNAME
                                FROM DEPT d
                                WHERE d.DEPTNO = e.DEPTNO)
                        as "Dept_name")),
    XMLELEMENT("salary", e.SAL),
    XMLELEMENT("Hiredate", e.HIREDATE))
FROM EMP e
WHERE EMPNO = 7839;
```

```
<Emp EMPNO="7839" ENAME="KING">
  <Dept DEPTNO="10" Dept_name="ACCOUNTING"/>
  <salary>5000</salary>
  <Hiredate>1981-11-17</Hiredate>
</Emp>
```

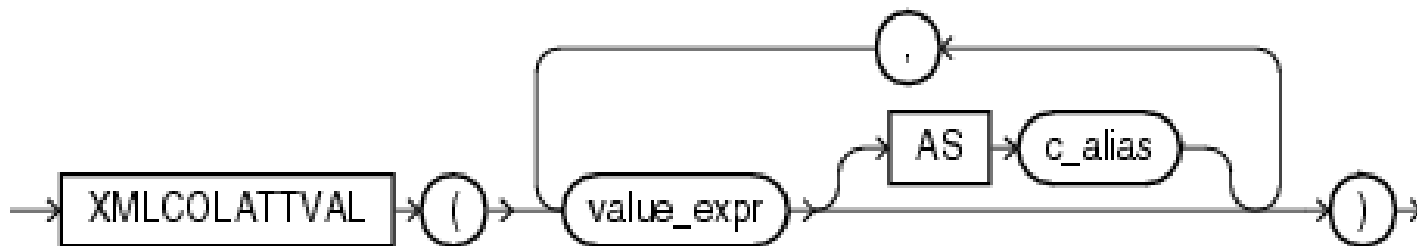

SELECT

```
XMLELEMENT("Emp",  
  XMLATTRIBUTES(e.EMPNO, e.ENAME),  
  XMLELEMENT("Dept",  
    XMLATTRIBUTES(e.DEPTNO, (SELECT d.DNAME  
                                FROM DEPT d  
                                WHERE d.DEPTNO = e.DEPTNO)  
                                as "Dept_name")),  
  XMLELEMENT("salary", e.SAL),  
  XMLELEMENT("Hiredate", e.HIREDATE))  
FROM EMP e  
WHERE EMPNO = 7839;
```

```
<Emp EMPNO="7839" ENAME="KING">  
  <Dept DEPTNO="10" Dept_name="ACCOUNTING"/>  
  <salary>5000</salary>  
  <Hiredate>1981-11-17</Hiredate>  
</Emp>
```

XMLColAttVal создает фрагмент XML, а затем расширяет полученный XML так, чтобы каждый фрагмент XML имел столбец имени с именем атрибута. Вы можете использовать предложение AS с alias, чтобы изменить значение атрибута name на что-то иное, чем имя столбца.

Вы должны указать значение для value_expr. Если значение_выражения равно нулю, то никакой элемент не возвращается.

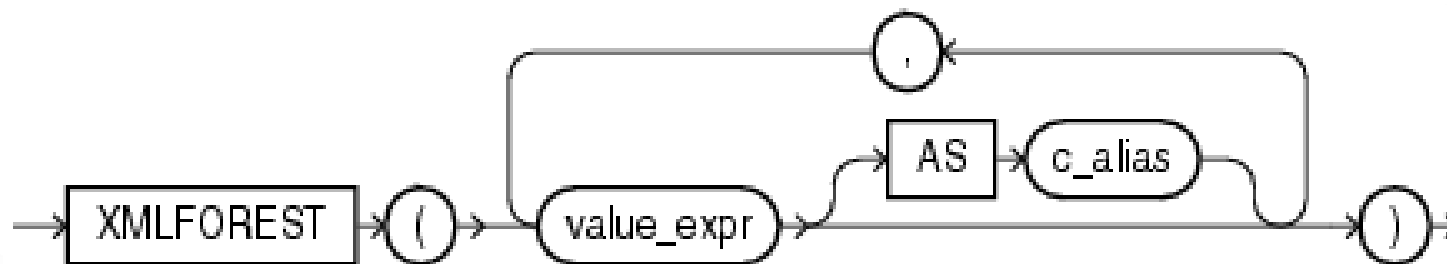


```
SELECT  
  XMLELEMENT("Employee",  
    XMLCOLATTVAL(EMPNO, ENAME, SAL, COMM))  
FROM EMP  
WHERE EMPNO = 7839;
```

```
<Employee>  
  <column name="EMPNO">7839</column>  
  <column name="ENAME">KING</column>  
  <column name="SAL">5000</column>  
  <column name="COMM"/>  
</Employee>
```

XMLForest преобразует каждый из своих параметров аргумента в XML, а затем возвращает фрагмент XML, который является объединением этих преобразованных аргументов.

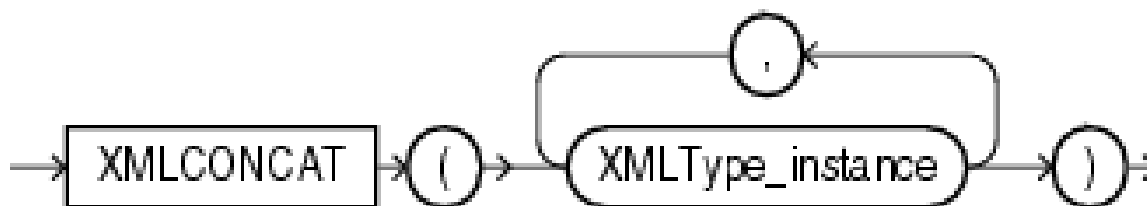
- Если `value_expr` является скалярным выражением, то вы можете опустить предложение `AS`, и Oracle Database использует имя столбца в качестве имени элемента.
- Если `value_expr` является типом объекта или коллекцией, тогда предложение `AS` является обязательным, и Oracle использует указанный `c_alias` в качестве включающего тега. `C_alias` может содержать до 4000 символов.
- Если `value_expr` имеет значение `null`, тогда для этого `value_expr` не создается никаких элементов.



```
SELECT
  XMLELEMENT("Employee",
    XMLFOREST(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL,
      COMM))
FROM EMP
WHERE EMPNO = 7839;
```

```
<Employee>
  <EMPNO>7839</EMPNO>
  <ENAME>KING</ENAME>
  <JOB>PRESIDENT</JOB>
  <HIREDATE>1981-11-17</HIREDATE>
  <SAL>5000</SAL>
</Employee>
```

XMLConcat создает фрагмент XML путем объединения нескольких экземпляров XMLType. Функция XMLConcat принимает произвольное количество экземпляров XMLType и объединяет их вместе. Если одно из значений NULL, то оно игнорируется в результате. Если все значения равны ПУСТО (NULL), результат будет ПУСТО (NULL). Эта форма используется для объединения произвольного количества экземпляров XMLType в одну строку.



```
SELECT
  XMLConcat(XMLElement("EMPNO", e.EMPNO),
            XMLElement("ENAME", e.ENAME)) RESULT
FROM EMP e;
```

```
<EMPNO>7839</EMPNO><ENAME>KING</ENAME>
<EMPNO>7698</EMPNO><ENAME>BLAKE</ENAME>
<EMPNO>7782</EMPNO><ENAME>CLARK</ENAME>
<EMPNO>7566</EMPNO><ENAME>JONES</ENAME>
<EMPNO>7788</EMPNO><ENAME>SCOTT</ENAME>
...
<EMPNO>7934</EMPNO><ENAME>MILLER</ENAME>
```

XMLAgg - это агрегатная функция. Он принимает набор фрагментов XML и возвращает агрегированный документ XML. Все аргументы, возвращающие значение null, исключаются из результата.



SELECT

```
XMLELEMENT("Department",  
  XMLATTRIBUTES(DEPT.DEPTNO, DEPT.DNAME),  
  XMLAGG(XMLELEMENT("Employee", EMP.ENAME)  
    ORDER BY EMP.ENAME))
```

FROM EMP

```
JOIN DEPT ON DEPT.DEPTNO = EMP.DEPTNO
```

```
WHERE EMP.DEPTNO = 30
```

```
GROUP BY DEPT.DEPTNO, DEPT.DNAME;
```

```
<Department DEPTNO="30" DNAME="SALES">
```

```
  <Employee>ALLEN</Employee>
```

```
  <Employee>BLAKE</Employee>
```

```
  <Employee>JAMES</Employee>
```

```
  <Employee>MARTIN</Employee>
```

```
  <Employee>TURNER</Employee>
```

```
  <Employee>WARD</Employee>
```

```
</Department>
```

```
select vEname, vSal, vSalgrade
from
  XmlTable('Department'
    passing xmltype(
      '<Department DEPTNO="10" DNAME="ACCOUNTING">
        <Employee Emp_number="7782" JOB="MANAGER">
          <ENAME>CLARK</ENAME>
          <SAL>2450</SAL>
          <Salgrade>4</Salgrade>
        </Employee>
      </Department>')
    columns
      vEname   varchar2(100) path 'Employee/ENAME',
      vSal     integer        path 'Employee/SAL',
      vSalgrade integer       path 'Employee/Salgrade'
  );
```

https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions228.htm

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?

www.compassplus.ru