

### Exercise #1: XML resource management

1. Download the `customers.xml` document from <http://eric.univ-lyon2.fr/~jdarmont/docs/customers.xml>.
2. Connect to Oracle. In an anonymous PL/SQL block, declare a string variable `xml_doc` of size 50 and initialize it to  `'/public/username_cust.xml'` (e.g.,  `'/public/m100_cust.xml'`). Declare another string variable `xml_data` of size 2500 and initialize it to the contents of `customers.xml` (cut and paste). Finally, declare a boolean variable `res`. In the code section, call function `DBMS_XDB.createResource` with the parameters `xml_doc` and `xml_data`. Store the function result into `res`. Test!
3. Execute the PL/SQL block again. Conclusion? Before the call to function `DBMS_XDB.createResource`, add into the code section a call to procedure `DBMS_XDB.deleteResource` with the parameter `xml_doc`. Test!
4. After the calls to procedure `DBMS_XDB.deleteResource` and function `DBMS_XDB.createResource`, respectively, add into the code section calls to function `DBMS_XDB.existsResource` with the parameter `xml_doc`. Store the function result into `res` and display the message “XML document exists.” if `res` is true, and “XML document does not exist.” otherwise. Test!
5. Since repeating the code for testing whether the XML document exists is not a good idea, “factorize” it in a procedure. Test!

### Exercise #2: XQueries over XML and relational data

#### General form of an XQuery under Oracle:

```
SELECT XMLQuery('insert XQuery here' RETURNING CONTENT) FROM DUAL;
```

1. List the names of all customers from document  `/public/username_cust.xml`.
2. Knowing that a relational table can be queried with XQuery by replacing the document clause `doc("document_path")` by `ora:view("TABLE_NAME")`, and that the “root element” of a table is `ROW`, list the names (`ENAME`) of all employees from the table `EMP` created during semester #1 (you may copy it from `DARMONT.EMP` if you deleted it). Note: you may omit the `ROW` element by using a `//` path to `ENAME`.
3. List employee names (`ENAME`) and department names (`DNAME`) from relational tables `EMP` and `DEPT`. You may copy `DEPT` from `DARMONT.DEPT` if you deleted it.

4. Copy relational table `DARMONT.CUST_TYPE` on your own account. Its extension follows.

```
CUST_ID  TYPE
-----
cust01   Good
cust02   Average
cust03   Very good
cust04   Very good
cust05   Average
```

List customer names from document  `/public/username_cust.xml` along with their type from `CUST_TYPE` (XML-relational join!).

#### Variant form of XQuery under Oracle:

```
SELECT * FROM XMLTable('insert XQuery here')i
```

`XMLTable` is a virtual relational table with only one attribute `COLUMN_VALUE`. Each tuple in `XMLTable` is an XML fragment (of type `XMLType`).

5. List customers from document  `/public/username_cust.xml` using an `XMLTable` query.
6. Convert the XML attributes and elements `@id`, `name` and `phone` into relational data by adding a clause of the following form to the previous query.  

```
COLUMNS rel_attribute1 type_of_attribute1 PATH 'xml_element_or_attribute1',
         rel_attribute2 type_of_attribute2 PATH 'xml_element_or_attribute2',
         ...
```
7. Delete document  `/public/username_cust.xml`.