Database Architectures

**Practical Assessment #2 (PR2):**

**XML Extension**

***Students:*** *Carlos Del Blanco Garcia*

*Jordi Bericat Ruz*

***Professor:*** *Maria Teresa Bordas Garcia*

***Term:*** *Autumn 2020-21 (Aula 1)*

*Table of Contents*

[Activity 1 1](#_Toc55303619)

[a) 1](#_Toc55303620)

[b) 2](#_Toc55303621)

[Activity 2 3](#_Toc55303622)

[a) 3](#_Toc55303623)

[b) 3](#_Toc55303624)

[c) 3](#_Toc55303625)

[Activity 3 4](#_Toc55303626)

[a) 4](#_Toc55303627)

[b) 4](#_Toc55303628)

# Activity 1

## a)

## b)

The main purpose of the extensive markup language (xml) is to stablish a proper communication mechanism among applications. To achieve this, it is necessary to strictly define a structure of elements (known as vocabulary) which implies a set rules and constraint. Here is where xml schemas come into play since they allow defining that so-called vocabulary with a very high degree of details regarding the application data particularities.

That said, to define the xml schema that will stablish the required vocabulary and set of rules for the xml structure proposed in this activity statement, we will proceed as follows:

**<!-- ################### definition of the xml schema #################### -->**

<?xml version = **"1.0"** encoding = **"ISO-8859-1"** ?>

<!-- see comments section [0] -->

<xsd:schema><!-- see comments section [1] -->

**<!-- ################### definition of simple elements ################## -->**

<xsd:element name=**"id\_type"**><!--see comments section [2.1.3]-->

<xsd:simpleType>

<xsd:restriction base=**"xsd:positiveInteger"**>

<xsd:maxInclusive value=**"9999"**/>

</xsd:restriction>

</xsd:simpleType>

</xsd:element>

<xs:element name=**"year"** type=**"xs:integer"**/><!--see comments section [2.2.1]-->

<xs:element name=**"month"** type=**"xs:integer"**/><!--see comments section [2.2.1]-->

<xs:element name=**"day"** type=**"xs:integer"**/><!--see comments section [2.2.1]-->

<xs:element name=**"idRegion"** type=**"xs:integer"**/><!--see comments section [2.3.1]-->

<xs:element name=**"descripton"** type=**"xs:string"**/><!--see comments section [2.3.2]-->

<xs:element name=**"hospitalised"** type=**"xs:integer"**/><!--see comm. section[2.4.1]-->

<xs:element name=**"ICU"** type=**"xs:integer"**/><!-- see comments section [2.4.2]-->

<xsd:element name=**"gender"**><!--see comments section [2.4.3]-->

<xsd:simpleType>

<xsd:restriction base=**"xsd:NMTOKEN"**>

<xsd:enumeration value=**"female"** />

<xsd:enumeration value=**"male"** />

<xsd:enumeration value=**"other"** />

</xsd:restriction>

</xsd:simpleType>

</xsd:element>

**<!-- ##################### definition of attributes ##################### -->**

<xs:attribute name=**"id"** type=**"xs:id\_type"**/><!-- see comments section [2.1.3] -->

**<!-- ##################### definition of complex elements ##################### -->**

<xsd:element name=**"date"**><!-- see comments section [2.2] -->

<xsd:complexType>

<xsd:sequence>

<xs:element ref=**"year"**/>

<xs:element ref=**"month"**/>

<xs:element ref=**"day"**/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name=**"region"**><!-- see comments section [2.3] -->

<xsd:complexType>

<xsd:sequence>

<xs:element ref=**"idRegion"**/>

<xs:element ref=**"description"**/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name=**"patients"**><!-- see comments section [2.4] -->

<xsd:complexType>

<xsd:sequence>

<xs:element ref=**"hospitalised"**/>

<xs:element ref=**"ICU"**/>

<xs:element ref=**"gender"**/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

**<!-- ##################### root element ##################### -->**

<xs:element name=**"COVID"**><!-- see comments section [2.1.1] -->

<xs:complexType>

<xs:sequence>

<xs:element ref=**"date"**/>

<xs:element ref=**"region"**/>

<xs:element ref=**"patients"** maxOccurs=**"10"**/>

</xs:sequence>

<xs:attribute ref=**"id"** use=**"required"**/><!--comm. sections [2.1.2] & [2.1.3]-->

</xs:complexType>

</xs:element>

<!—xml schema definition end -->

</xs:schema>

**<!--**

**################################################################################**

**# #**

**# COMMENTS SECTION #**

**# #**

**################################################################################**

[0] -> First of all, we should decide which design method we'll use to define the xml

schema. As seen at Bibliography [#1][https://www.w3schools.com/xml/schema\_example.asp],

there are three available approaches:

1- "Simplest-yet-messy" approach: This way, "to create the schema we

could simply follow the structure in the XML document and define each element

as we find it". (footnote: literal citation from Bibliography [#1], section: "

Create an XML Schema")

2- "Divided Schema" approach: "The next design method is based on defining all elements

and attributes first, and then referring to them using the ref attribute" (footnote:

literal citation from Bibliography [#1], section: "Create an XML Schema"). This way

we achieve element types eutilization and also gets easier to read and maintain the

xml code in complex structures.

3- "Use of Name Types" approach:

Finally, we decided to use the 2nd design approach (Divide the Schema) since que

won't be upgrading the xml structure (the statement says nothing about scalability

and reusability of types), but at the same time we want to desing a readable (it has

to be assessed) and maintanable xml squema (so both team members / students who

participate on its elaboration can better undersand and improve it).

[1] -> The activity statement doesn't give any information about namespaces

(xmlns), hence we neither include any reference to it in the schema root element

declaration, nor to the "targetNamespace" attribute (the XML Schema will be

assigned to the NULL namespace).

[2] -> NOTES ABOUT ELEMENTS & ATTRIBUTES DEFINITIONS:

[2.1] -> NODE "COVID":

[2.1.1] -> The "COVID" node will be the root element of this xml schema.

On the other hand, we'll be considering that its subelements will

be appearing in the same order on the instance xml documents.

Therefore, will have to use the primitive "sequence" in the "COVID"

element definition.

[2.1.2] -> ATTRIBUTE "id" ("is mandatory" constraint): it must be declared as a

mandatory attribute, since it contains a "foreign key" value. To do so,

we have to explicitly declare it with the "use" attribute, as well as

the "required" attribute (references: module 2, page 49 &

https://www.w3schools.com/xml/schema\_simple\_attributes.asp).

# Activity 2

## a)

## b)

## c)

# Activity 3

## a)

## b)

# Bibliography

1. XML Schema Example at w3.org 🡪 <https://www.w3schools.com/xml/schema_example.asp>