

Normalizacion - Desgin

Juan Ignacio Raggio

May 3, 2025

Contents

1	Anomalias	1
2	Para no tener estos inconvenientes, se requiere tener una buena estructura	1
2.1	Por ejemplo si tengo conferencias en las que se publican papers, cada conferencia se realiza anualmente.	1
3	Como usar keys en xsd	2
3.1	unique (aparicion optativa y unica)	2
3.2	key (aparicion obligatoria y unica)	2
3.3	keyref	2

1 Anomalias

- Efectos no deseados que dificultan la manipulacion de la informacion
 - Redundancia de datos -> Anmoalias

2 Para no tener estos inconvenientes, se requiere tener una buena estructura

2.1 Por ejemplo si tengo conferencias en las que se publican papers, cada conferencia se realiza anualmente.

Seria un mal diseño tener los años asociados a los papers pues el año corresponde a las distintas ediciones de las conferencias, en caso de que quiera borrar los años repetidos, estaria perdiendo papers

- Conclusion: El año determina la edicion (año -> edicion)

3 Como usar keys en xsd

- Hay dos formas:

3.1 unique (aparicion optativa y unica)

```
...
<xs:element name="curso" type="cursoType" maxOccurs="unbounded">
  <xs:unique name="clave_curso">
    <xs:selector xpath="./alumno"/>
    <xs:field xpath="@legajo"/>
  </xs:unique>
</xs:element>
...
```

3.2 key (aparicion obligatoria y unica)

```
...
<xs:element name="curso" type="cursoType" maxOccurs="unbounded">
  <xs:key name="clave_curso">
    <xs:selector xpath="./alumno"/>
    <xs:field xpath="@legajo"/>
  </xs:key>
</xs:element>
...
```

- El mismo efecto de **unicidad**, podemos conseguirlo con el comando key. La diferencia con unique es que con key el elemento o atributo declarado como tal debe aparecer obligatoriamente en el xml. Con unique, la aparición es optativa.
- La expresion xpath en este caso es reducida
 - Tiene que tener un path relativo
 - Tiene que hacer referencia a datos dentro del elemento o inferiores
 - No puede tener predicados

3.3 keyref

En xsd existe un tag llamado keyref que permite tener referencias a keys, ejemplo:

```

<xs:element name="universidad">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="curso" type="cursoType" maxOccurs="unbounded">
        <xs:unique name="clave_curso">
          <xs:selector xpath="./alumno"/>
          <xs:field xpath="@legajo"/>
        </xs:unique>
      </xs:element>
      <xs:element name="info_st" type="studentType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:key name="PK_InfoSt">
    <xs:selector xpath="./info_st"/>
    <xs:field xpath="@legajo"/>
  </xs:key>
  <xs:keyref name="FK_st" refer="PK_InfoSt">
    <xs:selector xpath="./curso/alumno"/>
    <xs:field xpath="@legajo"/>
  </xs:keyref>
</xs:element>

```

- En nuestro ejemplo, hemos declarado como key el atributo legajo de info_{st}. Luego declaramos dentro del curso que los atributos legajo deben hacer referencia a un legajo que exista dentro de info_{st}.
- El tipo del legajo del estudiante puede ser ID en cuyo caso no podrá repetirse en el documento, o podría ser de otro tipo y al declararlo como key (o unique) no se va a poder repetir dentro del contexto. Los atributos declarados en keref no necesitan ser IDREF: pueden serlo pero no es obligatorio