Données du web

HMIN 103

Données du Web : XML Relationnel

Élèves :
Adel Terki
Emile Youssef

Enseignant :
Federico Ulliana

13 novembre 2018

Table des matières

1 Warm-up, Fonctionnalités Oracle pour XML : Stockage CLOB et Binary XML, Interrogation

```
DROP TABLE tweet_CLOB;
DROP TABLE tweet_binaryxml;
DROP TABLE film_binaryxml;

CREATE TABLE tweet_CLOB (xmlContent varchar(20), tweet CLOB);

CREATE TABLE tweet_binaryxml (xmlContent varchar(20), tweetXml XMLTYPE)
XMLTYPE COLUMN tweetXml STORE AS  BINARY XML;


CREATE TABLE film_binaryxml (xmlContent varchar(20), filmXml XMLTYPE)
XMLTYPE COLUMN filmXml STORE AS  BINARY XML;


INSERT INTO tweet_binaryxml (xmlContent,tweetXml)
VALUES ('tweet', sys.xmltype.createxml('
<tweets>
  <tweet id="t1" author="u1">
    <timestamp GMT="2"> 30 </timestamp>
    <message length="140" type="text" color="black" lang="en">
      <ref_user> example </ref_user>
      <text> absolutely smashed it at </text>
      <hashtags>
        <ref_tag>mtvlivelockdown</ref_tag>
        <ref_tag>I&lt;3XML</ref_tag>
      </hashtags>
      <text> ! Catch him at the official </text>
      <ref_user> clubmtvuk </ref_user>
      <text> after party tonight @10 pm </text>
    </message>
    <responses>
        <response>yo !<date>12/01/18</date></response>
        <response>waw!<date>14/02/18</date></response>
        <response>lets go <date>16/04/18</date></response>
    </responses>
    <gps>
      <latitude> 0 </latitude>
      <longitude> 0 </longitude>
      <altitude> 0 </altitude>
    </gps>
    <url_media>pic.com/TfXd2D3</url_media>
  </tweet>

  <tweet id="t2" author="u2">
    <timestamp GMT="2"> 31 </timestamp>
    <message length="140" type="text" color="black" lang="en">
      <ref_user> example </ref_user>
      <text> absolutely smashed it at </text>
      <hashtags>
        <ref_tag>mtvlivelockdown</ref_tag>
        <ref_tag>efforts</ref_tag>
      </hashtags>
      <text> ! Catch him at the official </text>
```

```
54        <ref_user> clubmtvuk </ref_user>
55        <text> after party tonight @10 pm </text>
56      </message>
57      <responses>
58          <response>yo !<date>12/01/18</date></response>
59          <response>waw!<date>14/02/18</date></response>
60          <response>let s go <date>16/04/18</date></response>
61      </responses>
62      <gps>
63        <latitude> 0 </latitude>
64        <longitude> 0 </longitude>
65        <altitude> 0 </altitude>
66      </gps>
67      <url_media>pic.com/TfXd2D3</url_media>
68    </tweet>
69
70    <tweet id="t3" author="u2">
71      <timestamp GMT="2"> 32 </timestamp>
72      <message length="140" type="text" color="black" lang="en">
73        <ref_user> example </ref_user>
74        <text> absolutely smashed it at </text>
75        <text> ! Catch him at the official </text>
76        <ref_user> clubmtvuk </ref_user>
77        <text> after party tonight @10 pm </text>
78      </message>
79      <responses>
80          <response>yo !<date>12/01/18</date></response>
81          <response>waw!<date>14/02/18</date></response>
82          <response>lets go <date>16/04/18</date></response>
83      </responses>
84      <gps>
85        <latitude> 0 </latitude>
86        <longitude> 0 </longitude>
87        <altitude> 0 </altitude>
88      </gps>
89      <url_media>pic.com/TfXd2D3</url_media>
90    </tweet>
91
92    <tweet id="t4" author="u3">
93      <timestamp GMT="2"> 33 </timestamp>
94      <message length="140" type="text" color="black" lang="en">
95        <ref_user> example </ref_user>
96        <text> absolutely smashed it at </text>
97        <text> ! Catch him at the official </text>
98        <ref_user> clubmtvuk </ref_user>
99        <text> after party tonight @10 pm </text>
100     </message>
101     <gps>
102       <latitude> 0 </latitude>
103       <longitude> 0 </longitude>
104       <altitude> 0 </altitude>
105     </gps>
106     <url_media>pic.com/TfXd2D3</url_media>
107   </tweet>
108
109   <tweet id="t5" author="u3">
110     <timestamp GMT="2"> 34</timestamp>
```

```
111      <message length="140" type="text" color="black" lang="en">
112        <ref_user> example </ref_user>
113        <text> absolutely smashed it at </text>
114        <text> ! Catch him at the official </text>
115        <ref_user> clubmtvuk </ref_user>
116        <text> after party tonight @10 pm </text>
117      </message>
118      <gps>
119        <latitude> 0 </latitude>
120        <longitude> 0 </longitude>
121        <altitude> 0 </altitude>
122      </gps>
123      <url_media>pic.com/TfXd2D3</url_media>
124    </tweet>
125
126    <user id="u1" OS="Windows">
127      <name> MTV Music </name>
128      <city> Montpellier </city>
129      <country> France </country>
130      <url_avatar> null </url_avatar>
131      <follow_count> 24 </follow_count>
132      <sub_count> 94728472 </sub_count>
133      <description> MTV Music UK </description>
134    </user>
135
136    <user id="u2" OS="Windows">
137      <name> NRJ Music </name>
138      <city> Paris </city>
139      <country> France </country>
140      <url_avatar> null </url_avatar>
141      <follow_count> 29 </follow_count>
142      <sub_count> 1597383 </sub_count>
143      <description> MTV Music UK </description>
144    </user>
145
146    <user id="u3" OS="Windows">
147      <name> W9 Hits </name>
148      <city> Berlin </city>
149      <country> Germany </country>
150      <url_avatar> null </url_avatar>
151      <follow_count> 41 </follow_count>
152      <sub_count> 12042 </sub_count>
153      <description> MTV Music UK </description>
154    </user>
155
156  </tweets>
157    ') );
158
159  SELECT EXTRACT(tweetXml,'//tweet/@id') FROM tweet_binaryxml;
160  SELECT EXTRACT(tweetXml,'//tweet/message') FROM tweet_binaryxml;
161  SELECT EXTRACT(tweetXml,'//user/@id') FROM tweet_binaryxml;
162
163  tweet_binaryxml
164  SELECT XMLQUERY('for $x in //user
165    for $y in //tweet
166      where $y/@author=$x/@id
167        return
```

```
168         <result>
169           <pair
170             tweetID ="{$y/@id}"
171             author="{$x/name/text()}"
172             />
173         </result>'
174       PASSING tweetXml RETURNING CONTENT)
175 FROM tweet_binaryxml;
176
177 SELECT XMLQUERY('
178   for $x in //tweet
179       return
180       if  ($x/responses) then
181       <retweet idtweet="{$x/@id}"><content>{$x/message}</content>
182         {$x/responses/response/date}
183         </retweet>
184         else
185         <nonRetwitted idtweet="{$x/@id}"/> '
186       PASSING tweetXml RETURNING CONTENT)
187 FROM tweet_binaryxml;
188
189 SELECT XMLQUERY('
190 for $x in //user
191 for $y in //tweet
192       where $y/@author=$x/@id
193         return
194         <result>
195         <name>{$x/name/text()}</name>
196         <dates>{$y/timestamp/text()}</dates>
197         </result>'
198       PASSING tweetXml RETURNING CONTENT)
199 FROM tweet_binaryxml;
```

WarmUp.sql

2 Stockage Monet

```
1  DROP TABLE presse;
2  DROP TABLE presse_journal;
3  DROP TABLE presse_journal_nom;
4  DROP TABLE presse_journal_directeur;
5  DROP TABLE presse_journal_directeur_nom;
6  DROP TABLE presse_journal_directeur_prenom;
7  DROP TABLE presse_journal_article;
8  DROP TABLE presse_journal_article_corps;
9  DROP TABLE presse_journalistes;
10 DROP TABLE presse_journalistes_journaliste;
11 DROP TABLE presse_journalistes_journaliste_anonymisation;
12
13
14 CREATE TABLE presse (
15   node     VARCHAR(5),
16   txtval   VARCHAR(5),
17   numval   NUMERIC(5),
18   CONSTRAINT PK_PRESSE PRIMARY KEY(node)
```

```
19 );
20
21 CREATE TABLE presse_journal(
22   node  VARCHAR(5),
23   txtval  VARCHAR(5),
24   numval  NUMERIC(5)
25   CONSTRAINT PK_PRESSE_JOURNAL PRIMARY KEY(node)
26 );
27
28 CREATE TABLE presse_journal_nom(
29   node  VARCHAR(5),
30   txtval  VARCHAR(5),
31   numval  NUMERIC(5)
32   CONSTRAINT PK_PRESSE_JOURNAL_NOM PRIMARY KEY(node)
33 );
34
35 CREATE TABLE presse_journal_directeur(
36   node  VARCHAR(5),
37   txtval  VARCHAR(5),
38   numval  NUMERIC(5)
39   CONSTRAINT PK_PRESSE_JOURNAL_DIRECTEUR PRIMARY KEY(node)
40 );
41
42
43 CREATE TABLE presse_journal_directeur_nom(
44   node  VARCHAR(5),
45   txtval  VARCHAR(5),
46   numval  NUMERIC(5)
47   CONSTRAINT PK_PRESSE_JOURNAL_DIRECTEUR_NOM PRIMARY KEY(node)
48 );
49
50 CREATE TABLE presse_journal_directeur_prenom(
51   node  VARCHAR(5),
52   txtval  VARCHAR(5),
53   numval  NUMERIC(5)
54   CONSTRAINT PK_PRESSE_JOURNAL_DIRECTEUR_PRENOM PRIMARY KEY(node)
55 );
56
57 CREATE TABLE presse_journal_article(
58   node  VARCHAR(5),
59   txtval  VARCHAR(5),
60   numval  NUMERIC(5),
61   CONSTRAINT PK_PRESSE_JOURNAL_ARTICLE PRIMARY KEY(node)
62 );
63
64 CREATE TABLE presse_journal_article_auteur(
65   node  VARCHAR(5),
66   txtval  VARCHAR(5),
67   numval  NUMERIC(5),
68   CONSTRAINT PK_PRESSE_JOURNAL_ARTICLE_AUTEUR PRIMARY KEY(node)
69 );
70
71 CREATE TABLE presse_journal_article_titre(
72   node  VARCHAR(5),
73   txtval  VARCHAR(5),
74   numval  NUMERIC(5),
75   CONSTRAINT PK_PRESSE_JOURNAL_ARTICLE_TITRE PRIMARY KEY(node)
```

```
76 );
77
78 CREATE TABLE presse_journal_article_corps(
79   node  VARCHAR(5),
80   txtval  VARCHAR(5),
81   numval  NUMERIC(5)
82   CONSTRAINT PK_PRESSE_JOURNAL_ARTICLE_CORPS PRIMARY KEY(node)
83 );
84
85 CREATE TABLE presse_journalistes(
86   node  VARCHAR(5),
87   txtval  VARCHAR(5),
88   numval  NUMERIC(5)
89   CONSTRAINT PK_PRESSE_JOURNALISTES PRIMARY KEY(node)
90 );
91
92 CREATE TABLE presse_journalistes_journaliste(
93   node  VARCHAR(5),
94   txtval  VARCHAR(5),
95   numval  NUMERIC(5),
96   CONSTRAINT PK_PRESSE_JOURNALISTES_JOURNALISTE PRIMARY KEY(NODE)
97 );
98
99 CREATE TABLE presse_journalistes_journaliste_idj(
100   node  VARCHAR(5),
101   txtval  VARCHAR(5),
102   numval  NUMERIC(5),
103   CONSTRAINT PK_PRESSE_JOURNALISTES_JOURNALISTE_IDJ PRIMARY KEY(node)
104 );
105
106 CREATE TABLE presse_journalistes_journaliste_pseudonyme(
107   node  VARCHAR(5),
108   txtval  VARCHAR(5),
109   numval  NUMERIC(5),
110   CONSTRAINT PK_PRESSE_JOURNALISTES_JOURNALISTE_PSEUDONYME PRIMARY KEY(node
      )
111 );
112
113 CREATE TABLE presse_journalistes_journaliste_anonymisation(
114   node  VARCHAR(5),
115   txtval  VARCHAR(5),
116   numval  NUMERIC(5)
117   CONSTRAINT PK_PRESSE_JOURNALISTES_JOURNALISTE_ANONYMISATION PRIMARY KEY(
      node)
118 );
119
120 INSERT INTO presse  (node,txtval,numval)
121 VALUES (n1,NULL,NULL);
122
123 INSERT INTO presse_journal  (node,txtval,numval)
124 VALUES (n2,NULL,NULL);
125
126 INSERT INTO presse_journal_nom  (node,txtval,numval)
127 VALUES (n3,NULL,NULL);
128
129 INSERT INTO presse_journal_directeur (node,txtval,numval)
130 VALUES (n4,NULL,NULL);
```

```
131
132 INSERT INTO presse_journal_directeur_nom (node,txtval,numval)
133 VALUES (n5,"Ulach",NULL);
134
135 INSERT INTO presse_journal_directeur_prenom (node,txtval,numval)
136 VALUES (n6,"Fred",NULL);
137
138 INSERT INTO presse_journal_article  (node,txtval,numval)
139 VALUES (n8,NULL,NULL);
140
141
142 INSERT INTO presse_journal_article_auteur (node,txtval,numval)
143 VALUES (n9,"Emile",NULL);
144
145 INSERT INTO presse_journal_article_corps (node,txtval,numval)
146 VALUES (n10,"faitDivers",NULL);
147
148 INSERT INTO presse_journal_article_titre (node,txtval,numval)
149 VALUES (n11,"viol",NULL);
150
151 INSERT INTO presse_journalistes (node,txtval,numval)
152 VALUES (n12,NULL,NULL);
153
154 INSERT INTO presse_journalistes_journaliste (node,txtval,numval)
155 VALUES (n13,NULL,NULL);
156
157 INSERT INTO presse_journalistes_journaliste_idj (node,txtval,numval)
158 VALUES (n14,NULL,164);
159
160 INSERT INTO presse_journalistes_journaliste_pseudonyme  (node,txtval,numval
    ) VALUES (n15,"MCFred",NULL);
161
162 INSERT INTO presse_journalistes_journaliste_anonymisation (node,txtval,
    numval) VALUES (n16,"non",NULL);
163
164 --Xpath to SQL queries--
165
166 --/presse/journal/directeur/nom/text()
167 SELECT txtval FROM presse_journal_directeur_nom;
168
169 --/presse/journal/directeur/prenom/text()
170 SELECT txtval FROM presse_journal_directeur_prenom;
171
172 --/presse/journalistes/journaliste/@id
173 SELECT numval FROM presse_journalistes_journaliste_idj;
```

Monet.sql

3 Stockage schema-aware

3.1 Derivation du schéma relationel du DTD

3.1.1 Simplification des expressions régulières

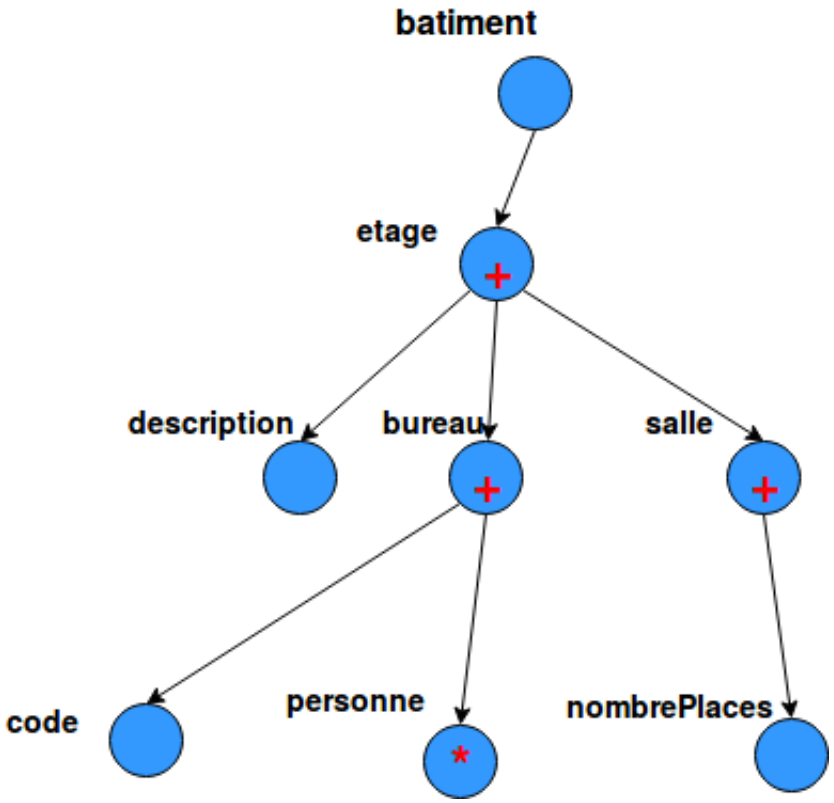
DTD initial :

```
<!DOCTYPE batiment [
 <!ELEMENT batiment (etage)+>
<!ELEMENT etage (description,(bureau+|salle+))>
<!ELEMENT description (#PCDATA)>
<!ELEMENT bureau (code, personne*)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT personne (#PCDATA)>
<!ELEMENT salle (nombrePlaces)>
<!ELEMENT nombrePlaces (#PCDATA)>]>
```

DTD transformé :

```
<!DOCTYPE batiment [
 <!ELEMENT batiment etage+ >
<!ELEMENT etage (description|(bureau+|salle+))>
<!ELEMENT description (#PCDATA)>
<!ELEMENT bureau (code|(personne*))>
<!ELEMENT salle (nombrePlaces)>]>
```

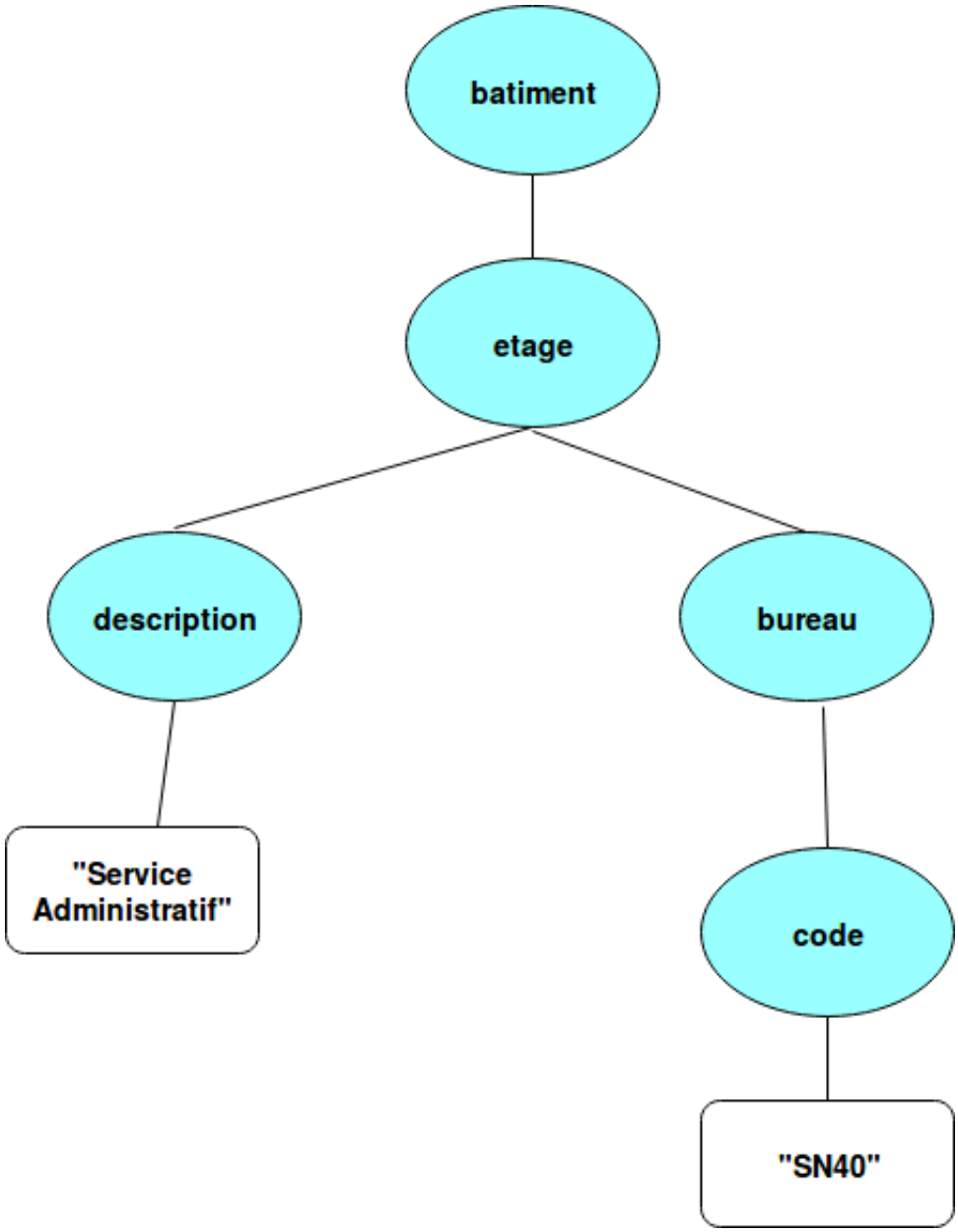
3.1.2 Graphe représantant le DTD obtenu



3.1.3 Creation des relations

batiment(**batimentID**)
etage(**etageID**,**batimentID**,description : string)
bureau(**bureauID**,**etageID**,code : string)
salle(**salleID**,**etageID**,nombrePlaces : string)
personne(**personneID**,**bureauID**,prs : string)

3.2 Remplissage des tables



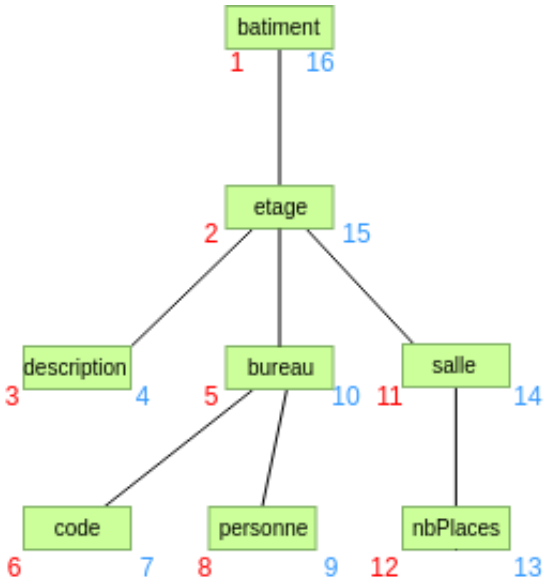
```sql
INSERT INTO batiment (batimentID) VALUES (1);
INSERT INTO etage (etageID,batimentID,description) VALUES (2,1,"Service
    Administratif");
INSERT INTO bureau(bureauID,etageID,code) VALUES (5,2,"SN40");
```

3.3 Traduction de requêtes en SQL

```sql
--/batiment/etage/description/text()
SELECT descrition FROM etage;
--/batiment/etage*
SELECT * FROM batiment,etage WHERE batiment.batimentID=etage.batimentID;
--/batiment/etage/bureau/code/text()
SELECT code FROM bureau ;
```

4 Interval-encoding avec SAX

4.1 Encodage begin/end



begin	end	par	tag	type
1	16		batiment	ELT
2	15	1	etage	ELT
3	4	2	description	ELT
5	10	2	bureau	ELT
6	7	5	code	ELT
8	9	5	personne	ELT
11	14	2	salle	ELT
12	13	11	nbPlaces	ELT