Donnes du Web - TD3 - XML-Relationnel

Basil Dali - Yannis Naidja

Octobre 2019

Chapitre 1

Stockage schema-unaware : Verical-Edge vs Monet

1.1 Considrons le document XML pour la presse que vous avez propos en rponse la question1du TD1

1.2 laide du langage SQL, implmenter les schmas de stockage Vertical-Edge et Monet associs au docu-ment. Ensuite, peupler les tables correspondantes

Voir document XML presse en annexe.

1.2.1 Vertical-Edge

```
CREATE TABLE PRESSE (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE JOURNAL (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
```

```
        txtval  varchar(30),
        numval int
);

CREATE TABLE ARTICLE (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);
CREATE TABLE TITRE (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE AUTEUR (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE CORPS (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);
CREATE TABLE JOURNALISTES (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);
CREATE TABLE JOURNALISTE (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
```

```
        numval int
);
CREATE TABLE JOURNALISTEID (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE ANONYMOUS (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);
CREATE TABLE PSEUDO (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE NOM (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRENOM (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE DIRECTEUR (
        source int,
        target int NOT NULL PRIMARY KEY,
        ordinal int,
        txtval  varchar(30),
```

```
105         numval int
106 );
107
108 INSERT INTO PRESSE (target)                              VALUES
    ↪  (0);
109 INSERT INTO JOURNAL (source, target, ordinal)           VALUES (0,
    ↪  1, 1);
110 INSERT INTO NOM (source, target, ordinal, txtval)       VALUES (1,
    ↪  2, 1, 'CNEWS');
111 INSERT INTO DIRECTEUR (source, target, ordinal)         VALUES (1,
    ↪  3, 1);
112 INSERT INTO NOM (source, target, ordinal, txtval)       VALUES (3,
    ↪  4, 1, 'Pepega');
113 INSERT INTO PRENOM (source, target, ordinal, txtval)    VALUES (3,
    ↪  5, 2, 'Kekw');
114 INSERT INTO ARTICLE (source, target, ordinal)           VALUES (1,
    ↪  7, 1);
115 INSERT INTO CORPS (source, target, ordinal, txtval)     VALUES (7,
    ↪  10, 1, 'Des fake news');
116 INSERT INTO TITRE (source, target, txtval)             VALUES (7,
    ↪  8, 'fake');
117 INSERT INTO AUTEUR(source, target, txtval)             VALUES (7,
    ↪  9, 'j1');
118 INSERT INTO ARTICLE (source, target, ordinal)          VALUES (1,
    ↪  11, 1);
119 INSERT INTO CORPS (source, target, ordinal, txtval)    VALUES
    ↪  (11, 14, 1,' Encore des fake news');
120 INSERT INTO TITRE (source, target, txtval)             VALUES
    ↪  (11, 12, 'news');
121 INSERT INTO AUTEUR(source, target, txtval)             VALUES
    ↪  (11, 13, 'j1');
122 INSERT INTO JOURNALISTES (source, target, ordinal)     VALUES (0,
    ↪  15, 2);
123 INSERT INTO JOURNALISTE (source, target, ordinal)      VALUES
    ↪  (15, 16, 1);
124 INSERT INTO JOURNALISTEID (source, target, txtval)     VALUES
    ↪  (16, 19, 'j1');
125 INSERT INTO NOM (source, target, ordinal, txtval)      VALUES
    ↪  (16, 17, 1, 'Vuillard');
126 INSERT INTO PRENOM (source, target, ordinal, txtval)   VALUES
    ↪  (16, 18, 1, 'Eric');
127 INSERT INTO JOURNALISTE (source, target, ordinal)      VALUES
    ↪  (15, 20, 2);
128 INSERT INTO JOURNALISTEID (source, target, txtval)     VALUES
    ↪  (20, 23, 'j2');
```

```
129 INSERT INTO NOM (source, target, ordinal, txtval)      VALUES
    ↪  (20, 21, 1, 'Dupont');
130 INSERT INTO PRENOM (source, target, ordinal, txtval)   VALUES
    ↪  (20, 22, 1, 'Jean');
131 INSERT INTO ANONYMOUS(source, target, ordinal, txtval) VALUES
    ↪  (20, 24, 1, 'non');
```

1.2.2 Monet

```
1  CREATE TABLE PRESSE (
2          node int,
3          txtval  varchar(30),
4          numval int
5  );
6  CREATE TABLE PRESSE_JOURNAL (
7          node int,
8          txtval  varchar(30),
9          numval int
10 );
11
12 CREATE TABLE PRESSE_JOURNAL_NOM (
13         node int,
14         txtval  varchar(30),
15         numval int
16 );
17 CREATE TABLE PRESSE_JOURNAL_DIRECTEUR (
18         node int,
19         txtval  varchar(30),
20         numval int
21 );
22
23 CREATE TABLE PRESSE_JOURNAL_DIRECTEUR_NOM (
24         node int,
25         txtval  varchar(30),
26         numval int
27 );
28
29 CREATE TABLE PRESSE_JOURNAL_DIRECTEUR_PRENOM (
30         node int,
31         txtval  varchar(30),
32         numval int
33 );
34
35 CREATE TABLE PRESSE_JOURNAL_ARTICLE  (
36         node int,
37         txtval  varchar(30),
```

```
        numval int
);

CREATE TABLE PRESSE_JOURNAL_ARTICLE_CORPS  (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNAL_ARTICLE_TITRE  (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNAL_ARTICLE_AUTEUR  (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNALISTES (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNALISTES_JOURNALISTE (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNALISTES_JOURNALISTE_NOM (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNALISTES_JOURNALISTE_PRENOM (
        node int,
        txtval  varchar(30),
        numval int
);

CREATE TABLE PRESSE_JOURNALISTES_JOURNALISTE_IDJ (
```

```
84          node int,
85          txtval  varchar(30),
86          numval int
87  );
88
89  CREATE TABLE PRESSE_JOURNALISTES_JOURNALISTE_ANONYMISATION (
90          node int,
91          txtval  varchar(30),
92          numval int
93  );
94
95  INSERT INTO PRESSE(node)
    ↪  VALUES(0);
96  INSERT INTO PRESSE_JOURNAL(node)
    ↪  VALUES(1);
97  INSERT INTO PRESSE_JOURNAL_NOM(node,txtval)
    ↪  VALUES(2,'CNEWS');
98  INSERT INTO PRESSE_JOURNAL_DIRECTEUR(node)
    ↪  VALUES(3);
99  INSERT INTO PRESSE_JOURNAL_DIRECTEUR_NOM(node,txtval)
    ↪  VALUES(4,'Pepega');
100 INSERT INTO PRESSE_JOURNAL_DIRECTEUR_PRENOM(node,txtval)
    ↪  VALUES(5,'KEKW');
101 INSERT INTO PRESSE_JOURNAL_ARTICLE(node)
    ↪  VALUES(7);
102 INSERT INTO PRESSE_JOURNAL_ARTICLE_TITRE(node, txtval)
    ↪  VALUES(8, 'fake');
103 INSERT INTO PRESSE_JOURNAL_ARTICLE_AUTEUR(node, txtval)
    ↪  VALUES(9, 'j1');
104 INSERT INTO PRESSE_JOURNAL_ARTICLE_CORPS(node,txtval)
    ↪  VALUES (10,'fakenews');
105 INSERT INTO PRESSE_JOURNAL_ARTICLE(node)
    ↪  VALUES(11);
106 INSERT INTO PRESSE_JOURNAL_ARTICLE_TITRE(node, txtval)
    ↪  VALUES(12, 'news');
107 INSERT INTO PRESSE_JOURNAL_ARTICLE_AUTEUR(node, txtval)
    ↪  VALUES(13, 'j1');
108 INSERT INTO PRESSE_JOURNAL_ARTICLE_CORPS(node,txtval)
    ↪  VALUES (14,' more fakenews');
109 INSERT INTO PRESSE_JOURNALISTES(node)
    ↪  VALUES (15);
110 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE(node)
    ↪  VALUES (16);
111 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_NOM(node,txtval)
    ↪  VALUES (17,'Vuillard');
```

```
112 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_PRENOM(node,txtval)
    ↪  VALUES (18,'Eric');
113 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_IDJ(node,txtval)
    ↪  VALUES (19, 'j1');
114 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE(node)
    ↪  VALUES (20);
115 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_NOM(node,txtval)
    ↪  VALUES (21,'Dupont');
116 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_PRENOM(node,txtval)
    ↪  VALUES (22,'Jean');
117 INSERT INTO PRESSE_JOURNALISTES_JOURNALISTE_IDJ(node,txtval)
    ↪  VALUES (23, 'j2');
118 INSERT INTO
    ↪  PRESSE_JOURNALISTES_JOURNALISTE_ANONYMISATION(node,txtval)
    ↪  VALUES (24, 'oui');
```

1.3 laide du langage SQL, exprimer cinq requtes XPath de votre choix sur chaque schma

1.3.1 Le nom du journal

```
1 /presse/journal/nom
```

Vertical-Edge

```
1 SELECT N.txtval nom_journal
2 FROM PRESSE  P,
3      JOURNAL J,
4      NOM     N
5 WHERE
6        P.Target = J.Source
7 AND    J.Target = N.Source;
```

Monet

```
1 SELECT txtval nom_journal
2 FROM   PRESSE_JOURNAL_NOM;
```

1.3.2 Le nom du directeur

```
1 /presse/journal/directeur/nom
```

Vertical-Edge

```
1 SELECT N.txtval nom_directeur
2 FROM PRESSE    P,
```

```
      JOURNAL   J,
      DIRECTEUR D,
      NOM       N
WHERE
       P.Target = J.Source
AND    J.Target = D.Source
AND    D.Target = N.Source;
```

Monet

```
SELECT txtval nom_directeur
FROM PRESSE_JOURNAL_DIRECTEUR_NOM;
```

1.3.3 Le titre de tout les articles

```
/presse/journal/article/@titre
```

Vertical-Edge

```
SELECT T.txtval titre_article
FROM PRESSE  P,
     JOURNAL J,
     ARTICLE A,
     TITRE   T
WHERE P.Target = J.Source
AND   J.Target = A.Source
AND   A.Target = T.Source;
```

Monet

```
SELECT txtval titre_article
FROM PRESSE_JOURNAL_ARTICLE_TITRE;
```

1.3.4 Le nombre d article

```
/presse/journal/count(article)
```

Vertical-Edge

```
SELECT COUNT(A.Target) nombre_d_article
FROM PRESSE  P,
     JOURNAL J,
     ARTICLE A
WHERE P.Target = J.Source
AND   J.Target = A.Source;
```

Monet

```
SELECT COUNT(A.node) nombre_d_article
FROM PRESSE_JOURNAL_ARTICLE A;
```

1.3.5 Les identifiants des journalistes

```
/presse/journal/journalistes/journaliste/@idJ
```

Vertical-Edge

```
SELECT J3.txtval identifiant_de_journaliste
FROM PRESSE        P,
     JOURNALISTES  J1,
     JOURNALISTE   J2,
     JOURNALISTEID J3
WHERE P.Target = J1.Source
AND   J1.Target = J2.Source
AND   J2.Target = J3.Source;
```

Monet

```
SELECT txtval identifiant_de_journaliste
FROM PRESSE_JOURNALISTES_JOURNALISTE_IDJ;
```

Chapitre 2

Stockage schema-aware : Verical-Edge vs Monet

2.1 A partir de la DTD pour les batiments presentee dans lenonce du TD1, definir un schema de stockagerelationnel suivant la methode presentee en cours.

2.1.1 DTD batiment

```
<!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) >]>
```

2.1.2 Suppression des symboles +

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

```
<!ELEMENT nombrePlaces (#PCDATA) >]>
```

2.1.3 Suppression de l'ordre et des correlations

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

2.1.4 Simplifications

r |r* est equivalent a r*

```
<!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) >]>
```

2.1.5 Representation sous forme de graphe

Voir graphe en annexe

2.1.6 Relations

- batiment(batimentID : integer, flagRoot : integer)
- etage(etageID : integer, batimentID : integer, description : string)
- bureau(bureauID : integer, etageID : integer, code : string)
- personne(personneID : integer, bureauID : integer)
- salle(salleID : integer, etageID : integer, nombreDePlace : integer)

2.1.7 Creation des tables

```
CREATE TABLE BATIMENT (
       batimentID int NOT NULL PRIMARY KEY,
       flagRoot int NOT NULL
);
```

```
5
6  CREATE TABLE ETAGE (
7          etageID int NOT NULL PRIMARY KEY,
8          batimentID int NOT NULL,
9          description varchar(30),
10         CONSTRAINT fk_batimentID
11         FOREIGN KEY (batimentID)
12         REFERENCES BATIMENT (batimentID)
13 );
14
15 CREATE TABLE BUREAU (
16         bureauID int NOT NULL PRIMARY KEY,
17         etageID int NOT NULL,
18         code varchar(10),
19         CONSTRAINT fk_bureau_etageID
20         FOREIGN KEY (etageID)
21         REFERENCES ETAGE (etageID)
22 );
23
24 CREATE TABLE PERSONNE (
25         personneID int NOT NULL PRIMARY KEY,
26         bureauID int NOT NULL,
27         CONSTRAINT fk_bureauID
28         FOREIGN KEY (bureauID)
29         REFERENCES BUREAU (bureauID)
30 );
31
32 CREATE TABLE SALLE (
33         salleID int NOT NULL PRIMARY KEY,
34         etageID int NOT NULL,
35         nombreDePlace int,
36         CONSTRAINT fk_salle_etageID
37         FOREIGN KEY (etageID)
38         REFERENCES ETAGE (etageID)
39 );
```

2.2 Peupler les tables avec des lignes correspondants au document XML que vous propose en reponse a laquestion 1 du TD1

```
1 INSERT INTO BATIMENT (batimentID, flagRoot)          VALUES (0,
  ↪  1);
2 INSERT INTO ETAGE (etageID, batimentID, description) VALUES (0,
  ↪  0, 'descriptionETAGE1');
```

```
INSERT INTO ETAGE (etageID, batimentID, description) VALUES (1,
↪  0, 'descriptionETAGE2');
INSERT INTO SALLE (salleID, etageID, nombreDePlace)  VALUES (0,
↪  0, 25);
INSERT INTO SALLE (salleID, etageID, nombreDePlace)  VALUES (1,
↪  0, 50);
INSERT INTO BUREAU (bureauID, etageID, code)         VALUES (0,
↪  1, 'B02');
INSERT INTO PERSONNE (personneID, bureauID)          VALUES (0,
↪  0);
INSERT INTO BUREAU (bureauID, etageID, code)         VALUES (1,
↪  1, 'B03');
INSERT INTO PERSONNE (personneID, bureauID)          VALUES (1,
↪  1);
INSERT INTO PERSONNE (personneID, bureauID)          VALUES (2,
↪  1);
```

2.3 A laide du langage SQL, exprimer cinq requtes XPath de votre choix

2.3.1 Les descriptions d'etage du batiment d'identifiant 0

```
//batiment[@id=0]/description/text()
```

Requte SQL

```
SELECT DESCRIPTION
FROM ETAGE
WHERE batimentID = 0;
```

2.3.2 Le nombre de place des salles de l etage d identifiant 0 du batiment d identifiant 0

```
//batiment[@id=0]/etage[@id=0]/salle/nombredeplace
```

Requte SQL

```
SELECT S.NOMBREDEPLACE
FROM SALLE S JOIN ETAGE E ON S.etageID = E.etageID
WHERE E.batimentID = 0 AND E.etageID = 0;
```

2.3.3 Le nombre de personne travaillant dans des bureaux

```
count(//batiment/etage/bureau/personne)
```

Requte SQL

```
SELECT count(*)
FROM personne;
```

2.3.4 Le code des bureaux de l etage 1 du batiment 0

```
//batiment[@id=0]/etage[@id=1]/bureau/@code
```

Requte SQL

```
SELECT B.code
FROM BUREAU B JOIN ETAGE E ON B.etageID = E.etageID
WHERE E.batimentID = 0 AND E.etageID = 1;
```

2.3.5 Le nombre d'etage du batiment 0

```
count(//batiment[@id=0]/etage)
```

Requte SQL

```
SELECT COUNT(*)
FROM ETAGE
WHERE batimentID = 0;
```

Chapitre 3

Oracle-XML

3.1 Crer des tables pour enregistrer vos documents XML en utilisant le diffrentes options de stockage.

3.1.1 CLOB

```
1 CREATE TABLE tweet_CLOB (
2        text_content varchar(20),
3        xml_content XMLTYPE)
4        XMLTYPE xml_content STORE AS CLOB;
```

3.1.2 BINARY

```
1 CREATE TABLE tweet_BINARY (
2        text_content varchar(20),
3        xml_content XMLTYPE)
4        XMLTYPE xml_content STORE AS BINARY XML;
```

3.2 Enregistrez votre document XML pour les Tweets dans la base avec linstruction suivante (attention auxcaractres ” ” et ” ” dans le copier-coller.)

3.2.1 CLOB

```
INSERT INTO
↪  tweet_CLOB(text_content,xml_content)VALUES('tweet.xml',
↪  sys.xmltype.createxml('
<tweeter>
  <users>
    <user id="U41">
      <user_name>
        basil_dalie
      </user_name>
      <first_name>
        Basil
      </first_name>
      <last_name>
        Dalie
      </last_name>
      <profile>
        https://twitter.com/basil_dalie
      </profile>
    </user>
    <user id="U02">
      <user_name>
        Cicero
      </user_name>
      <first_name>
        Ci
      </first_name>
      <last_name>
        Cero
      </last_name>
      <profile>
        https://twitter.com/Cicero
      </profile>
    </user>
    <user id="U43">
      <user_name>
        alex_not
      </user_name>
```

```
      <first_name>
        Alex
      </first_name>
      <last_name>
        Not
      </last_name>
      <profile>
        https://twitter.com/alex_n
      </profile>
    </user>
  </users>
  <tweets>
    <tweet id="T42" author_ref="U41">
      <header>
        <date>
          1758312000
        </date>
        <timezone>
          <standard>
            UTC
          </standard>
          <offset>
            1
          </offset>
        </timezone>
        <location>
          <latitude>
            3.876716
          </latitude>
          <longitude>
            43.610769
          </longitude>
          <city>
            Montpellier
          </city>
          <country>
            France
          </country>
        </location>
        <language>
          Latin
        </language>
        <retweets>
          <retweet ref="T04" />
          <retweet ref="T05" />
          <retweet ref="T03" />
```

```
        </retweets>
        <answers>
          <answer ref="T02" />
        </answers>
        <operating_system>
          Linux x86_64
        </operating_system>
        <images>
          <image id="I01">

↪ https://images2.minutemediacdn.com/image/upload/c_crop,h_1193,w_2121,x_0,y_64/f_auto,q_a
          </image>
          <image id ="I02">
            https://website.com/image.jpg
          </image>
        </images>
        <videos>
          <video id="V01">
            https://website.com/video.avi
          </video>
        </videos>
      </header>
      <body>
        <hashtag>
          #I3XML
        </hashtag>
        <text>
          Lorem ipsum dolor sit amet, consectetur adipiscing
↪ elit, sed do eiusmod tempor incididunt ut labore et dolore
↪ magna aliqua. Ut enim ad minim veniam, quis nostrud
↪ exercitation ullamco laboris nisi ut aliquip ex ea commodo
↪ consequat. Duis aute irure dolor in reprehenderit in
↪ voluptate velit esse cillum dolore eu fugiat nulla pariatur.
↪ Excepteur sint occaecat cupidatat non proident, sunt in culpa
↪ qui officia deserunt
        </text>
        <text>
          anim id est laborum
        </text>
        
        <user_ref ref="U02">
          \@Cicero
        </user_ref>
        <text>
```

```
118           Sed ut perspiciatis unde omnis iste natus error sit
    ↪  voluptatem accusantium doloremque laudantium, totam rem
    ↪  aperiam, eaque ipsa quae ab illo inventore veritatis et quasi
    ↪  architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam
    ↪  voluptatem quia voluptas sit aspernatur aut odit aut fugit,
    ↪  sed quia consequuntur magni dolores eos qui ratione
    ↪  voluptatem sequi nesciunt. Neque porro quisquam est, qui
    ↪  dolorem ipsum quia dolor sit amet, consectetur, adipisci
    ↪  velit, sed quia non numquam eius modi tempora incidunt ut
    ↪  labore et dolore magnam aliquam quaerat voluptatem. Ut enim
    ↪  ad minima veniam, quis nostrum exercitationem ullam corporis
    ↪  suscipit laboriosam, nisi ut aliquid ex ea commodi
    ↪  consequatur? Quis autem vel eum iure reprehenderit qui in ea
    ↪  voluptate velit esse quam nihil molestiae consequatur, vel
    ↪  illum qui dolorem eum fugiat quo voluptas nulla pariatur?
119         </text>
120         <video_ref ref="V01" />
121       </body>
122     </tweet>
123   </tweets>
124 </tweeter>'));
```

3.2.2 BINARY

```
1   INSERT INTO
    ↪  tweet_BINARY(text_content,xml_content)VALUES('tweet.xml',
    ↪  sys.xmltype.createxml('
2   <tweeter>
3     <users>
4       <user id="U41">
5         <user_name>
6           basil_dalie
7         </user_name>
8         <first_name>
9           Basil
10        </first_name>
11        <last_name>
12          Dalie
13        </last_name>
14        <profile>
15          https://twitter.com/basil_dalie
16        </profile>
17      </user>
18      <user id="U02">
19        <user_name>
20          Cicero
```

```
      </user_name>
      <first_name>
        Ci
      </first_name>
      <last_name>
        Cero
      </last_name>
      <profile>
        https://twitter.com/Cicero
      </profile>
    </user>
    <user id="U43">
      <user_name>
        alex_not
      </user_name>
      <first_name>
        Alex
      </first_name>
      <last_name>
        Not
      </last_name>
      <profile>
        https://twitter.com/alex_n
      </profile>
    </user>
  </users>
  <tweets>
    <tweet id="T42" author_ref="U41">
      <header>
        <date>
          1758312000
        </date>
        <timezone>
          <standard>
            UTC
          </standard>
          <offset>
            1
          </offset>
        </timezone>
        <location>
          <latitude>
            3.876716
          </latitude>
          <longitude>
            43.610769
```

```
            </longitude>
            <city>
              Montpellier
            </city>
            <country>
              France
            </country>
          </location>
          <language>
            Latin
          </language>
          <retweets>
            <retweet ref="T04" />
            <retweet ref="T05" />
            <retweet ref="T03" />
          </retweets>
          <answers>
            <answer ref="T02" />
          </answers>
          <operating_system>
            Linux x86_64
          </operating_system>
          <images>
            <image id="I01">

↪ https://images2.minutemediacdn.com/image/upload/c_crop,h_1193,w_2121,x_0,y_64/f_auto,q_a
            </image>
            <image id ="I02">
              https://website.com/image.jpg
            </image>
          </images>
          <videos>
            <video id="V01">
              https://website.com/video.avi
            </video>
          </videos>
        </header>
        <body>
          <text>
```

```
          Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt
        </text>
        <text>
          anim id est laborum
        </text>
        
        <user_ref ref="U02">
          \@Cicero
        </user_ref>
        <text>
          Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?
        </text>
        <video_ref ref="V01" />
      </body>
    </tweet>
  </tweets>
</tweeter>'));
```

3.3 laide des commandes suivantes, excuter5requtes Xquery et5requtes XQuery parmi celles que-vous avez proposes pour le TP prcdent.

3.3.1 CLOB

Les noms des auteurs des tweets.

```
/tweeter/users/user[@id =
↪ /tweeter/tweets/tweet/@author_ref]/concat(first_name, ' ',
↪ last_name)
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/users/user[@id=/tweeter/tweets/tweet/@author_ref]/user_n
↪ AS nom FROM tweet_CLOB;
```

Les tweets de lutilisateur dont lid est "u41".

```
/tweeter/tweets/tweet[@author_ref='U41']
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet[@author_ref=''U41'']')
↪ AS tweet FROM tweet_CLOB;
```

Les tweets contenants lhashtag #I3XML"

```
/tweeter/tweets/tweet/body/hashtag[contains(text(),
↪ '#I3XML')]/ancestor::tweet
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet/body/hashtag[contains(text(),
↪ ''#I3XML'')]') FROM tweet_CLOB;
```

Les retweets du tweet dont lid est "t42".

```
/tweeter/tweets/tweet[@id = /tweeter/tweets/tweet[@id =
↪ "T42"]/header/retweets/retweet/@ref]
```

```
SELECT EXTRACT(xml_content, '/tweeter/tweets/tweet[@id =
↪ /tweeter/tweets/tweet[@id =
↪ "T42"]/header/retweets/retweet/@ref]') AS tweet FROM
↪ tweet_CLOB;
```

Les tweet sans hashtags.

```
/tweeter/tweets/tweet/body[count(hashtag) = 0]/parent::tweet
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet/body[count(hashtag)
↪ = 0]/parent::tweet') FROM tweet_CLOB;
```

3.3.2 BINARY

Les noms des auteurs des tweets.

```
/tweeter/users/user[@id =
↪ /tweeter/tweets/tweet/@author_ref]/concat(first_name, ' ',
↪ last_name)
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/users/user[@id=/tweeter/tweets/tweet/@author_ref]/user_n
↪ AS nom FROM tweet_BINARY;
```

Les tweets de lutilisateur dont lid est "u41".

```
/tweeter/tweets/tweet[@author_ref='U41']
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet[@author_ref=''U41'']')
↪ AS tweet FROM tweet_BINARY;
```

Les tweets contenants lhashtag #I3XML"

```
/tweeter/tweets/tweet/body/hashtag[contains(text(),
↪ '#I3XML')]/ancestor::tweet
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet/body/hashtag[contains(text(),
↪ ''#I3XML'')]') FROM tweet_BINARY;
```

Les retweets du tweet dont lid est "t42".

```
/tweeter/tweets/tweet[@id = /tweeter/tweets/tweet[@id =
↪ "T42"]/header/retweets/retweet/@ref]
```

```
SELECT EXTRACT(xml_content, '/tweeter/tweets/tweet[@id =
↪ /tweeter/tweets/tweet[@id =
↪ "T42"]/header/retweets/retweet/@ref]') AS tweet FROM
↪ tweet_BINARY;
```

Les tweet sans hashtags.

```
/tweeter/tweets/tweet/body[count(hashtag) = 0]/parent::tweet
```

```
SELECT
↪ EXTRACT(xml_content,'/tweeter/tweets/tweet/body[count(hashtag)
↪ = 0]/parent::tweet') FROM tweet_BINARY;
```

3.3.3 CLOB

Crez une liste de paires tweet-auteur, avec chaque paire contenue dans un element result.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
for  $tweet in /tweeter/tweets/tweet
where $auteur/@id = $tweet/@author_ref
return
<result>
{ $tweet }
{ $auteur }
</result>' PASSING xml_content RETURNING CONTENT)FROM
↪  tweet_CLOB;
```

Listez les utilisateurs de la plateforme en ordre alphabtique.

```
SELECT XMLQUERY('for $user in tweeter/users/user
order by upper-case($user/user_name/text()) ascending
return  $user' PASSING xml_content RETURNING CONTENT)FROM
↪  tweet_CLOB;
```

Pour chaque utilisateur, listez le nom de lutilisateur et la date de tous ses tweets, le tout regroup dansun lment result.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
return
<result>
  <nom>
    { $auteur/user_name/text()}
  </nom>
  {
    for $tweet in /tweeter/tweets/tweet
    where $auteur/@id = $tweet/@author_ref
    return $tweet/header/date
  }
</result>' PASSING xml_content RETURNING CONTENT)FROM
↪  tweet_CLOB;
```

Listez les utilisateurs qui ont publi un tweet qui a t retwitt au moins deux fois.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
  for $tweet in /tweeter/tweets/tweet
  where $auteur/@id = $tweet/@author_ref and
↪  count($tweet/header/retweets/retweet) > 1
  return $auteur' PASSING xml_content RETURNING CONTENT)FROM
↪  tweet_CLOB;
```

Pour chaque tweet, listez son contenu et la date de ses deux premires rponses. Rajoutez un element vide <nonRetwitted/>sil na pas t retwitt.

```
SELECT XMLQUERY('for $tweet in /tweeter/tweets/tweet
return
<tweet>
  <contenu>
    { $tweet/body/text}
  </contenu>
  <premieres-reponses>
    {
/tweeter/tweets/tweet[@id=$tweet/header/answers/answer[1]/@ref]/header/date
}
    {
/tweeter/tweets/tweet[@id=$tweet/header/answers/answer[2]/@ref]/header/date
}
  </premieres-reponses>
  {
    if (count($tweet/header/retweets/retweet) = 0) then
      <nonRetwitted />
    else()
  }
</tweet>' PASSING xml_content RETURNING CONTENT)FROM
tweet_CLOB;
```

3.3.4 Binary

Crez une liste de paires tweet-auteur, avec chaque paire contenue dans un element result.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
for  $tweet in /tweeter/tweets/tweet
where $auteur/@id = $tweet/@author_ref
return
<result>
{ $tweet }
{ $auteur }
</result>' PASSING xml_content RETURNING CONTENT)FROM
tweet_BINARY;
```

Listez les utilisateurs de la plateforme en ordre alphabtique.

```
SELECT XMLQUERY('for $user in tweeter/users/user
order by upper-case($user/user_name/text()) ascending
return  $user' PASSING xml_content RETURNING CONTENT)FROM
tweet_BINARY;
```

Pour chaque utilisateur, listez le nom de lutilisateur et la date de tous ses tweets, le tout regroup dansun lment result.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
return
<result>
  <nom>
    { $auteur/user_name/text()}
  </nom>
  {
    for $tweet in /tweeter/tweets/tweet
    where $auteur/@id = $tweet/@author_ref
    return $tweet/header/date
  }
</result>' PASSING xml_content RETURNING CONTENT)FROM
↪ tweet_BINARY;
```

Listez les utilisateurs qui ont publi un tweet qui a t retwitt au moins deux fois.

```
SELECT XMLQUERY('for $auteur in /tweeter/users/user
  for $tweet in /tweeter/tweets/tweet
  where $auteur/@id = $tweet/@author_ref and
↪ count($tweet/header/retweets/retweet) > 1
  return $auteur' PASSING xml_content RETURNING CONTENT)FROM
↪ tweet_BINARY;
```

Pour chaque tweet, listez son contenu et la date de ses deux premires rponses. Rajoutez un element vide <nonRetwitted/>sil na pas t retwitt.

```
SELECT XMLQUERY('for $tweet in /tweeter/tweets/tweet
return
<tweet>
  <contenu>
    { $tweet/body/text}
  </contenu>
  <premieres-reponses>
    {
↪ /tweeter/tweets/tweet[@id=$tweet/header/answers/answer[1]/@ref]/header/date
↪ }
    {
↪ /tweeter/tweets/tweet[@id=$tweet/header/answers/answer[2]/@ref]/header/date
↪ }
  </premieres-reponses>
  {
    if (count($tweet/header/retweets/retweet) = 0) then
```

```
13          <nonRetwitted />
14        else()
15      }
16    </tweet>' PASSING xml_content RETURNING CONTENT)FROM
    ↪  tweet_BINARY;
```

2

Chapitre 4

Interval-encoding avec SAX

4.1 Illustrer lencodage (1) begin/end et (2) Dewey de lXML du document propos pour les btiments

4.1.1 BeginEnd

Voir arbre et graphe begin/end en annexe

4.1.2 Dewey

Voir arbre Dewey en annexe

4.2 laide du langage SQL, crer le schma de stockage pour les intervalles et ensuite peupler la tableNODE avec les lignes correspondantes au document propos pour les btiments

```
1 CREATE TABLE NODE (
2         begin int NOT NULL,
3         end int NOT NULL,
4         par int,
5         tag varchar(20),
6         type varchar(20) NOT NULL
7 );
8
9 INSERT INTO NODE (begin, end, tag, type) VALUES (1, 50,
  ↪  'batiment', 'ELT');
```

```
10 INSERT INTO NODE                          VALUES (2, 19, 1,
   ↪  'etage', 'ELT');
11 INSERT INTO NODE                          VALUES (3, 6, 2,
   ↪  'description', 'ELT');
12 INSERT INTO NODE (begin, end, par, type) VALUES (4, 5, 3,
   ↪  'TEXT');
13 INSERT INTO NODE                          VALUES (7, 12, 2,
   ↪  'salle', 'ELT');
14 INSERT INTO NODE                          VALUES (8, 11, 7,
   ↪  'nombrePlace', 'ELT');
15 INSERT INTO NODE (begin, end, par, type) VALUES (9, 10, 8,
   ↪  'TEXT');
16 INSERT INTO NODE                          VALUES (13, 18, 2,
   ↪  'salle', 'ELT');
17 INSERT INTO NODE                          VALUES (14, 17, 13,
   ↪  'nombrePlace', 'ELT');
18 INSERT INTO NODE (begin, end, par, type) VALUES (9, 10, 8,
   ↪  'TEXT');
19 INSERT INTO NODE                          VALUES (20, 49, 1,
   ↪  'etage', 'ELT');
20 INSERT INTO NODE                          VALUES (21, 24, 20,
   ↪  'description', 'ELT');
21 INSERT INTO NODE (begin, end, par, type) VALUES (22, 23, 21,
   ↪  'TEXT');
22 INSERT INTO NODE                          VALUES (25, 34, 20,
   ↪  'bureau', 'ELT');
23 INSERT INTO NODE                          VALUES (26, 29, 25,
   ↪  'code', 'ELT');
24 INSERT INTO NODE (begin, end, par, type) VALUES (27, 28, 26,
   ↪  'TEXT');
25 INSERT INTO NODE                          VALUES (30, 33, 25,
   ↪  'personne', 'ELT');
26 INSERT INTO NODE (begin, end, par, type) VALUES (31, 32, 30,
   ↪  'TEXT');
27 INSERT INTO NODE                          VALUES (35, 48, 20,
   ↪  'bureau', 'ELT');
28 INSERT INTO NODE                          VALUES (36, 39, 35,
   ↪  'code', 'ELT');
29 INSERT INTO NODE (begin, end, par, type) VALUES (37, 38, 36,
   ↪  'TEXT');
30 INSERT INTO NODE                          VALUES (40, 43, 35,
   ↪  'personne', 'ELT');
31 INSERT INTO NODE (begin, end, par, type) VALUES (41, 42, 40,
   ↪  'TEXT');
32 INSERT INTO NODE                          VALUES (44, 47, 35,
   ↪  'personne', 'ELT');
```

```
INSERT INTO NODE (begin, end, par, type) VALUES (45, 46, 44,
↪  'TEXT');
```

4.3 laide de la classeSaxParser(.java sur Moodle), programmer lencodage par intervalles begin/end

```
public class SaxParser extends DefaultHandler {
  /** Constants used for JAXP 1.2 */
  static final String JAXP_SCHEMA_LANGUAGE =
      "http://java.sun.com/xml/jaxp/properties/schemaLanguage";
  static final String W3C_XML_SCHEMA =
      "http://www.w3.org/2001/XMLSchema";
  static final String JAXP_SCHEMA_SOURCE =
      "http://java.sun.com/xml/jaxp/properties/schemaSource";

  private Deque<Integer> stackBegin;
  private Integer counter;

  // Parser calls this once at the beginning of a document
  public void startDocument() throws SAXException {
      stackBegin = new ArrayDeque<>();
      counter = 0;
  }

  // Parser calls this for each opening of an element in a
  ↪  document
  public void startElement(String namespaceURI, String localName,
                           String qName, Attributes atts)
      throws SAXException
  {
      stackBegin.push(++counter);
  }

  // Parser calls this for each end of an element in a document
  public void endElement(String namespaceURI, String localName,
                           String qName)
      throws SAXException
  {
      int begin = stackBegin.pop();
      int end = ++counter;
      Integer parent = stackBegin.peek();

      String parentStr;
```

```
37          if (parent == null) {
38              parentStr = "NULL";
39          } else {
40              parentStr = String.valueOf(parent);
41          }
42
43          String output = "INSERT INTO NODE (begin, end, parent, tag,
            ↪  nodetype) VALUES(";
44          output += begin + ", " + end + ", " + parentStr + ", '" +
            ↪  qName + "', 'element');";
45
46          System.out.println(output);
47      }
48
49      // Parser calls this once after parsing a document
50      public void endDocument() throws SAXException {
51      }
52
53      // Parser calls after parsing a text node
54      public void characters(char[] ch, int start, int length) throws
        ↪  SAXException
55      {
56          String str = new String(ch, start, length);
57          str = str.replace(" ", "").replace("\n", "");
58          // System.out.println(str.length());
59
60          if (str.length() != 0) {
61              int begin = ++counter;
62              int end = ++counter;
63              String output = "INSERT INTO NODE (begin, end, parent,
                ↪  tag, nodetype) VALUES(";
64              output += begin + ", " + end + ", " + stackBegin.peek()
                ↪  + ", NULL, 'text');";
65
66              System.out.println(output);
67          }
68      }
69  }
```

4.4 Testez le programme sur le document XML concernant les Tweets, et rformulez 5 requtes Xquery sur ce document

4.4.1 Schema de stockage

```
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(5, 6, 4, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(4, 7, 3, 'user_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(9, 10, 8, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(8, 11, 3, 'first_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(13, 14, 12, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(12, 15, 3, 'last_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(17, 18, 16, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(16, 19, 3, 'profile', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(3, 20, 2, 'user', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(23, 24, 22, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(22, 25, 21, 'user_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(27, 28, 26, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(26, 29, 21, 'first_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(31, 32, 30, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(30, 33, 21, 'last_name', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(35, 36, 34, NULL, 'text');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(34, 37, 21, 'profile', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(21, 38, 2, 'user', 'element');
INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(41, 42, 40, NULL, 'text');
```

```
20 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(40,
   ↪ 43, 39, 'user_name', 'element');
21 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(45,
   ↪ 46, 44, NULL, 'text');
22 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(44,
   ↪ 47, 39, 'first_name', 'element');
23 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(49,
   ↪ 50, 48, NULL, 'text');
24 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(48,
   ↪ 51, 39, 'last_name', 'element');
25 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(53,
   ↪ 54, 52, NULL, 'text');
26 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(52,
   ↪ 55, 39, 'profile', 'element');
27 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(39,
   ↪ 56, 2, 'user', 'element');
28 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(2,
   ↪ 57, 1, 'users', 'element');
29 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(61,
   ↪ 62, 60, NULL, 'text');
30 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(64,
   ↪ 65, 63, NULL, 'text');
31 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(63,
   ↪ 66, 60, 'date', 'element');
32 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(67,
   ↪ 68, 60, NULL, 'text');
33 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(70,
   ↪ 71, 69, NULL, 'text');
34 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(73,
   ↪ 74, 72, NULL, 'text');
35 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(72,
   ↪ 75, 69, 'standard', 'element');
36 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(76,
   ↪ 77, 69, NULL, 'text');
37 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(79,
   ↪ 80, 78, NULL, 'text');
38 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(78,
   ↪ 81, 69, 'offset', 'element');
39 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(82,
   ↪ 83, 69, NULL, 'text');
40 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(69,
   ↪ 84, 60, 'timezone', 'element');
41 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(85,
   ↪ 86, 60, NULL, 'text');
42 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(88,
   ↪ 89, 87, NULL, 'text');
```

```
43 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(91,
   ↪  92, 90, NULL, 'text');
44 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(90,
   ↪  93, 87, 'latitude', 'element');
45 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(94,
   ↪  95, 87, NULL, 'text');
46 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(97,
   ↪  98, 96, NULL, 'text');
47 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(96,
   ↪  99, 87, 'longitude', 'element');
48 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(100,
   ↪  101, 87, NULL, 'text');
49 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(103,
   ↪  104, 102, NULL, 'text');
50 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(102,
   ↪  105, 87, 'city', 'element');
51 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(106,
   ↪  107, 87, NULL, 'text');
52 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(109,
   ↪  110, 108, NULL, 'text');
53 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(108,
   ↪  111, 87, 'country', 'element');
54 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(112,
   ↪  113, 87, NULL, 'text');
55 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(87,
   ↪  114, 60, 'location', 'element');
56 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(115,
   ↪  116, 60, NULL, 'text');
57 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(118,
   ↪  119, 117, NULL, 'text');
58 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(117,
   ↪  120, 60, 'language', 'element');
59 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(121,
   ↪  122, 60, NULL, 'text');
60 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(124,
   ↪  125, 123, NULL, 'text');
61 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(126,
   ↪  127, 123, 'retweet', 'element');
62 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(128,
   ↪  129, 123, NULL, 'text');
63 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(130,
   ↪  131, 123, 'retweet', 'element');
64 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(132,
   ↪  133, 123, NULL, 'text');
65 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(134,
   ↪  135, 123, 'retweet', 'element');
```

```
66 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(136,
   ↪ 137, 123, NULL, 'text');
67 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(123,
   ↪ 138, 60, 'retweets', 'element');
68 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(139,
   ↪ 140, 60, NULL, 'text');
69 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(142,
   ↪ 143, 141, NULL, 'text');
70 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(144,
   ↪ 145, 141, 'answer', 'element');
71 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(146,
   ↪ 147, 141, NULL, 'text');
72 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(141,
   ↪ 148, 60, 'answers', 'element');
73 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(149,
   ↪ 150, 60, NULL, 'text');
74 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(152,
   ↪ 153, 151, NULL, 'text');
75 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(151,
   ↪ 154, 60, 'operating_system', 'element');
76 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(155,
   ↪ 156, 60, NULL, 'text');
77 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(158,
   ↪ 159, 157, NULL, 'text');
78 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(161,
   ↪ 162, 160, NULL, 'text');
79 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(160,
   ↪ 163, 157, 'image', 'element');
80 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(164,
   ↪ 165, 157, NULL, 'text');
81 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(167,
   ↪ 168, 166, NULL, 'text');
82 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(166,
   ↪ 169, 157, 'image', 'element');
83 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(170,
   ↪ 171, 157, NULL, 'text');
84 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(157,
   ↪ 172, 60, 'images', 'element');
85 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(173,
   ↪ 174, 60, NULL, 'text');
86 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(176,
   ↪ 177, 175, NULL, 'text');
87 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(179,
   ↪ 180, 178, NULL, 'text');
88 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(178,
   ↪ 181, 175, 'video', 'element');
```

```
89  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(182,
    ↪  183, 175, NULL, 'text');
90  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(175,
    ↪  184, 60, 'videos', 'element');
91  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(60,
    ↪  185, 59, 'header', 'element');
92  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(187,
    ↪  188, 186, NULL, 'text');
93  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(190,
    ↪  191, 189, NULL, 'text');
94  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(189,
    ↪  192, 186, 'text', 'element');
95  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(193,
    ↪  194, 186, NULL, 'text');
96  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(196,
    ↪  197, 195, NULL, 'text');
97  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(195,
    ↪  198, 186, 'text', 'element');
98  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(199,
    ↪  200, 186, NULL, 'text');
99  INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(201,
    ↪  202, 186, 'image_ref', 'element');
100 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(203,
    ↪  204, 186, NULL, 'text');
101 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(206,
    ↪  207, 205, NULL, 'text');
102 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(205,
    ↪  208, 186, 'user_ref', 'element');
103 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(209,
    ↪  210, 186, NULL, 'text');
104 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(212,
    ↪  213, 211, NULL, 'text');
105 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(211,
    ↪  214, 186, 'text', 'element');
106 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(215,
    ↪  216, 186, NULL, 'text');
107 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(217,
    ↪  218, 186, 'video_ref', 'element');
108 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(186,
    ↪  219, 59, 'body', 'element');
109 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(59,
    ↪  220, 58, 'tweet', 'element');
110 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(223,
    ↪  224, 222, NULL, 'text');
111 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(226,
    ↪  227, 225, NULL, 'text');
```

```
112 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(225,
    ↪ 228, 222, 'date', 'element');
113 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(229,
    ↪ 230, 222, NULL, 'text');
114 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(232,
    ↪ 233, 231, NULL, 'text');
115 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(235,
    ↪ 236, 234, NULL, 'text');
116 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(234,
    ↪ 237, 231, 'standard', 'element');
117 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(238,
    ↪ 239, 231, NULL, 'text');
118 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(241,
    ↪ 242, 240, NULL, 'text');
119 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(240,
    ↪ 243, 231, 'offset', 'element');
120 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(244,
    ↪ 245, 231, NULL, 'text');
121 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(231,
    ↪ 246, 222, 'timezone', 'element');
122 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(247,
    ↪ 248, 222, NULL, 'text');
123 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(250,
    ↪ 251, 249, NULL, 'text');
124 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(253,
    ↪ 254, 252, NULL, 'text');
125 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(252,
    ↪ 255, 249, 'latitude', 'element');
126 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(256,
    ↪ 257, 249, NULL, 'text');
127 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(259,
    ↪ 260, 258, NULL, 'text');
128 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(258,
    ↪ 261, 249, 'longitude', 'element');
129 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(262,
    ↪ 263, 249, NULL, 'text');
130 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(265,
    ↪ 266, 264, NULL, 'text');
131 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(264,
    ↪ 267, 249, 'city', 'element');
132 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(268,
    ↪ 269, 249, NULL, 'text');
133 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(271,
    ↪ 272, 270, NULL, 'text');
134 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(270,
    ↪ 273, 249, 'country', 'element');
```

```
135 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(274,
    ↪ 275, 249, NULL, 'text');
136 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(249,
    ↪ 276, 222, 'location', 'element');
137 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(277,
    ↪ 278, 222, NULL, 'text');
138 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(280,
    ↪ 281, 279, NULL, 'text');
139 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(279,
    ↪ 282, 222, 'language', 'element');
140 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(283,
    ↪ 284, 222, NULL, 'text');
141 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(286,
    ↪ 287, 285, NULL, 'text');
142 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(285,
    ↪ 288, 222, 'retweets', 'element');
143 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(289,
    ↪ 290, 222, NULL, 'text');
144 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(292,
    ↪ 293, 291, NULL, 'text');
145 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(294,
    ↪ 295, 291, 'answer', 'element');
146 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(296,
    ↪ 297, 291, NULL, 'text');
147 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(298,
    ↪ 299, 291, 'answer', 'element');
148 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(300,
    ↪ 301, 291, NULL, 'text');
149 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(302,
    ↪ 303, 291, 'answer', 'element');
150 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(304,
    ↪ 305, 291, NULL, 'text');
151 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(291,
    ↪ 306, 222, 'answers', 'element');
152 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(307,
    ↪ 308, 222, NULL, 'text');
153 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(310,
    ↪ 311, 309, NULL, 'text');
154 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(309,
    ↪ 312, 222, 'operating_system', 'element');
155 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(222,
    ↪ 313, 221, 'header', 'element');
156 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(315,
    ↪ 316, 314, NULL, 'text');
157 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(318,
    ↪ 319, 317, NULL, 'text');
```

```
158 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(317,
    ↪  320, 314, 'hashtag', 'element');
159 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(321,
    ↪  322, 314, NULL, 'text');
160 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(324,
    ↪  325, 323, NULL, 'text');
161 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(323,
    ↪  326, 314, 'text', 'element');
162 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(327,
    ↪  328, 314, NULL, 'text');
163 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(330,
    ↪  331, 329, NULL, 'text');
164 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(332,
    ↪  333, 329, NULL, 'text');
165 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(334,
    ↪  335, 329, NULL, 'text');
166 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(329,
    ↪  336, 314, 'hashtag', 'element');
167 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(337,
    ↪  338, 314, NULL, 'text');
168 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(340,
    ↪  341, 339, NULL, 'text');
169 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(339,
    ↪  342, 314, 'text', 'element');
170 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(343,
    ↪  344, 314, NULL, 'text');
171 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(345,
    ↪  346, 314, 'video_ref', 'element');
172 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(314,
    ↪  347, 221, 'body', 'element');
173 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(221,
    ↪  348, 58, 'tweet', 'element');
174 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(351,
    ↪  352, 350, NULL, 'text');
175 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(354,
    ↪  355, 353, NULL, 'text');
176 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(353,
    ↪  356, 350, 'date', 'element');
177 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(357,
    ↪  358, 350, NULL, 'text');
178 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(360,
    ↪  361, 359, NULL, 'text');
179 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(363,
    ↪  364, 362, NULL, 'text');
180 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(362,
    ↪  365, 359, 'standard', 'element');
```

```
181 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(366,
    ↪  367, 359, NULL, 'text');
182 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(369,
    ↪  370, 368, NULL, 'text');
183 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(368,
    ↪  371, 359, 'offset', 'element');
184 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(372,
    ↪  373, 359, NULL, 'text');
185 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(359,
    ↪  374, 350, 'timezone', 'element');
186 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(375,
    ↪  376, 350, NULL, 'text');
187 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(378,
    ↪  379, 377, NULL, 'text');
188 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(381,
    ↪  382, 380, NULL, 'text');
189 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(380,
    ↪  383, 377, 'latitude', 'element');
190 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(384,
    ↪  385, 377, NULL, 'text');
191 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(387,
    ↪  388, 386, NULL, 'text');
192 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(386,
    ↪  389, 377, 'longitude', 'element');
193 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(390,
    ↪  391, 377, NULL, 'text');
194 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(393,
    ↪  394, 392, NULL, 'text');
195 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(392,
    ↪  395, 377, 'city', 'element');
196 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(396,
    ↪  397, 377, NULL, 'text');
197 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(399,
    ↪  400, 398, NULL, 'text');
198 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(398,
    ↪  401, 377, 'country', 'element');
199 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(402,
    ↪  403, 377, NULL, 'text');
200 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(377,
    ↪  404, 350, 'location', 'element');
201 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(405,
    ↪  406, 350, NULL, 'text');
202 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(408,
    ↪  409, 407, NULL, 'text');
203 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(407,
    ↪  410, 350, 'language', 'element');
```

```
204 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(411,
    ↪ 412, 350, NULL, 'text');
205 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(413,
    ↪ 414, 350, 'retweets', 'element');
206 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(415,
    ↪ 416, 350, NULL, 'text');
207 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(417,
    ↪ 418, 350, 'answers', 'element');
208 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(419,
    ↪ 420, 350, NULL, 'text');
209 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(422,
    ↪ 423, 421, NULL, 'text');
210 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(421,
    ↪ 424, 350, 'operating_system', 'element');
211 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(350,
    ↪ 425, 349, 'header', 'element');
212 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(427,
    ↪ 428, 426, NULL, 'text');
213 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(430,
    ↪ 431, 429, NULL, 'text');
214 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(429,
    ↪ 432, 426, 'text', 'element');
215 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(433,
    ↪ 434, 426, NULL, 'text');
216 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(436,
    ↪ 437, 435, NULL, 'text');
217 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(435,
    ↪ 438, 426, 'hashtag', 'element');
218 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(439,
    ↪ 440, 426, NULL, 'text');
219 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(442,
    ↪ 443, 441, NULL, 'text');
220 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(441,
    ↪ 444, 426, 'hashtag', 'element');
221 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(426,
    ↪ 445, 349, 'body', 'element');
222 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(349,
    ↪ 446, 58, 'tweet', 'element');
223 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(449,
    ↪ 450, 448, NULL, 'text');
224 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(452,
    ↪ 453, 451, NULL, 'text');
225 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(451,
    ↪ 454, 448, 'date', 'element');
226 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(455,
    ↪ 456, 448, NULL, 'text');
```

```
227 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(458,
    ↪ 459, 457, NULL, 'text');
228 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(461,
    ↪ 462, 460, NULL, 'text');
229 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(460,
    ↪ 463, 457, 'standard', 'element');
230 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(464,
    ↪ 465, 457, NULL, 'text');
231 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(467,
    ↪ 468, 466, NULL, 'text');
232 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(466,
    ↪ 469, 457, 'offset', 'element');
233 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(470,
    ↪ 471, 457, NULL, 'text');
234 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(457,
    ↪ 472, 448, 'timezone', 'element');
235 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(473,
    ↪ 474, 448, NULL, 'text');
236 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(476,
    ↪ 477, 475, NULL, 'text');
237 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(479,
    ↪ 480, 478, NULL, 'text');
238 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(478,
    ↪ 481, 475, 'latitude', 'element');
239 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(482,
    ↪ 483, 475, NULL, 'text');
240 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(485,
    ↪ 486, 484, NULL, 'text');
241 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(484,
    ↪ 487, 475, 'longitude', 'element');
242 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(488,
    ↪ 489, 475, NULL, 'text');
243 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(491,
    ↪ 492, 490, NULL, 'text');
244 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(490,
    ↪ 493, 475, 'city', 'element');
245 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(494,
    ↪ 495, 475, NULL, 'text');
246 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(497,
    ↪ 498, 496, NULL, 'text');
247 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(496,
    ↪ 499, 475, 'country', 'element');
248 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(500,
    ↪ 501, 475, NULL, 'text');
249 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(475,
    ↪ 502, 448, 'location', 'element');
```

```
250 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(503,
    ↪  504, 448, NULL, 'text');
251 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(506,
    ↪  507, 505, NULL, 'text');
252 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(505,
    ↪  508, 448, 'language', 'element');
253 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(509,
    ↪  510, 448, NULL, 'text');
254 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(511,
    ↪  512, 448, 'retweets', 'element');
255 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(513,
    ↪  514, 448, NULL, 'text');
256 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(515,
    ↪  516, 448, 'answers', 'element');
257 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(517,
    ↪  518, 448, NULL, 'text');
258 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(520,
    ↪  521, 519, NULL, 'text');
259 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(519,
    ↪  522, 448, 'operating_system', 'element');
260 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(448,
    ↪  523, 447, 'header', 'element');
261 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(525,
    ↪  526, 524, NULL, 'text');
262 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(528,
    ↪  529, 527, NULL, 'text');
263 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(527,
    ↪  530, 524, 'text', 'element');
264 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(524,
    ↪  531, 447, 'body', 'element');
265 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(447,
    ↪  532, 58, 'tweet', 'element');
266 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(535,
    ↪  536, 534, NULL, 'text');
267 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(538,
    ↪  539, 537, NULL, 'text');
268 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(537,
    ↪  540, 534, 'date', 'element');
269 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(541,
    ↪  542, 534, NULL, 'text');
270 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(544,
    ↪  545, 543, NULL, 'text');
271 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(547,
    ↪  548, 546, NULL, 'text');
272 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(546,
    ↪  549, 543, 'standard', 'element');
```

```
273 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(550,
    ↪  551, 543, NULL, 'text');
274 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(553,
    ↪  554, 552, NULL, 'text');
275 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(552,
    ↪  555, 543, 'offset', 'element');
276 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(556,
    ↪  557, 543, NULL, 'text');
277 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(543,
    ↪  558, 534, 'timezone', 'element');
278 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(559,
    ↪  560, 534, NULL, 'text');
279 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(562,
    ↪  563, 561, NULL, 'text');
280 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(565,
    ↪  566, 564, NULL, 'text');
281 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(564,
    ↪  567, 561, 'latitude', 'element');
282 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(568,
    ↪  569, 561, NULL, 'text');
283 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(571,
    ↪  572, 570, NULL, 'text');
284 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(570,
    ↪  573, 561, 'longitude', 'element');
285 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(574,
    ↪  575, 561, NULL, 'text');
286 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(577,
    ↪  578, 576, NULL, 'text');
287 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(576,
    ↪  579, 561, 'city', 'element');
288 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(580,
    ↪  581, 561, NULL, 'text');
289 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(583,
    ↪  584, 582, NULL, 'text');
290 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(582,
    ↪  585, 561, 'country', 'element');
291 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(586,
    ↪  587, 561, NULL, 'text');
292 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(561,
    ↪  588, 534, 'location', 'element');
293 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(589,
    ↪  590, 534, NULL, 'text');
294 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(592,
    ↪  593, 591, NULL, 'text');
295 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(591,
    ↪  594, 534, 'language', 'element');
```

```
296 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(595,
    ↪  596, 534, NULL, 'text');
297 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(597,
    ↪  598, 534, 'retweets', 'element');
298 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(599,
    ↪  600, 534, NULL, 'text');
299 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(601,
    ↪  602, 534, 'answers', 'element');
300 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(603,
    ↪  604, 534, NULL, 'text');
301 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(606,
    ↪  607, 605, NULL, 'text');
302 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(605,
    ↪  608, 534, 'operating_system', 'element');
303 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(534,
    ↪  609, 533, 'header', 'element');
304 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(611,
    ↪  612, 610, NULL, 'text');
305 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(614,
    ↪  615, 613, NULL, 'text');
306 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(613,
    ↪  616, 610, 'text', 'element');
307 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(610,
    ↪  617, 533, 'body', 'element');
308 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(533,
    ↪  618, 58, 'tweet', 'element');
309 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(621,
    ↪  622, 620, NULL, 'text');
310 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(624,
    ↪  625, 623, NULL, 'text');
311 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(623,
    ↪  626, 620, 'date', 'element');
312 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(627,
    ↪  628, 620, NULL, 'text');
313 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(630,
    ↪  631, 629, NULL, 'text');
314 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(633,
    ↪  634, 632, NULL, 'text');
315 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(632,
    ↪  635, 629, 'standard', 'element');
316 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(636,
    ↪  637, 629, NULL, 'text');
317 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(639,
    ↪  640, 638, NULL, 'text');
318 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(638,
    ↪  641, 629, 'offset', 'element');
```

```
319 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(642,
    ↪  643, 629, NULL, 'text');
320 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(629,
    ↪  644, 620, 'timezone', 'element');
321 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(645,
    ↪  646, 620, NULL, 'text');
322 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(648,
    ↪  649, 647, NULL, 'text');
323 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(651,
    ↪  652, 650, NULL, 'text');
324 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(650,
    ↪  653, 647, 'latitude', 'element');
325 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(654,
    ↪  655, 647, NULL, 'text');
326 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(657,
    ↪  658, 656, NULL, 'text');
327 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(656,
    ↪  659, 647, 'longitude', 'element');
328 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(660,
    ↪  661, 647, NULL, 'text');
329 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(663,
    ↪  664, 662, NULL, 'text');
330 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(662,
    ↪  665, 647, 'city', 'element');
331 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(666,
    ↪  667, 647, NULL, 'text');
332 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(669,
    ↪  670, 668, NULL, 'text');
333 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(668,
    ↪  671, 647, 'country', 'element');
334 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(672,
    ↪  673, 647, NULL, 'text');
335 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(647,
    ↪  674, 620, 'location', 'element');
336 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(675,
    ↪  676, 620, NULL, 'text');
337 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(678,
    ↪  679, 677, NULL, 'text');
338 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(677,
    ↪  680, 620, 'language', 'element');
339 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(681,
    ↪  682, 620, NULL, 'text');
340 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(683,
    ↪  684, 620, 'retweets', 'element');
341 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(685,
    ↪  686, 620, NULL, 'text');
```

```
342 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(687,
    ↪  688, 620, 'answers', 'element');
343 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(689,
    ↪  690, 620, NULL, 'text');
344 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(692,
    ↪  693, 691, NULL, 'text');
345 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(691,
    ↪  694, 620, 'operating_system', 'element');
346 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(620,
    ↪  695, 619, 'header', 'element');
347 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(697,
    ↪  698, 696, NULL, 'text');
348 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(700,
    ↪  701, 699, NULL, 'text');
349 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(702,
    ↪  703, 699, NULL, 'text');
350 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(699,
    ↪  704, 696, 'text', 'element');
351 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(696,
    ↪  705, 619, 'body', 'element');
352 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(619,
    ↪  706, 58, 'tweet', 'element');
353 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(709,
    ↪  710, 708, NULL, 'text');
354 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(712,
    ↪  713, 711, NULL, 'text');
355 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(711,
    ↪  714, 708, 'date', 'element');
356 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(715,
    ↪  716, 708, NULL, 'text');
357 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(718,
    ↪  719, 717, NULL, 'text');
358 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(721,
    ↪  722, 720, NULL, 'text');
359 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(720,
    ↪  723, 717, 'standard', 'element');
360 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(724,
    ↪  725, 717, NULL, 'text');
361 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(727,
    ↪  728, 726, NULL, 'text');
362 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(726,
    ↪  729, 717, 'offset', 'element');
363 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(730,
    ↪  731, 717, NULL, 'text');
364 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(717,
    ↪  732, 708, 'timezone', 'element');
```

```
365 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(733,
    ↪ 734, 708, NULL, 'text');
366 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(736,
    ↪ 737, 735, NULL, 'text');
367 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(739,
    ↪ 740, 738, NULL, 'text');
368 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(738,
    ↪ 741, 735, 'latitude', 'element');
369 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(742,
    ↪ 743, 735, NULL, 'text');
370 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(745,
    ↪ 746, 744, NULL, 'text');
371 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(744,
    ↪ 747, 735, 'longitude', 'element');
372 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(748,
    ↪ 749, 735, NULL, 'text');
373 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(751,
    ↪ 752, 750, NULL, 'text');
374 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(750,
    ↪ 753, 735, 'city', 'element');
375 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(754,
    ↪ 755, 735, NULL, 'text');
376 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(757,
    ↪ 758, 756, NULL, 'text');
377 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(756,
    ↪ 759, 735, 'country', 'element');
378 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(760,
    ↪ 761, 735, NULL, 'text');
379 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(735,
    ↪ 762, 708, 'location', 'element');
380 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(763,
    ↪ 764, 708, NULL, 'text');
381 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(766,
    ↪ 767, 765, NULL, 'text');
382 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(765,
    ↪ 768, 708, 'language', 'element');
383 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(769,
    ↪ 770, 708, NULL, 'text');
384 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(771,
    ↪ 772, 708, 'retweets', 'element');
385 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(773,
    ↪ 774, 708, NULL, 'text');
386 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(775,
    ↪ 776, 708, 'answers', 'element');
387 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(777,
    ↪ 778, 708, NULL, 'text');
```

```
388 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(780,
    ↪  781, 779, NULL, 'text');
389 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(779,
    ↪  782, 708, 'operating_system', 'element');
390 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(708,
    ↪  783, 707, 'header', 'element');
391 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(785,
    ↪  786, 784, NULL, 'text');
392 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(788,
    ↪  789, 787, NULL, 'text');
393 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(787,
    ↪  790, 784, 'text', 'element');
394 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(784,
    ↪  791, 707, 'body', 'element');
395 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(707,
    ↪  792, 58, 'tweet', 'element');
396 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(58,
    ↪  793, 1, 'tweets', 'element');
397 INSERT INTO NODE (begin, end, parent, tag, nodetype) VALUES(1,
    ↪  794, NULL, 'tweeter', 'element');
```

4.4.2 Requtes SQL

Les ancetres des tweets

```
1 /tweeter/tweets/ancestor::node()
```

```
1 SELECT N3.*
2 FROM NODE N1, NODE N2, NODE N3
3 WHERE N1.tag = 'tweeter'   AND
4       N2.tag = 'tweets'    AND
5       N2.parent = N1.begin AND
6       N2.begin > N3.begin  AND
7       N2.end < N3.end;
```

Les prenoms des utilisateurs

```
1 /tweeter/users/user/child::first_name
```

```
1 SELECT N4.*
2 FROM NODE N1, NODE N2, NODE N3, NODE N4
3 WHERE N1.tag = 'tweeter'    AND
4       N2.tag = 'users'      AND
5       N3.tag = 'user'       AND
6       N4.tag = 'first_name' AND
7       N2.parent = N1.begin  AND
8       N3.parent = N2.begin  AND
9       N4.parent = N3.begin;
```

Les descendants des en-ttes de tweets

```
/tweeter/tweets/tweet/header/descendant::node()
```

```
SELECT N5.*
FROM NODE N1, NODE N2, NODE N3, NODE N4, NODE N5
WHERE N1.tag = 'tweeter'   AND
      N2.tag = 'tweets'    AND
      N3.tag = 'tweet'     AND
      N4.tag = 'header'    AND
      N2.parent = N1.begin AND
      N3.parent = N2.begin AND
      N4.parent = N3.begin AND
      N4.begin < N5.begin  AND
      N4.end   > N5.end;
```

Le frere suivant de users si le nom de balise est tweets

```
  /tweeter/users/following-sibling::tweets
```

```
SELECT N3.*
FROM NODE N1, NODE N2, NODE N3
WHERE N1.tag = 'tweeter'    AND
      N2.tag = 'users'      AND
      N3.tag = 'tweets'     AND
      N2.parent = N3.parent AND
      N3.begin > N2.end;
```

Le frre prcedent de tweets dont le nom est users

```
/tweeter/tweets/preceding-sibling::users
```

```
SELECT N3.*
FROM NODE N1, NODE N2, NODE N3
WHERE N1.tag = 'tweeter'    AND
      N2.tag = 'tweets'     AND
      N3.tag = 'users'      AND
      N2.parent = N3.parent AND
      N3.end < N2.begin;
```

Les noeuds parents des noeuds dont le nom de balise est offset et dont les ancestres sont dans l ordre ascendant timezone, header, tweet, tweets et tweeter

```
/tweeter/tweets/tweet/header/timezone/parent::*
```

```
SELECT N7.*
FROM NODE N1, NODE N2, NODE N3, NODE N4, NODE N5, NODE N6, NODE
↪  N7
```

```
3  WHERE N1.tag = 'tweeter'   AND
4        N2.tag = 'tweets'    AND
5        N3.tag = 'tweet'     AND
6        N4.tag = 'header'    AND
7        N5.tag = 'timezone'  AND
8        N6.tag = 'offset'    AND
9        N2.parent = N1.begin AND
10       N3.parent = N2.begin AND
11       N4.parent = N3.begin AND
12       N5.parent = N4.begin AND
13       N6.parent = N5.begin AND
14       N7.begin  = N6.parent;
```

4.5 Testez le programme avec le fichier XMLhttp ://www.ins.cwi.nl/ et reportez le temps dexecution

La dure d'execution affiche pour une xecution avec l'outil GNU time est 1 minute et 15,130s. Cependant, une grande partie du temps est occup par les affichages des instructions d'insertions, en effet en redirigeant la sortie standard vers l'entre standard du programme GNU tail, la dure d'execution pour une execution est d'approximativement 7 secondes.

Annexe A

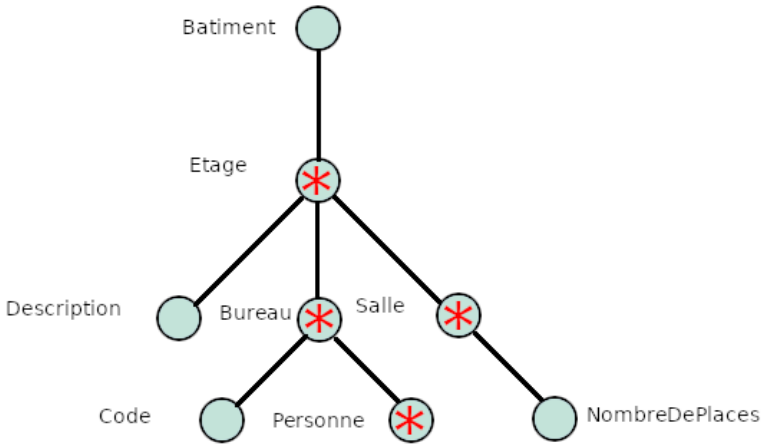
XML - Presse

```xml
<presse>
  <journal>
    <nom>
      Cnews
    </nom>
    <directeur>
      <nom>
        Pepega
      </nom>
      <prenom>
        Kekw
      </prenom>
    </directeur>
    <article titre="fake" auteur="j1">
      <corps>
        Des fake news
      </corps>
    </article>
    <article titre="news" auteur="j1">
      <corps>
        Encore des fake news
      </corps>
    </article>
  </journal>
  <journalistes>
    <journaliste idJ="j1">
      <nom>
        vuillard
      </nom>
      <prenom>
        eric
```

```
32        </prenom>
33      </journaliste>
34      <journaliste idJ="j2" anonymisation="oui">
35        <nom>
36          Dupont
37        </nom>
38        <prenom>
39          Jean
40        </prenom>
41      </journaliste>
42    </journalistes>
43  </presse>
```

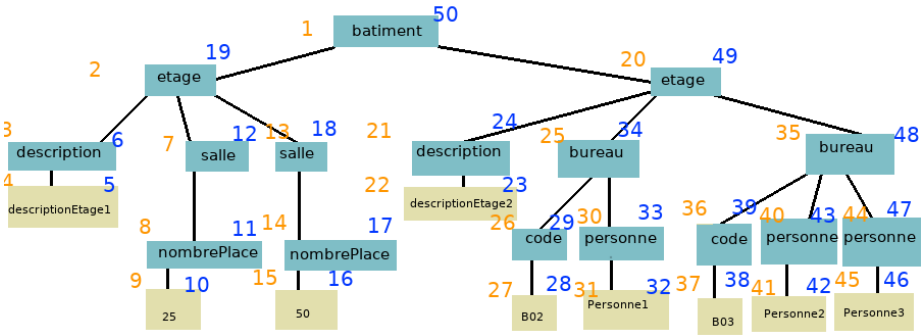
Annexe B

Representation sous forme de graphe


Batiment
Etage
Description
Bureau
Salle
Code
Personne
NombreDePlaces

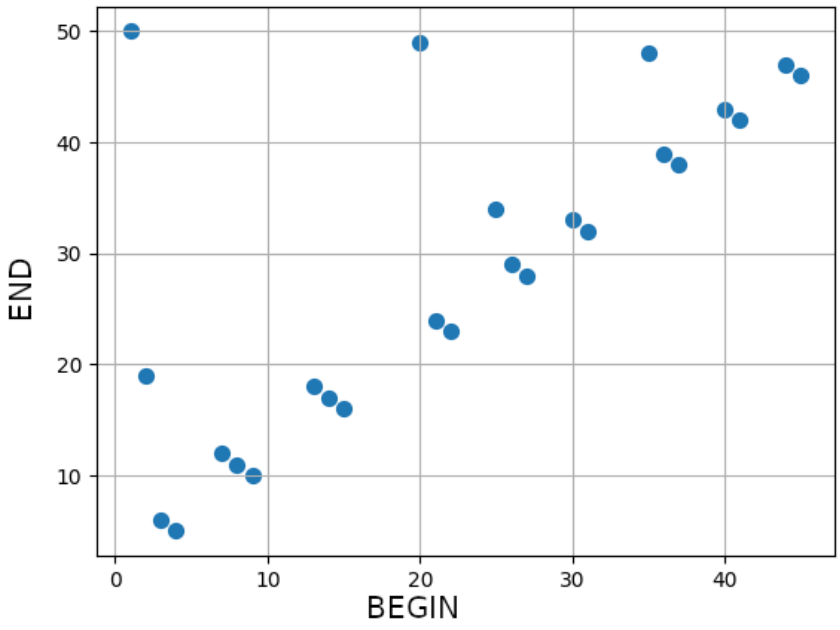
Annexe C

Arbre begin/end pour le document batiment


batiment
etage
etage
description
salle
salle
description
bureau
bureau
descriptionEtage1
nombrePlace
nombrePlace
descriptionEtage2
code
personne
code
personne
personne
25
50
B02
Personne1
B03
Personne2
Personne3

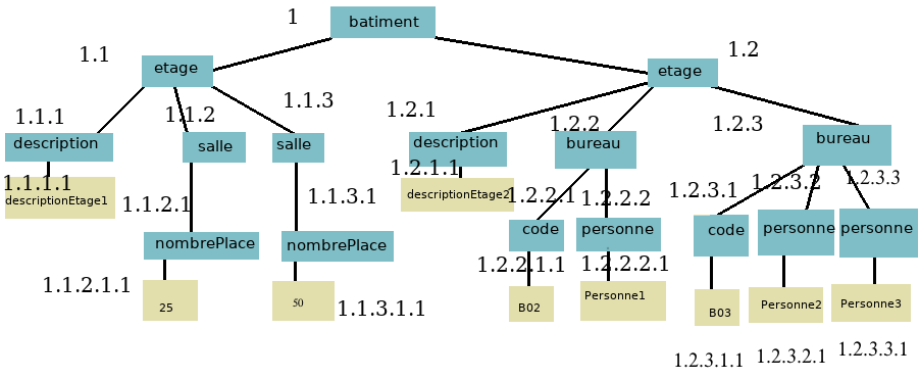
Annexe D

Graphe begin/end pour le document batiment


END
BEGIN
50
40
30
20
10
0
10
20
30
40

Annexe E

Arbre Dewey pour le document batiment


1
batiment
1.1
etage
1.2
etage
1.1.1
description
1.1.2
salle
1.1.3
salle
1.2.1
description
1.2.2
bureau
1.2.3
bureau
1.1.1.1
descriptionEtage1
1.1.2.1
nombrePlace
1.1.3.1
nombrePlace
1.2.1.1
descriptionEtage2
1.2.2.1
code
1.2.2.2
personne
1.2.3.1
code
1.2.3.2
personne
1.2.3.3
personne
1.1.2.1.1
25
1.1.3.1.1
50
1.2.2.1.1
B02
1.2.2.2.1
Personne1
1.2.3.1.1
B03
1.2.3.2.1
Personne2
1.2.3.3.1
Personne3