Université Catholique de Louvain

LINGI2172 - Databases

Mission 3 - Database Design

Auteur :
Baugnies Benjamin (6020-10-00)
Colson Olivier (5039-10-00)
Vanwelde Romain (3143-10-00)

Group 3

Superviseurs :
Pr. Bernard Lambeau
Antoine Cailliau

6 mai 2014

1 Introduction

Telling a story is challenging. Indeed, in order to build a “good" scenario, one must think of a lot of different aspects and ensure coherence between all these points.

Some can manage it using diagrams, other can rely on tons of paper sheets referencing each other or, more reasonably, use a computer to store all their documents. However, with all these ways of working, the same problem arises : as the storyline and background get denser, it becomes more and more difficult to ensure that no contradiction appears. This is a big problem, since contradictions ruin the feeling of reality that must always be given by a good scenario. How about asking the computer to gather, interpret and display all this information in a clean and understandable way ?

Our project can be defined as a “narration manager". Its goal is to make it easier for people to write coherent and complex scenarios without either becoming mad or cancelling their project because of its increasing complexity. It is intended for all "story makers" (videogame makers, film makers, roleplayers, writers, ...), and is thus meant to be generic and conveniently adapt to various situations, as well as user preferences and priorities in the story (for example, some users could want to define a precise date for each event happening in their story, as others could prefer to focus on the relations binding all the characters together).

Possibilities of telling a story are infinite, yet time and coordination constraints often limit what is actually possible to achieve. It is now time to push these limits away.

2 Elementary Facts

Bellow are some elementary facts we wrote to better understand what to do, which relation exists between all the entities.

2.1 About characters

Pierre is from Bruxelles.
Pierre is born on 28/12/1992
Jean is born on 01/04/1992
Benjamin is born on 03/06/1992
Jean is melancholic
Benjamin is member of the association "Les Petits Riens"

2.2 Characters relations

Pierre liked Benjamin from 9/12/2002 to 13/7/2007.
Benjamin liked Pierre from 10/11/2003 to 12/8/2009.
Jean doesn’t like Benjamin from 10/11/2003 to 12/8/2009.
Paul is Pierre’s father.

Pierre is Paul's son since 28/12/1992.

2.3 About events

Jean attended the event "The beer festival"
The "beer festival" took place at LLN
The "beer festival" lasted from 08/03/2014 to 18/03/2014.
The "beer festival" is "blablablablablablablabla" as description.

2.4 About places

Intel room is a sub-location of Réaumur's Map and is represented on square number 10.
Réaumur is a sub-location of LLN's Map and is represented on square number 5.
The LLN's Map represents the location "LLN"
The Réaumur's Map represents the location "Réaumur"
LLNMap has 10 square width, and represents a 5km distance.
LLNMAP has 20 square length, and represents a 10km distance.

3 ORM schema

The ORM schema is shown on the Figure 1. If some relations are too difficult to read, the numeric version is available in the Annexes directory of the zip file.

As we can see on the schema, there are 4 main entities which are `"Characters"`, `"Events"`, `"Place"`, and `"Map"`.

We will explain here above three specific case of the diagram :

Character - Relation relations involving time range, time or timeless notion.
The relations explain the relationships between different characters. They are uni-directional and the different kinds of relations can be defined by the user. We split these relations into three types to represent the fact that some relations are time-independent (e.g "... is my father"), some can start at a given time and be permanent and/or open-ended (e.g "... is my godfather since ... "), and finally some can last for only a while (e.g "... was my friend from ... to ... ").

Pseudo relation.
This relation involves two characters and a pseudonym. It describes the name used by the first character for the second one. This represents the fact that during the story, a given character might not know another's real name. We added this relation since this can have an impact on the story (he wouldn't realize others were talking about someone he knows for example). The corresponding table will also allow us the find all the pseudonyms a given person might go by.

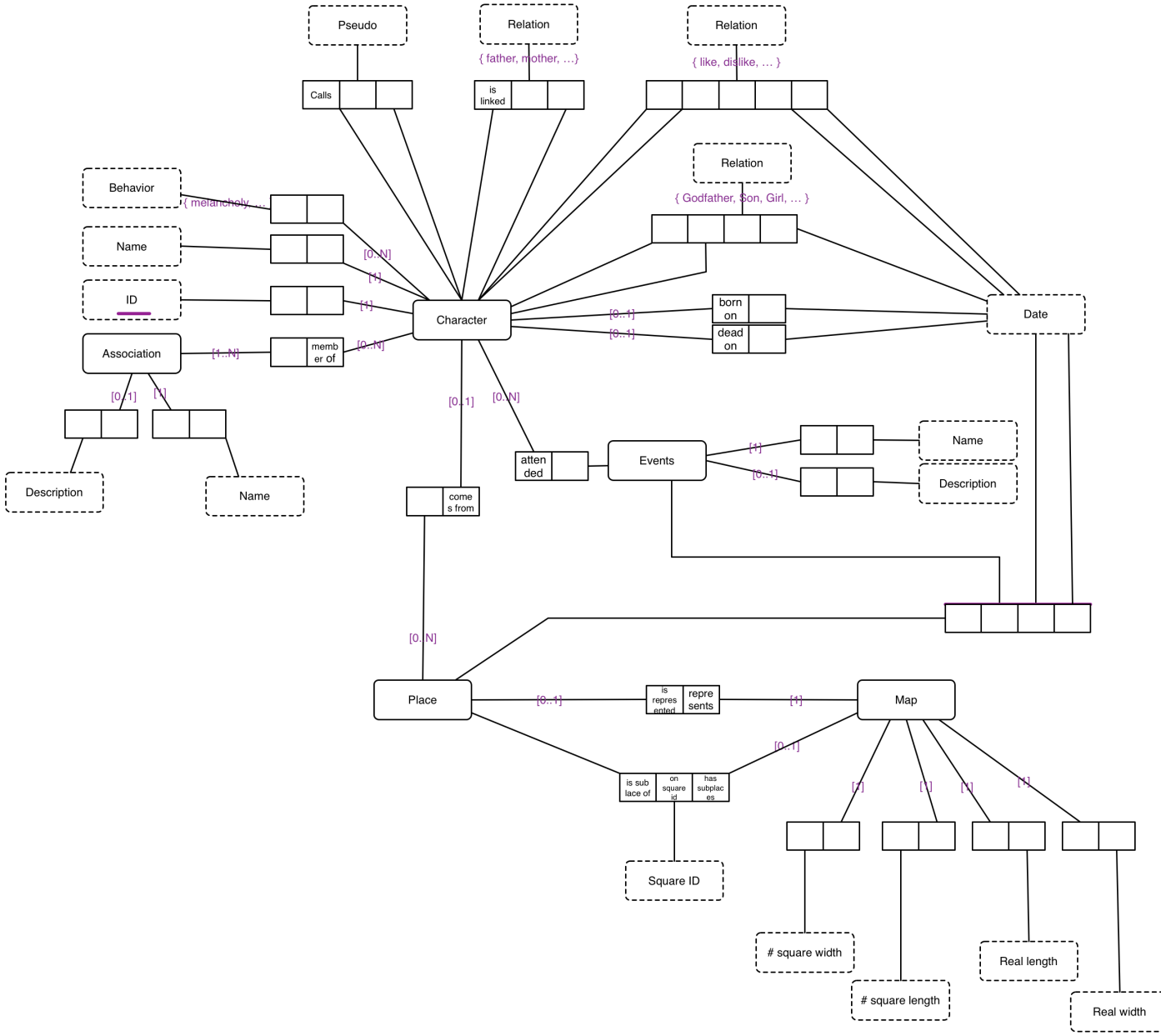

Figure 1 – ORM Schema

Place - Map relation.
This is a rather complex relation that we introduced to keep track of a story's geography at different levels. The first use is to allow users to situate events or characters. We can also define sub-places to refine locations. We can see that a place's map is optional. However, since a place's sub-places are linked to its map, it implicitly becomes required when we want to add levels. This structure allows us to chain an arbitrary number of levels with a "place - map - place - map - ... " hierarchy. This relation has two constraints that are not expressed in the database and will have to be verified in the software implementation. Firstly, the map square on which the sub-place is located must belong to the map's domain of possible squares (between 1 and (# square width)*(# square length)). Secondly, it is required that two places of the same level do not overlap.

4 Tutorial-D script

In the zip file, you can find files `structure_rel.d` and `data_rel.d`.

structure_rel.d has three differents parts. The **first** one is the definition of special types (All the entities ID, and one for some names too). Then, the **second** part is all the relvars, with all primary keys, and also unique keys (e.g. `MAPPEDPLACE` which has one key on the PLACEID and another on the MAPID). The third part is all the constraints which are foreign keys. We express first all the constraints implying CHARID, then implying ASSOCIATIONID, and so on.

data_rel.d fills the database with all elementary facts expressed before in the report.

5 Relvar predicates

In this section, we will give the relvar predicates that our database represents. The attributes of a relation are in bold, and the attributes that form the key are underlined.

About characters :

- The character **CHARACTERID** is named **NAME**. **ASSOCIATIONID**.
- The character **CHARACTERID** has a behavior that is **BEHAVIOR**.
- The character **CALLERID** knows the charactert **CALLEID** by the pseudonym **PSEUDONYME**.

About time :

- The character **CHARACTERID** was born on **BIRTH**.
- The character **CHARACTERID** died on **DEATH**.

About Associations :

- The association **ASSOCIATIONID** is named **NAME**.
- The association **ASSOCIATIONID** is described as **DESCRIPTION**.
- The character **CHARACTERID** is a member of the association.

About character relations :

- The relation **RELATIONID** is of the type **RELATIONSHIP**.
- The character **SOURCE** has a timeless relation with **TARGET** of the type **RELATIONID**.
- The character **SOURCE** started a permanent relation with **TARGET** of the type **RELATIONID** at the time **DATE**.
- The character **SOURCE** had a relation with **TARGET** of the type **RELATIONID** that started on **START** and ended on **ENDDATE**.

About places :

- The place **PLACEID** is called **PLACENAME**.
- The map **MAPID** has a width of **WIDTH** split into **NUMWIDTH** sections, and a length **LENGHT** split into **NUMLENGTH** sections.
- The place **PLACEID** is represented by the map **MAPID**.
- The place **PLACEID** is locate on the square **SQUAREID** of the **MAPID** map.
- The character **CHARACTERID** originates from **PLACEID**.

About events :

- The event **EVENTID** is named **NAME**.
- The event **EVENTID** is described as **DESCRIPTION**.
- The character **CHARACTERID** attended **EVENTID**.
- The event **EVENTID** happened at **PLACEID**, started on **BEGINNING**, and ended on **ENDDATE**.

6 SQL script

In the zip file, you can find files `structure.sql` which builds the entire structure of the database in SQL. This script is idempotent, and is build with the same structure as the tutorial-D script. Indeed, we first create all the tables with the primary keys/unique keys, then we alter them to add the foreign key constraints.

The file `data.sql` is also available in the zip file. This one is also idempotent. It builds the structure of the database as the previous file, then it fills the database with all the elementary facts presented above in the report.

Annexes

A Code Rel

```
1  TYPE NAME  POSSREP {NAME CHAR};
2  TYPE CHARACTER#   POSSREP {CHARACTERNUM CHAR};
3  TYPE EVENT#     POSSREP {EVENTNUM CHAR};
4  TYPE MAP#       POSSREP {MAPNUM CHAR};
5  TYPE PLACE# POSSREP {PLACENUM CHAR};
6  TYPE ASSOCIATION#   POSSREP {ASSOCIATIONNUM CHAR};
7  TYPE RELATION#  POSSREP {RELATIONNUM CHAR};
8  TYPE ENHANCEDDATE   POSSREP {YEAR INTEGER, MONTH INTEGER, DAY INTEGER};
9
10
11
12
13 VAR CHARACTER   BASE RELATION {CHARACTERID CHARACTER#, NAME NAME} KEY {
       CHARACTERID};
14
15 VAR ASSOCIATION BASE RELATION {CHARACTERID CHARACTER#, ASSOCIATIONID
       ASSOCIATION#} KEY {CHARACTERID , ASSOCIATIONID};
16 VAR ORIGINATES  BASE RELATION {CHARACTERID CHARACTER#, PLACEID PLACE#} KEY
       {CHARACTERID};
17 VAR BEHAVIOR    BASE RELATION {CHARACTERID CHARACTER#, BEHAVIOR CHAR} KEY {
       CHARACTERID, BEHAVIOR};
18 VAR PSEUDO  BASE RELATION {CALLERID CHARACTER#, CALLEDID CHARACTER#,
       PSEUDONYME NAME} KEY {CALLERID, CALLEDID, PSEUDONYME};
19
20 VAR TIMELESSRELATION    BASE RELATION {SOURCE CHARACTER#, TARGET CHARACTER
       #, RELATIONID RELATION#} KEY {SOURCE, TARGET, RELATIONID};
21 VAR DATERELATION    BASE RELATION {SOURCE CHARACTER#, TARGET CHARACTER#,
       RELATIONID RELATION#, DATE ENHANCEDDATE} KEY {SOURCE, TARGET, RELATIONID
       , DATE};
22 VAR RANGERELATION   BASE RELATION {SOURCE CHARACTER#, TARGET CHARACTER#,
       RELATIONID RELATION#, START ENHANCEDDATE, ENDDATE ENHANCEDDATE} KEY {
       SOURCE, TARGET, RELATIONID, START, ENDDATE};
23
24
25 VAR BIRTH   BASE RELATION {CHARACTERID CHARACTER#, BIRTH ENHANCEDDATE} KEY
       {CHARACTERID};
26 VAR DEATH   BASE RELATION {CHARACTERID CHARACTER#, DEATH ENHANCEDDATE} KEY
       {CHARACTERID};
27
28 VAR ATTENDS BASE RELATION {CHARACTERID CHARACTER#, EVENTID EVENT#} KEY {
       CHARACTERID, EVENTID};
29
30 VAR RELATIONLIST    BASE RELATION {RELATIONID RELATION#, RELATIONTYPE CHAR}
        KEY {RELATIONID};
31
32 VAR EVENTNAME   BASE RELATION {EVENTID EVENT#, NAME CHAR} KEY {EVENTID};
33 VAR EVENTDESCRIPTION    BASE RELATION {EVENTID EVENT#, DESCRIPTION CHAR}
       KEY {EVENTID};
34
```

```
35 VAR ASSOCIATIONNAME BASE RELATION {ASSOCIATIONID ASSOCIATION#, NAME CHAR}
       KEY {ASSOCIATIONID};
36 VAR ASSOCIATIONDESCRIPTION  BASE RELATION {ASSOCIATIONID ASSOCIATION#,
       DESCRIPTION CHAR} KEY {ASSOCIATIONID};
37
38
39 VAR MAP       BASE RELATION {MAPID MAP#, NUMWIDTH INTEGER, NUMLENGTH INTEGER,
        WIDTH RATIONAL, LENGTH RATIONAL} KEY {MAPID};
40
41 VAR PLACE    BASE RELATION {PLACEID PLACE#, PLACENAME NAME} KEY {PLACEID};
42
43
44 VAR SUBPLACE     BASE RELATION {PLACEID PLACE#, SQUAREID INTEGER, MAPID MAP
       #} KEY {PLACEID, SQUAREID, MAPID};
45
46
47 VAR MAPPEDPLACE BASE RELATION {PLACEID PLACE#, MAPID MAP#} KEY {PLACEID}
       KEY {MAPID};
48
49 VAR EVENT    BASE RELATION {EVENTID EVENT#, PLACEID PLACE#, BEGINNING
       ENHANCEDDATE, ENDDATE ENHANCEDDATE} KEY {EVENTID, PLACEID, BEGINNING,
       ENDDATE};
50
51
52
53
54 CONSTRAINT C1 ASSOCIATION {CHARACTERID} <= CHARACTER {CHARACTERID};
55
56 CONSTRAINT C2 ORIGINATES {CHARACTERID} <= CHARACTER {CHARACTERID};
57 CONSTRAINT C3 BEHAVIOR {CHARACTERID} <= CHARACTER {CHARACTERID};
58 CONSTRAINT C4 (PSEUDO RENAME {CALLERID AS CHARACTERID}){CHARACTERID} <=
       CHARACTER {CHARACTERID};
59 CONSTRAINT C5 (PSEUDO RENAME {CALLEDID AS CHARACTERID}){CHARACTERID} <=
       CHARACTER {CHARACTERID};
60
61 CONSTRAINT C6 (TIMELESSRELATION RENAME {SOURCE AS CHARACTERID}) {
       CHARACTERID} <= CHARACTER {CHARACTERID};
62 CONSTRAINT C7 (DATERELATION RENAME {SOURCE AS CHARACTERID}) {CHARACTERID}
       <= CHARACTER {CHARACTERID};
63 CONSTRAINT C8 (RANGERELATION RENAME {SOURCE AS CHARACTERID}) {CHARACTERID}
       <= CHARACTER {CHARACTERID};
64
65 CONSTRAINT C9 (TIMELESSRELATION RENAME {TARGET AS CHARACTERID}){CHARACTERID
       } <= CHARACTER {CHARACTERID};
66 CONSTRAINT C10 (DATERELATION RENAME {TARGET AS CHARACTERID}){CHARACTERID}
       <= CHARACTER {CHARACTERID};
67 CONSTRAINT C11 (RANGERELATION RENAME {TARGET AS CHARACTERID}){CHARACTERID}
       <= CHARACTER {CHARACTERID};
68
69 CONSTRAINT C12 BIRTH {CHARACTERID} <= CHARACTER {CHARACTERID};
70 CONSTRAINT C13 DEATH {CHARACTERID} <= CHARACTER {CHARACTERID};
71
72 CONSTRAINT C14 ATTENDS {CHARACTERID} <= CHARACTER {CHARACTERID};
73
74
75
```

```
76
77 CONSTRAINT C15 ASSOCIATION {ASSOCIATIONID} <= ASSOCIATIONNAME {
       ASSOCIATIONID};
78 CONSTRAINT C16 ASSOCIATIONDESCRIPTION {ASSOCIATIONID} <= ASSOCIATIONNAME {
       ASSOCIATIONID};
79
80
81 CONSTRAINT C17 ORIGINATES {PLACEID} <= PLACE {PLACEID};
82 CONSTRAINT C18 SUBPLACE {PLACEID} <= PLACE {PLACEID};
83 CONSTRAINT C19 MAPPEDPLACE {PLACEID} <= PLACE {PLACEID};
84 CONSTRAINT C20 EVENT {PLACEID} <= PLACE {PLACEID};
85
86
87 CONSTRAINT C21 TIMELESSRELATION {RELATIONID} <= RELATIONLIST {RELATIONID};
88 CONSTRAINT C22 DATERELATION {RELATIONID} <= RELATIONLIST {RELATIONID};
89 CONSTRAINT C23 RANGERELATION {RELATIONID} <= RELATIONLIST {RELATIONID};
90
91
92 CONSTRAINT C31 ATTENDS {EVENTID} <= EVENTNAME {EVENTID};
93 CONSTRAINT C32 EVENTDESCRIPTION {EVENTID} <= EVENTNAME {EVENTID};
94 CONSTRAINT C33 EVENT {EVENTID} <= EVENTNAME {EVENTID};
95
96 CONSTRAINT C34 MAPPEDPLACE {MAPID} <= MAP {MAPID};
97 CONSTRAINT C35 SUBPLACE {MAPID} <= MAP {MAPID};
```

structure_rel.d

B Code SQL

```sql
DROP TABLE IF EXISTS CHARACTER CASCADE;
DROP TABLE IF EXISTS ASSOCIATION CASCADE;
DROP TABLE IF EXISTS ORIGINATES CASCADE;
DROP TABLE IF EXISTS BEHAVIOR CASCADE;
DROP TABLE IF EXISTS PSEUDO CASCADE;
DROP TABLE IF EXISTS TIMELESSRELATION CASCADE;
DROP TABLE IF EXISTS DATERELATION CASCADE;
DROP TABLE IF EXISTS RANGERELATION CASCADE;
DROP TABLE IF EXISTS BIRTH CASCADE;
DROP TABLE IF EXISTS DEATH CASCADE;
DROP TABLE IF EXISTS ATTENDS CASCADE;
DROP TABLE IF EXISTS RELATIONLIST CASCADE;
DROP TABLE IF EXISTS EVENTNAME CASCADE;
DROP TABLE IF EXISTS EVENTDESCRIPTION CASCADE;
DROP TABLE IF EXISTS ASSOCIATIONNAME CASCADE;
DROP TABLE IF EXISTS ASSOCIATIONDESCRIPTION CASCADE;
DROP TABLE IF EXISTS MAP CASCADE;
DROP TABLE IF EXISTS PLACE CASCADE;
DROP TABLE IF EXISTS SUBPLACE CASCADE;
DROP TABLE IF EXISTS SQUAREID CASCADE;
DROP TABLE IF EXISTS MAPPEDPLACE CASCADE;
DROP TABLE IF EXISTS EVENT CASCADE;
DROP TYPE IF EXISTS ENHANCEDDATE;

CREATE TYPE ENHANCEDDATE as (YEAR INTEGER, MONTH INTEGER, DAY INTEGER);

CREATE TABLE CHARACTER(
  CHARACTERID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
  NAME TEXT NOT NULL
);

CREATE TABLE ASSOCIATION(
  CHARACTERID CHARACTER VARYING(10) NOT NULL,
  ASSOCIATIONID CHARACTER VARYING(10) NOT NULL,
  PRIMARY KEY (CHARACTERID,ASSOCIATIONID)
);

CREATE TABLE ORIGINATES(
  CHARACTERID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
  PLACEID CHARACTER VARYING(10) NOT NULL
);

CREATE TABLE BEHAVIOR(
  CHARACTERID CHARACTER VARYING(10) NOT NULL,
  BEHAVIOR CHARACTER VARYING(42) NOT NULL,
  PRIMARY KEY (CHARACTERID,BEHAVIOR)
);

CREATE TABLE PSEUDO(
  CALLERID CHARACTER VARYING(10) NOT NULL,
  CALLEDID CHARACTER VARYING(10) NOT NULL,
  PSEUDONYME CHARACTER VARYING(42) NOT NULL,
  PRIMARY KEY (CALLERID,CALLEDID,PSEUDONYME)
```

```
54 );
55
56 CREATE TABLE TIMELESSRELATION(
57    SOURCE CHARACTER VARYING(10) NOT NULL,
58    TARGET CHARACTER VARYING(10) NOT NULL,
59    RELATIONID CHARACTER VARYING(10) NOT NULL,
60    PRIMARY KEY (SOURCE,TARGET,RELATIONID)
61 );
62
63 CREATE TABLE DATERELATION(
64    SOURCE CHARACTER VARYING(10) NOT NULL,
65    TARGET CHARACTER VARYING(10) NOT NULL,
66    RELATIONID CHARACTER VARYING(10) NOT NULL,
67    DATE ENHANCEDDATE NOT NULL,
68    PRIMARY KEY (SOURCE,TARGET,RELATIONID,DATE)
69 );
70
71 CREATE TABLE RANGERELATION(
72    SOURCE CHARACTER VARYING(10) NOT NULL,
73    TARGET CHARACTER VARYING(10) NOT NULL,
74    RELATIONID CHARACTER VARYING(10) NOT NULL,
75    START ENHANCEDDATE NOT NULL,
76    ENDDATE ENHANCEDDATE NOT NULL,
77    PRIMARY KEY (SOURCE,TARGET,RELATIONID,START,ENDDATE)
78 );
79
80 CREATE TABLE BIRTH(
81    CHARACTERID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
82    BIRTH ENHANCEDDATE NOT NULL
83 );
84
85 CREATE TABLE DEATH(
86    CHARACTERID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
87    DEATH ENHANCEDDATE NOT NULL
88 );
89
90 CREATE TABLE ATTENDS(
91    CHARACTERID CHARACTER VARYING(10) NOT NULL,
92    EVENTID CHARACTER VARYING(10) NOT NULL,
93    PRIMARY KEY (CHARACTERID,EVENTID)
94 );
95
96 CREATE TABLE RELATIONLIST(
97    RELATIONID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
98    RELATIONTYPE CHARACTER VARYING(42) NOT NULL
99 );
100
101 CREATE TABLE EVENTNAME(
102    EVENTID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
103    NAME CHARACTER VARYING(42) NOT NULL
104 );
105
106 CREATE TABLE EVENTDESCRIPTION(
107    EVENTID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
108    DESCRIPTION CHARACTER VARYING(42) NOT NULL
109 );
```

```
110
111 CREATE TABLE ASSOCIATIONNAME(
112    ASSOCIATIONID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
113    NAME CHARACTER VARYING(42) NOT NULL
114 );
115
116 CREATE TABLE ASSOCIATIONDESCRIPTION(
117    ASSOCIATIONID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
118    DESCRIPTION TEXT NOT NULL
119 );
120
121 CREATE TABLE MAP(
122    MAPID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
123    NUMWIDTH INTEGER NOT NULL,
124    NUMLENGTH INTEGER NOT NULL,
125    WIDTH DOUBLE PRECISION NOT NULL,
126    LENGTH DOUBLE PRECISION NOT NULL
127 );
128
129 CREATE TABLE PLACE(
130    PLACEID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
131    PLACENAME CHARACTER VARYING(42) NOT NULL
132 );
133
134 CREATE TABLE SUBPLACE(
135    PLACEID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
136    SQUAREID INTEGER NOT NULL,
137    MAPID CHARACTER VARYING(10) NOT NULL
138 );
139
140 CREATE TABLE SQUAREID(
141    PLACEID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
142    SQUAREID CHARACTER VARYING(10) NOT NULL
143 );
144
145 CREATE TABLE MAPPEDPLACE(
146    PLACEID CHARACTER VARYING(10) PRIMARY KEY NOT NULL,
147    MAPID CHARACTER VARYING(10) UNIQUE NOT NULL
148 );
149
150 CREATE TABLE EVENT(
151    EVENTID CHARACTER VARYING(10) NOT NULL,
152    PLACEID CHARACTER VARYING(10) NOT NULL,
153    BEGINNING ENHANCEDDATE NOT NULL,
154    ENDDATE ENHANCEDDATE NOT NULL,
155    PRIMARY KEY (EVENTID,PLACEID,BEGINNING,ENDDATE)
156 );
157
158 ALTER TABLE ONLY ASSOCIATION
159      ADD CONSTRAINT C1 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
           CHARACTERID);
160
161 ALTER TABLE ONLY ORIGINATES
162      ADD CONSTRAINT C2 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
           CHARACTERID);
163
```

```
164 ALTER TABLE ONLY BEHAVIOR
165     ADD CONSTRAINT C3 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
        CHARACTERID);
166
167 ALTER TABLE ONLY PSEUDO
168     ADD CONSTRAINT C4 FOREIGN KEY (CALLERID) REFERENCES CHARACTER(
        CHARACTERID);
169
170 ALTER TABLE ONLY PSEUDO
171     ADD CONSTRAINT C5 FOREIGN KEY (CALLEDID) REFERENCES CHARACTER(
        CHARACTERID);
172
173 ALTER TABLE ONLY TIMELESSRELATION
174     ADD CONSTRAINT C6 FOREIGN KEY (SOURCE) REFERENCES CHARACTER(CHARACTERID
        );
175
176 ALTER TABLE ONLY DATERELATION
177     ADD CONSTRAINT C7 FOREIGN KEY (SOURCE) REFERENCES CHARACTER(CHARACTERID
        );
178
179 ALTER TABLE ONLY RANGERELATION
180     ADD CONSTRAINT C8 FOREIGN KEY (SOURCE) REFERENCES CHARACTER(CHARACTERID
        );
181
182 ALTER TABLE ONLY TIMELESSRELATION
183     ADD CONSTRAINT C9 FOREIGN KEY (TARGET) REFERENCES CHARACTER(CHARACTERID
        );
184
185 ALTER TABLE ONLY DATERELATION
186     ADD CONSTRAINT C10 FOREIGN KEY (TARGET) REFERENCES CHARACTER(
        CHARACTERID);
187
188 ALTER TABLE ONLY RANGERELATION
189     ADD CONSTRAINT C11 FOREIGN KEY (TARGET) REFERENCES CHARACTER(
        CHARACTERID);
190
191 ALTER TABLE ONLY BIRTH
192     ADD CONSTRAINT C12 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
        CHARACTERID);
193
194 ALTER TABLE ONLY DEATH
195     ADD CONSTRAINT C13 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
        CHARACTERID);
196
197 ALTER TABLE ONLY ATTENDS
198     ADD CONSTRAINT C14 FOREIGN KEY (CHARACTERID) REFERENCES CHARACTER(
        CHARACTERID);
199
200 ALTER TABLE ONLY ASSOCIATION
201     ADD CONSTRAINT C15 FOREIGN KEY (ASSOCIATIONID) REFERENCES
        ASSOCIATIONNAME(ASSOCIATIONID);
202
203 ALTER TABLE ONLY ASSOCIATIONDESCRIPTION
204     ADD CONSTRAINT C16 FOREIGN KEY (ASSOCIATIONID) REFERENCES
        ASSOCIATIONNAME(ASSOCIATIONID);
205
```

```
206 ALTER TABLE ONLY ORIGINATES
207     ADD CONSTRAINT C17 FOREIGN KEY (PLACEID) REFERENCES PLACE(PLACEID);
208
209 ALTER TABLE ONLY SUBPLACE
210     ADD CONSTRAINT C18 FOREIGN KEY (PLACEID) REFERENCES PLACE(PLACEID);
211
212 ALTER TABLE ONLY MAPPEDPLACE
213     ADD CONSTRAINT C19 FOREIGN KEY (PLACEID) REFERENCES PLACE(PLACEID);
214
215 ALTER TABLE ONLY EVENT
216     ADD CONSTRAINT C20 FOREIGN KEY (PLACEID) REFERENCES PLACE(PLACEID);
217
218 ALTER TABLE ONLY TIMELESSRELATION
219     ADD CONSTRAINT C21 FOREIGN KEY (RELATIONID) REFERENCES RELATIONLIST(
        RELATIONID);
220
221 ALTER TABLE ONLY DATERELATION
222     ADD CONSTRAINT C22 FOREIGN KEY (RELATIONID) REFERENCES RELATIONLIST(
        RELATIONID);
223
224 ALTER TABLE ONLY RANGERELATION
225     ADD CONSTRAINT C23 FOREIGN KEY (RELATIONID) REFERENCES RELATIONLIST(
        RELATIONID);
226
227 ALTER TABLE ONLY ATTENDS
228     ADD CONSTRAINT C31 FOREIGN KEY (EVENTID) REFERENCES EVENTNAME(EVENTID);
229
230 ALTER TABLE ONLY EVENTDESCRIPTION
231     ADD CONSTRAINT C32 FOREIGN KEY (EVENTID) REFERENCES EVENTNAME(EVENTID);
232
233 ALTER TABLE ONLY EVENT
234     ADD CONSTRAINT C33 FOREIGN KEY (EVENTID) REFERENCES EVENTNAME(EVENTID);
235
236 ALTER TABLE ONLY MAPPEDPLACE
237     ADD CONSTRAINT C34 FOREIGN KEY (MAPID) REFERENCES MAP(MAPID);
238
239 ALTER TABLE ONLY SUBPLACE
240     ADD CONSTRAINT C35 FOREIGN KEY (MAPID) REFERENCES MAP(MAPID);
```

structure.sql