Data Model

KSU Students Portal

Version 1.0 Beta

Submitted in partial fulfillment of the requirements of the degree of MSE

Javier Ramos Rodríguez
CIS 895 – MSE Project
Kansas State University
Table of Contents

1. Description ................................................................. 3
2. E-R Diagram ............................................................... 4
3. UML Data Model .......................................................... 6
4. Tables ........................................................................ 7
5. SQL Code ..................................................................... 9
1. Description

The main entity that we are interested is the application user. We want to store information such the name, address, courses, mayor, home university, country…

A registered user will be able to publish articles, events and links. We want to store some information about those entities like, the header, the actual text, the date or the user that publish such item.

A user will also have a blog where he/she can publish new blog entries. Also, user will have the chance to send messages to each other.

With this information we will create an Entity-Relation diagram to model the data and its relations.
2. E-R Diagram
In the diagram I only show the relevant attributes to make it easier to read. The relations without names mean the usual “belongs to” or “has” relationship. A user will have associated a blog; the blog entity will have several entries. As we mention before a user have the possibility to publish links, event and articles. The problem that we face here is that some user will be interested in some articles or events according to their profiles. So I decided to create a separate entity call Profile were we keep the attributes that we will use to filter the information to allow users to receive only the information that they are interested according to the profile. The attributes that we are going to filter with will be the home university, language, country, mayor or courses. The articles and events will also have some information associated just like the user but it will not be exactly the same, so I have created another entity called Filter.

User and Message entities have to relations: one that relates the messages and their authors (From field) and another relation that associates the message and its possible destinations.
3. UML Data Model
4. Tables

Next, I’ll show the transformation of the E-R diagram to tables applying the E-R transformation rules.

- USER: login, password, f_name, l_name, email, street, city, state, visibility, about, p_id
- PROFILE: p_id, h_univ, mayor, co_code
- BLOG: b_id, visibility, p_id, u_id
- BLOG_ENTRY: be_id, picture, text, be_date, b_id
- MESSAGE: m_id, subject, m_text, m_date, m_author
- MESSAGE_DEST: m_id, login
- COURSE: c_num, c_name
- ARTICLE: a_id, a_header, a_text, a_url, a_date, a_author, f_id
- EVENT: e_id, e_header, e_text, e_date, e_author, f_id
- LINK: l_id, link, l_date, l_author
- FILTER: f_id, h_univ, mayor, co_code, lan_code, c_num
- PROFILE_COURSE: p_id, c_num
- PROFILE_LANGUAGE: p_id, lan_code
- COUNTRY: co_code, country
- LANGUAGE: lan_code, language

Basically I did the next transformations:

- For each entity I created a table.
- For many-to-many relations I created a new table with the primary keys of the entities involved.
- For one-to-many relations I put the primary key of the one side as foreign key in the many side.
- I have created two new tables Country and Language that were originally attributes but since they will be shared by Filter and Profile is better to have a fixed list of countries or languages.
The visibility attribute in the user marks whether the blog or the user info. can be seen by unregistered users or just registered users. A special value of the p_id will be used to mark events and articles that are sent to everyone including unregistered users. The country attribute in the User entity means the country that the user is from and the country attribute in the Profile means the countries that he is interested in.
5. SQL Code

```sql
create table APP_USER ( login varchar2(50) primary key, password varchar2(80) NOT NULL, f_name varchar2(100) NOT NULL, l_name varchar2(100) NOT NULL, email varchar2(100) NOT NULL, street varchar2(200), city varchar2(100), about varchar2(3999), visibility integer not null, state varchar2(100) , p_id number(38) references PROFILE(p_id ));
create table PROFILE ( p_id number(38) primary key, h_univ varchar2(200), mayor varchar2(300), co_code varchar2(3) references COUNTRY(co_code));
create table COURSE (c_num varchar2(10) primary key, c_name varchar2(100) not null);
create table COUNTRY (co_code varchar2(3) primary key , country varchar2(100) not null );
create table LANGUAGE (lan_code varchar2(3) primary key, language varchar2(100) not null);
create table PROFILE_COURSE ( p_id number(38) references PROFILE(p_id), c_num varchar2(10) references COURSE(c_num));
create table PROFILE_LANGUAGE ( p_id number(38) references PROFILE(p_id), lan_code varchar2(3) references LANGUAGE(lan_code));
create table EVENT (e_id number(38) primary key, e_header varchar2(100) NOT NULL, e_text varchar2(1000), e_date DATE NOT NULL, visibility integer, e_author varchar2(80) REFERENCES APP_USER(login), f_id number(38) references FITLER(f_id));
create table ARTICLE (a_id number(38) primary key, a_header varchar2(100) NOT NULL, a_text varchar2(1000), a_url varchar2(300), a_date DATE NOT NULL, a_author varchar2(80) REFERENCES APP_USER(login), f_id number(38) references FITLER(f_id));
create table FILTER (f_id number(38) primary key, co_code varchar2(3) references COUNTRY(co_code), c_num varchar2(10) references COURSE(c_num), h_univ varchar2 (200), mayor varchar2(300), lan_code varchar2(3) references LANGUAGE(lan_code));
create table LINK (l_id number(38) primary key, link varchar2(400) NOT NULL, l_date DATE NOT NULL, l_author varchar2(80) REFERENCES APP_USER(login));
create table MESSAGE ( m_id  number(38) primary key, subject varchar2(200) not null, m_text varchar2(2000), m_date DATE not null, m_author varchar2(100) REFERENCES APP_USER(login) not null );
create table MESSAGE_DEST ( m_id  number(38) references MESSAGE(m_id), u_id varchar2(50) references APP_USER(login));
create table BLOG ( b_id  number(38) primary key, p_id number(38) references PROFILE(p_id), visibility integer not null, u_id number(38) references APP_USER(login));
create table BLOG_ENTRY ( be_id  number(38) primary key, be_id  number(38) references BLOG(b_id), picture varchar2(300) not null, text varchar2(3999), be_date DATE not null, b_id number(38) references BLOG(b_id));
```