Database for a simple forum website

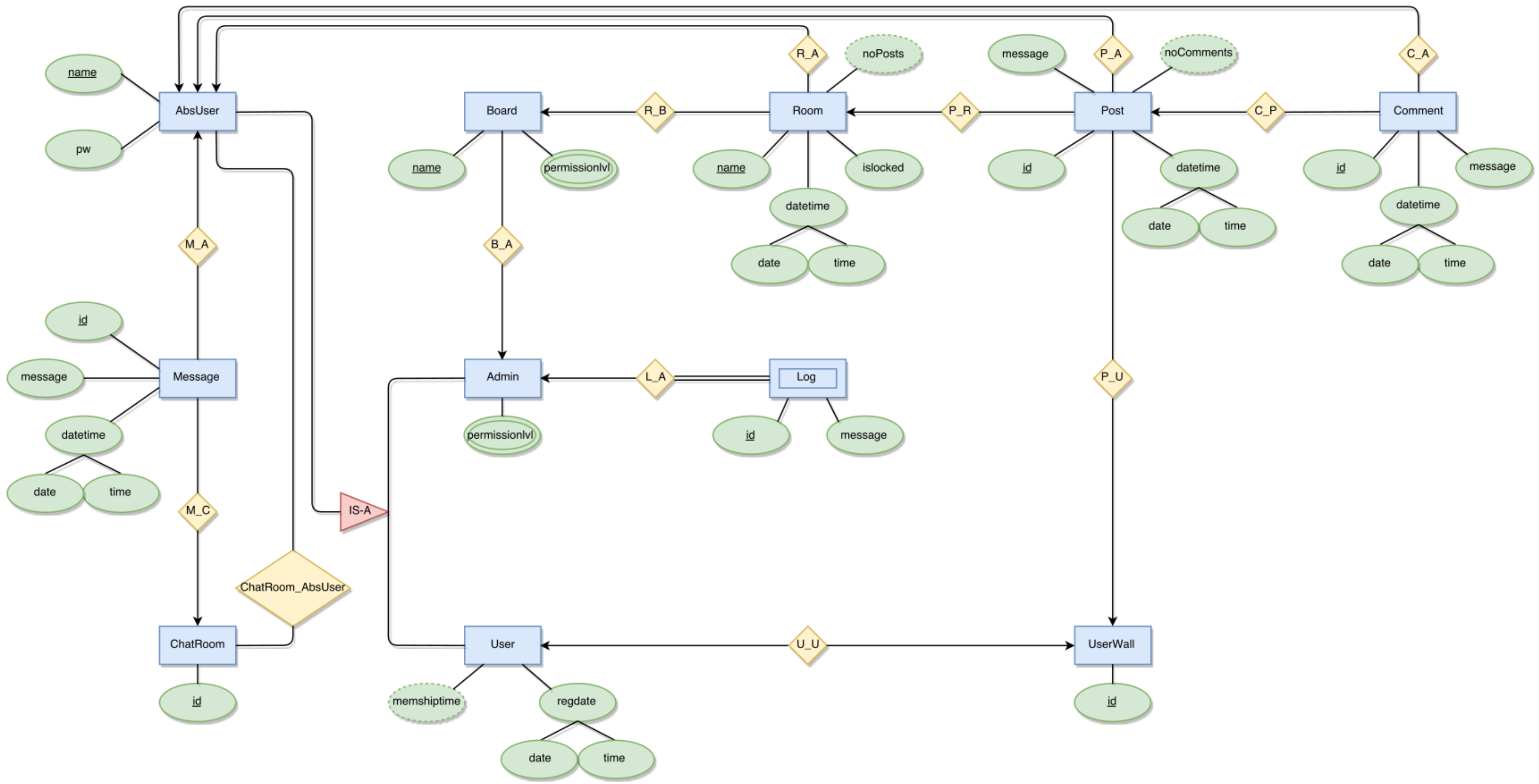
- Project by: Váraljai Péter
- Project Supervisor: Gábor Csaba

Introduction

I would like to create a small database to show examples of all types of entities and relationships. I will store the data of a small forum, including useres, admins, their messages, post, and comments. (ect)

Description

- **Board**: Figurarly this is the "main menu" every **Room** is listed under different **Board**. It can be created or modified by any **Admin** who has a right permission level.
- **Room**: **Rooms** are attached to **Boards**. Every **Room** contain different **Post** from different **Users**. It can be created by any **User**. It has creation time, and it can be locked by any Admin.
- **Post**: **Post** are attached to **Rooms** or **UserWalls**. Users can **Comment** any **Posts** (except if the **Room** is locked). It has creation time, belongs to an **User**, and contains a message.
- **Comment**: Every **User** can **Comment** their/other **Users Post**. It has creation time, belongs to an **User**, and contains a message.
- **ChatRoom**: **Chatrooms** are automaticaly created rooms where **Users** and **Admins** can send **Messages** to each other. Multiple **Users** can be in the same **ChatRoom**, and **Users** can be in multiple **ChatRooms**.
- **Message**: Every **User** can send **Messages** to other **Users** are in the same **ChatRoom**. It has creation time, belongs to a **User**, and contains a message.
- **AbsUser**: This is the abstract user. It stores username and hashed and encrypted password.
- **User**: This is an ISA subclass of **AbsUser**. It has registration date and **UserWall**.
- **Admin**: This is an ISA subclass of **AbsUser**. It has permission level, and a **Logger**.
- **UserWall**: This is belongs to a **User**. Other **Users** can **Post** their opinions here.
- **Log**: This is a weak table. It is belongs to an **Admin** and cointains every modifications an **Admin** done.


name
pw
AbsUser
Board
name
permissionlvl
R_B
Room
R_A
noPosts
name
islocked
datetime
date
time
P_R
Post
message
P_A
noComments
id
datetime
date
time
C_P
Comment
C_A
id
message
datetime
date
time
M_A
B_A
id
Message
message
datetime
date
time
M_C
IS-A
Admin
permissionlvl
L_A
Log
id
message
P_U
ChatRoom_AbsUser
ChatRoom
id
User
memshiptime
regdate
date
time
U_U
UserWall
id

Relational model

- **Board**(name, permissionlvl, rAdmin)
- **Room**(name, islocked, date, time, rBoard, rAdmin)
- **Post**(id, message, date, time, rRoom, rUserWall, rAbsUser)
- **Comment**(id, message, date, time, rPost, rAbsUser)
- **ChatRoom**(id)
- **ChatRoom_AbsUser**(rChatRoom,rAbsUser)
- **Message**(id, message, date, time, rChatroom, rAbsUser)
- **AbsUser**(name, pw)
- **User**(iAbsUser, regdate, regtime)
- **Admin**(iAbsUser, permissionlvl)
- **UserWall**(rUser)
- **Log**(rAdmin,message)

Querry and View
example

- Melyek azok a Postok melyek alatt a 2 legfőbb Admin is Commentelt?
- What are the ids of the Posts, where the 2 Big Bosses are both Commented?

```sql
create view TheBigBosses as
    select *
    from
    (
        select Admin.iAbsUser as username
        from Admin
        order by Admin.permissionlevel desc
    )
    where rownum <=2
;

create view PnC as
    select Post.id as pid, Comment2.id as cid, Comment2.rAbsUser as username
    from Post, Comment2
    where Comment2.rPost = Post.id
;

select PnC.pid
from PnC, TheBigBosses
where PnC.username = TheBigBosses.username
group by PnC.pid
having  min(PnC.username) = ( select min(TheBigBosses.username) from TheBigBosses )
    and max(PnC.username) = ( select max(TheBigBosses.username) from TheBigBosses )
;

drop view TheBigBosses;
drop view PnC;
```

Output:

```
       PID
----------
 453140951
 549571820
 946073490
 964047727
1352258535
  56601904
  40506796
1267821697
1216134633
  64251270
 579701661
 858015113
 380910673
 831785068
1108668441
```

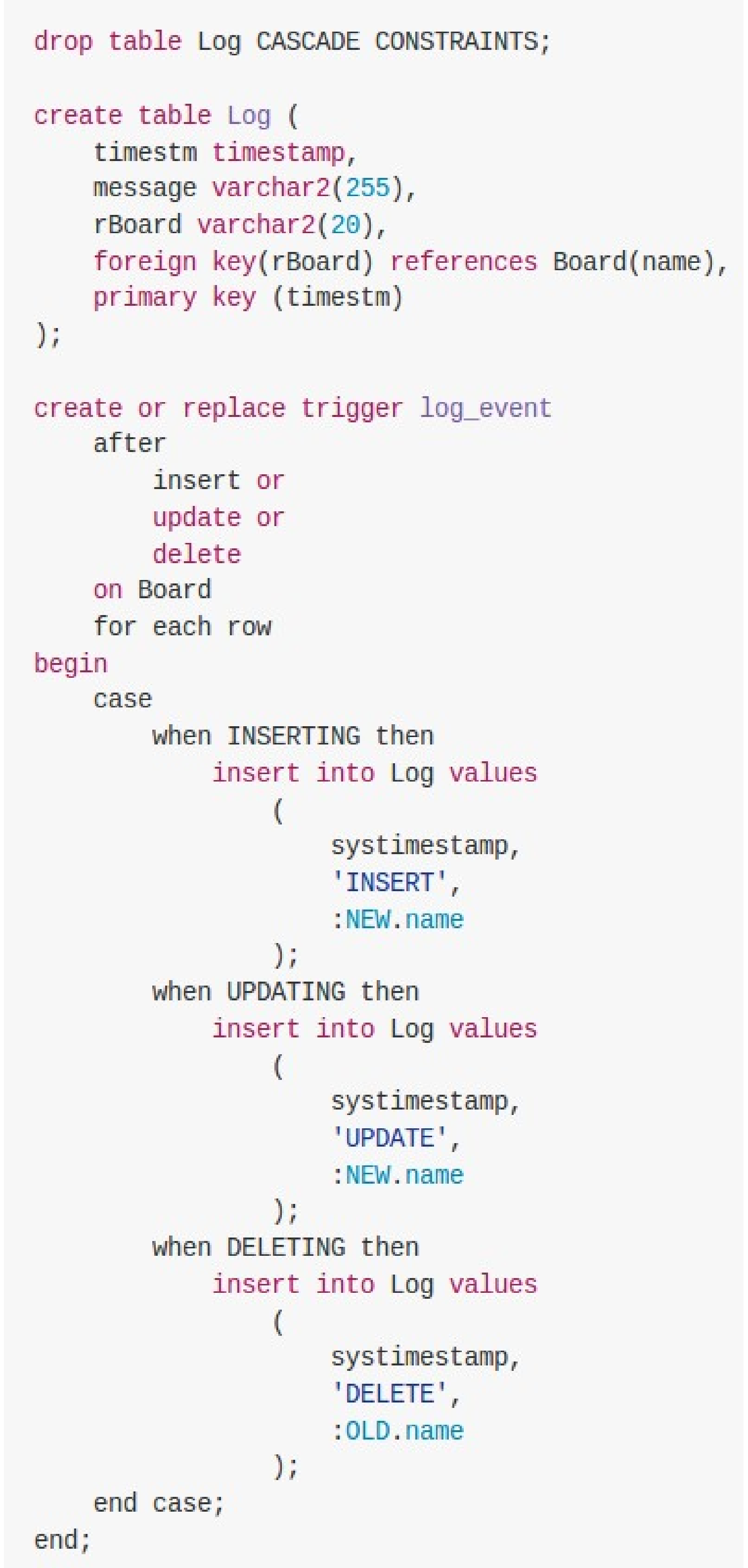
Relational algebra:

```
TheBigBosses: σ rownum() > 0 and rownum()
≤ 2 ρ t1 ( τ Admin.permissionlevel desc ρ
username←Admin.iAbsUser π Admin.iAbsUser,
permissionlevel Admin)

SnC: ρ pid←Post.id, cid←Comment2.id,
username←Comment2.rAbsUser π Post.id,
Comment2.id, Comment2.rAbsUser σ
Comment2.rPost = Post.id Post × Comment2

π username (SnC ÷ TheBigBosses)
```

Trigger



```sql
drop table Log CASCADE CONSTRAINTS;

create table Log (
    timestm timestamp,
    message varchar2(255),
    rBoard varchar2(20),
    foreign key(rBoard) references Board(name),
    primary key (timestm)
);

create or replace trigger log_event
    after
        insert or
        update or
        delete
    on Board
    for each row
begin
    case
        when INSERTING then
            insert into Log values
                (
                    systimestamp,
                    'INSERT',
                    :NEW.name
                );
        when UPDATING then
            insert into Log values
                (
                    systimestamp,
                    'UPDATE',
                    :NEW.name
                );
        when DELETING then
            insert into Log values
                (
                    systimestamp,
                    'DELETE',
                    :OLD.name
                );
    end case;
end;
```