Table: User, Solution, Problem, Tag

The problem-solution web page is called ‘CodeQnA’.

Programmers are the clients of my web page. When they have an issue or a problem, they can post their doubts on the page. They can also answer freely when they know the solution to the problem. Users can search the posting by tag. Tag is the key point of the posting so that other users can easily search.

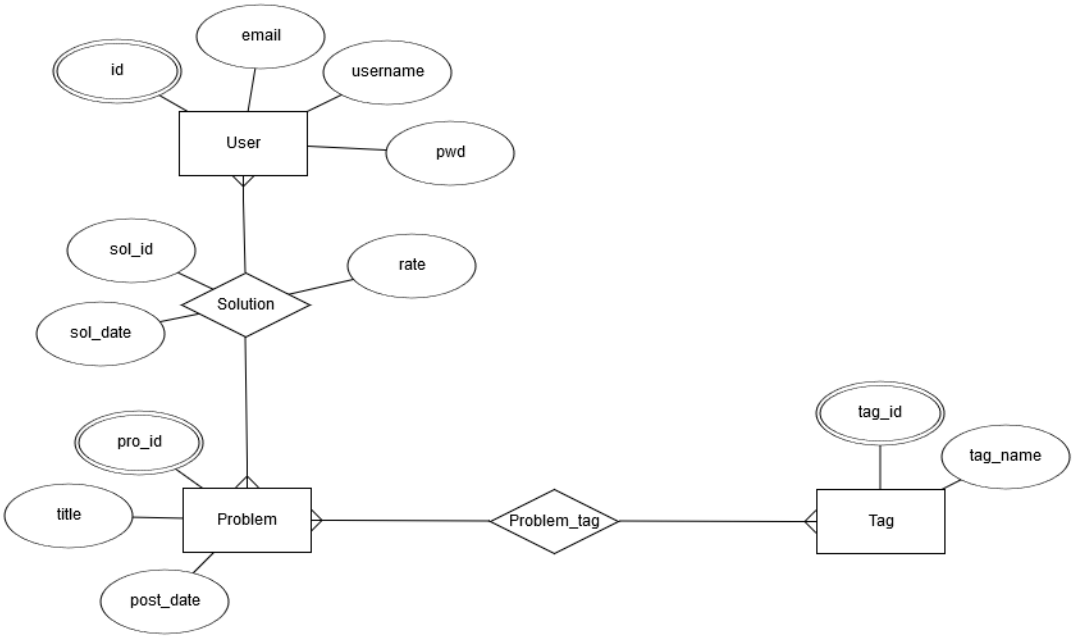


Table USER_c has the information of the user on the web page. They are the client of the developer. They have the user_id, email, user name, and password. The username is used to identify themselves by nickname. We can know by username who uploaded the posting. Email and password are used to log in. user_id is a primary key to identify users and it increments automatically.

Users can post their problems they have met while coding, and also give a solution to the problem. Users can answer below Problems posted, and every user can access to read and give a rating to the problem.

Table PROBLEM has information about Problem. Users upload their problems to be solved on the web page. Each problem has an id called pro_id which is unique. And they have the title, and the date it was posted.

Each problem can have many solutions, because users have different approaches to the problem.

Table SOLUTION has information about the Solution to the posted problem. Users can post the solution below problem or rate the solution implemented.

When users think the solution is the best answer to the problem, they can rate it by giving up to 5 points for good answer, and 1 points for bad answer.

Table TAG consists of tag_id, and tag_name. The tag is for users to search for the posting. tag_name is a programming language used in the problem. This helps users to understand the format of the code explained. We can search for the solution by selecting programming languages that we are considering. We can select tag from 'C', 'C++', 'Python', 'JAVA', 'JAVASCRIPT', 'Perl', 'Ruby', 'C#'.

Table PROBLEM_TAG makes connection with the problem and the tag. Tag is used to search posted problem.

```
CREATE TABLE User_c(
User_id NUMBER(4) CONSTRAINT pk_userid PRIMARY KEY,
email VARCHAR(30) CONSTRAINT nn_user_email NOT NULL,
username VARCHAR (20) CONSTRAINT fk_user_username NOT NULL,
pwd VARCHAR (20) CONSTRAINT nn_user_pwd NOT NULL
);

CREATE TABLE Problem(
pro_id VARCHAR (4) CONSTRAINT pk_problem PRIMARY KEY,
title VARCHAR(30) CONSTRAINT nn_problem_title NOT NULL,
post_date DATE
);

CREATE TABLE Solution(
sol_id NUMBER (4),
user_id NUMBER(4) CONSTRAINT fk_solution_id references USER_c,
pro_id VARCHAR (4) CONSTRAINT fk_solution references PROBLEM,
sol_date DATE,
rate NUMBER (4) CONSTRAINT ck_problem_rate CHECK (rate BETWEEN 0 and 5)
```

```
);

CREATE TABLE Tag(

tag_id VARCHAR (4) CONSTRAINT pk_tag_tagid PRIMARY KEY,

tag_name VARCHAR (20) CONSTRAINT ck_Tag_tagname CHECK (tag_name IN
('C', 'C++', 'Python', 'Java', 'JavaScript', 'Perl', 'Ruby', 'C#'))

);

CREATE TABLE Problem_tag(

pro_id VARCHAR(4) CONSTRAINT fk_pt_proid references PROBLEM,

tag_id VARCHAR (4) CONSTRAINT fk_pt_tagid references TAG

);


INSERT INTO User_c (user_id, email, username, pwd) values (1,
'jkb00001@red.ujaen.es', 'Bea', '12345');

INSERT INTO User_c (user_id, email, username, pwd) values (2,
'he@red.ujaen.es', 'Hen', '222');

INSERT INTO User_c (user_id, email, username, pwd) values (3,
'esma@red.ujaen.es', 'Esma', 'nurk');

INSERT INTO User_c (user_id, email, username, pwd) values (4,
'byul@red.ujaen.es', 'ByulHee', 'em23');

INSERT INTO User_c (user_id, email, username, pwd) values (5,
'love@red.ujaen.es', 'Emily', '1222345');

INSERT INTO User_c (user_id, email, username, pwd) values (6,
'meeeme@red.ujaen.es', 'Noelia', '212345');


INSERT INTO Problem (pro_id, title, post_date) values ('A001', 'Please help
me','02-MAY-2019');

INSERT INTO Problem (pro_id, title, post_date) values ('A002', 'I have a 404
error','22-ABR-2019');

INSERT INTO Problem (pro_id, title, post_date) values ('A003', 'What is
this?','07-MAY-2019');
```

```
INSERT INTO Problem (pro_id, title, post_date) values ('A004', 'Help','27-ABR-2019');


INSERT INTO Solution (sol_id, user_id, pro_id, sol_date, rate) values ('111', '2', 'A001', '04-MAY-2019','0');

INSERT INTO Solution (sol_id, user_id, pro_id, sol_date, rate) values ('222', '1', 'A002', '07-MAY-2019','0');

INSERT INTO Solution (sol_id, user_id, pro_id, sol_date, rate) values ('111', '3', 'A002', '28-ABR-2019','0');

INSERT INTO Solution (sol_id, user_id, pro_id, sol_date, rate) values ('111', '3', 'A003', '08-MAY-2019','0');


INSERT INTO Tag (tag_id, tag_name) values ('T1', 'C++');

INSERT INTO Tag (tag_id, tag_name) values ('T2', 'JavaScript');

INSERT INTO Tag (tag_id, tag_name) values ('T3', 'JavaScript');

INSERT INTO Tag (tag_id, tag_name) values ('T4', 'Perl');

INSERT INTO Tag (tag_id, tag_name) values ('T5', 'C');


INSERT INTO Problem_tag (pro_id, tag_id) values ('A001', 'T1');

INSERT INTO Problem_tag (pro_id, tag_id) values ('A002', 'T2');

INSERT INTO Problem_tag (pro_id, tag_id) values ('A003', 'T3');


CREATE VIEW tag_namelist

AS SELECT tag_name FROM Tag;


SELECT u.username, p.post_date from User_c u, problem p, solution s where u.user_id=s.user_id and p.pro_id=s.pro_id order by post_date desc;

select distinct post_date from Problem order by post_date;

select p.title, count(p.pro_id) as num_solution from problem p, solution s where p.pro_id=s.pro_id group by p.title order by num_solution;
```

```
SELECT u.username, s.sol_date from User_c u, Solution s where
u.user_id=s.user_id;

SELECT u.email, u.username, p.post_date, s.sol_date from User_c u, Problem p,
Solution s where u.user_id=s.user_id and p.pro_id=s.pro_id;

SELECT tag_name, count(tag_name) as num_tag_name from Tag group by
tag_name having count(tag_name)>0;

UPDATE solution SET rate='3' where pro_id='A002' and sol_id='222';

SELECT u.username, sum(s.rate) as total_rate from user_c u, solution s where
u.user_id=s.user_id group by u.username;


INSERT INTO solution (sol_id, user_id, pro_id, sol_date, rate)

(SELECT '555', user_id, pro_id, '08-MAY-2019', 0 FROM solution

  WHERE sol_date='07-MAY-2019');

INSERT INTO Problem_tag (pro_id, tag_id)

 (SELECT pro_id, 'T5' from Problem_tag where pro_id='A001');

UPDATE solution

 SET rate = (select avg(rate) from solution) where rate = 0;
```