A
Project Report
on

Competitive Coding Platform Database

Developed by

PARTH DEDANIYA(IT022) – Department of IT, DD University
JEEL DHAMSANIYA(IT026) - Department of IT, DD University

Guided By
Internal Guide:
Prof. Mukesh M Goswami
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
October-2021

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled **"Competitive Coding Platform Database"** is a bonafied report of the work carried out by

1) **PARTH DEDANIYA** , Student ID No : **19ITUOS074**
2) **JEEL DHAMSANIYA** , Student ID No : **19ITUOS082**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2021-2022.

Prof. Mukesh M Goswami
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “Competitive Coding Platform Database”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Mukesh M Goswami, who took been interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System.

We would also like to thank all our lecturers.

Finally, we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project. We thank one and all.

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

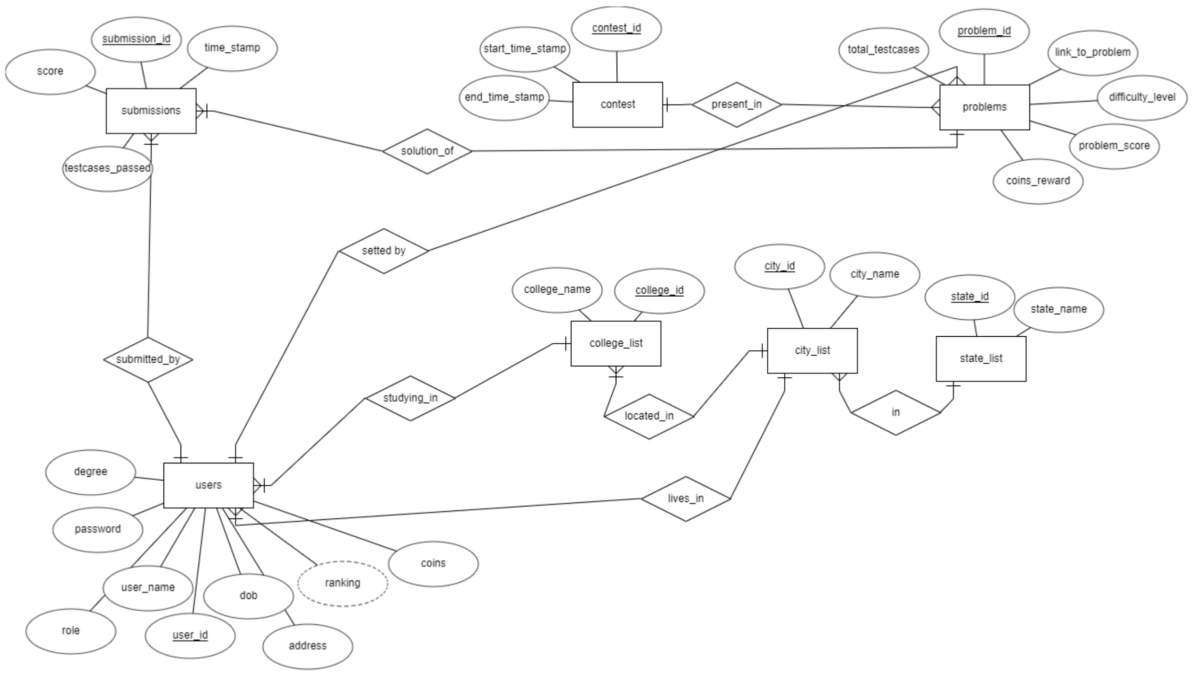
1.1 CURRENT SYSTEM

We have designed a database of Competitive Coding Platform. Currently it becomes very hard to get job in big giants companies and same companies are also in search of good programmers apart of their academic results. By the help of this kind of platform a company can find a programmer with their kind of expectation.

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

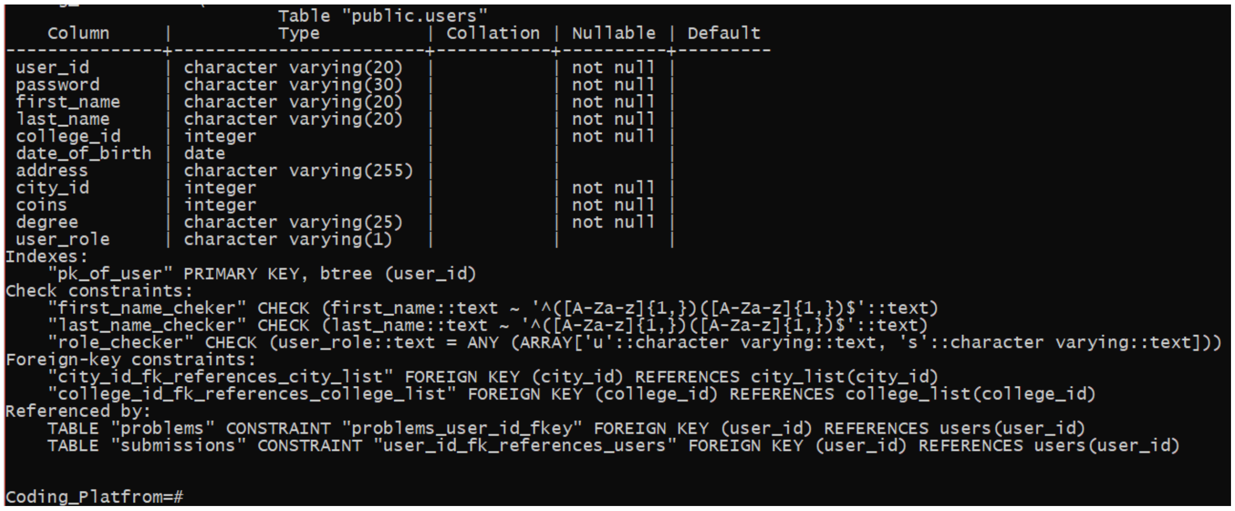
- User can easily participate in various ongoing coding contests.
- They can also submit problems as a problem setter.
- Can submit multiple submissions for getting good score.
- After submission the coin reward is updated accordingly.

2. E-R DIAGRAM


submission_id
time_stamp
score
submissions
testcases_passed
solution_of
contest_id
start_time_stamp
end_time_stamp
contest
present_in
total_testcases
problem_id
link_to_problem
problems
difficulty_level
problem_score
coins_reward
setted by
submitted_by
college_name
college_id
college_list
city_id
city_name
city_list
state_id
state_name
state_list
studying_in
located_in
in
lives_in
degree
users
password
user_name
role
user_id
dob
ranking
coins
address

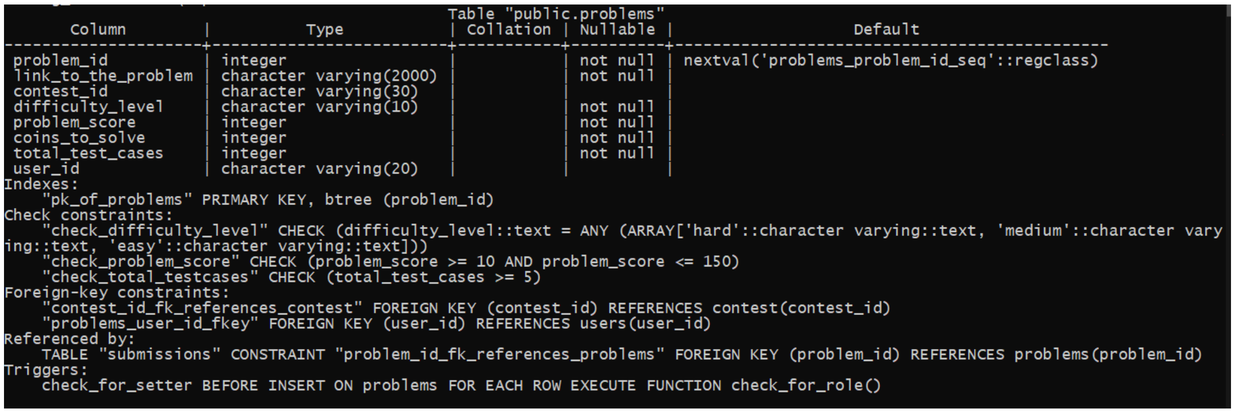
3. DATA DICTIONARY

3.1 Users



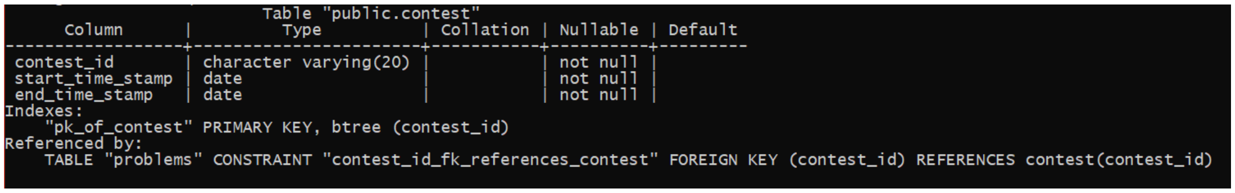
```
                          Table "public.users"
    Column      |          Type          | Collation | Nullable | Default
----------------+------------------------+-----------+----------+---------
 user_id        | character varying(20)  |           | not null |
 password       | character varying(30)  |           | not null |
 first_name     | character varying(20)  |           | not null |
 last_name      | character varying(20)  |           | not null |
 college_id     | integer                |           | not null |
 date_of_birth  | date                   |           |          |
 address        | character varying(255) |           |          |
 city_id        | integer                |           | not null |
 coins          | integer                |           | not null |
 degree         | character varying(25)  |           | not null |
 user_role      | character varying(1)   |           |          |
Indexes:
    "pk_of_user" PRIMARY KEY, btree (user_id)
Check constraints:
    "first_name_cheker" CHECK (first_name::text ~ '^([A-Za-z]{1,})([A-Za-z]{1,})$'::text)
    "last_name_checker" CHECK (last_name::text ~ '^([A-Za-z]{1,})([A-Za-z]{1,})$'::text)
    "role_checker" CHECK (user_role::text = ANY (ARRAY['u'::character varying::text, 's'::character varying::text]))
Foreign-key constraints:
    "city_id_fk_references_city_list" FOREIGN KEY (city_id) REFERENCES city_list(city_id)
    "college_id_fk_references_college_list" FOREIGN KEY (college_id) REFERENCES college_list(college_id)
Referenced by:
    TABLE "problems" CONSTRAINT "problems_user_id_fkey" FOREIGN KEY (user_id) REFERENCES users(user_id)
    TABLE "submissions" CONSTRAINT "user_id_fk_references_users" FOREIGN KEY (user_id) REFERENCES users(user_id)


Coding_Platfrom=#
```

3.2 Problems



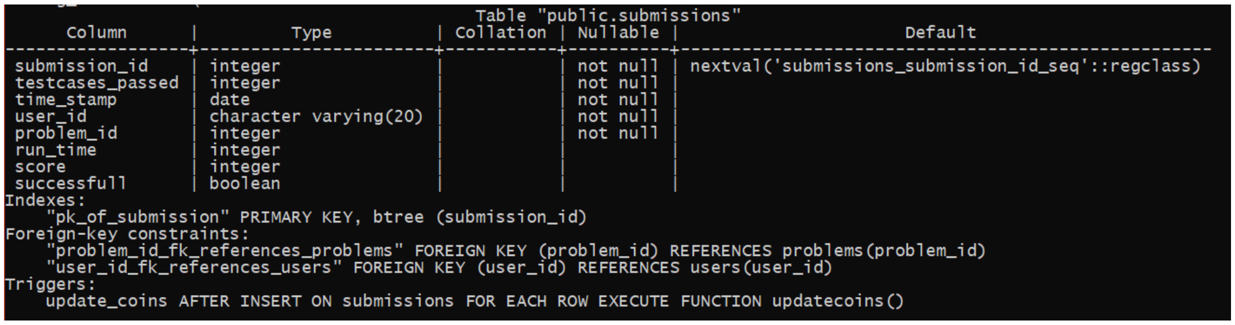
```
                                          Table "public.problems"
       Column        |          Type           | Collation | Nullable |                   Default
---------------------+-------------------------+-----------+----------+----------------------------------------------
 problem_id          | integer                 |           | not null | nextval('problems_problem_id_seq'::regclass)
 link_to_the_problem | character varying(2000) |           | not null |
 contest_id          | character varying(30)   |           |          |
 difficulty_level    | character varying(10)   |           | not null |
 problem_score       | integer                 |           | not null |
 coins_to_solve      | integer                 |           | not null |
 total_test_cases    | integer                 |           | not null |
 user_id             | character varying(20)   |           |          |
Indexes:
    "pk_of_problems" PRIMARY KEY, btree (problem_id)
Check constraints:
    "check_difficulty_level" CHECK (difficulty_level::text = ANY (ARRAY['hard'::character varying::text, 'medium'::character vary
ing::text, 'easy'::character varying::text]))
    "check_problem_score" CHECK (problem_score >= 10 AND problem_score <= 150)
    "check_total_testcases" CHECK (total_test_cases >= 5)
Foreign-key constraints:
    "contest_id_fk_references_contest" FOREIGN KEY (contest_id) REFERENCES contest(contest_id)
    "problems_user_id_fkey" FOREIGN KEY (user_id) REFERENCES users(user_id)
Referenced by:
    TABLE "submissions" CONSTRAINT "problem_id_fk_references_problems" FOREIGN KEY (problem_id) REFERENCES problems(problem_id)
Triggers:
    check_for_setter BEFORE INSERT ON problems FOR EACH ROW EXECUTE FUNCTION check_for_role()
```

3.3 Contests



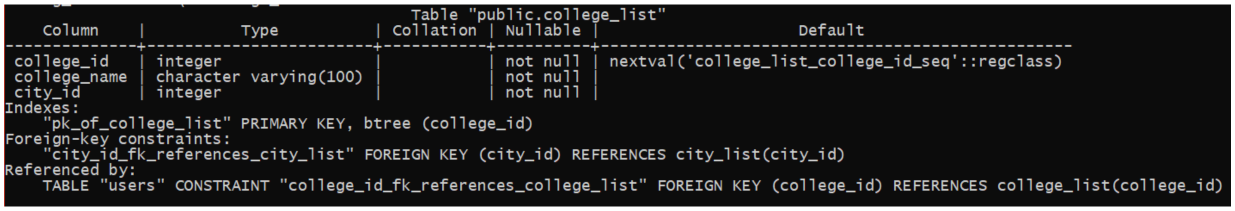
```
                          Table "public.contest"
      Column      |         Type          | Collation | Nullable | Default
------------------+-----------------------+-----------+----------+---------
 contest_id       | character varying(20) |           | not null |
 start_time_stamp | date                  |           | not null |
 end_time_stamp   | date                  |           | not null |
Indexes:
    "pk_of_contest" PRIMARY KEY, btree (contest_id)
Referenced by:
    TABLE "problems" CONSTRAINT "contest_id_fk_references_contest" FOREIGN KEY (contest_id) REFERENCES contest(contest_id)
```

3.4 Submissions



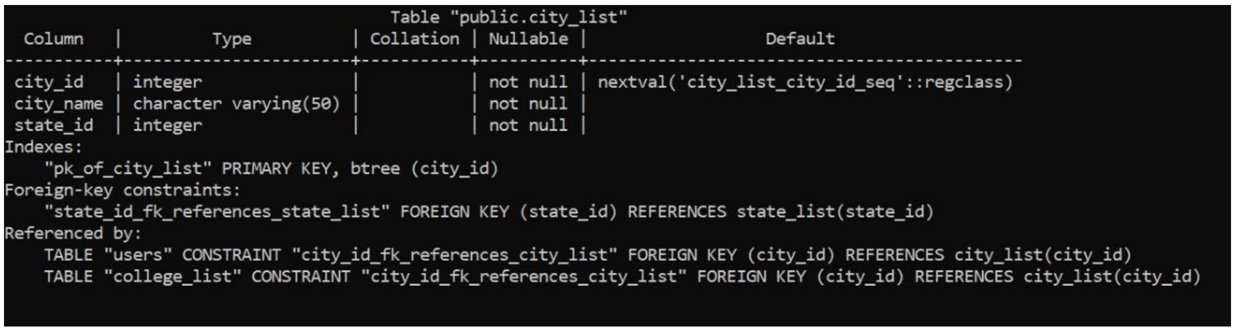
```
                                    Table "public.submissions"
     Column       |         Type          | Collation | Nullable |                        Default
------------------+-----------------------+-----------+----------+--------------------------------------------------------
 submission_id    | integer               |           | not null | nextval('submissions_submission_id_seq'::regclass)
 testcases_passed | integer               |           | not null |
 time_stamp       | date                  |           | not null |
 user_id          | character varying(20) |           | not null |
 problem_id       | integer               |           | not null |
 run_time         | integer               |           |          |
 score            | integer               |           |          |
 successfull      | boolean               |           |          |
Indexes:
    "pk_of_submission" PRIMARY KEY, btree (submission_id)
Foreign-key constraints:
    "problem_id_fk_references_problems" FOREIGN KEY (problem_id) REFERENCES problems(problem_id)
    "user_id_fk_references_users" FOREIGN KEY (user_id) REFERENCES users(user_id)
Triggers:
    update_coins AFTER INSERT ON submissions FOR EACH ROW EXECUTE FUNCTION updatecoins()
```

3.5 College_list



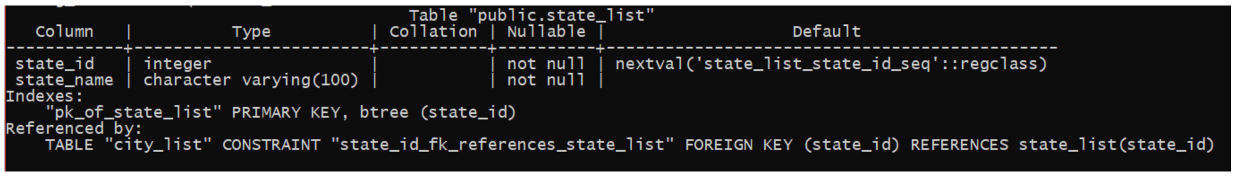
```
                                   Table "public.college_list"
    Column    |          Type          | Collation | Nullable |                    Default
--------------+------------------------+-----------+----------+-----------------------------------------------
 college_id   | integer                |           | not null | nextval('college_list_college_id_seq'::regclass)
 college_name | character varying(100) |           | not null |
 city_id      | integer                |           | not null |
Indexes:
    "pk_of_college_list" PRIMARY KEY, btree (college_id)
Foreign-key constraints:
    "city_id_fk_references_city_list" FOREIGN KEY (city_id) REFERENCES city_list(city_id)
Referenced by:
    TABLE "users" CONSTRAINT "college_id_fk_references_college_list" FOREIGN KEY (college_id) REFERENCES college_list(college_id)
```

3.6 City_list



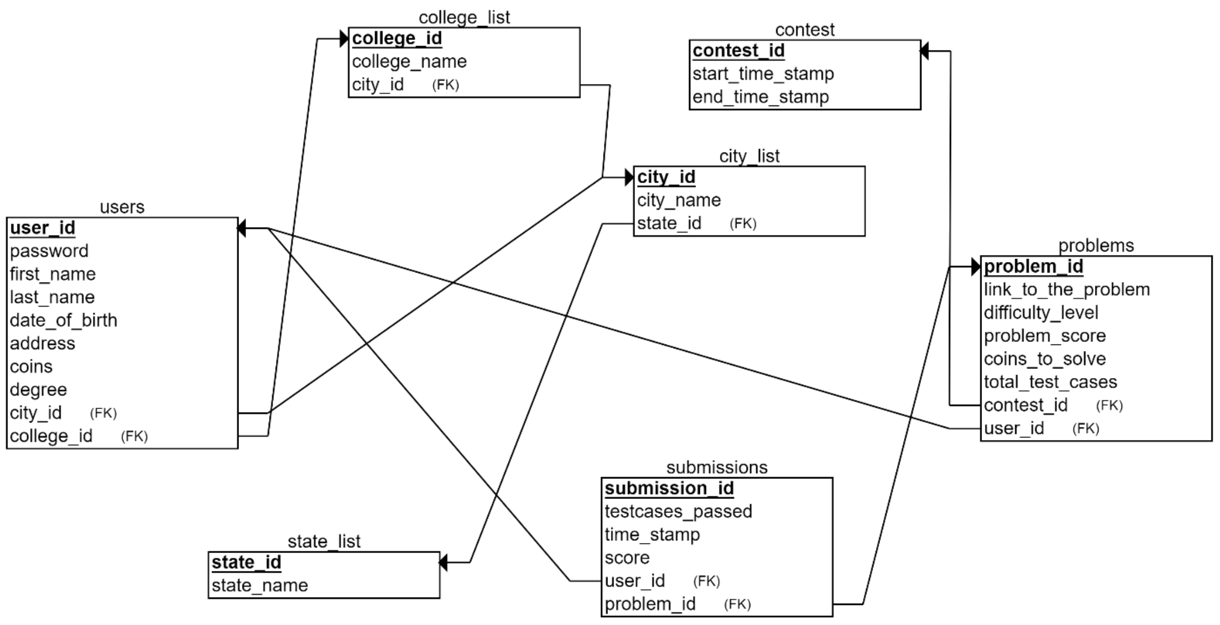
```
                                Table "public.city_list"
  Column   |         Type          | Collation | Nullable |                 Default
-----------+-----------------------+-----------+----------+-----------------------------------------
 city_id   | integer               |           | not null | nextval('city_list_city_id_seq'::regclass)
 city_name | character varying(50) |           | not null |
 state_id  | integer               |           | not null |
Indexes:
    "pk_of_city_list" PRIMARY KEY, btree (city_id)
Foreign-key constraints:
    "state_id_fk_references_state_list" FOREIGN KEY (state_id) REFERENCES state_list(state_id)
Referenced by:
    TABLE "users" CONSTRAINT "city_id_fk_references_city_list" FOREIGN KEY (city_id) REFERENCES city_list(city_id)
    TABLE "college_list" CONSTRAINT "city_id_fk_references_city_list" FOREIGN KEY (city_id) REFERENCES city_list(city_id)
```

3.7 State_list



```
                                   Table "public.state_list"
   Column   |          Type          | Collation | Nullable |                  Default
------------+------------------------+-----------+----------+---------------------------------------------
 state_id   | integer                |           | not null | nextval('state_list_state_id_seq'::regclass)
 state_name | character varying(100) |           | not null |
Indexes:
    "pk_of_state_list" PRIMARY KEY, btree (state_id)
Referenced by:
    TABLE "city_list" CONSTRAINT "state_id_fk_references_state_list" FOREIGN KEY (state_id) REFERENCES state_list(state_id)
```

4. SCHEMA DIAGRAM



5. DATABASE IMPLEMENTATION

5.1 CREATE TABLES

5.1.1 Users

```
CREATE TABLE public.users
(
    user_id character varying(20) NOT NULL,
    password character varying(30) NOT NULL,
    first_name character varying(20) NOT NULL,
    last_name character varying(20) NOT NULL,
    college_id integer NOT NULL,
    date_of_birth date,
    address character varying(255),
    city_id integer NOT NULL,
    coins integer NOT NULL,
    degree character varying(25) NOT NULL,
    user_role character varying(1),
    CONSTRAINT pk_of_user PRIMARY KEY (user_id),
    CONSTRAINT city_id_fk_references_city_list FOREIGN KEY (city_id)
        REFERENCES public.city_list (city_id),
    CONSTRAINT college_id_fk_references_college_list FOREIGN KEY (college_id)
        REFERENCES public.college_list (college_id),
    CONSTRAINT first_name_cheker CHECK (first_name ~ '^([A-Za-z]{1,})([A-Za-
    z]{1,})$'),
    CONSTRAINT last_name_checker CHECK (last_name ~ '^([A-Za-z]{1,})([A-Za-
    z]{1,})$'),
    CONSTRAINT role_checker CHECK (user_role = ANY (ARRAY['u', 's']))
)
```

5.1.2 Contest

```
CREATE TABLE public.contest
(
    contest_id character varying(20) NOT NULL,
    start_time_stamp date NOT NULL,
    end_time_stamp date NOT NULL,
    CONSTRAINT pk_of_contest PRIMARY KEY (contest_id)
)
```

5.1.3 Problems

```
CREATE TABLE public.problems
(
    problem_id integer NOT NULL,
    link_to_the_problem character varying(2000 NOT NULL,
    contest_id character varying(30),
    difficulty_level character varying(10) NOT NULL,
    problem_score integer NOT NULL,
    coins_to_solve integer NOT NULL,
    total_test_cases integer NOT NULL,
    user_id character varying(20),
    CONSTRAINT pk_of_problems PRIMARY KEY (problem_id),
    CONSTRAINT contest_id_fk_references_contest FOREIGN KEY (contest_id)
        REFERENCES public.contest (contest_id),
    CONSTRAINT problems_user_id_fkey FOREIGN KEY (user_id)
        REFERENCES public.users (user_id),
    CONSTRAINT check_difficulty_level CHECK (difficulty_level = ANY
    (ARRAY['hard', 'medium', 'easy'])),
    CONSTRAINT check_problem_score CHECK (problem_score >= 10 AND
    problem_score <= 150),
    CONSTRAINT check_total_testcases CHECK (total_test_cases >= 5)
)
```

5.1.4 Submissions

```
CREATE TABLE public.submissions
(
    submission_id integer NOT NULL,
    testcases_passed integer NOT NULL,
    time_stamp date NOT NULL,
    user_id character varying(20) NOT NULL,
    problem_id integer NOT NULL,
    run_time integer,
    score integer,
    successfull boolean,
    CONSTRAINT pk_of_submission PRIMARY KEY (submission_id),
    CONSTRAINT problem_id_fk_references_problems FOREIGN KEY (problem_id)
        REFERENCES public.problems (problem_id),
    CONSTRAINT user_id_fk_references_users FOREIGN KEY (user_id)
        REFERENCES public.users (user_id)
)
```

5.1.5 College_list

```
CREATE TABLE public.college_list
(
   college_id integer NOT NULL,
   college_name character varying(100) NOT NULL,
   city_id integer NOT NULL,
   CONSTRAINT pk_of_college_list PRIMARY KEY (college_id),
   CONSTRAINT city_id_fk_references_city_list FOREIGN KEY (city_id)
      REFERENCES public.city_list (city_id)
)
```

5.1.6 City_list

```
CREATE TABLE public.city_list
(
   city_id integer NOT NULL,
   city_name character varying(50) NOT NULL,
   state_id integer NOT NULL,
   CONSTRAINT pk_of_city_list PRIMARY KEY (city_id),
   CONSTRAINT state_id_fk_references_state_list FOREIGN KEY
(state_id)
      REFERENCES public.state_list (state_id)
)
```

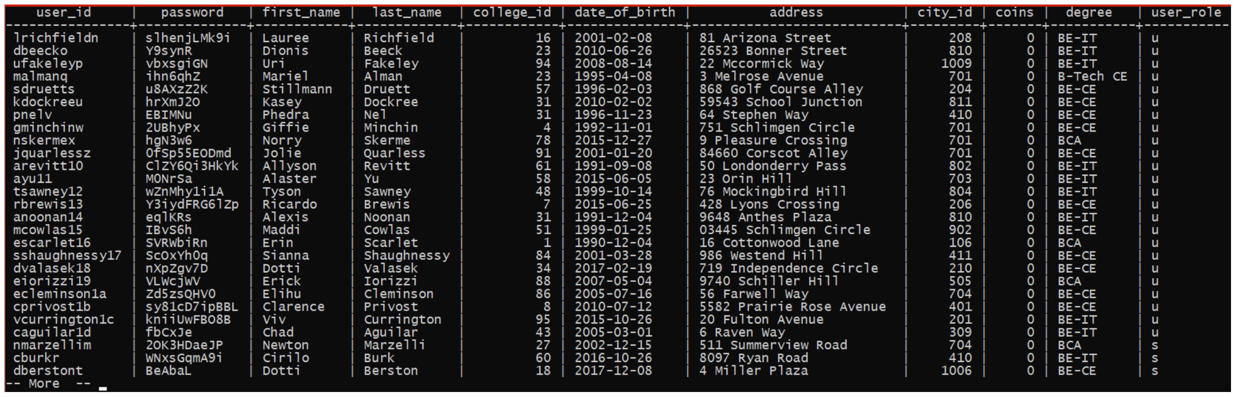
5.1.7 State_list

```
CREATE TABLE public.state_list
(
   state_id integer NOT NULL,
   state_name character varying(100) NOT NULL,
   CONSTRAINT pk_of_state_list PRIMARY KEY (state_id)
)
```

5.2 INSERT DATA VALUE

5.2.1 Users

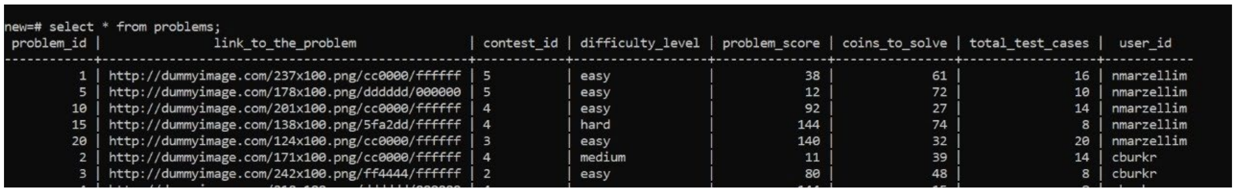
insert into users values (‘lrichfieldn’,’slhenjLMk9i’,’Lauree’,’Richfield’,16,’2001-02-08’,’81 Arizona Street’,208,0,’BE-IT’,’u’);



```
    user_id      |   password    | first_name |  last_name  | college_id | date_of_birth |         address           | city_id | coins |   degree    | user_role
-----------------+---------------+------------+-------------+------------+---------------+---------------------------+---------+-------+-------------+-----------
 lrichfieldn     | slhenjLMk9i   | Lauree     | Richfield   |         16 | 2001-02-08    | 81 Arizona Street         |     208 |     0 | BE-IT       | u
 dbeecko         | Y9synR        | Dionis     | Beeck       |         23 | 2010-06-26    | 26523 Bonner Street       |     810 |     0 | BE-IT       | u
 ufakeleyp       | vbxsgiGN      | Uri        | Fakeley     |         94 | 2008-08-14    | 22 Mccormick Way          |    1009 |     0 | BE-IT       | u
 malmanq         | ihn6qhZ       | Mariel     | Alman       |         23 | 1995-04-08    | 3 Melrose Avenue          |     701 |     0 | B-Tech CE   | u
 sdruetts        | u8AXzZ2K      | Stillmann  | Druett      |         57 | 1996-02-03    | 868 Golf Course Alley     |     204 |     0 | BE-CE       | u
 kdockreeu       | hrXmJ2O       | Kasey      | Dockree     |         31 | 2010-02-02    | 59543 School Junction     |     811 |     0 | BE-CE       | u
 pnelv           | EBIMNu        | Phedra     | Nel         |         31 | 1996-11-23    | 64 Stephen Way            |     410 |     0 | BE-CE       | u
 gminchinw       | 2UBhyPx       | Giffie     | Minchin     |          4 | 1992-11-01    | 751 Schlimgen Circle      |     701 |     0 | BE-CE       | u
 nskermex        | hgN3w6        | Norry      | Skerme      |         78 | 2015-12-27    | 9 Pleasure Crossing       |     701 |     0 | BCA         | u
 jquarlessz      | OfSp55EODmd   | Jolie      | Quarless    |         91 | 2001-01-20    | 84660 Corscot Alley       |     701 |     0 | BE-CE       | u
 arevitt10       | ClZY6Qi3HkYk  | Allyson    | Revitt      |         61 | 1991-09-08    | 50 Londonderry Pass       |     802 |     0 | BE-IT       | u
 ayu11           | MONrSa        | Alaster    | Yu          |         58 | 2015-06-05    | 23 Orin Hill              |     703 |     0 | BE-IT       | u
 tsawney12       | wZnMhy1i1A    | Tyson      | Sawney      |         48 | 1999-10-14    | 76 Mockingbird Hill       |     804 |     0 | BE-IT       | u
 rbrewis13       | Y3iydFRG6lZp  | Ricardo    | Brewis      |          7 | 2015-06-25    | 428 Lyons Crossing        |     206 |     0 | BE-CE       | u
 anoonan14       | eqlKRs        | Alexis     | Noonan      |         31 | 1991-12-04    | 9648 Anthes Plaza         |     810 |     0 | BE-IT       | u
 mcowlas15       | IBvS6h        | Maddi      | Cowlas      |         51 | 1999-01-25    | 03445 Schlimgen Circle    |     902 |     0 | BE-CE       | u
 escarlet16      | SVRWbiRn      | Erin       | Scarlet     |          1 | 1990-12-04    | 16 Cottonwood Lane        |     106 |     0 | BCA         | u
 sshaughnessy17  | ScOxYhOq      | Sianna     | Shaughnessy |         84 | 2001-03-28    | 986 Westend Hill          |     411 |     0 | BE-CE       | u
 dvalasek18      | nXpZgv7D      | Dotti      | Valasek     |         34 | 2017-02-19    | 719 Independence Circle   |     210 |     0 | BE-CE       | u
 eiorizzi19      | VLWcjWV       | Erick      | Iorizzi     |         88 | 2007-05-04    | 9740 Schiller Hill        |     505 |     0 | BCA         | u
 ecleminson1a    | Zd5zsQHVO     | Elihu      | Cleminson   |         86 | 2005-07-16    | 56 Farwell Way            |     704 |     0 | BE-CE       | u
 cprivost1b      | sy81cD7ipBBL  | Clarence   | Privost     |          8 | 2010-07-12    | 5582 Prairie Rose Avenue  |     401 |     0 | BE-CE       | u
 vcurrington1c   | kniiUwFBO8B   | Viv        | Currington  |         95 | 2015-10-26    | 20 Fulton Avenue          |     201 |     0 | BE-IT       | u
 caguilar1d      | fbCxJe        | Chad       | Aguilar     |         43 | 2005-03-01    | 6 Raven Way               |     309 |     0 | BE-IT       | u
 nmarzellim      | 2OK3HDaeJP    | Newton     | Marzelli    |         27 | 2002-12-15    | 511 Summerview Road       |     704 |     0 | BCA         | s
 cburkr          | WNxsGqmA9i    | Cirilo     | Burk        |         60 | 2016-10-26    | 8097 Ryan Road            |     410 |     0 | BE-IT       | s
 dberstont       | BeAbaL        | Dotti      | Berston     |         18 | 2017-12-08    | 4 Miller Plaza            |    1006 |     0 | BE-CE       | s
-- More  --
```

5.2.2 Problems

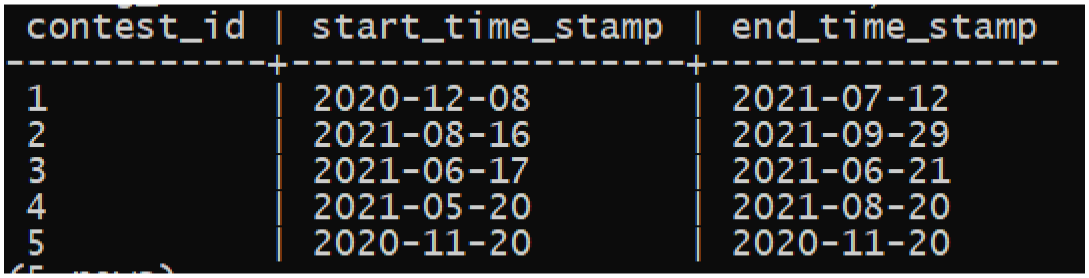
insert into problems values (1, 'http://dummyimage.com/237x100.png/cc0000/ffffff', 5, 'easy', 50, 30, 10, 'nmarzellim');



```
new=# select * from problems;
 problem_id |               link_to_the_problem               | contest_id | difficulty_level | problem_score | coins_to_solve | total_test_cases |  user_id
------------+--------------------------------------------------+------------+------------------+---------------+----------------+------------------+-----------
          1 | http://dummyimage.com/237x100.png/cc0000/ffffff | 5          | easy             |            38 |             61 |               16 | nmarzellim
          5 | http://dummyimage.com/178x100.png/dddddd/000000 | 5          | easy             |            12 |             72 |               10 | nmarzellim
         10 | http://dummyimage.com/201x100.png/cc0000/ffffff | 4          | easy             |            92 |             27 |               14 | nmarzellim
         15 | http://dummyimage.com/138x100.png/5fa2dd/ffffff | 4          | hard             |           144 |             74 |                8 | nmarzellim
         20 | http://dummyimage.com/124x100.png/cc0000/ffffff | 3          | easy             |           140 |             32 |               20 | nmarzellim
          2 | http://dummyimage.com/171x100.png/cc0000/ffffff | 4          | medium           |            11 |             39 |               14 | cburkr
          3 | http://dummyimage.com/242x100.png/ff4444/ffffff | 2          | easy             |            80 |             48 |                8 | cburkr
```

5.2.3 Contest

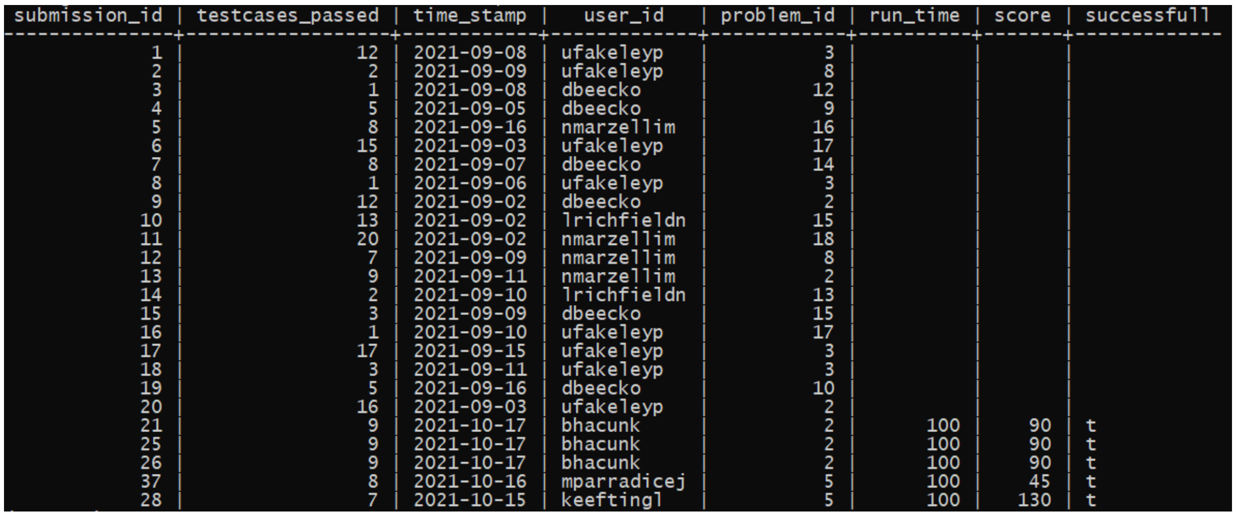
insert into contest values (1,’2020-12-08’,’2021-07-12’0);



```
 contest_id | start_time_stamp | end_time_stamp
------------+------------------+----------------
 1          | 2020-12-08       | 2021-07-12
 2          | 2021-08-16       | 2021-09-29
 3          | 2021-06-17       | 2021-06-21
 4          | 2021-05-20       | 2021-08-20
 5          | 2020-11-20       | 2020-11-20
```

5.2.4 Submissions

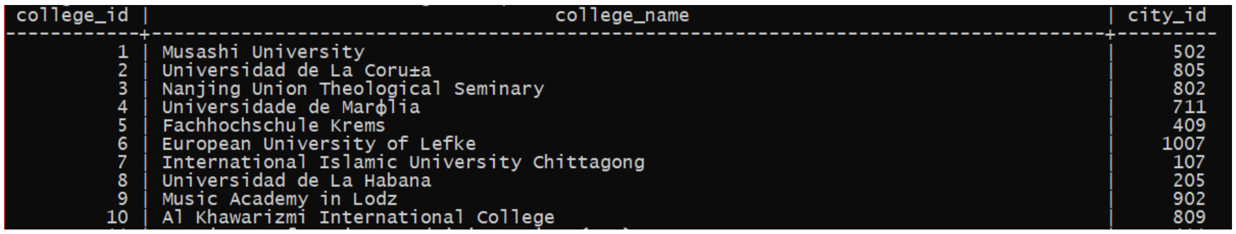
insert into submissions values (28, 7, '2021-10-15', 'keeftingl', 5, 100, 130, 't');



```
 submission_id | testcases_passed | time_stamp |   user_id    | problem_id | run_time | score | successfull
---------------+------------------+------------+--------------+------------+----------+-------+-------------
             1 |               12 | 2021-09-08 | ufakeleyp    |          3 |          |       |
             2 |                2 | 2021-09-09 | ufakeleyp    |          8 |          |       |
             3 |                1 | 2021-09-08 | dbeecko      |         12 |          |       |
             4 |                5 | 2021-09-05 | dbeecko      |          9 |          |       |
             5 |                8 | 2021-09-16 | nmarzellim   |         16 |          |       |
             6 |               15 | 2021-09-03 | ufakeleyp    |         17 |          |       |
             7 |                8 | 2021-09-07 | dbeecko      |         14 |          |       |
             8 |                1 | 2021-09-06 | ufakeleyp    |          3 |          |       |
             9 |               12 | 2021-09-02 | dbeecko      |          2 |          |       |
            10 |               13 | 2021-09-02 | lrichfieldn  |         15 |          |       |
            11 |               20 | 2021-09-02 | nmarzellim   |         18 |          |       |
            12 |                7 | 2021-09-09 | nmarzellim   |          8 |          |       |
            13 |                9 | 2021-09-11 | nmarzellim   |          2 |          |       |
            14 |                2 | 2021-09-10 | lrichfieldn  |         13 |          |       |
            15 |                3 | 2021-09-09 | dbeecko      |         15 |          |       |
            16 |                1 | 2021-09-10 | ufakeleyp    |         17 |          |       |
            17 |               17 | 2021-09-15 | ufakeleyp    |          3 |          |       |
            18 |                3 | 2021-09-11 | ufakeleyp    |          3 |          |       |
            19 |                5 | 2021-09-16 | dbeecko      |         10 |          |       |
            20 |               16 | 2021-09-03 | ufakeleyp    |          2 |          |       |
            21 |                9 | 2021-10-17 | bhacunk      |          2 |      100 |    90 | t
            25 |                9 | 2021-10-17 | bhacunk      |          2 |      100 |    90 | t
            26 |                9 | 2021-10-17 | bhacunk      |          2 |      100 |    90 | t
            37 |                8 | 2021-10-16 | mparradicej  |          5 |      100 |    45 | t
            28 |                7 | 2021-10-15 | keeftingl    |          5 |      100 |   130 | t
```

5.2.5 College_list

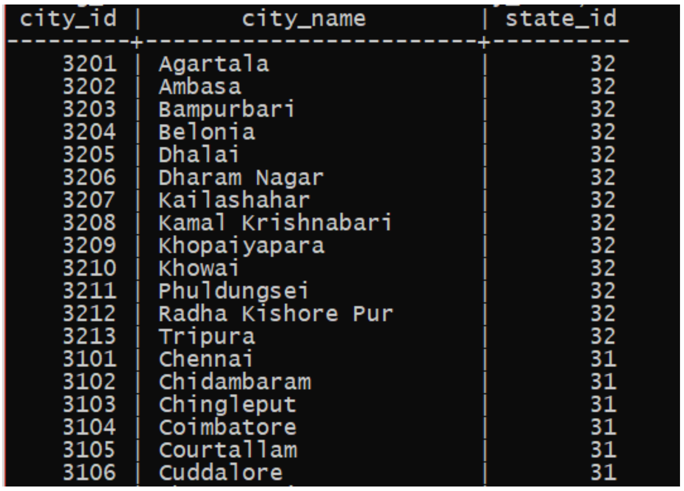
insert into college_list values (1,’Musashi University’,502);



```
 college_id |                                   college_name                                    | city_id
------------+-----------------------------------------------------------------------------------+---------
          1 | Musashi University                                                                |     502
          2 | Universidad de La Coru±a                                                          |     805
          3 | Nanjing Union Theological Seminary                                                |     802
          4 | Universidade de Mar⌀lia                                                           |     711
          5 | Fachhochschule Krems                                                              |     409
          6 | European University of Lefke                                                      |    1007
          7 | International Islamic University Chittagong                                       |     107
          8 | Universidad de La Habana                                                          |     205
          9 | Music Academy in Lodz                                                             |     902
         10 | Al Khawarizmi International College                                               |     809
```

5.2.6 City_list

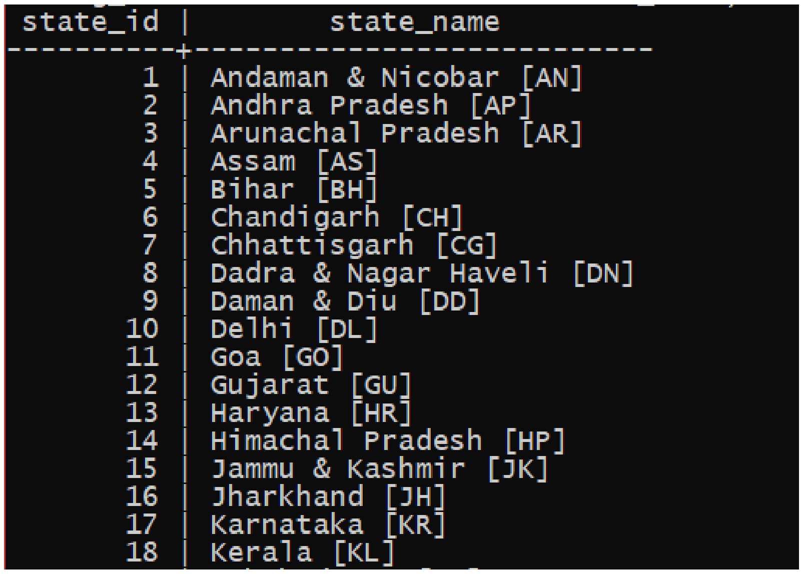
insert into city_list values (3201,’ ‘Agartala’,32);



```
 city_id |       city_name        | state_id
---------+------------------------+----------
    3201 | Agartala               |       32
    3202 | Ambasa                 |       32
    3203 | Bampurbari             |       32
    3204 | Belonia                |       32
    3205 | Dhalai                 |       32
    3206 | Dharam Nagar           |       32
    3207 | Kailashahar            |       32
    3208 | Kamal Krishnabari      |       32
    3209 | Khopaiyapara           |       32
    3210 | Khowai                 |       32
    3211 | Phuldungsei            |       32
    3212 | Radha Kishore Pur      |       32
    3213 | Tripura                |       32
    3101 | Chennai                |       31
    3102 | Chidambaram            |       31
    3103 | Chingleput             |       31
    3104 | Coimbatore             |       31
    3105 | Courtallam             |       31
    3106 | Cuddalore              |       31
```

5.2.7 State_list

insert into values (1,’Andaman & Nicobar [AN]’);

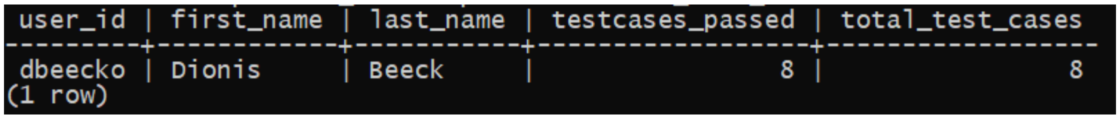


```
 state_id |         state_name
----------+--------------------------
        1 | Andaman & Nicobar [AN]
        2 | Andhra Pradesh [AP]
        3 | Arunachal Pradesh [AR]
        4 | Assam [AS]
        5 | Bihar [BH]
        6 | Chandigarh [CH]
        7 | Chhattisgarh [CG]
        8 | Dadra & Nagar Haveli [DN]
        9 | Daman & Diu [DD]
       10 | Delhi [DL]
       11 | Goa [GO]
       12 | Gujarat [GU]
       13 | Haryana [HR]
       14 | Himachal Pradesh [HP]
       15 | Jammu & Kashmir [JK]
       16 | Jharkhand [JH]
       17 | Karnataka [KR]
       18 | Kerala [KL]
```

5.3 QUERIES

5.3.1 Display the users who have submitted the problem successfully

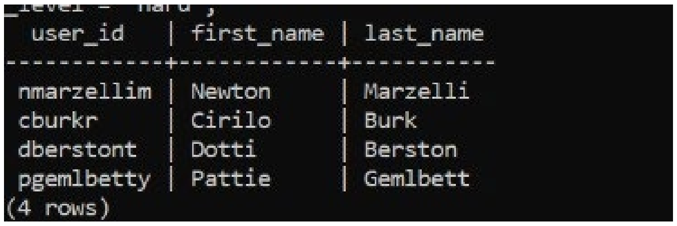
select users.user_id,first_name,last_name,submissions.testcases_passed,problems.total_test_cases from users,problems,submissions where submissions.user_id=users.user_id and problems.problem_id=submissions.problem_id and problems.total_test_cases=submissions.testcases_passed;



```
 user_id | first_name | last_name | testcases_passed | total_test_cases
---------+------------+-----------+------------------+------------------
 dbeecko | Dionis     | Beeck     |                8 |                8
(1 row)
```

5.3.2 Display the problem setter details who have uploaded hard level problems

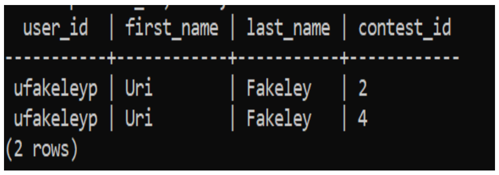
select users.user_id, users.first_name, users.last_name from users, problems where users.user_role = 's' and users.user_id = problems.user_id and problems.difficulty_level = 'hard';



```
  user_id   | first_name | last_name
------------+------------+-----------
 nmarzellim | Newton     | Marzelli
 cburkr     | Cirilo     | Burk
 dberstont  | Dotti      | Berston
 pgemlbetty | Pattie     | Gemlbett
(4 rows)
```

5.3.3 Display the list of contests user = ‘crama0’ has attended.

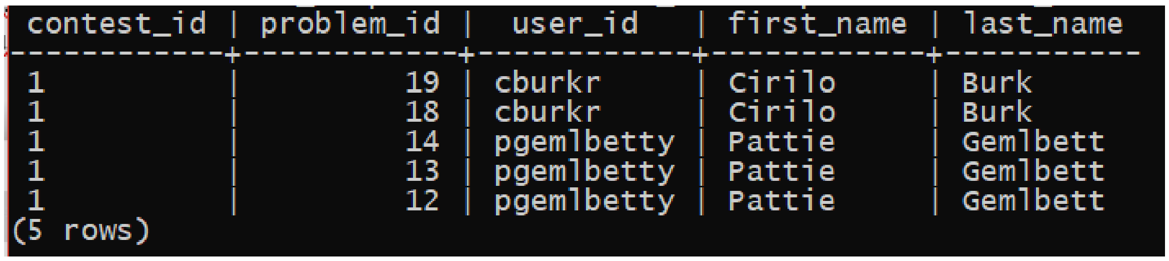
select distinct x.user_id,x.first_name,x.last_name,x,contest_id from ((select table1.user_id,table1.first_name,table1.last_name,submissions.problem_id from((select * from users where user_id=’ufakeleyp’) as table1 inner join submissions on table1.user_id=submissions.user_id)) as table2 inner join problems on table2.problem_id=problems.problem_id) as x;



```
  user_id  | first_name | last_name | contest_id
-----------+------------+-----------+------------
 ufakeleyp | Uri        | Fakeley   | 2
 ufakeleyp | Uri        | Fakeley   | 4
(2 rows)
```

5.3.4 Display the details of contest_id=1(total problems along with it problem setter details)

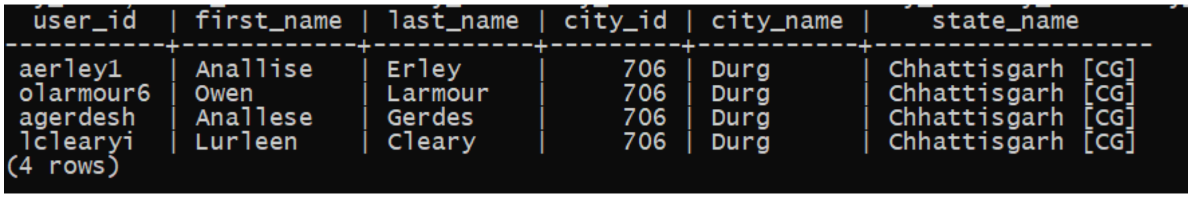
select
contest.contest_id,problems.problem_id,users.user_id,users.first_name,users.last_name
from contest,problems,users where contest.contest_id=problems.contest_id and
problems.user_id=users.user_id and contest.contest_id='1';



```
 contest_id | problem_id |  user_id   | first_name | last_name
------------+------------+------------+------------+-----------
 1          |         19 | cburkr     | Cirilo     | Burk
 1          |         18 | cburkr     | Cirilo     | Burk
 1          |         14 | pgemlbetty | Pattie     | Gemlbett
 1          |         13 | pgemlbetty | Pattie     | Gemlbett
 1          |         12 | pgemlbetty | Pattie     | Gemlbett
(5 rows)
```

5.3.5 Display users details who are living in city having city_id=706

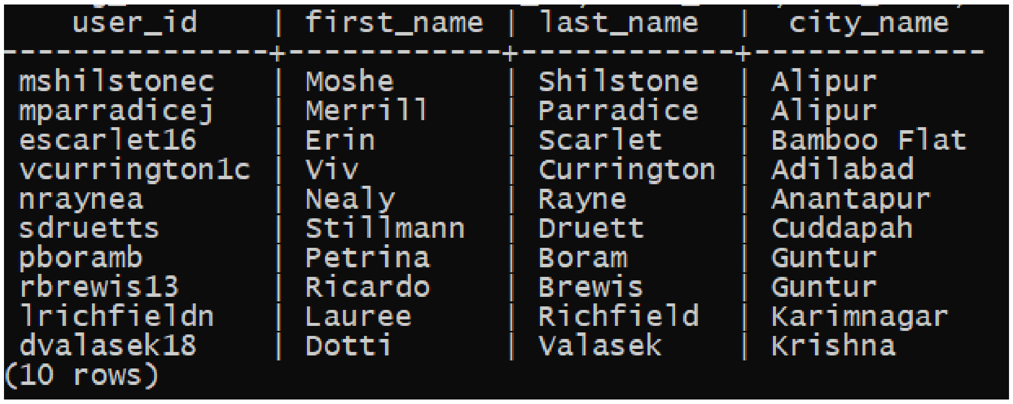
select
users.user_id,users.first_name,users.last_name,city_list.city_id,city_list.city_name,state_l
ist.state_name from users,city_list,state_list where city_list.city_id=706 and
users.city_id=city_list.city_id and city_list.state_id=state_list.state_id;



```
  user_id   | first_name | last_name | city_id | city_name |    state_name
------------+------------+-----------+---------+-----------+-------------------
 aerley1    | Anallise   | Erley     |     706 | Durg      | Chhattisgarh [CG]
 olarmour6  | Owen       | Larmour   |     706 | Durg      | Chhattisgarh [CG]
 agerdesh   | Anallese   | Gerdes    |     706 | Durg      | Chhattisgarh [CG]
 lclearyi   | Lurleen    | Cleary    |     706 | Durg      | Chhattisgarh [CG]
(4 rows)
```

5.3.6 Display 10 user's info along with the city name where he lives.

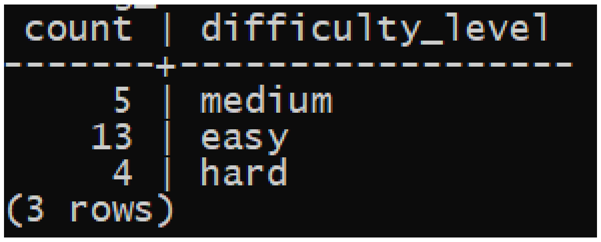
select user_id,first_name,last_name,city_list.city_name from users,city_list where users.city_id=city_list.city_id limit 10;



```
   user_id     | first_name | last_name  |  city_name
---------------+------------+------------+-------------
 mshilstonec   | Moshe      | Shilstone  | Alipur
 mparradicej   | Merrill    | Parradice  | Alipur
 escarlet16    | Erin       | Scarlet    | Bamboo Flat
 vcurrington1c | Viv        | Currington | Adilabad
 nraynea       | Nealy      | Rayne      | Anantapur
 sdruetts      | Stillmann  | Druett     | Cuddapah
 pboramb       | Petrina    | Boram      | Guntur
 rbrewis13     | Ricardo    | Brewis     | Guntur
 lrichfieldn   | Lauree     | Richfield  | Karimnagar
 dvalasek18    | Dotti      | Valasek    | Krishna
(10 rows)
```

5.3.7 Display no. of easy, medium and hard problems.

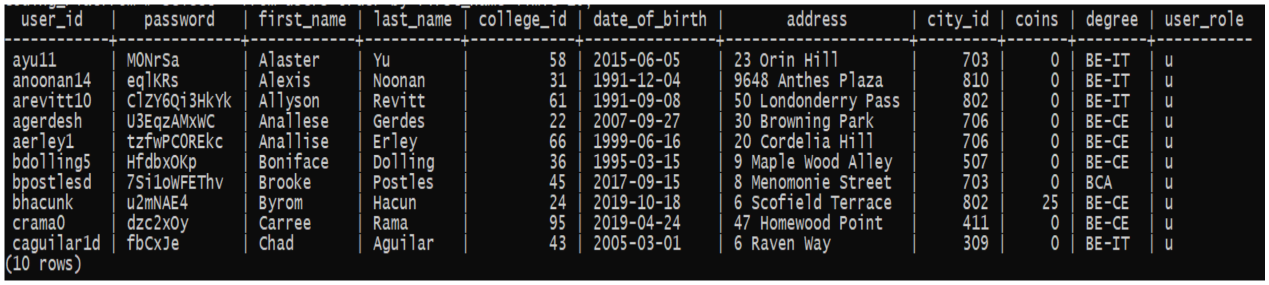
select count(difficulty_level),difficulty_level from problems group by difficulty_level;



```
 count | difficulty_level
-------+------------------
     5 | medium
    13 | easy
     4 | hard
(3 rows)
```

5.3.8 Display the first 10 user info sorted according to first name

Select * from users order by first_name limit 10;



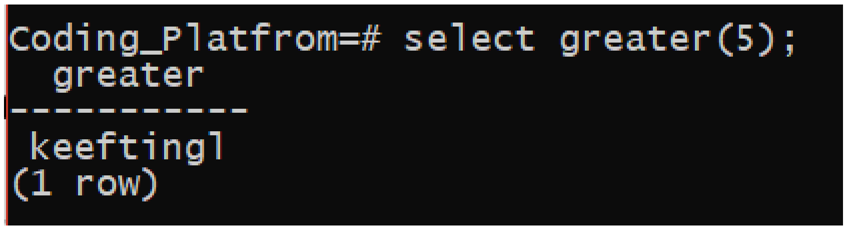
```
  user_id   |   password   | first_name | last_name | college_id | date_of_birth |       address        | city_id | coins | degree | user_role
------------+--------------+------------+-----------+------------+---------------+----------------------+---------+-------+--------+-----------
 ayu11      | MONrSa       | Alaster    | Yu        |         58 | 2015-06-05    | 23 Orin Hill         |     703 |     0 | BE-IT  | u
 anoonan14  | eqlKRs       | Alexis     | Noonan    |         31 | 1991-12-04    | 9648 Anthes Plaza    |     810 |     0 | BE-IT  | u
 arevitt10  | ClZY6Qi3HkYk | Allyson    | Revitt    |         61 | 1991-09-08    | 50 Londonderry Pass  |     802 |     0 | BE-IT  | u
 agerdesh   | U3EqzAMxWC   | Anallese   | Gerdes    |         22 | 2007-09-27    | 30 Browning Park     |     706 |     0 | BE-CE  | u
 aerley1    | tzfwPCOREkc  | Anallise   | Erley     |         66 | 1999-06-16    | 20 Cordelia Hill     |     706 |     0 | BE-CE  | u
 bdolling5  | HfdbxOKp     | Boniface   | Dolling   |         36 | 1995-03-15    | 9 Maple Wood Alley   |     507 |     0 | BE-CE  | u
 bpostlesd  | 7Si1oWFEThv  | Brooke     | Postles   |         45 | 2017-09-15    | 8 Menomonie Street   |     703 |     0 | BCA    | u
 bhacunk    | u2mNAE4      | Byrom      | Hacun     |         24 | 2019-10-18    | 6 Scofield Terrace   |     802 |    25 | BE-CE  | u
 crama0     | dzc2xOy      | Carree     | Rama      |         95 | 2019-04-24    | 47 Homewood Point    |     411 |     0 | BE-CE  | u
 caguilar1d | fbCxJe       | Chad       | Aguilar   |         43 | 2005-03-01    | 6 Raven Way          |     309 |     0 | BE-IT  | u
(10 rows)
```

5.4 FUNCTIONS

5.4.1 Create a function to display user having maximum score for given problem.

-
  ```
  create function greater(pid problems.problem_id%type) returns varchar
  language plpgsql
  as
  $$
  declare
                  uid varchar(10);
                  sc integer;
  begin
        select user_id into uid from submissions where score = (select max(score) from
        submissions where problem_id in (select problem_id from submissions where
        problem_id=pid));
        return uid;
   end;
   $$
  ```

- **Output:**



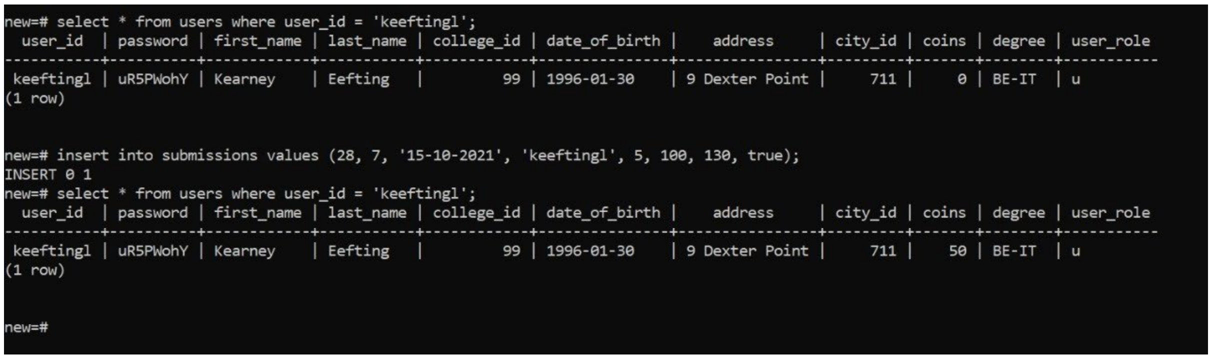
```
Coding_Platfrom=# select greater(5);
  greater
-----------
 keefting1
(1 row)
```

5.5 TRIGGERS

5.5.1 Create a trigger to update user's coin after making a submission.

-
  ```
  create function updateCoins()
  returns trigger
  language plpgsql
  as
  $$
      declare

          testCases int;
          userid varchar(20);
          pid int;

          coins_tc record;
          coins int;
          total_tc int;
          current_coins int;

      begin

          testCases := new.testcases_passed;
          userid := new.user_id;
          pid := new.problem_id;


              select total_test_cases, coins_to_solve into coins_tc from problems where
                  problem_id = pid;
              select coins into current_coins from users where user_id = userid;
              current_coins :=
                  (testCases/coins_tc.total_test_cases::float)*coins_tc.coins_to_solve +
                  current_coins;

              update users set coins = current_coins where user_id = userid;
          return new;
      end;
   $$;

  create trigger update_coins after insert on submissions for each row execute procedure
   updateCoins();
  ```

- **Output:**

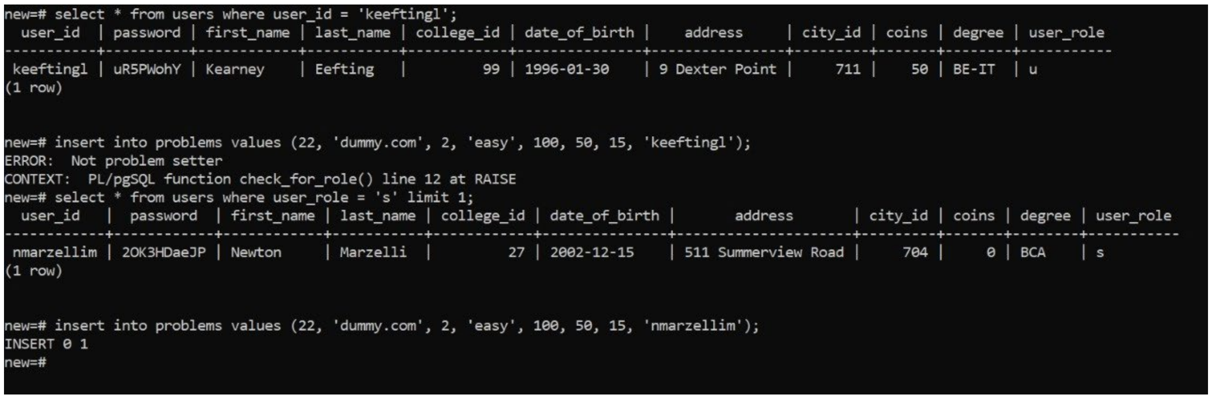


```
new=# select * from users where user_id = 'keeftingl';
  user_id   | password | first_name | last_name | college_id | date_of_birth |     address     | city_id | coins | degree | user_role
------------+----------+------------+-----------+------------+---------------+-----------------+---------+-------+--------+-----------
 keeftingl | uR5PWohY | Kearney    | Eefting   |         99 | 1996-01-30    | 9 Dexter Point |     711 |     0 | BE-IT  | u
(1 row)


new=# insert into submissions values (28, 7, '15-10-2021', 'keeftingl', 5, 100, 130, true);
INSERT 0 1
new=# select * from users where user_id = 'keeftingl';
  user_id   | password | first_name | last_name | college_id | date_of_birth |     address     | city_id | coins | degree | user_role
------------+----------+------------+-----------+------------+---------------+-----------------+---------+-------+--------+-----------
 keeftingl | uR5PWohY | Kearney    | Eefting   |         99 | 1996-01-30    | 9 Dexter Point |     711 |    50 | BE-IT  | u
(1 row)


new=#
```

5.5.2 Create a trigger to check whether user can create a problem or not.

-
  ```
  create or replace function check_for_role()
  returns trigger
  language plpgsql
  as
  $$
  declare
      r varchar(1);
      uid users.user_id%type;
  begin
      uid := new.user_id;
      select user_role into r from users where users.user_id = uid;
      if r='u' then
          raise exception 'Not problem setter';
      end if;
      return new;
  end;
  $$;

  create trigger "check_for_setter" before insert on problems for each row execute
   procedure check_for_role();
  ```

- **Output:**



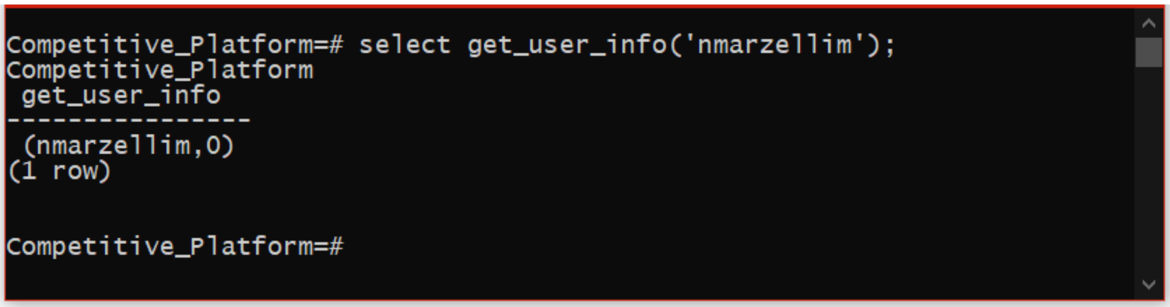
```
new=# select * from users where user_id = 'keeftingl';
  user_id  | password | first_name | last_name | college_id | date_of_birth |    address     | city_id | coins | degree | user_role
-----------+----------+------------+-----------+------------+---------------+----------------+---------+-------+--------+-----------
 keeftingl | uR5PWohY | Kearney    | Eefting   |         99 | 1996-01-30    | 9 Dexter Point |     711 |    50 | BE-IT  | u
(1 row)


new=# insert into problems values (22, 'dummy.com', 2, 'easy', 100, 50, 15, 'keeftingl');
ERROR:  Not problem setter
CONTEXT:  PL/pgSQL function check_for_role() line 12 at RAISE
new=# select * from users where user_role = 's' limit 1;
  user_id   |  password  | first_name | last_name | college_id | date_of_birth |       address       | city_id | coins | degree | user_role
------------+------------+------------+-----------+------------+---------------+---------------------+---------+-------+--------+-----------
 nmarzellim | 2OK3HDaeJP | Newton     | Marzelli  |         27 | 2002-12-15    | 511 Summerview Road |     704 |     0 | BCA    | s
(1 row)


new=# insert into problems values (22, 'dummy.com', 2, 'easy', 100, 50, 15, 'nmarzellim');
INSERT 0 1
new=#
```

5.6 CURSORS

5.6.1 Create a cursor to display value of coins for a given user.

-
  ```
  create or replace function get_user_info(uid users.user_id%type) returns record as $$
  declare
      d record;
      cur cursor for select user_id,coins from users;

  begin
      open cur;
      fetch next from cur into d;
      loop
      fetch cur into d;
          exit when d.user_id=uid;
          exit when not found;
      end loop;
      close cur;
      return d;
  end;
  $$
  language plpgsql;
  ```

- **Output:**

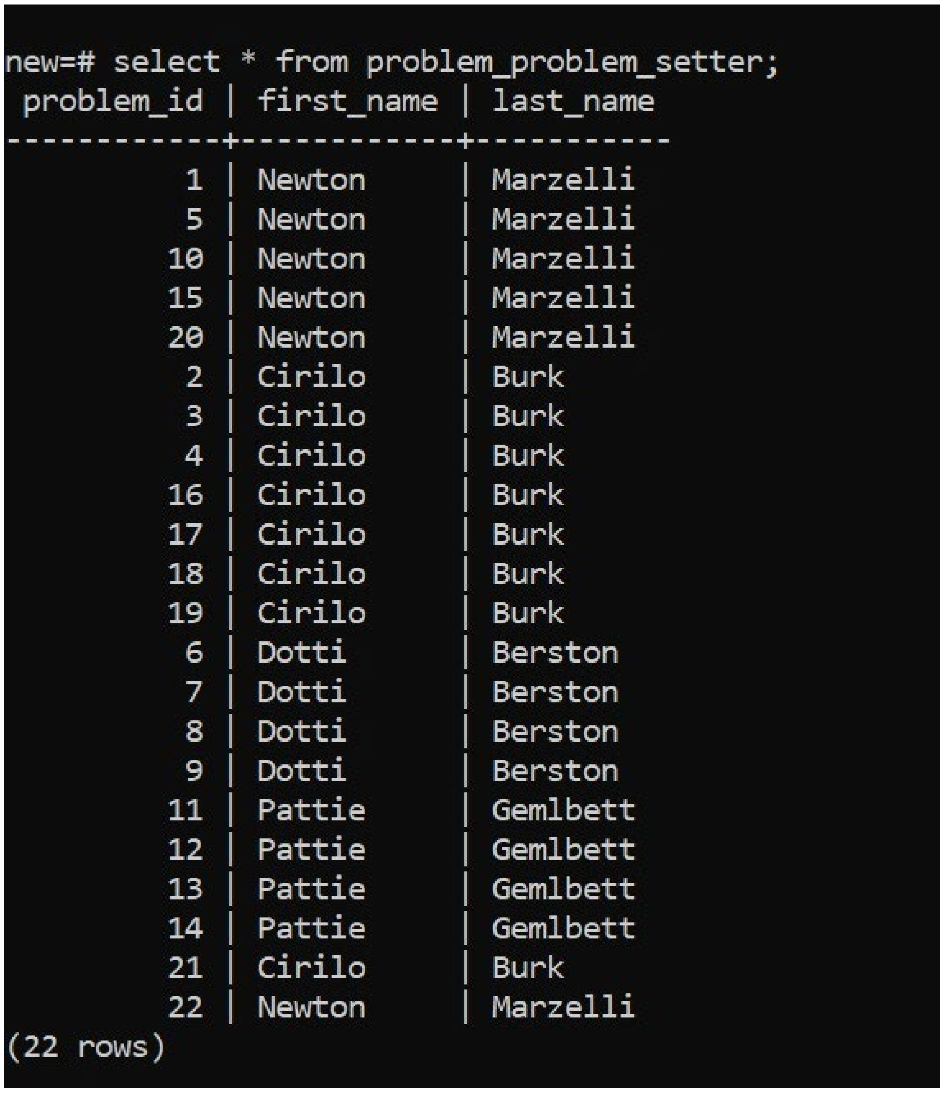


```
Competitive_Platform=# select get_user_info('nmarzellim');
Competitive_Platform
 get_user_info
----------------
 (nmarzellim,0)
(1 row)


Competitive_Platform=#
```

5.7 VIEWS

5.7.1 Create a view to display problems with their problem-setter's name.

create view problem_problem_setter as select problem_id, first_name, last_name from (problems inner join (select user_id as "uid", first_name, last_name from users where user_role='s') as "newt" on newt.uid=problems.user_id);



```
new=# select * from problem_problem_setter;
 problem_id | first_name | last_name
------------+------------+-----------
          1 | Newton     | Marzelli
          5 | Newton     | Marzelli
         10 | Newton     | Marzelli
         15 | Newton     | Marzelli
         20 | Newton     | Marzelli
          2 | Cirilo     | Burk
          3 | Cirilo     | Burk
          4 | Cirilo     | Burk
         16 | Cirilo     | Burk
         17 | Cirilo     | Burk
         18 | Cirilo     | Burk
         19 | Cirilo     | Burk
          6 | Dotti      | Berston
          7 | Dotti      | Berston
          8 | Dotti      | Berston
          9 | Dotti      | Berston
         11 | Pattie     | Gemlbett
         12 | Pattie     | Gemlbett
         13 | Pattie     | Gemlbett
         14 | Pattie     | Gemlbett
         21 | Cirilo     | Burk
         22 | Newton     | Marzelli
(22 rows)
```

6. FUTURE ENHANCEMENTS OF THE SYSTEM

6.6.1.1. We will design Front-end Design in HTML , CSS , JavaScript, ReactJs and Develop Bank-end in NodeJs.

6.6.1.2. For security purpose New Registration is done using OTP.

6.6.1.3. We will make database more consistent and We are making this database efficient and easy to implement with huge data capacity.

6.6.1.4. Methods and user data input will be lot easy after the implement of GUI.

6.6.1.5. We will also add some extra features so that the users can get answer for their complaints as fast as possible.

7. BIBLIOGRAPHY

7.6.1.1. For the successful implementation of this project we referred to many websites and books.
7.6.1.2. The schema was designed by taking ideas from website of election commission of india.
7.6.1.3. We created the ER Diagram and Schema Diagram on "Creatly.com".
7.6.1.4. Mostly we referred the online material for syntax of procedures, triggers, Exception and cursors.

Reference book:
Data Base System Concepts
-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Reference Websites:

7.6.1.5. https://www.stackoverflow.com/
7.6.1.6. https://www.mockaroo.com/
7.6.1.7. https://erdplus.com/
7.6.1.8. http://www.postgresqltutorial.com/