



Atal Bihari Vajpayee Indian Institute Of
Information Technology and Management

DATABASE SYSTEMS PROJECT

CODEFORCES DATABASE

Submission Date-10-Nov-2022

Submitted to- Dr. DEBANJAN SADHYA



Team Members -

Ankur Kumar Verma (2021IMG-009)

Dev Kudawla(2021IMG-023)

Himanshu Sharma (2021IMG-026)

Shaik Musharuff (2021IMG-056)

Vikram Singh (2021IMG-061)

Kodarapu Kaushik (2021IMG-031)

Contents

01 **Overview**

02 **ER Diagram**

03 **Assumptions**

04 **Relationship Sets and Cardinality**

05 **ER Diagram to Tables**

06 **Functional Dependencies**

08 **Decomposition(Using Normalisation)**

09 **Final Tables**

10 **Creating and inserting Data into Tables**

11 **Queries in Relational Algebra**

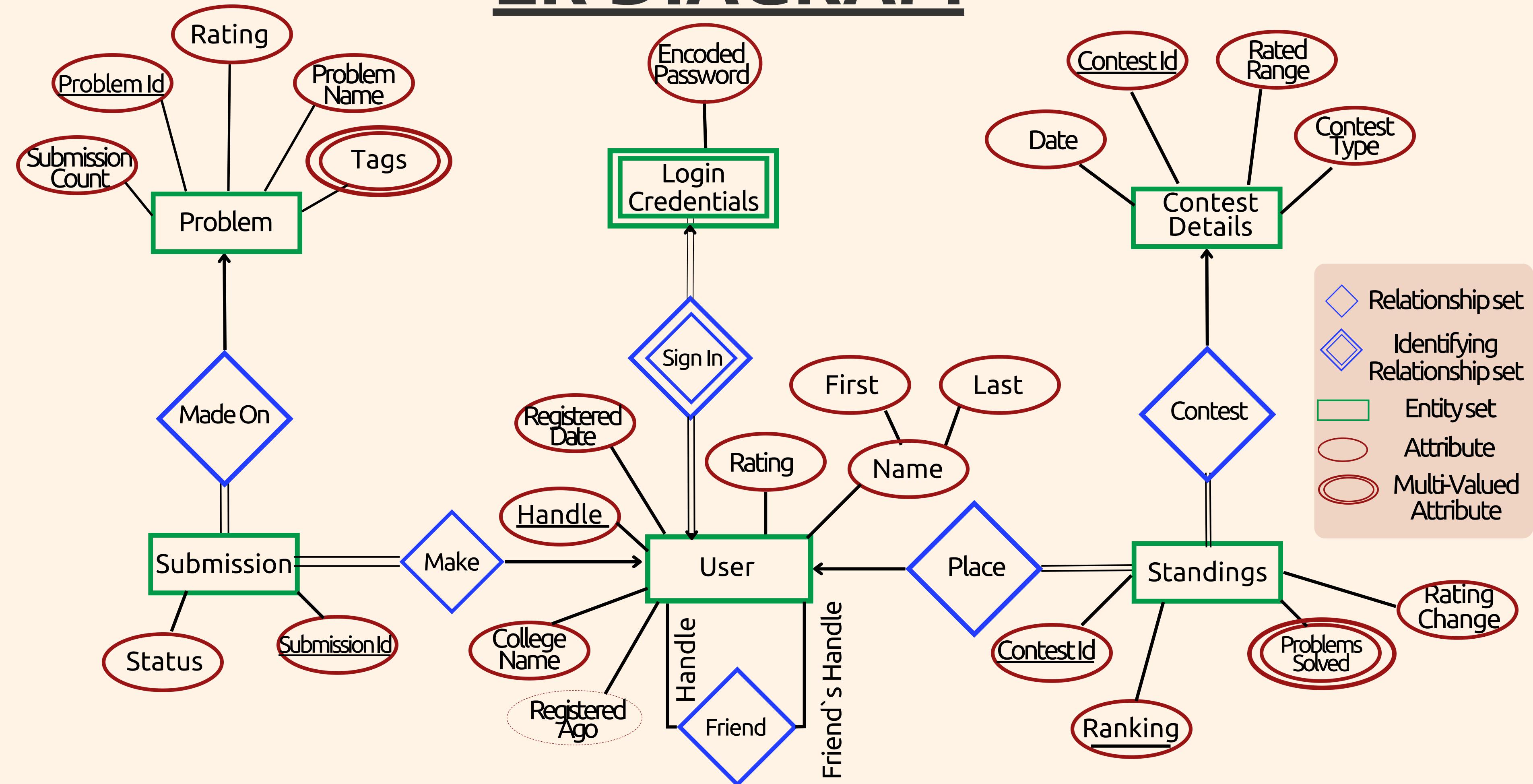
12 **Queries in MySql**

Overview



Codeforces is an online platform that hosts Competitive Programming Contests on regular basis. It was started and maintained by a group of Competitive Programmers from ITMO University led by Mike Mirzayanov. This Database System provides users to keep track about their Competitive Programming journey on Codeforces by providing the details of there performances in the contests, the levels of problems they have solved and where they stand among other participants. This Database also helps the user to see about their friends Competitive Programming journey.

ER DIAGRAM



Assumptions

- 01 Rating of a user cannot be negative.
- 02 In a given contest rank of two participants cannot be same.
- 03 Using the value of Rank in a given contest we can determine rating change.
- 04 We will be storing password in encoded form using hashing algorithm SHA-256.

Relationship Sets and Cardinality

• 01

Place

It is a one to many relationship between User and Standing as one user can have standings in many contests and one contest standing will be related to only user .Here Standing is showing total participation and User is showing partial participation.

• 02

Sign-In

it is an identifying relationship set. It is a one to one relationship between User and login credentials as a given user will have a single password to sign in. Here both login credential and User are showing total participation.

Relationship Sets and Cardinality

- 03 **Contest**
It is a one to many relationship between contest details and Standing as one contest will have many standing due to many users but a particular standing will be related to a particular contest .Here Standing is showing total participation and contest detail is showing partial participation.
- 04 **Friend**
(handle , friend 's handle)
It is unary relationship set on User entity set. It is a many to many relationship on User entity set as a user can have multiple friends and a particular user can be in friend list of many users. Here both are showing partial participation.

Relationship Sets and Cardinality

05

Make

It is a many to one relationship between submission and user as one submission will be made on a particular user but on a given user can make multiple submissions .Here submission is showing total participation and User is showing partial participation.

06

Made-On

It is a many to one relationship between submission and problem as one submission will be made on a particular problem but on a given problem there will be multiple submissions .Here submission is showing total participation and problem is showing partial participation.

ER DIAGRAM TO TABLES

- 01 **Submission**
submission id (primary key), handle (foreign key), problem id (foreign key), status
- 02 **Problem**
problem id (primary key), problem name, rating, submission count, tags
for tags we will be making another relation as it is a multivalued attribute.
- 03 **User**
handle (primary key) , first_name, last_name, registered date ,rating, college name, registered date

ER DIAGRAM TO TABLES

- 04
 - **Login Credentials**

handle(primary key), encoded password

- 05
 - **Friends**

handle(primary key), friend 's handle(primary key)

- 06
 - **Standings**

contest id (primary key), handle (primary key), ranking, problems solved

for problems solved we will be making another relation as it is a multivalued attribute.

- 07
 - **Contest**

contest id(primary key), contest type, rated range, datetime

Functional Dependencies

Schema	Functional Dependencies
Problem	Problem id->Problem Name,Rating,Submission count,Tag
standing	contest id,Rank->Handle,Rating change,Problem solved Ranking->Rating change
User	Handle->First name,Last name,Registered Date,College Name,Registered Ago,Rating Registered Date->Registered Ago
Contest	Contest id->Contest Type,Date,Rated Range
Submission	Submission Id->Handle,Problem Id,Status
Login Credentials	Handle->Encoded Password
Friends	

DECOMPOSITION(Using Normalisation)

01

Problem

1NF: Since Tags is not atomic as it can contains many values as a given problem can have many tags associated with it.

Now we split the Problem entity set into two entity different entity set

E1(Problem id,Problem Name,Rating,Submission count)

E2(Problem id,Tags)

2NF: Since there is no partial dependency so, it is in 2NF

3NF: Since there is no transitive dependency so, it is in 3NF

BCNF: Since there is no non-prime attribute defining a prime attribute so E1,E2 are in BCNF

new schemas are:

E1=(Problem id,Problem Name,Rating,Submission count)

E2=(Problem id,Tags)

Standing

1NF: Since Problem solved is not atomic as it can contains many values as in a given contest a given user can solve multiple problems.

Now we split the standing entity set into two entity different entity set

E1(contest id,Ranking,Handle,Rating change)

E2(Ranking,Problem id,Contest id)

2NF: Since there is a partial dependency i.e,

Ranking->Rating change

so we need to split E1 entity set into two different entity sets

E3(contest id,Ranking,Handle)

• E4(Rank,Rating change)

• **3NF:** Since there is no transitive dependency so, it is in 3NF

BCNF: Since there is no non-prime attribute defining a prime attribute

new schemas are:

E3=(contest id,Rank,Handle $)$

E2=(Ranking,Problem id,Contest id $)$

E4=(Ranking,Rating change $)$

03

User

1NF: Since the attributes are atomic i.e, attributes can't be divided into sub attributes

2NF: Since there is no partial dependency so,it is in 2NF

3NF: Since there is transitive dependency i.e,
Handle->First name,Last name,Registered Date,College Name,Registered Ago,Rating

Registered Date->Registered Ago

Now we split the standing entity set into two entity different entity set

E1(Handle,First name,Last name,Registered Date,College Name)

• E2(Registered Date,Registered Ago)

• **BCNF:** Since there is no non-prime attribute defining a prime attribute
new schemas are:

E1=(Handle,First name,Last name,Registered Date,College Name)

E2=(Registered Date,Registered Ago)

04

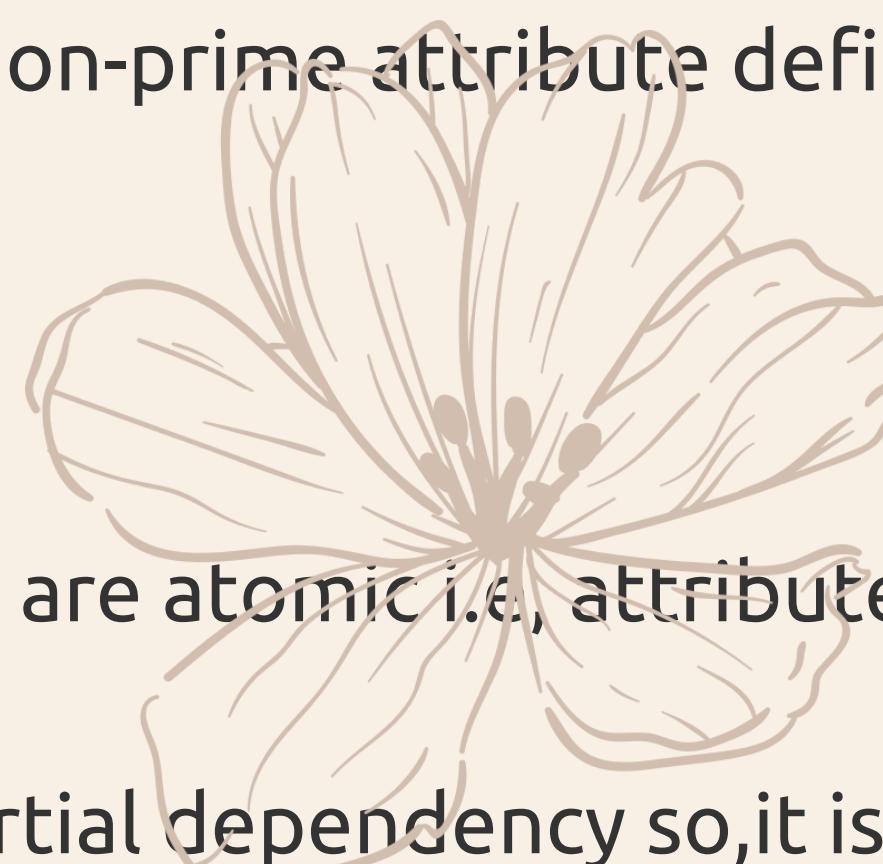
Contest

1NF: Since the attributes are atomic i.e, attributes can't be divided into sub attributes

2NF: Since there is no partial dependency so,it is in 2NF

3NF: Since there is no transitive dependency so,it is in 3NF

BCNF: Since there is no non-prime attribute defining a prime attribute so it is in BCNF



05

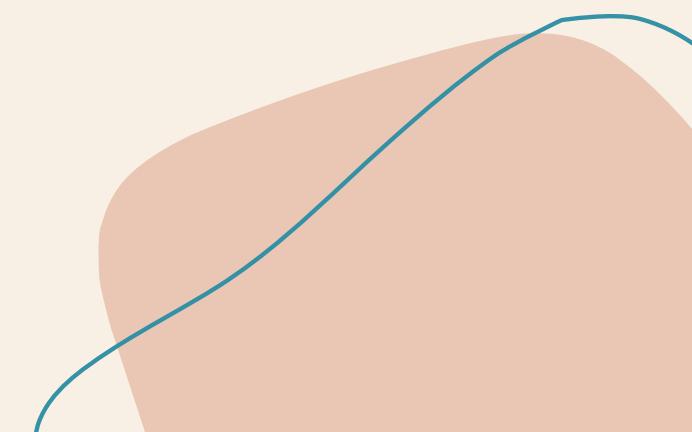
Submission

1NF: Since the attributes are atomic i.e, attributes can't be divided into sub attributes

2NF: Since there is no partial dependency so,it is in 2NF

3NF: Since there is no transitive dependency so,it is in 3NF

BCNF: Since there is no non-prime attribute defining a prime attribute so it is in BCNF



06

Login Credentials

1NF: Since the attributes are atomic i.e, attributes can't be divided into sub attributes

2NF: Since there is no partial dependency so,it is in 2NF

3NF: Since there is no transitive dependency so,it is in 3NF

BCNF: Since there is no non-prime attribute defining a prime attribute so it is in BCNF

Final Tables

- Problem(Problem id,Problem Name,Rating,Submission count)
- Tags(Problem id,Tags)
- Standings(contest id,Ranking,Handle)
- Standings_Problem_Solved(Ranking,Problem id,Contest id)
- Standings_Rating(Ranking,Rating change)
- User(Handle,First name,Last name,Registered Date,College Name,Rating)
- Registered_Date(Registered Date,Registered Ago)
- Contest(Contest Id,Contest Type,Rated Range,Date)
- Submission(Submission Id,Handle,Problem Id,Status)
- Login Credentials(Handle,Encoded Password)
- Friends(Handle,Friend's Handle)

Tables after Decomposition

Contest	
PK	Contest ID
	Contest Type
	Rated Range
	DateTime

Tags	
PK	Problem ID
PK	Problem Tag

Problem	
PK	Problem ID
	Problem Name
	Rating
	Submission Count

Submission	
PK	Submission ID
FK	Handle
FK	Problem ID
	Status

Login Credentials	
PK	Handle
	Encoded Password

User	
PK	Handle
	First_Name
	Last_Name
	Rating
FK	Registered Date
	College Name

Standings_Rating	
PK	Ranking
	Rating_change

Standings_Problem_Solved	
PK	Ranking
PK	Problem ID
PK	Contest ID

Friends	
PK	Handle
PK	Friends Handle

Standings	
PK	Contest ID
FK	Handle
PK	Ranking

Register_date	
PK	Registered_date
	Registered_ago

Creating and Inserting Data into Tables

User Table

```
create table user(handle varchar(20) primary key, first_name varchar(15) NOT NULL ,last_name varchar(15) NOT NULL ,rating int NOT NULL , registered_date date , college_name varchar(30),foreign key(registered_date_fk) references register_date(registered_date));
```

```
insert into user values('saturnbored', 'Shivam', 'Ruhil', 1692, '2020-10-21', 'ABV-IIITM Gwalior'),  
( 'rohita', 'Rohit', 'Sigar', 1626, '2019-9-20', 'ABV-IIITM Gwalior'),  
( 'satyabratojha', 'Satyabrat', 'Ojha', 1325, '2021-12-20', 'ABV-IIITM Gwalior'),  
( 'himanshu_0502', 'Himanshu', 'Sharma', 1316, '2022-3-31', 'ABV-IIITM Gwalior'),  
( 'addy0110', 'Adarsh', 'Pandey', 1475, '2021-4-11', 'ABV-IIITM Gwalior'),  
( 'kal013', 'Kalash', 'Gupta', 2791, '2016-5-25', 'IIT Delhi'),  
( 'demoralizer', 'Utkarsh', 'Gupta', 2345, '2018-7-9', 'IIT BHU'),  
( 'priyanshdec31', 'Priyansh', 'Agrawal', 2145, '2017-11-16', 'IIIT Delhi'),  
( 'sudoBug', 'Arjan', 'Bal', 2187, '2017-6-8', 'ABV-IIITM Gwalior'),  
( 'tourist', 'Gennady', 'Korotkevich', 3912, '2008-12-31', 'ITMO University');
```

	handle	first_name	last_name	rating	registered_date	college_name
▶	addy0110	Adarsh	Pandey	1475	2021-04-11	ABV-IIITM Gwalior
	demoralizer	Utkarsh	Gupta	2345	2018-07-09	IIT BHU
	himanshu_0502	Himanshu	Sharma	1316	2022-03-31	ABV-IIITM Gwalior
	kal013	Kalash	Gupta	2791	2016-05-25	IIT Delhi
	priyanshdec31	Priyansh	Agrawal	2145	2017-11-16	IIIT Delhi
	rohita	Rohit	Sigar	1626	2019-09-20	ABV-IIITM Gwalior
	saturnbored	Shivam	Ruhil	1692	2020-10-21	ABV-IIITM Gwalior
	satyabratojha	Satyabrat	Ojha	1325	2021-12-20	ABV-IIITM Gwalior
	sudoBug	Arjan	Bal	2187	2017-06-08	ABV-IIITM Gwalior
	tourist	Gennady	Korotkevich	3912	2008-12-31	ITMO University

Creating and Inserting Data into Tables

Register_date Table

```
create table register_date(registered_date date primary key,registered_ago int);
```

```
insert into register_date values('2020-10-21',24),('2019-9-20',35),('2021-12-20',11),('2022-3-31',7),('2021-4-11',18),  
('2016-5-25',65),('2018-7-9',39),('2017-11-16',47),('2017-6-8',52),('2008-12-31',145);
```

	registered_date	registered_ago
▶	2008-12-31	145
	2016-05-25	65
	2017-06-08	52
	2017-11-16	47
	2018-07-09	39
	2019-09-20	35
	2020-10-21	24
	2021-04-11	18
	2021-12-20	11
	2022-03-31	7

Creating and Inserting Data into Tables

Submission Table

```
create table submission(submission_id char(9) primary key,handle_fk varchar(20),problem_id_fk char(5),status varchar(25), foreign key(handle_fk)
references user(handle),foreign key(problem_id_fk) references problem(problem_id));
```

```
insert into submission values('176547962','himanshu_0502','1000A','Accepted'),('176549696','himanshu_0502','1000B','Accepted'),
('176578904','himanshu_0502','1000C','Accepted'),('176588062','satyabratojha','1001A','Accepted'),
('176595902','satyabratojha','1001B','Wrong Answer'),('176647962','satyabratojha','1001B','Accepted'),
('176678962','satyabratojha','1001C','Accepted'),('176849962','satyabratojha','1001D','Wrong Answer'),
('176638962','saturnbored','1001B','Accepted'),('176697962','saturnbored','1001C','Accepted'),
('176749062','saturnbored','1001D','Accepted'),('176875962','satyabratojha','1001D','Accepted'),
('176594262','saturnbored','1001A','Accepted'),('177178994','himanshu_0502','1002A','Accepted'),
('177179904','satyabratojha','1002A','Accepted'),('177181204','himanshu_0502','1002B','Accepted'),
('177182394','satyabratojha','1002B','Accepted'),('177182604','satyabratojha','1002C','Accepted'),
('177183104','himanshu_0502','1002C','Accepted'),('177183904','satyabratojha','1002D','Accepted'),
('177184204','himanshu_0502','1002D','Accepted'),('177185604','himanshu_0502','1002E','Accepted'),
('177185704','satyabratojha','1002E','Accepted'),('177585704','tourist','1003A','Accepted'),
('177585804','tourist','1003B','Accepted'),('177586004','tourist','1003C','Accepted'),
('177586456','tourist','1003D','Accepted'),('177586989','tourist','1003E','Accepted'),
('177588998','tourist','1003F','Accepted'),('177591267','tourist','1003G','Accepted'),
('177599908','tourist','1003H','Accepted');
```

submission_id	handle_fk	problem_id_fk	status
176547962	himanshu_0502	1000A	Accepted
176549696	himanshu_0502	1000B	Accepted
176578904	himanshu_0502	1000C	Accepted
176588062	satyabratojha	1001A	Accepted
176594262	saturnbored	1001A	Accepted
176595902	satyabratojha	1001B	Wrong Answer
176638962	saturnbored	1001B	Accepted
176647962	satyabratojha	1001B	Accepted
176678962	satyabratojha	1001C	Accepted
176697962	saturnbored	1001C	Accepted
176749062	saturnbored	1001D	Accepted
176849962	satyabratojha	1001D	Wrong Answer
176875962	satyabratojha	1001D	Accepted
177178994	himanshu_0502	1002A	Accepted
177179904	satyabratojha	1002A	Accepted
177181204	himanshu_0502	1002B	Accepted
177182394	satyabratojha	1002B	Accepted
177182604	satyabratojha	1002C	Accepted
177183104	himanshu_0502	1002C	Accepted
177183904	satyabratojha	1002D	Accepted
177184204	himanshu_0502	1002D	Accepted
177185604	himanshu_0502	1002E	Accepted
177185704	satyabratojha	1002E	Accepted
177585704	tourist	1003A	Accepted
177585804	tourist	1003B	Accepted
177586004	tourist	1003C	Accepted
177586456	tourist	1003D	Accepted
177586989	tourist	1003E	Accepted
177588998	tourist	1003F	Accepted
177591267	tourist	1003G	Accepted
177599908	tourist	1003H	Accepted

Creating and Inserting Data into Tables

Problem Table

```
create table problem(problem_id char(5) primary key , problem_name varchar(20),rating int , submission_count int NOT NULL);
```

```
insert into problem values('1000A', 'XOR Product', 1000, 12348),
('1000B', 'Palindrome Find', 1200, 9048),('1000C', 'Swap Game', 1500, 5098),('1000D', 'Another Problem', 1900, 1282),
('1000E', 'MEX vs MED', 2500, 248),('1001A', 'Nim', 800, 21348),('1001B', 'Product Power', 1100, 15348),
('1001C', 'Prime Number', 1400, 6473),('1001D', 'AND Oracle', 1900, 2356),('1001E', 'Permutation', 2500, 589),
('1001F', 'Balance Sequence', 2900, 145),('1001G', 'Fibonacci DP', 3300, 67),('1001H', 'Is this fft', 3500, 8),
('1002A', 'Easy One', 800, 23067),('1002B', 'Step Move', 800, 21567),
('1002C', 'Array Balance', 900, 17567),('1002D', 'Sum of Two', 1100, 11567),('1002E', 'Increasing Sequence', 1400, 7367),
('1002F', 'Find Substring', 1700, 2567),('1003A', 'Random Tests', 800, 25567),('1003B', 'Alice and Bob', 800, 26567),
('1003C', 'Game is Game', 800, 22567),('1003D', 'Power XOR', 1100, 18567),('1003E', 'Swap Game', 1300, 15567),
('1003F', 'Find Waldo', 1400, 7567),('1003G', 'Prmutation Break', 1500, 5934),('1003H', 'Subsequence J', 1600, 1567),
('1002G', 'Buzz!', 2800, 507);
```

problem_id	problem_name	rating	submission_count
1000A	XOR Product	1000	12348
1000B	Palindrome Find	1200	9048
1000C	Swap Game	1500	5098
1000D	Another Problem	1900	1202
1000E	MEX vs MED	2500	248
1001A	Nim	800	21348
1001B	Product Power	1100	15348
1001C	Prime Number	1400	6473
1001D	AND Oracle	1900	2356
1001E	Permutation	2500	589
1001F	Balance Sequence	2900	145
1001G	Fabonacci DP	3300	67
1001H	Is this fft	3500	8
1002A	Easy One	800	23067
1002B	Step Move	800	21567
1002C	Array Balance	900	17567
1002D	Sum of Two	1100	11567
1002E	Increasing Sequence	1400	7367
1002F	Find Substring	1700	2567
1002G	Buzz!	2000	507
1003A	Random Tests	800	25567
1003B	Alice and Bob	800	26567
1003C	Game is Game	800	22567
1003D	Power XOR	1100	18567
1003E	Swap Game	1300	15567
1003F	Find Waldo	1400	7567
1003G	Prmutation Break	1500	5934
1003H	Subsequence J	1600	1567

Creating and Inserting Data into Tables

Login_Credentials Table

```
→ create table login_credentials(email_id varchar(25) primary key, handle_fk varchar(20), password char(64),  
~ foreign key(handle_fk) references user(handle));
```

```
insert into login_credentials values('hs034@gmail.com','himanshu_0502','6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b'),  
( 'so164@gmail.com', 'satyabratojha', 'd4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35' ),  
( 'sb980@gmail.com', 'saturnbored', '4e07408562bedb8b60ce05c1decfe3ad16b72230967de01f640b7e4729b49fce' ),  
( 'ra178@gmail.com', 'rohita', '4b227777d4dd1fc61c6f884f48641d02b4d121d3fd328cb08b5531fcacdabf8a' ),  
( 'tr007@gmail.com', 'tourist', 'ef2d127de37b942baad06145e54b0c619a1f22327b2ebbcfbec78f5564afe39d' ),  
( 'ay194@gmail.com', 'addy0110', 'e7f6c011776e8db7cd330b54174fd76f7d0216b612387a5ffcfb81e6f0919683' ),  
( 'k1199@gmail.com', 'kal013', '7902699be42c8a8e46fbbe4501726517e86b22c56a189f7625a6da49081b2451' ),  
( 'dm764@gmail.com', 'demoralizer', '2c624232cdd221771294dfbb310aca000a0df6ac8b66b696d90ef06fdefb64a3' ),  
( 'pa164@gmail.com', 'priyanshdec31', '19581e27de7ced00ff1ce50b2047e7a567c76b1cbaebabe5ef03f7c3017bb5b7' ),  
( 'sb164@gmail.com', 'sudoBug', '4a44dc15364204a80fe80e9039455cc1608281820fe2b24f1e5233ade6af1dd5' );
```

	handle	password
▶	addy0110	e7f6c011776e8db7cd330b54174fd76f7d0216b...
	demoralizer	2c624232cdd221771294dfbb310aca000a0df6a...
	himanshu_0502	6b86b273ff34fce19d6b804eff5a3f5747ada4ea...
	kal013	7902699be42c8a8e46fbbb4501726517e86b22c...
	priyanshdec31	19581e27de7ced00ff1ce50b2047e7a567c76b1...
	rohita	4b227777d4dd1fc61c6f884f48641d02b4d121d...
	saturnbored	4e07408562bedb8b60ce05c1decfe3ad16b7223...
	satyabratojha	d4735e3a265e16eee03f59718b9b5d03019c07...
	sudoBug	4a44dc15364204a80fe80e9039455cc16082818...
	tourist	ef2d127de37b942baad06145e54b0c619a1f223...

Creating and Inserting Data into Tables

Standings Table

```
create table standings(contest_id char(4), ranking int, handle_fk varchar(20), primary key (contest_id,ranking),
foreign key(handle_fk) references user(handle));  
  
insert into standings values('1000',1290,'himanshu_0502'),('1001',756,'satyabratojha'),('1001',689,'saturnbored'),
('1002',889,'himanshu_0502'),('1002',886,'satyabratojha'),('1003',1,'tourist');
```

	contest_id	ranking	handle_fk
▶	1000	1290	himanshu_0502
	1002	889	himanshu_0502
	1001	689	saturnbored
	1001	756	satyabratojha
	1002	886	satyabratojha
	1003	1	tourist

Creating and Inserting Data into Tables

Standings_Rating Table

```
create table standings_rating(ranking int primary key, rating_change int);  
  
insert into standings_rating values(1290,133),(756,219),(689,35),(889,93),(886,100),(1,0);
```

ranking	rating_change
1	0
689	35
756	219
886	100
989	98
1290	133

Creating and Inserting Data into Tables

Standings_Problem_Solved Table

```
create table standings_problem_solved(ranking int,problem_id char(5),
contest_id char(4),primary key(ranking,problem_id,contest_id));
```

```
insert into standings_problem_solved values(1290,'1000A','1000'),(1290,'1000B','1000'),(1290,'1000C','1000'),
(756,'1001A','1001'),(756,'1001B','1001'),(756,'1001C','1001'),
(756,'1001D','1001'),(689,'1001A','1001'),(689,'1001B','1001'),
(689,'1001C','1001'),(689,'1001D','1001'),(889,'1002A','1002'),
(889,'1002B','1002'),(889,'1002C','1002'),(889,'1002D','1002'),
(889,'1002E','1002'),(886,'1002A','1002'),(886,'1002B','1002'),
(886,'1002C','1002'),(886,'1002D','1002'),(886,'1002E','1002'),
(1,'1003A','1003'),(1,'1003B','1003'),(1,'1003C','1003'),
(1,'1003D','1003'),(1,'1003E','1003'),(1,'1003F','1003'),
(1,'1003G','1003'),(1,'1003H','1003');
```

ranking	problem_i	contest_id
1	1003A	1003
1	1003B	1003
1	1003C	1003
1	1003D	1003
1	1003E	1003
1	1003F	1003
1	1003G	1003
1	1003H	1003
689	1001A	1001
689	1001B	1001
689	1001C	1001
689	1001D	1001
756	1001A	1001
756	1001B	1001
756	1001C	1001
756	1001D	1001
886	1002A	1002
886	1002B	1002
886	1002C	1002
886	1002D	1002
886	1002E	1002
889	1002A	1002
889	1002B	1002
889	1002C	1002
889	1002D	1002
889	1002E	1002
1290	1000A	1000
1290	1000B	1000
1290	1000C	1000

Creating and Inserting Data into Tables

Tags Table

```
create table tags(problem_id char(5), problem_tag varchar(25), primary key(problem_id,problem_tag));  
  
insert into tags values('1000A','greedy'),('1000A','bitmasks'),('1000B','strings'),('1000C','games'),  
('1000D','graph'),('1000D','dfs and similar'),('1000E','math'),('1000E','dp'),  
('1000E','combinatorics'),('1001A','games'),('1001A','greedy'),('1001B','number theory'),  
('1001C','math'),('1001D','greedy'),('1001D','dp'),('1001D','bitmasks'),  
('1001E','constructive algorithms'),('1001E','greedy'),('1001F','strings'),('1001F','implementation'),  
('1001F','graph'),('1001F','flows'),('1001G','dp'),('1001H','fft'),('1001H','math'),  
('1002A','greedy'),('1002B','implementation'),('1002B','brute force'),  
('1002C','greedy'),('1002C','two pointers'),('1002D','constructive algorithms'),('1002E','greedy'),  
('1002E','sorting'),('1002F','strings'),('1002F','brute force'),  
('1002G','binary search'),('1002G','greedy'),('1002G','dp'),('1003A','math'),('1003B','two pointers'),  
('1003C','games'),('1003C','greedy'),('1003D','math'),('1003D','bitmasks'),('1003E','greedy'),  
('1003E','games'),('1003E','implementation'),('1003F','dp'),('1003F','shortest path'),  
('1003G','divide and conquer'),('1003G','greedy'),('1003H','two pointers'),('1003H','binary search'),('1003H','dp');
```

problem	problem_tag		
1000A	bitmasks	1002B	implementation
1000A	greedy	1002C	greedy
1000B	strings	1002C	two pointers
1000C	games	1002D	constructive algorithms
1000D	dfs and similar	1002E	greedy
1000D	graph	1002E	sorting
1000E	combinatorics	1002F	brute force
1000E	dp	1002F	strings
1000E	math	1002G	binary search
1001A	games	1002G	dp
1001A	greedy	1002G	greedy
1001B	number theory	1003A	math
1001C	math	1003B	two pointers
1001D	bitmasks	1003C	games
1001D	dp	1003C	greedy
1001D	greedy	1003D	bitmasks
1001E	constructive algorithms	1003D	math
1001E	greedy	1003E	games
1001F	flows	1003E	greedy
1001F	graph	1003E	implementation
1001F	implementation	1003F	dp
1001F	strings	1003F	shortest path
1001G	dp	1003G	divide and conquer
1001H	fft	1003G	greedy
1001H	math	1003H	binary search
1002A	greedy	1003H	dp
1002B	brute force	1003H	two pointers

Creating and Inserting Data into Tables

Friends Table

```
create table friends(handle varchar(20),friends_handle varchar(20),primary key(handle,friends_handle));  
  
insert into friends values('himanshu_0502','satyabratojha'),('himanshu_0502','saturnbored'),  
('himanshu_0502','rohita'),('himanshu_0502','tourist'),  
('himanshu_0502','kal013'),('satyabratojha','himanshu_0502'),  
('satyabratojha','saturnbored'),('satyabratojha','tourist'),  
('saturnbored','addy0110'),('saturnbored','tourist'),  
('saturnbored','rohita'),('saturnbored','sudoBug');
```

	handle	friends_handle
▶	himanshu_0502	kal013
	himanshu_0502	rohita
	himanshu_0502	saturnbored
	himanshu_0502	satyabratojha
	himanshu_0502	tourist
	saturnbored	addy0110
	saturnbored	rohita
	saturnbored	sudoBug
	saturnbored	tourist
	satyabratojha	himanshu_0502
	satyabratojha	saturnbored
	satyabratojha	tourist

Creating and Inserting Data into Tables

Contest Table

```
create table contest(contest_id char(4) primary key, contest_type varchar(25) ,rated_range int ,Date datetime);

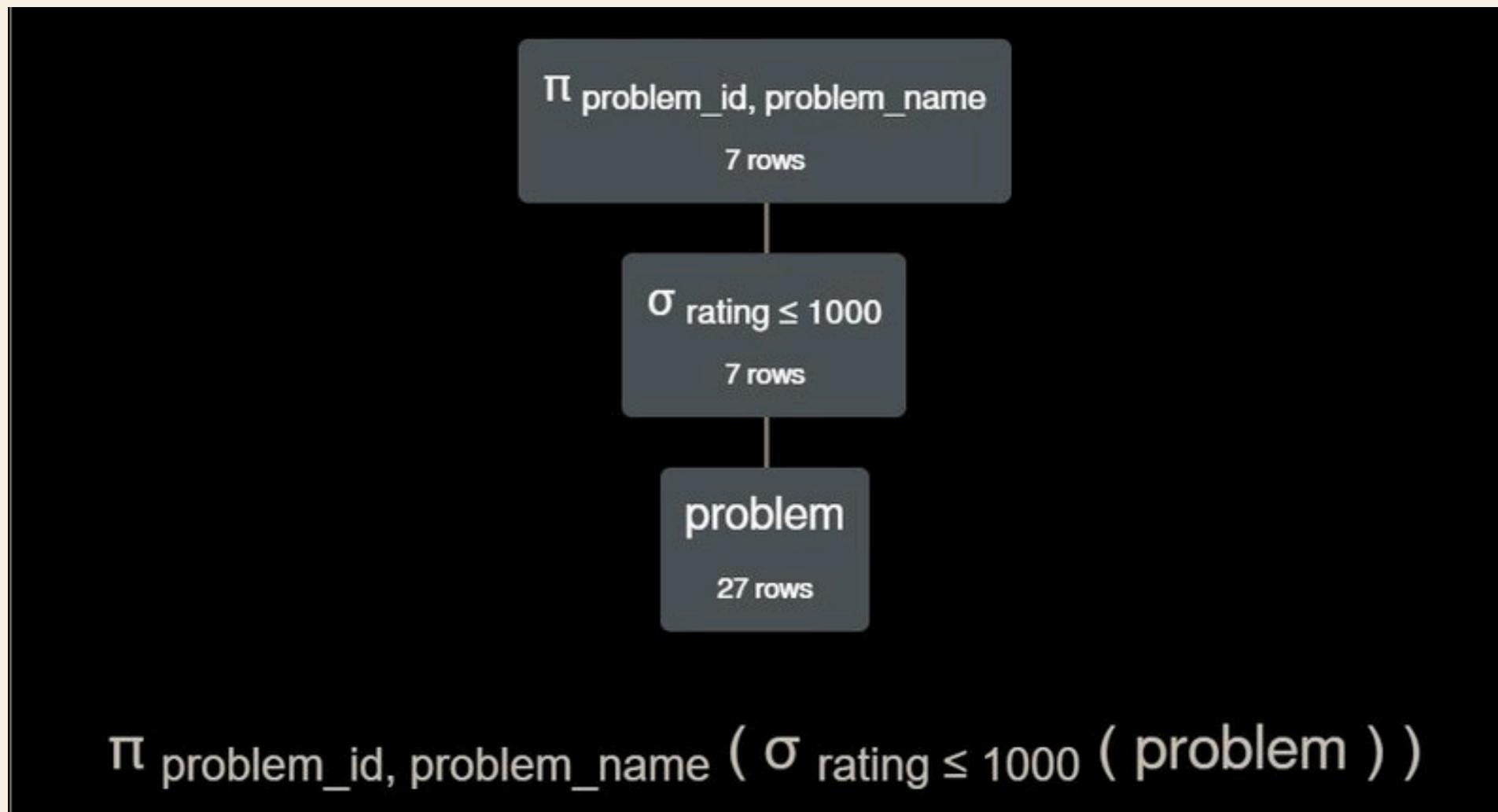
insert into contest values('1000', 'Codeforces Round Div 2',2099,'2022-9-10 20:05:00'),
('1001', 'Codeforces Global Round',9999,'2022-9-13 20:05:00'),('1002', 'Codeforces Round Div 3',1599,'2022-9-15 17:05:00'),
('1003', 'Codeforces Round Div 4',1399,'2022-9-16 13:35:00');
```

contest_id	contest_type	rated_range	Date
1000	Codeforces Round Div 2	2099	2022-09-10 20:05:00
1001	Codeforces Global Round	9999	2022-09-13 20:05:00
1002	Codeforces Round Div 3	1599	2022-09-15 17:05:00
1003	Codeforces Round Div 4	1399	2022-09-16 13:35:00

Queries in Relational Algebra

Q1. List of all problems whose rating is less than or equal to 1000.

Query:

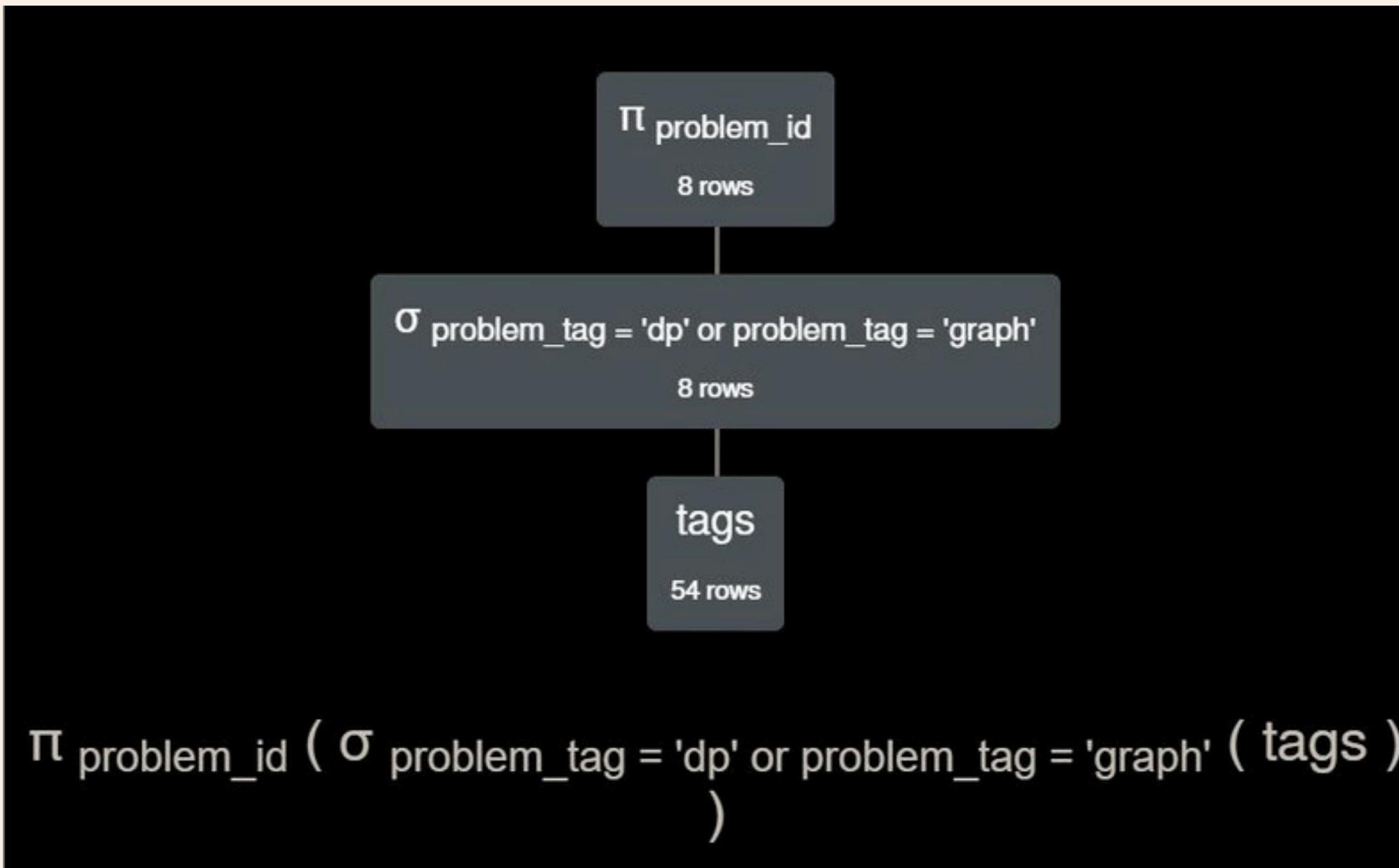


Output:

problem.problem_id	problem.problem_name
'1001A'	'Nim'
'1002A'	'Easy One'
'1002B'	'Step Move'
'1002C'	'Array Balance'
'1003A'	'Random Tests'
'1003B'	'Alice and Bob'
'1003C'	'Game is Game'

Q2. list of all problem with tags dp(dynamic programming) or graph.

Query:

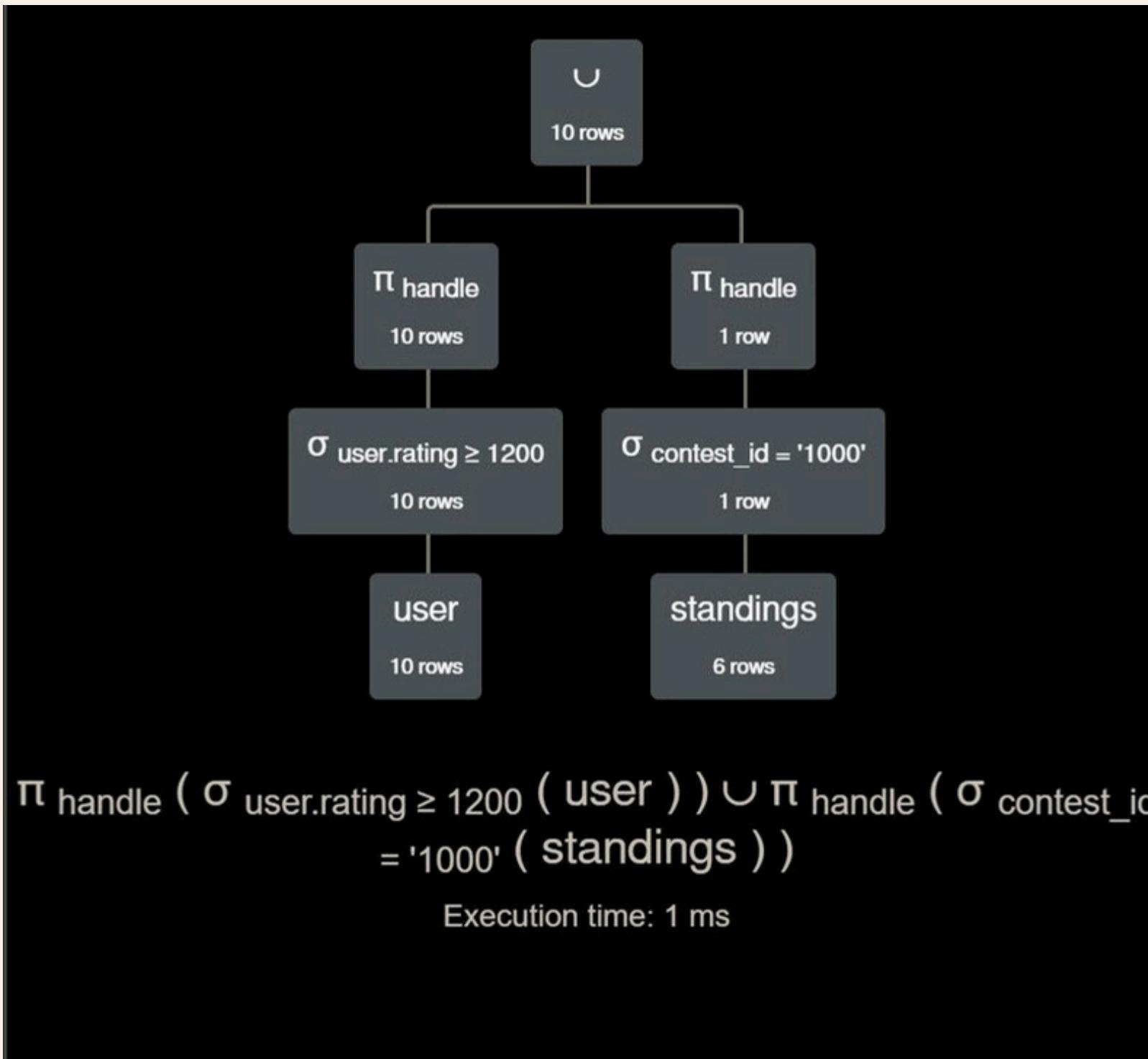


Output:

tags.problem_id
'1000D'
'1000E'
'1001D'
'1001F'
'1001G'
'1002G'
'1003F'
'1003H'

Q3. Name all users who either have rating greater than or equal to 1200 or participated in contest where contest id='1000'.

Query:

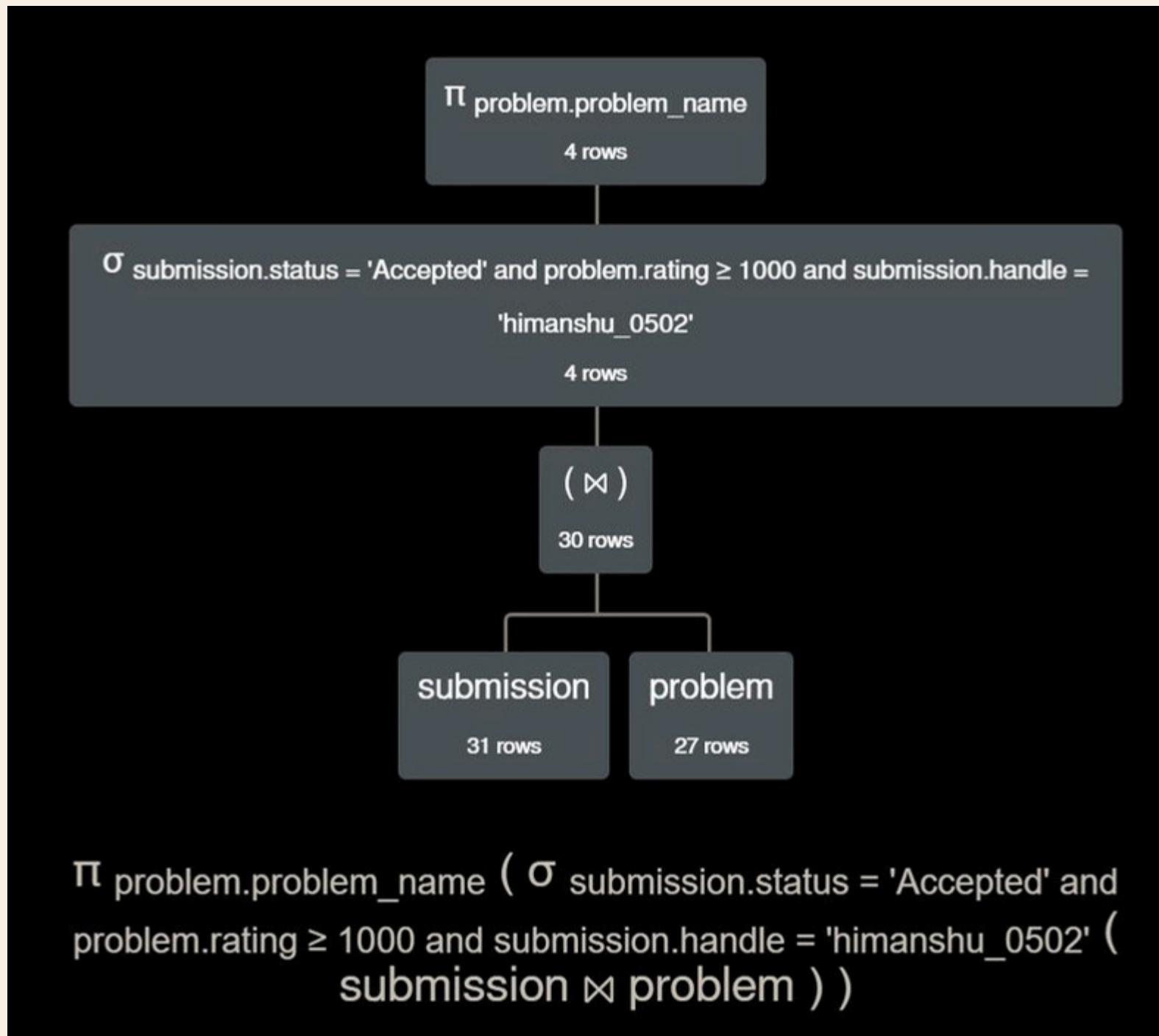


Output:

user.handle
'saturnbored'
'rohita'
'satyabratojha'
'himanshu_0502'
'addy0110'
'kal013'
'demoralizer'
'priyanshdec31'
'sudoBug'
'tourist'

Q4. name of problems of rating ≥ 1000 solved by 'himanshu_0502'.

Query:

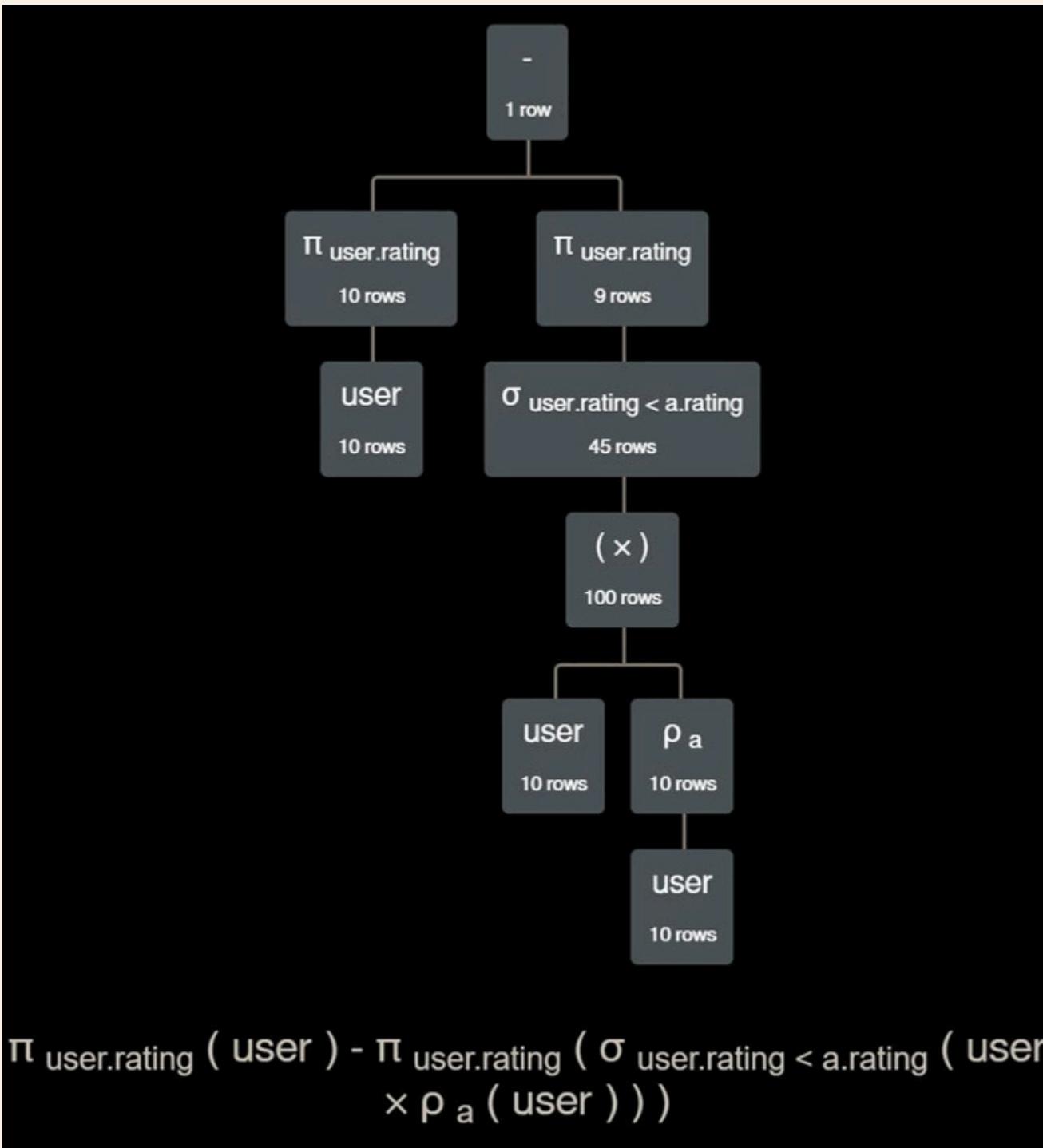


Output:

problem.problem_name
'Palindrome Find'
'Swap Game'
'Sum of Two'
'Increasing Sequence'

Q5. Maximum rating attained by any user.

Query:



Output:

user.rating
3912

Queries in MySql

Q1. List of all friends of user 'himanshu_0502'.

Query:

```
select friends_handle from friends where handle = 'himanshu_0502';
```

Output:

	friends_handle
▶	kal013
▶	rohitaa
▶	saturnbored
▶	satyabratojha
▶	tourist

Queries in MySql

Q2. User who had registered before 10 years ago.

Query:

```
select handle from user natural join register_date where registered_ago>10;
```

Output:

Handle
louis123

Queries in MySql

Q3. Find the maximum rating of every college in descending order.

Query:

```
select college_name,max(rating) from user group by college_name order by rating desc;
```

Output:

college_name	max(rating)
IITD University	2912
IIT Delhi	2791
IIT BHU	2345
IIT Delhi	2145
ABV-IIITM Gwalior	2187

Queries in MySql

Q4. details of all users whose first name starts with 'a' or ends with 'sh'.

Query:

```
select * from user where first_name like "a%" or first_name like "%sh";
```

Output:

Handle	first_name	last_name	rating	registered_date	college_name
addy0110	Adarsh	Pandey	1475	2021-04-11	ABV-IIITM Gwalior
demoraker	Utkarsh	Gupta	2345	2018-07-09	IIT BHU
kal013	Kalash	Gupta	2791	2016-05-25	IIT Delhi
pryanishdec01	Priyansh	Agrawal	2145	2017-11-16	IIT Delhi
sudoBug	Arjan	Bal	2187	2017-06-08	ABV-IIITM Gwalior

Queries in MySql

Q5. Handle name of all users who are friend of either 'himanshu_0502' or 'satyabratojha'

Query:

```
select friends_handle from friends where handle = 'himanshu_0502' union
(select friends_handle from friends where handle = 'satyabratojha'))
```

Output:

friends_handle
ka013
rohit8
ss_tumbored
satyabratojha
tourist
himanshu_0502



Thank You