DBMS Project

On

Online Coding Platform Database Designed by

THALLADA PAVAN SIDDARTHA ,22CSB0C09, CSE-A

KOTIPALLI CHAITANYA KARTHIK,22CSB0A13, CSE-A

Problem Statement:

This database that we have constructed in our project basically gives a broad overview about how a Coding site or coding platform works, in general there are many coding sites that we use daily, we have analysed how the data is managed in those sites and have constructed an efficient DB.

This database consists of only essential components (entities) that covers almost all features of an efficient coding site, giving an edge to

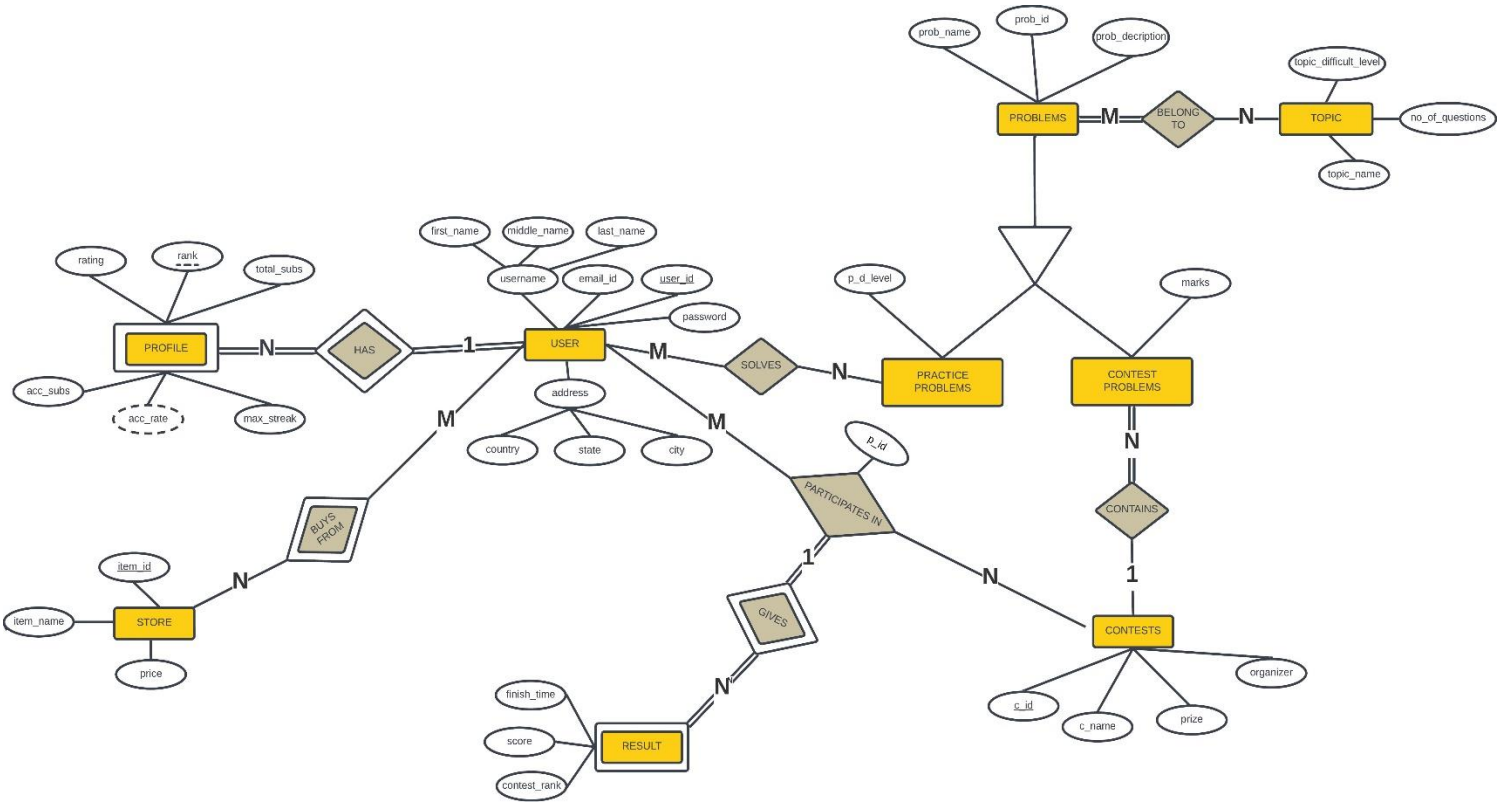
- Participate in contests that occurs weekly,
- Announcing the results,
- Practising problems, and
- Even the Store that helps them to buy the necessary goodies.

It mainly focuses on the entities and the relationships between them with all the key constraints and participation constraints.

ASSUMPTIONS :-

1. We have assumed that no two users will have the same profile.
2. A problem can include 1 or more topics.
3. No two participants will get the same result after the contest.

ER diagram :



A relation is said to be in 1NF ,

“ If there are no multivalued attributes , composite attributes , all columns have unique names . ”

Relational Schema :

1.Entity :- **USER**

Attributes :-

- user_id
- username(composite attribute)
- email_id
- address(composite attribute)

1NF :- This relation does not satisfy 1NF condition as there is a composite attribute. To bring this into 1NF, sub attributes are to be inserted instead of main attributes.

Attributes :-

- user_id
- first_name
- middle_name
- last_name
- email_id
- Password
- country
- State
- city

Functional dependencies :-

- user_id ➡ first_name
- user_id ➡ middle_name
- user_id ➡ last_name
- user_id ➡ email_id

- user_id ➡ password
- user_id ➡ country
- user_id ➡ state
- user_id ➡city
- city ➡ state
- state ➡ country
- email_id ➡ user_id
- email_id➡ first_name
- email_id➡ middle_name
- email_id ➡ last_name
- email_id ➡ country
- email_id ➡ password
- email_id ➡ city
- email_id ➡ state

2NF :- As we can see in the above functional dependencies the determinant (X in X ➡ Y) is a single attribute i.e, all our candidate keys are single attributes . so , there exists no proper subset of candidate keys.Therefore there is no chance of existence of partial dependency . Hence this table is in 2NF.

3NF :- In the above functional dependencies we can say that city and state are non-prime attributes and they are determining some other non-prime attributes .Hence, we can say that there is a transitive dependency . Therefore this table is not in 3NF. To bring this table into 3NF we should do lossless decomposition.

Decomposing the above table into 3 tables.

1) **USER** - user_id, first_name, middle_name,last_name, email_id, password, city
2) **CITY** - city, state
3) **STATE** - state, country

We can say that this is a valid/ lossless decomposition because the intersection of city and state tables i.e, state determines the whole state table and their join i.e city, state, country when intersected with the user table gives city which in turn determines the join.

BCNF :- In all the above modified tables , only the superkeys are determining all other attributes . Hence, we can say that the table is in BCNF .

Note :- Here,it is shown as user,city,state but the creation will be done in state,city,user order only.

2.Entity :- PROFILE

Attributes :-

- user_id
- rating
- rank
- max_streak
- total_subs
- acc_subs
- acc_rate(derived)

Functional dependencies :-

- user_id ➡ rank
- user_id ➡ rating
- user_id ➡ max_streak
- user_id ➡ total_subs
- user_id ➡ acc_subs

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF :- In this table, there is a single candidate key i.e , user_id and it only a single attribute so there is no proper subset of this therefore there is no partial dependency. Hence this table is in 2nd normal form.

3NF :- In this table all functional dependencies are from candidate key(prime attribute) to non prime attributes . Therefore there is no transitive dependency. Hence this table is in 3rd normal form.

BCNF :- all functional dependencies are from superkey.
i.e user_id to all other attributes. Therefore this table is in BCNF.

3.Entity :- STORE

Attributes :-

- item_id
- item_name
- price

Functional dependencies :-

- item_id ➡ item_name
- item_id ➡ price

1NF :- this relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF :- In this table there is a single candidate key i.e , user_id and it only a single attribute so there is no proper subset of this therefore there is no partial dependency. Hence this table is in 2nd normal form.

3NF :- In this table all functional dependencies are from candidate key(prime attribute(item_id)) to non prime attributes(item_name,price) . Therefore there is no transitive dependency. Hence this table is in 3rd normal form.

BCNF :- all functional dependencies are from superkey
i.e item_id to all other attributes. Therefore this table is in BCNF.

4.Entity :- BUYS_FROM

Attributes :-

- user_id,item_id

There are no functional dependencies.
As there are no dependencies and all the attributes together form a primary key. So the table satisfies all normal forms conditions. Therefore its in 1,2,3,BC normal forms.

5.Entity :- PROBLEMS

Attributes :-

- prob_id
- prob_name
- prob_description

Functional dependencies :-

- prob_id ➡ prob_name
- prob_id ➡ prob_description

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3NF & BCNF :- In this table there are only 2 dependencies where a candidate key is determining other attributes. Therefore as the superkey is acting as determinant in all the dependencies , this table is in BCNF which means this is also in 2nd and 3rd normal forms.

6.Entity :- TOPIC

Attributes :-

- topic_name
- no_of_questions
- topic_difficulty_level

Functional dependencies :-

- topic_name ➡ no_of_questions

- topic_name ➡ topic_difficulty_level

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3NF & BCNF :- In this table there are only 2 dependencies where a candidate key is determining other attributes. Therefore as the superkey is acting as determinant in all the dependencies , this table is in BCNF which means this is also in 2nd and 3rd normal forms.

7.Entity :- BELONGS_TO

Attributes :-

- prob_id,topic_name

There are no functional dependencies.

As there are no dependencies and all the attributes together form a primary key. So the table satisfies all normal forms conditions. Therefore its in 1,2,3,BC normal forms.

8.Entity :- PRACTICE_PROBLEMS

Attributes :-

- prob_id
- p_d_level

Functional dependencies :-

- prob_id ➡ p_d_level

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3 NF & BCNF :- In this table there is only one dependency where a full candidate key is determining another attribute . therefore there is no chance of existence of partial dependency . hence this table satisfies all normal forms.

9.Entity :- CONTESTS

Attributes :

- c_id
- c_name
- prize
- organizer

Functional dependencies :-

- c_id ➡ c_name
- c_id ➡ prize
- c_id ➡ organizer

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3 NF & BCNF :- In this table there are only dependencies from primary key to all other keys which says that this table is in BCNF . Therefore this table is in 2nd 3rd normal form too.

10.Entity :- CONTEST_PROBLEMS

Attributes :-

- c_id,prob_id
- marks

Functional dependencies :-

- c_id,prob_id ➡ marks

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3 NF & BCNF :- in this table there is only one functional dependency in which the full candidate key is determining a non prime attribute. Which satisfies the BCNF condition. Therefore the table is in 1NF,2NF,BCNF.

11.Entity :- PARTICIPATES_IN

Attributes :-

- user_id
- c_id
- p_id

Functional dependencies :-

- user_id,c_id ➡ p_id
- p_id ➡ user_id,c_id

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3 NF & BCNF :- there are two candidate keys in this table . one is p_id and other is user_id + c_id as all dependencies are from one full candidate key to another so there is no existence of partial dependency or transitive dependency . Therefore the table is in 2NF,3NF and BCNF.

12.Entity :- RESULTS

Attributes :-

- p_id
- contest_rank
- finish_time ● score

Functional dependencies :-

- p_id ➡ contest_rank
- p_id ➡ finish_time
- p_id ➡ score

1NF :- This relation satisfies 1NF condition. Therefore it is in 1st normal form.

2NF & 3 NF & BCNF :- In this table there are only dependencies from primary key to all other keys which says that this table is in BCNF . Therefore this table is in 2nd 3rd normal form too.

13.Entity :- SOLVES

Attributes :-

- user_id,prob_id

There are no functional dependencies. As there are no dependencies and all the attributes together form a primary key. So the table satisfies all normal forms conditions. Therefore, its in 1,2,3,BC normal forms.

CREATING TABLES :-

1) STATE

```
Create table STATE (
    State varchar(20), Country
    varchar(20),
```

```
        primary key(state)
 );
```

2)CITY

```
Create table CITY
(
        City varchar(20), State varchar(20),
        primary key(city)
);
```

3)USER

```
Create table user
(
user_id varchar(10) ,
first_name varchar(20),
middle_name varchar(20),
last_name varchar(20),
email_id varchar(30),
password varchar(20),
city varchar(20) ,
primary key(user_id)
);
```

4)PROFILE

```
Create table Profile (
user_id varchar(10),
```

```
rating decimal(2,1),
rankk int,
max_streak int,
total_subs int,
Acc_subs int,
Primary key(user_id),
Foreign key(user_id) references user(user_id)
);
```

5)STORE

```
Create table STORE
(
Item_id int ,
Item_name varchar(20),
price int,
Primary key(item_id)
);
```

6) BUYS_FROM

```
Create table buys_from (
        Item_id int,
        User_id varchar(10),
        Primary key(user_id,item_id),
        Foreign key(user_id) references user(user_id),
        Foreign key(item_id) references store(item_id)
);
```

7)PROBLEMS

```
Create table Problems
(
Prob_id varchar (10)  ,
Prob_name varchar(50),
Prob_description varchar(300) ,
primary key (prob_id)
);
```

8) TOPIC

```
Create table Topic
(
        Topic_name varchar(30) Primary key,
        No_of_questions int,
        Diffuclty_level varchar(10)
);
```

9) BELONG_TO

```
Create table belong_to (
        Prob_id varchar(10),
        Topic_name varchar(30),
        Primary key(topic_name,prob_id),
        Foreign key (prob_id) references Problems(prob_id),
        Foreign key (topic_name) references topic(topic_name)
```

```
);
```

10) PRACTICE_PROBLEMS

```
Create table Practice_problems
(
        prob_id varchar(10)   not null primary key,
         p_d_level varchar(10) ,
        Foreign key (Prob_id) references Problems(Prob_id)
);
```

11) CONTESTS

```
Create table Contest
(
        c_id int not null primary key,
        c_name varchar(20),
        Prize int ,
        Organizer varchar(255)
);
```

12) CONTEST_PROBLEMS

```
Create table contest_problems (
        prob_id varchar(10) ,
        c_id int,
        Marks int,
        Primary key(Prob_id,C_id),
        Foreign key (Prob_id) references Problems(Prob_id), Foreign key (c_id) references
        contests(c_id)
);
```

13) PARTICIPATES_IN

```
Create table participates_in (
        c_id int,
        User_id varchar(10),
        P_id int primary key,
        Foreign key (c_id) references contests(c_id),
        Foreign key(user_id) references user(user_id)
);
```

14) RESULTS

```
Create table Results
(
        P_id int primary key,
        Contest_rank int,
        score int,
        finish_time datetime
);
```

15) SOLVES

```
Create table solves
(
User_id varchar(20),
Prob_id varchar(20),,
Primary key(user_id,prob_id),
Foreign key(user_id) references user(user_id),
Foreign key (Prob_id) references Problems(Prob_id)
)
```

Inserting data :

STATE :-

insert into STATE ("Telangana", "India") ,('Beijing', 'China'), ('Tokyo', 'Japan'), ("Tripura", "India") , ("Hassen", 'Germany') , ('Washington', 'America'), ('Islamabad', 'Pakistan') ,('Assam', 'India'), ('Shanghai', 'China'),('Geneva', 'Switzerland');

CITY:-

Insert into CITY ('HYD', 'Telangana') , ('Haidan', 'Beijing') , ('Machida', 'Tokyo') ,

('Agartala', 'Tripura'), ('Frankfurt', 'Hassen'), ('Seattle', 'Washington'), ('Wah', 'Islamabad'),

('Guwahati', 'Assam'), ('Suzhou', 'Shanghai'), ('Vessy', 'Geneva');

USER :-

insert into user

('m3g4n','meghana','kumari','pasikanti','meghanapasikanti@gmail.com','12345','HYD'),

('happy_1','bharath','kumar','reddy','bharathreddy@gmail.com','123456789', 'Machida'),

('asam_123','asam',null,'samba','asamsamba@gmail.com','test1', 'Haidan'),

('leo','sujit','das',null,'sujitdas@gmail.com','password','Agartala'),

('hema_123','hema','kumari','landa','hemalanda@gmail.com','12345678','Frankfurt'),

('sruth_56','sruthi','singh','pulusu','sruthipulusu@gmail.com','zinch','Seattl
e'),

('cyberpunk','ashok',null,'valasa','ashokvalasa@gmail.com','G_czechout'
,'Wah'),

('noob','rama',null,'rao','ramarao@gmail.com','asdf', 'Guwahati'),

('akhil','dinesh','thakur','reddy','dineshreddy@gmail.com','querty','Suzhou'
),

('snake_eyes','manoj','kumar','reddy','manojreddy@gmail.com','12345678
90','Vessy'),

('icecuber','kalyan','siddartha','induri','kalyaninduri@gmail.com','asdfghjkl'
,'Agartala');

PROFILE :-

insert into PROFILE values ('m3g4n', 5 , 1, 200, 15, 13),
('happy_1', 5 , 2, 228 , 33, 21), ('asam_123', 4.8 , 3, 10 , 15 , 10),
('leo', 4.7 , 4, 90 , 16, 12), ('hema_123', 4.6, 5 , 300 , 14, 8),
('sruth_56', 4.5 , 6, 100, 12, 2), ('cyberpunk',4.4 ,7 , 20,50 ,4),
('noob',4.3 ,8 ,5 ,10 ,7), ('akhil', 4.2,9 ,3 ,10 ,7),
('snake_eyes', 4.1,10,50 , 79, 7), ('icecuber', 4.0, 11, 27, 16,5);

STORE :-

insert into STORE values(1, 'bag',1000), (2, 'bottle', 800),
(3, 't-shirt', 500), (4, 'bag', 700), (5, 't-shirt', 600);

BUYS_FROM :-

insert into buys_from values (2, 'm3g4n'), (5, 'happy_1'), (4, 'cyberpunk'),(3, 'snake_eyes'), (1, 'cyberpunk'), (4, 'noob'), (3, 'asam_123');

PROBLEMS :-

insert into problems values

('CODA01','Welcome world.','pd1'),

('CODA02', 'Beautiful String','pd2'),

('CODA03', 'Minimum no of operations','pd3'),

('ACMU01', 'Am i prime?','pd4'),

('ACMU02', 'Nth tribonacci number','pd5'),

('ACMU03', 'Kth largest number','pd6'),

('ACMU04', 'Valid string','pd7'),

('CDR01', 'XOR array','pd8'),

('CDR02', 'Deepest nodes sum','pd9'),

('CDR03', 'Is graph bipartite?','pd10'),

('CODW01', 'Palindrome check','pd11'),

('CODW02', 'Sliding Window','pd12'),

('DZER01', 'Reverse string','pd21'),

('DZER02', 'Spiral Order','pd13'),

('DZER03', 'Tower of hanoi','pd14'),

('KCR01', 'Water Bottles','pd15'),

('KCR02', 'LCM problem','pd16'),

('KCR03', 'Smallest path','pd17'),

('CODE01', 'Binary search','pd18'),

('CODE02', 'Next Permutation','pd19'),

('CODE03', 'Range Queries.','pd20') ;

TOPIC :-

Insert into topic values ('Arrays',3 ,'easy'), (' Binary search', 3,'easy'),
('Strings',3 ,'easy'), ('Stacks',2 ,'easy'),('Sorting',3,'medium'),
('Queue',2 ,'easy'), ('Two Pointers',2 ,'medium'),
('Depth First search',2 ,'medium'), ('Greedy', 3,'medium'),
('Recursion',3 ,'hard'), ('Back tracking',3 ,'hard'),
('Bitmanipulation',2,'medium'), ('Hashing',2 ,'hard'), ('Graph',1
,'hard'),('Binary Trees', 2,'hard'), ('Linked Lists',1 ,'medium'),
('Dynamic programming',1 ,'hard');

BELONG_TO :-

Insert into belong_to

values('CODA02' , 'Arrays');

('KCR02' , 'Arrays');

('KCR03' , 'Arrays');

('KCR01' , ' Binary search');

('KCR02' , ' Binary search');

('CODA01' , ' Binary search');

('CODA03' , 'Strings'),

('CDR02' , 'Strings'), ('CDR03' , 'Strings'),('ACMU04' , 'Sorting'), ('ACMU02' , 'Sorting'),

('CODA02' , 'Stacks'),('ACMU03' , 'Stacks'),('ACMU02' , 'Two Pointers'),

('ACMU01' , 'Two Pointers'),

('DZER02' , 'Depth First search'),('KCR03' , 'Depth First search'),('DZER02' , 'Greedy'),

('KCR01' , 'Greedy'),('DZER03' , 'Greedy'),('DZER03' , 'Recursion'),

('DZER01' , 'Recursion'),('CODW02' , 'Recursion'),('CODA02' , 'Back tracking'),

('CODE03' , 'Back tracking'),('CODA03' , 'Back tracking'),

('ACMU02' , 'Bitmanipulation'),('CDR03' , 'Bitmanipulation'),('CODA02' , 'Hashing'),

('ACMU03' , 'Hashing'),('ACMU04' , 'Binary Trees'),('DZER03' , 'Binary Trees'),

('KCR02' , 'Graph'),('CODE01' , 'Linked Lists'),('CODE03' , 'Dynamic programming');

PRACTICE_PROBLEMS :-

```
insert into practice_problems
values('CODA01','easy'),
('CODA02','medium'), ('CODA03','hard'),
('ACMU01','easy'), ('ACMU02','easy'),
('ACMU03','medium'),
('ACMU04','medium'), ('CDR01','easy'),
('CDR02','easy'), ('CDR03','hard'),
('CODW01','medium'),
('CODW02','hard'), ('DZER01','easy'),
('DZER02','medium'), ('DZER03','hard'),
('KCR01','easy'),

('KCR02','hard'), ('KCR03','medium'),
('CODE01','medium'), ('CODE02','hard'),
('CODE03','easy');
```

CONTEST :-

```
Insert into contests

values(0001,'Codeathon 3.o',1500 ,'CSEA'),

(0002,'ACM Unlock', 1200,'ECEA'),

(0003,'Coderbyte',2000 ,'IIIH'),

(0004,'Code wars 4.0',1000 ,'Harbour Space Univer sity'),

(0005,'Divide By zero',10000 ,'Harvard University'),

(0006,'Kick Start',5000 ,'CodeNation'),

(0007,'Hash Code',2500 ,'CSEA');
```

CONTEST_PROBLEMS :-

```
insert into contest_problems

values ('CODA01',0001 , 100),
```

('CODA02', 0001, 200),

('CODA03', 0001, 300),

('ACMU01',0002 ,100),

('ACMU02',0002 ,150),

('ACMU03', 0002,150),

('ACMU04', 0002,200),

('CDR01', 0003, 150),

('CDR02',0003 , 150),

('CDR03', 0003, 200),

('CODW01', 0004,200), ('CODW02',0004 , 300),

('DZER01',0005 ,400),

('DZER02',0005 ,300),

('DZER03',0005 ,400),

('KCR01',0006 ,200),

('KCR02', 0006, 150),

('KCR03',0006 , 350),

('CODE01',0007 ,400),

('CODE02',0007 ,150),

('CODE03',0007 , 300) ;

PARTICIPATES_IN :-

Insert into participates_in

(0001 ,'m3g4n',2024001),

(0001,'happy_1',2024002),

(0001 ,'asam_123',2024003),

(0002 ,'happy_1',2024004),

(0002 ,'m3g4n',2024005),

(003,'asam_123',2024006),

(0003,'leo',2024007),

(0003 ,'hema_123',2024008),

(0004,'hema_123',2024009),
(0004,'cyberpunk',2024010),

(0004,'icecuber',2024011),

(0005,'m3g4n',2024012),

(0005,'happy_1',2024013),

(0005,'noob',2024014),

(0005,'leo',2024015),

(0005 ,'icecuber',2024016),

(0005,'snake_eyes',2024017),
(0006,'hema_123',2024018),

(0006,'sruth_56',2024019),

(0006 ,'m3g4n',2024020),

(0007 ,'akhil',2024021),

(0007,'asam_123',2024022),

(0007,'noob',2024023),

(0007,'cyberpunk',2024024);

RESULTS :-

Insert into Results

(2024001, 1,600 , '2020-08-23 12:45:03'),

(2024002, 2,400 , '2024-03-23 21:50:43'),

(2024003, 3,350 , '2024-03-23 08:00:03'),

(2024004,1 , 400, '2023-03-22 05:46:03'),

(2024005 ,2 ,250 , '2023-03-20 05:46:03'),

(2024006 , 3,450 , '2023-10-22 05:00:45'),

(2024007 ,2 ,350 , '2023-10-22 12:13:33'),

(2024008 ,1,450 , '2023-10-22 05:25:41'),

(2024009 , 1, 600, '2018-04-15 04:41:45'),

(2024010 , 2,750 , '2018-04-16 05:41:45'),

(2024011, 3,400 , '2018-04-17 04:25:45'),

(2024012,1 , 700, '2024-04-15 15:42:45'),

(2024013, 2,500 , '2024-04-19 19:19:19'),

(2024014 , 3,550 , '2024-04-13 13:31:45'),

(2024015 ,4 ,600 , '2024-04-15 14:41:00'),

(2024016 ,5 ,650 , '2024-04-17 23:54:45'),

(2024017,6 ,400 , '2024-04-15 04:32:15'),

(2024018,1 ,350, '2017-02-15 21:32:15'),

(2024019,2 ,450, '2017-02-15 04:39:15'),

(2024020 ,3 ,550 , '2017-02-15 13:52:15'),

(2024021,1 ,500 , '2023-07-15 14:05:00'),

(2024022,2 , 450, '2023-07-14 02:05:00'),

(2024023,3 ,450 , '2023-07-16 09:09:03'),

(2024024,4 ,300 , '2023-07-14 00:00:07');

SOLVES:-

insert into solves

values
('leo','ACMU01'),('akhil','ACMU02'),('hema_123','CODE01'),('noob','CODE0
2'),('asam_123','KCR03'),('happy_1','KCR03'),('cyberpunk','KCR02'),('hema_
123','KCR01'),('m3g4n','DZER03'),('icecuber','DZER01');

RELATIONAL SCHEME :