

Dhirubhai Ambani Institute of Information and Communication Technology



IT214 – Database Management System

Database to manage online competitive

Coding Platform

Lab Group - 5 : Team - 2 (G5-T2)

Group Members :-

Representative: Karthik Raman Balamurugan - 202301407

(Contact Number: +91 7418878768)

Hriday Asnani - 202301282

Harshil Vithlani - 202301431

L Venkata Subramanyam - 202301280

This Submission contains the ER Diagram, Relational Schema and the BCNF Normalization Proofs along with the minimal FD set

Submitted alongside in separate textfiles are:

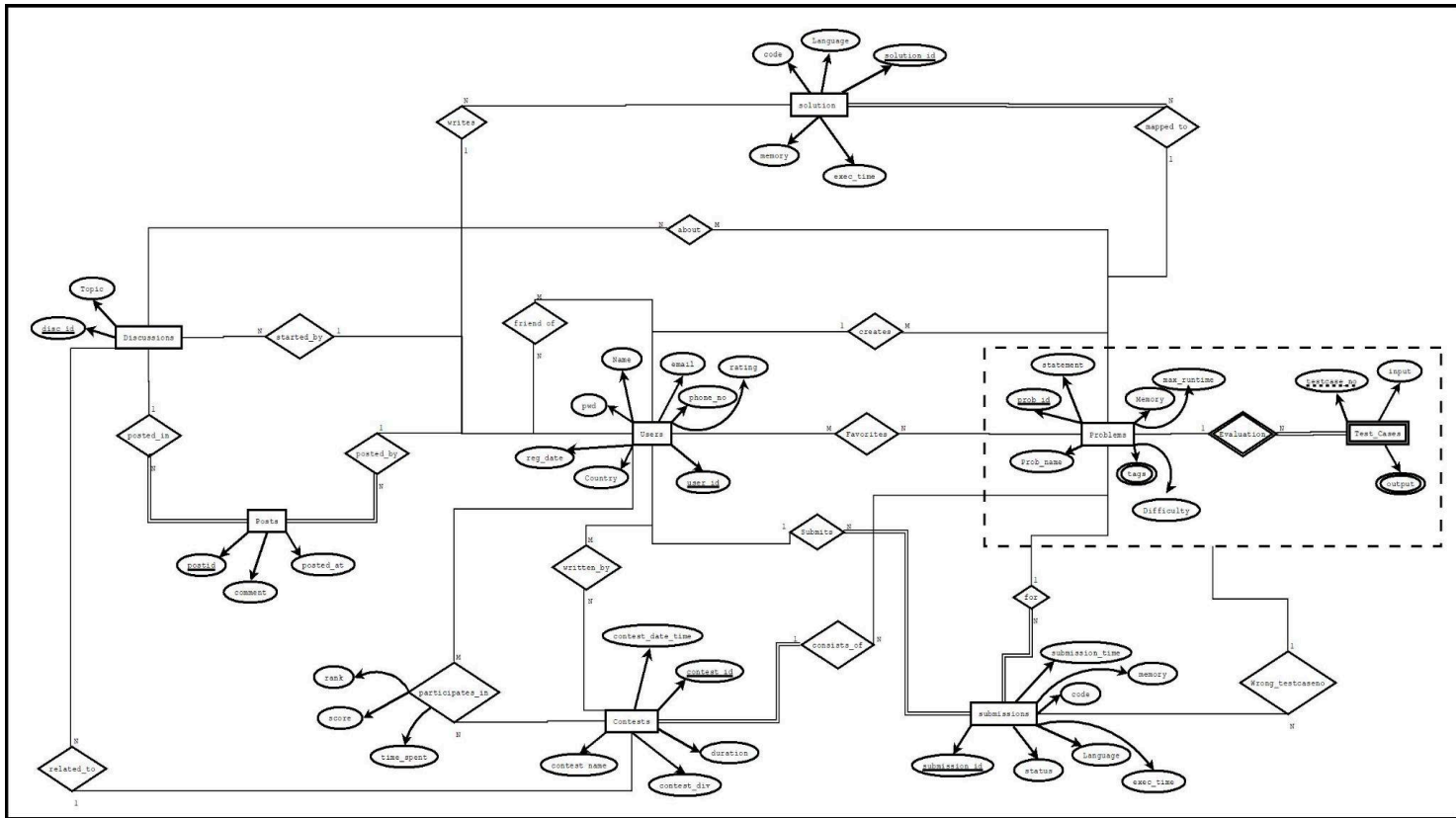
- DDL Script
- INSERT statements' script
- Queries' SQL Statements

Special Note : Our project only deals with the data storing and retrieval aspects (database management) of the online coding platform, i.e., the area concerned with our course. However, we have not looked into the running and evaluation of codes, since it is outside the scope of DBMS.

We are considering that our project requires and depends on an external compiler/IDE to compile codes, run them, compare them and evaluate them, in order to make it an actual complete competitive coding platform. Our project is treating all these aspects (program codes and others) as simple data provided by the compiler software.

Nevertheless, we can assure that we have dealt with the portion concerning DBMS (our course) to the best of our knowledge and hard work to get as close as we can to an actual platform.

(1) Entity Relationship - ER Diagram



Attached is the google photos link to view the entire image of the ER diagram clearly, as it was difficult to fit the large diagram here in portrait : [ER Image](#)

```

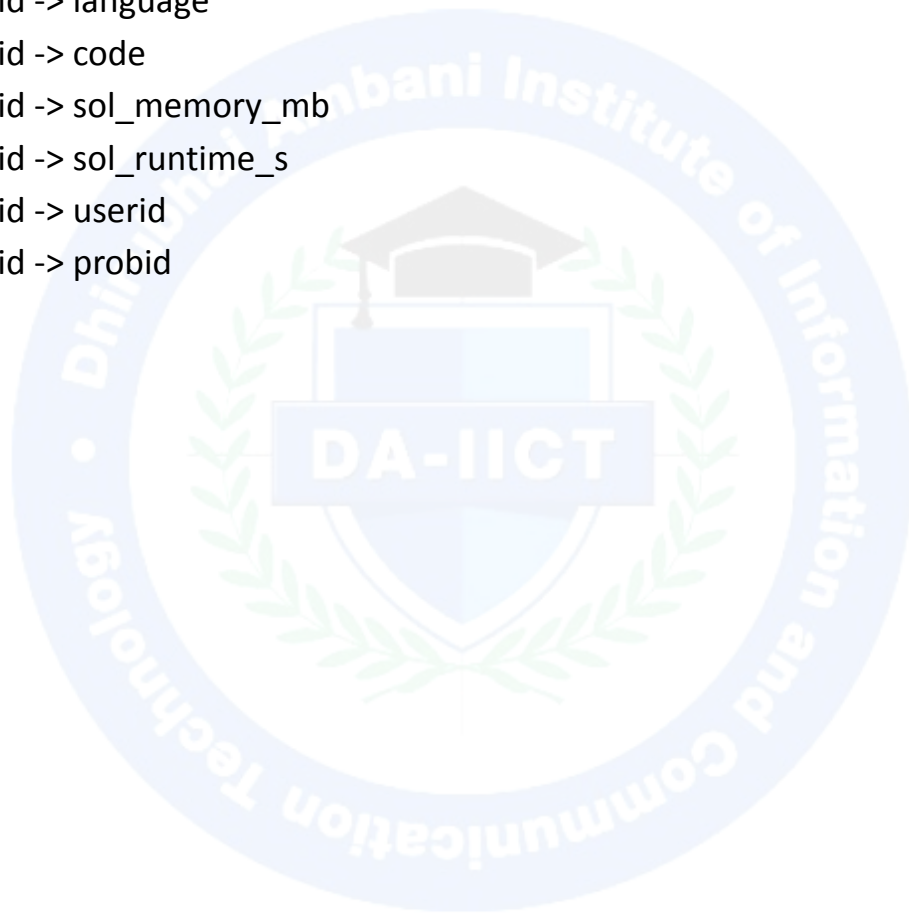
    erDiagram
        Contest ||--o{ User : "participates in"
        Contest ||--o{ Problems : "Written by"
        Contest ||--o{ Solutions : "Favourites"
        Contest ||--o{ Discussions : "Discussions"
        Contest ||--o{ Submissions : "Submissions"
        Contest ||--o{ Posts : "Posts"
        Contest ||--o{ Friend_of : "Friend of"
        Contest ||--o{ Favourites : "Favourites"
        Contest ||--o{ Test_cases : "Test cases"
        Contest ||--o{ Output : "Output"
        Contest ||--o{ tags : "tags"
        Contest ||--o{ About : "About"
        User ||--o{ Problems : "Problems"
        User ||--o{ Solutions : "Solutions"
        User ||--o{ Discussions : "Discussions"
        User ||--o{ Submissions : "Submissions"
        User ||--o{ Posts : "Posts"
        User ||--o{ Friend_of : "Friend of"
        User ||--o{ Favourites : "Favourites"
        User ||--o{ Test_cases : "Test cases"
        User ||--o{ Output : "Output"
        User ||--o{ tags : "tags"
        User ||--o{ About : "About"
        Problems ||--o{ Solutions : "Solutions"
        Problems ||--o{ Discussions : "Discussions"
        Problems ||--o{ Submissions : "Submissions"
        Problems ||--o{ Posts : "Posts"
        Problems ||--o{ Friend_of : "Friend of"
        Problems ||--o{ Favourites : "Favourites"
        Problems ||--o{ Test_cases : "Test cases"
        Problems ||--o{ Output : "Output"
        Problems ||--o{ tags : "tags"
        Problems ||--o{ About : "About"
        Solutions ||--o{ Discussions : "Discussions"
        Solutions ||--o{ Submissions : "Submissions"
        Solutions ||--o{ Posts : "Posts"
        Solutions ||--o{ Friend_of : "Friend of"
        Solutions ||--o{ Favourites : "Favourites"
        Solutions ||--o{ Test_cases : "Test cases"
        Solutions ||--o{ Output : "Output"
        Solutions ||--o{ tags : "tags"
        Solutions ||--o{ About : "About"
        Discussions ||--o{ Submissions : "Submissions"
        Discussions ||--o{ Posts : "Posts"
        Discussions ||--o{ Friend_of : "Friend of"
        Discussions ||--o{ Favourites : "Favourites"
        Discussions ||--o{ Test_cases : "Test cases"
        Discussions ||--o{ Output : "Output"
        Discussions ||--o{ tags : "tags"
        Discussions ||--o{ About : "About"
        Submissions ||--o{ Posts : "Posts"
        Submissions ||--o{ Friend_of : "Friend of"
        Submissions ||--o{ Favourites : "Favourites"
        Submissions ||--o{ Test_cases : "Test cases"
        Submissions ||--o{ Output : "Output"
        Submissions ||--o{ tags : "tags"
        Submissions ||--o{ About : "About"
        Posts ||--o{ Friend_of : "Friend of"
        Posts ||--o{ Favourites : "Favourites"
        Posts ||--o{ Test_cases : "Test cases"
        Posts ||--o{ Output : "Output"
        Posts ||--o{ tags : "tags"
        Posts ||--o{ About : "About"
        Friend_of ||--o{ Favourites : "Favourites"
        Friend_of ||--o{ Test_cases : "Test cases"
        Friend_of ||--o{ Output : "Output"
        Friend_of ||--o{ tags : "tags"
        Friend_of ||--o{ About : "About"
        Favourites ||--o{ Test_cases : "Test cases"
        Favourites ||--o{ Output : "Output"
        Favourites ||--o{ tags : "tags"
        Favourites ||--o{ About : "About"
        Test_cases ||--o{ Output : "Output"
        Test_cases ||--o{ tags : "tags"
        Test_cases ||--o{ About : "About"
        Output ||--o{ tags : "tags"
        Output ||--o{ About : "About"
        tags ||--o{ About : "About"
        About ||--o{ Friend_of : "Friend of"
        About ||--o{ Favourites : "Favourites"
        About ||--o{ Test_cases : "Test cases"
        About ||--o{ Output : "Output"
        About ||--o{ tags : "tags"
        About ||--o{ About : "About"
  
```

- **Remark** : We have made a few considerable changes to our ER Diagram and Relational Schema since the initial submission. We are hereby submitting the final versions.

(3) Minimal FD Set

userid -> Name
userid -> email
userid -> phone_no
userid -> rating
userid -> pwd
userid -> reg_date
userid -> country
probid -> statement
probid -> max_memory_mb
probid -> max_runtime_s
probid -> difficulty
probid -> prob_name
probid -> creator_id
probid -> contestid
{probid, testcaseno} -> input
discid -> topic
postid -> comment
postid -> posted_at
postid -> userid
postid -> discid
contestid -> contest_name
contestid -> duration_hrs
contestid -> contest_div
contestid -> contest_date_time
{userid , contestid } -> rank
{userid , contestid } -> score
{userid , contestid } -> time_spent_hrs
subid -> submission_time

subid -> code
subid -> memory_s
subid -> status
subid -> language
subid -> runtime_s
subid -> userid
subid -> probid
subid -> wrong_testcaseno
solutionid -> language
solutionid -> code
solutionid -> sol_memory_mb
solutionid -> sol_runtime_s
solutionid -> userid
solutionid -> probid



(4) Proof that relations are in BCNF

1. 'Users' relation :

- Attributes :

Users {Userid, Name, email, phone_no, rating, pwd, reg_date, country}

- Functional dependencies :

userid -> Name

userid -> email

userid -> phone_no

userid -> rating

userid -> pwd

userid -> reg_date

userid -> country

Let X = User_id

$X^+ = \{\text{userid, Name, email, phone_no, rating, pwd, reg_date, country}\}$

Thus, **Primary key = userid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Users' is userid, which is the primary key of this relation, so "Users" is in **BCNF**.

2. 'Friend_of' relation:

- Attributes:

Friend_of {userid, friendid}

- The only functional dependency existing here is the trivial one:

{userid, friendid} -> {userid, friendid}

Thus, the entire set of attributes, i.e, {userid, friendid} is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "Friend_of" is in **BCNF**.

3. 'Problems' relation :

- Attributes :

Problems {probid, statement, memory, max_runtime, difficulty, prob_name, tags, creator_id, contestid}

- Functional dependencies :

probid -> prob_name
probid -> statement
probid -> max_memory_mb
probid -> max_runtime_s
probid -> difficulty
probid -> tags
probid -> creator_id
probid -> contestid

Let X = probid

$X^+ = \{\text{probid, statement, max_memory_mb, max_runtime_s, difficulty, prob_name, tags, creator_id, contestid}\}$

Thus, **Primary key = probid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Problems' is probid , which is the primary key of this relation, so "Problems" **is in BCNF**.

4. 'Tags' Relation:

- Attributes:

Friend_of {probid, tagname}

- The only functional dependency existing here is the trivial one:

{probid, tagname} -> {probid, tagname}

Thus, the entire set of attributes, i.e, **{probid, tagname}** is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "Tags" **is in BCNF**.

5. 'Favorites' Relation:

- Attributes:
Friend_of {userid, probid}
- The only functional dependency existing here is the trivial one:
 $\{userid, probid\} \rightarrow \{userid, probid\}$

Thus, the entire set of attributes, i.e, **{userid, probid}** is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "Favorites" is in **BCNF**.

6. 'Test_cases' relation :

- Attributes :
Test_cases {probid, testcaseno, input}
- Functional dependencies :
 $\{probid, testcaseno\} \rightarrow input$

Let $X = \{probid, testcaseno\}$

$X^+ = \{probid, testcaseno, input\}$

Thus, **Primary key = {probid, testcaseno}**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Test_cases' is {probid, testcaseno} , which is the primary key of this relation, so "Test_cases" is in **BCNF**.

7. 'Output' Relation:

- Attributes:
Friend_of { probid, testcaseno, output}
- The only functional dependency existing here is the trivial one:
 $\{probid, testcaseno, output\} \rightarrow \{probid, testcaseno, output\}$

Thus, the entire set of attributes, i.e, **{probid, testcaseno, output}** is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "Output" **is in BCNF**.

8. 'Discussions' relation :

- Attributes :
Discussions {discid, topic}
- Functional dependencies :
 $discid \rightarrow topic$

Let $X = discid$

$X^+ = \{discid, topic\}$

Thus, **Primary key = discid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation Discussions is discid , which is the primary key of this relation, so "Discussions" **is in BCNF**.

9. 'About' Relation:

- Attributes:
Friend_of { discid,probid}
- The only functional dependency existing here is the trivial one:
 $\{discid,probid\} \rightarrow \{discid,probid\}$

Thus, the entire set of attributes, i.e, **{discid, probid}** is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "About" is in **BCNF**.

10. 'Posts' Relation:

- Attributes :
Posts {postid, comment, posted_at, userid, discid}
- Functional dependencies :
postid \rightarrow comment
postid \rightarrow posted_at
postid \rightarrow userid
postid \rightarrow discid

Let $X = \text{postid}$

$X^+ = \{\text{postid, comment, posted_at, userid, discid}\}$

Thus, **Primary key = postid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Posts' is postid , which is the primary key of this relation, so "Posts" is in **BCNF**.

11. 'Contests' relation :

- Attributes :

Contests {contestid, contest_name, contest_div, duration, contest_date_time}

- Functional dependencies :

contestid \rightarrow contest_name

contestid \rightarrow contest_div

contestid \rightarrow duration_hrs

contestid \rightarrow contest_date_time

Let $X = \text{contestid}$

$X^+ = \{\text{contest_name, contest_div, duration_hrs, contest_date_time}\}$

Thus, **Primary key = contestid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Contests' is contestid , which is the primary key of this relation, so "Contests" is in **BCNF**.

12. 'Participates_in' relation :

- Attributes :

Participates_in {userid, contestid, rank, score, time_spent_hrs}

- Functional dependencies :

{userid, contestid} \rightarrow rank

{userid, contestid} \rightarrow score

{userid, contestid} \rightarrow time_spent_hrs

Let $X = \{\text{userid, contestid}\}$

$X^+ = \{\text{userid, contest_id, rank, score, time_spent_hrs}\}$

Thus, **Primary key = {userid, contestid}**

The left side of all the FDs in the minimal set of FDs for the relation 'Participates_in' is {userid, contestid}, which is the primary key of this relation, so "Participates_in" is in **BCNF**.

13. 'Written_by' Relation:

- Attributes:
Friend_of { userid, contestid }
- The only functional dependency existing here is the trivial one:
 $\{userid, contestid\} \rightarrow \{userid, contestid\}$

Thus, the entire set of attributes, i.e, {userid, contestid} is the primary key and the relation complies with BCNF as it is not violating any of its rules.

We can see that the left side of the only FD in the minimal FD set for this relation is equivalent to the Primary Key itself, so "Written_by" is in BCNF.

14. 'Submissions' relation :

- Attributes :
Submissions {subid, code, memory_mb, submission_time, status, language, runtime_s, wrong_testcaseno, userid, probid}
- Functional dependencies :
subid \rightarrow submission_time
subid \rightarrow code
subid \rightarrow memory_mb
subid \rightarrow status
subid \rightarrow language
subid \rightarrow runtime_s
subid \rightarrow userid
subid \rightarrow probid
subid \rightarrow wrong_testcaseno

Let $X = \text{subid}$

$X^+ = \{\text{subid, code, memory_mb, submission_time, status, language, runtime_s, wrong_testcaseno, userid, probid}\}$

Thus, **Primary key = subid**

We can see that the left side of all the FDs in the minimal set of FDs for the relation 'Submissions' is **subid** , which is the primary key of this relation, so “Submissions” is in **BCNF**.



15. 'Solution' relation :

- Attributes :
Solutions {solutionid, code, language, sol_memory_mb, sol_runtime_s , userid, probid}
- Functional dependencies :
solutionid -> language
solutionid -> code
solutionid -> sol_memory_mb
solutionid -> sol_runtime_s
solutionid -> userid
solutionid -> probid

Let X = solutionid

$X^+ = \{\text{solutionid, code, language, sol_memory_mb, sol_runtime_s, userid, probid}\}$

Thus, **Primary key = solutionid**

The left side of all the FDs in the minimal set of FDs for the relation 'Solutions' is **solutionid**, which is the primary key of this relation, so "Solutions" is in **BCNF**.

→ We have added the screenshots of outputs for initial few queries here in the report file to display their functioning and efficiency.
The accompanied text file contains the entire list of query statements.

- **List of Queries for the Coding Platform database : -**

1. List and details of all platform users in descending order of rating:

SELECT userid, name, rating, email, phone_no, reg_date, country FROM users ORDER BY rating DESC;

Output :

| | userid [PK] integer | name character varying (18) | rating integer | email character varying (30) | phone_no character varying (16) | reg_date timestamp without time zone | country character varying (20) |
|----|------------------------|--------------------------------|-------------------|---------------------------------|------------------------------------|-----------------------------------------|-----------------------------------|
| 1 | 4 | Diana | 1600 | diana@example.com | 4567890123 | 2023-07-12 00:00:00 | USA |
| 2 | 12 | Lalit | 1600 | lalit@sample.in | 9567801234 | 2025-03-01 00:00:00 | India |
| 3 | 7 | Gautam | 1550 | gautam@sample.in | 9012345678 | 2024-08-09 00:00:00 | India |
| 4 | 10 | Jai | 1500 | jai@sample.in | 9345678012 | 2025-02-14 00:00:00 | India |
| 5 | 1 | Alice | 1500 | alice@example.com | 1234567890 | 2024-12-08 00:00:00 | USA |
| 6 | 11 | Kiran | 1450 | kiran@sample.com | 9456780123 | 2023-11-03 00:00:00 | USA |
| 7 | 6 | Fiona | 1400 | fiona@example.com | 6789012345 | 2024-10-31 00:00:00 | Germany |
| 8 | 9 | Isha | 1350 | isha@sample.in | 9234567801 | 2024-12-25 00:00:00 | India |
| 9 | 2 | Bob | 1300 | bob@example.com | 2345678901 | 2024-11-15 00:00:00 | Canada |
| 10 | 8 | Harsha | 1250 | harsha@sample.in | 9123456780 | 2024-09-30 00:00:00 | India |
| 11 | 3 | Charlie | 1200 | charlie@example.com | 3456789012 | 2025-01-05 00:00:00 | UK |
| 12 | 5 | Evan | 1100 | evan@example.com | 5678901234 | 2024-03-20 00:00:00 | Australia |

2. List of winners (participants with rank-1) for all contests:

SELECT c.contest_name, c.contest_div AS division, u.name AS winner_name, u.email, pi.score, pi.time_spent_hrs FROM contests c JOIN participates_in pi ON c.contestid = pi.contestid JOIN users u ON pi.userid = u.userid WHERE pi.rank = 1 ORDER BY c.contest_date_time DESC;

Output :

| | contest_name text | division integer | winner_name character varying (18) | email character varying (30) | score integer | time_spent_hrs integer |
|---|----------------------|---------------------|---------------------------------------|---------------------------------|------------------|---------------------------|
| 1 | New Year Sprint | 1 | Alice | alice@example.com | 99 | 2 |
| 2 | Winter Bash | 3 | Harsha | harsha@sample.in | 92 | 2 |
| 3 | Summer Showd... | 4 | Jai | jai@sample.in | 90 | 2 |
| 4 | Tech Fiesta | 2 | Diana | diana@example.com | 93 | 2 |
| 5 | Autumn Cup | 2 | Evan | evan@example.com | 88 | 1 |
| 6 | Spring Challenge | 1 | Alice | alice@example.com | 100 | 2 |

3. LEADERBOARD DISPLAY for any given contest (say contestid = 1):

```
SELECT u.userid, u.name, u.rating, pi.rank, pi.score, pi.time_spent_hrs,
c.contest_name FROM users u JOIN participates_in pi ON u.userid = pi.userid JOIN
contests c ON pi.contestid = c.contestid WHERE pi.contestid = 1 ORDER BY pi.rank
ASC;
```

Output :

Data OutputMessagesNotifications

SQL

Showing r

| | userid integer | name character varying (18) | rating integer | rank integer | score integer | time_spent_hrs integer | contest_name text |
|----|-------------------|--------------------------------|-------------------|-----------------|------------------|---------------------------|----------------------|
| 1 | 1 | Alice | 1500 | 1 | 100 | 2 | Spring Challenge |
| 2 | 2 | Bob | 1300 | 2 | 95 | 2 | Spring Challenge |
| 3 | 3 | Charlie | 1200 | 3 | 90 | 2 | Spring Challenge |
| 4 | 4 | Diana | 1600 | 4 | 85 | 2 | Spring Challenge |
| 5 | 7 | Gautam | 1550 | 5 | 80 | 2 | Spring Challenge |
| 6 | 8 | Harsha | 1250 | 6 | 75 | 2 | Spring Challenge |
| 7 | 9 | Isha | 1350 | 7 | 70 | 2 | Spring Challenge |
| 8 | 10 | Jai | 1500 | 8 | 65 | 2 | Spring Challenge |
| 9 | 11 | Kiran | 1450 | 9 | 60 | 2 | Spring Challenge |
| 10 | 12 | Lalit | 1600 | 10 | 55 | 2 | Spring Challenge |

4. List of all problems with their status of submissions:

```
SELECT p.probid, p.prob_name, s.subid, s.status, s.wrong_testcaseno AS
rejected_testcase, u.name AS submitter_name, s.submission_time FROM
problems p LEFT JOIN submissions s ON p.probid = s.probid LEFT JOIN users u ON
s.userid = u.userid ORDER BY p.probid ASC, s.subid ASC;
```

Data OutputMessagesNotifications

SQL

Showing rows: 1 to 21Page N

| | probid integer | prob_name character varying (50) | subid integer | status character varying (20) | rejected_testcase integer | submitter_name character varying (18) | submission_time timestamp without time zone |
|----|-------------------|-------------------------------------|------------------|----------------------------------|------------------------------|------------------------------------------|------------------------------------------------|
| 1 | 101 | Sum of Two Numbers | 401 | Accepted | [null] | Alice | 2025-02-03 18:30:52 |
| 2 | 101 | Sum of Two Numbers | 407 | Accepted | [null] | Gautam | 2025-04-09 12:05:50 |
| 3 | 102 | Reverse a String | 402 | Wrong_Answer | 2 | Bob | 2024-12-28 09:15:30 |
| 4 | 102 | Reverse a String | 408 | Accepted | [null] | Harsha | 2025-01-10 13:15:25 |
| 5 | 102 | Reverse a String | 411 | Compile_Error | [null] | Bob | 2025-01-13 11:40:32 |
| 6 | 102 | Reverse a String | 412 | Compile_Error | [null] | Bob | 2025-01-14 00:15:30 |
| 7 | 103 | Find Graph Path | 403 | TLE | [null] | Charlie | 2025-04-05 10:20:00 |
| 8 | 103 | Find Graph Path | 404 | Wrong_Answer | 1 | Diana | 2025-04-06 14:05:15 |
| 9 | 103 | Find Graph Path | 409 | Wrong_Answer | 1 | Isha | 2025-04-11 08:55:10 |
| 10 | 103 | Find Graph Path | 413 | Accepted | [null] | Kiran | 2025-04-15 12:25:50 |
| 11 | 104 | Sort an Array | 405 | Accepted | [null] | Evan | 2025-02-12 16:10:45 |
| 12 | 104 | Sort an Array | 410 | Accepted | [null] | Jai | 2025-04-12 09:45:00 |

5. Average Rating of users per country;

SELECT country, AVG(rating) AS avg_rating, COUNT(*) AS total_users FROM users
GROUP BY country;

Data OutputMessagesNotifications

SQL

| | country character varying (20) | avg_rating numeric | total_users bigint |
|---|-----------------------------------|-----------------------|-----------------------|
| 1 | Australia | 1100.0000000000000000 | 1 |
| 2 | UK | 1200.0000000000000000 | 1 |
| 3 | USA | 1516.6666666666666667 | 3 |
| 4 | Germany | 1400.0000000000000000 | 1 |
| 5 | Canada | 1300.0000000000000000 | 1 |
| 6 | India | 1450.0000000000000000 | 5 |

END

- We are extremely grateful to our professor - PM Jat Sir for giving us the opportunity to work on such an enriching project and also making us equipped with the requisite set of skills and knowledge throughout the semester. We would also like to extend this gratitude to our respected TAs for their guidance and support during the entire span of the project.



Thank You !!!