

USER

- Problems of certain tags and difficulty

```
SELECT * FROM
    (SELECT * FROM PROBLEMS where Rating_Difficulty =
'Hard')result_query1
JOIN
    (SELECT * FROM PROBLEMS_TAGS where Tag =
'graphs')result_query2
ON result_query1.Problem_ID = result_query2.Problem_ID
```

- Solved and unsolved problems

```
use `cp-stats`;
Select rq4.Problem_ID, rq4.`Name`, rq4.Rating_Difficulty
from
    (Select Problems.Problem_ID, `Name`, Rating_Difficulty
from Problems where rating_difficulty= 'Hard')rq4
left join
    (select rq1.problem_id from(SELECT Problems.Problem_ID
FROM PROBLEMS where rating_difficulty= 'Hard')rq1
JOIN(SELECT * FROM SOLVED where USER_ID = 'U-1')rq2 ON
rq1.PROBLEM_ID = rq2.problem_id)rq3
on rq4.Problem_ID = rq3.Problem_ID
where rq3.Problem_ID is Null
```

- Repos

- a. Give me repos of certain tag

```
SELECT * FROM
REPO_TEMPLATES
JOIN
    (SELECT * FROM REPO_TAGS where Tag in ('greedy',
'dp'))result_query1
ON REPO_TEMPLATES.name=result_query1.name
```

- b. Give me repos of certain language

```
SELECT * FROM
REPO_TEMPLATES
WHERE Template_Language = 'C++'
```

- c. Update repo content

```
Update repo_templates set Template_Content='xyz has abc'
where ID='U-1' and `Name`='kupang' and
Template_Name='Giant Pizza' and Template_Language='C++';
```

- Blogs
 - a. Give me blogs of a certain tag


```
SELECT * FROM
BLOGS
JOIN
(SELECT * FROM BLOGS_TAGS where Tag= 'dp')result_query1
ON BLOGS.ID = result_query1.ID
```
 - b. Give me the highest liked blogs of a certain tag


```
SELECT * FROM blogs WHERE (ID, `Name`, `Date`, 'greedy')
IN (SELECT * FROM blogs_tags) ORDER BY likes DESC;
```
- Check if I have been preferred (premium)


```
SELECT ID, IF(ID IN (select User_ID from preferred), 'Yes',
'No') AS Preferred FROM premium_users;
```
- List of my groups along with role & date of joining


```
SELECT * FROM `cp-stats`.groups Inner JOIN (SELECT * FROM
MEMBER_OF WHERE ID = 'U-1')result_query1 ON `groups`.Group_ID =
result_query1.Group_ID;
```
- List groups by avg rating desc


```
select Group_ID, AVG(Rating) as AvgRating from member_of,
registered where member_of.ID = registered.User_ID group by
Group_ID order by AvgRating DESC;
```
- List other members of my group


```
SELECT * FROM MEMBER_OF WHERE Group_ID =$GROUP_ID, ID!=$ID
```
- List my following


```
SELECT * FROM
USERS
JOIN
(SELECT * FROM FOLLOWING WHERE User_ID=$ID)result_query1
ON USERS.ID = result_query1.Following_ID
```
- Sort my followers on basis of popularity


```
select User_ID, Followers from num_followers right join (select
* from following where Following_ID='U-147')res on res.User_ID
= num_followers.ID order by Followers desc;
```

GROUP ADMIN

- Sort by group members on the basis of popularity (followers)

```
SELECT Group_ID, member_of.ID, Followers FROM
`cp-stats`.member_of, num_followers where member_of.ID =
num_followers.ID and Group_ID=1 order by Followers desc;
```

- *Sort by group members on the basis of ratings*

```
SELECT Group_ID, User_ID, Rating, Handle from registered,
(SELECT * FROM `cp-stats`.member_of where Group_ID='19')res
where res.ID=registered.User_ID order by Rating desc;
```

RECRUITER (Contains OLAP Majorly)

- *List preferred users (on basis of ratings)*

```
select preferred.User_ID, Rating from preferred, registered
where preferred.User_ID=registered.User_ID and
preferred.Recruiter_ID='R-1' order by Rating desc;
```

- *List recruited users*

```
SELECT * FROM
USERS
JOIN
(SELECT * FROM RECRUITED WHERE Recruiter_ID =
'R-1')result_query1
ON USERS.ID = result_query1.User_ID
```

- *Sort recruitees on basis of popularity*

```
SELECT User_ID, Followers FROM recruited, num_followers where
Recruiter_ID='R-11' and User_ID=ID order by Followers desc;
```

- *List users by*

a. Location

```
SELECT * FROM USER WHERE Location_City = 'Delhi' OR
Location_Country = 'India'
```

b. Rating

```
SELECT User_ID, rating FROM registered ORDER BY rating
DESC;
```

c. Strength

```
SELECT * FROM USERS WHERE ID IN (SELECT User_ID FROM
user_strength WHERE Tag='dp' AND Category='Strength');
```

d. Language

```
SELECT * FROM
USERS
```

```

JOIN
(SELECT * FROM USER_LANGUAGES WHERE Language =
'C++')result_query1
ON USERS.ID = result_query1.ID

```

- Analyze users by their number of solves on all tags and difficulties of problems

```

SELECT User_ID, Tag, Rating_Difficulty, COUNT(*) AS solves FROM
problems, (SELECT User_ID,solved.Problem_ID AS Problem_ID,Tag
FROM solved,problems_tags WHERE
(solved.Problem_ID=problems_tags.Problem_ID)) res WHERE
res.Problem_ID=problems.Problem_ID GROUP BY User_ID, Tag,
Rating_Difficulty WITH ROLLUP;

```

- Analyze users by their submissions in language

```

SELECT User_ID, Language, COUNT(*) as probs_solved from solved
group by User_ID, Language order by User_ID ;

```

- Analyze by their number of solves on all difficulties of problems in the past year

```

SELECT solved.User_ID, problems.Rating_Difficulty, COUNT(*) AS
Prob_Solved FROM solved, problems WHERE
solved.Problem_ID=problems.Problem_ID AND solved.Date >
DATE_ADD(NOW(), INTERVAL -1 YEAR) GROUP BY solved.User_ID,
problems.Rating_Difficulty WITH ROLLUP;

```

Programming Organization

- List all my users

```

select * from users
join
(select * from registered where Programming_Organisation_ID =
'P-1')rql
on rql.user_id = users.ID
where rql.user_id is not null

```

- Contests

- a. Give me contests after a certain date

```

SELECT * FROM CONTESTS WHERE DateOfContest > '2018-10-10'

```

- b. Maximum liked contests

```

SELECT * FROM CONTESTS WHERE Likes = (SELECT MAX(Likes)
FROM CONTESTS)

```

- How many ads have been posted

```
SELECT COUNT(Ads) FROM ORGANISATION_PAYSTO WHERE  
Organization_ID = 'P-1'
```

Admin

- *Increase all users trust rating by blogs count * constant whose blogs have more than 50 likes in the past year*

```
UPDATE users, (select ID, COUNT(*) as cnt from blogs where  
Likes > 50 and Date > DATE_ADD(NOW(), INTERVAL -1 YEAR) group  
by ID) as res set TrustRating = TrustRating + (res.cnt * 5)  
where users.ID = res.ID;
```

- *Reported accounts on basis of unblocking date*

```
SELECT * from users  
join  
blocks  
where users.ID = blocks.User_ID order by Date_Unblock;
```