SQL PROJECT

HELLO!

Team leader –

Rhythm Pathania(UE184064)

Team Member –

Siddharth(UE185142)

" A SQL query walks into a bar, walks up to 2 tables and ask "Can I join you?"

INDEX

▶ Objective

▶ Target Audience

▶ Technology Used

▶ Project Demonstration

▶ Future scope

▶ Conclusion

OBJECTIVE

We were given a project and we had to create a database and solve the query given

PROBLEM STATEMENT:

Samantha interviews many candidates from different colleges using coding challenges and contests. Write a query to print the *contest_id*, *hacker_id*, *name*, and the sums of *total_submissions*, *total_accepted_submissions*, *total_views*, and *total_unique_views* for each contest sorted by *contest_id*. Exclude the contest from the result if all four sums are 0.

TARGET AUDIENCE

The target audience is all the HR managers who are conducting recruitment drives in various colleges , this project can help them to retrieve a particular information about the candidates.

TECHNOLOGY USED

SQL AND MYSQL

SQL-

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. This tutorial will give you a quick start to SQL. It covers most of the topics required for a basic understanding of SQL and to get a feel of how it works.

MySql-

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

PROJECT DEMONSTRATION

FINAL QUERY

RESULT

```
                                      SSIONS),

                          _VIEWS)

              ON CON.CONTEST_ID = COL.CONTEST_ID
          GE CHA ON COL.COLLEGE_ID = CHA.COLLEGE_ID

          ELECT CHALLENGE_ID,
                SUM(TOTAL_VIEWS) AS TOTAL_VIEWS,
                SUM(TOTAL_UNIQUE_VIEWS) AS TOTAL_UNIQUE_VIEWS
  ->    FROM VIEW_STATS
  ->    GROUP BY CHALLENGE_ID) VS ON CHA.CHALLENGE_ID = VS.CHALLENGE_ID
  -> LEFT JOIN
  ->    (SELECT CHALLENGE_ID,
  ->          SUM(TOTAL_SUBMISIONS) AS TOTAL_SUBMISIONS,
  ->          SUM(TOTAL_ACCEPTED_SUBMISSIONS) AS TOTAL_ACCEPTED_SUBMISSIONS
  ->    FROM SUBMISSION_STATS
  ->    GROUP BY CHALLENGE_ID) SS ON CHA.CHALLENGE_ID = SS.CHALLENGE_ID
  -> GROUP BY CON.CONTEST_ID,
  ->          CON.HACKER_ID,
  ->          CON.NAME
  -> HAVING SUM(TOTAL_SUBMISIONS) != 0
  -> OR SUM(TOTAL_ACCEPTED_SUBMISSIONS) != 0
  -> OR SUM(TOTAL_VIEWS) != 0
  -> OR SUM(TOTAL_UNIQUE_VIEWS) != 0
  -> ORDER BY CONTEST_ID;
+------------+-----------+--------+----------------------+-----------------------------+------------------+-----------------------+
| CONTEST_ID | HACKER_ID | NAME   | SUM(TOTAL_SUBMISIONS) | SUM(TOTAL_ACCEPTED_SUBMISSIONS) | SUM(TOTAL_VIEWS) | SUM(TOTAL_UNIQUE_VIEWS) |
+------------+-----------+--------+----------------------+-----------------------------+------------------+-----------------------+
|      66406 |     17973 | Rose   |                  111 |                          39 |              156 |                    56 |
|      66556 |     79153 | Angela |                  191 |                          32 |               86 |                    21 |
|      94828 |     80275 | Frank  |                  150 |                          38 |               76 |                    32 |
+------------+-----------+--------+----------------------+-----------------------------+------------------+-----------------------+
3 rows in set (0.17 sec)

mysql>
```

FUTURE SCOPE

In future this project can be used even by placement cells of various colleges to manage the data of students sitting for the placements . It will help them to manage each and every record of them also small changes can be done easily .

CONCLUSION

We have successfully created a query which solve the problem statement given . Not only this problem but many more similar problems can be solved using SQL to benefit the user.

THANKS!

Any questions?