



SCS1309 Database Management

Lab Sheet - MySQL 05

Create a database to store information about a Global Research Collaboration Platform (GRCP). The platform includes researchers, their organizations, projects, publications, and conference participation records. You have been tasked with managing and optimizing this database to handle complex queries and improve performance.

Database Schema Overview

1. Researchers

Resear cherID	Name	Age	Nationality	Organization ID	AreaOfExpertise
1	Dr. Sudesh	45	USA	101	Artificial Intelligence
2	Dr. Wikrama	50	Sri Lanka	102	Quantum Computing

2. Organizations

OrganizationID	OrgName	Country
101	MIT	45
102	UCSC	50

3. Projects

ProjectID	ProjectName	StartDate	EndDate	LeadResearcherID
201	AI Ethics Framework	2024-12-31	2024-12-31	1

4. Publications

PublicationID	Title	PublishedDate	ProjectID	ResearcherID
301	“AI and Ethics”	2024-05-01	201	1

5. Conferences

ConferenceID	Name	Location	ConferenceDate
401	Global AI Summit	London	2024-06-15

6. Participation

ParticipationID	ResearcherID	ConferenceID
1	1	401

Tasks:

1. Write a query to list all projects that have at least one associated publication. Use a subquery to determine the relevant projects.
2. Write a query to list organizations that have more than the average number of researchers working on projects. Use a nested subquery and aggregate functions.
3. Identify researchers who are involved in more than one project. Use a subquery to find such researchers based on their participation records.
4. Write a query to list publication details along with the project name and the researcher's name who authored it.
5. Create a view *OrgRankings* to display rankings of organizations based on the number of projects they lead. Include columns for organization name, project count, and rank.
6. Create a view *ResearcherDetails* that includes only non-sensitive details (e.g., name, age, and organization) of researchers. Exclude private attributes like nationality.
7. Create a view *ActiveOrganizations* that lists only organizations whose researchers participated in at least three conferences.
8. Explain how a materialized view could store the top 10 most frequently cited

- publications and discuss how to maintain it. (Include the SQL logic and outline periodic refresh strategies using triggers or scheduled jobs.)
9. Create a view named *MultiProjectResearchers* that lists researchers who have contributed to multiple projects. Include columns: *ResearcherName*, *ProjectCount*, and *OrganizationsName*.
 10. Create a view named *ActiveResearchers* that includes only researchers who have participated in at least two conferences and authored at least one publication.
 11. Create a view named *ResearchImpact* that ranks researchers based on their total number of *publications* and *conference participations* combined.
 12. Create a view named *InterdisciplinaryProjects* to list projects led by researchers from different areas of expertise. Include columns: *ProjectName*, *LeadResearcher*, and *DistinctExpertiseCount*.
 13. Create a view named *ResearcherCollaborations* showing pairs of researchers who have co-authored at least one publication.
 14. Create a dynamic view named *RecentPublications* to list details of publications from the last 6 months. Include the columns: *PublicationTitle*, *PublishedDate*, *ResearcherName*, and *ProjectName*.
 - a. Write a query to retrieve all records from the *RecentPublications* view for projects led by researchers from the organization "MIT."
 15. Partition the data by creating a view named *OngoingProjects* to display details of projects that have not yet ended. Include columns: *ProjectName*, *LeadResearcher*, and *RemainingDays*.
 - a. Add a calculated column *RemainingDays* to the view, which calculates the days left until the *EndDate* using the current date.
 16. To optimize performance when querying publication details for a specific project, create an index on the *ProjectID* and *ResearcherID* columns in the *Publications* table.
 17. After creating the index, write a query to retrieve the *PublicationID*, *Title*, *PublishedDate*, *ProjectName*, and *ResearcherName* for all publications linked to the project with *ProjectID* = 201.
 18. The *Participation* table contains millions of records, and queries often involve retrieving participation details based on both *ResearcherID* and *ConferenceID*.
 - a. Create a composite index on the *ResearcherID* and *ConferenceID* columns in the *Participation* table to optimize queries.
 - b. After creating the index, write a query to retrieve the *Name* of researchers and *ConferenceName* for all records where the *ResearcherID* = 1 and *ConferenceID* = 401.
 19. Queries often join the *Conferences*, *Participation*, and *Researchers* tables to retrieve details about researchers attending specific conferences.
 - a. Create an index on the *ConferenceID* column in the *Participation* table and on the *ResearcherID* column in the *Researchers* table.
 - b. After creating these indexes, write a query to retrieve the *Name* of conferences, *ResearcherName*, and *AreaOfExpertise* for conferences held in "London" attended by researchers specializing in "Artificial Intelligence."

20. The *Publications* table frequently joins with the *Projects* and *Researchers* tables to retrieve publication details.
- a. Create indexes on the *ProjectID* column in both the *Publications* and *Projects* tables.
 - b. After creating the indexes, write a query to retrieve all publication titles and their project names for projects started after "2024-01-01."