Employee Performance Mapping – SQL Project Report

Objective

This project supports the HR department of **ScienceQtech** in employee performance mapping based on position, experience, and ratings. It utilizes SQL to extract insights from employee and project datasets to assess employee strengths, compensation, and overall contributions to the company.

Database Design and Data Import

A MySQL database named **`employee`** was created. Three CSV datasets were loaded:

- `emp_record_table`
- `data_science_team`
- `proj_table`

These tables represent key organizational dimensions:

- Employee characteristics
- Project details
- Data science team organization

Entity-Relationship Design

Relationships were defined as follows:

- Employees are linked to projects via `PROJ_ID`
- Reporting structures are mapped via `MANAGER_ID`

These relationships help track employee performance through inter-entity interactions.

Basic Data Retrieval

Basic SQL queries were used to extract:

- Employee names
- Departments
- Genders
- IDs

These outputs were foundational in constructing HR reports.

Performance Filtering

Employees were segmented by performance rating into:

- Low
- Average
- High performers

This categorization helps guide decisions regarding training, upskilling, or promotion.

Data Transformation

Transformations included:

- Concatenating employee names
- Salary computations (e.g., bonuses)

These outputs allowed deeper insight into financial implications at both individual and departmental levels.

Organizational Hierarchy

Analysis of the `MANAGER_ID` field enabled:

- Construction of reporting hierarchies
- Identification of team leads, managers, and presidents

This insight is crucial for team performance evaluation.

Department Analysis

Using **`UNION`** and filtering, performance across departments (e.g., Finance vs Healthcare) was compared. This enabled:

- Cross-functional evaluation
- Structural efficiency comparison

Window Functions for Insights

Window functions such as:

- `RANK()`
- `MAX() OVER (PARTITION BY)`

were applied to:

- Rank employees within departments
- Compare departmental top performers

Stored Procedures and Functions

Custom SQL logic was implemented:

- **Stored procedures** to fetch employees by experience levels
- **User-defined functions** to validate experience against role standards

These tools added flexibility and reusability to HR operations.

Query Optimization

Query performance was analyzed using `EXPLAIN`. Optimizations included:

- Indexing on frequently searched fields (e.g., `FIRST_NAME`)

This significantly improved performance with large datasets.

Geographic and Salary Analysis

Analytical functions calculated average salaries by:

- Continent
- Country

These insights supported:

- Regional salary pattern analysis
- Strategic compensation planning

Conclusion

This project demonstrates the transformation of real-world HR needs into robust SQL logic. By efficiently extracting, transforming, and analyzing data, actionable insights were provided for:

- Performance mapping
- Compensation strategy
- Organizational structure evaluation