# EMPLOYEE PERFORMANCE IN AN ORGANISATION Documentation

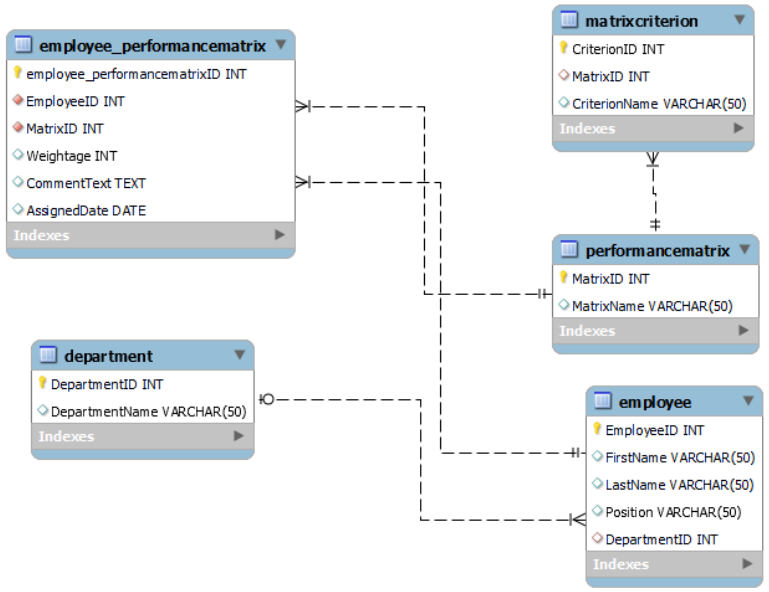
## 1. Domain Description and Relevance:

The chosen domain is a real-time Performance Management System (PMS) designed to track and manage employee performance within an organisation. This system allows for the evaluation of employees based on predefined performance criteria and matrices. The relevance of this system lies in its ability to provide timely and accurate insights into employee performance, aiding in decision-making, talent management, and overall organisational improvement.

- Background Study and Requirement Analysis:
    - Conducted interviews with stakeholders to understand the performance evaluation process.
    - Identified key entities: Employee, Department, PerformanceMatrix, MatrixCriterion, Employee_PerformanceMatrix.
    - Analysed the relationships between entities and their attributes.
    - Identified normalisation requirements to eliminate redundancy and ensure data integrity.

## 2. Entity-Relationship Diagram (ERD):

The ERD was created using MySQL Workbench to visually represent the relationships between tables.




## 3. Table Descriptions:

*Employee Table:*
EmployeeID (Primary Key): Unique identifier for each employee.
FirstName: First name of the employee.
LastName: Last name of the employee.
Position: Job position of the employee.
DepartmentID (Foreign Key): References the Department table.

*Department Table:*
DepartmentID (Primary Key): Unique identifier for each department.
DepartmentName: Name of the department.

*PerformanceMatrix Table:*
MatrixID (Primary Key): Unique identifier for each performance matrix.
MatrixName: Name of the performance matrix.

*MatrixCriterion Table:*
CriterionID (Primary Key): Unique identifier for each performance criterion.
MatrixID (Foreign Key): References the PerformanceMatrix table.
CriterionName: Name of the performance criterion.

*Employee_PerformanceMatrix Table:*
EmployeeID (Primary Key, Foreign Key): References the Employee table.
MatrixID (Primary Key, Foreign Key): References the PerformanceMatrix table.
Weightage: Numerical value representing the weightage of the criterion.
CommentText: Textual comments on employee performance.
AssignedDate: Date when the performance matrix was assigned.

Relationships

- *Many-to-One Relationship:*
    - *Employee to Department*: Many employees can belong to one department, as indicated by the foreign key (DepartmentID) in the Employee table referencing the primary key (DepartmentID) in the Department table.
    - *MatrixCriterion to PerformanceMatrix*: Many criteria can be associated with one performance matrix, as indicated by the foreign key (MatrixID) in the MatrixCriterion table referencing the primary key (MatrixID) in the PerformanceMatrix table.
    - *Employee_PerformanceMatrix to Employee*: Many performance records can be associated with one employee, as indicated by the foreign key (EmployeeID) in the Employee_PerformanceMatrix table referencing the primary key (EmployeeID) in the Employee table.

    - *Employee_PerformanceMatrix to PerformanceMatrix:* Many performance records can be associated with one performance matrix, as indicated by the foreign key (MatrixID) in the Employee_PerformanceMatrix table referencing the primary key (MatrixID) in the PerformanceMatrix table.

- *One-to-One Relationship:*
  No explicit one-to-one relationships are defined in the provided tables.

However, it's worth noting that the primary key in each table uniquely identifies a single record, implying a one-to-one relationship between primary keys and corresponding records within each table.

## 4. Normalisation Process:

The database design follows normalisation principles to eliminate redundancy and ensure data integrity.

- *1NF (First Normal Form)*: All tables have atomic values, and there are no repeating groups.
- *2NF (Second Normal Form):* All non-prime attributes are fully functionally dependent on the primary key.
- *3NF (Third Normal Form)*: No transitive dependencies exist; all attributes are functionally dependent on the primary key.

Normalisation was achieved by breaking down tables into smaller, more manageable ones and establishing relationships between them. This ensures efficient data storage, reduces redundancy, and maintains data integrity.

By adhering to normalisation principles, the database design is well-structured to support real-time performance management within an organisation. The ERD and table descriptions provide a clear understanding of the system's architecture and its components.