

# EMPLOYEE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted for the partial fulfillment for the award of “Python Full Stack Development”

SUBMITTED BY  
VENKAT S

UNDER GUIDANCE OF  
MR.VALARIVAN

BESANT TECHNOLOGY

## OBJECTIVE :

The Employee Management System project is designed to create a sophisticated and efficient MySQL database solution that encompasses various critical aspects of human resource management. The system will include comprehensive modules for managing departments, where users can define and organize departments, and assign employees accordingly. Employee records will be meticulously maintained, capturing essential tables such as ” **Department, Employee, Salary , Project , Employee Project , PerformanceReviews.**” The salary module will track compensation data, including base salaries, bonuses, and total salary, ensuring up-to-date and accurate payroll information. In addition, the project management component will allow for the creation, tracking, and management of projects, assigning employees to projects based on their roles and skills. The system will also feature an employee-project assignment feature to monitor and manage employee involvement in various projects. Performance reviews will be systematically recorded, enabling the evaluation of employee performance and providing insights for management decisions. The project aims to enhance organizational efficiency by offering robust data management, facilitating strategic decision-making, and ensuring data integrity through a well-organized relational database design. This comprehensive approach will support effective human resource operations, improve workflow management, and provide valuable analytics for organizational growth.

## **CREATE DATABASE :**

**Create Schema Employee\_Management\_System**

**Use Employee\_Management\_System**

## **Query For Department :**

```
create table Departments(dept_Id int primary key , dept_Name varchar(30) ,  
Location varchar(30)) ;
```

```
insert into Departments values (1, 'HR', 'New York'),  
(2, 'Developer', 'Los Angeles'),  
(3, 'Developer', 'Chicago'),  
(4, 'Tester', 'San Francisco'),  
(5, 'Developer', 'Seattle'),  
(6, 'Support', 'Boston'),  
(7, 'Tester', 'Miami'),  
(8, 'Developer', 'Houston'),  
(9, 'Support', 'Phoenix'),  
(10, 'Tester', 'Austin'),  
(11, 'Developer', 'Dallas'),  
(12, 'Tester', 'Denver'),  
(13, 'Admin', 'Philadelphia'),  
(14, 'Customer Service', 'San Diego'),  
(15, 'Supply Chain', 'Portland'),  
(16, 'Developer', 'Las Vegas'),  
(17, 'Corporate Strategy', 'Atlanta'),  
(18, 'HR', 'Charlotte'),  
(19, 'Compliance', 'San Jose'),  
(20, 'Support', 'Orlando');
```

**(or)**

```
INSERT INTO Departments (dept_id, dept_name, location)  
SELECT  
    n,  
    CONCAT('Dept_', n),  
    CASE  
        WHEN n % 2 = 0 THEN 'New York'
```

```

        WHEN n % 3 = 0 THEN 'San Francisco'
        ELSE 'Chicago'
    END
FROM
    (SELECT n FROM (SELECT @row := @row + 1 AS n FROM (SELECT 0
    UNION ALL SELECT 1) t1 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t2
    CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t3 CROSS JOIN (SELECT 0
    UNION ALL SELECT 1) t4 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t5
    CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t6, (SELECT @row := 0) r)
    num LIMIT 95) as nums;

select * from departments;

```

**OUTPUT :**

	dept_Id	dept_Name	Location
►	1	HR	New York
	2	Developer	Los Angeles
	3	Developer	Chicago
	4	Tester	San Francisco
	5	Developer	Seattle
	6	Support	Boston
	7	Tester	Miami

### **Query For Employees :**

```
create table Employees (emp_Id int auto_increment primary key , Emp_Name  
varchar(30) , Dept_Id int , Position varchar(50) , Salary int ,  
Hire_date Date , Email varchar(50) , Phone_No varchar(30));
```

```
insert into Employees(Emp_Name , Dept_Id , Position , Salary , Hire_date ,  
Email , Phone_no) values
```

```
('Arun Kumar', 1, 'Software Engineer', 75000, 20200101,  
'arunkumar1@example.com', '+91-9812345670'),  
('Venkat' ,3,'Web Developer' ,80000,20240901 , 'skvenkat@gmail.com'  
, '+91-6386357683'),  
('Lakshmi Narayanan', 2, 'Data Analyst', 68000, 20200215,  
'lakshminarayanan2@example.com', '+91-9812345671'),  
('Ravi Chandran', 3, 'Project Manager', 90000, 20200310,  
'ravichandran3@example.com', '+91-9812345672'),  
('Vidhya Rani', 1, 'Software Engineer', 72000, 20200420,  
'vidhyarani4@example.com', '+91-9812345673'),  
('Murugan Sundar', 4, 'HR Specialist', 60000, 20200505,  
'murugansundar5@example.com', '+91-9812345674'),  
('Saravanan Bala', 5, 'Accountant', 65000, 20200612,  
'saravananbala6@example.com', '+91-9812345675'),  
('Priya Selvam', 1, 'DevOps Engineer', 78000, 20200725,  
'priyaselvam7@example.com', '+91-9812345676'),  
('Gokul Raj', 2, 'Data Scientist', 85000, 20200830, 'gokulraj8@example.com',  
, '+91-9812345677'),  
('Karthik Subramani', 3, 'Product Manager', 88000, 20200915,  
'karthiksubramani9@example.com', '+91-9812345678'),  
('Meena Devi', 4, 'HR Manager', 72000, 20201005,  
'meenadevi10@example.com', '+91-9812345679'),  
('Anand Kumar', 1, 'Software Engineer', 74000, 20201111,  
'anandkumar11@example.com', '+91-9812345680'),  
('Deepa Lakshmi', 2, 'Data Analyst', 69000, 20201222,  
'deepalakshmi12@example.com', '+91-9812345681'),
```

```

('Vijay Kumar', 3, 'Project Manager', 92000, 20210101,
'vijaykumar13@example.com', '+91-9812345682'),
('Sundar Rajan', 1, 'Software Engineer', 76000, 20210115,
'sundarrajan14@example.com', '+91-9812345683'),
('Divya Rani', 4, 'HR Specialist', 62000, 20210210, 'divyarani15@example.com',
'+91-9812345684'),
('Rajesh Kumar', 5, 'Accountant', 66000, 20210305,
'rajeshkumar16@example.com', '+91-9812345685'),
('Siva Priya', 1, 'DevOps Engineer', 79000, 20210420,
'sivapriya17@example.com', '+91-9812345686'),
('Bala Murugan', 2, 'Data Scientist', 86000, 20210515,
'balamurugan18@example.com', '+91-9812345687'),
('Ramesh Kumar', 3, 'Product Manager', 89000, 20210630,
'rameshkumar19@example.com', '+91-9812345688');

```

(or)

```

INSERT INTO Employees (emp_name, dept_id, position, salary, hire_date,
email, phone)
SELECT
    CONCAT('Employee_', n),
    FLOOR(1 + (RAND() * 100)), -- Random dept_id between 1 and 100
    CASE
        WHEN n % 2 = 0 THEN 'Manager'
        WHEN n % 3 = 0 THEN 'Developer'
        ELSE 'Analyst'
    END,
    FLOOR(30000 + (RAND() * 50000)), -- Random salary between 30,000 and
80,000
    DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 3650) DAY), --
Random hire_date in the last 10 years
    CONCAT('employee_', n, '@company.com'),
    CONCAT('555-', LPAD(FLOOR(RAND() * 10000), 4, '0'))
FROM
    (SELECT n FROM (SELECT @row := @row + 1 AS n FROM (SELECT 0
UNION ALL SELECT 1) t1 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t2
CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t3 CROSS JOIN (SELECT 0
UNION ALL SELECT 1) t4 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t5

```

CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t6, (SELECT @row := 0) r  
num LIMIT 1000) as nums;

## **OUTPUT :**

emp_Id	Emp_Name	Dept_Id	Position	Salary	Hire_date	Email	Phone_No
1	Arun Kumar	1	Software Engineer	75000	2020-01-01	arunkumar1@example.com	+91-9812345670
2	Venkat	3	Web Developer	80000	2024-09-01	skvenkat@gmail.com	+91-6386357683
3	Lakshmi Narayanan	2	Data Analyst	68000	2020-02-15	lakshminarayanan2@example.com	+91-9812345671
4	Ravi Chandran	3	Project Manager	90000	2020-03-10	ravichandran3@example.com	+91-9812345672
5	Vidhya Rani	1	Software Engineer	72000	2020-04-20	vidhyarani4@example.com	+91-9812345673
6	Murugan Sundar	4	HR Specialist	60000	2020-05-05	murugansundar5@example.com	+91-9812345674
7	Saravanan Bala	5	Accountant	65000	2020-06-12	saravananbala6@example.com	+91-9812345675
8	Priya Selvam	1	DevOps Engineer	78000	2020-07-25	priyaselvam7@example.com	+91-9812345676
9	Gokul Raj	2	Data Scientist	85000	2020-08-30	gokulraj8@example.com	+91-9812345677
10	Karthik Subramani	3	Product Manager	88000	2020-09-15	karthiksubramani9@example.com	+91-9812345678
11	Meena Devi	4	HR Manager	72000	2020-10-05	meenadevi10@example.com	+91-9812345679
12	Anand Kumar	1	Software Engineer	74000	2020-11-11	anandkumar11@example.com	+91-9812345680
13	Deepa Lakshmi	2	Data Analyst	69000	2020-12-22	deepalakshmi12@example.com	+91-9812345681
14	Vijay Kumar	3	Project Manager	92000	2021-01-01	vijaykumar13@example.com	+91-9812345682
15	Sundar Rajan	1	Software Engineer	76000	2021-01-15	sundarrajan14@example.com	+91-9812345683

## **Query For Salaries :**

create table Salaries (Salary\_Id int auto\_increment primary key , Emp\_Id int ,Base\_Salary int , Bonus int , Total\_Salary varchar(30), Pay\_Date date);

insert into Salaries(Emp\_Id ,Base\_salary,Bonus,Total\_salary ,pay\_date) values  
(1, 70000, 5000, 75000, '2023-08-01'),  
(2, 80000, 6000, 86000, '2023-08-01'),  
(3, 65000, 4500, 69500, '2023-08-01'),  
(4, 72000, 5200, 77200, '2023-08-01'),  
(5, 95000, 8000, 103000, '2023-08-01'),  
(6, 83000, 6300, 89300, '2023-08-01'),

```
(7, 60000, 4000, 64000, '2023-08-01'),
(8, 88000, 7000, 95000, '2023-08-01'),
(9, 55000, 3500, 58500, '2023-08-01'),
(10, 90000, 7500, 97500, '2023-08-01'),
(11, 62000, 3000, 65000, '2023-08-01'),
(12, 72000, 4200, 76200, '2023-08-01'),
(13, 48000, 2000, 50000, '2023-08-01'),
(14, 53000, 2500, 55500, '2023-08-01'),
(15, 67000, 3500, 70500, '2023-08-01');
```

**(or)**

```
INSERT INTO Salaries (emp_id, base_salary, bonus, pay_date)
SELECT
    emp_id,
    salary,
    FLOOR(RAND() * 10000), -- Random bonus up to 10,000
    DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 365) DAY) -- Random
pay_date in the last year
FROM
    Employees;
```

### **OUTPUT :**

Salary_Id	Emp_Id	Base_Salary	Bonus	Total_Salary	Pay_Date
1	1	70000	5000	75000	2023-08-01
2	2	80000	6000	86000	2023-08-01
3	3	65000	4500	69500	2023-08-01
4	4	72000	5200	77200	2023-08-01
5	5	95000	8000	103000	2023-08-01
6	6	83000	6300	89300	2023-08-01
7	7	60000	4000	64000	2023-08-01
8	8	88000	7000	95000	2023-08-01
9	9	55000	3500	58500	2023-08-01
10	10	90000	7500	97500	2023-08-01
11	11	62000	90000	65000	2023-08-01
12	12	72000	4200	76200	2023-08-01
13	13	48000	2000	50000	2023-08-01
14	14	53000	2500	55500	2023-08-01
15	15	67000	3500	70500	2023-08-01

### **Query For Projects :**

create table Projects (Project\_Id int auto\_increment primary key , Project\_Name varchar(30) , Start\_date date , End\_date date , budget int);

insert into Projects(Project\_name,Start\_date,End\_date ,budget) values ('Project Alpha', '2024-01-01', '2024-06-30', 500000),  
('Project Beta', '2024-02-01', '2024-07-31', 600000),  
('Project Gamma', '2024-03-01', '2024-08-31', 550000),  
('Project Delta', '2024-04-01', '2024-09-30', 450000),  
('Project Epsilon', '2024-05-01', '2024-10-31', 700000),  
('Project Zeta', '2024-06-01', '2024-11-30', 650000),  
('Project Eta', '2024-07-01', '2024-12-31', 600000),  
('Project Theta', '2024-08-01', '2025-01-31', 750000),  
('Project Iota', '2024-09-01', '2025-02-28', 800000),  
('Project Kappa', '2024-10-01', '2025-03-31', 550000),  
('Project Lambda', '2024-11-01', '2025-04-30', 500000),  
('Project Mu', '2024-12-01', '2025-05-31', 600000),  
('Project Nu', '2025-01-01', '2025-06-30', 650000),  
('Project Xi', '2025-02-01', '2025-07-31', 700000),  
('Project Omicron', '2025-03-01', '2025-08-31', 550000),

**(or)**

INSERT INTO Projects (project\_name, start\_date, end\_date, budget)  
SELECT  
    CONCAT('Project\_', n),  
    DATE\_SUB(CURDATE(), INTERVAL FLOOR(RAND() \* 365 \* 2) DAY), --  
Random start\_date in the last 2 years  
    DATE\_ADD(CURDATE(), INTERVAL FLOOR(RAND() \* 365) DAY), -- Random  
end\_date within the next year  
    FLOOR(50000 + (RAND() \* 150000)) -- Random budget between 50,000 and  
200,000  
FROM  
    (SELECT n FROM (SELECT @row := @row + 1 AS n FROM (SELECT 0  
UNION ALL SELECT 1) t1 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t2



CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t3 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t4 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t5 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t6, (SELECT @row := 0) r) num LIMIT 200) as nums;

### **OUTPUT :**

Project_Id	Project_Name	Start_date	End_date	budget
1	Project Alpha	2024-01-01	2024-06-30	500000
2	Project Beta	2024-02-01	2024-07-31	600000
3	Project Gamma	2024-03-01	2024-08-31	550000
4	Project Delta	2024-04-01	2024-09-30	450000
5	Project Epsilon	2024-05-01	2024-10-31	700000
6	Project Zeta	2024-06-01	2024-11-30	650000
7	Project Eta	2024-07-01	2024-12-31	600000
8	Project Theta	2024-08-01	2025-01-31	750000
9	Project Iota	2024-09-01	2025-02-28	800000
10	Project Kappa	2024-10-01	2025-03-31	550000
11	Project Lambda	2024-11-01	2025-04-30	500000
12	Project Mu	2024-12-01	2025-05-31	600000
13	Project Nu	2025-01-01	2025-06-30	650000
14	Project Xi	2025-02-01	2025-07-31	700000
15	Project Omicron	2025-03-01	2025-08-31	550000

### **Query For EmployeeProjects :**

CREATE TABLE EmployeeProjects (

```

emp_id INT,
project_id INT,
role VARCHAR(50)
);
INSERT INTO EmployeeProjects (emp_id, project_id, role)
SELECT
    emp_id,
    FLOOR(1 + (RAND() * 200)), -- Random project_id between 1 and 200
    CASE
        WHEN n % 2 = 0 THEN 'Team Member'
        WHEN n % 3 = 0 THEN 'Lead'
        ELSE 'Consultant'
    END
FROM
    Employees
    CROSS JOIN (SELECT @row := @row + 1 AS n FROM (SELECT 0 UNION
ALL SELECT 1) t1 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t2 CROSS
JOIN (SELECT 0 UNION ALL SELECT 1) t3 CROSS JOIN (SELECT 0 UNION
ALL SELECT 1) t4 CROSS JOIN (SELECT 0 UNION ALL SELECT 1) t5 CROSS
JOIN (SELECT 0 UNION ALL SELECT 1) t6, (SELECT @row := 0) r) as nums
LIMIT 3000;

```

### **OUTPUT :**

emp_id	project_id	role
1	75	Team Member
1	142	Lead
1	83	Team Member
1	189	Consultant
1	98	Team Member
1	119	Consultant
1	102	Team Member
1	153	Lead
1	58	Team Member
1	29	Consultant
1	171	Team Member
1	166	Consultant

### **Query For PerformanceReviews :**

create table PerformanceReviews(Review\_Id int auto\_increment primary key,  
Emp\_Id int , Review\_date date , Review\_Score int , Comments Text);

```
INSERT INTO PerformanceReviews (emp_id, review_date, review_score,  
comments)  
SELECT  
    emp_id,  
    DATE_SUB(CURDATE(), INTERVAL FLOOR(RAND() * 365) DAY), -- Random  
review_date in the last year  
    FLOOR(1 + (RAND() * 5)), -- Random review_score between 1 and 5  
    CASE  
        WHEN n % 2 = 0 THEN 'Excellent performance'  
        WHEN n % 3 = 0 THEN 'Needs improvement'  
        ELSE 'Satisfactory performance'  
    END  
FROM  
    Employees  
    CROSS JOIN (SELECT @row := @row + 1 AS n FROM (SELECT 0 UNION  
ALL SELECT 1)  
t1 CROSS JOIN (SELECT 0 UNION ALL SELECT 1)  
t2 CROSS JOIN (SELECT 0 UNION ALL SELECT 1)  
t3 CROSS JOIN (SELECT 0 UNION ALL SELECT 1)  
t4 CROSS JOIN (SELECT 0 UNION ALL SELECT 1)  
t5 CROSS JOIN (SELECT 0 UNION ALL SELECT 1)  
t6, (SELECT @row := 0) r) as nums LIMIT 1000;
```

select \* from performancereviews;

## **OUTPUT :**

Review_Id	Emp_Id	Review_date	Review_Score	Comments
1	1	2024-03-23	5	Excellent performance
2	1	2024-05-22	3	Needs improvement
3	1	2024-08-14	3	Excellent performance
4	1	2024-06-22	3	Satisfactory performance
5	1	2023-08-27	3	Excellent performance
6	1	2024-05-01	1	Satisfactory performance
7	1	2023-08-27	3	Excellent performance
8	1	2024-08-08	5	Needs improvement
9	1	2023-10-26	1	Excellent performance
10	1	2024-04-19	2	Satisfactory performance

## **1 .COMPLEX JOINS TO RETRIEVE DATA :**

SELECT

Employees.emp\_name,  
Departments.dept\_name,  
EmployeeProjects.role,  
Projects.project\_name

FROM

Employees

JOIN

Departments ON Employees.dept\_id = Departments.dept\_id

JOIN

EmployeeProjects ON Employees.emp\_id = EmployeeProjects.emp\_id

JOIN

Projects ON EmployeeProjects.project\_id = Projects.project\_id;

## OUTPUT :

emp_name	dept_name	role	project_name
Arun Kumar	HR	Team Member	Project Gamma
Arun Kumar	HR	Lead	Project Sigma II
Arun Kumar	HR	Team Member	Project Lambda IV
Arun Kumar	HR	Consultant	Project Rho IV
Arun Kumar	HR	Team Member	Project Beta V
Arun Kumar	HR	Consultant	Project Tau
Arun Kumar	HR	Team Member	Project Beta
Arun Kumar	HR	Lead	Project Epsilon III
Arun Kumar	HR	Team Member	Project Kappa III
Arun Kumar	HR	Consultant	Project Epsilon II
Arun Kumar	HR	Team Member	Project Psi III

## 2. FIND THE HIGHEST SALARY OF EACH DEPARTMENT :

```
SELECT
    dept_name,
    MAX(salary) AS highest_salary
FROM
    Employees
JOIN
    Departments ON Employees.dept_id = Departments.dept_id
GROUP BY
    Dept_name;
```

## OUTPUT :

dept_name	highest_salary
HR	82000
Developer	96000
Tester	90000
Support	83000
Admin	48000
Customer Service	53000
Supply Chain	67000
Corporate Strategy	92000
Compliance	83000
Innovation	86000
Product Management	95000
Risk Management	86000
Tax	66000

### 3.SALARY UPDATE WITH ROLLBACK ;

UPDATE Salaries

SET base\_salary = base\_salary \* 1.1

WHERE emp\_id = 1;

### OUTPUT :

Salary_Id	Emp_Id	Base_Salary	Bonus	Total_Salary	Pay_Date
1	1	70000	5000	75000	2023-08-01
2	2	80000	6000	86000	2023-08-01
3	3	65000	4500	69500	2023-08-01
4	4	72000	5200	77200	2023-08-01
5	5	95000	8000	103000	2023-08-01
6	6	83000	6300	89300	2023-08-01
7	7	60000	4000	64000	2023-08-01
8	8	88000	7000	95000	2023-08-01
9	9	55000	3500	58500	2023-08-01
10	10	90000	7500	97500	2023-08-01
11	11	62000	90000	65000	2023-08-01
12	12	72000	4200	76200	2023-08-01
13	13	48000	2000	50000	2023-08-01
14	14	53000	2500	55500	2023-08-01
15	15	67000	3500	70500	2023-08-01