Human Resource Management System

Software Development Practices (SQL)

12/18/2019

Submitted To
Marek. M.Sukiennik

Submitted By
Venu Sukanya. Talari

Table of Contents

1. Introduction:

Human resource management system (HRMS) is software where an organization can track both internal and external people. Internal tracking means to maintain existing employees, consultants and contractors while external people will be the applicants. Through this system organization will be able to create and understand the link between employee and organization as well to each other. The system maintain all the basic information related to employees like name, address, educational background, work experience, salary, bonus, project details etc. the system will also help organization to track an employee`s leaves information, vacations, shifting details, attendance as well as performance management. In back time HRMS was not that important and organizations maintain every information manually, but now trend has been changed and technology take over everything, organizations adapting new technologies and making their departments more digitalized and smart. HRMS is becoming one of the most important parts of each organization, because this department is basically responsible for current employees' management and to recruit the best possible talent for the organization which helps organization to achieve their desired goals.

1.1 Aim of the Project:

The aim of the study is to provide the organization a HRMS database so that organization can keep its employees activities and information in a systematic way.

1.2 Objectives of the project:

The main objective of the project is to provide a systematic setup to the administration so that they can easily maintain their current employees´ data as well the new hire. By using HRMS administration can reduce the time of daily activities such as, employee detail, attendance, performance etc. this project is consist of five important modules which are as following,

- Person management (Employee, Consultant, Contractor)
- Payroll management (Salary, leaves, bonus, shifts)
- Performance & Appraisal (Projects, standards, gap measurement, rewards)
- Training and development (On-job and future development courses)
- Applicant search and tracking (job posting, tracking, applicant status)

1.3 Overview of HRMS:

In this project HR manager is basically responsible for maintenance of all HRMS activities as we can see in the following diagram. Where each module is linked with HRMS and performing some activities and then return back to the HRM system and update daily activities and this is a continuous process. HR manager and admin keep check and balance to maintain the system.

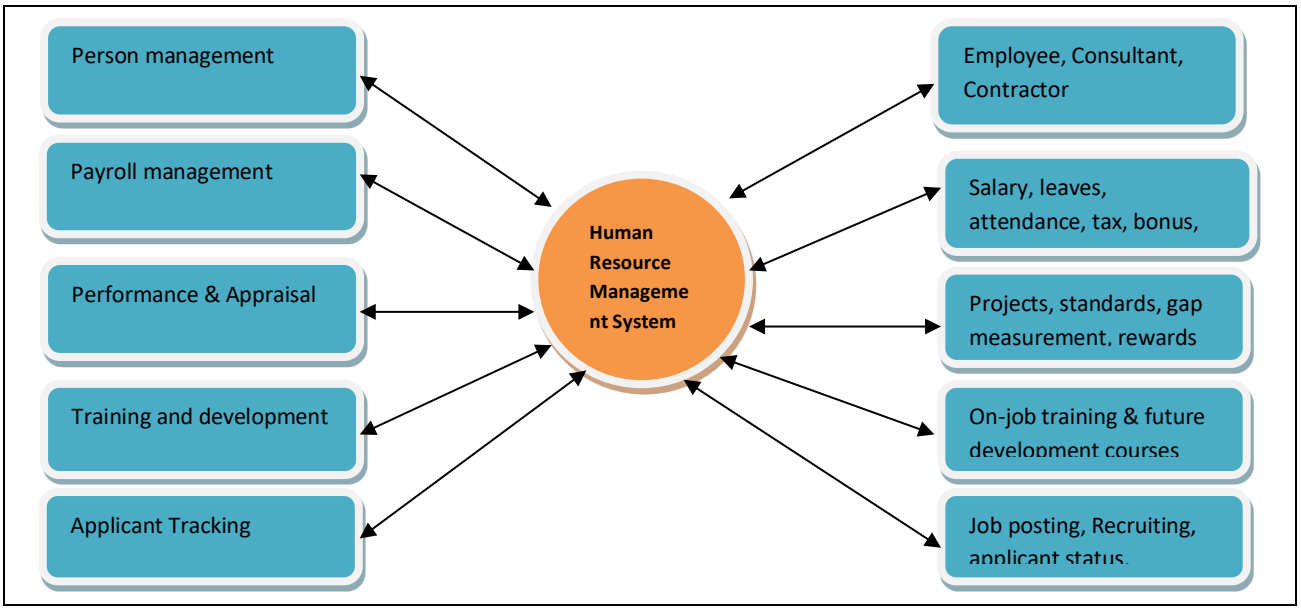


Figure: 01 Basic Activities of HRMS

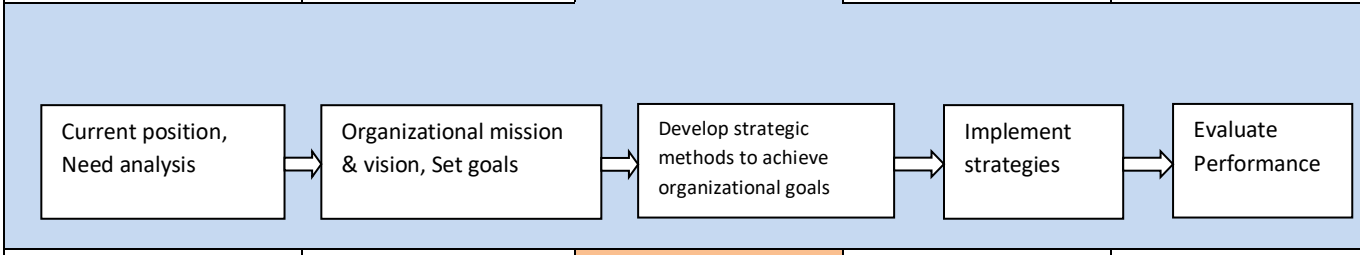
1.4 Objectives of the Report:

The report is divided into four parts, where in first part SIPOC (Supplier, input, Process, Output and Customer) method will be discussed, in second ERD (Entity Relationship Diagram) design and flowchart will be discussed, while the third part of the report will provide the database functionality which covers, main database which provide all the basic information about employees, moreover stored procedures, triggers, functions, views and indexes will also be discussed. Finally, the forth part is about front-end of the project.

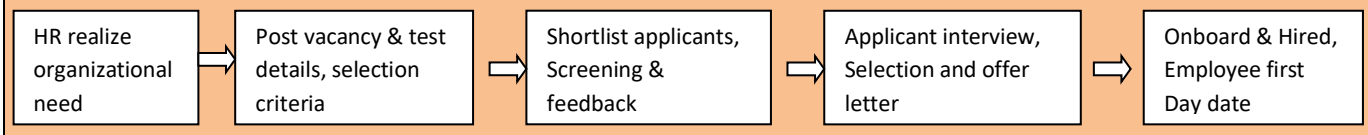
1. SIPOC Method:
2. ERD Design
3. ERD Flowchart
4. HRM Database Design and Functionality

2. SIPOC Method:

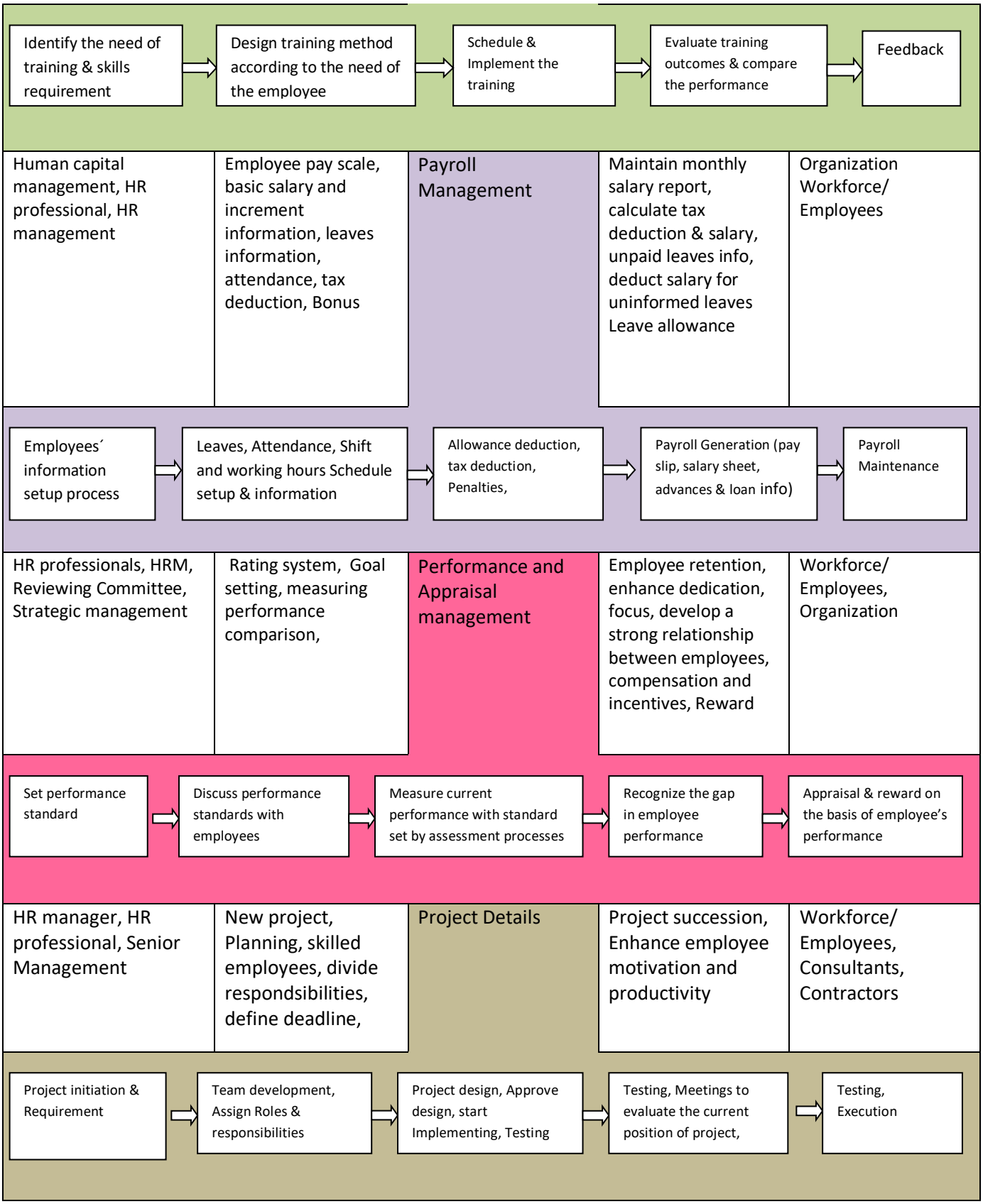
Supplier	Input	Process	Output	Customer
Organization`s strategic management, HR senior management,	Organization mission and vision	Strategic planning	Achieved business objectives and goal	Human resource management and HR professionals



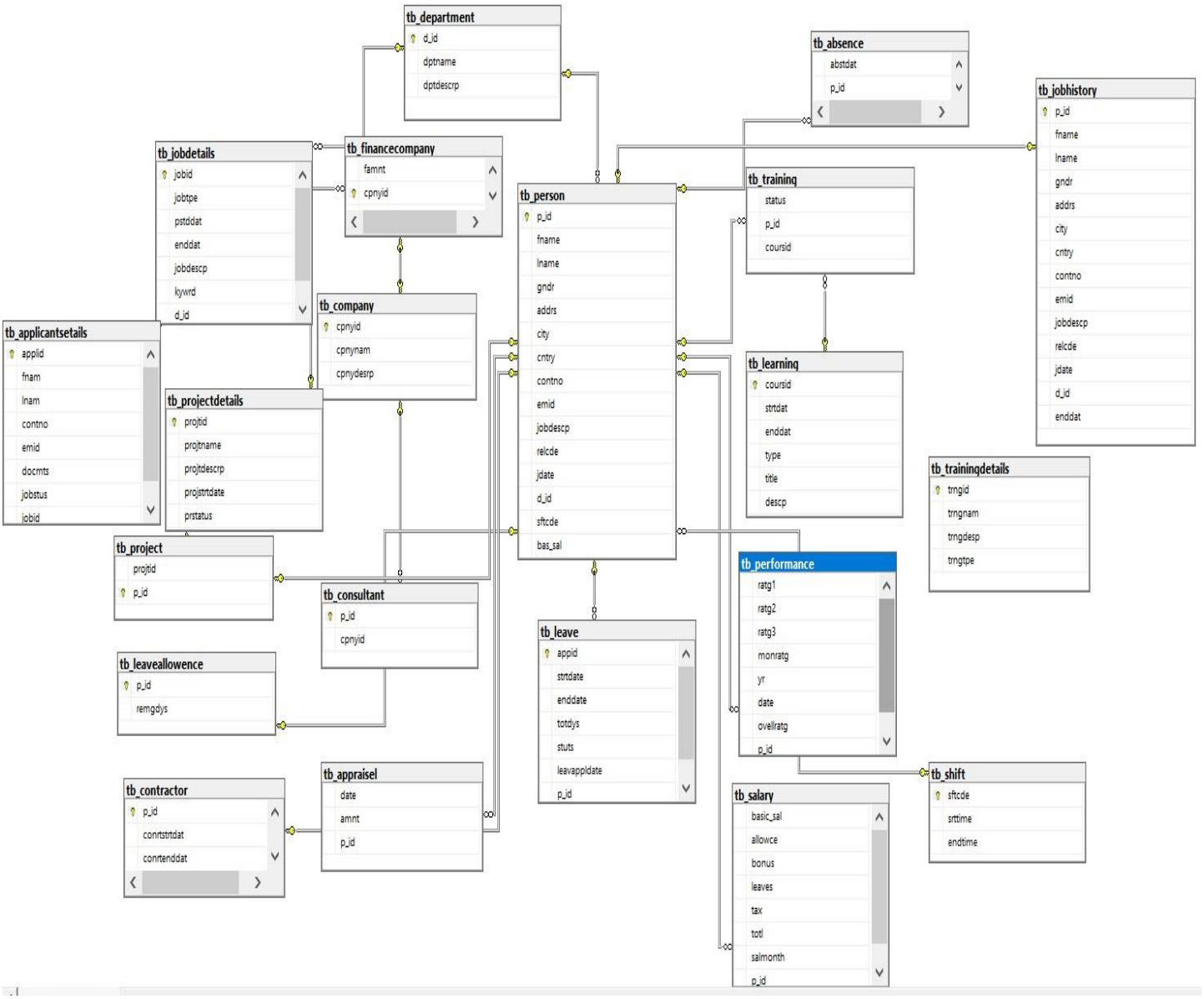
Recruiting manager, human resource manager, HR professionals, HTM	To analyze organizational need, recruitment method, social media, marketing strategies, orientation program, applicant selection methods, job advertisement	Applicant Hiring (Talent acquisition Management)	Found the desired talent and skilled employees	Consultants, social media, job websites, job seekers, applicants



HR professionals, business experts, Vendors	Analyze need of training and learning, requirement of learning, skills, employer knowledge, abilities, mentoring	Training & Learning management	Enhance employees skills, Assessment, motivation, future development,	New employees, existing employees,

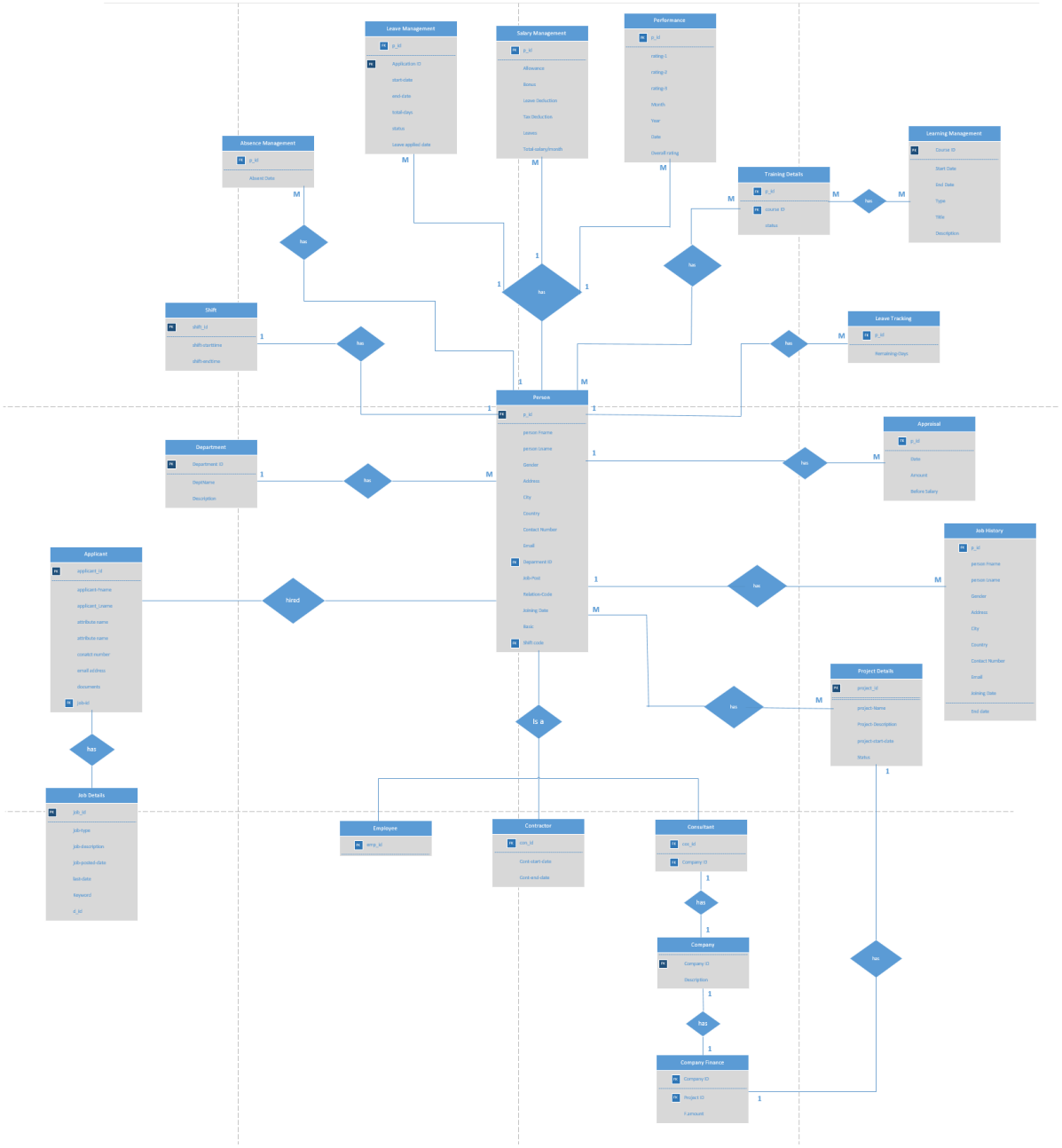

Identify the need of training & skills requirement
Design training method according to the need of the employee
Schedule & Implement the training
Evaluate training outcomes & compare the performance
Feedback
Human capital management, HR professional, HR management
Employee pay scale, basic salary and increment information, leaves information, attendance, tax deduction, Bonus
Payroll Management
Maintain monthly salary report, calculate tax deduction & salary, unpaid leaves info, deduct salary for uninformed leaves Leave allowance
Organization Workforce/ Employees
Employees´ information setup process
Leaves, Attendance, Shift and working hours Schedule setup & information
Allowance deduction, tax deduction, Penalties,
Payroll Generation (pay slip, salary sheet, advances & loan info)
Payroll Maintenance
HR professionals, HRM, Reviewing Committee, Strategic management
Rating system, Goal setting, measuring performance comparison,
Performance and Appraisal management
Employee retention, enhance dedication, focus, develop a strong relationship between employees, compensation and incentives, Reward
Workforce/ Employees, Organization
Set performance standard
Discuss performance standards with employees
Measure current performance with standard set by assessment processes
Recognize the gap in employee performance
Appraisal & reward on the basis of employee's performance
HR manager, HR professional, Senior Management
New project, Planning, skilled employees, divide respondsibilities, define deadline,
Project Details
Project succession, Enhance employee motivation and productivity
Workforce/ Employees, Consultants, Contractors
Project initiation & Requirement
Team development, Assign Roles & responsibilities
Project design, Approve design, start Implementing, Testing
Testing, Meetings to evaluate the current position of project,
Testing, Execution

3. ERD (Entity Relationship Diagram) Logical Design:



3.1 ERD (Entity Relationship Diagram) Flowchart:

ERD is the basic model to represent the relations between the entities with the specification of Primary key and Foreign Key. We have established the relations between the entities – Person, Employee, Contractor, Consultant, Company, Finance Company, Department, Shift details, Leave Management, Salary Management, Performance management, Training details, Leave Tracking, Appraisal, Job History and Project details.


Leave Management
Salary Management
Performance
Absence Management
Learning Management
Training Details
has
Shift
Leave Tracking
Person
Department
Appraisal
Applicant
Job History
hired
Project Details
is a
Job Details
Employee
Contractor
Consultant
Company
Company Finance

4. HRMS Database design and Functionality:

4.1 Database tables:

4.1.1 Person table:

```
CREATE TABLE [dbo].[tb_person] (

    [p_id]     NUMERIC (18)  NOT NULL,

    [fname]    VARCHAR (50)  NOT NULL,

    [lname]    VARCHAR (50)  NOT NULL,

    [gndr]     CHAR (10)     NOT NULL,

    [addrs]    NVARCHAR (50) NOT NULL,

    [city]     NVARCHAR (50) NOT NULL,

    [cntry]    NVARCHAR (50) NOT NULL,

    [contno]   NUMERIC (18)  NOT NULL,

    [emid]     NVARCHAR (50) NOT NULL,

    [jobdescp] NVARCHAR (50) NOT NULL,

    [relcde]   CHAR (10)     NOT NULL,

    [jdate]    DATE          NOT NULL,

    [d_id]     NUMERIC (18)  NOT NULL,

    [sftcde]   NUMERIC (18)  NOT NULL,

    [bas_sal]  NUMERIC (18)  NULL,

    PRIMARY KEY CLUSTERED ([p_id] ASC),

    CONSTRAINT [FK_tb_person1] FOREIGN KEY ([sftcde]) REFERENCES [dbo].[tb_shift] ([sftcde]),

    CONSTRAINT [FK_tb_person] FOREIGN KEY ([d_id]) REFERENCES [dbo].[tb_department] ([d_id])

);
```

4.2.2 Training Details:

```
CREATE TABLE [dbo].[tb_trainingdetails] (

    [trngid]   NUMERIC (18) NOT NULL,

    [trngnam]  VARCHAR (50) NOT NULL,

    [trngdesp] VARCHAR (50) NOT NULL,
```

```
    [trngtpe]  NCHAR (10)   NOT NULL,
    PRIMARY KEY CLUSTERED ([trngid] ASC)
);
```

4.2.3 Training Assignment

```
CREATE TABLE [dbo].[tb_training] (
    [status]  VARCHAR (50) NOT NULL,
    [p_id]    NUMERIC (18) NOT NULL,
    [coursid] NUMERIC (18) NOT NULL,
    CONSTRAINT [FK_tb_training] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id]),
    CONSTRAINT [FK_tb_training1] FOREIGN KEY ([coursid]) REFERENCES [dbo].[tb_learning] ([coursid])
);
```

4.2.4 Shift

```
CREATE TABLE [dbo].[tb_shift] (
    [sftcde]  NUMERIC (18) NOT NULL,
    [srttime] NUMERIC (18) NOT NULL,
    [endtime] NUMERIC (18) NOT NULL,
    PRIMARY KEY CLUSTERED ([sftcde] ASC)
);
```

4.2.5 Salary

```
CREATE TABLE [dbo].[tb_salary] (
    [basic_sal] NUMERIC (18) NOT NULL,
    [allowce]   NUMERIC (18) NOT NULL,
    [bonus]     NUMERIC (18) NOT NULL,
    [leaves]    NUMERIC (18) NOT NULL,
    [tax]       NUMERIC (18) NOT NULL,
    [totl]      NUMERIC (18) NOT NULL,
```

```
    [salmonth]  DATE         NOT NULL,
    [p_id]      NUMERIC (18) NOT NULL,
    CONSTRAINT [FK_tb_salary] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
);
```

4.2.6 Project Details

```
CREATE TABLE [dbo].[tb_projectdetails] (
    [projtid]      NUMERIC (18)  NOT NULL,
    [projtname]    NVARCHAR (50) NOT NULL,
    [projtdescrp]  VARCHAR (50)  NOT NULL,
    [projstrtdate] DATE          NOT NULL,
    [prstatus]     NCHAR (10)    NULL,
    PRIMARY KEY CLUSTERED ([projtid] ASC)
);
```

4.2.7 Project Assignment

```
CREATE TABLE [dbo].[tb_project] (
    [projtid] NUMERIC (18) NOT NULL,
    [p_id]    NUMERIC (18) NOT NULL,
    PRIMARY KEY CLUSTERED ([p_id] ASC),
    CONSTRAINT [FK_tb_project] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id]),
    CONSTRAINT [FK_prjctid] FOREIGN KEY ([projtid]) REFERENCES [dbo].[tb_projectdetails] ([projtid])
);
```

4.2.8 Performance

```
CREATE TABLE [dbo].[tb_performance] (
    [ratg1]     NCHAR (10)   NOT NULL,
    [ratg2]     NCHAR (10)   NOT NULL,
```

```
    [ratg3]     NCHAR (10)   NOT NULL,
    [monratg]   NCHAR (10)   NOT NULL,
    [yr]        NCHAR (10)   NOT NULL,
    [date]      DATE         NOT NULL,
    [ovellratg] NCHAR (10)   NOT NULL,
    [p_id]      NUMERIC (18) NOT NULL,
    CONSTRAINT [FK_tb_performance] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
);
```

4.2.9 Leave Allowance

```
CREATE TABLE [dbo].[tb_leaveallowence] (
    [p_id]    NUMERIC (18) NOT NULL,
    [remgdys] NUMERIC (18) NOT NULL,
    PRIMARY KEY CLUSTERED ([p_id] ASC),
    CONSTRAINT [FK_tb_leaveallowence_ToTable] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person]
([p_id])
);
```

4.2.10 Leave record Table

```
CREATE TABLE [dbo].[tb_leave] (
    [appid]        NUMERIC (18) NOT NULL,
    [strtdate]     DATE         NOT NULL,
    [enddate]      DATE         NOT NULL,
    [totdys]       NUMERIC (18) NOT NULL,
    [stuts]        CHAR (10)    NULL,
    [leavappldate] DATE         NOT NULL,
    [p_id]         NUMERIC (18) NOT NULL,
    CONSTRAINT [PK_tb_leave] PRIMARY KEY CLUSTERED ([appid] ASC),
    CONSTRAINT [FK_tb_leave] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
```

```
);
```

4.2.11 Course

```
CREATE TABLE [dbo].[tb_learning] (
    [coursid] NUMERIC (18) NOT NULL,
    [strtdat] DATE         NOT NULL,
    [enddat]  DATE         NOT NULL,
    [type]    VARCHAR (50) NOT NULL,
    [title]   VARCHAR (50) NOT NULL,
    [descp]   VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([coursid] ASC)
);
```

4.2.12 Job History

```
CREATE TABLE [dbo].[tb_jobhistory] (
    [p_id]     NUMERIC (18)  NOT NULL,
    [fname]    VARCHAR (50)  NOT NULL,
    [lname]    VARCHAR (50)  NOT NULL,
    [gndr]     CHAR (10)     NOT NULL,
    [addrs]    NVARCHAR (50) NOT NULL,
    [city]     NVARCHAR (50) NOT NULL,
    [cntry]    NVARCHAR (50) NOT NULL,
    [contno]   NUMERIC (18)  NOT NULL,
    [emid]     NVARCHAR (50) NOT NULL,
    [jobdescp] NVARCHAR (50) NOT NULL,
    [relcde]   CHAR (10)     NOT NULL,
    [jdate]    DATE          NOT NULL,
    [d_id]     NUMERIC (18)  NOT NULL,
```

```
    [enddat]   DATE          NOT NULL,
    PRIMARY KEY CLUSTERED ([p_id] ASC),
    CONSTRAINT [FK_tb_jobhistory] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
);
```

4.2.13 Job Details

```
CREATE TABLE [dbo].[tb_jobdetails] (
    [jobid]    NUMERIC (18)  NOT NULL,
    [jobtpe]   NVARCHAR (50) NOT NULL,
    [pstddat]  DATE          NOT NULL,
    [enddat]   DATE          NOT NULL,
    [jobdescp] VARCHAR (50)  NOT NULL,
    [kywrd]    NCHAR (10)    NOT NULL,
    [d_id]     NUMERIC (18)  NOT NULL,
    PRIMARY KEY CLUSTERED ([jobid] ASC),
    CONSTRAINT [FK_tb_jobdetails] FOREIGN KEY ([d_id]) REFERENCES [dbo].[tb_department] ([d_id])
);
```

4.2.14 Finance:

```
CREATE TABLE [dbo].[tb_financecompany] (
    [famnt]   NUMERIC (18) NOT NULL,
    [cpnyid]  NUMERIC (18) NOT NULL,
    [projtid] NUMERIC (18) NOT NULL,
    PRIMARY KEY CLUSTERED ([cpnyid] ASC),
    CONSTRAINT [FK_tb_financecompany] FOREIGN KEY ([cpnyid]) REFERENCES [dbo].[tb_company]
([cpnyid]),

    CONSTRAINT [FK_tb_financecompany1] FOREIGN KEY ([projtid]) REFERENCES [dbo].[tb_projectdetails]
([projtid])
```

```
);
```

4.2.15 Department

```
CREATE TABLE [dbo].[tb_department] (
    [d_id]      NUMERIC (18) NOT NULL,
    [dptname]   VARCHAR (50) NOT NULL,
    [dptdescrp] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([d_id] ASC)
);
```

4.2.16 Contractor

```
CREATE TABLE [dbo].[tb_contractor] (
    [p_id]         NUMERIC (18) NOT NULL,
    [conrtstrtdat] DATE         NOT NULL,
    [conrtenddat]  DATE         NOT NULL,
    PRIMARY KEY CLUSTERED ([p_id] ASC),
    CONSTRAINT [FK_tb_contractor] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
);
```

4.2.17 Consultant:

```
CREATE TABLE [dbo].[tb_consultant] (
    [p_id]   INT          NOT NULL,
    [cpnyid] NUMERIC (18) NOT NULL,
    PRIMARY KEY CLUSTERED ([p_id] ASC),
    CONSTRAINT [FK_tb_consultant] FOREIGN KEY ([cpnyid]) REFERENCES [dbo].[tb_company] ([cpnyid])
);
```

4.2.18 Company

```
CREATE TABLE [dbo].[tb_company] (
    [cpnyid]    NUMERIC (18) NOT NULL,
    [cpnynam]   VARCHAR (50) NOT NULL,
    [cpnydesrp] VARCHAR (50) NOT NULL,
    PRIMARY KEY CLUSTERED ([cpnyid] ASC)
);
```

4.2.19 Appraisal

```
CREATE TABLE [dbo].[tb_appraisel] (
    [date] DATE         NOT NULL,
    [amnt] NUMERIC (18) NOT NULL,
    [p_id] NUMERIC (18) NOT NULL,
    CONSTRAINT [FK_tb_appraisel] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])
);
```

4.2.20 Applicant Details

```
CREATE TABLE [dbo].[tb_applicantsetails] (
    [applid]  NUMERIC (18)  NOT NULL,
    [fnam]    VARCHAR (50)  NOT NULL,
    [lnam]    VARCHAR (50)  NOT NULL,
    [contno]  NCHAR (10)    NOT NULL,
    [emid]    NVARCHAR (50) NOT NULL,
    [docmts]  NVARCHAR (50) NOT NULL,
    [jobstus] NCHAR (10)    NOT NULL,
```

```
    [jobid]   NUMERIC (18)  NOT NULL,

    PRIMARY KEY CLUSTERED ([applid] ASC),

    CONSTRAINT [FK_tb_applicantsetails] FOREIGN KEY ([jobid]) REFERENCES [dbo].[tb_jobdetails]
([jobid])

);
```

4.2.21 Absence

```
CREATE TABLE [dbo].[tb_absence] (

    [abstdat] DATE         NOT NULL,

    [p_id]    NUMERIC (18) NOT NULL,

    CONSTRAINT [FK_tb_absence] FOREIGN KEY ([p_id]) REFERENCES [dbo].[tb_person] ([p_id])

);
```

4.2 Stored Procedures:

4.2.1 SP 01- selecting First name:

// Selecting fname

```
        CREATE PROCEDURE SelectAllPerson

        @fname varchar(50)

AS

begin

        SELECT* from tb_person where fname=@fname

End
```

4.2.2 SP 02: Selecting Last name:

// Selecting lname

```
CREATE PROCEDURE SelectAllPerson1

@1name varchar(50)
```

```
AS

begin

SELECT* from tb_person where lname=@lname

End
```

4.2.3 SP: 03 Selecting Person-ID

```
// Selecting person id

CREATE PROCEDURE serch_prsn1

   @pid numeric

AS

begin

SELECT * from tb_person where p_id=@pid

End
```

4.2.4 SP: 04 Selecting Country

```
// Selecting Country
CREATE PROCEDURE serch_prsn1
   @cntry varchar(50)
AS
begin
   SELECT * from tb_person where cntry=@cntry
   End
```

4.2.5 SP: 05 Selecting City

```
// Selecting City

CREATE PROCEDURE serch_prsn2

   @city numeric

AS

begin

   SELECT * from tb_person where city=@city

   End
```

4.2.6 SP: 06 Selecting Contact# & Email Id

Selecting Contact no. and mail ID

```
CREATE PROCEDURE serch_prsn3
   @cont numeric, @mil nvarchar(50)
AS
begin
   SELECT * from tb_person where contno=@contno and emid=@emid
   End
```

4.2.7 SP: 07 Selecting Gender

// Selecting Gender

```
CREATE PROCEDURE serch_prsn4
   @gen char(10)
AS
begin
   SELECT * from tb_person where gndr=@gen
   End
```

4.2.8 SP: 08 selecting Shift ST & ET According to person ID

// selecting shift start and end time according to the Person ID

```
CREATE PROCEDURE serch_prsn5
   @sfcd numeric, @pid numeric
AS
begin
   SELECT tb_shift.srttime, tb_shift.endtime
```

```
   from tb_shift, tb_person
   where tb_shift.sftcde = tb_person.sftcde and
       tb_person.p_id=@sfcd
   End
```

4.2.9 SP: 09 Selecting Contractor SD & ED as/person Id

```
// Selecting contractor start and end date as per the person ID
CREATE PROCEDURE serch_prsn6
   @contr numeric, @pid numeric
AS
begin
   SELECT tb_contractor.conrtstrtdat, tb_contractor.conrtenddat
   from tb_contractor, tb_person
   where tb_contractor.conid = tb_person.p_id and
       tb_person.p_id=@contr
   End
```

4.2.10 SP: 10 Using Inner Joins

```
// Using inner joins
 CREATE PROCEDURE serch_prsn7
   @empID numeric, @departID numeric
AS
begin
   SELECT tb_person.p_id
   from tb_person
     inner join tb_department
```

```
    on tb_person.persondeptID=tb_department.d_id
  End
```

4.2.11 SP: 11 select all record from person and contractor by id

```
//*Select all record from person and contractor by id*/
CREATE PROCEDURE cntctrbyid
  @id numeric,
Begin
select tb_person.* from tb_person,tb_contractor
where tb_person.p_id=@id
AND tb_person.p_id=tb_contractor.p_id
End
```

4.2.12 SP:12 Select Record from Person & Consultant by Id

```
//*Select all record from person and consultant by id*/
CREATE PROCEDURE cnsltntbyid
  @id numeric,
Begin
AS select tb_person.* from tb_person, tb_consultant
 where tb_person.p_id=@id
AND tb_person.p_id= tb_consultant.cpnyid
End
```

4.2.13 SP: 13 Retrieve Salary from Person table

```
//RetriveSalary from table person
AS
Begin
```

```
Select bas_sal from tb_person

where p_id="+Convert.ToInt16(textBox1.Text)+""

End
```

4.2.14 SP: 14 update salary when appraisal given

```
//* Update salary when appraisal is given */

CREATE PROCEDURE cnsltntbyid

   @id numeric,
@sal numeric

AS

Begin

update tb_person set bas_sal=@sal+(select bas_sal from tb_person

 where p_id=@id )

where p_id=@id

End
```

4.2.15 SP: 15 Absence Retrieval

```
//* To retrieve the total no of absence in a month */

CREATE PROCEDURE clcltabsn

   @mnth INT,

   @id INT
AS

Begin

   SELECT COUNT(*) AS aa FROM dbo.tb_absence

WHERE (SELECT DATEPART(MONTH,tb_absence.abstdat) from tb_absence)= @mnth

AND p_id=@id;

RETURN aa

End
```

4.2.16 SP: 16 Select all list of Employee

```
//* Select All list of employe */

AS

Begin

select * from tb_person

where p_id=" + Convert.ToInt16(txt_eid.Text) + "

AND relcde='"+cde+"'";

End
```

4.3 Triggers:

4.3.1 Trigger for a employee's record after Updation , Insertion or Deletion

```
Create trigger dbo.tb_person_trigger
on dbo.tb_person
after UPDATE, INSERT, DELETE
as
declare @p_id numeric (18,0) ,@user varchar(20), @activity
varchar(20);
if exists(SELECT * from inserted) and exists (SELECT * from deleted)

Begin

   SET @activity = &#39;UPDATE&#39;;
   SET @user = SYSTEM_USER;
   SELECT @p_id = p_id from inserted i;
   INSERT into dbo.tb_person_Audit(p_id,Activity, DoneBy) values

   (@p_id,@activity,@user);

End

If exists (Select * from inserted) and not exists(Select * from deleted)

Begin

    SET @activity = &#39;INSERT&#39;;
```

```
        SET @user = SYSTEM_USER;

        SELECT @p_id = p_id from inserted i;
        INSERT into dbo.tb_person_Audit(p_id,Activity, DoneBy)
values(@p_id,@activity,@user);
End
    If exists(select * from deleted) and not exists(Select * from inserted)
Begin
        SET @activity = &#39;DELETE&#39;;
        SET @user = SYSTEM_USER;
        SELECT @p_id = p_id from deleted i;
        INSERT into dbo.tb_person_Audit (p_id,Activity, DoneBy)
        values(@p_id,@activity,@user);
End
```

4.3.2 Trigger for a employee's record who left the company

```
CREATE TRIGGER dbo.tb_person_job_history_trigger
    ON dbo.tb_person
    AFTER DELETE
AS
    BEGIN
        SET NOCOUNT ON;

        DECLARE @p_id numeric (18,0)

        SELECT @p_id = dbo.tb_person.p_id
FROM dbo.tb_person
        INSERT INTO dbo.tb_jobhistory (p_id,enddat,fname,lname,jdate)
        Select p_id,getdate(),fname,lname,jdate from dbo.tb_person a
        where a.p_id = @p_id
```

END

4.4 Database Functionality:

- Insert/update/Delete new Entry of Employ/contractor/Consultant
- Add salary of Employ/contractor at the time of Insertion
- Automatic Allocate Number of Allowed Leaves at the time of Insertion
- Tracking the information of Department
- Keeping record of new job position
- Keeping records of applicant of applied for job
- Assign the courses to the person
- View performance and give appraisal to person
- Automated process to update the salary at the time of Appraisal
- Keep record of Leave application and make a decision whether it approved or not
- Count the person's absence in month and calculate automated deduction for salary
- Automated salary calculation
- Manage finances for company and project
- Keeping record of person even after leaving the job
- And other automation of manual function of HR System

5. Conclusion:

Overall human resource management system (HRMS) is systematic way to maintain current employees as well as new employees. This project is all about to maintain existing employees personal detail like name, address, contact, job start date, job end date, email id etc. the system is also maintaining employees payroll, leaves, absence, performance, incentives, reward and project details. Furthermore, administration (HR) can easily search, update, insert, and delete person`s information. Stored procedures, triggers and all the database functionality has provided in the project. Database tables' creation code and connectivity has mentioned in the project in detail.