Database System
Payroll Management System

Introduction:

Payroll Management System is basically an idea to transform all the manual HR activities into an automated system.

Purpose of PMS:

The new system must include the following:

- Allow to only admin to login and control the record of all.
- Ability to interface with the existing data.
- Ability to automate calculate the promotion of the employee and salary on the basis of attendance.
- Generate report, these report contain monthly attendance, leaves , and salary.

Motivation behind PMS:

Every organization have HR activities to run their business smoothly. So there is a huge market out there for this Payroll Management System. This system allows a company to plan its HR costs more effectively, as well as to manage them and control them without needing to allocate too many resources toward them. This Software enables HR managers to create, edit and analyze reports more accurately.

Objective of PMS

It save many hours of HR mangers from doing pencil and paper work to save the information manually and then checking those thousands of document and make reports from them is a complete headache. So Payroll system make HR activities more efficient and effective, allow them to share information with greater ease.

Scope of the PMS:

The scope of the Payroll system is very widely used. There are some aspect focused like in the personal aspects this is concerned with selection, payroll, attendance and leaves, new registration of employee, resignation of employee etc. From industrial aspect this covers union management relations joint discussion, collective bargaining, injustice and disciplinary procedures, settlement of disputes etc. Payroll system includes the efficient and effective approach to any industry or any company management. It avoids the papers work, papers work can be lost but if this info has back end then company never feel any crises of information of employees

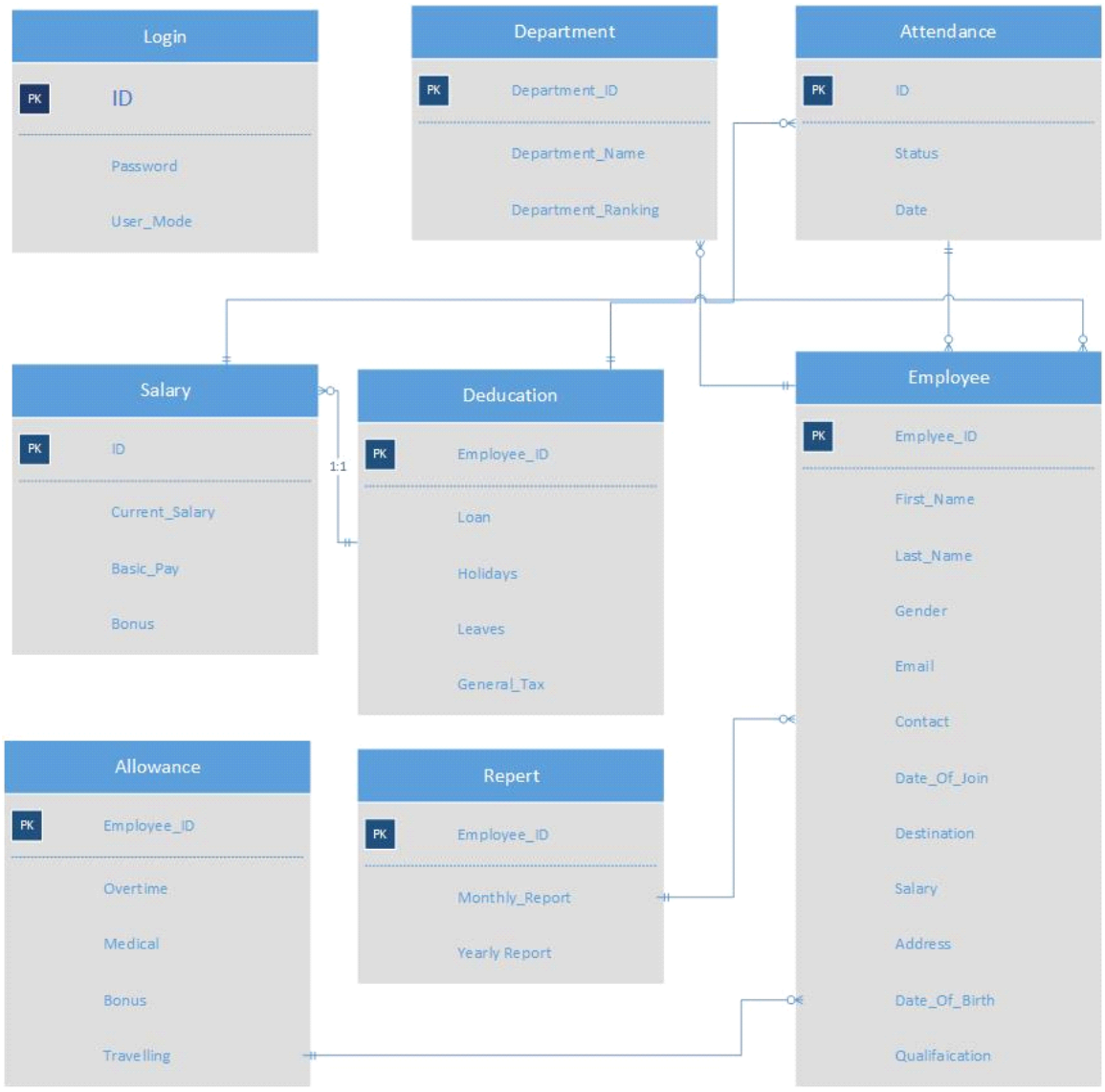
Requirement Specification:

This system perform the following functionality

- Employee
    - Add
    - Update
    - Remove
- User
    - Add
    - Update
    - Remove
- Attendance
    - Add
    - Update
    - Report
- Salary
    - Generate Salary

- Report

ERD:



Relational Schema

DDL:

Tables:

Login:

Create:

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

```
    [EMPLOYEE_ID] [int] NOT NULL,
    [PASSWORD] [varchar](50) NOT NULL
) ON [PRIMARY]
```

Alter:

```
ALTER PROCEDURE [dbo].[Pro_Login]
    -- Add the parameters for the stored procedure here
AS
BEGIN
CREATE TABLE [dbo].[LOGINN](
    [EMPLOYEE_ID] [int] NOT NULL,
    [PASSWORD] [varchar](50) NOT NULL
) ON [PRIMARY]

END
```

Employee:

```
CREATE PROCEDURE [dbo].[Pro_Employee]
    -- Add the parameters for the stored procedure here
AS
BEGIN
CREATE TABLE [dbo].[Employee](
    [Employee_ID] [int] NOT NULL,
    [Salary_ID] [int] NOT NULL,
    [Department_ID] [int] NOT NULL,
    [Allowance_ID] [int] NOT NULL,
    [Report_ID] [int] NOT NULL,
    [FName] [varchar](50) NOT NULL,
    [LName] [varchar](50) NOT NULL,
    [Email] [varchar](50) NOT NULL,
    [Contact] [char](11) NOT NULL,
    [DOJ] [datetime] NOT NULL,
```

```
	[Destination] [varchar](50) NOT NULL,
	[Gender] [bit] NOT NULL,
 CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED
(
	[Employee_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]


SET ANSI_PADDING OFF

ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT [FK_Employee_Department] FOREIGN
KEY([Department_ID])
REFERENCES [dbo].[Department] ([Department_ID])

ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_Employee_Department]

ALTER TABLE [dbo].[Employee]  WITH CHECK ADD  CONSTRAINT [FK_Employee_Report] FOREIGN
KEY([Report_ID])
REFERENCES [dbo].[Report] ([Report_ID])

ALTER TABLE [dbo].[Employee] CHECK CONSTRAINT [FK_Employee_Report]
END

GO
```

Department

```
CREATE PROCEDURE [dbo].[Pro_Department]
	-- Add the parameters for the stored procedure here

AS
BEGIN
```

```
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	CREATE TABLE [dbo].[Department](
	[Department_ID] [int] NOT NULL,
	[Employee_ID] [int] NOT NULL,
	[Department_Name] [varbinary](50) NOT NULL,
	[Department_ranking] [int] NOT NULL,
 CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED
(
	[Department_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

END

GO
```

Attendance

```
CREATE PROCEDURE [dbo].[Pro_Attendance]
	-- Add the parameters for the stored procedure here

AS
BEGIN
CREATE TABLE [dbo].[Attendance](
	[Employee_ID] [int] NOT NULL,
	[Salary_ID] [int] NOT NULL,
	[Department_ID] [int] NOT NULL,
	[Status] [bit] NOT NULL,
	[Date] [datetime] NOT NULL,
 CONSTRAINT [PK_Attendance] PRIMARY KEY CLUSTERED
(
```

```
        [Employee_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]



ALTER TABLE [dbo].[Attendance]  WITH CHECK ADD  CONSTRAINT [FK_Attendance_Department]
FOREIGN KEY([Department_ID])

REFERENCES [dbo].[Department] ([Department_ID])


ALTER TABLE [dbo].[Attendance] CHECK CONSTRAINT [FK_Attendance_Department]

aLTER TABLE [dbo].[Attendance]  WITH CHECK ADD  CONSTRAINT [FK_Attendance_Salary] FOREIGN
KEY([Salary_ID])

REFERENCES [dbo].[Salary] ([Salary_ID])


ALTER TABLE [dbo].[Attendance] CHECK CONSTRAINT [FK_Attendance_Salary]

END


GO
```

Deduction:

```
CREATE PROCEDURE [dbo].[Pro_Deduction]
        -- Add the parameters for the stored procedure here
AS
BEGIN
        -- SET NOCOUNT ON added to prevent extra result sets from
        -- interfering with SELECT statements.
        CREATE TABLE [dbo].[Deduction](
        [Employee_ID] [int] NOT NULL,
        [Salary_ID] [int] NOT NULL,
        [Holidays] [int] NOT NULL,
        [Leaves] [int] NOT NULL,
        [GeneralTax] [int] NOT NULL,
```

```
	[Loan] [int] NOT NULL,
 CONSTRAINT [PK_Deduction] PRIMARY KEY CLUSTERED
(
	[Employee_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
END
GO
```

Allowance:

```
CREATE PROCEDURE [dbo].[Pro_Allowance]
	-- Add the parameters for the stored procedure here
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets fr
	CREATE TABLE [dbo].[Allowance](
	[Allowance_ID] [int] NOT NULL,
	[Salary_ID] [int] NOT NULL,
	[Overtime] [int] NOT NULL,
	[Medical] [int] NOT NULL,
	[Bonus] [int] NOT NULL,
	[Travelling] [int] NOT NULL,
 CONSTRAINT [PK_Allowance] PRIMARY KEY CLUSTERED
(
	[Allowance_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]



END
```

```
GO
```

Salary:

```
CREATE PROCEDURE [dbo].[Pro_Salary]


AS

BEGIN

CREATE TABLE [dbo].[Salary](

        [Salary_ID] [int] NOT NULL,

        [Employee_ID] [int] NOT NULL,

        [Basic_Pay] [int] NOT NULL,

        [Bonus] [int] NOT NULL,

        [Current_Salary] [int] NOT NULL,

 CONSTRAINT [PK_Salary] PRIMARY KEY CLUSTERED

(

        [Salary_ID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

END


GO
```

Report

```
CREATE PROCEDURE [dbo].[Pro_Report]

        -- Add the parameters for the stored procedure here

AS

BEGIN

CREATE TABLE [dbo].[Report](

        [Report_ID] [int] NOT NULL,

        [Employee_ID] [int] NOT NULL,

        [YearlyReport] [bit] NOT NULL,

        [MonthlyReport] [bit] NOT NULL,
```

```
 CONSTRAINT [PK_Report] PRIMARY KEY CLUSTERED

(

	[Report_ID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]


ALTER TABLE [dbo].[Report]  WITH CHECK ADD  CONSTRAINT [FK_Report_Employee] FOREIGN
KEY([Employee_ID])

REFERENCES [dbo].[Employee] ([Employee_ID])

ALTER TABLE [dbo].[Report] CHECK CONSTRAINT [FK_Report_Employee]

END


GO
```

DML:

```
string login = "select * from LOGIN where username = '" + user + "' and pass = '" + password + "'";


   string query = "insert into Employee
(Employee_ID,FName,LName,Gender,Email,Contact,DOJ,Destination,Salary,Address,DOB,Qualification)
values (\'" + Convert.ToInt32(id) + "\',\'" + firstname + "\',\'" + lastname + "\',\'" + gender + "\',\'" + email + "\',\'" +
contact + "\',\'" + Convert.ToDateTime(dateofjoin) + "\',\'" + designation + "\',\'" + Convert.ToInt32(salary) + "\',\'"
+ Address + "\',\'" + Convert.ToDateTime(dateofbirth) + "\',\'" + qualification + "\');";


  string  query=   "update Employee set FName = '"+ Fname+"',LName = '"+LName+"' ,Email =
'"+email+"',Contact = '"+contact+"',Destination = '"+designation+"',Address = '"+address+"',Qualification =
'"+qualification+"' where Employee_id= '"+Convert.ToInt32(employeeid);
```