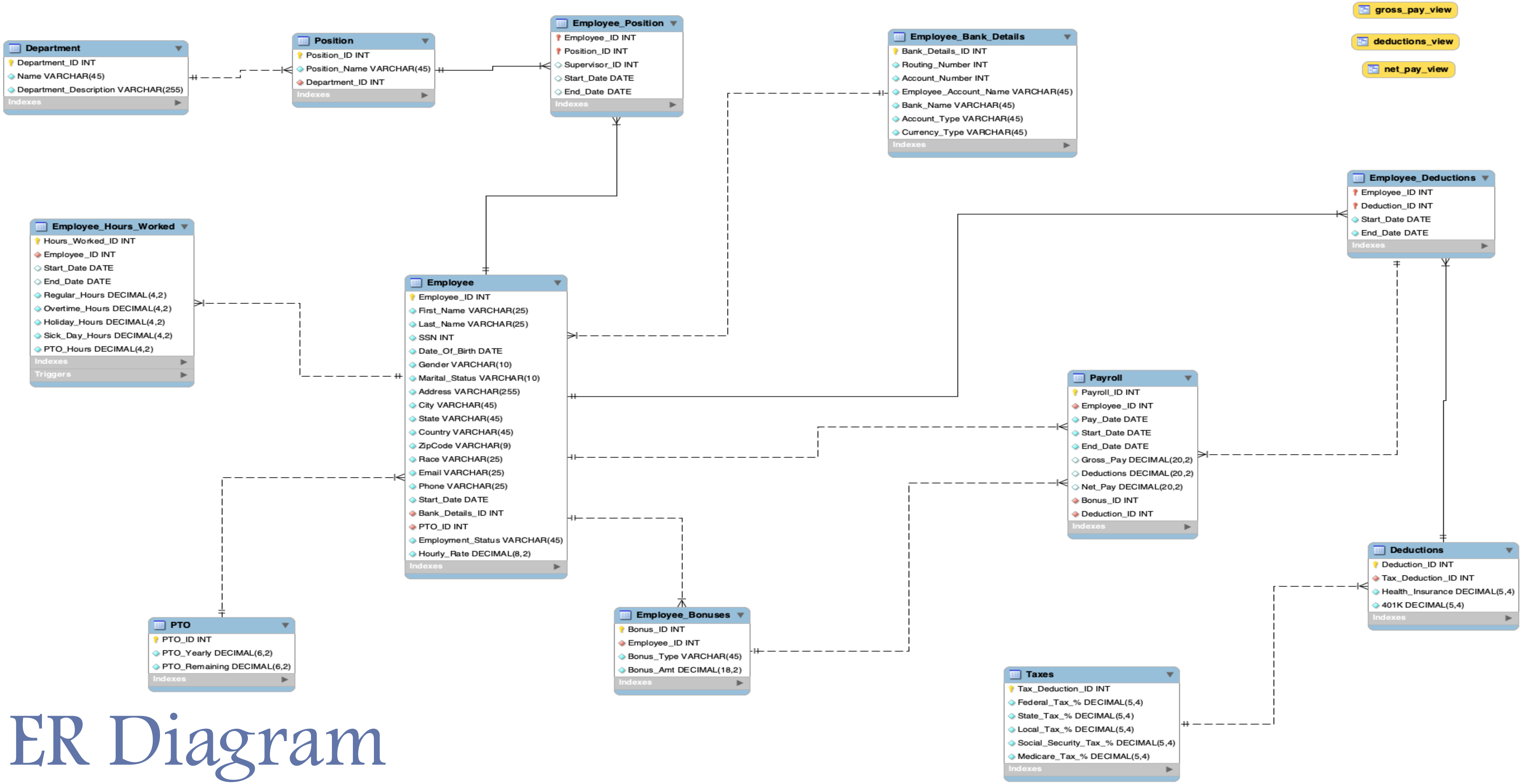
Group 8

Payroll Management System

Business Plan

Problem Statement - Payroll Management System using spreadsheets tends to become messy and time-consuming

Goal - Design, develop and implement a relational database payroll management system to automate the payroll calculation



Department
Department_ID INT
Name VARCHAR(45)
Department_Description VARCHAR(255)
Indexes

Position
Position_ID INT
Position_Name VARCHAR(45)
Department_ID INT
Indexes

Employee_Position
Employee_ID INT
Position_ID INT
Supervisor_ID INT
Start_Date DATE
End_Date DATE
Indexes

Employee_Bank_Details
Bank_Details_ID INT
Routing_Number INT
Account_Number INT
Employee_Account_Name VARCHAR(45)
Bank_Name VARCHAR(45)
Account_Type VARCHAR(45)
Currency_Type VARCHAR(45)
Indexes

gross_pay_view
deductions_view
net_pay_view

Employee_Deductions
Employee_ID INT
Deduction_ID INT
Start_Date DATE
End_Date DATE
Indexes

Employee_Hours_Worked
Hours_Worked_ID INT
Employee_ID INT
Start_Date DATE
End_Date DATE
Regular_Hours DECIMAL(4,2)
Overtime_Hours DECIMAL(4,2)
Holiday_Hours DECIMAL(4,2)
Sick_Day_Hours DECIMAL(4,2)
PTO_Hours DECIMAL(4,2)
Indexes
Triggers

Employee
Employee_ID INT
First_Name VARCHAR(25)
Last_Name VARCHAR(25)
SSN INT
Date_Of_Birth DATE
Gender VARCHAR(10)
Marital_Status VARCHAR(10)
Address VARCHAR(255)
City VARCHAR(45)
State VARCHAR(45)
Country VARCHAR(45)
ZipCode VARCHAR(9)
Race VARCHAR(25)
Email VARCHAR(25)
Phone VARCHAR(25)
Start_Date DATE
Bank_Details_ID INT
PTO_ID INT
Employment_Status VARCHAR(45)
Hourly_Rate DECIMAL(8,2)
Indexes

Payroll
Payroll_ID INT
Employee_ID INT
Pay_Date DATE
Start_Date DATE
End_Date DATE
Gross_Pay DECIMAL(20,2)
Deductions DECIMAL(20,2)
Net_Pay DECIMAL(20,2)
Bonus_ID INT
Deduction_ID INT
Indexes

Deductions
Deduction_ID INT
Tax_Deduction_ID INT
Health_Insurance DECIMAL(5,4)
401K DECIMAL(5,4)
Indexes

PTO
PTO_ID INT
PTO_Yearly DECIMAL(6,2)
PTO_Remaining DECIMAL(6,2)
Indexes

Employee_Bonuses
Bonus_ID INT
Employee_ID INT
Bonus_Type VARCHAR(45)
Bonus_Amt DECIMAL(18,2)
Indexes

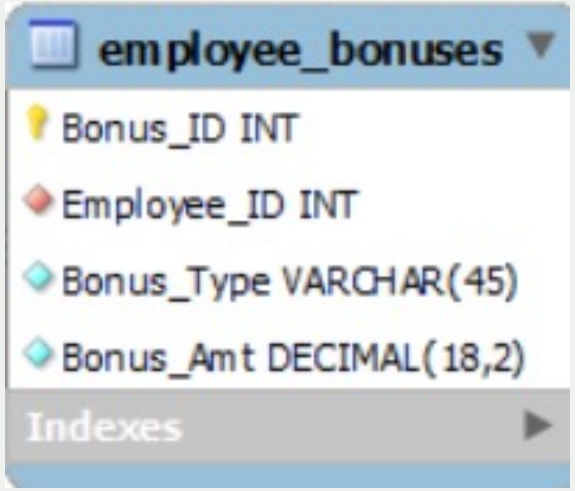
Taxes
Tax_Deduction_ID INT
Federal_Tax_% DECIMAL(5,4)
State_Tax_% DECIMAL(5,4)
Local_Tax_% DECIMAL(5,4)
Social_Security_Tax_% DECIMAL(5,4)
Medicare_Tax_% DECIMAL(5,4)
Indexes

ER Diagram

Relational Data Model

Relation	Degrees	Attributes	Description
EMPLOYEE	20	(Employee_ID, First_Name, Last_Name, SSN, Date_Of_Birth, Gender, Marital_Status, Address, City, State, Country, ZipCode, Race, Email, Phone, Start_Date, Bank_Details_ID, PTO_ID, Employment_Status, Hourly_Rate)	Has key personnel related attributes and keys for tables with similar information for each employee
PTO	3	(PTO_ID, PTO_Yearly, PTO_Remaining)	Has PTO related attributes for each employee.
EMPLOYEE_BANK_DETAILS	7	(Bank_Details_ID, Routing_Number, Account_Number, Employee_Account_Name, Bank_Name, Account_Type, Currency_Type)	Has bank related attributes for each employee
EMPLOYEE_POSTION	5	(EmployeeID, Position_ID, Supervisor_ID, Start_Date, End_Date	Has position specific attributes for each employee.
POSITION	3	(Position_ID, Position_Name, Department_ID)	Has position specific attributes for positions existing a the firm.
DEPARTMENT	3	(Department_ID, Name, Department_Description)	Has department specific attributes for departments existing at the firm.
EMPLOYEE_HOURS_WORKED	9	(Hours_Worked_ID, Employee_ID, Start_Date, End_Date, Regular_Hours, Overtime_Hours, Holiday_Hours, Sick_Day_Hours, PTO_Hours)	Has biweekly hours attributes for each employee tracking the type of hours worked during the period.
PAYROLL	10	(Payroll_ID, Employee_ID, Pay_Date, Start_Date, End_Date, Gross_Pay, Deductions, Net_Pay, Bonus_ID, Deduction_ID)	Has payment attributes made for to each employee biweekly.
EMPLOYEE DEDUCTIONS	4	(Deduction_ID, Employee_ID, Start_Date, End_Date)	Interaction table with EmployeeID and DeductionID attributes.
DEDUCTIONS	4	(Deduction_ID, Tax_Deduction_ID, Health_Insurance, 401K)	Has deduction attributes for each type of deduction available.
TAXES	6	(Tax_Deduction_ID, Federal_Tax_%, State_Tax_%, Local_Tax_%, Social_Security_%, Medicare_Tax)	Has attributes for tax scenarios that apply to employees of the frim.
EMPLOYEE_BONUSES	4	(Bonus_ID, Employee_ID, Bonus_Type, Bonus_Amt)	Has attributes for bonuses awarded to each employee.

3NF Normalization



employee_bonuses

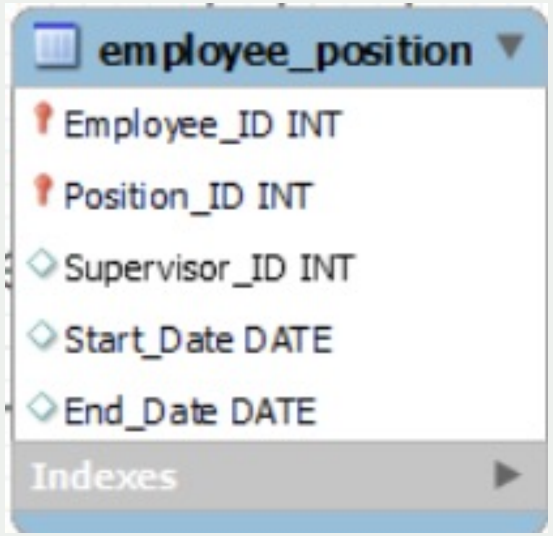
Bonus_ID INT

Employee_ID INT

Bonus_Type VARCHAR(45)

Bonus_Amt DECIMAL(18,2)

Indexes



employee_position

Employee_ID INT

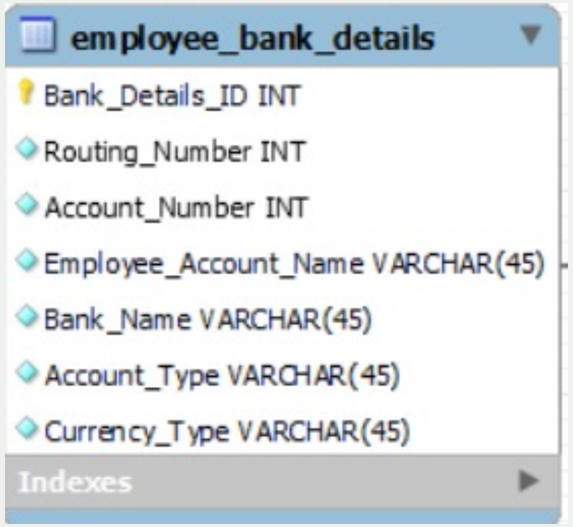
Position_ID INT

Supervisor_ID INT

Start_Date DATE

End_Date DATE

Indexes



employee_bank_details

Bank_Details_ID INT

Routing_Number INT

Account_Number INT

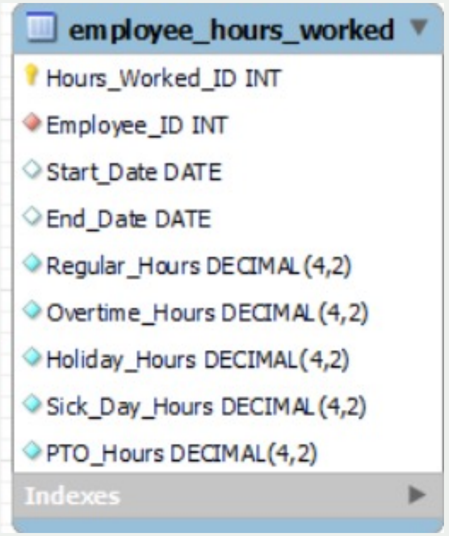
Employee_Account_Name VARCHAR(45)

Bank_Name VARCHAR(45)

Account_Type VARCHAR(45)

Currency_Type VARCHAR(45)

Indexes



employee_hours_worked

Hours_Worked_ID INT

Employee_ID INT

Start_Date DATE

End_Date DATE

Regular_Hours DECIMAL(4,2)

Overtime_Hours DECIMAL(4,2)

Holiday_Hours DECIMAL(4,2)

Sick_Day_Hours DECIMAL(4,2)

PTO_Hours DECIMAL(4,2)

Indexes

Functions

totalHours()

Returns total hours worked based on input that contains the employee's ID, payroll start date and end date.

get_gross_pay()

Returns the gross pay value for the employee based on input employee's ID, payroll start date and end date.

get_deductions()

Returns the deduction value for the employee based on input employee's ID, payroll start date and end date.

get_net_pay()

Returns the net pay value for the employee based on input employee's ID, payroll start date and end date.

Procedures

- HighestPaidEmployee()
  - SQL Call of Stored Procedure for Highest Paid Employee
- pay_calculation()
  - Calls three stored functions, including get_gross_pay(), get_deductions(), and get_net_pay() to fill the values of \`Gross_Pay\`, \`Deductions\` and \`Net_Pay\` in \`Payroll\` table.
- Create_New_Employee()
  - Standardizes Employee Creation and limits access to object creation.
- Create_New_PTO_Record()
  - Creates a new PTO record with standard values.
- Create_New_Emp_Bank_Details_Record()

Triggers

- Trigger 1: Check to see if the remaining PTO is enough for the requested amount **before inserting** data.
- Trigger 2: To Check if the remaining PTO is enough for the requested amount **before updating** data.

Event

- PTO_Annual_Reset
  - Event is written to run each year and the start date is computed based on the day the event is written to the database

Views

GROSS_PAY_VIEW

DEDUCTIONS_VIEW

NET_PAY_VIEW

Assumptions

Assumptions

- Employee and Payroll -- main entities
- Employee Positions
- Employee Pay Unit
- Employee PTO
- Employee Deductions
- Employee Bonuses
- Payroll table – three fields are filled automatically

SQL Show and Tell

SQL Show and Tell

- Min – Method 1 To fill the gross pay, deductions, net pay via three functions and one procedure
- Min – Method 2 To fill the gross pay, deductions, net pay via three views and one update query
- Harry – Create New PTO Record
- Harry – Trigger 1
- Harry – Event 1

Thank you for your listening!
Do you have any questions?