**Main Entities**

**Company Entity**

import jakarta.persistence.\*;

import jakarta.validation.constraints.\*;  
import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Company {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@NotBlank

private String name;

private String photo;

private String address;

@Email

@Column(unique = true)

private String contactEmail;

@Temporal(TemporalType.DATE)

private Date establishmentDate;

}

**Branch Entity**

import jakarta.persistence.\*;

import jakarta.validation.constraints.NotBlank;

import lombok.\*;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Branch {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "company\_id", nullable = false)

private Company company;

@NotBlank

private String name;

private String address;

private String contactInfo;

}

### Department Entity

import jakarta.persistence.\*;

import jakarta.validation.constraints.NotBlank;

import lombok.\*;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "branch\_id", nullable = false)

private Branch branch;

@NotBlank

private String name;

}

### User Entity

import jakarta.persistence.\*;

import jakarta.validation.constraints.\*;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "department\_id", nullable = false)

private Department department;

@NotBlank

@Column(unique = true)

private String userName;

@Email

@NotBlank

@Column(unique = true)

private String email;

@NotBlank

private String password;

private String phone;

@Enumerated(EnumType.STRING)

private WorkSchedule workSchedule;

@Temporal(TemporalType.DATE)

private Date joiningDate;

@DecimalMin("0.0")

private Double basicSalary;

@Enumerated(EnumType.STRING)

private Role role;

@Min(0)

private Integer leaveBalance;

}

### Attendance Entity

import jakarta.persistence.\*;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Attendance {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@Temporal(TemporalType.DATE)

private Date date;

@Temporal(TemporalType.TIME)

private Date inTime;

@Temporal(TemporalType.TIME)

private Date outTime;

private Double workingHours;

private Double overtime;

}

### Advance Salary Entity

import jakarta.persistence.\*;

import jakarta.validation.constraints.\*;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class AdvanceSalary {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@DecimalMin("0.0")

private Double amount;

@Temporal(TemporalType.DATE)

private Date date;

@Enumerated(EnumType.STRING)

private ApprovalStatus approvalStatus = ApprovalStatus.PENDING;

@Temporal(TemporalType.DATE)

private Date repaymentBackSchedule;

private Double repaymentBackAmount;

}

**Bonus Entity**

import jakarta.persistence.\*;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Bonus {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@Temporal(TemporalType.DATE)

private Date date;

@Enumerated(EnumType.STRING)

private BonusType type;

private Double bonusAmount;

private Double totalBonus;

}

**Salary Entity**

import jakarta.persistence.\*;

import jakarta.validation.constraints.DecimalMin;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Salary {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@DecimalMin("0.0")

private Double basicSalary;

private Double overtimeSalary;

private Boolean isCurrent = true;

private Double bonus;

private Double advanceSalary;

@Transient

private Double grossSalary;

private Double deductions;

@Transient

private Double netSalary;

@Temporal(TemporalType.DATE)

private Date date;

// the other fields, grossSalary and netSalary are Calculated from service

}

### Payment Receipt Entity

import jakarta.persistence.\*;

import jakarta.validation.constraints.DecimalMin;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class PaymentReceipt {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(unique = true)

private String receiptNumber;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@Temporal(TemporalType.DATE)

private Date date;

@ManyToOne

@JoinColumn(name = "salary\_id", nullable = false)

private Salary salary;

@DecimalMin("0.0")

private Double amount;

}

### Leave Entity

import jakarta.persistence.\*;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Leave {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@Enumerated(EnumType.STRING)

private LeaveType leaveType;

@Temporal(TemporalType.DATE)

private Date startDate;

@Temporal(TemporalType.DATE)

private Date endDate;

@Enumerated(EnumType.STRING)

private LeaveStatus status = LeaveStatus.PENDING;

private Integer leaveBalance;

}

### Feedback Entity

import jakarta.persistence.\*;

import jakarta.validation.constraints.Max;

import jakarta.validation.constraints.Min;

import lombok.\*;

import java.sql.Date;

@Data  
@AllArgsConstructor  
@NoArgsConstructor

@Entity

public class Feedback {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@ManyToOne

@JoinColumn(name = "user\_id", nullable = false)

private User user;

@Min(1)

@Max(5)

private Integer rating;

private String comments;

@Temporal(TemporalType.DATE)

private Date date;

}

@Enumerated(EnumType.STRING)

public enum WorkSchedule {

DAY\_TIME, MORNING\_TIME

}

@Enumerated(EnumType.STRING)

public enum Role {

ADMIN, EMPLOYEE, MANAGER

}

@Enumerated(EnumType.STRING)

public enum ApprovalStatus {

PENDING, APPROVED, REJECTED

}

@Enumerated(EnumType.STRING)

public enum BonusType {

PERFORMANCE\_BONUS, FESTIVAL\_BONUS

}

@Enumerated(EnumType.STRING)

public enum LeaveType {

SICK\_LEAVE, CASUAL\_LEAVE, RESERVE\_LEAVE

}

@Enumerated(EnumType.STRING)

public enum LeaveStatus {

PENDING, APPROVED, REJECTED

}

**DATABASE Queries’**

-- Company Entity

CREATE TABLE Company (

ID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(255) NOT NULL,

Address TEXT,

ContactEmail VARCHAR(255) UNIQUE,

EstablishmentDate DATE

);

-- Branch Entity

CREATE TABLE Branch (

ID INT PRIMARY KEY AUTO\_INCREMENT,

CompanyID INT,

Name VARCHAR(255) NOT NULL,

Address TEXT,

ContactInfo VARCHAR(255),

FOREIGN KEY (CompanyID) REFERENCES Company(ID)

);

-- Department Entity

CREATE TABLE Department (

ID INT PRIMARY KEY AUTO\_INCREMENT,

BranchID INT,

Name VARCHAR(255) NOT NULL,

FOREIGN KEY (BranchID) REFERENCES Branch(ID)

);

-- User Entity

CREATE TABLE User (

ID INT PRIMARY KEY AUTO\_INCREMENT,

DepartmentID INT,

UserName VARCHAR(255) NOT NULL UNIQUE,

Email VARCHAR(255) UNIQUE NOT NULL,

Password VARCHAR(255) NOT NULL,

Phone VARCHAR(15),

WorkSchedule ENUM('Day Time', 'Morning Time') NOT NULL,

JoiningDate DATE,

BasicSalary DECIMAL(10, 2),

Role ENUM('Admin', 'Employee', 'Manager') NOT NULL,

LeaveBalance INT DEFAULT 0,

FOREIGN KEY (DepartmentID) REFERENCES Department(ID)

);

-- Attendance Entity

CREATE TABLE Attendance (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Date DATE NOT NULL,

InTime TIME NOT NULL,

OutTime TIME NOT NULL,

WorkingHours DECIMAL(5, 2),

Overtime DECIMAL(5, 2),

FOREIGN KEY (UserID) REFERENCES User(ID)

);

-- Advance Salary Entity

CREATE TABLE AdvanceSalary (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Amount DECIMAL(10, 2) NOT NULL,

Date DATE NOT NULL,

ApprovalStatus ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending',

RepaymentBackSchedule DATE,

RepaymentBackAmount DECIMAL(10, 2),

FOREIGN KEY (UserID) REFERENCES User(ID)

);

-- Bonus Entity

CREATE TABLE Bonus (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Date DATE NOT NULL,

Type ENUM('Performance Bonus', 'Festival Bonus') NOT NULL,

BonusAmount DECIMAL(10, 2),

TotalBonus DECIMAL(10, 2) GENERATED ALWAYS AS (BonusAmount) STORED,

FOREIGN KEY (UserID) REFERENCES User(ID)

);

-- Salary Entity

CREATE TABLE Salary (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

BasicSalary DECIMAL(10, 2) NOT NULL,

OvertimeSalary DECIMAL(10, 2),

isCurrent BOOLEAN DEFAULT TRUE,

Bonus DECIMAL(10, 2),

AdvanceSalary DECIMAL(10, 2),

GrossSalary DECIMAL(10, 2) GENERATED ALWAYS AS (BasicSalary + OvertimeSalary + Bonus) STORED,

Deductions DECIMAL(10, 2),

NetSalary DECIMAL(10, 2) GENERATED ALWAYS AS (GrossSalary - Deductions) STORED,

Date DATE NOT NULL,

FOREIGN KEY (UserID) REFERENCES User(ID)

);

-- Payment Receipt Entity

CREATE TABLE PaymentReceipt (

ID INT PRIMARY KEY AUTO\_INCREMENT,

ReceiptNumber VARCHAR(255) NOT NULL UNIQUE,

UserID INT,

Date DATE NOT NULL,

SalaryID INT,

Amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (UserID) REFERENCES User(ID),

FOREIGN KEY (SalaryID) REFERENCES Salary(ID)

);

-- Leave Entity

CREATE TABLE Leave (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

LeaveType ENUM('Sick Leave', 'Casual Leave', 'Reserve Leave') NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

Status ENUM('Pending', 'Approved', 'Rejected') DEFAULT 'Pending',

LeaveBalance INT,

FOREIGN KEY (UserID) REFERENCES User(ID)

);

-- Feedback Entity

CREATE TABLE Feedback (

ID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Rating INT CHECK (Rating >= 1 AND Rating <= 5),

Comments TEXT,

Date DATE NOT NULL,

FOREIGN KEY (UserID) REFERENCES User(ID)

);

**General Entities**

Company:

(ID)

(Name)

(Address)

(Contact Information)/Email

establishmentDate

Branch:

(ID)

(Company ID)

(Name)

(Address)

(Contact Information)

Department:

(ID)

(Branch ID)

(Name)

User:

(ID)

(Department ID)

(Name)/UserName

(Email)

Password

Phone

Work Schedule (Day time or Morning Time)

(Joining Date)

Basic Salary (Hourly)

Role (Admin, Employee, Manager)

Leave Balance

Attendance:

(ID)

(User ID)

(Date)

In Time

Out Time

(Working Hours)/ Total Hours

(Overtime)

Advance Salary:

(ID)

(User ID)

(Amount)

(Date)

Approval Status (Pending, Approved, Rejected)

Repayment Back Schedule

Repayment Back Amount

Bonus:

(ID)

(User ID)

(Date)

Type (e.g., performance bonus, festival bonus),

festival Bonus Amount / Performance Bonus Amount

Total Bonus

Salary:

(ID)

(User ID)

(Monthly Salary)/Basic Salary

(Overtime Salary)

isCurrent (boolean)

(Bonus)

advance Salary

Gross Salary

Deductions

(Net Salary)

(Date)

Payment Receipt:

(ID)

Receipt Number

(User ID)

(Date)

Salary ID / (Amount)

Leave:

(ID)

(User ID)

(Leave Type) (e.g., sick leave, Casual/ Reserve Leave)

(Start Date)

(End Date)

Status (Pending, Approved, Rejected)

(Leave Balance) \*\*\*\*

Feedback:

(ID)

(User ID)

Rating

feedback/(Comments)

(Date)