A payroll system typically involves multiple tables to manage various aspects of employee compensation, taxation, and related information. Here are five common tables you might find in a payroll system:

1. **Employee Information Table:** This table stores basic details about each employee, such as their name, employee ID, contact information, job title, department, and hire date. It serves as a central repository of employee data.
2. **Salary and Compensation Table:** This table tracks the compensation details for each employee. It includes information about their salary or hourly wage, pay grade, overtime rates, bonuses, and other forms of compensation.
3. **Time and Attendance Table:** This table records the hours worked by employees. It includes details about regular hours, overtime hours, time off taken (such as vacation or sick leave), and any other attendance-related data. This information is crucial for accurate salary calculation.
4. **Tax and Deductions Table:** Here, tax-related information for each employee is stored. This includes tax withholding details, such as federal, state, and local tax rates, as well as other deductions like health insurance premiums, retirement contributions, and other voluntary deductions.
5. **Payroll Processing Log Table:** This table maintains a log of payroll processing activities. It includes records of each payroll run, including the date, pay period covered, total wages, taxes withheld, deductions, and net pay for each employee. Keeping a history of payroll runs is vital for auditing and reporting purposes.

These are just a few examples of tables that might exist in a payroll system. Depending on the complexity of the organization and its payroll needs, there could be additional tables for more specialized information, such as employee benefits, pension plans, or even integration with external systems for financial reporting.

**create database payroll\_functions\_db;**

**use payroll\_functions\_db;**

**-- Employee Information Table**

**CREATE TABLE Employee (**

**employee\_id INT PRIMARY KEY,**

**first\_name VARCHAR(50),**

**last\_name VARCHAR(50),**

**contact\_number VARCHAR(15),**

**job\_title VARCHAR(100),**

**department VARCHAR(100),**

**hire\_date DATE**

**);**

**-- Salary and Compensation Table**

**CREATE TABLE Compensation (**

**compensation\_id INT PRIMARY KEY,**

**employee\_id INT,**

**base\_salary DECIMAL(10, 2),**

**overtime\_rate DECIMAL(5, 2),**

**bonus DECIMAL(10, 2),**

**FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)**

**);**

**-- Time and Attendance Table**

**CREATE TABLE Attendance (**

**attendance\_id INT PRIMARY KEY,**

**employee\_id INT,**

**work\_date DATE,**

**regular\_hours DECIMAL(5, 2),**

**overtime\_hours DECIMAL(5, 2),**

**time\_off\_hours DECIMAL(5, 2),**

**FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)**

**);**

**-- Tax and Deductions Table**

**CREATE TABLE Deductions (**

**deduction\_id INT PRIMARY KEY,**

**employee\_id INT,**

**federal\_tax\_rate DECIMAL(5, 2),**

**state\_tax\_rate DECIMAL(5, 2),**

**health\_insurance DECIMAL(10, 2),**

**retirement\_contribution DECIMAL(10, 2),**

**FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)**

**);**

**-- Payroll Processing Log Table**

**CREATE TABLE PayrollLog (**

**payroll\_id INT PRIMARY KEY,**

**payroll\_date DATE,**

**pay\_period\_start DATE,**

**pay\_period\_end DATE,**

**employee\_id INT,**

**total\_wages DECIMAL(10, 2),**

**taxes\_withheld DECIMAL(10, 2),**

**deductions DECIMAL(10, 2),**

**net\_pay DECIMAL(10, 2),**

**FOREIGN KEY (employee\_id) REFERENCES Employee(employee\_id)**

**);**

**INSERT INTO Employee (employee\_id, first\_name, last\_name, contact\_number, job\_title, department, hire\_date)**

**VALUES**

**(1, 'John', 'Doe', '123-456-7890', 'Sales Manager', 'Sales', '2022-01-15'),**

**(2, 'Jane', 'Smith', '987-654-3210', 'Marketing Specialist', 'Marketing', '2021-08-20'),**

**(3, 'Michael', 'Johnson', '555-123-4567', 'Software Engineer', 'Engineering', '2022-03-10'),**

**(4, 'Sarah', 'Williams', '333-555-8888', 'HR Specialist', 'HR', '2021-11-05'),**

**(5, 'David', 'Miller', '444-222-1111', 'Financial Analyst', 'Finance', '2021-12-02'),**

**(6, 'Emily', 'Brown', '222-333-4444', 'Senior Designer', 'Design', '2021-10-18'),**

**(7, 'Daniel', 'Wilson', '666-777-8888', 'Marketing Manager', 'Marketing', '2022-04-25'),**

**(8, 'Olivia', 'Martinez', '111-222-3333', 'Sales Representative', 'Sales', '2022-05-12'),**

**(9, 'Aiden', 'Lee', '888-777-5555', 'Software Developer', 'Engineering', '2021-09-08'),**

**(10, 'Sophia', 'Garcia', '555-666-9999', 'Finance Manager', 'Finance', '2022-06-30'),**

**(11, 'Liam', 'Nguyen', '666-555-4444', 'HR Coordinator', 'HR', '2022-02-15'),**

**(12, 'Isabella', 'Rodriguez', '222-444-6666', 'Designer', 'Design', '2021-07-11'),**

**(13, 'Mason', 'Gonzalez', '333-222-5555', 'Software Engineer', 'Engineering', '2021-03-28'),**

**(14, 'Ava', 'Lopez', '444-555-6666', 'Sales Associate', 'Sales', '2022-07-05'),**

**(15, 'Noah', 'Perez', '555-444-3333', 'Marketing Coordinator', 'Marketing', '2021-04-22'),**

**(16, 'Emma', 'Hernandez', '111-333-5555', 'Designer', 'Design', '2022-03-10'),**

**(17, 'Sophia', 'Gonzalez', '777-888-4444', 'HR Manager', 'HR', '2021-01-02'),**

**(18, 'Jackson', 'Johnson', '666-555-2222', 'Software Developer', 'Engineering', '2022-02-18'),**

**(19, 'Oliver', 'Smith', '333-222-7777', 'Financial Analyst', 'Finance', '2022-06-15'),**

**(20, 'Lucas', 'Lee', '555-777-9999', 'Designer', 'Design', '2021-09-30');**

**INSERT INTO Compensation (compensation\_id, employee\_id, base\_salary, overtime\_rate, bonus)**

**VALUES**

**(1, 1, 75000.00, 1.5, 5000.00),**

**(2, 2, 60000.00, 1.25, 3000.00),**

**(3, 3, 80000.00, 2.0, 7000.00),**

**(4, 4, 55000.00, 1.2, 2000.00),**

**(5, 5, 65000.00, 1.3, 4000.00),**

**(6, 6, 70000.00, 1.4, 3500.00),**

**(7, 7, 72000.00, 1.5, 4200.00),**

**(8, 8, 58000.00, 1.2, 2500.00),**

**(9, 9, 85000.00, 1.6, 6000.00),**

**(10, 10, 68000.00, 1.3, 3800.00),**

**(11, 11, 52000.00, 1.1, 1800.00),**

**(12, 12, 70000.00, 1.4, 3200.00),**

**(13, 13, 75000.00, 1.5, 4500.00),**

**(14, 14, 62000.00, 1.25, 2700.00),**

**(15, 15, 60000.00, 1.3, 3300.00),**

**(16, 16, 68000.00, 1.35, 3900.00),**

**(17, 17, 50000.00, 1.1, 1600.00),**

**(18, 18, 80000.00, 1.6, 5200.00),**

**(19, 19, 72000.00, 1.4, 3700.00),**

**(20, 20, 67000.00, 1.3, 3400.00);**

**INSERT INTO Attendance (attendance\_id, employee\_id, work\_date, regular\_hours, overtime\_hours, time\_off\_hours)**

**VALUES**

**(1, 1, '2023-08-01', 8.0, 2.0, 0.0),**

**(2, 2, '2023-08-01', 8.0, 1.5, 0.0),**

**(3, 3, '2023-08-01', 8.0, 3.0, 0.0),**

**(4, 4, '2023-08-01', 8.0, 1.0, 0.0),**

**(5, 5, '2023-08-01', 8.0, 1.0, 0.0),**

**(6, 6, '2023-08-01', 8.0, 0.0, 0.0),**

**(7, 7, '2023-08-01', 8.0, 1.5, 0.0),**

**(8, 8, '2023-08-01', 8.0, 1.0, 0.0),**

**(9, 9, '2023-08-01', 8.0, 2.5, 0.0),**

**(10, 10, '2023-08-01', 8.0, 1.5, 0.0),**

**(11, 11, '2023-08-01', 8.0, 0.5, 0.0),**

**(12, 12, '2023-08-01', 8.0, 1.0, 0.0),**

**(13, 13, '2023-08-01', 8.0, 2.0, 0.0),**

**(14, 14, '2023-08-01', 8.0, 1.0, 0.0),**

**(15, 15, '2023-08-01', 8.0, 1.5, 0.0),**

**(16, 16, '2023-08-01', 8.0, 1.0, 0.0),**

**(17, 17, '2023-08-01', 8.0, 0.5, 0.0),**

**(18, 18, '2023-08-01', 8.0, 2.5, 0.0),**

**(19, 19, '2023-08-01', 8.0, 1.0, 0.0),**

**(20, 20, '2023-08-01', 8.0, 1.5, 0.0);**

**INSERT INTO Deductions (deduction\_id, employee\_id, federal\_tax\_rate, state\_tax\_rate, health\_insurance, retirement\_contribution)**

**VALUES**

**(1, 1, 0.15, 0.05, 100.00, 200.00),**

**(2, 2, 0.12, 0.04, 80.00, 150.00),**

**(3, 3, 0.18, 0.06, 120.00, 250.00),**

**(4, 4, 0.10, 0.03, 70.00, 130.00),**

**(5, 5, 0.14, 0.04, 90.00, 180.00),**

**(6, 6, 0.15, 0.05, 95.00, 200.00),**

**(7, 7, 0.16, 0.05, 100.00, 210.00),**

**(8, 8, 0.11, 0.03, 75.00, 140.00),**

**(9, 9, 0.20, 0.07, 130.00, 270.00),**

**(10, 10, 0.13, 0.04, 85.00, 170.00),**

**(11, 11, 0.09, 0.03, 60.00, 120.00),**

**(12, 12, 0.15, 0.05, 100.00, 190.00),**

**(13, 13, 0.17, 0.05, 110.00, 220.00),**

**(14, 14, 0.12, 0.04, 80.00, 160.00),**

**(15, 15, 0.14, 0.04, 90.00, 180.00),**

**(16, 16, 0.13, 0.04, 85.00, 170.00),**

**(17, 17, 0.08, 0.02, 50.00, 100.00),**

**(18, 18, 0.19, 0.06, 125.00, 260.00),**

**(19, 19, 0.16, 0.05, 100.00, 200.00),**

**(20, 20, 0.14, 0.04, 90.00, 180.00);**

**INSERT INTO PayrollLog (payroll\_id, payroll\_date, pay\_period\_start, pay\_period\_end, employee\_id, total\_wages, taxes\_withheld, deductions, net\_pay)**

**VALUES**

**(1, '2023-08-31', '2023-08-16', '2023-08-31', 1, 5000.00, 800.00, 400.00, 3800.00),**

**(2, '2023-08-31', '2023-08-16', '2023-08-31', 2, 4000.00, 600.00, 300.00, 3100.00),**

**(3, '2023-08-31', '2023-08-16', '2023-08-31', 3, 5500.00, 900.00, 450.00, 4150.00),**

**(4, '2023-08-31', '2023-08-16', '2023-08-31', 4, 3800.00, 560.00, 280.00, 2960.00),**

**(5, '2023-08-31', '2023-08-16', '2023-08-31', 5, 4500.00, 720.00, 360.00, 3420.00),**

**(6, '2023-08-31', '2023-08-16', '2023-08-31', 6, 4800.00, 800.00, 400.00, 3600.00),**

**(7, '2023-08-31', '2023-08-16', '2023-08-31', 7, 5200.00, 850.00, 425.00, 3925.00),**

**(8, '2023-08-31', '2023-08-16', '2023-08-31', 8, 4100.00, 610.00, 305.00, 3185.00),**

**(9, '2023-08-31', '2023-08-16', '2023-08-31', 9, 6000.00, 1000.00, 500.00, 4500.00),**

**(10, '2023-08-31', '2023-08-16', '2023-08-31', 10, 4700.00, 730.00, 365.00, 3605.00),**

**(11, '2023-08-31', '2023-08-16', '2023-08-31', 11, 3500.00, 520.00, 260.00, 2720.00),**

**(12, '2023-08-31', '2023-08-16', '2023-08-31', 12, 4800.00, 800.00, 400.00, 3600.00),**

**(13, '2023-08-31', '2023-08-16', '2023-08-31', 13, 5200.00, 850.00, 425.00, 3925.00),**

**(14, '2023-08-31', '2023-08-16', '2023-08-31', 14, 3900.00, 570.00, 285.00, 3045.00),**

**(15, '2023-08-31', '2023-08-16', '2023-08-31', 15, 4000.00, 600.00, 300.00, 3100.00),**

**(16, '2023-08-31', '2023-08-16', '2023-08-31', 16, 4600.00, 740.00, 370.00, 3490.00),**

**(17, '2023-08-31', '2023-08-16', '2023-08-31', 17, 3300.00, 480.00, 240.00, 2580.00),**

**(18, '2023-08-31', '2023-08-16', '2023-08-31', 18, 5800.00, 920.00, 460.00, 4420.00),**

**(19, '2023-08-31', '2023-08-16', '2023-08-31', 19, 5100.00, 810.00, 405.00, 3885.00),**

**(20, '2023-08-31', '2023-08-16', '2023-08-31', 20, 4800.00, 800.00, 400.00, 3600.00);**

**Exercise 1: Question: Retrieve the absolute difference between the base salary and health insurance deduction for employee 1.**

**SELECT**

**employee\_id,**

**ABS((select**

**base\_salary**

**from compensation where employee\_id = 1) - health\_insurance) AS abs\_difference**

**FROM Deductions WHERE employee\_id = 1;**

**Answer: This query retrieves the absolute difference between the base salary and health insurance deduction for employee 1.**

**Exercise 2: Question: Retrieve the rounded total wages for employee 3.**

**SELECT employee\_id, ROUND(total\_wages) AS rounded\_wages FROM PayrollLog WHERE employee\_id = 3;**

**Answer: This query retrieves the rounded total wages for employee 3.**

**Exercise 3: Question: Retrieve the ceiling of the overtime pay (overtime hours \* overtime rate) for employee 5.**

**SELECT**

**employee\_id,**

**CEIL((select**

**overtime\_hours**

**from Attendance where employee\_id=5**

**) \* overtime\_rate) AS ceil\_overtime\_pay**

**FROM Compensation WHERE employee\_id = 5;**

**Answer: This query retrieves the ceiling of the calculated overtime pay for employee 5.**

**Exercise 4: Question: Retrieve the floor of the net pay (total wages - deductions) for employee 7.**

**SELECT employee\_id, FLOOR(total\_wages - deductions) AS floor\_net\_pay FROM PayrollLog WHERE employee\_id = 7;**

**Answer: This query retrieves the floor of the net pay for employee 7.**

**Exercise 5: Question: Retrieve the remainder after dividing the total wages by 1000 for employee 10.**

**SELECT employee\_id, MOD(total\_wages, 1000) AS wage\_remainder FROM PayrollLog WHERE employee\_id = 10;**

**Answer: This query retrieves the remainder after dividing the total wages by 1000 for employee 10.**

**Exercise 6: Question: Retrieve the result of raising 2 to the power of 4.**

**SELECT POWER(2, 4) AS power\_result;**

**Answer: This query calculates 2 raised to the power of 4.**

**Exercise 7: Question: Retrieve the square root of the regular hours worked by employee 13.**

**SELECT employee\_id, SQRT(regular\_hours) AS sqrt\_regular\_hours FROM Attendance WHERE employee\_id = 13;**

**Answer: This query retrieves the square root of the regular hours worked by employee 13.**

**Exercise 8: Question: Retrieve a random number between 1 and 100.**

**SELECT FLOOR(RAND() \* 100) + 1 AS random\_number;**

**Answer: This query generates a random number between 1 and 100 using the RAND function.**

**Exercise 9: Question: Retrieve the absolute value of the difference between the total wages and deductions for employee 15.**

**SELECT employee\_id, ABS(total\_wages - deductions) AS abs\_wage\_deduction\_diff FROM PayrollLog WHERE employee\_id = 15;**

**Answer: This query retrieves the absolute value of the difference between the total wages and deductions for employee 15.**

**Exercise 10: Question: Retrieve the ceiling value of the square root of the base salary for employee 20.**

**SELECT employee\_id, CEIL(SQRT(base\_salary)) AS ceil\_sqrt\_base\_salary FROM Employee WHERE employee\_id = 20;**

**Answer: This query retrieves the ceiling value of the square root of the base salary for employee 20.**

**Exercise 11: Question: Retrieve the square root of the sum of base salary and bonus for employee 7.**

**SELECT employee\_id, SQRT(base\_salary + bonus) AS sqrt\_sum\_salary\_bonus FROM Compensation WHERE employee\_id = 7;**

**Exercise 12: Question: Calculate the result of raising the total wages to the power of 2 for employees with a job title containing "Engineer."**

**SELECT employee\_id, POWER(total\_wages, 2) AS power\_total\_wages FROM PayrollLog WHERE job\_title LIKE '%Engineer%';**

**Exercise 13: Question: Calculate the sum of the absolute differences between overtime hours and regular hours for all employees.**

**SELECT SUM(ABS(overtime\_hours - regular\_hours)) AS sum\_abs\_hour\_diff FROM Attendance;**

**Exercise 14: Question: Retrieve the ceiling of the division of total wages by base salary for employees hired before '2023-06-01.'**

**SELECT employee\_id, CEIL(total\_wages / base\_salary) AS ceil\_wages\_salary\_ratio FROM PayrollLog WHERE hire\_date < '2023-06-01';**

**Exercise 15: Question: Calculate the square root of the product of base salary and bonus for employee 12.**

**SELECT employee\_id, SQRT(base\_salary \* bonus) AS sqrt\_product\_salary\_bonus FROM Compensation WHERE employee\_id = 12;**

**Exercise 16: Question: Determine the floor of the division of total wages by state tax rate for employees with a department containing "Sales."**

**SELECT employee\_id, FLOOR(total\_wages / state\_tax\_rate) AS floor\_wages\_tax\_ratio FROM PayrollLog WHERE department LIKE '%Sales%';**

**Exercise 17: Question: Calculate the sum of the square roots of federal tax rate for employees with an employee ID greater than 10.**

**SELECT SUM(SQRT(federal\_tax\_rate)) AS sum\_sqrt\_federal\_tax FROM Deductions WHERE employee\_id > 10;**

**Exercise 18: Question: Retrieve the ceiling of the difference between overtime hours and regular hours for employee 19.**

**SELECT employee\_id, CEIL(overtime\_hours - regular\_hours) AS ceil\_hour\_diff FROM Attendance WHERE employee\_id = 19;**

**Exercise 19: Question: Calculate the absolute difference between the power of total wages and base salary for employees with a bonus greater than 100.**

**SELECT employee\_id, ABS(POWER(total\_wages, 2) - base\_salary) AS abs\_power\_wages\_salary\_diff FROM PayrollLog WHERE bonus > 100;**

**Exercise 20: Question: Retrieve the floor of the square root of federal tax rate for employees with a hire date before '2023-03-01.'**

**SELECT employee\_id, FLOOR(SQRT(federal\_tax\_rate)) AS floor\_sqrt\_federal\_tax FROM Deductions WHERE hire\_date < '2023-03-01';**

**Exercise 21: Question: Retrieve employees with a total wages greater than the square of their base salary.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > POWER(base\_salary, 2);**

**Exercise 22: Question: Retrieve employees with a health insurance deduction greater than the cube root of their overtime hours.**

**SELECT \* FROM Employee WHERE (SELECT health\_insurance FROM Deductions WHERE employee\_id = Employee.employee\_id) > POWER((SELECT overtime\_hours FROM Attendance WHERE employee\_id = Employee.employee\_id), 1/3);**

**Exercise 23: Question: Retrieve employees with total wages greater than the sum of their base salary and bonus.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > (SELECT base\_salary + bonus FROM Compensation WHERE employee\_id = Employee.employee\_id);**

**Exercise 24: Question: Retrieve employees with a total wages less than the product of their overtime hours and federal tax rate.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) < (SELECT overtime\_hours \* federal\_tax\_rate FROM Attendance, Deductions WHERE Attendance.employee\_id = Employee.employee\_id AND Deductions.employee\_id = Employee.employee\_id);**

**Exercise 25: Question: Retrieve employees with a total wages greater than the ceiling value of their base salary divided by the square root of their bonus.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > CEIL((SELECT base\_salary / SQRT(bonus) FROM Compensation WHERE employee\_id = Employee.employee\_id));**

**Exercise 26: Question: Retrieve employees with a total wages greater than the result of rounding up the product of their base salary and federal tax rate.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > CEIL((SELECT base\_salary \* federal\_tax\_rate FROM Compensation WHERE employee\_id = Employee.employee\_id));**

**Exercise 27: Question: Retrieve employees with a health insurance deduction greater than the remainder of dividing their bonus by 100.**

**SELECT \* FROM Employee WHERE (SELECT health\_insurance FROM Deductions WHERE employee\_id = Employee.employee\_id) > MOD((SELECT bonus FROM Compensation WHERE employee\_id = Employee.employee\_id), 100);**

**Exercise 28: Question: Retrieve employees with a total wages greater than the square root of the sum of their base salary and overtime pay.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > SQRT((SELECT base\_salary + (overtime\_hours \* overtime\_rate) FROM Compensation, Attendance WHERE Compensation.employee\_id = Employee.employee\_id AND Attendance.employee\_id = Employee.employee\_id));**

**Exercise 29: Question: Retrieve employees with a total wages less than the power of their base salary raised to the floor of their overtime hours.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) < POWER((SELECT base\_salary FROM Compensation WHERE employee\_id = Employee.employee\_id), FLOOR((SELECT overtime\_hours FROM Attendance WHERE employee\_id = Employee.employee\_id)));**

**Exercise 30: Question: Retrieve employees with a total wages greater than a random number between 500 and 1000.**

**SELECT \* FROM Employee WHERE (SELECT total\_wages FROM PayrollLog WHERE employee\_id = Employee.employee\_id) > RAND() \* (1000 - 500) + 500;**