A Report

on

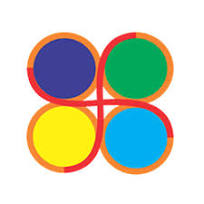
Micro Project

Employee Payroll Schema

**Subject: Relational Database Management System**

**Subject Code: 4330702**

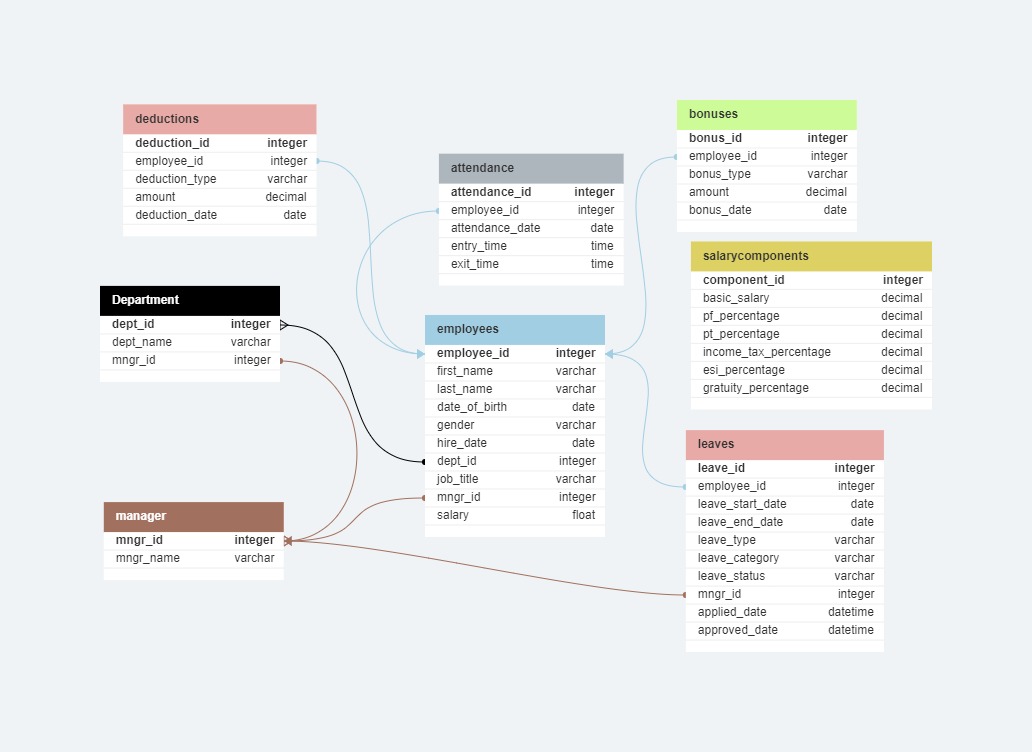
**Branch: Computer Engineering**



**Government Polytechnic, Ahmedabad**

**Team:**

* **Kavan Bhavsar - 226170307015**
* **Bhavya Bagadia - 226170307004**
* **Khush Panchal - 226170307075**



Introduction:

The given database schema comprises several tables that together manage and store information about an employee payroll system. Here's a summary breakdown of each table's purpose:

1. Manager Table (manager):

- Tracks information about managers in the company.

- Attributes: manager ID (unique identifier), manager name.

2. Department Table (department): [Not explicitly created in the provided schema but referenced in the Employees table]

- Holds information about departments in the company.

3. Employees Table (employees):

- Stores details about individual employees.

- Attributes: employee ID (unique identifier), first and last name, date of birth, gender, hire date, department ID (linked to department table), job title, manager ID, salary.

- Foreign key references: department ID links to department table, manager ID references manager table.

4. Leaves Table (leaves):

- Manages employees' leave-related data.

- Attributes: leave ID (unique identifier), employee ID (linked to employees table), leave start and end date, leave type, category, status, manager ID, applied and approved dates.

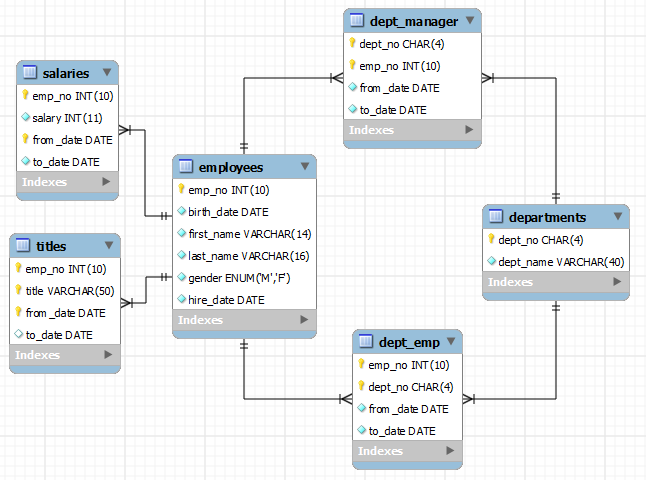
- Foreign key references: employee ID links to employees table, manager ID references manager table.

5. Deductions Table (deductions):

- Records deductions made from employee salaries.

- Attributes: deduction ID (unique identifier), employee ID (linked to employees table), deduction type, amount, deduction date.

- Foreign key references: employee ID links to employees table.



6. Bonuses Table (bonuses):

- Stores information about bonuses given to employees.

- Attributes: bonus ID (unique identifier), employee ID (linked to employees table), bonus type, amount, bonus date.

- Foreign key references: employee ID links to employees table.

7. Attendance Table (attendance):

- Manages employee attendance records.

- Attributes: attendance ID (unique identifier), employee ID (linked to employees table), attendance date, entry and exit times.

- Foreign key references: employee ID links to employees table.

8. Salary Components Table (salarycomponents):

- Stores specific salary components and their percentages.

- Attributes: component ID (unique identifier), basic salary, percentages for PF (Provident Fund), PT (Professional Tax), income tax, ESI (Employee State Insurance), gratuity.

This database setup seems to cover various aspects of managing employee-related data, including their personal details, leaves, attendance, salary components, deductions, bonuses, and managerial information.

# QUERIES:

## To create Database:

    CREATE TABLE manager (

        mngr\_id NUMBER(5) PRIMARY KEY,

        mngr\_name VARCHAR2(250) NOT NULL

    );

    -- Creating Department Table

    -- Creating Employees Table

    CREATE TABLE employees (

        employee\_id NUMBER(5) PRIMARY KEY,

        first\_name VARCHAR2(250) NOT NULL,

        last\_name VARCHAR2(250) NOT NULL,

        date\_of\_birth DATE NOT NULL,

        gender VARCHAR2(210) NOT NULL,

        hire\_date DATE NOT NULL,

        dept\_id NUMBER(5) NOT NULL,

        job\_title VARCHAR2(250) NOT NULL,

        mngr\_id NUMBER(5) NOT NULL,

        salary FLOAT(10) NOT NULL,

        FOREIGN KEY (dept\_id) REFERENCES department(dept\_id),

        FOREIGN KEY (mngr\_id) REFERENCES manager(mngr\_id)

        ON DELETE CASCADE

    );

    -- Creating Manager Table

    -- Creating Leaves Table

    CREATE TABLE leaves (

        leave\_id NUMBER(5) PRIMARY KEY,

        employee\_id NUMBER(5) NOT NULL,

        leave\_start\_date DATE NOT NULL,

        leave\_end\_date DATE NOT NULL,

        leave\_type VARCHAR2(50) NOT NULL,

        leave\_category VARCHAR2(50) NOT NULL,

        leave\_status VARCHAR2(20) NOT NULL,

        mngr\_id NUMBER(5) NOT NULL,

        applied\_date TIMESTAMP NOT NULL,

        approved\_date TIMESTAMP NOT NULL,

        FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id),

        FOREIGN KEY (mngr\_id) REFERENCES manager(mngr\_id)

        ON DELETE CASCADE

    );

    -- Creating Deductions Table

    CREATE TABLE deductions (

        deduction\_id NUMBER(5) PRIMARY KEY,

        employee\_id NUMBER(5) NOT NULL,

        deduction\_type VARCHAR2(50) NOT NULL,

        amount DECIMAL(10,2) NOT NULL,

        deduction\_date DATE NOT NULL,

        FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)

        ON DELETE CASCADE

    );

    -- Creating Bonuses Table

    CREATE TABLE bonuses (

        bonus\_id NUMBER(5) PRIMARY KEY,

        employee\_id NUMBER(5) NOT NULL,

        bonus\_type VARCHAR2(50) NOT NULL,

        amount DECIMAL(10,2) NOT NULL,

        bonus\_date DATE NOT NULL,

        FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)

        ON DELETE CASCADE

    );

    -- Creating Attendance Table

    CREATE TABLE attendance (

        attendance\_id NUMBER(5) PRIMARY KEY,

        employee\_id NUMBER(5) NOT NULL,

        attendance\_date DATE NOT NULL,

        entry\_time TIMESTAMP NOT NULL,

        exit\_time TIMESTAMP NOT NULL,

        FOREIGN KEY (employee\_id) REFERENCES employees(employee\_id)

        ON DELETE CASCADE

    );

    -- Creating SalaryComponents Table

    CREATE TABLE salarycomponents (

        component\_id NUMBER(5) PRIMARY KEY,

        basic\_salary DECIMAL(10,2) NOT NULL,

        pf\_percentage DECIMAL(5) NOT NULL,

        pt\_percentage DECIMAL(5) NOT NULL,

        income\_tax\_percentage DECIMAL(5) NOT NULL,

        esi\_percentage DECIMAL(5) NOT NULL,

        gratuity\_percentage DECIMAL(5) NOT NULL

    );

## To insert the data:

INSERT INTO deductions VALUES (201, 1001, 'Health Insurance', 150.00, TO\_DATE('2023-01-15', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (202, 1001, '401(k) Contribution', 200.50, TO\_DATE('2023-02-20', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (203, 1005, 'Tax Withholding', 300.75, TO\_DATE('2023-03-10', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (204, 1003, 'Union Dues', 50.00, TO\_DATE('2023-04-05', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (205, 1006, 'Pension Contribution', 180.25, TO\_DATE('2023-05-12', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (206, 1007, 'Education Expenses', 120.00, TO\_DATE('2023-06-22', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (207, 1008, 'Garnishments', 75.50, TO\_DATE('2023-07-18', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (208, 1010, 'Health Savings Account', 250.75, TO\_DATE('2023-08-30', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (209, 1011, 'Charitable Contributions', 30.00, TO\_DATE('2023-09-12', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (210, 1012, 'Child Support', 100.25, TO\_DATE('2023-10-05', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (211, 1011, 'Employee Stock Purchase Plan', 180.00, TO\_DATE('2023-11-08', 'YYYY-MM-DD'));

INSERT INTO deductions VALUES (212, 1012, 'Dependent Care FSA', 90.50, TO\_DATE('2023-12-20', 'YYYY-MM-DD'));

## PL/SQL block to Find Data:

set serveroutput on;

DECLARE

   v\_employee\_id NUMBER := &v\_employee\_id; -- Replace with an actual employee ID

   v\_manager\_id NUMBER := &v\_manager\_id;    -- Replace with an actual manager ID

   -- Cursor for employee information

   CURSOR emp\_info\_cur IS

      SELECT e.employee\_id, e.first\_name, e.last\_name, d.dept\_name

      FROM employees e

      JOIN department d ON e.dept\_id = d.dept\_id

      WHERE e.employee\_id = v\_employee\_id;

   -- Cursor for approved leaves

   CURSOR approved\_leaves\_cur IS

      SELECT leave\_id, leave\_start\_date, leave\_end\_date, leave\_type, leave\_category

      FROM leaves

      WHERE employee\_id = v\_employee\_id AND leave\_status = 'Approved';

   -- Cursor for attendance details

   CURSOR attendance\_cur IS

      SELECT attendance\_date, entry\_time, exit\_time

      FROM attendance

      WHERE employee\_id = v\_employee\_id;

BEGIN

   -- Retrieve and print employee information

   FOR emp\_info\_rec IN emp\_info\_cur

   LOOP

      DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || emp\_info\_rec.employee\_id);

      DBMS\_OUTPUT.PUT\_LINE('First Name: ' || emp\_info\_rec.first\_name);

      DBMS\_OUTPUT.PUT\_LINE('Last Name: ' || emp\_info\_rec.last\_name);

      DBMS\_OUTPUT.PUT\_LINE('Department: ' || emp\_info\_rec.dept\_name);

   END LOOP;

   -- Retrieve and print approved leaves

   FOR leaves\_rec IN approved\_leaves\_cur

   LOOP

      DBMS\_OUTPUT.PUT\_LINE('Leave ID: ' || leaves\_rec.leave\_id);

      DBMS\_OUTPUT.PUT\_LINE('Start Date: ' || TO\_CHAR(leaves\_rec.leave\_start\_date, 'YYYY-MM-DD'));

      DBMS\_OUTPUT.PUT\_LINE('End Date: ' || TO\_CHAR(leaves\_rec.leave\_end\_date, 'YYYY-MM-DD'));

      DBMS\_OUTPUT.PUT\_LINE('Leave Type: ' || leaves\_rec.leave\_type);

      DBMS\_OUTPUT.PUT\_LINE('Leave Category: ' || leaves\_rec.leave\_category);

   END LOOP;

   -- Retrieve and print attendance details

   FOR attendance\_rec IN attendance\_cur

   LOOP

      DBMS\_OUTPUT.PUT\_LINE('Attendance Date: ' || TO\_CHAR(attendance\_rec.attendance\_date, 'YYYY-MM-DD'));

      DBMS\_OUTPUT.PUT\_LINE('Entry Time: ' || TO\_CHAR(attendance\_rec.entry\_time, 'HH24:MI:SS'));

      DBMS\_OUTPUT.PUT\_LINE('Exit Time: ' || TO\_CHAR(attendance\_rec.exit\_time, 'HH24:MI:SS'));

   END LOOP;

END;

/