

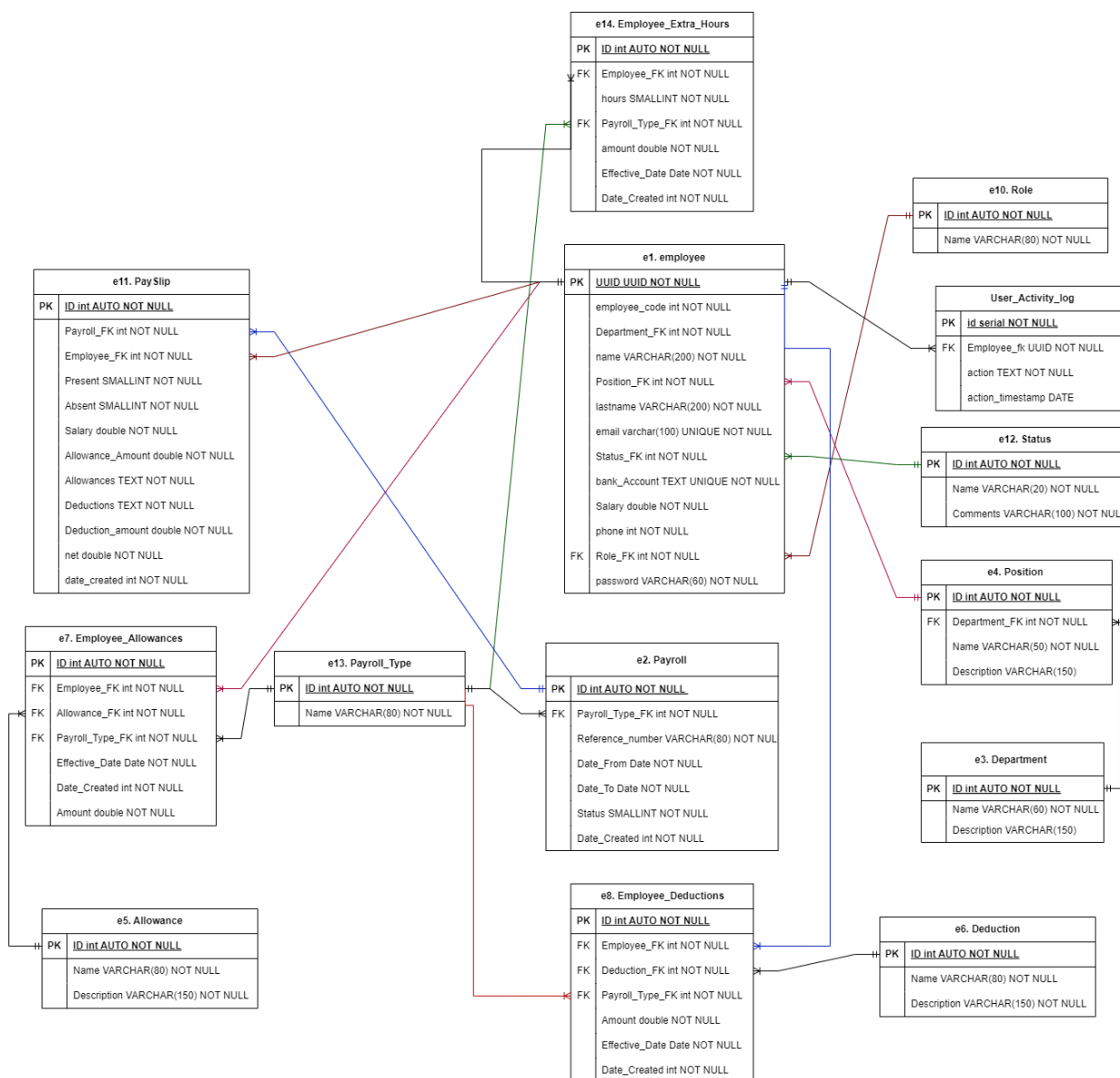
Nicolás Guevara Herrán
nguevarah@udistrital.edu.co

Database Design Overview

The database design for the Payroll Management System was constructed to ensure minimal redundancy and supports the integrity and consistency of the data. The core entities include 'Employee', 'Department', 'Position', 'Allowance', 'Deduction', 'Payroll', 'Payslip', 'Role', and 'UserActivityLog'.

Entity-Relationship Diagram

The ER diagram illustrates the relationships between these entities.



As technical decisions, the Employee table was created using an UUID as an identifier, since, it's not recommended to use personal information of the employee or an auto incremental identifier for security purposes.

The Employee is related to the UserActivityLog and Role table, because the system is intended to be used by the personnel of Trimeca S.A., so it is needed that the employees are the same users that will be interacting with the system. The UserActivityLog was created for security reasons, since it's needed to know who is changing something on the database, and to keep track of the changes.

In some cases, like for example, in the EmployeeExtraHours table, it was used dates in Date and int format (as a timestamp), just because those were topics seen on the course, and wanted to implement them both.

Metadata

Position Table

| Column Name | Data Type | Description |
|---------------|--------------|------------------------------------|
| id | integer | Primary Key |
| name | varchar(50) | Position name |
| description | varchar(150) | Position description |
| department_fk | integer | Foreign key referencing Department |

Role Table

| Column Name | Data Type | Description |
|-------------|-------------|-------------|
| id | integer | Primary Key |
| name | varchar(80) | Role name |

Allowance Table

| Column Name | Data Type | Description |
|-------------|--------------|-----------------------|
| id | integer | Primary Key |
| name | varchar(80) | Allowance name |
| description | varchar(150) | Allowance description |

Deduction Table

| Column Name | Data Type | Description |
|-------------|--------------|-----------------------|
| id | integer | Primary Key |
| name | varchar(80) | Deduction name |
| description | varchar(150) | Deduction description |

Department Table

| Column Name | Data Type | Description |
|-------------|--------------|------------------------|
| id | integer | Primary Key |
| name | varchar(60) | Department name |
| description | varchar(150) | Department description |

Employee Table

| Column Name | Data Type | Description |
|---------------|--------------|----------------------------------|
| UUID | uuid | Primary key |
| employee_code | varchar(10) | Unique employee code |
| name | varchar(200) | Employee first name |
| position_fk | integer | Foreign key referencing Position |
| lastname | varchar(200) | Employee last name |
| email | varchar(100) | Employee email |
| status_fk | integer | Foreign key referencing Status |
| bank_account | text | Employee bank account number |
| salary | numeric | Employee salary |
| phone | varchar(15) | Employee phone number |
| password | varchar(60) | Employee password |
| role_fk | integer | Foreign key referencing Role |

Employee Allowances Table

| Column Name | Data Type | Description |
|-----------------|-----------|-------------------------------------|
| id | integer | Primary key |
| employee_fk | uuid | Foreign key referencing Employee |
| allowance_fk | integer | Foreign key referencing Allowance |
| payroll_type_fk | integer | Foreign key referencing PayrollType |
| effective_date | varchar | Effective date |
| date_created | bigint | Date created |
| amount | numeric | Allowance amount |

Employee Deductions Table

| Column Name | Data Type | Description |
|-----------------|-----------|-------------------------------------|
| id | integer | Primary key |
| employee_fk | uuid | Foreign key referencing Employee |
| deduction_fk | integer | Foreign key referencing Deduction |
| payroll_type_fk | integer | Foreign key referencing PayrollType |
| effective_date | varchar | Effective date |
| date_created | bigint | Date created |
| amount | numeric | Allowance amount |

Employee Extra Hours Table

| Column Name | Data Type | Description |
|-----------------|-----------|-------------------------------------|
| id | integer | Primary key |
| hours | smallint | Number of extra hours |
| amount | numeric | Amount paid for extra hours |
| effective_date | date | Effective date |
| date_created | bigint | Date created |
| employee_fk | uuid | Foreign key referencing Employee |
| payroll_type_fk | integer | Foreign key referencing PayrollType |

Payroll Table

| Column Name | Data Type | Description |
|------------------|-------------|-------------------------------------|
| id | integer | Primary key |
| payroll_type_fk | integer | Foreign key referencing PayrollType |
| reference_number | varchar(80) | Unique reference number |
| status | smallint | Payroll status |
| date_from | date | Start date of the payroll period |
| date_to | date | End date of the payroll period |
| date_created | bigint | Date created |

Payroll Type Table

| Column Name | Data Type | Description |
|-------------|-------------|-------------------|
| id | integer | Primary key |
| name | varchar(80) | Payroll Type name |

Pay slip Table

| Column Name | Data Type | Description |
|------------------|-----------|----------------------------------|
| id | integer | Primary key |
| payroll_fk | integer | Foreign key referencing Payroll |
| employee_fk | uuid | Foreign key referencing Employee |
| present | smallint | Numer of days present |
| absent | smallint | Numer of days absent |
| salary | numeric | salary amount |
| allowance_amount | numeric | Total allowance amount |
| net | numeric | Net salary |
| deduction_amount | numeric | Total deduction amount |
| date_created | date | Date created |

Status Table

| Column Name | Data Type | Description |
|-------------|--------------|---------------------|
| id | integer | Primary key |
| name | varchar(20) | Status name |
| comments | varchar(100) | Additional comments |

User Activity Log Table

| Column Name | Data Type | Description |
|------------------|-----------|--|
| id | integer | Primary key |
| action | text | Description of the action (UPDATE, DELETE, INSERT, etc.) |
| action_timestamp | timestamp | Timestamp of the action |
| employee_fk | uuid | Foreign key referencing Employee |

Web Services Definition

Get Employees with Position and Department

- **Endpoint:** /employees_with_position_and_department
- **Method:** GET
- **Description:** Retrieves a list of all employees along with their positions and departments.

Get Allowances of an Employee

- **Endpoint:** `/employees/{employee_uuid}/allowances`
- **Method:** GET
- **Description:** Retrieves all allowances for a specific employee.
- **Parameters:**
 - `employee_uuid`: UUID of the employee

Get Deductions of an Employee

- **Endpoint:** `/employees/{employee_uuid}/deductions`
- **Method:** GET
- **Description:** Retrieves all deductions for a specific employee.
- **Parameters:**
 - `employee_uuid`: UUID of the employee

Get Payroll Summary for an Employee

- **Endpoint:** `/employees/{employee_uuid}/payroll_summary`
- **Method:** GET
- **Description:** Retrieves the payroll summary for a specific employee.
- **Parameters:**
 - `employee_uuid`: UUID of the employee

Get Employee Payment History

- **Endpoint:** `/employees/{employee_uuid}/payment_history`
- **Method:** GET
- **Description:** Retrieves the payment history for a specific employee.
- **Parameters:**
 - `employee_uuid`: UUID of the employee

Get Employees with Extra Hours

- **Endpoint:** `/employees/extra_hours`
- **Method:** GET
- **Description:** Retrieves employees with extra hours worked in a specific date range.
- **Parameters:**
 - `start_date`: Start date in the format YYYY-MM-DD
 - `end_date`: End date in the format YYYY-MM-DD

Get Employee Counts per Status

- **Endpoint:** `/employees/status_counts`
- **Method:** GET
- **Description:** Retrieves the count of employees per status.

Add Employee

- **Endpoint:** /employees/add
- **Method:** POST
- **Description:** Adds a new employee.

Update Employee Position

- **Endpoint:** /employees/{employee_uuid}/position
- **Method:** PUT
- **Description:** Updates the position of an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Update Employee Salary

- **Endpoint:** /employees/{employee_uuid}/salary
- **Method:** PUT
- **Description:** Updates the salary of an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Update Employee Status

- **Endpoint:** /employees/{employee_uuid}/status
- **Method:** PUT
- **Description:** Updates the status of an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Update Employee Email

- **Endpoint:** /employees/{employee_uuid}/email
- **Method:** PUT
- **Description:** Updates the email of an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Update Employee Phone

- **Endpoint:** /employees/{employee_uuid}/phone
- **Method:** PUT
- **Description:** Updates the phone number of an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Delete Employee

- **Endpoint:** /employees/{employee_uuid}
- **Method:** DELETE
- **Description:** Deletes an employee.
- **Parameters:**
 - employee_uuid: UUID of the employee

Get Departments Report

- **Endpoint:** /departments/report
- **Method:** GET
- **Description:** Retrieves a report of departments with the number of employees and their respective salaries.

Add Department

- **Endpoint:** /departments/add
- **Method:** POST
- **Description:** Adds a new department.

Delete Department

- **Endpoint:** /departments/{department_id}
- **Method:** DELETE
- **Description:** Deletes a department and all associated positions and employees.
- **Parameters:**
 - department_id: ID of the department

Add Position

- **Endpoint:** /positions/add
- **Method:** POST
- **Description:** Adds a new position.

Update Position Title

- **Endpoint:** /positions/{position_id}/title
- **Method:** PUT
- **Description:** Updates the title of a position.
- **Parameters:**
 - position_id: ID of the position

Get Users with Roles

- **Endpoint:** /users_with_roles
- **Method:** GET
- **Description:** Retrieves all users with their roles.

Get User Counts per Role

- **Endpoint:** /users/role_counts
- **Method:** GET
- **Description:** Retrieves the count of users per role.

Add Allowance

- **Endpoint:** /allowances/add
- **Method:** POST
- **Description:** Adds a new allowance.

Delete Employee Allowance

- **Endpoint:** /employee_allowances/{allowance_id}
- **Method:** DELETE
- **Description:** Deletes an allowance for an employee.
- **Parameters:**
 - allowance_id: ID of the allowance

Add Employee Allowance

- **Endpoint:** /employee_allowances/add
- **Method:** POST
- **Description:** Adds an allowance for an employee.

Add Deduction

- **Endpoint:** /deductions/add
- **Method:** POST
- **Description:** Adds a new deduction.

Delete Employee Deduction

- **Endpoint:** /employee_deductions/{deduction_id}
- **Method:** DELETE
- **Description:** Deletes a deduction for an employee.
- **Parameters:**
 - deduction_id: ID of the deduction

Add Employee Deduction

- **Endpoint:** /employee_deductions/add
- **Method:** POST
- **Description:** Adds a deduction for an employee.

Get Payrolls in Date Range

- **Endpoint:** /payrolls
- **Method:** GET
- **Description:** Retrieves payrolls within a specific date range.
- **Parameters:**
 - **start_date:** Start date in the format YYYY-MM-DD
 - **end_date:** End date in the format YYYY-MM-DD

Add Payroll

- **Endpoint:** /payrolls/add
- **Method:** POST
- **Description:** Adds a new payroll.

Delete Payroll by Date Created

- **Endpoint:** /payrolls/{date_created}
- **Method:** DELETE
- **Description:** Deletes payrolls created on a specific date.
- **Parameters:**
 - **date_created:** Date created in epoch time

Delete Payrolls in Date Range

- **Endpoint:** /payroll/date_range
- **Method:** DELETE
- **Description:** Deletes payrolls within a specific date range.
- **Parameters:**
 - **start_date:** Start date in the format YYYY-MM-DD
 - **end_date:** End date in the format YYYY-MM-DD

Add Payslip

- **Endpoint:** /payslips/add
- **Method:** POST
- **Description:** Adds a new payslip.

Add Employee Extra Hours

- **Endpoint:** `/employee_extra_hours/add`
- **Method:** POST
- **Description:** Adds extra hours worked by an employee.

Update Employee Extra Hours

- **Endpoint:** `/employees/extra_hours/{extra_hours_id}`
- **Method:** PUT
- **Description:** Updates the extra hours worked by an employee.
- **Parameters:**
 - `extra_hours_id`: ID of the extra hours record

Delete Employee Extra Hours

- **Endpoint:** `/employee_extra_hours/{extra_hours_id}`
- **Method:** DELETE
- **Description:** Deletes extra hours worked by an employee.
- **Parameters:**
 - `extra_hours_id`: ID of the extra hours record

Add User Activity Log

- **Endpoint:** `/user_activity_log/add`
- **Method:** POST
- **Description:** Adds a new user activity log.

Get User Activity Summary

- **Endpoint:** `/user_activity_summary`
- **Method:** GET
- **Description:** Retrieves a summary of user activities, including actions performed and the roles of users.

Get Employee Payroll Summary

- **Endpoint:** `/views/employee_payroll_summary`
- **Method:** GET
- **Description:** Retrieves a consolidated view of the payroll summary for each employee, including base salary, total allowances, total deductions, and net salary.

Get Department Employee Summary

- **Endpoint:** `/views/department_employee_summary`
- **Method:** GET

- **Description:** Retrieves the number of employees and the average salary for each department.

Get Department Salary Budget

- **Endpoint:** `/views/department_salary_budget`
- **Method:** GET
- **Description:** Retrieves the total salary budget for each department, including base salaries and allowances.

Get Employee Attendance Summary

- **Endpoint:** `/views/employee_attendance_summary`
- **Method:** GET
- **Description:** Retrieves the attendance summary for each employee, including total days present and absent.

Filling the Database

For filling the database, it was created a python script (Data.py), which uses the faker library to generate the dummy data. For feeding the database, first, all the possible contents that may be on the database are erased (because at the time, there was some mistake, so some tables were fed, and others not), then, the sequence is restarted, because in some tables, the sequence started for example from 50, and not from 1. Finally, all the data is filled with the multiple methods created for each table.

Tests on the Web Services

It was used Postman, which is an application capable of testing web services, so basically, it is just needed to define the method (if it is a GET method, POST, etc.), write the URL created for the specific web service and depending on what type of method it is being tested, it is needed to add or not a JSON body to interact with the method. In the case of the image, it is a GET method that is being tested, and as a result, it returned a dictionary with the information.

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8080/views/employee_attendance_summary`. The response status is 200 OK, and the body is a JSON array of employee attendance data.

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

```
1 {
2   {
3     "employee_name": "Daniel Mahoney",
4     "total_days_present": 10,
5     "total_days_absent": 26
6   },
7   {
8     "employee_name": "Alyssa Gray",
9     "total_days_present": 36,
10    "total_days_absent": 40
11  },
12  {
13    "employee_name": "Calvin Thompson",
14    "total_days_present": 29,
15    "total_days_absent": 17
16  },
17  {
18    "employee_name": "David Jones",
19    "total_days_present": 40,
20    "total_days_absent": 25
21  },
22  {
23    "employee_name": "John Doe",
24    "total_days_present": 20,
25    "total_days_absent": 0
26  },
27  {
28    "employee_name": "Sarah Turner",
29    "total_days_present": 30,
```