~~Step 1: Set Up the Environment~~

~~Install Node.js: Ensure Node.js is installed on your system.~~

~~Set Up MySQL: Install and set up MySQL server.~~

~~Initialize Node Project: Create a new folder for your project and run npm init to start a new Node.js project.~~

~~Install Required Packages: Install inquirer for command-line interactions and a MySQL Node.js client (like mysql2 or sequelize) for database interactions.~~

~~Step 2: Design Database Schema~~

~~Create SQL Schema: Define the tables for department, role, and employee as per the given schema.~~

~~Create seeds.sql: Optionally, write a script to seed the database with initial data.~~

~~Step 3: Implement Core Functionalities~~

~~Database Connection: Write a module to handle the connection to the MySQL database.~~

~~Build Command-Line Interface: Use inquirer to create the main menu and sub-menus based on the acceptance criteria.~~

~~Implement View Functions: Code the functions to view departments, roles, and employees.~~

~~Implement Add Functions: Code the functions to add a new department, role, and employee.~~

~~Implement Update Function: Code the function to update an employee's role.~~

Step 4: Implement Bonus Functionalities

Update Employee Managers: Add functionality to update employee managers.

View Employees by Manager: Implement a feature to view employees under a specific manager.

View Employees by Department: Implement a feature to view employees in a specific department.

Delete Departments, Roles, Employees: Add options to delete departments, roles, and employees.

View Department Budget: Implement a function to calculate the total salary budget of a department.

\*\*\* VIEW ALL TABLES FEATURE

Step 5: Testing

Unit Testing: Write tests for each function to ensure they work as expected.

Integration Testing: Test the application as a whole to ensure all parts work together seamlessly.

Step 6: Documentation and Final Touches

Write Documentation: Document the usage of the application and its features.

Refinement: Refine the user interface and user experience based on feedback or personal assessment.

Step 7: Submission

Prepare Submission: Ensure all code is properly commented and formatted.

Submit the Project: Follow the guidelines provided by your school for submission.

Remember, each step may require iterative development and testing. It's essential to frequently test your application to catch and fix any issues early in the development process. Good luck with your project!

BONUS FEATURES

1. Update Employee Managers

Functionality: Allow the user to update an employee's manager.

Implementation Steps:

Create a function, say updateEmployeeManager, in queries.js to update the manager ID for a specific employee in the database.

In cli.js, create a function to prompt the user to select an employee and their new manager using Inquirer, then call updateEmployeeManager with the appropriate arguments.

Ensure you have a method to list employees and managers for selection.

2. View Employees by Manager

Functionality: Allow the user to view all employees who report to a specific manager.

Implementation Steps:

Add a new query function in queries.js, for example, getEmployeesByManager, to fetch employees based on a manager ID.

Implement a prompt in cli.js to allow the user to select a manager, then display the employees who report to that manager.

3. View Employees by Department

Functionality: Enable viewing all employees in a particular department.

Implementation Steps:

Add a function, perhaps getEmployeesByDepartment, in queries.js to retrieve employees based on department ID.

Add a prompt in cli.js to select a department and use the function to display the relevant employees.

4. Delete Departments, Roles, Employees

Functionality: Provide options to delete existing departments, roles, or employees.

Implementation Steps:

Create functions in queries.js for each deletion operation: deleteDepartment, deleteRole, and deleteEmployee.

In cli.js, add prompts to select which department, role, or employee to delete, and then call the respective function.

5. View Department Budget

Functionality: Calculate the total salary budget of a department.

Implementation Steps:

Create a function in queries.js, like calculateDepartmentBudget, which sums up the salaries of all employees in a department.

Add a prompt in cli.js to select a department, then display the total budget using the calculated sum.