Create an Employee Tracking application

* This is an Employee Tracking application that allows a user to view and manage departments, roles, and the employees in a company.
* Create a command-line application that accepts user input when started
* Present the user with the following options:
  + view all departments,
  + view all roles,
  + view all employees,
  + add a department,
  + add a role,
  + add an employee,
  + update an employee role
* When the user chooses to view all departments:
  + present a formatted table showing department names and department ids
* When the user chooses to view all roles:
  + Present user with the
    - job title,
    - role id,
    - the department that role belongs to,
    - the salary for that role
* When the user chooses to view all employees:
  + Present user with a formatted table showing:
    - employee data,
    - employee ids,
    - first names,
    - last names,
    - job titles,
    - departments,
    - salaries,
    - managers that the employees report to
* When the user chooses to add a department:
  + Prompt the user to enter:
    - the name of the department and
    - that department is added to the database
* When the user chooses to add a role:
  + Prompt the user to enter:
    - the name,
    - salary,
    - department for the role
    - that role is added to the database
* When the user chooses to add an employee:
  + Prompt the user to enter the employee’s:
    - first name,
    - last name,
    - role,
    - manager,
    - that employee is added to the database
* When the user chooses to update an employee role:
  + Prompt the user to select:
    - an employee to update
    - their new role
    - update information in the database

**Grading Requirements**

This Challenge is graded based on the following criteria:

**Deliverables: 10%**

* Your GitHub repository containing your application code.

**Walkthrough Video: 27%**

* A walkthrough video that demonstrates the functionality of the employee tracker must be submitted, and a link to the video should be included in your README file.
* The walkthrough video must show all the technical acceptance criteria being met.
* The walkthrough video must demonstrate how a user would invoke the application from the command line.
* The walkthrough video must demonstrate a functional menu with the options outlined in the acceptance criteria.

**Technical Acceptance Criteria: 40%**

* Satisfies all the preceding acceptance criteria plus the following:
  + Uses the [Inquirer package (Links to an external site.)](https://www.npmjs.com/package/inquirer).
  + Uses the [MySQL2 package (Links to an external site.)](https://www.npmjs.com/package/mysql2) to connect to a MySQL database.
  + Uses the [console.table package (Links to an external site.)](https://www.npmjs.com/package/console.table" \t "_blank) to print MySQL rows to the console.
* Follows the table schema outlined in the homework instructions.

**Repository Quality: 13%**

* Repository has a unique name.
* Repository follows best practices for file structure and naming conventions.
* Repository follows best practices for class/id naming conventions, indentation, quality comments, etc.
* Repository contains multiple descriptive commit messages.
* Repository contains a high-quality README with description and a link to a walkthrough video.

**Application Quality 10%**

* The application user experience is intuitive and easy to navigate.

**Bonus**

Fulfilling any of the following can add up to 20 points to your grade. Note that the highest grade you can achieve is still 100:

* Application allows users to update employee managers (2 points).
* Application allows users to view employees by manager (2 points).
* Application allows users to view employees by department (2 points).
* Application allows users to delete departments, roles, and employees (2 points for each).
* Application allows users to view the total utilized budget of a department—in other words, the combined salaries of all employees in that department (8 points).