Labor, Adrian R.
I101
EDM

```
CREATE DATABASE companydb;

USE companydb;

CREATE DATABASE


 -- Task 1: Create a table named employees with the following fields:

CREATE TABLE employees (

   employee_ID INT AUTO_INCREMENT PRIMARY KEY, -- employee_id: Unique integer, auto-increment, primary key.

   employee_NAME VARCHAR(255) NOT NULL, -- employee_name: String (VARCHAR) with up to 255 characters, not null.

   manager_ID INT,

   FOREIGN KEY (manager_id) REFERENCES employee_tbl(employee_ID) -- manager_id: Integer, foreign key referencing employee_id in the same table (employees).

);


-- Task 2: Create a table named departments with the following fields:

CREATE TABLE departments (

   department_ID INT AUTO_INCREMENT PRIMARY KEY, -- department_id: Unique integer, auto-increment, primary key

   department_NAME VARCHAR(100) NOT NULL -- department_name: String (VARCHAR) with up to 255 characters, not null.

);


-- Task 3: Create a table named employee_departments with the following fields:

CREATE TABLE employee_departments (

   employee_ID INT, -- employee_id: Integer,

   department_ID INT, -- department_id: Integer,

   PRIMARY KEY (employee_ID, department_ID), -- Composite primary key (employee_id, department_id).

   FOREIGN KEY (employee_ID) REFERENCES employees(employee_ID), -- foreign key referencing employee_id in employees.
```

Labor, Adrian R.
I101
EDM

```
   FOREIGN KEY (department_ID) REFERENCES departments(department_ID) -- foreign key referencing department_id in departments.

);


CREATE TABLE projects (

   project_NAME VARCHAR(255) NOT NULL, -- project_name: String (VARCHAR) with up to 255 characters, not null.

   PRIMARY KEY (project_NAME) -- The PRIMARY KEY constraint ensures that the project_NAME column contains unique and non-null values, effectively identifying each record in the table uniquely

);


-- Task 4: Create a table named employee_projects with the following fields:

CREATE TABLE employee_projects (

   employee_ID INT, -- employee_id: Integer,

   FOREIGN KEY (employee_ID) REFERENCES employees(employee_ID) -- foreign key referencing employee_id in employees.

);


-- Task 5: Create a table named managers with the following fields:

CREATE TABLE managers (

   manager_ID INT AUTO_INCREMENT PRIMARY KEY, -- manager_id: Unique integer, auto-increment, primary key.

   employee_ID INT, -- employee_id: Integer

   FOREIGN KEY (employee_ID) REFERENCES employees(employee_ID) -- foreign key referencing employee_id in employees.

);
```