Employee Management

Employee table

Name	Type	Comments
id	number	unique – auto generated and incremented – primary key
fname	varchar	not null
mname	varchar	
lname	varchar	not null
manager_id	number	reference to the employee id (Hint: for top level CEO, there is no manager associated)
compensation_id	number	foreign key to Compensation table
address_id	number	foreign key to address table
degree_id	number	foreign key to degree table

Compensation table

Name	Type	Comments
id	number	primary key
doj	date	Date of join
dept_id	foreign key	foreign key to Department table. Not null
title	varchar	Title/position
salary	number	

Degree table

Name	Type	Comments
id	number	primary key
name	varchar	name of degree
year_passed	date	
institution_name	varchar	
gpa	varchar	

Address table

Name	Type	Comments
id	number	primary key
type	varchar	home, or office
line1	varchar	not null
line2	varchar	
state	varchar	not null
country	varchar	not null
zipcode	varchar	not null

- Develop a DAO class to wrap all the SQL details and for public interfacing
- Model your Employee management using POJOs and create the relations using object compositions/aggregations
- Some of the API of the DAO
    - **addEmployee(List<Employee>)**
    - **deleteEmployee(Employee or by id)**
    - **updateEmployee(Employee)**
    - **findAllEmployees()** – returns collection of Employee
    - **findEmployee(emp id)**
    - **findEmployeeByMaxSalary()** (you need to return the Employee info who has the max salary)
    - **findEmployeeBySecondMaxSalary()** (You can write SQLs or stored procs – up to you)
    - find all the managers of a same department
- write JUnit test cases for all the public API
- use collections
- take an XML file that represents the Employee hierarchy as input file and add the employees in to database using the addEmployee method of the DAO – use JAXB to create the object graph
- once the info is stored in the db, develop a program that accepts user input and keeps on running
- - Option 1 – Display details of the employee by id - id is given by the user input
- - Option 2 – delete the employee by id – id is given by user input

- - Option 3 – find max salary of the employees (no input needed, just select the option)
- - Option 4 – find second max salary (same as above)
- - Option 5 – quit the program