**CS 431 Fall 2021 Midterm**

Open book. You are encouraged to use Postgresql to try out your queries

1. (20% each) Write SQL to do the following. Statements should work regardless of the actual data in the database:

1a. Get the names of projects controlled by the Administration department

SELECT PName *-- getting project name*

FROM Project

JOIN Department ON Project.DNum = Department.DNumber *-- joining project and department*

WHERE DName = 'Administration'; *-- where the department name is 'Administration'*

1b. Get the names of managers of departments which do not have any male employees

SELECT FName, MInit, LName *-- getting FName, MInit, LName*

FROM Employee

JOIN Department ON Department.MgrSSN = Employee.SSN *-- joining employee and department*

WHERE Department.DNumber IN ( *-- using IN becasue the inner query might return more than one value*

*-- returns the number of the department where there are no male employees*

    SELECT DNumber

    FROM Department

    JOIN Employee ON Department.DNumber = Employee.DNo

    GROUP BY DNumber

    HAVING SUM(CASE WHEN Employee.Sex = 'M' THEN 1 ELSE 0 END) = 0 *-- if the sum of male employees is 0 then get the department number*

);

2. Write SQL for the Company database to (10% each part)

2a. Insert data to show that the current database model does not protect against employees supervising themselves

INSERT INTO Employee VALUES

('John', 'S', 'Doe', 875159325, '1945-08-15', '123 main st, Bellaire, TX', 'M', 45000, 875159325, 4 ); *-- inserting the same SSN for the SuperSSN*

2b. Remove the Houston location for department 5 (don’t worry about projects located in Houston)

DELETE FROM dept\_locations

WHERE dept\_locations.dlocation = 'Houston' AND dept\_locations.dnum = 5; *-- deleting the Houston location of the department 5*

2c. Add a column startDate to Employee to indicate when they started – this should not be null. You need to modify existing records to allow you to prevent null values in the column

ALTER TABLE Employee ADD startDate DATE NOT NULL DEFAULT current\_date;

*-- current date represents when the employee started*

3. You are asked to modify the entity-relationship (ER) model of the Company database to support a more structured approach to how projects are staffed. (30%)

* Each Project has a set of Positions
  + A Position has a unique id, a name (ex: for a software app there might be an architect, a user experience engineer, an artist, 2 developers, a QA engineer, and a manager), and a description (what the position’s responsibilities are)
  + Employees may serve in one or more Positions. This will replace the existing Works On relationship
  + Explain how you would modify the ER model to satisfy these new requirements. You can do this in text or by using an ER diagram (no need to duplicate existing information, and it’s fine to just have rectangles for the existing Employee and Project entities)

EER Diagram:

Diagram

Description automatically generated

Explanation:

New Entities:

* Position, Employee\_Position

Attributes:

* Position: id(key), name, description
* Employee\_Position: employee\_id, position\_id, project\_num

Relationships:

* Employee\_Position has Position: many-to-many, an employee can have many positions and positions can be given to many employees
* Employee\_Position has Project: many-to-many, an employee can have many projects and a project can have many employees
* Employee works on Employee\_Position: 1-to-many, optional for Employee to have positions and work on projects (ex: they might be a manager), required for Employee\_Position