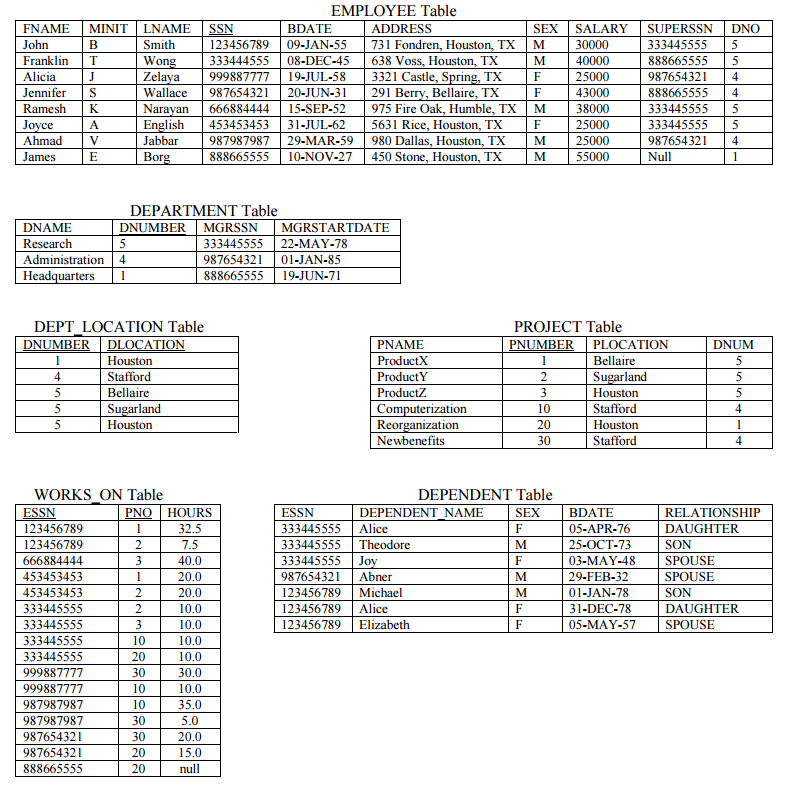
**CS 431 Fall 2021 Midterm**

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:
   1. Get the names of projects controlled by the Administration department
   2. Get the names of managers of departments which do not have any male employees
2. Write SQL for the Company database to (10% each part)
   1. Insert data to show that the current database model does not protect against employees supervising themselves
   2. Remove the Houston location for department 5 (don’t worry about projects located in Houston)
   3. 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
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)

Submitting: Put the answers to #1 and #2 in .sql files that can be executed (no need to worry about schemas), so only text, no word processor stuff. For #3, make sure your format is commonly readable – exporting to a .png or .pdf would be fine, for example

(above is snipped from: http://www.cs.umd.edu/projects/hpsl/classes/424-f98/hw3.pdf)

**Entity-relationship model**:

**Entities**: Employee, Department, Project, Dependent

**Attributes**: Employee: SSN(key), fname, minit, lname, bdate, address, sex, salary

Department: dnumber(key), dname, dlocation(multivalued)

Project: pnumber(key), pname, plocation

Dependent (weak, owned by Employee): dependent\_name(partial key), sex, bdate, relationship

**Relationships**: Employee works for Department: many-to-1, required participation for both

Employee manages Department: 1-to-1, optional for Employee, required for Department

Employee supervises Employee: 1-to-many, optional on both sides

Employee cares for Dependent: 1-to-many, optional for Employee, required for Dependent

Employee works on Project: many-to-many, required on both sides

Department controls Project: 1-to-many, optional for Department, required for Project