CSE-444 Final Exam
Autumn, 1998

I. (30 Pts.)

1. Render in the E/R model a database recording information about a car dealership, including:

   - For each vehicle, its vin (vehicle identification number), its make, model, year, and color.
   - For each combination of model and year, a base_price.
   - For each employee, their ssn (social security number), their name, their dept_id (department identification number), their dept_name, and their manager_ssn.
   - For each sale, the vin of the vehicle sold, the ssn of the salesman, the sale_price, and the sale_date.

2. Convert your E/R diagram to a relational database schema, keeping in mind the following constraints:

   Vehicles have unique vins. Any 2 vehicles that are the same model are also the same make. Any 2 vehicles with the same model and year also have the same base_price. Employees have unique ssns. Any 2 employees with the same dept_id have the same dept_name and manager_ssn. A vehicle can only be sold once.

   For each relation in your schema,

   - What are the keys of the relation?
   - What are the BCNF violations, if any? (List only FDs with one attribute on the right side.)
   - Decompose the relation, as necessary, into collections of BCNF relations.

3. Write an SQL query that considers all sales closed on December 16th, 1998. The query should output the vin of the vehicle sold and the name of the manager of the salesperson who made the sale.

4. What should be done in order to best speed up this query?

    - Store the sale relation as a sorted file on the sale_date attribute.
    - Create a clustered B+-tree index on the sale_date attribute of the sale relation.
    - Create a hashed index on the sale_date attribute of the sale relation.

    Explain your choice.

II. (20 Pts.) Suppose we have the following relations:
Salary(emp-id, amount)
Dept (emp-id, dept-id)
Name (emp-id, name)

1. Write an SQL query that obtains the pairs (name, salary) only for those employees who earn more than the average salary of their department.

2. Write an SQL query that returns the set of employee id's for those employees that earn more money than someone outside their department (i.e., return employee X if there is an employee Y in a department other than X's, who earns less than X).

III. (10 Pts.) Consider the following two schedules for interleaving the actions of transactions T1 and T2. For each schedule, decide whether it is serializable, and explain why. Note: the schedules depict the order of actions from left to right.

- T1:R(A), T2:R(A), T1:R(A), T1:W(A)
- T1:R(A), T1:W(A), T2:R(A), T1:R(A), T1:W(A)

IV. (25 Pts.) Consider the following relations:

Dept (emp-id, dept-id)

Head (dept-id, emp-id)

Salary(emp-id, amount)

The number of tuples in the relations are: Dept: 100,000, Head: 300, Salary, 100,000.

In each of the relations, a page of memory can fit 10 tuples. Every employee has a single salary, every department has at most 2 heads, and id's are unique for employees and departments.

The relation Dept is stored as a heap file with no indexes. The relation Head has a hash index on the pair (dept-id,emp-id), and the relation Salary has an index on emp-id.

Consider the following query:

```
SELECT emp-id, salary
FROM   Dept, Head, Salary
WHERE  Dept.dept-id=Head.dept-id AND
       Head.emp-id=Salary.emp-id
```

Suppose our query optimizer supports index nested loop join, hash-join and sort-merge join.

- what are the plans for two-way joins that a System-R optimizer will consider? (hint: recall that System-R does not consider Cartesian Products). For each plan, what is the estimated number of I/O's it would take (state any additional assumptions you are making to compute the number of I/Os).
- What plan will System-R choose for answering the query, and why. What is its expected I/O cost?

V. (15 Pts.) Define the following terms in 4 lines or less:

1. Clustered index
2. Predicate pushdown
3. The impedance mismatch
4. Primary key
5. ACID