**Tutorial 6**

**GC 312**

**Problem 1:**

Consider the following relational schema and SQL query. The schema captures information about employees, departments, and company ﬁnances (organized on a per department basis).

**Emp (eid: integer, did: integer, sal: integer, hobby: char(20))**

**Dept (did: integer, dname: char(20), ﬂoor: integer, phone: char(10))**

**Finance (did: integer, budget: real, sales: real, expenses: real)**

Consider the following query:

SELECT D.dname, F.budget

FROM Emp E, Dept D, Finance F

WHERE E.did=D.did AND D.did=F.did AND D.ﬂoor=1 AND E.sal ≥ 59000 AND E.hobby = ‘yodeling’

1. Identify a relational algebra expression that reﬂects the order of operations that a decent query optimizer would choose.

2. List the join orders (i.e., orders in which pairs of relations can be joined together to compute the query result) that a relational query optimizer will consider.

3. Suppose that the following additional information is available: Unclustered B+ tree indexes exist on Emp.did, Emp.sal, Dept.ﬂoor, Dept.did, and Finance.did. The system’s statistics indicate that employee salaries range from10,000 to 60,000, employees enjoy 200 diﬀerent hobbies, and the company owns two ﬂoors in the building. There are a total of 50,000 employees and 5,000 departments (each with corresponding ﬁnancial information) in the database. The DBMS used by the company has just one join method available, namely, index nested loops.

For each of the query’s base relations (Emp, Dept and Finance) estimate the number of tuples that would be initially selected from that relation if all of the non-join predicates on that relation were applied to it before any join processing begins.