Problem 1 [50 pts]

Briefly answer the following questions.

1. In the context of query optimization, what is an SQL query block?
2. Define the term reduction factor.
3. Describe a situation in which projection should precede selection in processing a project-select query and describe a situation where the opposite processing order is better. (Assume that duplicate elimination for projection is done via sorting.)
4. If there are dense, unclustered (secondary) B+ tree indexes on both R.a and S.b, the join $R \bowtie_{a=b} S$ could be processed by doing a sort-merge type of join—without doing any sorting—by using these indexes.
   (a) Would this be a good idea if R and S each have only one tuple per page, or would it be better to ignore the indexes and sort R and S? Explain.
   (b) What if R and S each have many tuples per page? Again, explain.
5. Why does the System R optimizer consider only left-deep join trees? Give an example of a plan that would not be considered because of this restriction.
6. Explain the role of interesting orders in the System R optimizer.

Problem 2 [50 pts]

Consider the following relational schema:
Suppliers (sid: integer, sname: char(20), city: char(20))
Supply(sid: integer, pid: integer)
Parts(pid: integer, pname: char(20), price: real)

Suppose we are given the following query:
SELECT S.sname, P.pname
FROM Suppliers S, Parts P, Supply Y
WHERE S.sid = Y.sid AND Y.pid = P.pid AND
   S.city = 'Madison' AND P.price \leq 1000

Briefly answer the following questions.

1. What information about these relations does the query optimizer need to select a good query execution plan for the given query?
2. What indexes might help process this query? Explain briefly about your choices.
3. How does adding DISTINCT to the SELECT clause affect the plans produced?
4. How does adding ORDER BY sname to the query affect the plans produced?
5. How does adding GROUP BY sname to the query affect the plans produced?