CSE 344 Section 9

Problem 1

Given the following query with relations R(A,B) and S(C):

```
SELECT R.A, MAX(R.B) AS MaxR_B
  FROM R, S
 WHERE R.A = S.C AND R.B > 10
 GROUP BY R.A
```

Suppose that R and S are partitioned across 3 different machines using random block partitioning, and no indexes are available on any of the machines. Draw the relational algebra plan that you would use to execute the query above. Use only shuffle for joins. You do not need to indicate how joins are executed locally on each machine.

(Start the diagram with 3 nodes at the bottom.)

Problem 2

a) Consider relations R(a,b), S(c,d), and T(e,f). All three are horizontally random block partitioned across N = 3 machines (N1, N2, N3). Show a relational algebra plan for the following query and how it will be executed across the N = 3 machines. Use hash-join (a.k.a shuffle-join) operators. Include operators that need to re-shuffle data and add a note explaining how these operators will re-shuffle that data.

```
SELECT *
  FROM R, S, T
 WHERE R.b = S.c
       AND S.d = T.e
       AND (R.a - T.f) > 100
```

b) Now consider the case where the R relation is very large and both S and T are very small. Show a plan that uses broadcast joins to compute the result of the query.

Problem 3

Say you are designing a parallel relational database to store purchase data for manufacturing products. The tables are:

```
Manufacturer(mid, name, category, city, state)
Purchase(pid, mid, date, amount) -- foreign key mid to Manufacturer
```

Tuples in the purchase table record individual payments in dollar amounts to a manufacturer for purchases of some product. There is a large amount of data in both tables that would have to be spread between multiple machines. As the database designer, you know that the most common query that will be run on the system is:

```
SELECT m.mid, SUM(p.amount) AS total_revenue
  FROM Purchase p, Manufacturer m
 WHERE p.mid = m.mid
 GROUP BY m.mid
```

a) Describe in a few sentences how you would partition the data between machines if your goal is to maximize performance of the above query.

b) Now consider that instead of maximizing performance of any query, your goal is to minimize skew and store the data evenly across the machines. Describe in a few sentences how you would partition the data in that case.

Problem 4

Imagine that you are designing a parallel RDMS to digitize all the letters received by the post office. We assume that "letters" may only contain paper (ie, text) or DVDs (ie, binary blobs).

LetterID	**SenderAddr**	**RecipientAddr**	**Status**	**ContentType**	**Content**
12345	1600 Pennsylvania Ave	185 E Stevens Way	Delivered	Text	"Dear Hannah, I would like to request a regrade of the midterm …"
67890	3800 E Stevens Way	185 E Stevens Way	InTransit	DVD	*(lots and lots of 0s and 1s)*

Consider the partitioning strategies we know:

- Block (Horizontal)
- Range (Horizontal) - please specify attribute set
- Hash (Horizontal) - please specify attribute set
- Vertical - please specify attribute set

Which one would you choose under the following circumstances? If you choose Range, Hash, or Vertical partitioning, please specify the attributes that you would partition on.

a) The query load consisted solely of counts of in-transit letters (ie, "SELECT COUNT(*) WHERE Status='InTransit'").

b) All of the attributes have uniformly distributed data except for recipients (eg, the President of the United States gets an unusually large or unusually small amount of letters).

c) The query load consists primarily but not solely of randomly-sampled letter contents, grouped by sender (eg, "the President of the United States typically sends letters that start with "Dear Sir or Madam,…"). The database contains every letter sent since the creation of the Postal Service in 1771.