Student Id: 1605029

Answer to 12.1

a. Each of the queries $Q_1$ and $Q_2$ will be processed in single node since partitioning attribute and query attribute are same here.

b. It is highly likely that there are more tuples in "Dhaka" node compared to "Sherpur" node, and hence, there is a possibility for data skewing. As a result, Q1 will take more time compared to $Q_2$. To improve the processing time for $Q_1$, we can take the following measures

i. Repartition the relation based on other attribute

ii. Partition tuples virtually (say, tuples in each node can be virtually partitioned into 64 virtual nodes. Then, we can process the query parallely in those 64 nodes.)

But it will be tough to improve the processing time if skewness is significantly high.

Answer to 12.2

Here, partitioning attribute and range query attribute are same. Hence, query will be processed parallely at the nodes whose partition overlaps with the specified range of values ($N_{10}$ - $N_{29}$ in this case).

Answer to 12.3

Query ~~at~~ $Q_1$ will be processed in parallel since no query attribute is provided. The reason for $Q_2$ is ~~partition~~ partitioning attribute and range query attribute not being the same.

Answer to 12.4

a. The aggregate query will be processed in the following manner
   i. Partitioning attribute "district" and grouping attribute "income" are different. So person relation will be repartitioned based on the "income" attribute.
   ii. Aggregate value "count(NID)" will be computed locally at each node.

b. 80% of the total tuples are shifted and the rest of the tuples remain at the same node after repartitioning.
   ∴ Repartitioning cost = 160M × 0.8 = 128M tuples

c. We can reduce the cost of transferring tuples by using partial aggregation. In this type of aggregation, each node will contain locally aggregated tuples.
   Since we have 640 distinct income values, after

performing partial aggegation, each node will contain 640 ~~aggrut aggr~~ aggregated tuples. Repartitioning these partially aggregated tuples based on grouping attribute "income" will result in each node having exactly 640/64 = 10 merged tuples with distinct income values.

Optimized repartition cost: 630 out of 640 tuples will leave their node after repartitioning. So,

Repartitioning Cost = Number of tuples displaced

= 630 × 64

= 40, 320

This ~~number~~ cost is significantly smaller than the previous cost (128M). Thus, optimization is achieved.

Answer to 13-1

a. The steps are:

i. person' = exchangeOperator(person, "range-p", 64, "income")

ii. ~~Tut~~ Tuples at each node will be ~~repartitioned~~ sorted locally based on "income".

b. The steps are:

i. Tuples at each node will be sorted locally based on "income".

ii. person relation will be repartitioned and merged based on "income" by exchange operator.