Project 2 : Data Models and Query Languages (CSE 560)

Shubham Sharma, Person No.: 50290293, UBIT: ss628

April 24, 2019

Problem 1

The query S1 has an equivalent relational algebra formulation as follows:

$$Answer := \pi_{Ename}(Employees) - (\pi_{Ename}(Assign) - \pi_{Ename}(Employees))$$

In this relational algebra formula, projection of attribute Ename is taken from Employees relation and projection of Ename is taken from Assign relation. Thereafter, the former is taken out from the latter using difference operation. The result is taken out from the projection of attribute Ename is from the Employees relation to give us the employees assigned to exactly one project.

The query S2 has an equivalent relational algebra formulation as follows:

$$Mark\_Sal := \pi_{Salary}(\sigma_{Ename='Mark'}(Employee))$$

$$Answer := \pi_{Employee.Ename}(\sigma_{Employee.Salary>Mark.Salary}(Employee \times Mark\_Sal))$$

In this relational algebra formula, first all tuples are selected from relation Employee where the Ename is 'Mark'. From this projection of the attribute Salary is taken to give us Mark's salary as $Mark\_Sal$. In the second expression, first a cross product is taken of the relation Employee and $Mark\_Sal$, which adds Mark's salary to each tuple. Thereafter a selection is taken on the condition for whichever employee's salary is greater than Mark's salary and a projection is taken to get those employee's name.

The query S3 does not have an equivalent relational algebra formulation since here we require an aggregator function COUNT to compute the number of projects whose budget is higher as well as GROUP BY function for which the relational algebra formulation is not discussed in class.

The query S4 does not have an equivalent relational algebra formulation since here we require an aggregator function AVG to compute the average budget for an agency as well as GROUP BY function for which the relational algebra formulation is not discussed in class.

Problem 2

The figure 1 shows the graph for which the schema is defined using the relations $Nodetable$, which depicts the nodes in the graph and $Network\_DAG$, which depicts the edges between different nodes with $Node\_init$ as the initial node and $Node\_fin$ as the final node, as shown in figure 2 and 3. The results for query 1 is shown in figure 4. As we can compare with the graph, 2 nodes: B and C are directly reachable from A which is depicted in this result (and similarly for other nodes as well).

For query 2, we see in figure 5, for node A: 3 nodes are reachable, directly or indirectly: A, B and C.

For query 3, we see in figure 6, that the set of all the nodes not directly reachable from the node A are A, D and E.

For query 4, we see in figure 7, that the set of all the nodes not reachable, directly or indirectly, from the node A are D and E.

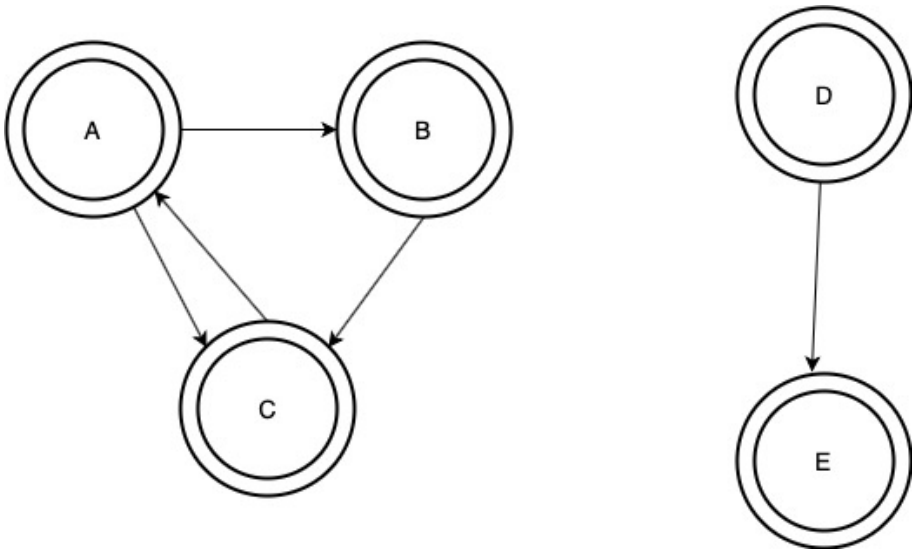

Figure 1: Graph taken as Schema

Nodename
A
B
C
D
E

Figure 2: Relation *Nodetable*

Node_init	Node_Fin
A	B
B	C
C	A
A	C
D	E

Figure 3: Relation *Network_DAG*

Nodename	NUM_OF_NODES_DIRECTLY_REACHABLE
A	2
B	1
C	1
D	1
E	0

Figure 4: Number of the nodes directly reachable from each node

Nodename	DIRECTLY_INDIRECTLY_REACHABLE
A	3
B	3
C	3
D	1
E	0

Figure 5: Number of the nodes reachable, directly or indirectly, from each node

Nodename
A
D
E
NULL

Figure 6: Set of all the nodes not directly reachable from the node A

Nodename
D
E

Figure 7: Set of all the nodes not reachable, directly or indirectly, from the node A