Group 11

Name: Woonki Kim

Email: kimwoon@oregonstate.edu

ONID: kimwoon

Name: Hyuntaek Oh

Email: ohhyun@oregonstate.edu

ONID ohhyun

2: Query Optimization

	Output Size	Cost	Plan
$R \bowtie_{B,C} S$	200	8.75k	$R \bowtie_{B,C} S$
$R \bowtie_B W$	40k	7.5k	$R \bowtie_B W$
$R \bowtie_A U$	40k	6.25k	$R \bowtie_A U$
$S \bowtie_B W$	60k	6.25k	$S \bowtie_B W$
$W \bowtie_D U$	20k	3.75k	$W \bowtie_D U$

< **Table 1.** Two Relation Natural Join >

	Output Size	Cost	Plan
$(R \bowtie S) \bowtie_B W$	2k	11.5k	$(R \bowtie S) \bowtie_B W$
$(R \bowtie S) \bowtie_A U$	2k	10.25k	$(R \bowtie S) \bowtie U$
$(R \bowtie W) \bowtie_{B,C} S$	4k	61.25k	$(R \bowtie W) \bowtie S$
$(R \bowtie W) \bowtie_{A,D} U$	4k	58.75k	$(R \bowtie W) \bowtie U$
$(R \bowtie U) \bowtie_{B,D} W$	4k	58.75k	$(R \bowtie U) \bowtie W$
$(R \bowtie U) \bowtie_{B,C} S$	2k	60k	$(R \bowtie U) \bowtie S$

$(S \bowtie W) \bowtie_D U$	600k	82.5k	$(S \bowtie W) \bowtie U$
$(S \bowtie W) \bowtie_{B,C} R$	4k	86.25k	$(S \bowtie W) \bowtie R$
$(W \bowtie U) \bowtie_{A,B} R$	4k	33.75k	$(W \bowtie U) \bowtie_{A,B} R$
$(W \bowtie U) \bowtie_B S$	600k	32.5k	$(W \bowtie U) \bowtie_B S$

< **Table 2.** Three Relation Natural Join >

	Output Size	Cost	Plan
$((R \bowtie S) \bowtie W) \bowtie U$	x	15.25k	$((R \bowtie S) \bowtie W) \bowtie U$
$((R \bowtie S) \bowtie U) \bowtie W$	x	15.25k	$((R \bowtie S) \bowtie U) \bowtie W$
$((R \bowtie W) \bowtie S) \bowtie U$	x	67.5k	
$((R \bowtie W) \bowtie U) \bowtie S$	x	67.5k	
$((R \bowtie U) \bowtie W) \bowtie S$	x	67.5k	
$((R \bowtie U) \bowtie S) \bowtie W$	x	65k	
$((S \bowtie W) \bowtie U) \bowtie R$	x	837.5k	
$((S \bowtie W) \bowtie R) \bowtie U$	x	92.5k	
$((W \bowtie U) \bowtie R) \bowtie S$	x	42.5k	
$((W \bowtie U) \bowtie S) \bowtie R$	x	787.5k	

< **Table 3.** Four Relation Join >

Thus, the most efficient join order according to the dynamic programming algorithm used in System-R is both $((R \bowtie S) \bowtie W) \bowtie U$ and $((R \bowtie S) \bowtie U) \bowtie W$, which costs 15.25k.

3: Query Containment

Consider the set of conjunctive queries with comparison operators < and >. Prove that the homomorphism theorem for conjunctive queries proved in the class extends to these queries or provide an example that shows it does not hold for them.

To prove the homomorphism theorem for conjunctive queries proved in the class extends to these queries, we consider if $q_1 \subseteq q_2$, then there is homomorphism from q_2 to q_1.

For example, $q_1(x) : -R(x, y), R(y, z), R(z, w).$

$$q_2(x) : -R(x, y), R(y, z).$$

Consider, $q_1(x) : -R(x, y), R(y, z), R(z, w), y < l,\ w > n$

$$q_2(x) : -R(x, y), R(y, z), z < m$$

Then, we cannot guarantee that $z \, in \, q_1$ is in $z \, in \, q_2$.

Let says we want to prove if $q_1 \subseteq q_2$,

$$Q_1(eid) : -Emp(eid, ename, age), Works(eid, did), Dept(did, dname, budget), age < 40, \quad budget > 30{,}000$$

$$Q_2(eid) : -Emp(eid, ename, age), Works(eid, did), age > 50$$

In the case above, if there are no comparison operators, it will hold the homomorphism. However, due to the range of age, it is obvious that Q_2 cannot contain Q_1. Since Q_1 has the range of age under 40 whereas Q_2 has the range of age over 50, which means there are no common tuples in this condition.

Therefore, there is no possibility to hold the homomorphism.