Solution Set 4

Solution Set provided courtesy of Shankar Ponnekanti and Sriram Raghavan

1 Problem 1.

(a) The SELECT keyword in SQL actually does the work of the projection operator ($\pi$) in relational algebra. The SELECT operator in relational algebra selects tuples from a relation based on a given condition whereas the SELECT keyword in SQL projects out a specified set of attributes from a given relation.

(b) The given relational algebra expression can be converted to the following SQL statement:

SELECT attribute-list
FROM R, S
WHERE R.B = S.B AND condition

2 Problem 2.

The converted relational algebra expressions are as follows:

(a) $\sigma_Q \text{ and } M(R) \bowtie \sigma_P \text{ and } M(S)$

(b) $(\sigma_Q(R) \bowtie S) \cup (\sigma_P(S) \bowtie R)$

(c) $\pi_E(\pi_D(\pi_C(\sigma_Q \text{ and } M(R)) \bowtie \sigma_P \text{ and } M(S)) \bowtie \pi_{D,E}(T))$

3 Problem 3

Plan 1. 1500 random I/O’s will be needed to read all the blocks of $r$. We need to determine for each tuple of $r$, the number of tuples of $s$ such that $r.B = s.B$. This is given by $n_s/V(B, S)$ provided we assume that all values $r.B$ also occur in $s$. This is the number of $s$ tuples that need to be retrieved for each $r$ tuple. Since for relation $s$, the index on $B$ is not clustered, this would require $n_s/V(B, S)$ I/O’s for each $r$ tuple. Doing the check on the $C$ attribute does not require further I/O’s. Hence
No. of I/Os = \( n_1 = 1500 + \frac{n_r n_s}{V(B, S)} \)
\[ = 1500 + \frac{750000 \times 250000}{50000} \]
\[ = 1500 + 3750000 \]
\[ = 3751500 \]

All these I/O’s are random I/O’s. We assume that time for random I/O is \( t_r \) and time for sequential I/O is \( t_s \). Then time needed for Plan 1 is \( n_1 t_r = 3751500 t_r \).

**Plan 2.** Again 1500 random I/O’s will be needed to read all the blocks of \( r \). The number of \( s \) tuples retrieved for each \( r \) tuple is \( n_s / V(C, S) \), assuming again that all values of \( C \) attribute in \( r \) are also present in \( s \). Since the index on the \( C \) attribute is clustered, these tuples will be present on approximately \( \frac{n_s}{V(C, S) \times 100} \) blocks. But all these I/O’s (except the first I/O) are sequential. So for every tuple of \( r \), there are \( \frac{n_s}{V(C, S) \times 100} - 1 \) sequential I/O’s and 1 random I/O (for the first block). Hence

\[
\text{Total no. of I/Os} = 1500 + \frac{n_r n_s}{V(C, S) \times 100} \]
\[ = 1500 + \frac{750000 \times 250000}{50 \times 100} \]
\[ = 1500 + 3750000 \]
\[ = 3751500 \]

\[
\text{No of sequential I/O's} = n_r (\frac{n_s}{V(C, S) \times 100} - 1) \]
\[ = 750000 \times 250000 \]
\[ = 36750000 \]
\[ = 10n_1 \text{ (approximately)} \]

\[
\text{No of random I/O's} = 1500 + n_r \]
\[ = 751500 \]
\[ = 0.2n_1 \text{ (approximately)} \]

Hence we have

\[
\text{Time for Plan 2} = 10n_1 t_s + 0.2n_1 t_r \]
\[ = n_1 t_r (10 \frac{t_s}{t_r} + 0.2) \]

If \( 10 \frac{t_s}{t_r} + 0.2 < 1 \), i.e., if \( \frac{t_s}{t_r} > 12.5 \), then we expect Plan 2 to do better. Otherwise, Plan 1 does better.

**\( V(C, S) = 500 \):** In this case, cost of Plan 1 remains the same. For Plan 2:

\[
\text{Total no. of I/Os} = 1500 + \frac{n_r n_s}{V(C, S) \times 100} \]
\[ \text{No of sequential I/O's} = n_r \left( \frac{n_s}{V(C, S) * 100} - 1 \right) \]

\[ = 750000 \left( \frac{250000}{50000} - 1 \right) \]

\[ = 300000 \]

\[ = 0.8n_1 \text{ (approximately)} \]

\[ \text{No of random I/O's} = 1500 + n_r \]

\[ = 751500 \]

\[ = 0.2n_1 \text{ (approximately)} \]

Time for Plan 2 is given by \( n_1 t_r (0.8\frac{n_s}{100} + 0.2) \) which is less than \( n_1 t_r \) since \( \frac{t_r}{t_r} < 1 \). Hence Plan 2 definitely does better.

**Assuming No of tuples returned depends on the domain size:** For Plan 1:

\[ \text{No. of I/Os} = n_1 = 1500 + \frac{n_r n_s}{DOM(B, S)} \]

\[ = 1500 + \frac{750000 \cdot 250000}{1000} \]

\[ = 187501500 \]

All these I/O’s are random I/O’s. Time needed for Plan 1 is \( n_1 t_r = 187501500 t_r \).

For Plan 2:

\[ \text{Total no. of I/Os} = 1500 + \frac{n_r n_s}{DOM(C, S) * 100} \]

\[ = 1500 + \frac{750000 \cdot 250000}{100 \cdot 100} \]

\[ = 1500 + 18750000 \]

\[ = 18751500 \]

\[ \text{No of sequential I/O’s} = n_r \left( \frac{n_s}{DOM(C, S) * 100} - 1 \right) \]

\[ = 750000 \left( \frac{250000}{10000} - 1 \right) \]

\[ = 1800000 \]

\[ = 0.1n_1 \text{ (approximately)} \]

\[ \text{No of random I/O’s} = 1500 + n_r \]

\[ = 751500 \]

\[ = 0.004n_1 \text{ (approximately)} \]

So time for Plan 2 is \( n_1 t_r (0.1\frac{n_r}{t_r} + 0.004) \). Obviously, Plan 2 does better in this case.