Homework 5

Hanlin Sun

November 2022

1 Question 1

1.1 Question-1A

$\Pi_{P.Address,P.Host}(\sigma_{P.PId=S.PId\ AND\ S.GId=G.GId\ AND\ G.name='Eileen'}$
$_{AND\ P.Type='house'\ AND\ P.MaxGuest\ >2\ AND\ P.MaxGuest\ <5})((P \times S) \times G)$

1.2 Question-1B

$\Pi_{P.Address,P.Host}(((\sigma_{P.Type='house'\ AND\ P.MaxGuest\ >2}$
$_{AND\ P.MaxGuest\ <5}\ P) \bowtie_{P.PId=S.PId}\ S) \bowtie_{S.GId=G.GId}\ (\sigma_{G.Name='Eileen'}G))$

1.3 Question-1C

$\Pi_{P.Address,P.Host}(\Pi_{Address,Host,PId}(\sigma_{P.Type='house'\ AND\ P.MaxGuest\ >2}$
$_{AND\ P.MaxGuest\ <5}\ P) \bowtie_{P.PId=S.PId}\ (\Pi_{PId,GId}\ S \bowtie_{S.GId=G.GId}\ (\Pi_{GId}\ (\sigma_{G.Name='Eileen'}G)))$

2 Question 2

2.1 Question-2A

Select Equality Estimation:
$Sel_{A=c} = \frac{1}{V(R,A)}$, where $V(R, A)$ is the number of distinct value in R.
In this case, we have distinct value of Guest's name $V(R, A)$ for 10,000, $T(R) = 50,000$.
so the estimate tuple number for $\sigma_{Name='Eileen'}Guest$ is:

$$\frac{T(R)}{V(R,A)} = \mathbf{5}.$$

2.2 Question-2B

Select And Estimation:
$Sel_{A=c \wedge A=b} = Sel_{A=c} \times Sel_{A=b}$.
For property, $T(R) = 4000$, $V(R, A)$ is 10, so $\frac{1}{V(R,A)}$ is 0.1 and $Sel_{MaxGuest>2 \wedge MaxGuest<5}$ is

$0.5 \times 80\% = 40\%$.

So, the final tuple number is:

$$4000 \times 10\% \times 40\% = \mathbf{160}$$

2.3 Question-2C

Set $f(A)$ to be the answer of 2-A.

Every Guest and every Property have participated in **some** Stay(one to many), and Stays are evenly distributed over Properties and Guests, so after join GId, Stay should have:

$$100000 \times \frac{f(A)}{50000} = 20f(A)$$

And $T(\pi_{PId}(Stay \bowtie f(A)))$=$T(Stay \bowtie f(A))$(No tuple eliminated),so the tuple number is $T(Stay \bowtie f(A))$.

After join, should have (2+4-1) attributes stored in the temp block. So the final block number is

$$\lceil (2+4-1) \times \frac{20f(A)}{400} \rceil = 0.25f(A)=\mathbf{2}$$

2.4 Question-2D

Set $f(B)$ to be the answer of 2-B.

Every Guest and every Property have participated in **some** Stay(one to many), and Stays are evenly distributed over Properties and Guests, so after join PId with property, Stay should have:

$$100000 \times \frac{f(A)}{4000} = 250f(A)$$

And $T(\pi_{Address,Host}(\pi_{PId,Address,Host}\ f(B) \bowtie_{PId} (\sigma_{PId}Stay))) = T(f(B) \bowtie (\sigma_{PId}Stay))$

so, the tuple number is $250f(B)$. And after join, due to maximum projection $\sigma_{PId,Address,Host}$, block need to store 3 attributes. So the block number is:

$$\frac{3 \times 250f(B)}{400} = 1.875f(B) = \mathbf{300}$$

3 Question 3

3.1 Question 3A

Since block nested loop required, the joint cost is:

$$\text{Cost: } M + \lceil (M/(B-2)) \rceil \times N$$

use smaller relation as outer. Stay has 10,000 blocks while Guest have 250 blocks, so Guest is the outer.(Buffer number is 22) So, the total cost will be:

$$250 + \lceil \tfrac{250}{22-2} \rceil \times 10,000 = 130,250$$

3.2 Question 3B

Since block nested loop required, the joint cost is:

$$\text{Cost: } M + \lceil (M/(B-2)) \rceil \times N$$

use smaller relation as outer. Stay has 10,000 blocks while Property have 50 blocks, so Property is the outer.(Buffer number is 22) So, the total cost will be:

$$50 + \lceil \tfrac{50}{22-2} \rceil \times 10,000 = 30,050$$

3.3 Question 3C

Since Stay and Guest are already ordered by GId but with no index, so using merge joint is the most efficient way.

For $\sigma_{Name='Eileen'Guest'}$, after selection has only 5 tuples, which is 1 block.

Then do the merge joint between Stay and $\sigma_{Name='Eileen'Guest'}$, since both of these sets were ordered, so no need to sort them. The sort cost is 0.

So the final cost will be:

$$Sort(Guest) + Sort(Stay) + (10000 + 1) = 10,001$$

3.4 Question 3D

No, the cost will not changed. Since Stay and Guest has already been ordered, they don't need to satisfied the sorting memory requirements. Only need 3 buffer.

3.5 Question 3E

In this case Stay was not ordered, so need to use external sort technique.

Number of Passes: $1 + \lceil log_{B-1}(\lceil N/B \rceil) \rceil = 1 + \lceil log_{22-1}(\lceil 10,000/22 \rceil) \rceil = 1 + 3 = 4$

Total cost: $2N\times$ (number of passes) $= 2 \times 10000 \times 4 = 80,000$ I/Os

Pass 0: $10000/22 = 455$ sorted runs of 22 pages each(last run only have 12 pages)

Pass 1: $455/21 = 22$ sorted runs of 462 pages each(last run only have 298 pages)

Pass 2: $22/21 = 2$ sorted runs of 9702 pages each(last run only have 298 page)

Pass 3: Sorting 10,000 page complete.

3.6 Question 3F

In part E, we need to sort the Stay table, so changing the buffer will certain affect the cost.

When the buffer size is 10, the total pass becomes:

$$1 + \lceil log_{10-1}(\lceil 10,000/10 \rceil) \rceil = 1 + 4 = 5$$

So the total cost becomes:

$$2N \times \text{ (number of passes)} = 2 \times 10000 \times 5 = 100,000$$

4 Question 4

4.1 Question 4A

These two queries is equivalent.

For query 1, it return the property's Address and Host where having property's type equals house and MaxGuest number between 2 and 5 in Property join Stay on p.PId=s.PId and Stay join Guest on s.GId = g.GId.

For query 2, it return the property's Address and Host where having property's type equals house and MaxGuest number between 2 and 5 in Property.

Overall. the return value was the same because every Guest and every Property have participated in some Stay, which means Property join Stay join Guest tuple number equals to Property's tuple number.

Should use Query 2, query 2 is more efficient because it returns the same value with query 1 but no need to do the join process.

4.2 Question 4B

Query 2 is more efficient.

For query 1, it need to join the Property and Guest and Stay together, so need more time to finish this process.

But for query 2, there is no need to join the table so can skip this process.

4.3 Question 4C

If Stay join Guest on s.GId=g.GId have many duplicated tuples, then in query 4 the distinct operation will bring more cost, in this case Q3 may be more efficient.

But if less or no duplicate tuples were created, Q4 will be more efficient.

4.4 Question 4D

Q5 is better, for exist operation, it will remove the duplicate tuples and optimize the storage, store less data than Q3.

4.5 Question 4E

Way 1: replace 'select *' with more specific range to reduce extra space.

After optimize, the query becomes:

With HouseID AS

(SELECT PId FROM Property WHERE p.type='house')

OctStay As

(SELECT Date_of_Stay FROM Stay WHERE Date_of_Stay='10/2/2021'),

SELECT g.GId, g.Name FROM Guest g JOIN Stay s ON g.GId=s.GId WHERE s.PId IN HouseID AND s.Date_of_Stay IN OctStay

Way 2: Join these table together instead of using With...As.

SELECT g.GId, g.Name

FROM Guest g JOIN Stay s ON g.GId=s.GId JOIN Property p ON S.PId =p.PId

WHERE p.type='house' AND s.Date_of_Stay = '10/2/2021'