Optimization of Logical Queries

Integrated Exercise 8.3 p3

Translate the following SQL query to the relational algebra, remove redundant joins and use the algebraic laws to produce a better query plan.

```
SELECT F.budget, E.eid
FROM Emp E, Dept D, Finance F
WHERE E.did = D.did AND D.did = F.did
  AND E.hobby = 'yodeling'
  AND D.floor NOT IN
  ( SELECT D2.floor FROM Dept D2, Finance F2
    WHERE NOT D2.dname = 'CID'
      OR  (F2.did = D2.did AND F2.expenses >= ALL
             (SELECT MAX(F3.expenses)
              FROM Finance F3
              WHERE F3.budget = F.budget)
          )
  )
```

Optimization of Logical Queries

Step 1. Normalize to Exists and Not Exists

```
SELECT F.budget, E.eid
FROM Emp E, Dept D, Finance F
WHERE E.did = D.did AND D.did = F.did AND E.hobby = 'yodeling'
  AND NOT EXISTS
  ( SELECT D2.floor FROM Dept D2, Finance F2
    WHERE D.floor = D2.floor AND
      ( NOT D2.dname = 'CID'
        OR  (F2.did = D2.did AND NOT EXISTS
                (SELECT MAX(F3.expenses)
                 FROM Finance F3
                 WHERE F3.budget = F.budget
                 HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
             )
      )
  )
```

Optimization of Logical Queries

Step 2. Normalize to the conjunctive normal form

```
SELECT F.budget, E.eid
FROM Emp E, Dept D, Finance F
WHERE E.did = D.did AND D.did = F.did AND E.hobby = 'yodeling'
  AND NOT EXISTS
  ( SELECT D2.floor FROM Dept D2, Finance F2
    WHERE (D.floor = D2.floor AND NOT D2.dname = 'CID')
       OR (D.floor = D2.floor AND F2.did = D2.did
           AND NOT EXISTS
                (SELECT MAX(F3.expenses)
                 FROM Finance F3
                 WHERE F3.budget = F.budget
                 HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
           )
  )
```

Optimization of Logical Queries

Step 3. Normalize to UNION

```
SELECT F.budget, E.eid
FROM Emp E, Dept D, Finance F
WHERE E.did = D.did AND D.did = F.did AND E.hobby = 'yodeling'
  AND NOT EXISTS
  ((SELECT D2.floor FROM Dept D2, Finance F2
    WHERE D.floor = D2.floor AND NOT D2.dname = 'CID')
   UNION
    (SELECT D2.floor FROM Dept D2, Finance F2
     WHERE D.floor = D2.floor AND F2.did = D2.did
           AND NOT EXISTS
                 (SELECT MAX(F3.expenses)
                  FROM Finance F3
                  WHERE F3.budget = F.budget
                  HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
    )
  )
```

Optimization of Logical Queries

Step 4. Translation of the innermost subquery

```
SELECT MAX(F3.expenses)
FROM Finance F3
WHERE F3.budget = F.budget
HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
```

$$e_1 := \boldsymbol{\pi}_{\texttt{MAX(F3.expenses),F.*,F2.*}}\, \boldsymbol{\sigma}_{\texttt{MAX(F3.expenses)>F2.expenses}}\, \boldsymbol{\gamma}_{\texttt{MAX(F3.expenses),F.*,F2.*}}$$
$$\boldsymbol{\sigma}_{\texttt{F3.budget=F.budget}}\, (\boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F3}}(\texttt{Finance}))$$

Optimization of Logical Queries

Step 5. Translation

```
(SELECT D2.floor FROM Dept D2, Finance F2
 WHERE D.floor = D2.floor AND F2.did = D2.did
       AND NOT EXISTS
            (SELECT MAX(F3.expenses)
             FROM Finance F3
             WHERE F3.budget = F.budget
             HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
)
```

Translating the From clause yields:

$$e_2 := \boldsymbol{\rho}_{\mathtt{D2}}(\mathtt{Dept}) \times \boldsymbol{\rho}_{\mathtt{F2}}(\mathtt{Finance}) \times \boldsymbol{\rho}_{\mathtt{D}}(\mathtt{Dept}) \times \boldsymbol{\rho}_{\mathtt{F}}(\mathtt{Finance})$$

Since this is a NOT EXISTS subquery, we decorrelate by means of an antijoin:

$$f_2 := (e_2 \,\overline{\bowtie}\, \pi_{\mathtt{F2.*,F.*}}(e_1))$$

Adding the WHERE and SELECT clauses (with the necessary parameters) gives:

$$e_3 := \boldsymbol{\pi}_{\mathtt{D2.floor,D.*,F.*}} \boldsymbol{\sigma}_{\mathtt{D.floor=D2.floor} \wedge \mathtt{F2.did=D2.did}}(f_2)$$

Optimization of Logical Queries

Step 6. Translation

```
((SELECT D2.floor FROM Dept D2, Finance F2
  WHERE D.floor = D2.floor AND NOT D2.dname = 'CID')
```

$$e_4 := \boldsymbol{\pi}_{\texttt{D2.floor,D.*}} \boldsymbol{\sigma}_{\texttt{D.floor=D2.floor} \wedge \texttt{D2.name} \neq \texttt{'CID'}} (\boldsymbol{\rho}_{\texttt{D2}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}))$$

Optimization of Logical Queries

Step 7. Translation of the union

```
((SELECT D2.floor FROM Dept D2, Finance F2
  WHERE D.floor = D2.floor AND NOT D2.dname = 'CID')
 UNION
  (SELECT D2.floor FROM Dept D2, Finance F2
   WHERE D.floor = D2.floor AND F2.did = D2.did
         AND NOT EXISTS
               (SELECT MAX(F3.expenses)
                FROM Finance F3
                WHERE F3.budget = F.budget
                HAVING MAX(F3.EXPENSES) > F2.EXPENSES)
  )
)
```

Notice that the schemas of e_3 and e_4 are not equivalent because they have other context relations. Therefore, to be able to take the union, we have to add the context relation F of e_3 to e_4.

$$e_5 := \pi_{\texttt{D2.floor,D.*,F.*}}(e_3) \cup \pi_{\texttt{D2.floor,D.*,F.*}}(e_4 \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}))$$

Optimization of Logical Queries

Step 8. Translation of the outermost query

Translation of the From clause yields:

$$e_6 := \boldsymbol{\rho}_{\mathtt{E}}(\mathtt{Emp}) \times \boldsymbol{\rho}_{\mathtt{D}}(\mathtt{Dept}) \times \boldsymbol{\rho}_{\mathtt{F}}(\mathtt{Finance})$$

Since this is a NOT EXISTS subquery, we decorrelate by means of an antijoin:

$$f_7 := e_6 \,\overline{\bowtie}\, \boldsymbol{\pi}_{\mathtt{D.*,F.*}}(e_5)$$

And translate the remaining WHERE and SELECT clauses:

$$e_7 := \boldsymbol{\pi}_{\mathtt{F.budget,E.eid}}(\boldsymbol{\sigma}_{\mathtt{E.did=D.did} \wedge \mathtt{D.did=F.did} \wedge \mathtt{E.hobby='yodeling'}}(f_7))$$

Whole expression:

$$\boldsymbol{\pi}_{\texttt{F.budget,E.eid}}($$
$$\boldsymbol{\sigma}_{\texttt{E.did=D.did}\wedge\texttt{D.did=F.did}\wedge\texttt{E.hobby='yodeling}}(\boldsymbol{\rho}_{\texttt{E}}(\texttt{Emp}) \times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance})))$$
$$\overline{\bowtie} \ (\boldsymbol{\pi}_{\texttt{D.*,F.*}} \boldsymbol{\sigma}_{\texttt{D.floor=D2.floor}\wedge\texttt{D2.name}\neq\texttt{'CID'}}(\boldsymbol{\rho}_{\texttt{D2}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance})$$
$$\times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}))$$
$$\cup \ \boldsymbol{\pi}_{\texttt{D.*F.*}} \ ((\boldsymbol{\sigma}_{\texttt{D.floor=D2.floor}\wedge\texttt{F2.did=D2.did}}(\boldsymbol{\rho}_{\texttt{D2}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance})$$
$$\times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance})))$$
$$\overline{\bowtie} \ (\pi_{\texttt{F2.*,F.*}} \boldsymbol{\sigma}_{\texttt{MAX(F3.expenses)>F2.expenses}} \boldsymbol{\gamma}_{\texttt{MAX(F3.expenses),F.*,F2.*}}$$
$$\boldsymbol{\sigma}_{\texttt{F3.budget=F.budget}} \ (\boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F3}}(\texttt{Finance}))))$$

Optimization of Logical Queries

Step 9. Removal of redundant joins

No atom can be removed from the following maximal subexpressions (why?)

- $\boldsymbol{\sigma}_{\texttt{E.did=D.did}\wedge\texttt{D.did=F.did}\wedge\texttt{E.hobby='yodeling}}(\boldsymbol{\rho}_{\texttt{E}}(\texttt{Emp}) \times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}))$
- $(\boldsymbol{\sigma}_{\texttt{D.floor=D2.floor}\wedge\texttt{F2.did=D2.did}}(\boldsymbol{\rho}_{\texttt{D2}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{D}}(\texttt{Dept}) \times \boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance})))$
- $\boldsymbol{\sigma}_{\texttt{F3.budget=F.budget}}(\boldsymbol{\rho}_{\texttt{F}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F2}}(\texttt{Finance}) \times \boldsymbol{\rho}_{\texttt{F3}}(\texttt{Finance})))))$

Optimization of Logical Queries

Step 9. Removal of redundant joins

Subexpression

$$\pi_{\mathtt{D.*,F.*}}\, \sigma_{\mathtt{D.floor=D2.floor} \wedge \mathtt{D2.name} \neq \text{'CID'}} (\rho_{\mathtt{D2}}(\mathtt{Dept}) \times \rho_{\mathtt{F2}}(\mathtt{Finance}) \times \rho_{\mathtt{D}}(\mathtt{Dept}) \times \rho_{\mathtt{F}}(\mathtt{Finance}))$$

Is not a select-project-join expression, because of the inequality. Therefore, no redundant join can be removed.

Therefore, in this example, no redundant join can be removed.

Optimization of Logical Queries

Step 10. Application of the algebraic laws

$$\pi_{\texttt{F.budget,E.eid}}\Big(\pi_{\texttt{F.*,E.eid}}\Big(\pi_{\texttt{E.eid,e.did}}\sigma_{\texttt{E.hobby}=\texttt{'yodeling'}}\rho_{\texttt{E}}(\texttt{Emp})$$
$$\underset{\texttt{E.did}=\texttt{D.did}}{\bowtie}\ \pi_{\texttt{D.did}}\rho_{\texttt{D}}(\texttt{Dept})\ \underset{\texttt{D.did}=\texttt{F.did}}{\bowtie}\ \rho_{\texttt{F}}(\texttt{Finance})\Big)\Big)$$
$$\overline{\bowtie}\ \Big(\pi_{\texttt{F.*}}\ \Big(\pi_{\texttt{D2.floor}}\sigma_{\texttt{D2.name}\neq\texttt{'CID'}}\rho_{\texttt{D2}}(\texttt{Dept})$$
$$\underset{\texttt{D.floor}=\texttt{D2.floor}}{\bowtie}\ \pi_{\texttt{D2.floor}}\rho_{\texttt{D}}(\texttt{Dept})\times\pi\rho_{\texttt{F2}}(\texttt{Finance})$$
$$\times\ \rho_{\texttt{F}}(\texttt{Finance})\Big)$$
$$\cup\ \pi_{\texttt{F.*}}\ \Big(\big(\pi_{\texttt{F.*,F2.*}}\big(\pi_{\texttt{D.floor}}\rho_{\texttt{D2}}(\texttt{Dept})\ \underset{\texttt{D.floor}=\texttt{D2.floor}}{\bowtie}\ \pi_{\texttt{D2.floor,D2.did}}\rho_{\texttt{D}}(\texttt{Dept})$$
$$\underset{\texttt{F2.did}=\texttt{D2.did}}{\bowtie}\ \rho_{\texttt{F2}}(\texttt{Finance})\times\rho_{\texttt{F}}(\texttt{Finance})\big)$$
$$\overline{\bowtie}\ \big(\pi_{\texttt{F2.*,F.*}}\sigma_{\texttt{MAX(F3.expenses)}>\texttt{F2.expenses}}\ \gamma_{\texttt{MAX(F3.expenses),F.*,F2.*}}$$
$$\big(\rho_{\texttt{F}}(\texttt{Finance})\ \underset{\texttt{F3.budget}=\texttt{F.budget}}{\bowtie}\ \pi_{\texttt{F3.budget,F3.expenses}}\rho_{\texttt{F3}}(\texttt{Finance})\times\rho_{\texttt{F2}}(\texttt{Finance})\big)\big)\big)\big)$$