Optimizing logical query plans

Exercises

Academic year 2017-2018

Algebraic laws

Exercise 1. Consider the following relational schema:

- Hotel(id, name, address)
- Room(rid, hid, type, price)
- Booking(hid, gid, date_from, date_to, rid)
- Guest(gid, name, address)

Translate the following SQL queries into the relational algebra and use the algebraic laws to improve the query plan.

1. SELECT R.rid, R.type, R.price
   FROM Room R, Booking B, Hotel H
   WHERE R.rid = B.rid AND B.hid = H.hid
   AND H.name = 'Hilton' AND R.price > 100

2. SELECT G.gid, G.name
   FROM Room R, Hotel H, Booking B, Guest G
   WHERE H.hid = B.hid AND G.gid = B.gid
   AND H.hid = R.hid AND H.name = 'Hilton'
   AND date_from >= '1-Oct-2003' AND date_to <= '31-Dec-2003'

Select-project-join expressions and conjunctive queries

Exercise 2. Consider the “beer drinkers database” consisting of the following relations:

- Visit(drinker, café)
- Appreciate(drinker, beer)
- Serve(café, beer)
Write both (1) select-project-join expressions and (2) conjunctive queries for the following queries:

1. Give every drinker \( d \) who visits a café that serves a beer appreciated by \( d \).
2. Give all pairs \((d,b)\) such that the café \( b \) serves a beer appreciated by \( d \).

**Exercise 3.** Consider a binary relation \( Q(A,B) \). Translate the following SQL queries into select-project-join expressions and then into conjunctive queries:

1. \( \text{SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3 WHERE Q1.B = Q2.A AND Q2.B = Q3.A} \)

**Exercise 4.** Consider the relations \( R(A,B), S(C), T(D,E), U(F,G), \) and \( V(A,B,C) \). Translate the following conjunctive queries into select-project-join expressions. What is the corresponding SQL query?

1. \( Q_1(x,y) \leftarrow S(x), T(x,3), U(x,y) \)
2. \( Q_2(y) \leftarrow S(x), R(x,y) \)
3. \( Q_3(x) \leftarrow V(x,n,s), R(x,a), T(a,'Boeing'), S(s) \)

**Containment and optimization of conjunctive queries**

**Exercise 5.** Consider the following conjunctive queries:

- \( Q_1(x,y) \leftarrow Q(x,a), Q(a,b), Q(b,y) \)
- \( Q_2(x,y) \leftarrow Q(x,a), Q(a,b), Q(b,c), Q(c,y) \)
- \( Q_3(x,y) \leftarrow Q(x,a), Q(a,1), Q(1,b), Q(b,y) \)
- \( Q_4(x,y) \leftarrow Q(x,y), Q(y,x) \)

Give all pairs \((Q_i, Q_j)\) such that \( Q_i \) is contained in \( Q_j \). Are there equivalent queries?

**Exercise 6.** Optimize the following conjunctive queries:

- \( Q_1(x,z) \leftarrow R(x,y), R(y,w), R(y,z) \)
Exercise 7. Consider the beer drinkers database again:

- Visit(drinker, café)
- Appreciate(drinker, beer)
- Serve(cafè, beer)

The query compiler has computed the following logical query plan:

\[ \pi_{B_1 \cdot \text{drinker}} \sigma_{B_1 \cdot \text{cafe} = B_2 \cdot \text{cafe}} \sigma_{L_1 \cdot \text{drinker}} \]

\[ \sigma_{L_1 \cdot \text{beer} = L_2 \cdot \text{beer}} \sigma_{L_2 \cdot \text{drinker} = \text{Jan}} \sigma_{L_1 \cdot \text{beer} = \text{S}} \sigma_{S \cdot \text{cafe} = B_2 \cdot \text{cafe}} \]

\[ (\rho_{B_1 \cdot \text{Visit}}) \times \rho_{B_2 \cdot \text{Visit}} \times \rho_{S \cdot \text{Serve}} \times \rho_{L_1 \cdot \text{Appreciate}} \times \rho_{L_2 \cdot \text{Appreciate}} \]

Optimize this plan by removing redundant joins.

Integrated exercises

Exercise 8. Consider the following relational schema, containing information on employees (Emp), departments (Dept), and finances (Finance):

- Emp(eid, did, sal, hobby)
- Dept(did, dname, floor, phone)
- Finance(did, budget, sales, expenses)

For each of the following SQL statements:

1. Translate the query into the relational algebra.
2. Remove redundant joins from the select-project-join subexpressions in the obtained logical query plan.
3. Make use of the algebraic laws to further optimize the obtained expression.

1. SELECT MAX(E.sal)
   FROM Emp E
   WHERE E.eid IN
   (SELECT E1.eid
FROM Emp E1, Emp E2, Dept D1, Dept D2, Finance F
WHERE F.budget = 100 AND E1.did = D1.did AND E1.did = F.did
AND E2.did = D2.did AND E2.did = F.did
AND D1.floor = 1 AND D2.dname = 'CID'
)
GROUP BY E.hobby

2. SELECT D.floor
FROM Dept D, Emp E
WHERE
  (D.floor = 1
  OR D.floor IN
    ( SELECT D2.floor FROM Dept D2, Finance F1
        WHERE F1.budget > 150 AND D2.did = F1.did)
  )
AND E.did = D.did
AND E.did IN (SELECT F2.did FROM Finance F2, Emp E2
    WHERE F2.did = E.did AND E2.did = D.did
    AND E2.eid = E.eid AND F2.expenses = 300)

3. 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
      )
    )
  )
)

Exercise 9. Consider the following relational schema:

- Suppliers(sid, sname, city)
- Supply(sid, pid)
- Parts(pid, pname, price)

For each of the following SQL statements:

1. Translate the query into the relational algebra.
2. Remove redundant joins from the select-project-join subexpressions in the obtained logical query plan.

3. Make use of the algebraic laws to further optimize the obtained expression.

1. SELECT S.sname, P.pname
   FROM Suppliers S1, Suppliers S2, Parts P, Supply Y
   WHERE S1.sid = Y.sid AND S2.sid = Y.sid AND Y.pid = P.pid
   AND S2.city = 'Madison' AND P.price <= 100

2. SELECT S.sname, S.city
   FROM Suppliers S, Parts P, Supply Y
   WHERE S.sid = Y.sid AND Y.pid = P.pid
   AND P.price IN
     (SELECT P2.price FROM Parts P2, Supply Y2
      WHERE Y2.pid = P2.pid and Y2.sid = S.sid)

3. SELECT MAX(P.price), S.sname
   FROM Parts P, Suppliers S
   WHERE S.city = 'Ham'
   AND (P.Price, S.city) IN
     (SELECT P2.Price, S2.city FROM Parts P2, Supply Y, Suppliers S2
      WHERE P2.pid = Y.sid AND Y.pid = S2.pid
      AND S.sid = S2.sid AND P.pid = P2.pid)
   GROUP BY S.sname