

Optimization of Logical Queries

Task:

Consider the following relational schema:

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

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

```
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
```

Optimization of Logical Queries

Solution

The translation gives us the following relational algebra expression:

$$\pi_{R.rid,R.type,R.price} \sigma_{R.rid=B.rid \wedge B.hid=H.hid \wedge H.name='Hilton' \wedge R.price>100} (\rho_R(\text{Room}) \times \rho_H(\text{Hotel}) \times \rho_B(\text{Booking}))$$

Optimization of Logical Queries

Solution

The translation gives us the following relational algebra expression:

$$\pi_{R.rid,R.type,R.price} \sigma_{R.rid=B.rid \wedge B.hid=H.hid \wedge H.name='Hilton' \wedge R.price>100} (\rho_R(\text{Room}) \times \rho_H(\text{Hotel}) \times \rho_B(\text{Booking}))$$

First, we split the selections:

$$\pi_{R.rid,R.type,R.price} \sigma_{R.rid=B.rid} \sigma_{B.hid=H.hid} \sigma_{H.name='Hilton'} \sigma_{R.price>100} (\rho_R(\text{Room}) \times \rho_H(\text{Hotel}) \times \rho_B(\text{Booking}))$$

And we push the selections:

$$\pi_{R.rid,R.type,R.price} \sigma_{R.rid=B.rid} (\sigma_{R.price>100} \rho_R(\text{Room}) \times \sigma_{B.hid=H.hid} (\sigma_{H.name='Hilton'} \rho_H(\text{Hotel}) \times \rho_B(\text{Booking})))$$

Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

$$\pi_{R.rid,R.type,R.price}(\sigma_{R.price>100} \rho_R(\text{Room}) \bowtie_{R.rid=B.rid} (\sigma_{H.name='Hilton'} \rho_H(\text{Hotel}) \bowtie_{B.hid=H.hid} \rho_B(\text{Booking})))$$

Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

$$\pi_{R.rid,R.type,R.price}(\sigma_{R.price>100} \rho_R(\text{Room}) \bowtie_{R.rid=B.rid} (\sigma_{H.name='Hilton'} \rho_H(\text{Hotel}) \bowtie_{B.hid=H.hid} \rho_B(\text{Booking})))$$

Finally, the projections are pushed:

$$\pi_{R.rid,R.type,R.price}(\pi_{R.rid,R.type,R.price} \sigma_{R.price>100} \rho_R(\text{Room}) \bowtie_{R.rid=B.rid} \pi_{B.rid}(\pi_{H.hid} \sigma_{H.name='Hilton'} \rho_H(\text{Hotel}) \bowtie_{B.hid=H.hid} \pi_{B.hid,B.rid} \rho_B(\text{Booking})))$$

Conjunctive queries

Task:

Consider a binary relation $Q(A, B)$. First translate the following SQL query into a select-project-join expression, and then into a conjunctive query:

```
SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3
WHERE Q1.B = Q2.A and Q2.B = Q3.A
```

Conjunctive queries

Task:

Consider a binary relation $Q(A, B)$. First translate the following SQL query into a select-project-join expression, and then into a conjunctive query:

```
SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3
WHERE Q1.B = Q2.A and Q2.B = Q3.A
```

Solution

The corresponding select-project-join expression is:

$$\pi_{Q1.A, Q3.B} \sigma_{Q1.B=Q2.A \wedge Q2.B=Q3.A} (\rho_{Q1}(Q) \times \rho_{Q2}(Q) \times \rho_{Q3}(Q))$$

Conjunctive queries

Task:

Consider a binary relation $Q(A, B)$. First translate the following SQL query into a select-project-join expression, and then into a conjunctive query:

```
SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3
WHERE Q1.B = Q2.A and Q2.B = Q3.A
```

Solution

The corresponding select-project-join expression is:

$$\pi_{Q_1.A, Q_3.B} \sigma_{Q_1.B=Q_2.A \wedge Q_2.B=Q_3.A} (\rho_{Q_1}(Q) \times \rho_{Q_2}(Q) \times \rho_{Q_3}(Q))$$

To translate this into a conjunctive query, we create an atom with distinct variables for each relation:

$$P(x_{Q_1.A}, x_{Q_3.B}) \leftarrow Q(x_{Q_1.A}, x_{Q_1.B}), Q(x_{Q_2.A}, x_{Q_2.B}), Q(x_{Q_3.A}, x_{Q_3.B})$$

Conjunctive queries

Task:

Consider a binary relation $Q(A, B)$. First translate the following SQL query into a select-project-join expression, and then into a conjunctive query:

```
SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3
WHERE Q1.B = Q2.A and Q2.B = Q3.A
```

Solution

The corresponding select-project-join expression is:

$$\pi_{Q_1.A, Q_3.B} \sigma_{Q_1.B=Q_2.A \wedge Q_2.B=Q_3.A} (\rho_{Q_1}(Q) \times \rho_{Q_2}(Q) \times \rho_{Q_3}(Q))$$

We then unify variables that must be equal:

$$P(x_{Q_1.A}, x_{Q_3.B}) \leftarrow Q(x_{Q_1.A}, x_{Q_1.B}), Q(x_{Q_1.B}, x_{Q_2.B}), Q(x_{Q_2.B}, x_{Q_3.B})$$

Conjunctive queries

Task:

Consider a binary relation $Q(A, B)$. First translate the following SQL query into a select-project-join expression, and then into a conjunctive query:

```
SELECT Q1.A, Q3.B FROM Q Q1, Q Q2, Q Q3
WHERE Q1.B = Q2.A and Q2.B = Q3.A
```

Solution

The corresponding select-project-join expression is:

$$\pi_{Q1.A, Q3.B} \sigma_{Q1.B=Q2.A \wedge Q2.B=Q3.A} (\rho_{Q1}(Q) \times \rho_{Q2}(Q) \times \rho_{Q3}(Q))$$

(Optionally), we rename the variables:

$$P(x, y) \leftarrow Q(x, k), Q(k, l), Q(l, y)$$

Conjunctive queries

Task:

Consider the relations $R(A, B)$, $S(C)$, $T(D, E)$, $U(F, G)$ and $V(A, B, C)$. Translate the following conjunctive query into a select-project-join expression. What is the corresponding SQL query?

$$Q_1(x, y) \leftarrow S(x), T(x, 3), U(x, y)$$

Conjunctive queries

Task:

Consider the relations $R(A, B)$, $S(C)$, $T(D, E)$, $U(F, G)$ and $V(A, B, C)$. Translate the following conjunctive query into a select-project-join expression. What is the corresponding SQL query?

$$Q_1(x, y) \leftarrow S(x), T(x, 3), U(x, y)$$

Solution

The select-project-join expression is:

$$\pi_{S.C, U.G} \sigma_{S.C=U.F} \sigma_{S.C=T.D} \sigma_{T.E=3} (S \times T \times U)$$

Conjunctive queries

Task:

Consider the relations $R(A, B)$, $S(C)$, $T(D, E)$, $U(F, G)$ and $V(A, B, C)$. Translate the following conjunctive query into a select-project-join expression. What is the corresponding SQL query?

$$Q_1(x, y) \leftarrow S(x), T(x, 3), U(x, y)$$

Solution

The select-project-join expression is:

$$\pi_{S.C, U.G} \sigma_{S.C=U.F} \sigma_{S.C=T.D} \sigma_{T.E=3} (S \times T \times U)$$

The corresponding SQL query is:

```
SELECT S.C, U.G
FROM S, T, U
WHERE S.C = U.F AND S.C = T.D AND T.E = 3
```

Containment and optimization of conjunctive queries

Recap

- A *substitution* of Q in D is a function that maps each variable occurring in Q to a constant in D .
- A *matching* of Q in D is a substitution σ such that $\sigma(\text{body}) \subseteq D$
- $Q(D) = \{\sigma(\text{head}) \mid \sigma \text{ a matching of } Q \text{ in } D\}$
- The *canonical database* of a query Q_i is the set of atoms D_i obtained from the body of Q , where each variable x is considered as a constant.
- To test whether $Q_i \subseteq Q_j$, it suffices to check whether the head of Q_i (considered as a fact) occurs in $Q_j(D_i)$.

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$?

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$? Candidate substitution:

$$x \mapsto x, y \mapsto y$$

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$? Candidate substitution:

$$x \mapsto x, y \mapsto y$$

Then, to match $Q(x, a)$ we would need $a \mapsto a$.

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$? Candidate substitution:

$$x \mapsto x, y \mapsto y, a \mapsto a$$

Then, to match $Q(a, b)$ we would need $b \mapsto b$.

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$? Candidate substitution:

$$x \mapsto x, y \mapsto y, a \mapsto a, b \mapsto b$$

Then, to match $Q(b, c)$ we would need $c \mapsto y$.

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_1)$? Candidate substitution:

$$x \mapsto x, y \mapsto y, a \mapsto a, b \mapsto b, c \mapsto y$$

But then, to match $Q(c, y)$ is mapped to $Q(y, y)$, which is not in D_1 !

Containment and optimization of conjunctive queries

Consider the following conjunctive queries:

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

Is $Q_1 \subseteq Q_2$? Is $Q_3 \subseteq Q_2$?

Solution: $Q_1 \subseteq Q_2$?

We construct the canonical database for Q_1 :

$$D_1 := \{Q(x, a), Q(a, b), Q(b, y)\}.$$

No candidate substitution is a matching. Hence, $Q_2(D_1) = \emptyset$ and $(x, y) \notin Q_2(D_1)$.

Therefore: $Q_1 \not\subseteq Q_2$ (we constructed a counterexample).

Containment and optimization of conjunctive queries

Solution: $Q_3 \subseteq Q_2$?

- $Q_3 : P(x, y) \leftarrow Q(x, a), Q(a, 1), Q(1, b), Q(b, y)$
- $Q_2 : P(x, y) \leftarrow Q(x, a), Q(a, b), Q(b, c), Q(c, y)$

We construct the canonical database for Q_3 :

$$D_3 := \{Q(x, a), Q(a, 1), Q(1, b), Q(b, y)\}.$$

Is $(x, y) \in Q_2(D_3)$?

Containment and optimization of conjunctive queries

Solution: $Q_3 \subseteq Q_2$?

- $Q_3 : P(x, y) \leftarrow Q(x, a), Q(a, 1), Q(1, b), Q(b, y)$
- $Q_2 : P(x, y) \leftarrow Q(x, a), Q(a, b), Q(b, c), Q(c, y)$

We construct the canonical database for Q_3 :

$$D_3 := \{Q(x, a), Q(a, 1), Q(1, b), Q(b, y)\}.$$

Yes! The following matching ensures that $(x, y) \in Q_2(D_3)$

$$[x \rightarrow x, y \rightarrow y, a \rightarrow a, b \rightarrow 1, c \rightarrow b]$$

Therefore: $Q_3 \subseteq Q_2$.

Optimization of conjunctive queries

Task

Optimize the following conjunctive query

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Optimization of conjunctive queries

Task

Optimize the following conjunctive query

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution

- The atom $R(x, y)$ cannot be removed (why?).

Optimization of conjunctive queries

Task

Optimize the following conjunctive query

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution

- The atom $R(x, y)$ cannot be removed (why?).
- We check whether $R(y, w)$ can be removed. Let P be the following conjunctive query:

$$P(x, z) \leftarrow R(x, y), R(y, z)$$

We must check whether $P \subseteq Q$ ($Q \subseteq P$ is trivially true).

Optimization of conjunctive queries

Task

Optimize the following conjunctive query

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution

- The atom $R(x, y)$ cannot be removed (why?).
- We check whether $R(y, w)$ can be removed. Let P be the following conjunctive query:

$$P(x, z) \leftarrow R(x, y), R(y, z)$$

We must check whether $P \subseteq Q$ ($Q \subseteq P$ is trivially true). Therefore, we construct the canonical database for P :

$$D := \{R(x, y), R(y, z)\}$$

The following matching ensures that $(x, z) \in Q(D)$, and hence that $P \subseteq Q$:

$$[x \rightarrow x, y \rightarrow y, w \rightarrow z, z \rightarrow z]$$

Optimization of conjunctive queries

Task

Optimize the following conjunctive query:

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution (continued)

- Since P is equivalent and “more optimal”, we now continue with optimizing query P .

$$P(x, z) \leftarrow R(x, y), R(y, z)$$

Optimization of conjunctive queries

Task

Optimize the following conjunctive query:

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution (continued)

- Since P is equivalent and “more optimal”, we now continue with optimizing query P .

$$P(x, z) \leftarrow R(x, y), R(y, z)$$

- The atom $R(y, z)$ cannot be removed (why?)

Optimization of conjunctive queries

Task

Optimize the following conjunctive query:

$$Q(x, z) \leftarrow R(x, y), R(y, w), R(y, z)$$

Solution (continued)

- Since P is equivalent and “more optimal”, we now continue with optimizing query P .

$$P(x, z) \leftarrow R(x, y), R(y, z)$$

- The atom $R(y, z)$ cannot be removed (why?)
- We cannot remove any other atom. Therefore, P is the minimal query equivalent to Q .

Integrated Exercise

Task

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 the following SQL statement:

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. By means of the algebraic laws, further optimize the obtained expression.

Integrated Exercise

Task (continued)

```
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
```

Integrated Exercise

Solution: translation into the relational algebra

First, we normalize the query to a form with only EXISTS and NOT EXISTS subqueries:

```
SELECT MAX(E.sal)
FROM Emp E
WHERE EXISTS
  (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'
        AND E1.eid = E.eid
   )
GROUP BY E.hobby
```

Integrated Exercise

Solution: translation into the relational algebra

Then, we translate the subquery in the following expression e_1 :

$$\begin{aligned} & \pi_{E_1.\text{eid}, E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \sigma_{F.\text{budget}=100 \wedge E_1.\text{did}=D_1.\text{did} \wedge E_1.\text{did}=F.\text{did}} \\ & \sigma_{E_2.\text{did}=D_2.\text{did} \wedge E_2.\text{did}=F.\text{did} \wedge D_1.\text{floor}=1 \wedge D_2.\text{dname}='CID' \wedge E_1.\text{eid}=E.\text{eid}} \\ & (\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{E_2}(\text{Emp}) \\ & \quad \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance})) \end{aligned}$$

And we translate the FROM-WHERE part of the outer query without subqueries:

$$e_2 := \rho_E(\text{Emp})$$

The decorrelation of the subquery gives:

$$e_3 := \hat{e}_2 \bowtie \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}}(e_1)$$

Notice that \hat{e}_2 is empty! Therefore, the translation of the complete query is:

$$e_4 := \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}}(e_1)$$

Integrated Exercise

Solution: translation into the relational algebra

This leads to (after merging projections):

$$\begin{aligned} & \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \\ & \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \sigma_{F.\text{budget}=100 \wedge E_1.\text{did}=D_1.\text{did} \wedge E_1.\text{did}=F.\text{did}} \\ & \sigma_{E_2.\text{did}=D_2.\text{did} \wedge E_2.\text{did}=F.\text{did} \wedge D_1.\text{floor}=1 \wedge D_2.\text{dname}='CID' \wedge E_1.\text{eid}=E.\text{eid}} \\ & (\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{E_2}(\text{Emp}) \\ & \quad \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance})) \end{aligned}$$

Integrated Exercise

Solution: translation into the relational algebra

The query only contains *one* (maximal) select-project-join subexpression:

$$\begin{aligned} & \pi_{E.eid, E.did, E.sal, E.hobby} \sigma_{F.budget=100 \wedge E_1.did=D_1.did \wedge E_1.did=F.did} \\ & \quad \sigma_{E_2.did=D_2.did \wedge E_2.did=F.did \wedge D_1.floor=1 \wedge D_2.dname='CID' \wedge E_1.eid=E.eid} \\ & \quad (\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{E_2}(\text{Emp}) \\ & \quad \quad \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance})) \end{aligned}$$

To remove redundant joins, we translate it to a conjunctive query:

$$\begin{aligned} Q_1(a_1, a_2, a_3, a_4) \leftarrow & \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Emp}(c_1, b_2, c_3, c_4), \\ & \text{Dept}(b_2, d_2, 1, d_4), \text{Dept}(b_2, 'CID', e_3, e_4), \\ & \text{Finance}(b_2, 100, f_3, f_4) \end{aligned}$$

Integrated Exercise

Solution: removal of redundant joins

$$Q_1(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Emp}(c_1, b_2, c_3, c_4), \\ \text{Dept}(b_2, d_2, 1, d_4), \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \\ \text{Finance}(b_2, 100, f_3, f_4)$$

- We cannot remove $\text{Emp}(a_1, a_2, a_3, a_4)$ and $\text{Finance}(b_2, 100, f_3, f_4)$ (why?)

Integrated Exercise

Solution: removal of redundant joins

$$Q_1(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Emp}(c_1, b_2, c_3, c_4), \\ \text{Dept}(b_2, d_2, 1, d_4), \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \\ \text{Finance}(b_2, 100, f_3, f_4)$$

- We cannot remove $\text{Emp}(a_1, a_2, a_3, a_4)$ and $\text{Finance}(b_2, 100, f_3, f_4)$ (why?)
- We check whether $\text{Emp}(a_1, b_2, b_3, b_4)$ can be removed. To this end, we build the canonical database of Q_1 without this atom:

$$D_2 = \{ \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(c_1, b_2, c_3, c_4), \text{Dept}(b_2, d_2, 1, d_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4) \}$$

Note that $(a_1, a_2, a_3, a_4) \notin Q_1(D_2)$ (why?), and it ensues that the atom cannot be removed from Q_1 .

Integrated Exercise

Solution: removal of redundant joins

$$Q_1(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Emp}(c_1, b_2, c_3, c_4), \\ \text{Dept}(b_2, d_2, 1, d_4), \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \\ \text{Finance}(b_2, 100, f_3, f_4)$$

- We check whether $\text{Emp}(c_1, b_2, c_3, c_4)$ can be removed. To this end, we build the canonical database of Q_1 without this atom:

$$D_3 = \{ \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Dept}(b_2, d_2, 1, d_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4) \}$$

This time, $(a_1, a_2, a_3, a_4) \in Q_1(D_3)$. Let Q_3 be the conjunctive query Q_1 without $\text{Emp}(a_1, b_2, b_3, b_4)$. We have just shown that $Q_3 \equiv Q_1$, and therefore that this atom can be removed. We can continue the optimization procedure with Q_3 .

Integrated Exercise

Solution: removal of redundant joins

$$Q_3(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Dept}(b_2, d_2, 1, d_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4)$$

- We check whether $\text{Dept}(b_2, d_1, 1, d_4)$ can be removed. To this end, we build the canonical database of Q_3 without this atom:

$$D_4 = \{ \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4) \}$$

Observe that $(a_1, a_2, a_3, a_4) \notin Q_3(D_4)$ (why?) and it ensues that the atom cannot be removed from Q_3 .

Integrated Exercise

Solution: removal of redundant joins

$$Q_3(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Dept}(b_2, d_2, 1, d_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4)$$

- We check whether $\text{Dept}(b_2, \text{'CID'}, e_3, e_4)$ can be removed. To this end, we build the canonical database of Q_3 without this atom:

$$D_5 = \{ \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \\ \text{Dept}(b_2, d_2, 1, d_4), \text{Finance}(b_2, 100, f_3, f_4) \}$$

Observe that $(a_1, a_2, a_3, a_4) \notin Q_3(D_5)$ (why?) and it ensues that the atom cannot be removed from Q_3 .

Integrated Exercise

Solution: removal of redundant joins

Thus, the optimized conjunctive query is:

$$Q_3(a_1, a_2, a_3, a_4) \leftarrow \text{Emp}(a_1, a_2, a_3, a_4), \text{Emp}(a_1, b_2, b_3, b_4), \text{Dept}(b_2, d_2, 1, d_4), \\ \text{Dept}(b_2, \text{'CID'}, e_3, e_4), \text{Finance}(b_2, 100, f_3, f_4)$$

And $\rho_{E_2}(\text{Emp})$ can be removed from the select-project-join expression (as well as the corresponding selections). The translation of Q_3 into a select-project-join expression is indeed:

$$\pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}}$$

$$\sigma_{F.\text{budget}=100 \wedge E_1.\text{did}=D_1.\text{did} \wedge E_1.\text{did}=F.\text{did} \wedge D_1.\text{floor}=1}$$

$$\sigma_{D_2.\text{did}=E_1.\text{did} \wedge D_2.\text{dname}=\text{'CID'} \wedge E_1.\text{eid}=E.\text{eid}}$$

$$(\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{D_1}(\text{Dept}))$$

$$\times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance}))$$

Integrated Exercise

Solution: application of the algebraic laws

The logical query plan for the whole SQL query where we removed the redundant joins is:

$$\begin{aligned} & \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \\ & \quad \sigma_{F.\text{budget}=100 \wedge E_1.\text{did}=D_1.\text{did} \wedge E_1.\text{did}=F.\text{did} \wedge D_1.\text{floor}=1} \\ & \quad \quad \sigma_{D_2.\text{did}=E_1.\text{did} \wedge D_2.\text{dname}='CID' \wedge E_1.\text{eid}=E.\text{eid}} \\ & \quad (\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance})) \end{aligned}$$

Now, we apply the algebraic laws. Pushing the selections gives:

$$\begin{aligned} & \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \sigma_{E_1.\text{eid}=E.\text{eid}} \\ & \quad (\rho_E(\text{Emp}) \times \sigma_{E_1.\text{did}=F.\text{did}} (\sigma_{D_2.\text{did}=E_1.\text{did}} \\ & \quad (\sigma_{E_1.\text{did}=D_1.\text{did}} (\rho_{E_1}(\text{Emp}) \times \sigma_{D_1.\text{floor}=1} (\rho_{D_1}(\text{Dept})))) \\ & \quad \times \sigma_{D_2.\text{dname}='CID'} \rho_{D_2}(\text{Dept})) \times \sigma_{F.\text{budget}=100} (\rho_F(\text{Finance})))) \end{aligned}$$

Integrated Exercise

Solution (continued)

Recognizing joins:

$$\begin{aligned} & \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \\ & \rho_E(\text{Emp}) \bowtie_{E_1.\text{eid}=E.\text{eid}} \left(\left(\left(\rho_{E_1}(\text{Emp}) \bowtie_{E_1.\text{did}=D_1.\text{did}} \sigma_{D_1.\text{floor}=1} \rho_{D_1}(\text{Dept}) \right) \right. \right. \\ & \left. \left. \bowtie_{E_1.\text{did}=D_2.\text{did}} \sigma_{D_2.\text{dname}='CID'}(\rho_{D_2}(\text{Dept})) \right) \right) \bowtie_{E_1.\text{did}=F.\text{did}} \sigma_{F.\text{budget}=100}(\rho_F(\text{Finance})) \end{aligned}$$

Pushing the projections:

$$\begin{aligned} & \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{sal}, E.\text{hobby}} \left(\pi_{E.\text{eid}, E.\text{sal}, E.\text{hobby}} \rho_E(\text{Emp}) \right) \\ & \quad \bowtie_{E_1.\text{eid}=E.\text{eid}} \pi_{E_1.\text{eid}} \left(\left(\left(\pi_{E_1.\text{did}, E_1.\text{eid}} \left(\pi_{E_1.\text{did}, E_1.\text{eid}} \rho_{E_1}(\text{Emp}) \right) \right. \right. \right. \\ & \quad \left. \left. \left. \bowtie_{E_1.\text{did}=D_1.\text{did}} \pi_{D_1.\text{did}} \sigma_{D_1.\text{floor}=1}(\rho_{D_1}(\text{Dept})) \right) \right) \right. \\ & \quad \left. \left. \left. \bowtie_{E_1.\text{did}=D_2.\text{did}} \pi_{D_2.\text{did}} \sigma_{D_2.\text{dname}='CID'}(\rho_{D_2}(\text{Dept})) \right) \right) \right) \\ & \quad \quad \quad \left. \left. \left. \bowtie_{E_1.\text{did}=F.\text{did}} \pi_{F.\text{did}} \sigma_{F.\text{budget}=100}(\rho_F(\text{Finance})) \right) \right) \right) \end{aligned}$$