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.

```sql
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:

\[ \pi_{R.\text{rid}, R.\text{type}, R.\text{price}} \sigma_{R.\text{rid}=B.\text{rid} \land B.\text{hid}=H.\text{hid} \land H.\text{name}=\text{Hilton} \land R.\text{price}>100} \]

\[ (\rho_{R}(\text{Room}) \times \rho_{H}(\text{Hotel}) \times \rho_{B}(\text{Booking})) \]
Optimization of Logical Queries

Solution

The translation gives us the following relational algebra expression:

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

First, we split the selections:

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

And we push the selections:

\[ \pi_{R.\text{rid}, R.\text{type}, R.\text{price}} \sigma_{R.\text{rid}=B.\text{rid}} (\sigma_{R.\text{price}>100} \rho_R(\text{Room}) \times \sigma_{B.\text{hid}=H.\text{hid}} (\sigma_{H.\text{name}=\text{Hilton}} \rho_H(\text{Hotel}) \times \rho_B(\text{Booking}))) \]
Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

$$\pi_{R.\text{rid}, R.\text{type}, R.\text{price}}(\sigma_{R.\text{price}>100} \rho_R(\text{Room}) \bowtie \left( \sigma_{\text{H.name}='\text{Hilton'}} \rho_H(\text{Hotel}) \bowtie \rho_B(\text{Booking}) \right)$$
Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

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

Finally, the projections are pushed:

\[
\pi_{R.\text{rid}, R.\text{type}, R.\text{price}}(\pi_{R.\text{rid}, R.\text{type}, R.\text{price}}(\sigma_{R.\text{price} > 100} \rho_R(\text{Room}))) \\
\Join_{R.\text{rid} = B.\text{rid}} \pi_{B.\text{hid}}(\pi_{H.\text{hid}}(\sigma_{H.\text{name} = 'Hilton'} \rho_H(\text{Hotel}))) \\
\Join_{B.\text{hid} = H.\text{hid}} \pi_{B.\text{hid}, B.\text{rid}}(\rho_B(\text{Booking})))
\]
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
```
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 $Q_1.A$, $Q_3.B$ FROM $Q$ $Q_1$, $Q$ $Q_2$, $Q$ $Q_3$
WHERE $Q_1.B = Q_2.A$ and $Q_2.B = Q_3.A$

Solution
The corresponding select-project-join expression is:

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

\[
\text{SELECT } Q1.A, Q3.B \text{ FROM } Q \ Q1, Q \ Q2, Q \ Q3 \\
\text{WHERE } Q1.B = Q2.A \text{ and } Q2.B = Q3.A
\]

Solution
The corresponding select-project-join expression is:

\[
\pi_{Q1.A,Q3.B} \sigma_{Q1.B=Q2.A \land Q2.B=Q3.A}(\rho_{Q1}(Q) \times \rho_{Q2}(Q) \times \rho_{Q3}(Q))
\]

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

\[
P(x_{Q1.A}, x_{Q3.B}) \leftarrow Q(x_{Q1.A}, x_{Q1.B}), Q(x_{Q2.A}, x_{Q2.B}), Q(x_{Q3.A}, x_{Q3.B})
\]
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

Solution
The corresponding select-project-join expression is:

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

We then unify variables that must be equal:

$$P(x_{Q1.A}, x_{Q3.B}) \leftarrow Q(x_{Q1.A}, x_{Q1.B}), Q(x_{Q1.B}, x_{Q2.B}), Q(x_{Q2.B}, x_{Q3.B})$$
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
```

Solution

The corresponding select-project-join expression is:

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

(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:

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

$$Q_1(x, y) ← S(x), T(x, 3), U(x, y)$$

Solution

The select-project-join expression is:

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

The corresponding SQL query is:

```sql
SELECT S.C, U.G
FROM S, T, U
WHERE C = F AND C = D AND 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 $\sigma$ 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 $Q_j(D_i)$. 

Solution of the exercises
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 \( Q_1. \) For ease of readability, and to avoid confusion, we denote the constants in this canonical database by \( \dot{x}, \dot{a}, \ldots \)

\[
D_1 := \{ Q(\dot{x}, \dot{a}), Q(\dot{a}, \dot{b}), Q(\dot{b}, \dot{y}) \}.
\]

Is \((\dot{x}, \dot{y}) \in Q_2(D_1)? \)
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 \( Q_1 \):

\[
D_1 := \{Q(\dot{x}, \dot{a}), Q(\dot{a}, \dot{b}), Q(\dot{b}, \dot{y})\}.
\]

Is \((\dot{x}, \dot{y}) \in Q_2(D_1)\) ? Candidate substitution:

\[
x \mapsto \dot{x}, \ y \mapsto \dot{y}
\]
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 \( Q_1 \):

\[
D_1 := \{Q(\dot{x}, \dot{a}), Q(\dot{a}, \dot{b}), Q(\dot{b}, \dot{y})\}.
\]

Is \((\dot{x}, \dot{y}) \in Q_2(D_1)\)? Candidate substitution:

\[
x \mapsto \dot{x}, \dot{y} \mapsto y
\]

Then, to mach \( Q(x, a) \) we would need \( a \mapsto \dot{a} \).
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 \(Q_1\):

\[D_1 := \{Q(\dot{x}, \dot{a}), Q(\dot{a}, \dot{b}), Q(\dot{b}, \dot{y})\}\]

Is \((\dot{x}, \dot{y}) \in Q_2(D_1)\)? Candidate substitution:

\[x \mapsto \dot{x}, y \mapsto \dot{y}, a \mapsto \dot{a}\]

Then, to match \(Q(a, b)\) we would need \(b \mapsto \dot{b}\).
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 $Q_1$:

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

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

$$x \mapsto \dot{x}, y \mapsto \dot{y}, a \mapsto \dot{a}, b \mapsto \dot{b}$$

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

Solution of the exercises 20
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 $Q_1$:

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

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

$$x \mapsto \dot{x}, y \mapsto \dot{y}, a \mapsto \dot{a}, b \mapsto \dot{b}, c \mapsto \dot{y}$$

But then, $Q(c, y)$ is mapped to $Q(\dot{y}, \dot{y})$, which is not in $D_1$! So, our candidate substitution is not a matching.
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 \( Q_1 \):

\[
D_1 := \{ Q(\dot{x}, \dot{a}), Q(\dot{a}, \dot{b}), Q(\dot{b}, \dot{y}) \}.
\]

No candidate substitution yielding \((\dot{x}, \dot{y})\) is a matching. Hence, \((\dot{x}, \dot{y}) \notin Q_2(D_1)\).

**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 $Q_3$:

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

Is $(\dot{x}, \dot{y}) \in Q_2(D_3)$?
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 $Q_3$:

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

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

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

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 \( P \):

\[ D := \{ R(\hat{x}, \hat{y}), R(\hat{y}, \hat{z}) \} \]

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

\[ [x \rightarrow \hat{x}, y \rightarrow \hat{y}, w \rightarrow \hat{z}, z \rightarrow \hat{z}] \]
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:

```sql
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 $e_1$:

$$\pi_{E_1.eid,E.eid,E.did,E.sal,E.hobby} \sigma_{F.budget=100 \land E_1.did=D_1.did \land E_1.did=F.did}$$

$$\sigma_{E_2.did=D_2.did \land E_2.did=F.did \land D_1.floor=1 \land D_2.dname='CID' \land E_1.eid=E.eid}$$

$$\rho_{E}(Emp) \times \rho_{E_1}(Emp) \times \rho_{E_2}(Emp)$$

$$\times \rho_{D_1}(Dept) \times \rho_{D_2}(Dept) \times \rho_{F}(Finance))$$

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

$$e_2 := \rho_{E}(Emp)$$

The decorrelation of the subquery gives:

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

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

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

Solution: translation into the relational algebra

This leads to (after merging projections):

$$\pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \sigma_{F.\text{budget}=100 \land E_1.\text{did}=D_1.\text{did} \land E_1.\text{did}=F.\text{did}} \sigma_{E_2.\text{did}=D_2.\text{did} \land E_2.\text{did}=F.\text{did} \land D_1.\text{floor}=1 \land D_2.\text{dname}=\text{'CID'} \land E_1.\text{eid}=E.\text{eid}} \left( \rho_{E}(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{E_2}(\text{Emp}) \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_{F}(\text{Finance}) \right)$$
Integrated Exercise

Solution: translation into the relational algebra

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

\[ \pi_{E.eid, E.did, E.sal, E.hobby} \sigma_{F.budget=100 \land E_1.did=D_1.did \land E_1.did=F.did \land E_2.did=D_2.did \land E_2.did=F.did \land D_1.floor=1 \land D_2.dname='CID' \land E_1.eid=E.eid} (\rho_{E}(Emp) \times \rho_{E_1}(Emp) \times \rho_{E_2}(Emp) \times \rho_{D_1}(Dept) \times \rho_{D_2}(Dept) \times \rho_{F}(Finance)) \]

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

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

Solution: removal of redundant joins

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

- 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

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

- 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:

\[ D_2 = \{ \text{Emp}(\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4), \text{Emp}(\dot{c}_1, \dot{b}_2, \dot{c}_3, \dot{c}_4), \text{Dept}(\dot{b}_2, \dot{d}_2, 1, \dot{d}_4), \text{Dept}(\dot{b}_2, 'CID', \dot{e}_3, \dot{e}_4), \text{Finance}(\dot{b}_2, 100, \dot{f}_3, \dot{f}_4) \} \]

Note that \( (\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4) \notin Q_1(D_2) \) (why?), and it ensues that the atom cannot be removed from \( Q_1 \).
Integrated Exercise

Solution: removal of redundant joins

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

- 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:

\[ D_3 = \{ \text{Emp}(\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4), \text{Emp}(\dot{a}_1, \dot{b}_2, \dot{b}_3, \dot{b}_4), \text{Dept}(\dot{b}_2, \dot{d}_2, 1, \dot{d}_4), \text{Dept}(\dot{b}_2, 'CID', \dot{e}_3, \dot{e}_4), \text{Finance}(\dot{b}_2, 100, \dot{f}_3, \dot{f}_4) \} \]

This time, \( (\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4) \in Q_1(D_3) \). 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

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

- 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:

\[ D_4 = \{ \text{Emp}(\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4), \text{Emp}(\hat{a}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4), \text{Dept}(\hat{b}_2, 'CID', \hat{e}_3, \hat{e}_4), \text{Finance}(\hat{b}_2, 100, \hat{f}_3, \hat{f}_4) \} \]

Observe that \( (\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4) \notin Q_3(D_4) \) (why?) and it ensues that the atom cannot be removed from \( Q_3 \).
Integrated Exercise

Solution: removal of redundant joins

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

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

\[ D_5 = \{ \text{Emp}(\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4), \text{Emp}(\hat{a}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4), \]
\[ \text{Dept}(\hat{b}_2, \hat{d}_2, 1, \hat{d}_4), \text{Finance}(\hat{b}_2, 100, \hat{f}_3, \hat{f}_4) \} \]

Observe that \((\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4) \notin Q_3(D_5) \) (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:

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

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:

\[
\pi_{E.eid, E.did, E.sal, E.hobby} \\
\sigma_{F.budget=100 \land E_1.did=D_1.did \land E_1.did=F.did \land D_1.floor=1} \\
\sigma_{D_2.did=E_1.did \land D_2.dname='CID' \land E_1.eid=E.eid} \\
(\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{D_1}(\text{Dept})) \\
\times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance}))
\]
Integrated Exercise

Solution: application of the algebraic laws

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

\[
\pi_{\text{MAX}(E.\text{sal})} \gamma E.\text{hobby}, \pi_{\text{MAX}(E.\text{sal})} \pi E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby} \\
\sigma F.\text{budget}=100 \land E_1.\text{did}=D_1.\text{did} \land E_1.\text{did}=F.\text{did} \land D_1.\text{floor}=1 \\
\sigma D_2.\text{did}=E_1.\text{did} \land D_2.\text{name}=\text{CID} \land E_1.\text{eid}=E.\text{eid} \\
(\rho_E(\text{Emp}) \times \rho_{E_1}(\text{Emp}) \times \rho_{D_1}(\text{Dept}) \times \rho_{D_2}(\text{Dept}) \times \rho_F(\text{Finance}))
\]

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

\[
\pi_{\text{MAX}(E.\text{sal})} \gamma E.\text{hobby}, \pi_{\text{MAX}(E.\text{sal})} \pi E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby} \sigma E_1.\text{eid}=E.\text{eid} \\
(\rho_E(\text{Emp}) \times \sigma_{E_1.\text{did}=F.\text{did}}(\sigma_{D_2.\text{did}=E_1.\text{did}} \\
(\sigma_{E_1.\text{did}=D_1.\text{did}}(\rho_{E_1}(\text{Emp}) \times \sigma_{D_1.\text{floor}=1}(\rho_{D_1}(\text{Dept}))) \\
\times \sigma_{D_2.\text{name}=\text{CID}'} \rho_{D_2}(\text{Dept})) \times \sigma_{F.\text{budget}=100}(\rho_F(\text{Finance}))))
\]
Solution (continued)

Recognizing joins:

\[ \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{eid}, E.\text{did}, E.\text{sal}, E.\text{hobby}} \rho_{E}(\text{Emp}) \times ((\rho_{E_1}(\text{Emp}) \times \sigma_{D_1.\text{floor}=1} \rho_{D_1}(\text{Dept})) \times \sigma_{D_2.\text{dname}=\text{'CID'}(\rho_{D_2}(\text{Dept}))}) \times \sigma_{F.\text{budget}=100}(\rho_{F}(\text{Finance}))) \]

Pushing the projections:

\[ \pi_{\text{MAX}(E.\text{sal})} \gamma_{E.\text{hobby}, \text{MAX}(E.\text{sal})} \pi_{E.\text{sal}, E.\text{hobby}}(\pi_{E.\text{eid}, E.\text{sal}, E.\text{hobby}} \rho_{E}(\text{Emp}) \times \pi_{E_1.\text{eid}}((\pi_{E_1.\text{did}, E_1.\text{did}}(\pi_{E_1.\text{did}, E_1.\text{did}}(\rho_{E_1}(\text{Emp})) \times \pi_{D_1.\text{did}}(\sigma_{D_1.\text{floor}=1}(\rho_{D_1}(\text{Dept})))) \times \pi_{D_2.\text{did}}(\sigma_{D_2.\text{dname}=\text{'CID'}(\rho_{D_2}(\text{Dept})))) \times \pi_{F.\text{did}}(\sigma_{F.\text{budget}=100}(\rho_{F}(\text{Finance})))) \times \rho_{F}(\text{Finance})))) \]