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:

$$\boldsymbol{\pi}_{\texttt{R.rid},\texttt{R.type},\texttt{R.price}}\ \boldsymbol{\sigma}_{\texttt{R.rid}=\texttt{B.rid}\ \wedge\ \texttt{B.hid}=\texttt{H.hid}\ \wedge\ \texttt{H.name}='\texttt{Hilton}'\ \wedge\ \texttt{R.price}>100} \\ (\boldsymbol{\rho}_{\texttt{R}}(\texttt{Room}) \times \boldsymbol{\rho}_{\texttt{H}}(\texttt{Hotel}) \times \boldsymbol{\rho}_{\texttt{B}}(\texttt{Booking}))$$

Optimization of Logical Queries

Solution

The translation gives us the following relational algebra expression:

$$\pi_{\texttt{R.rid},\texttt{R.type},\texttt{R.price}}\ \sigma_{\texttt{R.rid}=\texttt{B.rid}\ \wedge\ \texttt{B.hid}=\texttt{H.hid}\ \wedge\ \texttt{H.name}='\texttt{Hilton}'\ \wedge\ \texttt{R.price}>100} \left(\rho_{\texttt{R}}(\texttt{Room}) \times \rho_{\texttt{H}}(\texttt{Hotel}) \times \rho_{\texttt{B}}(\texttt{Booking})\right)$$

First, we split the selections:

$$\pi_{\texttt{R.rid},\texttt{R.type},\texttt{R.price}}\sigma_{\texttt{R.rid}=\texttt{B.rid}}\sigma_{\texttt{B.hid}=\texttt{H.hid}}\sigma_{\texttt{H.name}='\texttt{Hilton}'}\sigma_{\texttt{R.price}>100} \left(\rho_{\texttt{R}}(\texttt{Room}) \times \rho_{\texttt{H}}(\texttt{Hotel}) \times \rho_{\texttt{B}}(\texttt{Booking})\right)$$

And we push the selections:

$$\pi_{\texttt{R.rid},\texttt{R.type},\texttt{R.price}}\sigma_{\texttt{R.rid}=\texttt{B.rid}}\Big(\sigma_{\texttt{R.price}>100}\ \rho_{\texttt{R}}(\texttt{Room}) \times \sigma_{\texttt{B.hid}=\texttt{H.hid}}\left(\sigma_{\texttt{H.name}='\texttt{Hilton}'}\ \rho_{\texttt{H}}(\texttt{Hotel}) \times \rho_{\texttt{B}}(\texttt{Booking})\right)\Big)$$

Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

$$\boldsymbol{\pi}_{\texttt{R.rid,R.type,R.price}}\big(\boldsymbol{\sigma}_{\texttt{R.price>100}}\,\boldsymbol{\rho}_{\texttt{R}}(\texttt{Room}) \underset{\texttt{R.rid=B.rid}}{\bowtie} \big(\boldsymbol{\sigma}_{\texttt{H.name='Hilton'}}\,\boldsymbol{\rho}_{\texttt{H}}(\texttt{Hotel}) \underset{\texttt{B.hid=H.hid}}{\bowtie} \boldsymbol{\rho}_{\texttt{B}}(\texttt{Booking})\big)\big)$$

Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

$$\pi_{\texttt{R.rid,R.type,R.price}}\Big(\sigma_{\texttt{R.price>100}}\,\rho_{\texttt{R}}(\texttt{Room}) \underset{\texttt{R.rid=B.rid}}{\bowtie} \big(\sigma_{\texttt{H.name='Hilton'}}\,\rho_{\texttt{H}}(\texttt{Hotel}) \underset{\texttt{B.hid=H.hid}}{\bowtie} \rho_{\texttt{B}}(\texttt{Booking})\big)\Big)$$

Finally, the projections are pushed:

$$\pi_{\texttt{R.rid,R.type,R.price}}\Big(\pi_{\texttt{R.rid,R.type,R.price}}\,\sigma_{\texttt{R.price>100}}\,\rho_{\texttt{R}}(\texttt{Room}) \underset{\texttt{R.rid=B.rid}}{\bowtie} \pi_{\texttt{B.rid}}\big(\pi_{\texttt{H.hid}}\,\sigma_{\texttt{H.name='Hilton'}}\,\rho_{\texttt{H}}(\texttt{Hotel}) \underset{\texttt{B.hid=H.hid}}{\bowtie} \pi_{\texttt{B.hid,B.rid}}\,\rho_{\texttt{B}}(\texttt{Booking})\big)\Big)$$

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:

$$\boldsymbol{\pi}_{Q_1.A,Q_3.B}\boldsymbol{\sigma}_{Q_1.B=Q2.A\wedge Q_2.B=Q_3.A}(\boldsymbol{\rho}_{Q_1}(Q)\times\boldsymbol{\rho}_{Q_2}(Q)\times\boldsymbol{\rho}_{Q_3}(Q))$$

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:

```
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:

$$\boldsymbol{\pi}_{Q_1.A, Q_3.B} \boldsymbol{\sigma}_{Q_1.B = Q2.A \wedge Q_2.B = Q_3.A}(\boldsymbol{\rho}_{Q_1}(Q) \times \boldsymbol{\rho}_{Q_2}(Q) \times \boldsymbol{\rho}_{Q_3}(Q))$$

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:

$$\boldsymbol{\pi}_{Q_1.A,Q_3.B}\boldsymbol{\sigma}_{Q_1.B=Q2.A\wedge Q_2.B=Q_3.A}(\boldsymbol{\rho}_{Q_1}(Q)\times\boldsymbol{\rho}_{Q_2}(Q)\times\boldsymbol{\rho}_{Q_3}(Q))$$

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:

$$\boldsymbol{\pi}_{Q_1.A, Q_3.B} \boldsymbol{\sigma}_{Q_1.B = Q2.A \wedge Q_2.B = Q_3.A}(\boldsymbol{\rho}_{Q_1}(Q) \times \boldsymbol{\rho}_{Q_2}(Q) \times \boldsymbol{\rho}_{Q_3}(Q))$$

(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:

$$\boldsymbol{\pi}_{C,G} \boldsymbol{\sigma}_{C=F}\, \boldsymbol{\sigma}_{C=D}\, \boldsymbol{\sigma}_{E=3}\, (S \times T \times U)$$

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?

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

Solution

The select-project-join expression is:

$$\boldsymbol{\pi}_{C,G} \boldsymbol{\sigma}_{C=F}\, \boldsymbol{\sigma}_{C=D}\, \boldsymbol{\sigma}_{E=3}\, (S \times T \times U)$$

The corresponding SQL query is:

```
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 $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}(\mathrm{body}) \subseteq D$
- $Q(D) = \{\boldsymbol{\sigma}(\mathrm{head}) \mid \boldsymbol{\sigma}$ a matching of Q 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. 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:

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

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:

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

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 mach $Q(a, b)$ 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 mach $Q(b, c)$ we would need $c \mapsto \dot{y}$.

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:

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

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:

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

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

$$[x \to \dot{x}, y \to \dot{y}, a \to \dot{a}, b \to 1, c \to \dot{b}]$$

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:

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

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

  $$[x \to \dot{x}, y \to \dot{y}, w \to \dot{z}, z \to \dot{z}]$$

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:

$$\boldsymbol{\pi}_{E_1.\texttt{eid},E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}\, \boldsymbol{\sigma}_{F.\texttt{budget}=100 \wedge E_1.\texttt{did}=D_1.\texttt{did} \wedge E_1.\texttt{did}=F.\texttt{did}}$$
$$\boldsymbol{\sigma}_{E_2.\texttt{did}=D_2.\texttt{did} \wedge E_2.\texttt{did}=F.\texttt{did} \wedge D_1.\texttt{floor}=1 \wedge D_2.\texttt{dname}='\texttt{CID}' \wedge E_1.\texttt{eid}=E.\texttt{eid}}$$
$$(\boldsymbol{\rho}_E(\texttt{Emp}) \times \boldsymbol{\rho}_{E_1}(\texttt{Emp}) \times \boldsymbol{\rho}_{E_2}(\texttt{Emp})$$
$$\times\, \boldsymbol{\rho}_{D_1}(\texttt{Dept}) \times \boldsymbol{\rho}_{D_2}(\texttt{Dept}) \times \boldsymbol{\rho}_F(\texttt{Finance}))$$

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

$$e_2 := \boldsymbol{\rho}_E(\texttt{Emp})$$

The decorrelation of the subquery gives:

$$e_3 := \hat{e_2} \bowtie \boldsymbol{\pi}_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}(e_1)$$

Notice that $\hat{e_2}$ is empty! Therefore, the translation of the complete query is:

$$e_4 := \boldsymbol{\pi}_{\texttt{MAX}(E.\texttt{sal})}\, \boldsymbol{\gamma}_{E.\texttt{hobby},\texttt{MAX}(E.\texttt{sal})}\, \boldsymbol{\pi}_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}(e_1)$$

Integrated Exercise

Solution: translation into the relational algebra

This leads to (after merging projections):

$$\boldsymbol{\pi}_{\texttt{MAX}(E.\texttt{sal})}\, \boldsymbol{\gamma}_{E.\texttt{hobby},\texttt{MAX}(E.\texttt{sal})}$$
$$\boldsymbol{\pi}_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}\, \boldsymbol{\sigma}_{F.\texttt{budget}=100 \wedge E_1.\texttt{did}=D_1.\texttt{did} \wedge E_1.\texttt{did}=F.\texttt{did}}$$
$$\boldsymbol{\sigma}_{E_2.\texttt{did}=D_2.\texttt{did} \wedge E_2.\texttt{did}=F.\texttt{did} \wedge D_1.\texttt{floor}=1 \wedge D_2.\texttt{dname}='\texttt{CID}' \wedge E_1.\texttt{eid}=E.\texttt{eid}}$$
$$(\boldsymbol{\rho}_E(\texttt{Emp}) \times \boldsymbol{\rho}_{E_1}(\texttt{Emp}) \times \boldsymbol{\rho}_{E_2}(\texttt{Emp})$$
$$\times\, \boldsymbol{\rho}_{D_1}(\texttt{Dept}) \times \boldsymbol{\rho}_{D_2}(\texttt{Dept}) \times \boldsymbol{\rho}_F(\texttt{Finance}))$$

Integrated Exercise

Solution: translation into the relational algebra

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

$$\boldsymbol{\pi}_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}\,\boldsymbol{\sigma}_{F.\texttt{budget}=100\wedge E_1.\texttt{did}=D_1.\texttt{did}\wedge E_1.\texttt{did}=F.\texttt{did}}$$
$$\boldsymbol{\sigma}_{E_2.\texttt{did}=D_2.\texttt{did}\wedge E_2.\texttt{did}=F.\texttt{did}\wedge D_1.\texttt{floor}=1\wedge D_2.\texttt{dname}='\texttt{CID}'\wedge E_1.\texttt{eid}=E.\texttt{eid}}$$
$$(\boldsymbol{\rho}_E(\texttt{Emp})\times\boldsymbol{\rho}_{E_1}(\texttt{Emp})\times\boldsymbol{\rho}_{E_2}(\texttt{Emp})$$
$$\times\,\boldsymbol{\rho}_{D_1}(\texttt{Dept})\times\boldsymbol{\rho}_{D_2}(\texttt{Dept})\times\boldsymbol{\rho}_F(\texttt{Finance}))$$

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

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

Integrated Exercise

Solution: removal of redundant joins

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

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

Integrated Exercise

Solution: removal of redundant joins

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

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

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

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

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

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

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

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 $\texttt{Emp}(a_1, b_2, b_3, b_4)$. 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

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

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

$$D_4 = \{\texttt{Emp}(\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4), \texttt{Emp}(\dot{a}_1, \dot{b}_2, \dot{b}_3, \dot{b}_4),$$
$$\texttt{Dept}(\dot{b}_2, '\texttt{CID}', \dot{e}_3, \dot{e}_4), \texttt{Finance}(\dot{b}_2, 100, \dot{f}_3, \dot{f}_4)\}$$

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

Integrated Exercise

Solution: removal of redundant joins

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

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

  $$D_5 = \{\texttt{Emp}(\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{a}_4), \texttt{Emp}(\dot{a}_1, \dot{b}_2, \dot{b}_3, \dot{b}_4), \\ \texttt{Dept}(\dot{b}_2, \dot{d}_2, 1, \dot{d}_4), \texttt{Finance}(\dot{b}_2, 100, \dot{f}_3, \dot{f}_4)\}$$

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

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

And $\rho_{E_2}(\texttt{Emp})$ 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:

$$\pi_{E.\texttt{eid}, E.\texttt{did}, E.\texttt{sal}, E.\texttt{hobby}} \\ \sigma_{F.\texttt{budget}=100 \wedge E_1.\texttt{did}=D_1.\texttt{did} \wedge E_1.\texttt{did}=F.\texttt{did} \wedge D_1.\texttt{floor}=1} \\ \sigma_{D_2.\texttt{did}=E_1.\texttt{did} \wedge D_2.\texttt{dname}='\texttt{CID}' \wedge E_1.\texttt{eid}=E.\texttt{eid}} \\ (\boldsymbol{\rho}_E(\texttt{Emp}) \times \boldsymbol{\rho}_{E_1}(\texttt{Emp}) \times \boldsymbol{\rho}_{D_1}(\texttt{Dept}) \\ \times \boldsymbol{\rho}_{D_2}(\texttt{Dept}) \times \boldsymbol{\rho}_F(\texttt{Finance}))$$

Integrated Exercise

Solution: application of the algebraic laws

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

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

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

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

Integrated Exercise

Solution (continued)

Recognizing joins:

$$\pi_{\texttt{MAX}(E.\texttt{sal})}\gamma_{E.\texttt{hobby},\texttt{MAX}(E.\texttt{sal})}\pi_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}$$
$$\rho_E(\texttt{Emp}) \underset{E_1.\texttt{eid}=E.\texttt{eid}}{\bowtie} (((\rho_{E_1}(\texttt{Emp}) \underset{E_1.\texttt{did}=D_1.\texttt{did}}{\bowtie} \sigma_{D_1.\texttt{floor}=1}\rho_{D_1}(\texttt{Dept}))$$
$$\underset{E_1.\texttt{did}=D_2.\texttt{did}}{\bowtie} \sigma_{D_2.\texttt{dname}='\texttt{CID}'}(\rho_{D_2}(\texttt{Dept}))) \underset{E_1.\texttt{did}=F.\texttt{did}}{\bowtie} \sigma_{F.\texttt{budget}=100}(\rho_F(\texttt{Finance})))$$

Pushing the projections:

$$\pi_{\texttt{MAX}(E.\texttt{sal})}\gamma_{E.\texttt{hobby},\texttt{MAX}(E.\texttt{sal})}\pi_{E.\texttt{sal},E.\texttt{hobby}}(\pi_{E.\texttt{eid},E.\texttt{sal},E.\texttt{hobby}}\rho_E(\texttt{Emp})$$
$$\underset{E_1.\texttt{eid}=E.\texttt{eid}}{\bowtie} \pi_{E_1.\texttt{eid}}(((\pi_{E_1.\texttt{did},E_1.\texttt{eid}}(\pi_{E_1.\texttt{did},E_1.\texttt{eid}}\rho_{E_1}(\texttt{Emp})$$
$$\underset{E_1.\texttt{did}=D_1.\texttt{did}}{\bowtie} \pi_{D_1.\texttt{did}}\sigma_{D_1.\texttt{floor}=1}(\rho_{D_1}(\texttt{Dept})))$$
$$\underset{E_1.\texttt{did}=D_2.\texttt{did}}{\bowtie} \pi_{D_2.\texttt{did}}\sigma_{D_2.\texttt{dname}='\texttt{CID}'}\rho_{D_2}(\texttt{Dept})))$$
$$\underset{E_1.\texttt{did}=F.\texttt{did}}{\bowtie} \pi_{F.\texttt{did}}\sigma_{F.\texttt{budget}=100}(\rho_F(\texttt{Finance}))))$$