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:

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

First, we split the selections:

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

And we push the selections:

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

Optimization of logical queries

Solution (continued)

Then, the joins are recognized:

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

Finally, the projections are pushed:

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

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:

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

Which corresponds to the following conjunctive query:

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

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?

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

Solution

The select-project-join expression is:

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

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 $\boldsymbol{\sigma}$ such that $\boldsymbol{\sigma}(\text{body}) \subseteq D$
- $Q(D) = \{\boldsymbol{\sigma}(\text{head}) \mid \boldsymbol{\sigma}$ 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 $Q_3 \subseteq Q_2$?

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:

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

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

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

Therefore: $Q_3 \subseteq Q_2$.

Optimization of conjunctive queries

Task

Optimize the following conjunctive query

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

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 verified). 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 $P \subseteq Q$:

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

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?)
- 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):

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

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

$$\boldsymbol{\pi}_{E.\texttt{eid},E.\texttt{did},E.\texttt{sal},E.\texttt{hobby}}(e_1).$$

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?)
- 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:

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

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

Integrated Exercise

Solution: removal of redundant joins

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

- 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:

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

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

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

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

Integrated Exercise

Solution: removal of redundant joins

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

- 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:

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

  Indeed $(a_1, a_2, a_3, a_4) \notin Q_3(D_5)$ (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:

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

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:

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

Integrated Exercise

Solution: application of the algebraic laws

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

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

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

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

Integrated Exercise

Solution (continued)

Recognizing joins:

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

Pushing the projections:

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