Michail Kazimianec
Werner Nutt
Damiano Somenzi

Relational Algebra: Sample Solutions

Note that the solutions given here are *samples,* i.e., there may be many more ways to express these queries in relational algebra.

1. Write queries in relational algebra

Write the following queries in relational algebra.

1. *"Find the names of suppliers who supply some red part."*

   $$\pi_{\mathsf{sname}}((\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \bowtie \mathsf{Supplier}))$$

   Since there is not subscript under the joins, the joins are natural joins, i.e., the common attributes are equated.

2. *"Find the IDs of suppliers who supply some red or green part."*

   $$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}' \vee \mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog})$$

   An equivalent formulation uses the union operator

   $$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog} \cup \sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}).$$

   The latter version can be refined by pushing the projection through the union:

   $$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \cup \pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}).$$

3. *"Find the IDs of suppliers who supply some red part or are based at 21 George Street."*

   $$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \cup \pi_{\mathsf{sid}}(\sigma_{\mathsf{address}='\mathtt{21G.S.}'}(\mathsf{Supplier})).$$

4. *"Find the names of suppliers who supply some red part or are based at 21 George Street."*

   $$\pi_{\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog} \bowtie \mathsf{Supplier}) \cup \pi_{\mathsf{sname}}(\sigma_{\mathsf{address}='\mathtt{21G.S.}'}(\mathsf{Supplier})).$$

   Alternatively, we can pull the projection on sname to the top level.

   $$\pi_{\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog} \bowtie \mathsf{Supplier} \cup \sigma_{\mathsf{address}='\mathtt{21G.S.}'}(\mathsf{Supplier})).$$

5. *"Find the IDs of suppliers who supply some red part and some green part."*

$$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \cap \pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog})$$

Alternatively, we can replace the intersection with a join:

$$\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}) \bowtie \pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \mathsf{Catalog}).$$

6. *"Find pairs of* sid*s such that the supplier with the first* sid *charges more for some part than the supplier with the second* sid*."*

First, by creating temporary copies we introduce two versions of Catalog:

$$\begin{aligned} \mathsf{Cat1} &\leftarrow \mathsf{Catalog} \\ \mathsf{Cat2} &\leftarrow \mathsf{Catalog}. \end{aligned}$$

Then we join the two versions and take the projection:

$$\pi_{\mathsf{Cat1.sid},\mathsf{Cat2.sid}}(\mathsf{Cat1} \bowtie_{\mathsf{Cat1.pid}=\mathsf{Cat2.pid} \wedge \mathsf{Cat1.cost}>\mathsf{Cat2.cost}} \mathsf{Cat2})$$

Note that we have to qualify the attributes by the relation name because both relations have attributes with the same names.

7. *"Find the IDs of suppliers who supply only red parts."*

$$\pi_{\mathsf{sid}}(\mathsf{Supplier}) \setminus \pi_{\mathsf{sid}}(\mathsf{Catalog} \bowtie \sigma_{\mathsf{colour}\neq'\mathtt{red}'}(\mathsf{Part})).$$

8. *"Find the IDs of suppliers who supply every part."*

First, we observe that the projection of Catalog on the attributes sid and pid,

$$\pi_{\mathsf{sid},\mathsf{pid}}(\mathsf{Catalog})$$

contains all pairs of suppliers and the parts they supply, expressed by sids and pids.

The Cartesian product

$$\pi_{\mathsf{sid}}(\mathsf{Catalog}) \times \pi_{\mathsf{pid}}(\mathsf{Part})$$

contains all possible combinations of (1) the suppliers that supply something with (2) all the products.

If we take the set-theoretic difference of the second and the first relation,

$$\pi_{\mathsf{sid}}(\mathsf{Catalog}) \times \pi_{\mathsf{pid}}(\mathsf{Part}) \setminus \pi_{\mathsf{sid},\mathsf{pid}}(\mathsf{Catalog}),$$

we obtain those combinations of suppliers and parts where the supplier does not deliver the part. Let us store the difference in the temporary relation Temp1:

$$\mathsf{Temp1} \leftarrow \pi_{\mathsf{sid}}(\mathsf{Catalog}) \times \pi_{\mathsf{pid}}(\mathsf{Part}) \setminus \pi_{\mathsf{sid},\mathsf{pid}}(\mathsf{Catalog}).$$

Now, $\pi_{\mathsf{sid}}(\mathsf{Temp1})$, the projection of this relation onto the attribute sid, gives us the suppliers that do *not* supply some part, i.e., some part is missing in their range of products.

However, $\pi_{\mathsf{sid}}(\mathsf{Catalog})$, the projection of Catalog onto sid gives us the suppliers that *do* supply *at least* some part.

Therefore,

$$\pi_{\mathsf{sid}}(\mathsf{Catalog}) \setminus \pi_{\mathsf{sid}}(\mathsf{Temp1}),$$

the difference of the latter and the former, gives us the sid's of suppliers that

- supply something, and
- have no part missing in their range of products.

That means, it gives us the sid's of the suppliers that supply all parts.

2. Queries in relational algebra, what do they mean?

For each of the following relational algebra queries, say what they mean.

1. $\pi_{\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier})$

   "Find the names of suppliers supplying some red part for less than 100 Quid."

2. $\pi_{\mathsf{sname}}(\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog})) \bowtie \mathsf{Supplier})$

   "Find the names of suppliers supplying some red part for less than 100 Quid." This query is an optimized version of the previous one: from the join of Part and Catalog it retains only the projection onto sid. Then it uses sid to retrieve the snames of the suppliers.

3. $\pi_{\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier}) \cap \pi_{\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier})$

   "Find the names of suppliers such that there is a supplier with that name supplying some red part for less than 100 Quid and a supplier with that name supplying some green part for less than 100 Quid."

   The English rendering of the query sounds unnecessarily complicated. However, as the following example shows, the complicated formulation is needed to express the meaning of the query.

   Suppose there are two distinct suppliers, both with the name `'Smith'`, such that the first Smith supplies some red part for less than 100, and the second Smith supplies some green part for less than 100. Then the name `'Smith'` is returned by this query, even if the first Smith doesn't supply a green part, and the second doesn't supply a red part.

4. $\pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier}) \cap \pi_{\mathsf{sid}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier})$

   "Find the sids *of suppliers supplying some red part for less than 100 Quid and some green part for less than 100 Quid, and return only* sid*'s of suppliers recorded in the table* Supplier*."*

If there is a foreign key constraint on $\mathsf{Catalog}(\mathsf{sid})$, then the join with $\mathsf{Supplier}$ is superfluous. (Some query optimisers find out such redundant joins. For instance, Microsoft's SQL Server eliminates joins that are redundant because of foreign key constraints, and it eliminates parts of the query that are inconsistent.)

5. $\pi_{\mathsf{sname}}(\pi_{\mathsf{sid},\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{red}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier}) \cap \pi_{\mathsf{sid},\mathsf{sname}}(\sigma_{\mathsf{colour}='\mathtt{green}'}(\mathsf{Part}) \bowtie \sigma_{\mathsf{cost}<100}(\mathsf{Catalog}) \bowtie \mathsf{Supplier}))$

   "Find the names of suppliers supplying some red part for less than 100 Quid and some green part for less than 100 Quid."

   This is the query that intuitively makes most sense.