Weekly exercises IN3020&4020 (2026)
Week 3: Relational Algebra

1 Review

1. What is the domain of the Relational Algebra (i.e., what type of arguments take relational algebra operators)?

   Solution: The domain is the set of all relations (not schemas, not relation names, but relations themselves; however, we may sometimes write just relation names instead of relations in relational algebra expressions for brevity in the case when the relation state—that is, the tuples—are given somewhere else or not important).

2. List the operations of the core relational algebra (our version of) and briefly state the purpose of each.

   Solution: Core Relational Algebra consists of the following six operations:

   - selection $\sigma_C(R)$: select tuples from relation R that satisfy condition C,
   - projection $\pi_{Atts}(R)$: select attributes (columns) $Atts$,
   - Cartesian product $R \times S$: combine tuples from one relation with the tuples of another relation in all possible ways,
   - union $R \cup S$: set-theoretic union,
   - difference $R \setminus S$: set-theoretic difference,
   - renaming ρ_N: rename attributes as specified in list N.

3. When is each of the core operations defined?

   Solution:

   - Selection $\sigma_C(R)$ is defined when C mentions only attribute names of R and all operations in C are well-typed.

- Projection $\pi_{Atts}(R)$ is defined when $Atts$ mentions only attribute names of R.
- Cartesian product $R \times S$ is defined when R and S have no common attribute names.
- Union $R \cup S$ and difference $R \setminus S$ are defined when arguments R and S have the same signature (i.e., the attributes names with the same datatypes).
- Renaming ρ_N is defined when N is a list $A_1 \to A'_1, \ldots, A_n \to A'_n$ with $A_1, \ldots, A_n, A'_1, \ldots, A'_n$ all different attribute names such that $A_1, \ldots, A_n$ are attribute names of R and $A'_1, \ldots, A'_n$ are not.

4. What is a commutative operation? Give an example of a commutative operation from relational algebra.

   Solution: A binary operation (i.e., an operation with two arguments) is commutative if changing the order of the arguments does not change the result. For example, union and Cartesian product are commutative operations: $R \cup S = S \cup R$ and $R \times S = S \times R$ for all relations R, S (such that the operation are applicable).

5. What is an associative operation? Give an example of an associate operation from relational algebra.

   Solution: A binary operation is associative if, when applying the operation on three or more elements, the grouping (where we put the parenthesis) does not affect the result. Union and Cartesian product are associative: $(R \cup S) \cup T = R \cup (S \cup T)$ and $(R \times S) \times T = R \times (S \times T)$ for all relations R, S, T with the same signature.

6. What is a unary idempotent operation? What is a binary idempotent operation? Give one example of each from relational algebra.

   Solution: A unary idempotent operation is an operation that gives the same result if applied multiple times (on the same input). A binary operation *op* is idempotent if $e \; op \; e = e$. For example, selection and projection are unary idempotent operations: $\pi_{Attr}(\pi_{Attr}(R))$ for every relation R and some of its attributes $Attr$, while union is a binary idempotent operation: $R \cup R = R$ for every relation R.

2 Relation algebra computations

Consider the following two relations:

R:

A	B
1	2
3	4
4	5

S:

C	D
1	2
3	5
3	6
4	5
5	6

Using our core Relational Algebra for sets extended with natural join compute the following:

1. $\pi_A(R)$

   Solution: $\{\langle 1\rangle, \langle 3\rangle, \langle 4\rangle\}$

2. $\pi_B(S)$

   Solution: $\pi_B(S)$ is not defined because B is not an attribute in S (note that other versions of the Relational Algebra may have this different)

3. $\pi_C(S)$

   Solution: $\{\langle 1\rangle, \langle 3\rangle, \langle 4\rangle, \langle 5\rangle\}$ (note that the duplicate of 3 is removed).

4. $\sigma_{A=1}(R)$

   Solution: $\{\langle 1, 2\rangle\}$

5. $R \bowtie S$

   Solution: $\{\langle 1, 2, 1, 2\rangle, \langle 1, 2, 3, 5\rangle, \langle 1, 2, 3, 6\rangle, \ldots, \langle 4, 5, 5, 6\rangle\}$ (i.e., all quadruples of values such that the first two are a pair from R and the last two are a pair. from S). Note that in this case $R \bowtie S \equiv R \times S$ because R and S have no common attributes.

6. $(\rho_{B\to C}(R)) \bowtie S$

   Solution: $\{\langle 3, 4, 5\rangle, \langle 4, 5, 6\rangle\}$

7. $\sigma_{D=5}(\rho_{A\to C}(R) \bowtie S)$

   Solution: $\{\langle 3, 4, 5\rangle, \langle 4, 5, 5\rangle\}$

3 (*) Correctness of Relational Algebra expressions

Assuming the schemas in the previous section, which of the following expressions are valid (under our definitions) and which are not? Explain your answer.

1. $R \times R$

Solution: Not valid, because the arguments of $\times$ must have disjoint sets of attributes

2. $R \cup (\rho_{C \to B, D \to A}(S))$

   Solution: Valid, because the order of attributes in relation schemas are immaterial and hence both arguments of $\cup$ have the same signature as required.

3. $\sigma_{A > \texttt{ALL}(\pi_D(S))}(R)$

   Solution: Not valid, because `ALL` is not an operation over attributes names and values, as required in our definition (in particular, contrary to SQL, Relational Algebra does not allow for Relational Algebra expressions as sub-expressions in selection conditions).

4 Relation algebra expression

Consider four relations with the following schemas (the domains and the meaning of the attributes are as expected):

$$\begin{aligned} &\mathit{Product}\,[\mathit{manufacturer}, \mathit{model}, \mathit{type}] \\ &\mathit{PC}\,[\mathit{model}, \mathit{speed}, \mathit{RAM}, \mathit{HDD}, \mathit{price}] \\ &\mathit{Laptop}\,[\mathit{model}, \mathit{speed}, \mathit{RAM}, \mathit{HDD}, \mathit{screen}, \mathit{price}] \\ &\mathit{Printer}\,[\mathit{model}, \mathit{colour}, \mathit{type}, \mathit{price}] \end{aligned}$$

Express the following queries as relational algebra expressions using relational algebra operators σ, π, $\times$, $\cup$, $\setminus$, $\cap$, ρ, and $\bowtie$.

1. The PC models that have speed at least 2.50.

   Solution: $\pi_{\mathit{model}}\,(\sigma_{\mathit{speed} \geq 2.50}\,(\mathit{PC}))$

2. The manufacturers that sell printers, but not PCs.

   Solution:

$$\pi_{\mathit{manufacturer}}\left(\sigma_{\mathit{type}='\mathit{printer}'}\,(\mathit{Product})\right) \setminus \pi_{\mathit{manufacturer}}\left(\sigma_{\mathit{type}='\mathit{PC}'}\,(\mathit{Product})\right)$$

3. The manufacturers that make laptops with a HDD of at least 500 GB.

   Solution: $\pi_{\mathit{manufacturer}}\,(\sigma_{\mathit{HDD} \geq 500}\,(\mathit{Laptop}) \bowtie \mathit{Product})$

4. The model numbers and prices of all products made by manufacturer called HP.

   Solution:
   $\pi_{\mathit{model},\,\mathit{price}}\,(\sigma_{\mathit{manufacturer}='\mathit{HP}'}\,(\mathit{PC} \bowtie \mathit{Product}))$

$\cup$

$\pi_{model,\, price}\,(\sigma_{manufacturer='HP'}\,(Laptop \bowtie Product))$

$\cup$

$\pi_{model,\, price}\,(\sigma_{manufacturer='HP'}\,((\rho_{type \to ptype}\,(Printer)) \bowtie Product))$

Note that we need to rename *type* in the printer-part, because, if we do not rename, the natural join will also match *type* from *Products* with *type* in *Printer*.