RRA: A RESHAPED RELATIONAL ALGEBRA

by

V.M. Markowitz and Y. Raz

Technical Report #278

June 1983
ABSTRACT

The Relational Model has been acknowledged for its simple, well defined structural and manipulative parts. Initially limited in its capability to capture relevant parts of the real world semantics, the model underwent changes which have brought its structural part closer to the natural way of perceiving the information. Our objective is to propose a reshaping of the Relational Algebra in order to make it take advantage of such an enriched structure and draw closer to the natural way of communication, the natural language.

The Reshaped Relational Algebra (RRA) attempts to be more appealing and manageable than the Relational Algebra by drawing on analogies with natural language constructs, that is, the construction of complex algebraic expressions is made to follow patterns of natural language sentence combination.
When the Relational Model (RM) was proposed by Codd, it was intended to meet two main objectives [3]: data independence, by providing a clear boundary between the logical and physical aspects of database management; and communicability, by keeping the model simple enough so that the users could easily understand, use and communicate with one another about the data. The second objective was only partially fulfilled by the model. Relational query languages, although higher-level and easier to use than other query languages, have been found hard to manage by casual users; thus, for instance, comprehension difficulties, for some of the relational query languages, have been pointed out by human-factors studies such as [8]. The structural flatness of the initial form of the RM was at fault generally. It favored semantic misunderstandings and the expression of syntactically correct queries that were not reflecting the user's intended meaning. Moreover the user was compelled to adapt himself to a simple, but not necessarily natural, view of the real world.

A semantically richer framework for the RM structural part has been proposed in [7], the source of the following short review. The subsequent presentation of a more extended, compared to its original definition [2], Relational Algebra is also based on [7].

A relational database consists of a set of relations. Its structural definition, called schema, includes the specification of a set of domains and a collection of relation schemes.

A domain is a set of atomic elements or values. It has a name, an associated collocation of binary comparison operators including, at least, an equality test operator, and a possibly empty, collection of operations for the elements of the domain.
A relation has a name, a structural description called relation-scheme, and a value. The relation-scheme (r-scheme) is specified by a set of attribute-domain pairs:

\[ T = \{A_1: D_1, \ldots, A_n: D_n\}. \]

An attribute, \( A_i \), identifies the role played by a domain, \( D_i \), in a relation and has to be distinct in the respective r-scheme. The attribute set of a relation is denoted \( A \), that is \( A = \{A_1, \ldots, A_n\} \). The domain associated with an attribute \( a \in A \) will be denoted \( D_a \).

The relation value (r-value) of a relation is a subset of the indexed Cartesian product [7]:

\[ (A:D)^* = A_1:D_1 \times \ldots \times A_n:D_n = \{t:A \rightarrow \bigcup D_a \mid t \text{ is total and } \forall a \in A : t(a) \in D_a \}; \]

that is to say, the r-value is a set of tuples \( t = <A_1:d_1, \ldots, A_n:d_n> \) described as total functions on the set of attributes, \( A \), and having values, \( d_i \), in the associated domains. Such a tuple is an unordered set of attribute-value pairs.

Relations are manipulated by Relational Algebra (RA) operators. Let \( A \) and \( B \) be two set of attributes corresponding to the relations \( r \) and \( s \) respectively:

- \( t(A_i) \) denotes the value corresponding to the attribute \( A_i \) in tuple \( t \);
- \( t_1 \cdot t_2 \) denotes a tuple obtained by the concatenation, meaning in the present context the union, of the tuples \( t_1 \) and \( t_2 \); and
- \( t[Y] \) denotes a tuple containing components of \( t \) corresponding to the attribute set \( Y \).

The two relations above, \( r \) and \( s \), are said to be union-compatible if there is a total one-to-one correspondence between \( A \) and \( B \), such that corresponding attributes in \( A \) and \( B \) are associated with a same domain.
1. **Attribute Renaming.** Before applying any algebraic operator, it is possible to specify an attribute correspondence by renaming the attributes of the operators. The rename operation has the form: Rename \((r, M)\), where \(r\) denotes a relation and \(M\) is a one-to-one, not necessarily total, functional mapping \(\{(A_i \rightarrow C_j)\}\) with \(A_i \in A\). As a result, \(r\) will have renamed attributes, such that every image of \(A_i\) under \(M\), \(C_j\), will be associated with the domain of \(A_i\); in the tuples of \(r\) every attribute from the domain of \(M\), \(A_i\), is replaced by \(M(A_i)\). Consequently, the result of an algebraic operation will directly inherit the attributes from possibly renamed, operands, and, on its turn, may be subjected to renaming and appear as operand in another operation.

2. **Projection.** The projection of a relation \(r\) on \(Y \subseteq A\) is defined as:
\[
r[Y] = \{t[Y] | t \in r\}.
\]

3. **Cartesian Product.** Given two relations \(r\) and \(s\), such that \(A\) and \(B\) are disjoint, their Cartesian product is:
\[
r \times s = \{t | t = t_1t_2, t_1 \in r \quad \text{and} \quad t_2 \in s\}.
\]

4. **Union, Difference, Intersection.** Given two union-compatible relations \(r\) and \(s\), with the attributes of \(s\) renamed as their correspondents in \(r\), the union, difference or intersection of \(r\) and \(s\) are the known set operations involving \(r\) and \(s\) as operands. An intersecting and useful generalization of union is the bordered-union [7], which accepts operands that are not union-compatible.
Let two relations, \(r\) and \(s\), be not union-compatible and:
\[
x = A \cap B; \quad A' = A \setminus X; \quad B' = B \setminus X.
\]
The bordered-union of $r$ and $s$ is defined as:

$$\overline{r \cup s} = (r \times (B' : D^*) \cup (s \times (A' : D^*)$$

The bordered-union of $r$ and $s$ above, is the ordinary union of two union-compatible relations: one is obtained by bordering each tuple of $r$ by all the possible values for the attributes of $s$ that are not of $r$, $B'$; and the other is obtained by bordering each tuple of $s$ with all the possible values for the attributes of $r$ that are not of $s$, $A'$.

(5) **Selection.** Given a relation $r$ and $F$, a Boolean combination of atomic comparisons of the form $(A_i \theta c)$ or $(A_i \theta A_j)$, where $\theta$ is in the set of comparison operators associated with the domain of both $A_i$ and $A_j$, and $c$ is a constant, an $F$-selection applied on $r$ gives:

$$r[F] = \{ t \mid t \in r \text{ and } F \text{ is true when every } A_i \text{ in } F \text{ is replaced by } t(A_i) \}.$$  

When $F$ consists of a single atomic comparison, the selection is also called **restriction**.

(6) **Natural Join.** Given two relations $r$ and $s$, and $X = A \cap B$, their natural join is defined as:

$$r \bowtie s = \{ t \mid t \in r \times s, t_1 \in r, t_2 \in s \text{ and } t_1[x] = t_2[x] \}.$$  

(7) **$\theta$-Join.** Given two relations $r$ and $s$, such that $A$ and $B$ are disjoint, their $\theta$-join is defined as:

$$r[A_i \theta B_j]s = (r \times s)[\tilde{A}_i \theta \tilde{B}_j],$$

where $A_i \theta B_j$ complies with the above definition of $F$ (4).
(8) Generalized Division. Given two relations \( r \) and \( s \), with \( X = A \cap B \), the division of \( r \) by \( s \) is given by:

\[
r : s = \{ t \mid t = t_1 t_2, t_1 \in r[A-X], t_2 \in s[B-X] \text{ and } \forall t_3 (t_2 t_3 \in s \Rightarrow t_1 t_3 \in r) \}.
\]

(9) Aggregate Function [4]. Let \( r \) be a relation with \( A = X \cup Y \cup A_k \), such that \( X \) and \( Y \) are disjoint subsets of \( A \), none of them containing the single attribute \( A_k \). An aggregate function (af) manipulation of \( r \), \( \varphi_{af} \), results in a relation \( r' \) having the set of attributes \( A' = X \cup C_k \), where \( C_k \) is a new attribute associated with a domain containing the values of \( af \):

\[
r' = \varphi_{af} <X; A_k> \triangleq \{ t' \mid t'[X] \in r[X] \text{ and } t'(C_k) = af(\{ t \mid t \in r \text{ and } t'[X] = t[X] \}) \}.
\]

The aggregate-function operator partitions its input on the attributes \( X \), applies \( af \) to each partition and outputs the \( X \)-values, together with the \( af \)-value for, each partition, every value indexed by its corresponding attribute. The attribute subset \( X \) is called the \( af \)-component while the single attribute \( A_k \) is the \( af \)-argument.

We have seen that the domain orientation of the Relational Model defined in [7], brought its structural part closer to the natural way of perceiving the information. Accordingly, one would expect that the model's manipulative part would take advantage of such an enriched structure to draw closer to the natural way of communication, the natural language. To this effect, a reshaping of the Relational Algebra is presented below.
2. A RESHAPED RELATIONAL ALGEBRA

In the above presentation of the Relational Algebra we have presented only the elementary RA operations. It is known that RA is closed under composition, meaning that any algebraic expression can be used as operand in any other algebraic operation [7]. When embarking on the reshaping of the RA, we had in mind the construction of such complex expressions which we have intended to make follow patterns of natural language sentence composition.

A relation, \( r \), may be referenced in different places within an algebraic expression; the \( i \)-th reference to \( r \) will be denoted \( r^i_j \) and called relation-occurrence. A domain, on its turn, can be involved in several relation-occurrences. The correlation of the various appearances of a same domain is assured by correlating-symbols (c-symbols). A domain, \( D_k \), is associated in an expression, \( E \), with a set of c-symbols \( X_k = \{X^1_k, \ldots, X^n_k\} \), such that a c-symbol \( X^m_k \) correlates some \( i \) appearances of \( D_k \) as part of \( i \) different relation-occurrences within \( E \); \( i \) is called the correlating counter (c-counter) of the c-symbol \( X^m_k \). For a relation-occurrence in \( E \), \( r^i_j \), the pairing of the c-symbols with the domains is established by a one-to-one, not necessarily total, functional mapping from the attribute-set of the relation \( r \), into the union of all the c-symbol sets associated with the domains involved in \( E \); the c-symbol sets are assumed to be disjoint. Every \( r^i_j \) is associated in \( E \) with an operational-scheme (o-schéme), denoted as \( S(r^i_j) \); it consists of the set of (c-symbol: domain) pairs corresponding to \( r^i_j \) in \( E \). Remark that unlike the attribute which specifies the role played by a domain in a relation, the c-symbol has a correlational function within an expression.
The relation-occurrence is a temporary relation (t-relation) whose temporary structure is specified by the o-scheme and whose temporary r-value is obtained by applying the operator presented below.

2.1 Correlation Assessment

The correlation-assessment corresponds to the RA renaming. It establishes the r-values of the relation-occurrences involved in an algebraic expression E.

Let, $r$ be a relation having the attribute-set $A$; $A'$ be a subset of $A$; $r^j$ be a relation-occurrence, corresponding to $r$; $X_k$ be the set of c-symbols associated with $D_k$ in $E$: $\{X_k^1, \ldots, X_k^n\}$; $f$ be a one-to-one, total, functional mapping associated with $r^j$: $\{\langle A_k, X_k^j \rangle \}$, where $A_k \in A'$; and $Y$ be the range of $f$, $Y = f(A')$.

Informally, the correlation-assessment derives the r-value of $r^j$ by performing an index replacement in $r[A']$.

Given $r^j$ and its associated mapping $f$, the result of the correlation-assessment, taking $r^j$ and $f$ as arguments, is the r-value of $r^j$:

$$\{t: Y \rightarrow \bigcup_{y \in Y} D_y \mid \exists t' \in r(\forall A_k \in A': t'(A_k)) = t(A_k')\}.$$

2.2 Implicit Operations

An algebraic expression, $E$, specifies a derived, and temporary as well, relation $q$. The o-scheme associated with $q$, $S(q)$, is explicitly specified for $E$. Every (c-symbol: domain) pair of $S(q)$ has to appear in at least one of the o-schemes associated with $E$. 
Actually, \( S(q) \) asserts the set of global references in \( E \). The r-value of \( q \) is obtained as the result of evaluating \( E \). The temporary characteristic of \( q \) is manifested by its lack of an r-scheme and of a simple name; both would require an assignment operation which is beyond the scope of our reshaping.

Beside the referencing through c-symbols, some of the RRA operations imply a different kind of referencing that we shall call operational-referencing; it affects some of the elements involved in RRA selection, RRA \( \theta \)-join, RRA division and RRA aggregate-function operations (see below).

An algebraic expression, \( E \), is evaluated in several stages, where every stage covers an elementary algebraic operation. Initially, every c-counter is equal to the number of o-schemes, including \( S(q) \), in which the corresponding c-symbol appears associated with a same domain. Within an expression, any two \( t \)-relations, \( r' \) and \( r'' \), have associated a set of mutual references, \( M(r',r'') = S(r') \cap S(r'') \).

Let \( r' \) and \( r'' \) denote the operands, and \( r''' \) the intermediary result, of an elementary algebraic operation; its evaluation includes the following structural transformations:

(51) for every c-symbol, \( x \), appearing in \( M(r',r'') \):

\[
\text{c-counter}(x) := \text{c-counter}(x) - 1; \quad \text{and}
\]

(52) \( S(r') \) and \( S(r'') \) are replaced by:

\[
\hat{S}(r''') = \{ x: D \mid x: D \text{ is in } S(r') \text{ or } S(r'') \text{, such that } x \text{ is operationally referenced or c-counter}(x) > 1 \}.
\]

Thus, the o-scheme of \( r''' \) consists of those (c-symbol: domain) pairs from \( S(r') \cup S(r'') \) that still specify correlation, while the
elements of $S(r') \cup S(r'')$ not included in $S(r''')$ specify domains that have ceased to be referenced within $E$.

The evaluation of the $r$-value of $r'''$ consists of two steps:

(v1) The application of the homonym RA operator, possibly combined with an embedded-join defined below, resulting in $r$ with the o-scheme $S(r') \cup S(r'')$; and

(v2) the projection of $r$ on $S(r''')$ giving $r'''$.

The above evaluation steps are defining a single, elementary, RRA operation.

For some of the RA operators the attribute-sets of the operands have to be disjoint. This condition is loosened in RRA by allowing the set of mutual references to be non-empty for any operation. Consequently, the evaluation step (v1) go beyond applying the operator as known in RA, and comprises the following additional actions:

(a) every tuple obtained by applying the RA operator is retained only if all the labeled values, corresponding to couples of identical elements from $S(r')$ and $S(r'')$, are equal; and

(b) one labeled value in each of the above couples is subsequently suppressed from the retained tuples.

The above sequence is part of the RA natural join, therefore we shall call it embedded-join (e-join). The e-join, like the e-projection, is implied by the o-schemes of the operands of an RRA operation, and it implements the correlation requirements specified through the c-symbols. Remark that the RA natural join is a way of expressing correlation too.

The projection (v2) is implied by the form of the involved o-schemes and appears as embedded in the operation; we shall call it
embedded-projection (e-projection). In RRA the projection is left only in this implicit, o-scheme driven form.

2.3 The RRA Operators

Excepting the projection and the natural join, all the RA operators have direct correspondents in RRA. Keeping, roughly, their meaning the RRA operators are distinguished from their RA counterparts by the special significance given to the o-schemes. As shown above, the o-schemes of the t-relations involved in an RRA operation may trigger e-projections, e-joins or both, thus extending the semantic between the RA and RRA operators, namely on the role played by the o-schemes.

Let \( r', r'' \) and \( r''' \) denote the two operands and the result of an RRA operation, respectively.

Whenever \( M(r', r'') \) is non-empty, some of the RRA operations comprise the e-join defined above. At the same time, every RRA operation embeds an e-projection as shown above.

2.3.1 Cartesian Product

A non-empty \( M(r', r'') \) specifies the e-join that makes the RRA Cartesian product similar to the RA natural join; additionally, the RRA operation can include an e-projection.

2.3.2 Union, Difference, Intersection

Like in RA, the RRA union, RRA difference and RRA intersection require the union-compatibility as a prerequisite condition. Two t-relations, \( r' \) and \( r'' \), are said to be union-compatible within an expression \( E \), if \( S(r') = S(r'') = M(r', r'') \). The RRA operations differ from their RA counterparts by their potential of implying e-projections.
Notice that the RRA intersection appears as a special case of the RRA Cartesian product.

In the light of one reshaping, a useful operator would be the not-operator [7] defined as follows:

Let \( r \) be a relation with the attribute set \( A \),
\[
\neg r \triangleq (A \times D)^\ast - r.
\]
This operator is similar, in a way, to the bordered difference of [7]. Consequently, the difference of two union-compatible relations, \( r \) and \( s \), may be expressed as:
\[ r - s = r \cap (\neg s) . \]

2.3.3 \( \theta \)-join

The two operands of the \( \theta \)-comparison are operationally-referenced. A non-empty \( M(r', r'') \) means that an \( \theta \)-join is embedded in the RRA \( \theta \)-join. Let alone the \( e \)-projection, the RRA \( \theta \)-join has no direct RA counterpart.

2.3.4 Selection

All the operands of the \( \theta \)-comparisons belonging to the selectin formula are operationally-referenced. The RRA selection differs from its RA correspondent only on the \( e \)-projection possibly comprised by the RRA operation. Remark that it is possible to view the restriction as a special case of \( \theta \)-join, where the operands are an \( t \)-relation and a constant relation, or the same \( t \)-relation. We shall see in the next section that this view is supported by both the \( \theta \)-join and the restriction having the same natural language sentence combination correspondent.
2.3.5 Division

The o-scheme of a r-relation involved in an RRA division is partitioned into a division-set and a division-component, where the division-set is common to both operands of the RRA division, i.e. included in their set of mutual references. The elements of the division-component are operationally-referenced.

Let \( r' \) and \( r'' \) be the operands of an RRA division; their division set is \( D(r',r'') \subseteq M(r,r'') \), and the RA homonym of \( (v1) \) is a generalized division on \( D(r',r'') \); additionally, the RRA division could imply an e-join, specified, actually, only by \( M'(r',r'') = M(r,r'') - D(r',r'') \), and/or an e-projection. Like the RRA \( \theta \)-join, the RRA division has no direct counterpart in RA.

2.3.6 Aggregate-function

For the RRA aggregate-function (af) operations the o-scheme of the operand is partitioned into an af-component and an af-argument; all the elements of the af-component and the single element of the af-argument are operationally-referenced. The RRA aggregate-function cannot imply e-join.

An Example

We are supposing that every relation in a relational database is associated with, and described by, a simple natural language sentence consisting of a predicate and \( n \) object terms. In such a case the RRA operations correspond to rules governing the combination of these simple sentences. These correspondences has been analyzed in [6]. We shall mention them briefly during the presentation of an example.
Besides sentences associated with associations we shall encounter elementary statements of the form:

"A 0 c" or "A 0 B",

where A and B are object terms, c is a constant and 0 is a comparison operator; we shall call them restrictions.

Let \( r_1, r_2, \) and \( r_3 \) be associated with, and described by, the following sentences:

\[
\begin{align*}
(r_1) & \quad \text{'ITEM REQUESTED by DEPARTMENT in QTY.'} \\
(r_2) & \quad \text{'ITEM SUPPLIED by SUPPLIER to DEPARTMENT; and} \\
(r_3) & \quad \text{'ITEM STOCKED by SUPPLIER'.}
\end{align*}
\]

We shall analyze the evaluation of the following query expression, q:

'Find the DEPARTMENT and the ITEM such that the ITEM is REQUESTED by this DEPARTMENT in QTY \( \geq 10 \), and is supplied (to any DEPARTMENT) by a SUPPLIER that is STOCKING all the ITEMS supplied by him to this DEPARTMENT'.

In the above expression we can identify the following c-schemes:

\[
\begin{align*}
S(q) &= \{x_1: \text{DEPARTMENT}, x_2: \text{ITEM}\}; \\
S(r_1^1) &= \{x_1: \text{DEPARTMENT}, x_2: \text{ITEM}, x_3: \text{QTY}\}; \\
S(r_2^1) &= \{x_2: \text{ITEM}, x_4: \text{SUPPLIER}\}; \\
S(r_3^2) &= \{x_5: \text{ITEM}, x_4: \text{SUPPLIER}, x_1: \text{DEPARTMENT}\}.
\end{align*}
\]

The c-counters of the various c-symbols are initially:

\[
\begin{align*}
c\text{-counter} (x_1) &= 3; \\
c\text{-counter} (x_2) &= 3; \\
c\text{-counter} (x_3) &= 1 \text{ and } x_3 \text{ is also operationally referenced}; \\
c\text{-counter} (x_4) &= 3 \text{ and } x_4 \text{ is also operationally referenced}; \\
c\text{-counter} (x_5) &= 2.
\end{align*}
\]
The natural language referencing, explicit or implicit (e.g. textual contiguity), is assured by the c-symbols, easily deducible from the query expression and unambiguous.

The natural language sentence combination by relativization corresponds to RRA Cartesian product. The chaining of two sentences by a restriction of the form "A \theta B" corresponds to RRA \theta-join, while the chaining of a sentence with a restriction of the form "A \theta c" corresponds to RRA selection. The natural language coordination is expressed with RRA union, RRA not and RRA Cartesian product RRA division and and RRA aggregate function are used to express various natural language quantifiers.

The RRA expression corresponding to the above query is:

\[ r_1^1[\text{QTY} > 10] \times (r_1 \times (r_3 \cdot r_2)) \]

The evaluation is as follows:

1. \( r' = r_3^1 \cdot r_2^2 \), where \( D(r_3^1, r_2^2) = \{x_5: \text{ITEM}\} \),
   \[ M'(r_3^1, r_2^2) = M(r_3^1, r_2^2) - D(r_3^1, r_2^2) = \{x_4: \text{SUPPLIER}\} \],
   and
   \[ S(r') = \{x_4: \text{SUPPLIER}, x_1: \text{DEPARTMENT}\} \]
   with \( \text{c-counter (}x_4\text{)} \) becoming 2,
   and \( \text{c-counter (}x_5\text{)} \) becoming 1;

2. \( r'' = r_2^1 \times r' \), where \( M(r_2^1, r') = \{x_4: \text{SUPPLIER}\} \),
   and
   \[ S(r'') = \{x_2: \text{ITEM}, x_1: \text{DEPARTMENT}\} \]
   with \( \text{c-counter (}x_4\text{)} \) becoming 1;

3. \( r''' = r_1^1[\text{QTY} > 10] \), where \( S(r''') = \{x_2: \text{ITEM}, x_1: \text{DEPARTMENT}\} \)

4. \( r^{1v} = r'''' \times r' \), where \( M(r'', r''') = \{x_2: \text{ITEM}, x_1: \text{DEPARTMENT}\} \)
   and \( S(r^{1v}) = \{x_2: \text{ITEM}, x_1: \text{DEPARTMENT}\} \),
   with both \( \text{c-counter (}x_2\text{)} \) and \( \text{c-counter (}x_1\text{)} \) becoming 2.

The \( t \)-relation described by \( q \) is \( r^{1v} \).
3. CONCLUSION

We have proposed a reshaped Relational Algebra (RRA) to form the manipulative part of the RM. Excepting the projection and the natural join, the RRA operators correspond in form and, partly, in content to their RA homonyms; the RRA operators depart from their RA counterparts due to their more complex semantics implied by the operational schemes of the relation-occurrences involved in the RRA expressions.

Just as the RM structural part, as defined in [7], is closer to the way people perceive information, so there is a close relationship between the RRA operators and the natural language sentence combination. As such RRA could be considered as the basis of query languages having constructs close to the natural language ones. Such a query language, called ERROL (Entity-Relationship, Role Oriented, Query Language), is proposed in [5]. A more detailed presentation of a revised and extended reshaping of RA is in preparation.
REFERENCES


