Relational Algebra.
Part 1. Definitions.

Relational Algebra

Notation

$R, T, S, \ldots$ – relations.

$t, t_1, t_2, \ldots$ – tuples of relations.

$t^{(n)}$ – tuple with n attributes.

$t[1], t[2], \ldots, t[n]$ – 1st, 2nd,... nth attribute of tuple t.

$t.name$ – attribute *name* of tuple t (attribute names and numbers are **interchangable**).

$R.name, R[1], \ldots$ – attribute *name* or attribute number 1 (etc.) of relation R.

$R(X_1 : V_1, \ldots, X_n : V_n)$ – relational schema specifying relation R with n attributes $X_1, \ldots, X_n$ of types $V_1, \ldots, V_n$. We let V_i specify both the *type* and the set of possible values in it.

Base Operations

Union

Definition 1 *Let $R = \{t\}$ and $S = \{t'\}$ be two relations over **the same** relational schema. Then, the union of R and S, denoted $R \cup S$ is a relation T, such that:*

$$T = \{t'' | t'' \in R \ \ or \ \ t'' \in S\}.$$

Union of two relations combines in one relation all their tuples.

Note: it is important to note that for union to be applicable to two relations R and S, they *must have the **same** schema.*

Difference

Definition 2 *Let R and S be two relations over* **the same** *relational schema. Then, the difference of R and S, denoted $R - S$ is a relation T such that:*

$$T = \{t | t \in R \quad and \quad t \notin S\}.$$

Difference of two relations R and S is a relation that contains all tuples from R that are *not contained* in S.

Cartesian Product

Definition 3 *Let R and S be two relations. The cartesian product of R and S, denoted $R \times S$ is a relation T such that:*

$$T = \{t | t = (a_1, \ldots, a_n, b_1, \ldots, b_m) \wedge (a_1, \ldots, a_n) \in R \wedge (b_1, \ldots, b_m) \in S\}.$$

Cartesian product "glues" together all tuples of one relation with all tuples of the other one.

Selection

Definition 4 *Let database $\mathcal{D}$ consist of relations $R_1, \ldots R_n$. Let X, Y be attributes of one of the relations $R_1, \ldots R_n$. An atomic selection condition is an expression of one of the forms:*

$$X \ \mathsf{op}\ \alpha;$$

$$X \ \mathsf{op}\ Y,$$

where $\alpha \in V$ and V is X's domain; $\mathsf{op} \in \{=, \neq, <, >, \leq, \geq\}$ and X and Y have comparable types.

An atomic selection condition is a selection condition.
Let C_1 and C_2 be two selection conditions. Then $C_1 \wedge C_2$, $C_1 \vee C_2$ and $\neg C_1$ are selection conditions.

An expression is a valid selection condition only if it can be constructed using the procedure above.

Selection conditions specify "information needs" of the user.

Definition 5 *Let $t = (a_1, \ldots, a_n) \in R$ be a relational tuple. Let $C = X \ \mathsf{op}\ \alpha$ and $C' = X \mathsf{op}\ Y$ be two atomic selection conditions.*

t **satisfies** C iff $t.X \ \mathsf{op}\ \alpha$ is a true statement.

t **satisfies** C' iff $t.X \ \mathsf{op}\ t.Y$ is a true statement.

t satisfies $C_1 \wedge C_2$ iff t satisfies both C and C'.

t satisfies $C_1 \vee C_2$ iff t satisfies either C_1 or C_2.

t satisfies $\neg C_1$ iff t does not satisfy C_1.

The notion of satisfaction allows us to distinguish between the tuples that contain information sought by the user and those that do not.

Definition 6 *Let R be a relation and C be a selection condition. Selection on R with C denoted $\sigma_C(R)$ is a relation T such that:*

$$T = \{t | t \in R \wedge t \ satisfies \ C\}$$

Projection

Definition 7 *Let R be a relation with schema $(X_1 : V_1, \ldots, X_n : V_n)$. Let F be a sequence $Y_1, \ldots, Y_m$ of attributes from R. Then projection of R on F denoted $\pi_F(R)$ is a relation T such that*

$$T = \{t | t = (a_1, \ldots, a_m), (\exists t'' \in R)(\forall 1 \leq i \leq m) a_i = t''.Y_i\}$$

Projection throws out some attributes of the relation and possibly rearranges the order of the remaining ones.

Rename

This is a "utility" operation, which gives a new name to a given relation and (optionally) renames its attributes. Three possible ways in which it is used are:

- $\rho_S(R)$: relation R is now renamed to S. Attributes preserve their names.
- $\rho_{S(A_1,\ldots,A_n)}(R)$: relation R is now renamed to S. Its attributes are now named $A_1, \ldots, A_n$.
- $\rho_{R(A_1,\ldots,A_n)}(R)$: relation R keeps its name, but the attributes are renamed to $A_1, \ldots, A_n$.

Note: Renaming a relation is useful in situations when you need to write a relational algebra expression that involves using one relation several times for different purposes. It can be combined with the projection operation's power to rearrange the order of columns.

More on Conditions

A more generalized version of conditions used in selection and *join* operations may be described as follows.

> Let database $\mathcal{D}$ consist of relations $R_1, \ldots R_k$. Let $X_1, \ldots X_N$ be all attribute names used in $\mathcal{D}$.
>
> A valid identifier is one of the following: X, $R.Y$, c where X, Y are attribute names and Y is an attribute of relation R and c is a constant.
>
> An atomic condition is an expression of the form
>
> $$f(\mathcal{X}_1, \ldots, \mathcal{X}_m) \text{ op } g(\mathcal{Y}_1, \ldots, \mathcal{Y}_k),$$
>
> where op $\in \{=, \neq, >, <, \leq, \geq\}$, $\mathcal{X}_1, \ldots, \mathcal{X}_m, \mathcal{Y}_1, \ldots \mathcal{Y}_k$ are all valid identifiers and $f^{(m)}(.)$ and $g^{(k)}(.)$ are ***computable*** functions.
>
> Finally, atomic conditions are conditions and if C_1 and C_2 are conditions than so are $C_1 \wedge C_2$, $C_1 \vee C_2$ and $\neq C_1$.
>
> Valid conditions are only those that can be constructed using the procedures described above.

This definition of produces a larger set of conditions, some of which are no selection conditions. It also extends the set of selection conditions: e.g.,
$R.X_1 + R.X_2 \leq (R.X_3)^2 - 25$ is a valid selection constraint now.

Derived Operations

Some important relational algebra operations can be derived from the basic operations.

Intersection

Definition 8 *Let R and S be two relations with the* ***same*** *relational schema. An intersection of R and S, denoted $R \cap S$ is a relation T such that*

$$T = \{t | t \in R \quad and \quad t \in S\}$$

Intersection finds tuples common to two relations.

From set theory we know that $R \cap S = R - (R - S)$, hence, *intersection* is a derived operation in our relational algebra.

Division (Quotient)

Definition 9 *Let R be a relation of arity r with relational schema $X_1 : V_1, \ldots, X_r : V_r$ and S be a relation of arity $s < r$ $(s \neq 0)$ with schema $Y_1 : V_{r-s+1}, \ldots, Y_s : V_r$.*

The result of division of R on S (also known as the quotient of R and S), denoted R/S, is a relation T of arity $r - s$ with schema $X_1 : V_1, \ldots, X_{r-s} : V_{r-s}$ such that

$$T = \{t = (a_1, \ldots, a_{r-s}) | (\forall s = (a_{r-s+1}, \ldots, a_r) \in S)((a_1, \ldots, a_r) \in R)\}$$

Division finds all "prefixes" in one relation that have *every* "suffix" from the second relation.

Division can be derived as follows: Let $F = X_1, \ldots, X_{r-s}$. Then,

$$R/S = \pi_F(R) - \pi_F((\pi_F(R) \times S) - R).$$

Joins

The family of join operations allows us to "intelligently" combine together the contents of two (or more) different relations.

Θ-Join

Also known as *condition join*.

Definition 10 *Let R and S be two relations with attributes $(X_1, \ldots, X_n)$ and $(Y_1, \ldots, Y_m)$ respectively and let $C = R.X \Theta S.Y$ be a* join condition, *with $\Theta \in \{=, \neq, <, >, \leq, \geq\}$. A Θ-join of R and S, denoted $R \bowtie_C S$, is a relation T such that:*

$$T = \{t = (a_1, \ldots, a_n, b_1, \ldots, b_m) | r = (a_1, \ldots, a_n) \in R \wedge s = (b_1, \ldots, b_m) \in S \wedge s.X \Theta r.Y \quad is\ true \quad \}.$$

Join selects the tuples from the cartesian product of two relations that match a given constraint (typically, comparing the attributes from both relations). Θ-join can be expressed via selection and cartesian product as follows:

$$R \bowtie_C S = \sigma_C(R \times S).$$

Equijoin

Equijoin is a special type of Θ-join where the join condition is a conjunction of equalities: $R.X = S.Y$.

Natural Join

Natural Join is a type of equijoin of two relations.

Definition 11 *Let $R(X_1, \ldots, X_n)$ and $S(Y_1, \ldots, Y_m)$ be two relations and let $Z_1, \ldots Z_k$ **all** common attributes of R and S. Let $\mathcal{Y}' = (Y_1, \ldots, Y_m) - (Z_1, \ldots, Z_k)$. A **natural join** of R and S, denoted $R \bowtie S$ is a relation T such that*

$$T = \pi_{X_1,\ldots,X_n,\mathcal{Y}'}(R \bowtie_{R.Z_1=S.Z_1 \wedge \ldots R.Z_k=S.Z_k} S)$$

Natural join looks at **all common** attributes of two relations and joins on them, removing one set of common attributes from the final relation.

Semijoin

A semijoin of two relations performs a(ny) join operation of two relations and then projects the result onto the fields of one of the relations:

$$R \ltimes_\Theta S = \pi_R(R \bowtie_\theta S)$$
$$R \rtimes_\Theta S = \pi_S(R \bowtie_\theta S)$$

Examples

Consider the following five relations (**W** is at the bottom of the page):

R:

A	B	C	D
1	1	a	a
1	2	a	b
2	1	b	b
2	2	b	b
2	2	a	a

S:

B	E	F
1	a	c
1	b	a
2	a	a
3	b	b
2	c	a

T:

A	E	G
2	b	1
1	a	2
2	b	4
3	b	2
1	b	6

V:

A	E	G
2	b	1
3	a	2
3	b	1
1	a	2
1	c	0

Here are the results of some relational algebra operations on these tables.

$T \cup V$

A	E	G
2	b	1
1	a	2
2	b	4
3	b	2
1	b	6
3	a	2
3	b	1
1	c	0

$T - V$

A	E	G
2	b	4
3	b	2
1	b	6

$V - T$

A	E	G
3	a	2
3	b	1
1	c	0

$V \cap T$

A	E	G
2	b	1
1	a	2

$\sigma_{A=2}(R)$

A	B	C	D
2	1	b	b
2	2	b	b
2	2	a	a

$\sigma_{E \neq 'b'}(S)$

B	E	F
1	a	c
2	a	a
2	c	a

$\sigma_{G>2}(T)$

A	E	G
2	b	4
1	b	6

$\sigma_{G>A}(T)$

A	E	G
1	a	2

$\pi_{A,D}(R)$

A	D
1	a
1	b
2	b
2	a

$\pi_{F,B}(S)$

F	B
c	1
a	1
a	2
b	3

$\pi_E(T)$

E
b
a

$\pi_{E,G,A}(T)$

E	G	A
b	1	2
a	2	3
b	1	3
a	2	1
c	0	1

W:

A	B
2	1
2	2

R/W

C	D
b	b

$T \times W$

T.A	E	G	W.A	B
2	b	1	2	1
2	b	1	2	2
1	a	2	2	1
1	a	2	2	2
2	b	4	2	1
2	b	4	2	2
3	b	2	2	1
3	b	2	2	2
1	b	6	2	1
1	b	6	2	2

$W \times T$

W.A	B	T.A	E	G
2	1	2	b	1
2	1	1	a	2
2	1	2	b	4
2	1	3	b	2
2	1	1	b	6
2	2	2	b	1
2	2	1	a	2
2	2	2	b	4
2	2	3	b	2
2	2	1	b	6

$S \bowtie_{B>A} T$

B	S.E	F	A	T.E	G
2	a	a	**1**	a	2
2	a	a	**1**	b	6
3	b	b	**2**	b	1
3	b	b	**1**	a	2
3	b	b	**2**	b	4
3	b	b	**1**	B	6
2	c	a	**1**	a	2
2	c	a	**1**	b	6

$T \bowtie_{T.A=V.G} V$

T.A	T.E	T.G	V.A	V.E	V.G
2	b	1	3	a	**2**
2	b	1	1	a	**2**
1	a	2	2	b	**1**
1	a	2	3	b	**1**
2	b	4	3	a	**2**
2	b	4	1	a	**2**
1	b	6	2	b	**1**
1	b	6	3	b	**1**

$R \bowtie T$

A	B	C	D	E	G
1	1	a	a	a	2
1	1	a	a	b	6
1	2	a	b	a	2
1	2	a	b	b	6
2	1	b	b	b	1
2	1	b	b	b	4
2	2	b	b	b	1
2	2	b	b	b	4
2	1	a	a	b	1
2	1	a	a	b	4

$S \bowtie V$

B	E	F	A	G
1	a	c	3	2
1	a	c	1	2
1	b	a	2	1
1	b	a	3	1
2	a	a	3	2
2	a	a	1	2
3	b	b	2	1
3	b	b	3	1
2	c	a	1	0

$T \bowtie V$

A	E	G
2	b	1
1	a	2

$S \ltimes T$

B	E	F
1	a	c
1	b	a
2	a	a
3	b	b

$S \rtimes T$

A	E	G
2	b	1
1	a	2
2	b	4
3	b	2
1	b	6