Relational Calculus (RC) does not imply any connection with the branch of mathematics usually called ‘Calculus’; rather RC comes from the first order predicate calculus from the field of Logic. There are two types of RCs – tuple relational calculus (TRC) and domain relational calculus (DRC). Here the tuple relational calculus is discussed.

In TRC, expressions are of the form $\{t / \psi(t)\}$, where t is a tuple variable, i.e. a variable denoting a tuple of some fixed length and ψ is a formula built from atoms and a collection of operators to be defined.

The atoms of formulas ψ are of 3 types
:

i. R(s), where R is a relation name and s is a tuple variable. This atom stands for the assertion that s is a tuple in relation R.

ii. $s[i]\ \theta\ u[j]$, where u and s are tuple variables and θ is an arithmetic comparison operator (<, >, =). This atom stands for the assertion that i^{th} component of s stands in relation θ to the j^{th} component of u.

iii. $s[i]\ \theta\ a$ or $a\ \theta\ s[i]$ are same as above; only difference is that a is a constant.

Free and Bound variable : Informally, an occurrence of a variable in a formula is ‘bound’ if that variable has been introduced by a ‘for all’ or ‘there exists’ quantifier, and we say that the variable is ‘free’ if not.

The notion of a ‘free variable’ is analogous to that of a global variable in a programming language i.e. a variable defined outside the current procedure. On the other hand, a ‘bound variable’ is like a local variable, one that is defined in the procedure at hand and can’t be referenced from the outside.

Formulas and free and bound occurrences of tuple variable in these formulas are defined recursively as follows:

i. Every atom is a formula. All occurrences of tuple variables mentioned in the atom are free in this formula.

ii. If Ψ_1 and Ψ_2 are formulas, then $\Psi_1 \wedge \Psi_2$, $\Psi_1 \vee \Psi_2$ and $-\Psi_1$ are formulas asserting ‘Ψ_1 and Ψ_2 are both true’, ‘Ψ_1 or Ψ_2 or both true’ and ‘Ψ_1 is not true’ respectively. Occurrences of tuple variables are free or bound in $\Psi_1 \wedge \Psi_2$, $\Psi_1 \vee \Psi_2$ and $-\Psi_1$ as they are free or bound in Ψ_1 or Ψ_2 depending on where they occur. Note that an occurrence of a variable s could be bound in Ψ_1 while another occurrence of s is free in Ψ_2 or conversely.

iii. If Ψ is a formula then $(\exists s)(\Psi)$ is a formula. The symbol $\exists$ is a quantifier. The only other quantifier used is $\forall$, ‘for all’ described in iv. Below. Occurrences of s that are free in Ψ are bound to $(\exists s)$ in $(\exists s)(\Psi)$. The formula $(\exists s)(\Psi)$ asserts that there exists a value of s such that when we substitute this value for all free occurrences of s in Ψ, the formula Ψ becomes true. eg. $(\exists s)(R(s))$ says that relation R is not empty i.e. there exists a tuple s in R.

iv. If Ψ is a formula then $(\forall s)(\Psi)$ is a formula. Free occurrences of s in Ψ are bound to $(\forall s)$ in $(\forall s)(\Psi)$. The formula $(\forall s)(\Psi)$ asserts that whatever value of the appropriate arity we substitute for free occurrences of s in Ψ, the formula Ψ becomes true.

v. Parenthesis may be placed around formulas as needed. We assume the order of precedence is : arithmetic comparison operators highest, then the quantifiers $\exists$ and $\forall$, then $\neg$, $\wedge$, and $\vee$ in that order.

vi. Nothing else is a formula.

Ψ is psi.

Recall $\{t/ \Psi(t)\}$ is a tuple relational calculus expression where t is the only free variable.

Representation of five basic algebraic operators in TRC :

Union : $\{t/R(t) \vee S(t)\}$ → The set of tuples t such that t is in R or is in S

Union only makes sense if R and S have the same arity. The variable t is assumed to have some fixed length.

Difference : $\{t/ R(t) \wedge \neg S(t)\}$

Cartesian Product : $\{t^{(r+s)}/ (\exists u^{(r)})(\exists v^{(s)})(R(u) \wedge S(v) \wedge t[1] = u[1] \wedge \ldots \wedge t[r] = u[r]$
$\wedge t[r+1] = v[1] \wedge \ldots \wedge t[r+s] = v[s])\}$

It means R X S is the set of tuples t of length (r+s) such that there exists u and v, with u in R, v in S, the first r components of t form u and the next s components of t form v.

Projection : $\{t^{(k)}/ (\exists u)(R(u) \wedge t[1] = u[i_1] \wedge \ldots \wedge t[k] = u[i_k])\}$

The equivalent RA expression is -- $\Pi_{i1,\, i2,\, \ldots\, ik}$ (R)

Selection : The selection σ_F (R) is expressed by $\{t/ R(t) \wedge F'\}$ where F' is the formula F with each operand I denoting the i^{th} component replaced by t[I]

Example Schema 1:

TEACHER (tname,dept,tel_no, sub_title)
STUDENT (sname, course, hall)

STUDY (sub_title, sname, status, marks)

1. R.A : $\Pi_{tel\_no,\ sub\_title}$ ($\sigma_{tname = 'XYZ'}$ (TEACHER))

   TRC : {t^2/ (∃u (u ∈ TEACHER ∧ t [tel_no] = u [tel_no] ∧ t[sub_title] = u[sub_title] ∧ u[tname] = 'XYZ')}

2. R.A : Π_{sname} ($\sigma_{sub\_title = 'DBMS'}$ (STUDY))

   TRC : {t^1/ (∃u)(u ∈ STUDY ∧ t[sname] = u[sname] ∧ u[sub_title] = 'DBMS')}

3. R.A. : $\Pi_{sname,\ marks}$ ($\sigma_{course = 'IEP' \wedge sub\_title = 'DBMS' \wedge status = 'elective'}$ (STUDENT ⋈ STUDY))

   TRC : {t^2/ ∃u (u∈ STUDENT ∧ t[sname] = u[sname] ∧ u[course] = 'IEP') ∧ ∃v (v∈ STUDY ∧ v[sname] = u[sname] ∧ t[marks] = v[marks] ∧ v[sub_title] = 'DBMS' ∧ v[status] = 'elective')}

4. R.A. : $\Pi_{sname,\ hall}$ (STUDENT X $\sigma_{marks < 35}$ (STUDY))

   TRC : { t^2 / ∃u (u∈ STUDENT ∧ t[sname] = u[sname] ∧ t[hall] = u[hall] ∃v (v∈ STUDY ∧ v[sname] = u[sname] ∧ v[marks] < 35)}

Example Schema 2 :

EMP (e_no, e_name, sal)
WORKS_IN (e_no, dname, duration)
DEPT (dname, budget, floor)

1. Find all information about all overpaid employees

TRC : { e / e ∈ EMP ∧ e.sal >10,500}

2. Find names and salaries of all overpaid employees

TRC : {t^2 / (∃e) (e∈ EMP) ∧ t.sal > 10,500 ∧ t. e_name = e.e_name ∧ t.sal = e.sal}

3. Find names of overpaid CS department employees

TRC : {t^1/ ∃u ∈ EMP (u.sal > 10,500 ∧ t.ename = u.ename ∧ ∃w ∈ WORKS_IN ((w.eno = u.eno ∧ w. dname = 'CS'))}

4. Find names of employees who work in all departments

TRC : {t^1/ ∃u ∈ EMP (t.e_name = u.e_name ∧ ∀d ∈ DEPT (∃w ∈ WORKS_IN

$(w.e\_no = u.e\_no \wedge w.dname = d.dname))\}$

5. Find names of employees who work in CS department but not in Maths department.

<u>TRC</u> : $\{t^1 / \exists u \in EMP\ (t.e\_name = u.e\_name \wedge \exists w \in WORKS\_IN\ (w.dname = \text{'CS'} \wedge w.e\_no = u.e\_no) \wedge NOT\ \exists v \in WORKS\_IN\ (v.e\_no = u.e\_no \wedge v.dname = \text{'MATHS'}))\}$