Tuple Relational Calculus :-

⟹ Predicate Logic && First Order Logic.

$\boxed{\in, \wedge, \vee, \neg, \forall, \exists \text{ etc}}$

⟹ Non- Procedural Query Language

⟹ Query form : $\{T / P(T)\}$ ≡ SQL: SELECT A FROM R WHERE P

T : Tuple Variables

P(T) : Formula on tuple variable

↳ Retrieves tuples set those satisfying formula P(T)

$\{T / P(T)\}$

Output · Condition for output

$\boxed{\pi_A(\sigma_P(R \times S))}$ Relational algebra operators

» In tuple Relational calculus "Relational Calculus formula" are used.

Q: Supplier (Sid, Sname, rating)

Catalog (Sid, Pid, Cost)

Parts (Pid, Pname, color)

(1) Retrieve Suppliers whose name = "Alice"

$\boxed{\{S / S \in \text{supplier} \, (S.name = Alice)\}}$

```
SELECT *
FROM SUPPLIERER
WHERE  Name = 'Alice'
```

② Retrieve Sid's of supplier whose name = "Alice"

Free Tuple variable : Variable not bounded by quantifier

Bounded Variables : Variable is bounded only if preceded by quantifier.

$\exists$: { There exist }

$\forall$: { For all }

$\boxed{\exists S \in \text{supplier}\,(P(S))}$

$\boxed{\forall S \in \text{supplier}\,(P(S))}$

S = bounded variable here.

Some useful Results

① $\boxed{\forall x\,(P(x) \wedge Q(x)) = \forall x\,(P(x)) \wedge \forall x\,(Q(x))}$

② $\boxed{\neg \forall x\,(P(x)) = \exists x\ \neg(P(x))}$

③ $\boxed{\neg \exists x\,(P(x)) = \forall x\ \neg(P(x))}$

De morgan's law

④ $\boxed{\forall x\,[P(x) \longrightarrow Q(x)] = \forall x(\neg P(x) \vee Q(x))}$

⑤ $\boxed{\neg \forall x\,(P(x) \longrightarrow Q(x)) = \exists x\ \neg(P(x) \longrightarrow Q(x))}$

Acc. to FOLogic

$P \rightarrow Q = \neg P \vee Q$

$\neg(P \rightarrow Q) = P \wedge \neg Q$

$\boxed{\neg \forall x\,(P(x) \longrightarrow Q(x)) = \exists x\,(P(x) \wedge \neg Q(x))}$

$\boxed{\{T \mid \exists S \in \text{suppliers}\,(S.name = Alice \wedge T.sid = S.sid)\}}$

3) Retrieve ~~sid~~ Sname of supplier who supplied some part

$$\boxed{\{T / \exists S \in \underbrace{Supplier}_{} (\exists C \in \underbrace{catalog}_{\text{2 Tables}} (S.sid = C.sid) \wedge T.sname = S.sname\}}$$

4) Sid's of supplier supplied some red parts

$$\boxed{\{T / \exists c \in \underbrace{catalog}_{} (\exists P \in \underbrace{Parts}_{\text{2 Tables}} (P.color = RED \wedge P.pid = C.pid) \wedge T.sid = C.sid)\}}$$

5) Sid's of suppliers who supplied non-RED parts

Relational calculus

$$\pi_{\substack{\cancel{Sname} \\ Sid}} \left(\sigma_{(P.pid = C.pid) \wedge (P.color \neq Red)} (C \times P) \right)$$

that part should not be Red

we get part supplied by some supplier

Tuple relational calculus

$$\boxed{\{\ / \exists C \in catalog (\exists P \in Parts (P.color \neq RED \wedge P.pid = C.pid) \wedge T.sid = C.sid)\}}$$

UNSAFE QUERY → compliment of free variable (S')

\# $\{S / \neg S \in supplier\}$

S = free variable

new O/P = S' — infinite

$\{T / \neg \exists S \in Supplier \wedge T.sid = S.sid\}$

$\boxed{\neg \exists x (P(x)) = \forall x \neg (P(x))}$

— De morgan's

$\neg \exists S \in supplier$

$\forall S \ \neg (S \in supplier)$

not belongs to

O/P zero tuple

⇒ Compliment of free variable = unsafe.

⇒ Safe Tuple Relational calculus ≡ Basic Rel. Algebra

↳ Expressive Power

⇒ Basic Relational algebra operators

$\boxed{(\pi, \sigma, \times, \cup, \cap, -, \rho)}$

↳ Not possible to write with basic relational algebra

Not possible using Basic R.A

- Outer Join
- Aggregation
- Group by, ordering

so not possible using T.R calculus also