

Tuple Relational Calculus :-

⇒ Predicate logic & First Order logic.

$\in, \wedge, \vee, \neg, \forall, \exists$ etc

⇒ Non-Procedural Query Language

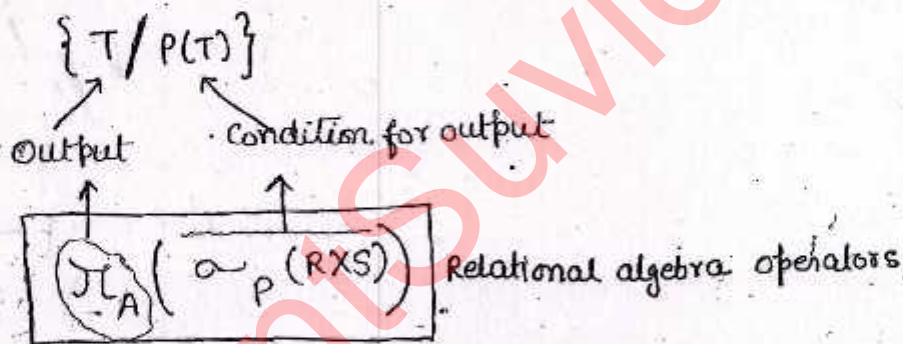
⇒ Query form: $\{T/P(T)\}$

SQL
SELECT A
FROM R
WHERE P

T: Tuple Variables

P(T): Formula on tuple variable

↳ Retrieves tuples set those satisfying formula P(T)



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

Suppliers (sid, sname, rating)

Catalog (sid, pid, cost)

Parts (pid, pname, color)

① Retrieve suppliers whose name = "Alice"

$\{S/S_{supplier} (S.name = Alice)\}$

SELECT *
FROM SUPPLIER
WHERE name = 'Alice'

② Retrieve sid's of suppliers 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 }

$\exists s \in \text{suppliers}(P(s))$

$\forall s \in \text{suppliers}(P(s))$

L s = bounded variable here.

Some useful Results

① $\forall x (P(x) \wedge Q(x)) = \forall x (P(x)) \wedge \forall x (Q(x))$

② $\neg \forall x (P(x)) = \exists x \neg (P(x))$

③ $\neg \exists x (P(x)) = \forall x \neg (P(x))$

De Morgan's law

④ $\forall x [P(x) \rightarrow Q(x)] = \forall x (\neg P(x) \vee Q(x))$

⑤ $\neg \forall x (P(x) \rightarrow Q(x)) = \exists x \neg (P(x) \rightarrow Q(x))$

Acc. to FO logic

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

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

$\neg \forall x (P(x) \rightarrow Q(x)) = \exists x (P(x) \wedge \neg Q(x))$

$\left\{ \begin{array}{l} T \\ \exists s \in \text{suppliers} (s.name = Alice \wedge T.sid = s.sid) \end{array} \right\}$

3) Retrieve sname of supplier who supplied some part.

$$\pi_{SNAME} \left(\exists E \in \text{supplier} \left(\exists C \in \text{catalog} (S.SID = C.SID) \wedge T.SNAME = S.SNAME \right) \right)$$

2 Tables

4) Sid's of supplier supplied some red parts

$$\pi_{SID} \left(\exists C \in \text{catalog} \left(\exists P \in \text{parts} (P.COLOR = RED \wedge P.PID = C.PID) \wedge T.SID = C.SID \right) \right)$$

2 Tables

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

Relational calculus

$$\pi_{SID} \left(\exists C \in \text{catalog} \left(\exists P \in \text{parts} (P.PID = C.PID) \wedge (P.COLOR \neq RED) \right) \right)$$

we get part supplied by some supplier

that part should not be red

Relational calculus

$$\pi_{SID} \left(\exists C \in \text{catalog} \left(\exists P \in \text{parts} (P.COLOR \neq RED \wedge P.PID = C.PID) \wedge T.SID = C.SID \right) \right)$$

UNSAFE QUERY → Compliment of free variable ('S')

{ S / T S ∈ supplier }

S = free variable

~~...~~

new o/p = S

infinite

{ T / ¬ ∃ S ∈ supplier ∧ T.SID = S.SID }

De Morgan's

¬ ∃ S ∈ supplier

∀ S ¬ (S ∈ supplier)

not belongs to 'S'

o/p zero tuple

⇒ Compliment of free variable = unsafe

⇒ Safe Tuple Relational Calculus \equiv Basic Rel. Algebra

↳ Expressive Power

⇒ Basic Relational algebra operators

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

↳ Not possible to write with basic relational algebra

Not possible using Basic RA

- » Outer Join
- » Aggregation
- » Group by, Ordering

so not possible using T.R. calculus also

StudentsSuvirdha.com