11/1/23

RELATIONAL ALGEBRA & RELATIONAL CALCULUS →

→ To work with Relational Database

→ RA → Procedural | RC → Non-Procedural

(Expressions → Sequence of Operations) | → What data we need (No op^n given)

■ Relational Algebra → ∘) Provides the foundation to work with Rel^n dB

∘) Useful for implementing & optimising the query

∘) Some of the concepts of Rel^n Alg is implemented in SQL.

eg:— DEPT (DCODE, DNAME, ...)

STUDENT (ROLL, NAME, ---, DCODE (FK))

In an expression →
1 or more rel^n → I/P
Operations → Applied
Rel^n(s) → O/P

~~Find stu.~~

Pred^n is check on every tuple

(#) SELECT Operation → [COMMUTATIVE ($\sigma_1 \sigma_2 R = \sigma_2 \sigma_1 R$)]

$\sigma_{predicate}(R)$

> >= < <= = != AND OR NOT

eg:— $\sigma_{DCODE = 'D1'}(STUDENT)$

O/P Rel^n →
- Schema → Same as I/P Rel^n
- State → Set of tuples satisfying the predicate

[CONJUNCTIVE ($\sigma_{P1} \sigma_{P2} R = \sigma_{P1\ AND\ P2} R$)]

PROJECT OPERATION → [NOT COMMUTATIVE]

$\pi_{list\_of\_attributes}(R)$

$\langle L_i \rangle \ \forall L_i \in R$

eg:— $\pi_{ROLL, NAME}(STUDENT)$

O/P →
- Schema → ROLL, NAME
- #Tuples → Likely to be same as #tuples in I/P Rel^n

[eg:— $\pi_{SCORE}(ST^D)$
→ Same scores are merged into 1]

(#) DUPLICATE ELIMINATING PROJECT Opⁿ → If Projected Attⁿ doesnot contain Superkey, after projection, multiple tuples maybe taken as 1.

$\pi_{a_1,a_2}(\sigma_{a_3>30}(R)) = \pi_{a_1,a_2}(\sigma_{a_3>30}(\pi_{a_1,a_2,a_3}(R)))$

[Broader Table narrowed → In Memory (Not Disk) → Faster → Optimisation]

First 4 are ~~Also~~ FUNDAMENTAL Operations

#) RENAME Operation →

$\rho_{R2(X,Y,Z,W)}(R1(A,B,C,D))$

General: [$R2(W,X,Y,Z) \leftarrow Op^n(R)$]

(Can use assignment for changing name also) $R2(X,Y,Z,W) \leftarrow \pi_{A,B,C,D}(R1(A,B,C,D))$

#) CARTESIAN PRODUCT →

$\underset{A,B}{R1} \times \underset{A,B}{R2}$

$\underbrace{\qquad\qquad}_{R1.A,\ R1.B,\ R2.A,\ R2.B}$

O/P Relⁿ →
- Schema → $Sch(R1) \cup Sch(R2)$ &
- State → $|R1| \times |R2|$ Relations

eg:— $\pi_{ROLL, NAME, DNAME}(\sigma_{STD.DCODE = DEPT.DCODE}(STUDENT \times DEPT))$

→ Qualification needed as conflicting attributes

Derived Operations

#) JOIN →

$R1 \bowtie_{Condition} R2$ eg:— $STD \bowtie_{STD.DCODE = DEPT.DCODE} DEPT$

If Condition is not given, equality of all common attributes is checked...

(Use RENAME to remove conflicting attⁿ on which = not done, if condⁿ not given) eg:— $STD \bowtie \rho_{DCODE,\ DNAME}\ DEPT$

⊛ If not based on '=' (equality) of attⁿ, (>, <, >=, etc.) → θ (Theta JOIN)

Later, choose any 1 attⁿ (common) (maybe) (Cannot join unequal attⁿ)

Same Score Students Pairs → $(\sigma_{STUDENT.SCORE = T.SCORE\ AND\ STUDENT.ROLL\ != T.ROLL}(STUDENT \times \rho_T(STUDENT)))$

Can give < /OR/ >

To avoid (2,3) (3,2)

FUND$^{\underline{M}}$ OP →

Select, Project, Rename, Car$^{\underline{t}}$ Pd$^{\underline{t}}$, Union, Minus

DER$^{\underline{v}}$ OP →

Intersection, Join, Natural Join