UNIT 2

Lecture 24

Domain Relational Calculus

Domain Relational Calculus

- A second form of relational calculus, called **domain relational calculus**, uses *domain* variables that take on values from an attributes domain, rather than values for an entire tuple.
- The domain relational calculus, however, is closely related to the tuple relational calculus.
- Domain relational calculus serves as the theoretical basis of the widely used QBE language, just as relational algebra serves as the basis for the SQL language.

Domain Relational Calculus

Formal Definition of Domain Relational Calculus

An expression in the domain relational calculus is of the form

$$\{< x1, x2, . . . , xn > \mid P(x1, x2, . . . , xn)\}$$

where $x1, x2, . . . , xn$ represent domain variables.

P represents a formula composed of atoms, as was the case in the tuple relational calculus.

Domain Relational Calculus

- An atom in the domain relational calculus has one of the following forms:
  - $< x1, x2, . . . , xn > \in r$, where r is a relation on n attributes and $x1, x2, . . . , xn$ are domain variables or domain constants.
  - $x \Theta y$, where x and y are domain variables and Θ is a comparison operator ($<, \leq, =, =, >, \geq$). We require that attributes x and y have domains that can be compared by Θ.
  - $x \Theta c$, where x is a domain variable, Θ is a comparison operator, and c is a constant in the domain of the attribute for which x is a domain variable.

Domain Relational Calculus

We build up formulae from atoms by using the following rules :

- An atom is a formula.
- If $P1$ is a formula, then so are $\neg P1$ and $(P1)$.
- If $P1$ and $P2$ are formulae, then so are $P1 \vee P2$, $P1 \wedge P2$, and $P1 \Rightarrow P2$.
- If $P1(x)$ is a formula in x, where x is a domain variable, then

$$\exists x\ (P1(x)) \text{ and } \forall x\ (P1(x))$$

are also formulae.

As a notational shorthand, we write

$$\exists a, b, c\ (P(a, b, c))$$

for

$$\exists a\ (\exists b\ (\exists c\ (P(a, b, c))))$$

DRC Example : 1

Display all the information of students whose branch is CSE.

RA : $\sigma_{\text{branch = "CSE"}}$ (STUDENT)

SQL > select * from student where br = 'CSE';

TRC : { t | t ϵ $STUDENT \ \wedge t[branch] = "CSE"\}$

DRC : {<r, sn, s, b, m, p> | <r, sn, s, b, m, p> ϵ $STUDENT \ \wedge b = "CSE"\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123

TRC Example : 2

Display all the information of students whose sem is 3.

RA : $\sigma_{sem = 3}$ (STUDENT)

SQL > select * from student where sem = 3;

TRC : { t | t ϵ $STUDENT \ \wedge t[sem] = 3\}$

DRC : {<r, sn, s, b, m, p> | <r, sn, s, b, m, p> ϵ $STUDENT \ \wedge s = 3\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

TRC Example : 3

Display all the information of students whose marks is greater than 50.

RA : $\sigma_{marks > 50}$ **(STUDENT)**

SQL > select * from student where marks > 50;

TRC : $\{ t \mid t \in STUDENT \wedge t[marks] > 50\}$

DRC : $\{<r, sn, s, b, m, p> \mid <r, sn, s, b, m, p> \in STUDENT \wedge m > 50\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Rollno	Sname	Sem	Branch	Marks	Pno
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
6	PINKI	3	ETC	90	123

TRC Example : 4

Display all the information of students whose branch is CSE or sem is 3.

RA : $\sigma_{branch\ =\ "CSE"\ \vee\ sem\ =\ 3}$ (STUDENT)

SQL > select * from student where branch = 'CSE' or sem = 3;

TRC : { t | t $\epsilon\ STUDENT\ \wedge t[branch] = "CSE"\ \vee t[sem] = 3\}$

DRC : {<r, sn, s, b, m, p> | <r, sn, s, b, m, p> $\epsilon\ STUDENT\ \wedge b = "CSE"\ \vee\ s = 3\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

TRC Example : 5

Display all the information of students whose branch is CSE and sem is 3.

RA : $\sigma_{branch\ =\ "CSE" \wedge sem\ =\ 3}$ (STUDENT)

SQL > select * from student where branch = 'CSE' and sem = 3;

TRC : { t | t $\epsilon\ STUDENT\ \wedge t[branch] = "CSE"\ \wedge\ t[sem] = 3\}$

DRC : {<r, sn, s, b, m, p> | <r, sn, s, b, m, p> $\epsilon\ STUDENT\ \wedge b = "CSE"\ \wedge\ s = 3\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121

TRC Example : 6

Display the names of all the students.

RA : Π_{sname} (STUDENT)

SQL > select distinct sname from student;

TRC : { t | $\exists\, s \in STUDENT\ (t[sname] = s[sname])\}$

DRC : {<sn> | $\exists < r, s, b, m, p > (< r, sn, s, b, m, p > \ \epsilon\ STUDENT)\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Sname
RAM
SHYAM
MOHAN
GOPAL
RINKI
PINKI

TRC Example : 7

Display the name and semester value of all the students.

RA : Π $_{\text{sname, sem}}$ (STUDENT)

SQL > select distinct sname, sem from student;

TRC : { t | $\exists\, s \in STUDENT\ (t[sname] = s[sname]\ \wedge t[sem] = s[sem])\}$

DRC : {<sn,s> | $\exists < r, b, m, p > (< r, sn, s, b, m, p > \ \in STUDENT)\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Sname	Sem
RAM	3
SHYAM	5
MOHAN	7
GOPAL	5
RINKI	3
PINKI	3

TRC Example : 8

Display the name of all the students of CSE branch.

RA : $\Pi_{sname}(\sigma_{branch = \text{“CSE”}}(STUDENT))$

SQL > select distinct sname from (select * from student where branch = ‘CSE’);

TRC : TRC : { t |$\exists\, s \in STUDENT\ (t[sname] = s[sname]\ \wedge s[branch] = \text{"}CSE\text{"})\}$

DRC : {<sn> | $\exists < r, s, b, m, p > (< r, sn, s, b, m, p >\ \in STUDENT\ \wedge b = \text{"}CSE\text{"})\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

OUTPUT

Sname
RAM
SHYAM
MOHAN

TRC Example : 9

Display the name of branches in which project 121 or 122 or both are running.

RA : $\Pi_{branch}(\sigma_{pno = 121}(STUDENT)) \cup \Pi_{branch}(\sigma_{pno = 122}(STUDENT))$

SQL > select distinct branch from student where pno = 121 Union select distinct branch from student where pno = 122;

TRC : $\{ t \mid \exists\, s \in STUDENT\ (t[branch] = s[branch]\ \wedge s[pno] = 121)$
$\vee \exists\, u \in STUDENT\ (t[branch] = u[branch]\ \wedge u[pno] = 122)\}$

DRC : $\{<b> \mid \exists < r, sn, s, m, p > (< r, sn, s, b, m, p > \ \in STUDENT\ \wedge p = 121)$
$\vee \exists < r, sn, s, m, p > (< r, sn, s, b, m, p > \ \in STUDENT\ \wedge p = 122)\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

Branch
CSE
IT

Branch
CSE
MECH

OUTPUT

Branch
CSE
IT
MECH

TRC Example : 10

Display the name of branches in which project 121 and 122 are running.

RA : $\Pi_{branch}(\sigma_{pno = 121}(STUDENT)) \cap \Pi_{branch}(\sigma_{pno = 122}(STUDENT))$

SQL > select distinct branch from student where pno = 121 intersect select distinct branch from student where pno = 122;

TRC : $\{ t \mid \exists s \in STUDENT\ (t[branch] = s[branch] \wedge s[pno] = 121)$
$\wedge \exists u \in STUDENT\ (t[branch] = u[branch] \wedge u[pno] = 122)\}$

DRC : $\{\langle b\rangle \mid \exists < r, sn, s, m, p > (< r, sn, s, b, m, p > \in STUDENT \wedge p = 121)$
$\wedge \exists < r, sn, s, m, p > (< r, sn, s, b, m, p > \in STUDENT \wedge p = 122)\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

Branch
CSE
IT

Branch
CSE
MECH

OUTPUT

Branch
CSE

TRC Example : 11

Display the name of branches in which project 121 is running but 122 is not.

RA : $\Pi_{branch}(\sigma_{pno = 121}(STUDENT)) - \Pi_{branch}(\sigma_{pno = 122}(STUDENT))$

SQL > select distinct branch from student where pno = 121 minus / except select distinct branch from student where pno = 122;

TRC : { t | $\exists\, s \in STUDENT\ (t[branch] = s[branch]\ \wedge s[pno] = 121)$

$\wedge\ \neg \exists u \in STUDENT\ (t[branch] = u[branch]\ \wedge u[pno] = 122)\}$

DRC : {<b> | $\exists < r, sn, s, m, p > (< r, sn, s, b, m, p >\ \in STUDENT\ \wedge p = 121)$

$\wedge \neg \exists < r, sn, s, m, p > (< r, sn, s, b, m, p >\ \in STUDENT\ \wedge p = 122)\}$

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

Branch
CSE
IT

Branch
CSE
MECH

OUTPUT

Branch
IT

TRC Example : 12

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{sname,\ pname}$ (STUDENT $\bowtie_{student.pno\ =\ project.pno}$ PROJECT)

Sname	Pname
RAM	P1
GOPAL	P1
SHYAM	P2
RINKI	P2
MOHAN	P3
PINKI	P3

SQL > select distinct sname, pname from student inner join project on student.pno = project.pno;
[SQL 99 syntax]

SQL > select distinct sname, pname from student, project where student.pno = project.pno;

TRC : $\{ t \mid \exists s \in \text{STUDENT } (t[sname] = s[sname]$

$\wedge \ \exists p \in \text{PROJECT } (t[pname] = p[pname] \ \wedge s[pno] = p[pno]))\}$

DRC : {<sn, pn> | $\exists < r, s, b, m, p1 > \left(< r, sn, s, b, m, p1 > \ \epsilon\ STUDENT \ \wedge \exists < p2, d > (< p2, pn, d >\right.$

TRC Example : 13

STUDENT

Rollno	Sname	Sem	Branch	Marks	Pno
1	RAM	3	CSE	40	121
2	SHYAM	5	CSE	50	122
3	MOHAN	7	CSE	55	123
4	GOPAL	5	IT	65	121
5	RINKI	3	MECH	40	122
6	PINKI	3	ETC	90	123

PROJECT

Pno	Pname	Duration
121	P1	10
122	P2	20
123	P3	30

$\Pi_{\text{branch, pno}}$ (STUDENT) $\div$ Π_{pno} (PROJECT)

OUTPUT

Branch
CSE

Equivalent SQL Query

SQL > select distinct branch from student s1
where not exists (select pno from project p
where not exists (select branch from student s2
where s1.branch = s2.branch and p.pno = s2.pno));

TRC : $\{\, t \mid \exists\, s \in STUDENT\ (t[branch] = s[branch] \wedge \forall\, p \in PROJECT \rightarrow (s[pno] = p[pno]))\}$

DRC : $\{<b> \mid \exists\, r, sn, s, m, p1\ (< r, sn, s, b, m, p1 > \in STUDENT\ \wedge$

Safety of Expressions

- A domain-relational-calculus expression may generate an infinite relation.
- Suppose that we write the expression

  $\{<r, sn, s, b, m, p> \mid \neg(<\mathrm{r, sn, s, b, m, p}> \in STUDENT)\}$
- There are infinitely many values that are not in STUDENT.
- Most of these values do not even appear in the database! Clearly, we do not wish to allow such expressions.
- This query is known as unsafe query in domain relational calculus.

Expressive Power of Languages

- When the domain relational calculus is restricted to safe expressions, it is equivalent in expressive power to the tuple relational calculus restricted to safe expressions.
- Since we noted earlier that the restricted tuple relational calculus is equivalent to the relational algebra, all three of the following are equivalent :
  - The basic relational algebra (without the extended relational algebra operations)
  - The tuple relational calculus restricted to safe expressions
  - The domain relational calculus restricted to safe expressions

GATE Questions

With regard to the expressive power of the formal relational query languages, which of the following statements is true?

(A) Relational algebra is more powerful than relational calculus.

(B) Relational algebra has the same power as relational calculus.

(C) Relational algebra has the same power as safe relational calculus.

(D) None of the above.

[GATE 2002]

GATE Questions

Which of the following relational calculus expressions is not safe?

(A) $\{t \mid \exists u \in R_1(t[A] = u[A] \wedge \neg \exists s \in R_2(t[A] = s[A])\}$

(B) $\{t \mid \forall u \in R_1(u[A] = \text{“}x\text{”} \Rightarrow \exists s \in R_2(t[A] = s[A]) \wedge s[A] = u[A]))\}$

(C) $\{t \mid \neg (t \in R_1)\}$

(D) $\{t \mid \exists u \in R_1(t[A] = u[A] \wedge \exists s \in R_2(t[A] = s[A])\}$ **[GATE 2001]**

GATE Questions

Which of the following tuple relational calculus expression(s) is/are equivalent to $\forall t \in r(P(t))$?

I. $\neg \exists t \in r(P(t))$

II. $\exists t \notin r(P(t))$

III. $\neg \exists t \in r(\neg P(t))$

IV. $\exists t \notin r(\neg P(t))$

(A) I only

(B) II only

(C) III only

(D) III and IV only

[GATE 2008]

GATE Questions

Consider a database that has the relation schems EMP (EmpId, EmpName, DeptId), and DEPT (DeptName, DeptId). Note that the DeptId can be permited to be NULL in the relation EMP. Consider the following queries on the database expressed in tuple relational calculus.

(I) $\{t \mid \exists u \in EMP(t[EmpName]=u[EmpName] \land \forall v \in DEPT(t[DeptId] \neq v[DeptId]))\}$

(II) $\{t \mid \exists u \in EMP(t[EmpName]=u[EmpName] \land \exists v \in DEPT(t[DeptId] \neq v[DeptId]))\}$

(III) $\{t \mid \exists u \in EMP(t[EmpName]=u[EmpName] \land \exists v \in DEPT(t[DeptId]=v[DeptId]))\}$

Which of the above queries are safe? **[GATE 2017]**

(A) (I) and (II) only

(B) (I) and (III) only

(C) (II) and (III) only

(D) (I) ,(II) and (III)

GATE Questions

Consider the relation employee(name, sex, supervisorName) with name as the key. supervisorName gives the name of the supervisor of the employee under consideration. What does the following Tuple Relational Calculus query produce?

$$\{e.name \mid employee(e) \land (\forall x)\,[\neg employee(x) \lor x.supervisorName \neq e.name \lor x.sex = \text{"male"}\,]\}$$

$$\{e.name \mid employee(e) \land (\neg \exists x)\,[employee(x) \land x.supervisorName = e.name \land x.sex \neq \text{"male"}\,]\}$$

(A) Names of employees with a male supervisor.

(B) Names of employees with no immediate male subordinates.

(C) Names of employees with no immediate female subordinates.

(D) Names of employees with a female supervisor. **[GATE 2007]**

GATE Questions

Consider the relation employee(name, sex, supervisorName) with name as the key. supervisorName gives the name of the supervisor of the employee under consideration. What does the following Tuple Relational Calculus query produce?

{e.name | employee(e) ∧

(∃x) [employee(x) ∧ x.supervisorName = e.name ∧ x.sex ≠ "male"] }

(A) Names of employees with a male supervisor.

(B) Names of employees with no immediate male subordinates.

(C) Names of employees with no immediate female subordinates.

(D) Names of employees with a female supervisor. **[GATE 2007]**

GATE Questions

The relational algebra expression equivalent to the following tuple calculus expression:

$\{t \mid t \in r \wedge (t[A] = 10 \wedge t[B] = 20)\}$

(A) $\sigma_{(A=10 \vee B=20)}(r)$

(B) $\sigma_{(A=10)}(r) \cup \sigma_{(B=20)}(r)$

(C) $\sigma_{(A=10)}(r) \cap \sigma_{(B=20)}(r)$

(D) $\sigma_{(A=10)}(r) - \sigma_{(B=20)}(r)$

[GATE 1999]

For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JlTp1iim6aOTRuw?sub_confirmation=1