Database Management System

UNIT 2

Relational data model and language

Lecture-1

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Today's Target

- Constraints in DBMS
- AKTU PYQs

Constraints in DBMS

- Relational constraints are the restrictions imposed on the database contents and operations.
- They ensure the correctness of data in the database.

Types of Constraints in DBMS-

- Domain constraint.
- Tuple Uniqueness constraint.
- Key constraint.
- NOT NULL
- UNIQUE
- DEFAULT
- CHECK

1. Domain Constraint

- Domain constraint defines the domain or set of values for an attribute.
- It specifies that the value taken by the attribute must be the atomic value from its domain.

Student

STU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
S004	Rahul	A

2. Tuple Uniqueness Constraint-

- Tuple Uniqueness constraint specifies that all the tuples must be necessarily unique in any relation.

Student

TU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
S004	Rahul	20

- This relation satisfies the tuple uniqueness constraint since here all the tuples are unique.

Student

TU_ID	Name	Age
S001	Akshay	20
S001	Akshay	20
S003	Shashank	20
S004	Rahul	20

- This relation does not satisfy the tuple uniqueness constraint since here all the tuples are not unique.

3. Key Constraint-

Key constraint specifies that in any relation-

- All the values of primary key must be unique.
- The value of primary key must not be null.

student

STU_ID	Name	Age
S001	Akshay	20
S001	Abhishek	21
S003	Shashank	20
S004	Rahul	20

4 NOT NULL:

- NOT NULL constraint makes sure that a column does not hold NULL value
- When we do not provide value for a particular column while inserting a record into a table, it takes NULL value by default.
- By specifying NULL constraint, we make sure that a particular column cannot have NULL values

5. UNIQUE:

- UNIQUE constraint enforces a column or set of columns to have unique values.
- If a column has a unique constraint, it means that particular column cannot have duplicate values in a table

6. DEFAULT

The DEFAULT constraint provides a default value to a column when inserting a record into a table there is no value provided while inserting record into table

7. CHECK:

- The CHECK constraint is used to limit the value range that can be placed in a column.
- If you define a CHECK constraint on a column it will allow only certain values for this column.

Integrity constraints

- It provide a way of ensuring the changes made by the authorized user do not result into the loss of data.
- Domain constraint
- Key constraint
- Entity Integrity constraint
- Referential Integrity constrain

1. Entity Integrity Constraint-

- It specifies that no attribute of primary key must contain a null value in any relation.
- This is because the presence of null value in the primary key violates the uniqueness property.

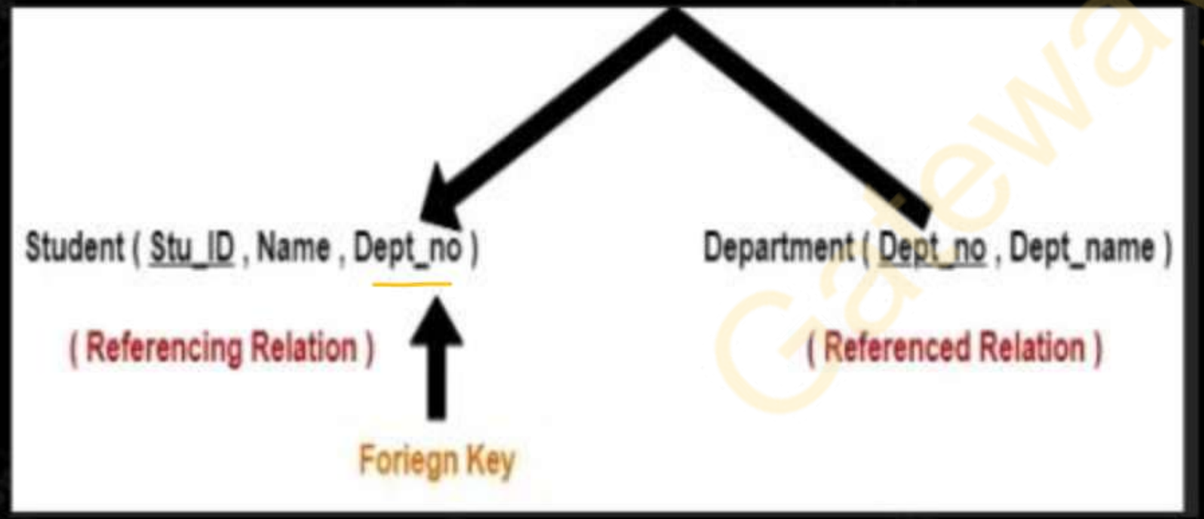
Student

TU_ID	Name	Age
S001	Akshay	20
S002	Abhishek	21
S003	Shashank	20
..	Rahul	2

This relation does not satisfy the entity integrity constraint as here the primary key contains a NULL value

2. Referential Integrity Constraint

- This constraint is enforced when a foreign key references the primary key of a relation.
- It specifies that all the values taken by the foreign key must either be available in the relation of the primary key or be null.



Student (Referencing)

STU_ID	Name	Dept_no
S001	Akshay	D10
S002	Abhishek	D10
S003	Shashank	D11
S004	Rahul	D14

Department (Referenced Relation)

Dept_no	Dept_name
D10	ASET
D11	ALS
D12	ASFL
D13	ASHS

Referential Integrity Constraint Violation-

There are following three possible causes of violation of referential integrity constraint-

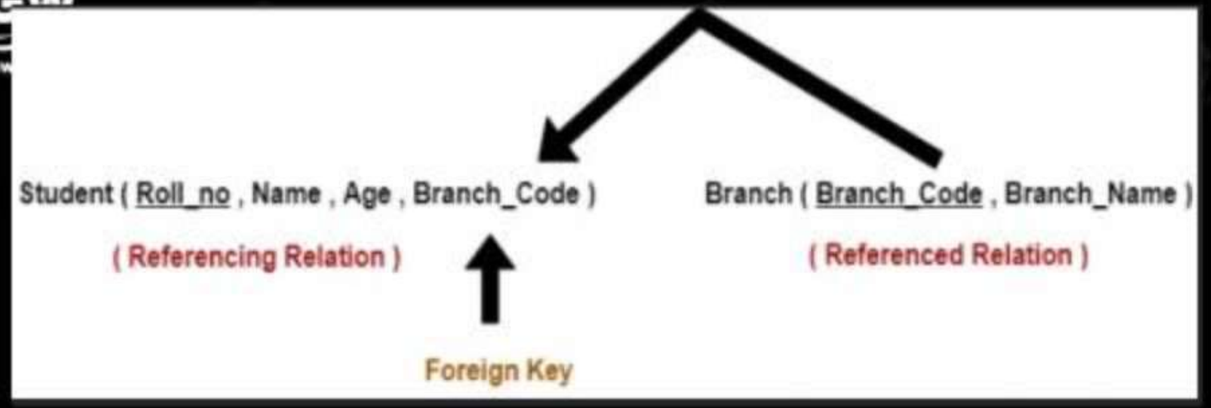
Cause-01: Insertion in a referencing relation.

Cause-02: Deletion from a referenced relation.

Cause-03: Updation in a referenced relation.

Cause-01: Insertion in a Referencing Relation-

- It is allowed to insert only those values in the referencing attribute which are already present in the value of the referenced attribute.
- Inserting a value in the referencing attribute which is not present in the value of the referenced attribute violates the referential integrity constraint



Student

(T-k)

Roll_no	Name	Age	Branch_Code
1	Rahul	22	CS
2	Anjali	21	CS
3	Teena	20	IT
4	Rahul	20	ME

Branch

Branch_Code	Branch_Name
CS	Computer Science
EE	Electronics Engineering
IT	Information Technology
CE	Civil Engineering

- In relation "Student", we can not insert any student having branch code ME (Mechanical Engineering).
- This is because branch code ME is not present in the relation "Branch".

Cause-02: Deletion from a Referenced Relation-

- It is not allowed to delete a row from the referenced relation if the referencing attribute uses the value of the referenced attribute of that row.
- Such a deletion violates the referential integrity constraint.

student(referencing relation)

F.K

Roll_no	Name	Age	Branch_Code
1	Rahul	22	CS
2	Anjali	21	CS
3	Teena	20	IT

branch(referenced Relation)

Branch_Code	Branch_Name
CS	Computer Science
EE	Electronics Engineering
IT	Information Technology
CE	Civil Engineering

- We can not delete a tuple from the relation "Branch" having branch code 'CS'.
- This is because the referencing attribute "Branch_Code" of the referencing relation "Student" references the value 'CS'.
- However, we can safely delete a tuple from the relation "Branch" having branch code 'CE'.
- This is because the referencing attribute "Branch_Code" of the referencing relation "Student" does not uses this value

Handling violation

1. On Delete Cascade.
2. Aborting or deleting the request for a deletion from the referenced relation if the value is used by the referencing relation.
3. The value being deleted from the referenced relation to NULL or some other value in the referencing relation if the referencing attribute uses that value.

STUDENT

Sid	Sname
S101	A
S102	B
S103	C

Referenced Relation

Insert – easily insert new tuple

Delete-

1.On delete cascade

(S101 ,A) simultaneously (b101,x,s101)

2. On delete set null

S101 ,A) simultaneously (b101,x, NULL)

3. ON DELETE SET DEFAULT

S101 ,A) simultaneously (b101,x, 0)

Book

Bid	Sname	Sid
b101	X	S101
b102	Y	S102
b103	Z	S103
b104	W	-

Referencing Relationship

Insert no easily not possible (referential intregity constraints must be maintained

Delete- easily delete

Cause-03: Updation in a Referenced Relation-

- It is not allowed to update a row of the referenced relation if the referencing attribute uses the value of the referenced attribute of that row.
- Such an updation violates the referential integrity constraint

student(referencing relation)

Roll_no	Name	Age	Branch_ Code
1	Rahul	22	CS
2	Anjali	21	CS
3	Teena	20	IT

branch(referenced Relation)

Branch_Code	Branch_Name
CS	Computer Science
EE	Electronics Engineering
IT	Information Technology
CE	Civil Engineering

- We can not update a tuple in the relation "Branch" having branch code 'CS' to the branch code 'CSE'.
- This is because referencing attribute "Branch_Code" of the referencing relation "Student" references the value 'CS'.

Handling violation

1. On Update Cascade
2. aborting or deleting the request for an updation of the referenced relation if the value is used by the referencing relation.

STUDENT

Sid	Sname
S101	A
S102	B
S103	C

Referenced Relation

UPDATE

S-101 updated in place of S101

Then this update done simultaneously on

book table also

Book

Bid	Sname	Sid
b101	X	S101
b102	Y	S102
b103	Z	S013
b104	W	-

Referencing Relationship

Update

Not possible due to referential integrity constraints

AKTU PYQS

Q1	What are the different integrity constraints.	AKTU 2021-22 AKTU 2022-23 AKTU 2020-21
Q.3	What is entity integrity constraints.	AKTU 2017-18
Q.2	Define constraints with its type.	AKTU 2018-19

Database Management System

UNIT 2

Relational data model and language

Lecture-2

Today's Target

- Relational Algebra
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

A (PRIMARY KEY)	C (FOREIGN KEY)
2	4
3	4
4	3
5	2
7	2
9	5
6	4

(AKTU)

ON delete cascade (2,4) how many tuple further deleted

(5,2) (7,2) (9,5)

Relational Algebra

- Relational Algebra is a procedural query language.
- Relational algebra mainly provides a theoretical foundation for relational databases and SQL.
- The main purpose of using Relational Algebra is to define operators that transform one or more input relations into an output relation.

Fundamental Operators

These are the basic/fundamental operators used in Relational Algebra.

- Selection(σ) (Unary operator)
- Projection(π) (Unary Operator)
- Union(U)
- Set Difference(-)
- Set Intersection(∩)
- Rename(ρ)
- Cartesian Product(X)

1.Selection(σ)

- The SELECT operation is used to select a subset of the tuples from a relation that satisfy a selection condition.
- The SELECT operation can also be visualized as a horizontal partition of the relation into two sets of tuples

1.Those tuples that satisfy the condition and are selected.

2. Those tuples that do not satisfy the condition and are discarded.

SYNTAX

$\sigma_c(R)$ $\quad \sigma_c(R)$

- The symbol σ (sigma) is used to denote the SELECT operator.
- c is the selection condition which is a Boolean expression(condition), we can have a single condition like Roll= 3 or a combination of conditions like X>2 AND Y<1
- R is a relational algebra expression, whose result is a relation.

Student

Roll	Name	Marks
1	A	60 ✓
2	B	70 ✓
3	C	23
4	D	60 ✓
5	E	25

tup1, tup2, tup3, 4, 5

Marks ≥ 60

Roll	Name	Marks
1	A	60
4	D	60
2	B	70

— O/P

Roll	Name	Marks
3	C	23
5	E	25

$\sigma_{(Marks \geq 60)}(Student)$

O/P

Roll	Name	Marks
3	C	23

$\sigma_{(Roll=3)}(Student)$

Roll	Name	Department	Fees	Team
1	Bikash	CSE	22000	A
2	Josh	CSE	34000	A
3	Kevin	ECE	36000	C
4	Ben	civil	56000	D
5	Rani	ECE	500	E

Student

1. Select all the students of Team A :

$\sigma_{\text{Team = 'A'}}$ (Student)

Roll	Name	Department	Fees	Team
1	Bikash	CSE	22000	A
2	Josh	CSE	34000	A

2.Select all the students of department ECE whose fees are greater than and equal to 10000 and belong to a Team other than A.

σ Department = 'ECE' and Fees >= 10000(σTeam != 'A' (Student))

Roll	Name	Department	Fees	Team
3	Kevin	ECE	36000	C

Note:

It is also called unary operator, which means it is applied to a single relation only

The selection operation is commutative that is,

$\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$

σ Department = 'ECE' and Fees >= 10000(σTeam != 'A' (Student))

σ Team!='A'(Department = 'ECE' and Fees >= 10000 (Student))

2.Projection(π)

- PROJECT operation, selects certain columns from the table and discards the other columns.
- The result of the PROJECT operation can hence be visualized as a vertical partition of the relation into two relations:

1. One has the needed columns (attributes) and contains the result of the operation. 2.the other contains the discarded columns

Syntax:

π_A (R)

- 'A' is the attribute list, it is the desired set of attributes from the attributes of relation(R),
- symbol 'π(pi)' is used to denote the Project operator,
- R is generally a relational algebra expression, which results in a relation.

(Duplicate values not allowed

Student

Roll	Nam	Mobile	Gendi
1	A	23	M
2	B	42	F
3	C	73	M
4	D	63	F
5	C	29	M

$\uparrow_{Roll, Name}(Student)$

→

Roll	Name
1	A
2	B
3	C
4	D
5	F

$\uparrow_{Gender}(Student)$

Gender
M
F

(this is Non-P.k or we can not with P.k)

$\uparrow_{Roll, Gender}$

Roll	Gender
1	M
2	F
3	M
4	F
5	M

Student

Roll	Name	Gendu
1	A	M
2	A	M
3	B	F
4	C	F
5	B	M

$\Pi_{Name, Gendu}$

Name	Gender
A	M
B	F
B	M
C	F

(Main hot tuple) = < tuples Output Relation

Employees

Employee ID	Name	Department	Salary
1	Alice	HR	60000
2	Bob	IT	70000
3	Charlie	IT	75000
4	David	HR	62000
5	Eva	Finance	80000

Select the name and salary of the all employee

$\pi_{Name, Salary}$ (Employees)

SQL : SELECT Name, Salary FROM Employees;

Name	Salary
Alice	60000
Bob	70000
Charlie	75000
David	62000
Eva	80000

Select all the data of the IT department employee

$\sigma_{Department='IT'}$ (Employees)

SQL: SQL Query : SELECT * FROM Employees

WHERE Department = 'IT';

EmployeeID

Employee ID	Name	Department	Salary
2	Bob	IT	70000
3	Charlie	IT	75000

Both projection and selection are used

$\Pi_{Name}(\sigma_{Department='IT'}(Employees))$

EmployeID	Name	Department	Salary
2	Bob	IT	70000
3	Charlie	IT	75000

Name
Bob
Charlie

Final Ans

Student

Roll	Name	Marks
1	A	60
2	A	60
3	B	70
4	C	20
5	D	30

$\Pi_{Name}(\sigma_{Marks >= 60}(Student))$

Roll	Name	Marks
1	A	60
2	A	60
3	B	70

Name
A
B

$\Pi_{Roll, Name}(\sigma_{marks >= 60}(Student))$

Roll	Name
1	A
2	A
3	B

Employee

Employee ID	Name	Department	Salary
1	Alice	HR	60000
2	Bob	IT	70000
3	Charlie	IT	75000
4	David	HR	62000
5	Eva	Finance	75000
6	Charl	IT	75000

$\pi_{Department, Salary}$ (Employees)

Department	Salary
HR	60000
IT	70000
IT	75000
HR	62000
Finance	75000

Employee

π_{Salary} (Employees)

Salary
60000
70000
75000
62000

- If the attribute list includes only nonkey attributes of R, duplicate tuples are likely to occur.

- The PROJECT operation removes any duplicate tuples, so the result of the PROJECT operation is a set of tuples
- The number of tuples in a relation resulting from a PROJECT operation is always less than or equal to the number of tuples in R.

Employee

Employee ID	Name	Department	Salary
1	Alice	HR	60000
2	Bob	IT	70000
3	Charlie	IT	75000
4	David	HR	62000
5	Eva	Finance	75000
6	Charl	IT	75000

$\pi_{Department}\ (\pi_{Department, Salary}\ (Employees)$

Department	Salary
HR	60000
IT	70000
IT	75000
HR	62000
Finance	75000

Department
HR
IT
Finance

Employee

Employee ID	Name	Department	Salary
1	Alice	HR	60000
2	Bob	IT	70000
3	Charlie	IT	75000
4	David	HR	62000
5	Eva	Finance	75000
6	Charl	IT	75000

$\pi_{Department,\ salary}\ (\pi_{Department}\ (Employees))$

Department
HR
IT
Finance

Not able to get the answer that shows, projection operator is not commutative

Category	Selection	Projection
Other Names	The selection operation is also known as horizontal partitioning.	The Project operation is also known as vertical partitioning.
Use	It is used to choose the subset of tuples from the relation that satisfies the given condition mentioned in the syntax of selection.	It is used to select certain required attributes, while discarding other attributes.
Partitioning	It partitions the table horizontally.	It partitions the table vertically.

Category	Selection	Projection
Operator Symbol	Select operator is denoted by Sigma symbol.	Project operator is denoted by Pi symbol.
Commutative	Selection is commutative.	Projection is not commutative.
Column Selection	Select is used to select all columns of a specific tuple.	Project is used to select specific columns.
Which used first	**The selection operation is performed before projection (if they are to be used together).**	**The projection operation is performed after selection (if they are to be used together).**

Student

Name	Roll	Mobile	Address	Marks
A	1	9263	A1	70
A	2	9263	AL	60
B	3	4279	A2	70
C	4	4478	A3	80
D	5	5867	A4	90

marks	Name
70	A
60	A
70	B

Name	Marks
A	70
A	60
B	70

70
60

① Select those address Whose Name is A and mark <= 60

$$\Pi_{address}\left(\sigma_{Name='A' \text{ and } marks <= 60}(Student)\right)$$

② Select Marks of all Student Whose marks <= 70

$$\Pi_{marks}\left(\sigma_{marks <= 70}(Student)\right)$$

③ Select Name & Marks of all Students Whose marks <= 70

$$\Pi_{Name, Marks}\left(\sigma_{marks <= 70}(Student)\right)$$

$$\Pi_{marks, Name}\left(\sigma_{marks <= 70}(Student)\right)$$

Database Management System

UNIT 2

Relational data model and language

Lecture-3

Today's Target

- Relational Algebra
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Union compatible

- Two relations R(A1, Az, ... , An) and S(B 1, Bz, ... , Bn) are said to be union compatible if they have

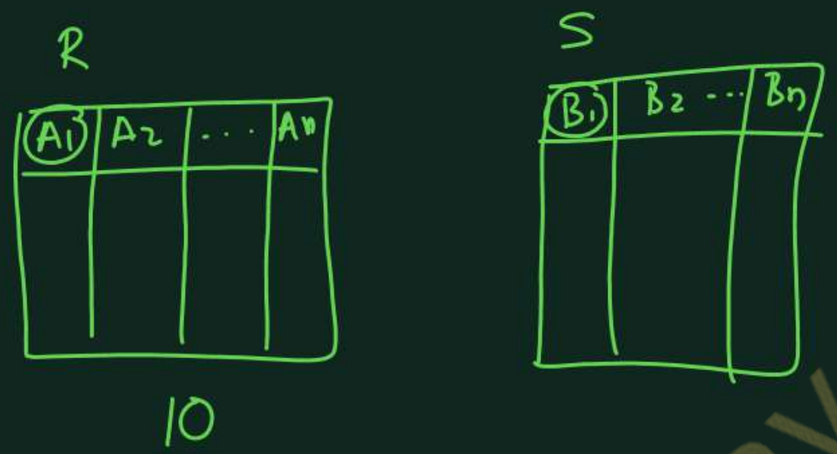
1. the same degree n
2. If $dom(A_i)=dom(B_i)$ $1<=i<=n$

- This means that the two relations have the same number of attributes, and each corresponding pair of attributes has the same domain.

We can define the three operations UNION, INTERSECTION, and SET DIFFERENCE on two union-compatible relations R and S follows-

1.union:

- The result of this operation, denoted by R U S,
- It is a relation that includes all tuples that are either in R or in S or in both Rand S.
- Duplicate tuples are eliminated



Depositor	
CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

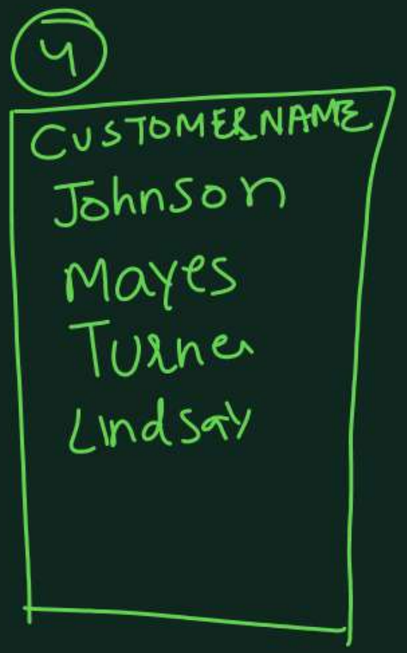
BORROW	
CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

(1) $\Pi_{CUSTOMER\_NAME}(Depositor) \cup \Pi_{CUSTOMER\_NAME}(Borrower) =$
$\Pi_{CUSTOMER\_NAME}(Borrower) \cup \Pi_{CUSTOMER\_NAME}(Depositor)$

(2) $\Pi_{CUSTOMER\_NAME}(Borrower) \cap \Pi_{CUSTOMER\_NAME}(Depositor)$ — Intersection

(3) $\Pi_{CUSTOMER\_NAME}(Borrower) - \Pi_{CUSTOMER\_NAME}(Depositor)$ — Set Difference | Minus

(4) $\Pi_{CUSTOMER\_NAME}(Depositor) - \Pi_{CUSTOMER\_NAME}(Borrower)$



CUSTOMER_NAME
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson
Curry
Williams
Mayes
Johnson

1 (Union)

2

CUSTOMER_NAME
Smith
Jones

(Intersection)

3

CUSTOMER_NAME
Jackson
Hayes
Willians
Curry

2.Intersection:

The result of this operation, denoted by $R \cap S$, is a relation that includes all tuples that are in both R and S.

3. set difference (or MINUS):

The result of this operation, denoted by R – S, is a relation that includes all tuples that are in R but not in S.

Q.1 To retrieve the social security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5.

$DEP5\_EMP \leftarrow \sigma_{DNO=5}(EMPLOYEE)$

$Result1 \leftarrow \pi_{SSN}(DEP5\_EMP)$

$Result2(SSN) \leftarrow \pi_{SUPERSSN}(DEP5\_EMP)$

$Result \leftarrow Result1 \cup Result2$

Result1

SSN
123456789
333445555
666884444
453453453

Result 2

SSN
333445555
888665555

SSN
123456789
333445555
666884444
453453453
888665555

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Cartesian Product

- CARTESIAN PRODUCT operation-also known as CROSS PRODUCT or CROSS JOIN-which is denoted by x.
- This is also a binary set operation, but the relations on which it is applied do not have to be union compatible.
- This operation is used to combine tuples from two relations in a combinatorial fashion.
- The operation applied by itself is generally meaningless.
- It is useful when followed by a selection that matches values of attributes coming from the component relations.

EMPLOYEE		
EMP_ID	EMP_NAME	EMP_DEPT
1	Smith	A
2	Harry	C
3	John	B

DEPARTMENT	
DEPT_NO	DEPT_NAME
A	Marketing
B	Sales
C	Legal

EMPLOYEE X DEPARTMENT

EMP_ID	EMP_NAME	EMP_DEPT	DEPT_NO	DEPT_NAME
1	Smith	A	A	Marketing
1	Smith	A	B	Sales
1	Smith	A	C	Legal
2	Harry	C	A	Marketing
2	Harry	C	B	Sales
2	Harry	C	C	Legal
3	John	B	A	Marketing
3	John	B	B	Sales
3	John	B	C	Legal

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Q.2 we want to retrieve a list of names of each female employee's dependent

$$\text{FemaleEMP} \leftarrow \sigma_{\text{Sex}=\text{'F'}}(\text{EMPLOYEE})$$

$$\text{EMPNAME} \leftarrow \pi_{\text{FName, LName, SSN}}(\text{Femal\_EMP})$$

$$\text{EMPDEPENDENTS} \leftarrow \text{EMPNAME} \times \text{DEPENDENT}$$

$$\text{ACTUAL\_DEPENDENT} \leftarrow \pi_{\text{SSN}=\text{ESSN}}(\text{EMPDEPENDENT})$$

$$\text{Result} \leftarrow \pi_{\text{FNAME, LNAME, DEPENDENT\_NAME}}(\text{ACTUAL\_DEPENDENT})$$

EMP_DEPENDENTS

FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	• • •
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	• • •
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	• • •
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	• • •
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	• • •
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	• • •
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	• • •
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	• • •
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	• • •
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	• • •
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	• • •
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	• • •
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	• • •
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	• • •
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	• • •
Joyce	English	453453453	333445555	Alice	F	1986-04-05	• • •
Joyce	English	453453453	333445555	Theodore	M	1983-10-25	• • •
Joyce	English	453453453	333445555	Joy	F	1958-05-03	• • •
Joyce	English	453453453	987654321	Abner	M	1942-02-28	• • •
Joyce	English	453453453	123456789	Michael	M	1988-01-04	• • •
Joyce	English	453453453	123456789	Alice	F	1988-12-30	• • •
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	• • •

ACTUAL_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	...
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT	FNAME	LNAME	DEPENDENT_NAME
	Jennifer	Wallace	Abner

Branch		
Branch_name	Branch_city	Assests

Account		
Account_no	Branch_name	balance

Loan		
Loan_no	Branch_name	amount

Customer		
Customer_name	Cust_street	Cust_city

Depositor	
Customer_name	Account_no

Borrower	
Customer_name	Loan_no

Q.3 Find the details of account having balance>=1000

$\sigma_{balance >= 1000}$ (Account)

NOTE
∧ - and
∨ - OR

Q.4 Find the details of the customers who lives in Delhi.

$\sigma_{Cust\_city = 'Delhi'}$ (Customer)

Q.5 Find the details of those loan having amount <=500 and from north Delhi

$\sigma_{amount <= 500 \wedge Branch\_name = 'NOLTH Delhi'}$ (Loan)

Q.6 Find those branch details which are in Delhi and having assets more than 10,00,00.

$\sigma_{Branch\_city = 'Delhi' \wedge Assests > 10,00,00}$ (Branch)

Q.7 Find those branch details which are in Delhi or having assets more than 10,00,00.

$\sigma_{Branch\_city = 'Delhi' \vee Assests > 10,00,00}$ (Branch)

Q.8 Find those account number where balance is less than 1000.

$$\Pi_{account\ no}\left(\sigma_{balance<1000}(Account)\right)$$

Q.9 Find the loan number which are from CP Branch plus amount>1000.

$$\Pi_{loan\_no}\left(\sigma_{Branch\_name='CP' \wedge amount>1000}(loan)\right)$$

Q.10 Find all branch name of bank.

$$\Pi_{Branch\_name}(Branch)$$

Q.11 Find the name of all customers who have loan

$$\Pi_{customer\_name}(Borrower)$$

Q.12 Find all the name of the customer, who are depositor or Borrower Loan

$$\Pi_{customer\_name}(Borrower) \cup \Pi_{customer\_name}(Depositor)$$

Q Find the customer name who are Depositor but not having any loan

$\Pi_{customer\_name}(Depositor) - \Pi_{Customer\_name}(Borrower)$

Q Find the customer name who borrow loan but not depositor

$\Pi_{(customer\_name)}(Borrower) - \Pi_{(customer\_name}(Depositor)$

Q Find the Branch name who have account not having loan

$\Pi_{Branch\_name}(Account) -$

$\Pi_{Branch\_name}(loan)$

Database Management System

UNIT 2

Relational data model and language

Lecture-4

Today's Target

- Relational Algebra
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Derived Operator

1. Join
   - Inner
   - Outer
2. Division

Join ⋈

- Join is a combination of a Cartesian product followed by a selection process.
- A Join operation pairs two tuples from different relations, if and only if a given join condition is satisfied.

CARTESIAN PRODUCT

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
D	E	F
6	7	2
4	5	8
7	6	2

A	B	C	D	E	F
1	2	3	6	7	2
1	2	3	4	5	8
1	2	3	7	6	2
4	5	6	6	7	2
4	5	6	4	5	8
4	5	6	7	6	2
7	8	9	6	7	2
7	8	9	4	5	8
7	8	9	7	6	2

Cartesian product $R_1 \times R_2$

A	B	C	D	E	F
4	5	6	4	5	8
7	8	9	6	7	2
7	8	9	4	5	8
7	8	9	7	6	2

$R_1 \underset{c>d}{\bowtie} R_2$

Cartesian product +selection (those tuple in which c>d)

$\sigma_{c>d}(R_1 \times R_2)$

JOIN

1.Inner join(Matching tuples)

- theta join
- Equi join
- Natural join
- Semi join
- Anti join

2. Outer join(Matching and non matching tuples)

- Left outer join
- Right outer join
- Full outer join

Theta (θ) Join $R1 \bowtie_{\theta} R2$

$(\theta) = \{<, <=, >, >=, ==, \neq\}$

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
D	E	F
6	7	2
4	5	8
7	6	2

(Equality)

A	B	C	D	E	F
4	5	6	4	5	8
7	8	9	6	7	2
7	8	9	4	5	8
7	8	9	7	6	2

$R1 \bowtie_{C>D} R2$

EQUI JOIN $R1 \bowtie_{C=D} R2$

A	B	C	D	E	F
4	5	6	6	7	2

- When Theta join uses only **equality** comparison operator, it is said to be equijoin

Degree and cardinality

$|R_1|_d + |R_2|_d = |R_1 + R_2|_d$ (theta join)

$|R_1|_d + |R_2|_d = |R_1 + R_2|_d$ (Equi join)

Cardinality (No of tuples in a Relation)

$$0 \leq |R_1 \bowtie_\theta R_2|_c \leq |R_1| \times |R_2|$$

Equi join

$$0 \leq |R_1 \bowtie R_2|_c \leq |R_1| \times |R_2|$$

R1	
A	B
1	2
3	4

degree = no of attribute in a Relation

R2	
C	D
5	6
7	8

$R_1 \bowtie_{B>C} R_2$ → 0

$R_1 \bowtie_{B<C} R_2$

A	B	C	D
1	2	5	6
1	2	7	8
3	4	5	6
		7	8

3 4

Degree and cardinality

Cardno	name	bid	DOJ	Cardno
1	A	23	6 Jul	1
2	abc	26	JJ~	
2	abc	27	8 J	

QUERIES

1. book(<u>bid</u>, title year_pub)
2. User(<u>card_no</u>,name)
3. Borrow(<u>bid,card_no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.1 Find the bid of all books issued to 'abc'

$$\Pi_{bid}\left(\sigma_{name='abc'}\left(User \bowtie_{user.cardno = Borrow.cardno} Borrow\right)\right)$$

Natural join ⋈

- Natural join does not use any comparison operator.
- It does not concatenate the way a Cartesian product does.
- We can perform a Natural Join only if there is at least one common attribute that exists between two relations.
- In addition, the attributes must have the same name/similar and domain.

Course		
ID	Course	Dept
CS01	Database	CS
ME01	Mechanics	ME
EE01	Electronics	EE

HoD	
Dept	Head
CS	Alex
ME	Maya
EE	Mira

Courses ⋈ HoD			
Dept	**CID**	**Course**	**Head**
CS	CS01	Database	Alex
ME	ME01	Mechanics	Maya
EE	EE01	Electronics	Mira

R1		
A	**B**	**C**
1	2	3
4	5	6
7	8	9

Difference Between-

Natural Join

R2		
C	**D**	**E**
3	6	9
9	5	4
8	7	6

$R_1 \bowtie R_2$

A	B	C	D	E
1	2	3	6	9
7	8	9	5	4

$R_1 \underset{R_1.C = R_2.C}{\bowtie} R_2$ {Equi join}

A	B	R1.C	R2.C	D	E
1	2	3	3	6	9
7	8	9	9	5	4

If two attribute are same

R1		
A	B	C
1	2	3
4	5	6
7	8	9

$R_1 \bowtie R_2$

R2		
B	C	D
2	3	9
6	5	4
8	9	6

A	B	C	D
1	2	3	9
7	8	9	6

Natural join work as intersection/ALL three attribute are same

$R_1 \bowtie R_2 = R_1 \cap R_2$

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
A	B	C
1	2	3
6	5	4
8	9	6

A	B	C
1	2	3

When natural join work as Cartesian product or when no attribute are same

$R_1 \bowtie R_2 = R_1 \times R_2$

R1		
A	B	C
1	2	3
4	5	6

R2		
D	E	F
1	2	3
6	5	4

A	B	C	D	E	F
1	2	3	1	2	3
1	2	3	6	5	4
4	5	6	1	2	4
4	5	6	6	5	4

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
C	D	E
3	6	9
9	8	2
7	6	3

green colored → Dangling tuples

A	B	C	D	E
1	2	3	6	9
7	8	9	8	2

$R_1 \bowtie R_2$

Degree and cardinality

$|R_1|_d + |R_2|_d - m = |R_1 \times R_2|_d$ — degree

m = No of common tuples

$0 \leq |R_1 \bowtie R_2| \leq |R_1 \times R_1|_c$

(Cardinality)

$R_1 \bowtie R_2$

R1		
A	B	C
1	2	3

R2		
C	D	E
6	7	8

R1		
A	B	C
1	2	3
1	6	3
8	9	3

R2		
C	D	E
3	6	9
3	5	2
3	8	2

A	B	C	D	E
1	2	3	6	9
1	2	3	5	2
1	2	3	8	2
1	6	3	6	9
1	6	3	5	2
1	6	3	8	2
8	9	3	6	9
8	9	3	5	2
8	9	3	8	2

Cartesian product

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.1 Find titles of all books supplies by abc.

$$\Pi_{title}\left(\sigma_{sname='abc'}\left(book \bowtie Supply\right)\right)$$

Q.2 Find bookid of all books issue abc.

$$\Pi_{bookid}\left(\sigma_{name='abc'}\left(User \bowtie Borrow\right)\right)$$

Q.3 Find title of all books issue to abc.

$$\Pi_{title}\left(\sigma_{name='abc'}\left(book \bowtie Borrow \bowtie User\right)\right)$$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no, name, city)
3. Borrow(bid, card_no, DOI)
4. Supply(bid, sname, price, dos)

Q.4 Find the name of all those supplier who have supplied book issue to 'abc'.

$$\Pi_{sname}\left(\sigma_{name='abc'}\left(Borrow \bowtie Supply \bowtie User\right)\right)$$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q. 5 Find the name of all those supplier who have supplies book of same title as issue to abc.

$$\Pi_{Sname}\Big(\big(\Pi_{title}(\sigma_{name='abc'}(book \bowtie borrow \bowtie usa))\big) \bowtie \big(\Pi_{title,\,Sname}(Supply \bowtie book)\big)\Big)$$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.6 Find the name of all user who lives in same city as that of abc.

$$\Pi_{name}\left(\Pi_{city}\left(\sigma_{name='abc'}(User)\right) \bowtie \Pi_{name,city}\left(\sigma_{name<>'abc'}(User)\right)\right)$$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q. 7 find the name of all those borrowers who have issued book on same date as issued by abc.

$$\uparrow_{name}\left(\uparrow_{DOI}\left(\sigma_{name='abc'}(User \bowtie borrow)\right) \bowtie \uparrow_{name,DOI}\left(\sigma_{name<>abc}(User \bowtie borrow)\right)\right)$$

SEMI JOIN ⋉

$R_1 \ltimes R_2$

A	B	C
1	2	3
4	5	6

ANTI JOIN ▷

C	D	E
3	6	9
6	7	3

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
C	D	E
3	6	9
4	5	2
6	7	3

$R_1 \triangleright R_2$

A	B	C	D	E
1	2	3	6	9
4	5	6	7	3

Outer Joins

- Theta Join, Equijoin, and Natural Join are called inner joins.
- An inner join includes only those tuples with matching attributes and the rest are discarded in the resulting relation.
- Therefore, we need to use outer joins to include all the tuples from the participating relations in the resulting relation. There are three kinds of outer joins – left outer join, right outer join, and full outer join.

R1		
A	B	C
1	2	3
4	5	6
7	8	9

R2		
C	D	E
3	6	7
8	5	4
6	3	2

NATURAL JOIN $R_1 \bowtie R_2$

A	B	C	D	E
1	2	3	6	7
4	5	6	3	2

Left outer join $R_1 \mathbin{⟕} R_2$

A	B	C	D	E
1	2	3	6	7
4	5	6	3	2
7	8	9	-	-

Right outer join $R_1 \mathbin{⟖} R_2$

A	B	C	D	E
1	2	3	6	7
4	5	6	3	2
-	-	8	5	2

Full outer join $\mathbin{⟗}$

A	B	C	D	E
1	2	3	6	7
4	5	6	3	2
7	8	9	-	-
-	-	8	5	2

Database Management System

UNIT 2

Relational data model and language

Lecture-5

Today's Target

- Relational Algebra Numericals
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Give the following queries in the relational algebra using the relational schema :

student(id, name)

Enrolled (id, code)

subject(code, lecturer) (AKTU 2019-20)

1. What are the names of students enrolled in cs3020

$$\Pi_{name}\left(\sigma_{code='CS302'}(Student \bowtie Enrolled)\right)$$

2Which subjects is Hector taking

$$\Pi_{Code}\left(\sigma_{nam='Hector'}(Student \bowtie Enrolled)\right)$$

Give the following queries in the relational algebra using the relational schema :

student(id, name)

Enrolled (id, code)

subject(code, lecturer)

3. Who teaches cs1500 ?

$$\Pi_{lecturer}\left(\sigma_{code='cs1500'}(Subject)\right)$$

4.Who teaches cs1500 or cs3020

$$\Pi_{lecturer}\left(\sigma_{code='cs1500' \vee code='cs3020'}(Subject)\right)$$

Give the following queries in the relational algebra using the relational schema :

student(id, name)

Enrolled (id, code)

subject(code, lecturer)

5. Who teaches at least two different subjects ?

$$\Pi_{S1.lecturer}\left(\rho_{S1}(Subject) \bowtie_{(S1.lecturer = S2.lecturer \wedge S1.code <> S2.code)} \rho_{S2}(Subject)\right)$$

6. What are the names of students in cs1500 or cs307

$$\Pi_{name}\left(\sigma_{code='CS1500'}(Student \bowtie Enrolled)\right) \cup \Pi_{name}\left(\sigma_{code='CS307'}(Student \bowtie Enrolled)\right)$$

Give the following queries in the relational algebra using the relational schema :

student(id, name)

Enrolled (id, code)

subject(code, lecturer)

7.What are the names of students in both cs 1500 and cs1200

$$\Pi_{name}\left(\sigma_{code='CS1500'}\left(Student \bowtie Enrolled\right)\right) \cap \Pi_{name}\left(\sigma_{code='CS1200'}\left(Student \bowtie Enrolled\right)\right)$$

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following :

1.List the codes of courses in which at least one student is registered (registered courses).

Π_{code} (Course $\bowtie$ Registered)

or

Π_{code} (Registered)

2. List the title of registered courses

Π_{title} (Course $\bowtie$ Registered)

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)(AKTU 2015-16/2017-18)

Use relational algebra to answer the following :

3. List the codes of courses for which no student is registered.

$\Pi_{code}(Course) - \Pi_{code}(Course \bowtie Registered)$

4. The titles of courses for which no student is registered

$\Pi_{title}(Course) - \Pi_{title}(Course \bowtie Registered)$

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following

5. Name of students and the titles of courses they registered to

$\Pi_{name,\ title}(Course \bowtie Registered \bowtie Student)$

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following

6. SSNs of students who are registered for both database systems and analysis of algorithms

$$\Pi_{SSN}\left(\sigma_{title = \text{'Database System'}}\left(Course \bowtie Registered \bowtie Student\right)\right) \cap \Pi_{SSN}\left(\sigma_{title = \text{'Analysis of Alg'}}\left(Course \bowtie Registered \bowtie Student\right)\right)$$

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following

7 The name of students who are registered for ~~both~~ for database systems ~~and~~ or analysis of algorithms.

$\Pi_{name}(\sigma_{title = Data\ base\ system}(Student \bowtie Course \bowtie Registered)) \cup$

$\Pi_{name}(\sigma_{title = 'Analysis\ of\ Algo'}(Student \bowtie Course \bowtie Registered))$

Division operator

R1(R)	
W	X
P	A
P	B
Q	A
Q	B
P	C
Q	D
M	A
R	B
Q	C

R2(S)
X
A
B

W
P
Q

$R_1 \div R_2$ SCR

$R_1 \div R_2 =$ R − S

R = {W, X}
S = {X}

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following

8. List of course in which all students are registered .

- $\Pi_{code, ssn}(Registered) \,/\, \Pi_{ssn}(student)$

Consider the following relations :

Student (ssn, name, address, major)

Course (code, title)

Registered (ssn, code)

Use relational algebra to answer the following

9. List of course in which all ECMP major students are registered

$$\Pi_{ssn, code}(Registered) \div \Pi_{ssn}\left(\sigma_{major = 'ECMP'}(Student)\right)$$

Database Management System

UNIT 2

Relational data model and language

Lecture-6

Today's Target

- Modification on database and division numerical

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Modification of database(tuple by tuple)

1.Deletion

R<-R-E

E{ Relation algebra expression whose output is union compatible with R}

2. Insertion

R<-RUE

3. Updation

Assignment Operator

$\leftarrow$

$A \leftarrow A \cup B$

NOTE

Division Operator

Don't prefer $\rightarrow$ $\div$ $\rightarrow$ prefer

$R_1 \div R_2$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.1 Delete all the entries of borrow relation.

Borrow ← Borrow − Borrow

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.2 Delete all the entries from borrow relation corresponding to cardn0= c101.

Borrow ← Borrow − $\sigma_{\substack{\text{cardno} \\ = c101}}$ (Borrow)

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

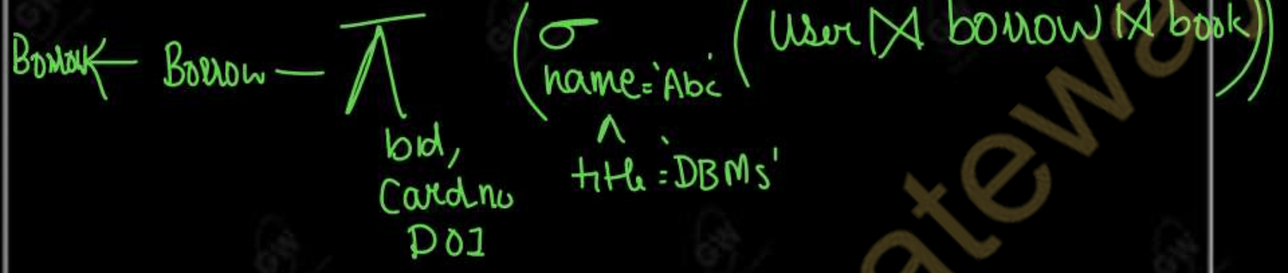
Q.3 Abc has return all the books.

$Borrow \leftarrow Borrow - \Pi_{bid, Cardno, DOI}\left(\sigma_{name='Abc'}\left(User \bowtie borrower\right)\right)$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

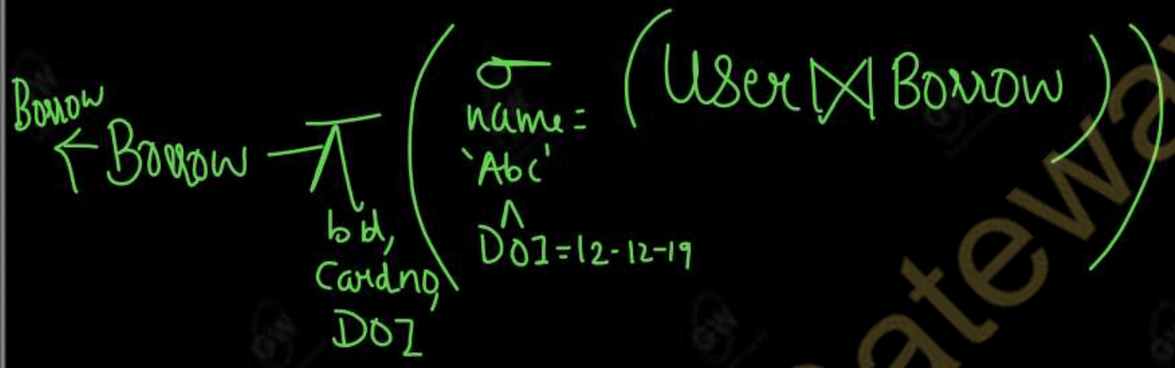
Q.4 Abc has return all DBMS book.



QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card no</u>,name,city)
3. Borrow(<u>bid,card no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.5 Abc has return all book issue on 12-12-19.



QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card no</u>,name,city)
3. Borrow(<u>bid,card no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.6 Delete all the entries of book having price more than 1000 from book table.

book ← book − $\pi_{bid, title, year\_pub}$ ($\sigma_{Price > 1000}$ (book ⋈ Supply))

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.7 Insert an entry into book having bid=101 year_pub =Y ,title=DBMS

book ← book ∪ {101, DBMS, Y}

QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card no</u>,name,city)
3. Borrow(<u>bid,card no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.8 Abc has issued a book on 21-2-18 having bid=b101

Borrow ∪ {b101, R, 21-2-18}

$$R \leftarrow \prod_{\text{Card\_no}} \left(\sigma_{\text{name='Abc'}} (\text{User}) \right)$$

QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card_no</u>,name,city)
3. Borrow(<u>bid,card_no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.9 Give 5% discount on all books.

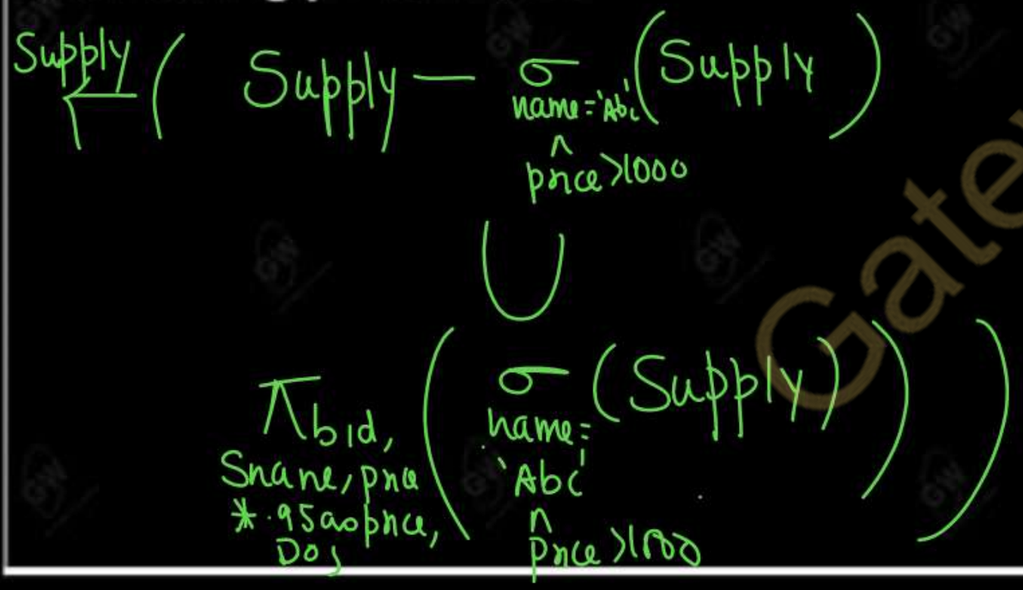
Supply ← ↑ (Supply)

bid, Sname, price * .95 as price, dos

QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card no</u>,name,city)
3. Borrow(<u>bid,card no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.10 Give 5% discount on all books supplied by Abc having price>1000.



QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.1 Find the name of that supplier who has supplied all the books issued to abc

$$\Pi_{bid,sname}(Supply) \div \Pi_{bid}\left(\sigma_{name='abc'}(User \bowtie Borrow)\right)$$

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.2 Find the names of those supplier who have supplied all the books of same title issue to abc

$$\Pi_{sname,title}(Supply \bowtie Book) \div \Pi_{title}(\sigma_{name='abc'}(User \bowtie Borrow \bowtie book))$$

AKTU QUESTION

Q.1	Consider the following schema for institute library: Student (RollNo, Name, Father_ Name, Branch) Book (ISBN, Title, Author, Publisher) Issue (RollNo, ISBN, Date-of –Issue) Write the following queries in SQL and relational algebra: (i) List roll number and name of all students of the branch 'CSE'. (ii) Find the name of student who has issued a book published by 'ABC' publisher. (iii) List title of all books and their authors issued to a student 'RAM'. (iv) List title of all books issued on or before December 1, 2020. (v) List all books published by publisher 'ABC'	AKTU 2023-24/ 2022-23

AKTU QUESTION		
Q.2	Consider the following Scheme: SUPPLIER (SUPPLIER ID, SUPPLIER_NAME, SUPPLIER_ADDRESS) PARTS (PART ID, PART_NAME, COLOR) CATALOG (SUPPLIER ID, PART ID, COST) Write the following queries in Relational Algebra (i) Find the name of the suppliers who supply Black Parts. (ii) Find the name of suppliers who supply both Blue and Black Parts. (iii) Find the name of suppliers who supply all Parts.	AKTU 2018-19

Database Management System

UNIT 2

Relational data model and language

Lecture-7

Today's Target

- Relational calculus
- PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Relational Calculus

- Relational calculus is a non-procedural query language.
- Relational Algebra has same power as Relational calculus.

They are of two types-

1. Tuple relational Calculus.
2. Domain relational Calculus.

1.TUPLE Relational calculus

- {t I COND(t)}

where t is a tuple variable and COND(t) is a conditional expression involving t. The result of such a query is the set of all tuples t that satisfy COND(t).

Many of the calculus expressions involves the use of Quantifiers. There are two types of quantifiers:

1. **Universal Quantifiers:** The universal quantifier denoted by ∀ is read as for all which means that in a given set of tuples exactly all tuples satisfy a given condition.
2. **Existential Quantifiers:** The existential quantifier denoted by ∃ is read as for all which means that in a given set of tuples there is at least one occurrences whose value satisfy a given condition.

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn

Dependent_name,sex,Bdate,Reationship)

Q.1 list all the employee details.

$\{t \mid Employee(t)\}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn Dependent_name,sex,Bdate,Reationship)

Q.2 find all employees whose salary is above 50,000.

$\{ t \mid Employee(t) \text{ AND } t.Salary > 50,000 \}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn Dependent_name,sex,Bdate,Reationship)

Q.3 find first name and last name employees whose salary is above 50,000.

$\{t.Fname, t.Lname \mid Employee(t) \text{ AND } t.Salary > 50,000\}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname, Dnumber, MGRSSN, MGRSTARTDATE)

Dept_location(Dnumber, Dlocation)

Project(Pname, Pnumber, Plocation, Dnum)

Works_on(EESN, Pno, Hours)

Dependent(Essn

Dependent_name, sex, Bdate, Reationship)

Q.4 Retrieve the birth date and address of the employee (or employees) whose name is 'John B. Smith.

$\{ t.Bdate, t.Address \mid Employee(t) \text{ AND } Fname = \text{'John'} \text{ AND } Minit = \text{'B'} \text{ AND } Lname = \text{'Smith'} \}$

Example Queries Using the Existential Quantifier

Q.5 Retrieve the name and address of all employees who work for the 'Research' department.

t

$\{ t.Fname, t.Minit, t.Lname, t.Address \mid Employee(t) \text{ AND } (\exists k)(Department(k) \text{ AND } k.Dname = \text{'Research'} \text{ AND } t.Dno = k.Dnumber) \}$

QUERIES

1. book(bid, title, year_pub)
2. User(card_no, name, city)
3. Borrow(bid, card_no, DOI)
4. Supply(bid, sname, price, dos)

Q.6 All the books Available in library.

$\{ t \mid book(t) \}$

Q.7 Give all the information about all Dbms books

$\{ t \mid book(t) \text{ AND } t.title = \text{'Dbms'} \}$

t →

bid	title	year_pub
1	DBMS	2016
2	A2	2017
3	DBMS	2018
4	AL	2015

1	DBMS	2016
3	DBMS	2018

QUERIES

1. book(bid, title ,year_pub)
2. User(card_no,name,city)
3. Borrow(bid,card_no, DOI)
4. Supply(bid,sname,price,dos)

Q.8 Give all the information about ~~all the~~ all user who lives in Delhi.

$\{ t \mid User(t) \text{ AND } t.city = \text{'Delhi'} \}$

Q title = 'DBMS' year_pub = 2018'

$\{ t \mid book(t) \text{ AND } t.title = \text{'DBMS'} \text{ AND } t.yearpub = 2018 \}$

QUERIES

1. book(<u>bid</u>, title ,year_pub)
2. User(<u>card no</u>,name,city)
3. Borrow(<u>bid,card no</u>, DOI)
4. Supply(<u>bid</u>,sname,price,dos)

Q.9 Find All book supplied by Abc

$\{t \mid book(t) \text{ AND } (\exists S)(Supply(S) \text{ AND } S.sname = \text{'Abc'} \text{ AND } S.bid = t.bid)\}$

Database Management System

UNIT 2

Relational data model and language

Lecture-8

Today's Target

- **Relational calculus**

AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn, Dependent_name,sex,Bdate,Reationship)

Q.1 Find the names of employees who have no dependents.

$\{ e.Fname, e.Lname \mid Employee(e) \text{ AND } (NOT\ (\exists d)(Dependent(d) \text{ AND } (e.SSN = d.ESSN))) \}$

↓ formula's

Universal

$\{ e.Fname \mid Employee(e) \text{ AND } ((\forall d)(NOT(Dependent(d)) \text{ OR } (NOT(e.SSN = d.ESSN)))) \}$

$((\forall x) P(x)) \rightarrow \text{NOT}((\forall x)(\text{NOT } P(x)))$

$((\exists x)(P(x)) \rightarrow \text{NOT}((\exists x)(\text{NOT } P(x)))$

$(\forall x)(P(x) \text{ and } Q(x)) \rightarrow \text{NOT}((\exists x)((\text{NOT } P(x)) \text{ OR } (\text{NOT}(Q(x)))))$

$(\forall x)(P(x) \text{ OR } Q(x)) \rightarrow \text{NOT}((\exists x)((\text{NOT } P(x)) \text{ AND } (\text{NOT}(Q(x)))))$

$(\exists x)(P(x) \text{ or } Q(x)) \rightarrow \text{NOT}((\forall x)((\text{NOT } P(x)) \text{ AND } (\text{NOT } Q(x))))$

$(\exists x)(P(x) \text{ and } Q(x)) \rightarrow \text{NOT}((\forall x)((\text{NOT } P(x)) \text{ OR } (\text{NOT } Q(x))))$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn

Dependent_name,sex,Bdate,Reationship)

Q.2 List the names of managers who have at least one dependent.

$\{ e.Fname, e.Lname \mid Employee(e) \text{ AND } ((\exists d)(\exists p)(department(d) \text{ And } (dependent(p) \text{ And } d.MGRSSN = e.SSN \text{ AND } p.ESSN = e.SSN)) \}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn Dependent_name,sex,Bdate,Reationship)

$\{ e.Fname, e.Lname \mid Employee(e) \wedge ((\exists x)(\exists w)(project(x) \wedge Work\_on(w) \wedge x.Dnum = 5 \wedge w.ESSN = e.SSN \wedge x.Pnumber = w.PNO))\}$

Q.3 Find the names of employees who work on all the projects controlled by department number 5.

$\{ e.Fname, e.Lname \mid Employee(e) \wedge (\forall x)(NOT\ Project(x) \vee (NOT(x.DUM = 5)) \text{ OR } ((\exists w))(Work\_on(w) \text{ AND } w.SSN = e.SSN \text{ AND } x.Pnumber = w.PNO)))\}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn Dependent_name,sex,Bdate,Reationship)

Q.4 Make a list of project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as manager of the controlling department for the project

$\{P.Pnumber \mid Project(P)$ And $((\exists e)(\exists w)$ $(Employee(e)$ AND $Works_on(w)$ AND $e.Lname='Smith'$ AND $e.SSN = w.ESSN$ AND $P.Pnumber = w.Essn))$

OR

$((\exists m)(\exists d)(department(d)$ AND $Employee(m)$ AND $m.Lname='Smith'$ AND $m.Ssn = d.MGRSSN$ AND $d.Dnumber = P.Dnum)))$

Safe Expressions

- Whenever we use universal quantifiers, existential quantifiers, or negation of predicates in a calculus expression, we must make sure that the resulting expression makes sense.
- A safe expression in relational calculus is one that is guaranteed to yield a finite number of tuples as its result; otherwise, the expression is called unsafe.

For example, the expression {t I NOT (EMPLOYEE(t))

- It is unsafe because it yields all tuples in the universe that are not EMPLOYEE tuples, which are infinitely numerous.

{t | NOT Employee(t) }

Database Management System

UNIT 2

Relational data model and language

Lecture-9

Today's Target

- Domain Calculus.
- AKTU PYQs

By PRAGYA RAJVANSHI

B.Tech, M.Tech(C.S.E.)

Domain relational calculus

- Domain calculus differs from tuple calculus in the type of variables used in formulas.
- Rather than having variables range over tuples, the variables range over single values from domains of attributes.
- An expression of the domain calculus is of the form
- {a1,a2,a3|P}

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn, Dependent_name,sex,Bdate,Reationship)

Q.1 Retrieve the birthdate and address of the employee whose name is 'John B. Smith.

$\{ u,v \mid (\exists q)(\exists r)(\exists s)(\text{Employee}(qrstuvwxyz) \text{ AND } q=\text{`John'} \text{ AND } r=\text{'B'} \text{ AND } s=\text{`Smith'}) \}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname, Dnumber, MGRSSN, MGRSTARTDATE)

Dept_location(Dnumber, Dlocation)

Project(Pname, Pnumber, Plocation, Dnum)

Works_on(EESN, Pno, Hours)

Dependent(Essn, Dependent_name, sex, Bdate, Reationship)

Q.2 Retrieve the name and address of all employees who work for the 'Research' department.

$\{ qrsv \mid (\exists m)(\exists z)(\exists l)(\text{Employee}(qrstuvwxyz) \text{ AND Department}(lmno) \text{ AND } l = \text{'Research'} \text{ AND } m = z) \}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn, Dependent_name,sex,Bdate,Reationship)

Q.3 For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, birth date, and address.

$\{ i k s u \mid (\exists l)(\exists m)(\exists n)(\exists t)$ (Employee(qrstuvwxyz) AND Department(lmno) AND Project(hijk) AND $j = \text{'stafford'}$ AND $k = m$ AND $n = t$)

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn, Dependent_name,sex,Bdate,Reationship)

Q.4 Find the names of employees who have no dependents.

$\{ q\,r\,s \mid (\exists t)(\text{Employee}(q\,r\,s\,t\,u\,v\,w\,x\,y\,z) \text{ AND } (\text{NOT}(\exists a)(\text{Dependent}(a\,b\,c\,d\,e) \text{ AND } a = t))) \}$

having dependent

$\{ q\,r\,s \mid (\exists t)(\exists a)(\text{Employee}(q\,r\,s\,t\,u\,v\,x\,y\,z) \text{ AND Dependent}(a\,b\,c\,d\,e) \text{ AND } a = t) \}$

Employee(Fname, Minit, Lname, SSN, Bdate, Address, Sex, Salary, SuperSsn, Dno)

Department(Dname,Dnumber,MGRSSN,MGRSTARTDATE)

Dept_location(Dnumber,Dlocation)

Project(Pname,Pnumber,Plocation,Dnum)

Works_on(EESN,Pno,Hours)

Dependent(Essn, Dependent_name,sex,Bdate,Reationship)

Q.5 List the names of managers who have at least one dependent.

$$\{ q\,r\,s \mid (\exists t)(\exists n)(\exists a)(\text{Employee}(q\,r\,s\,t\,u\,v\,w\,x\,y) \text{ AND Department}(l\,m\,n\,o) \text{ AND Dependent}(a\,b\,c\,d\,e) \text{ AND } t = n \text{ AND } t = a) \}$$

$$\{ n \mid (\exists L)(\text{Department}(l\,m\,n\,o) \text{ AND } L = \text{'AZMI'}) \}$$