Query Language

- After designing a data base, that is ER diagram followed by conversion in relational model followed by normalization and indexing, now next task is how to store, retrieve and modify data in the data base. Thought here we will be concentrating more on the retrieval part. Query languages are used for this purpose.

- **QUERY LANGUAGE**
    - A Languages using which user request some information from the database.
- **Procedural Query Language**
    - Here users instruct the system to performs a sequence of operations on the data base in order to compute the desired result.
    - Means user provides both what data to be retrieved and how data to be retrieved. e.g. Relational Algebra.
- **Non-Procedural Query Language**
    - In nonprocedural language, the user describes the desired information without giving a specific procedure for obtaining that information. What data to be retrieved e.g. Relational Calculus. **Tuple relational calculus, Domain relational calculus are** declarative query languages based on mathematical logic
- Relational Algebra (Procedural) and Relational Calculus (non-procedural) are mathematical system/ query languages which are used for query on relational model. RA and RC are not executed in any computer, they provide the fundamental mathematics on which SQL is based.
- SQL (structured query language) works on RDBMS, and it includes elements of both procedural or non-procedural query language.

Relational model	RDBMS
RA, RC	SQL
Algo	Code
Conceptual	Reality
Theoretical	Practical
Chess	Battle Field

RELATIONAL ALGEBRA

- RA like any other mathematical system provides a number of operators and use relations (tables) as operands and produce a new relation as their result.
- Every operator in the RA accepts (one or two) relation/table as input arguments and returns always a single relation instance as the result without a name.
- It also does not consider duplicity by default as it is based on set theory. Same query is written in RA and SQL the result may be different as SQL considers duplication.
- As it is pure mathematics no use of English keywords. Operators are represented using symbols.
- The relational algebra is a ***procedural query language***.
- The fundamental operations in the relational algebra are **select, project, union, set difference, Cartesian product, and Rename.**
- There are several other operations namely: **set intersection, natural join, and assignment.**
- The **select, project, and rename** operations are called **unary operations**, because they operate on one relation.
- **Union, Cartesian product and set difference** operate on pairs of relations and are, therefore, called **binary operations.**
- Relational algebra also provides the framework for query optimization.

- **<u>Relational schema</u>** - A **relation schema** R, denoted by R ($A_1, A_2, ..., A_n$), is made up of a relation name R and a list of attributes, $A_1, A_2, ..., A_n$. Each **attribute** Ai is the name of a role played by some domain D in the relation schema R. It is use to describe a Relation.
- E.g. Schema representation of Table **Student** is as –
  - **STUDENT (NAME, ID, CITY, COUNTRY, HOBBY).**
- **<u>Relational Instance</u>** - Relations with its data at particular instant of time.

Q If D_1, D_2, …... D_n are domains in a relational model, then the relation is a table, which is a subset of **(NET-JUNE-2013)**

(A) $D_1 + D_2 + ... + D_n$ **(B)** $D_1 \times D_2 \times ... \times D_n$

(C) $D_1 \cup D_2 \cup ... \cup D_n$ **(D)** $D_1 - D_2 - ... - D_n$

Ans: b

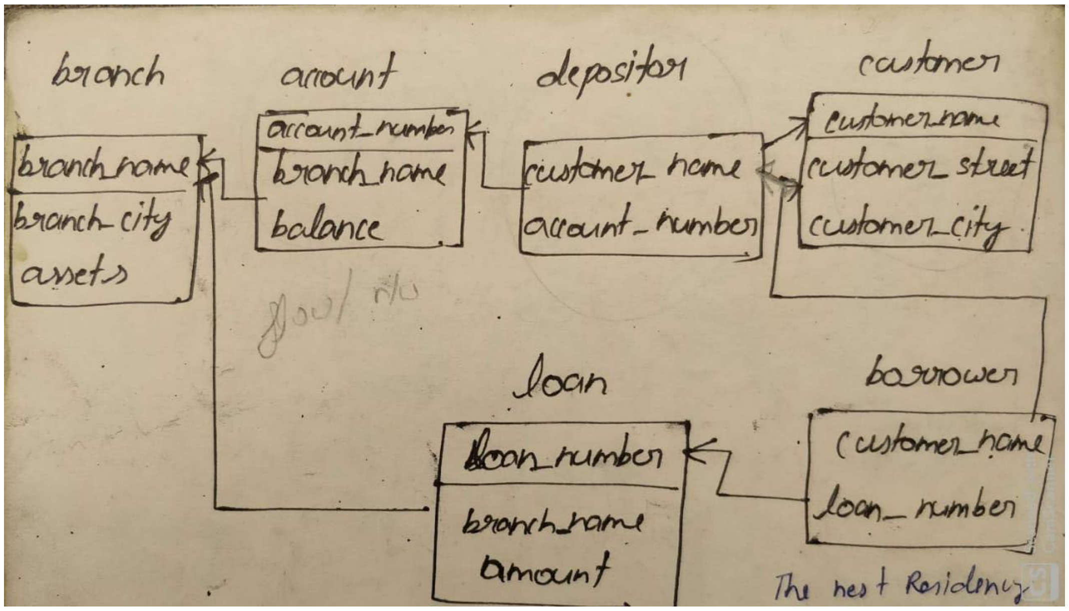
OPERATORS USED IN RELATIONAL ALGEBRA

- **BASIC / FUNDAMENTAL OPERATORS**

Name	Symbol
Select	(σ)
Project	(∏)
Union	(∪)
Set difference	(−)
Cross product	(X)
Rename	(ρ)

- **DERIVED OPERATORS**

Name	Symbol	DERIVED FROM
Join	(⋈)	(X)
Intersection	(∩)	(−) A∩B=A-(A-B)
Division	(÷)	(X,-, ∏)
Assignment	(=)	



The Project Operation (Vertical Selection)

- Main idea behind project operator is to select desired columns.
- The project operation is a unary operation that returns its argument relation, with certain attributes left out.
- Projection is denoted by the uppercase Greek letter pi (Π).
- **$\Pi_{\text{column\_name}}$ (table_name)**
- We list those attributes that we wish to appear in the result as a subscript to Π, argument relation follows in parentheses.
- Minimum number of columns selected can be 1, Maximum selected Columns can be n - 1.

- **Some points to remember**
  - Eliminates duplicate rows in a result relation by default.
  - $\prod_{A1, A2, An}(r)$, $A_1, A_2, \ldots, A_n$ refers to the set of attributes to be projected.
  - It is not commutative.

Q Write a RELATIONAL ALGEBRA query to find the name of all customer having bank account?

Q Write a RELATIONAL ALGEBRA query to find each loan number along with loan amount?

Q Write a RELATIONAL ALGEBRA query to find the name of all customer without duplication having bank account?

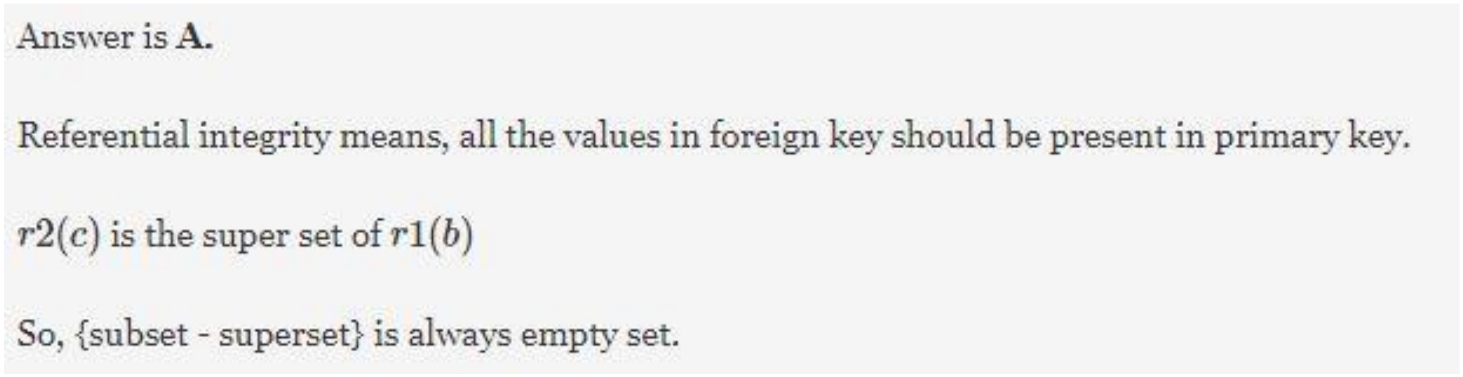
Q Write a RELATIONAL ALGEBRA query to find all the details of bank branches?

Q Suppose $R_1(A, B)$ and $R_2(C, D)$ are two relation schemas. Let r_1 and r_2 be the corresponding relation instances. B is a foreign key that refers to C in R_2. If data in r_1 and r_2 satisfy referential integrity constraints, which of the following is ALWAYS TRUE? **(Gate-2012) (2 Marks)**

a) $\prod B(r_1) - \prod C(r_2) = \emptyset$ **b)** $\prod C(r_2) - \prod B(r_1) = \emptyset$

c) $\prod B(r_1) = \prod C(r_2)$ **d)** $\prod B(r_1) - \prod C(r_2) \neq \emptyset$

Ans: a



Answer is **A.**

Referential integrity means, all the values in foreign key should be present in primary key.

$r2(c)$ is the super set of $r1(b)$

So, {subset - superset} is always empty set.

The Select Operation (Horizontal Selection)

- The select operation selects tuples that satisfy a given predicate/Condition p.
- Lowercase Greek letter sigma (σ) is used to denote selection.
- It is a unary operator.
- Eliminates only tuples/rows.
- $\sigma_{\text{condition}}$ (table_name)
- Predicate appears as a subscript to σ, the argument relation is in parentheses after the σ.
- Commutative in Nature, $\sigma_{p1}(\sigma_{p2}(r)) = \sigma_{p2}(\sigma_{p1}(r))$

Some points to remember

- We allow comparisons using =, ≠, <, >, ≤ and ≥ in the selection predicate.
- Using the connectives and (∧), or (∨), and not (¬), we can combine several predicates into a larger predicate.
- Minimum number of tuples selected can be 0, Maximum selected tuples can be all.
- Degree (Result relation) = degree (parent relation), where degree refers to no. of attributes.
- 0 <= cardinality (result relation) <= cardinality (parent relation), where cardinality refers to no. of tuples.

Q Write a RELATIONAL ALGEBRA query to find all account_no where balance is less the 1000?

Q Write a RELATIONAL ALGEBRA query to find branch name which is situated in Delhi and having assets less than 1,00,000?

Q Write a RELATIONAL ALGEBRA query to find branch name and account_no which has balance greater than equal to 1,000 but less than equal to 10,000?

Q Consider the following schemas: **(NET-DEC-2013)**
Branch = (Branch-name, Assets, Branch-city)
Customer = (Customer-name, Bank name, Customer-city)
Borrow = (Branch-name, loan number, customer account-number)
Deposit = (Branch-name, Account- number, Customer-name, Balance)
Using relational Algebra, the Query that finds customers who have balance more than 10,000 is ________

(A) $\pi_{\text{customer-name}}(\sigma_{\text{balance} > 10000}(\text{Deposit})$ **(B)** $\sigma_{\text{customer-name}}(\sigma_{\text{balance} > 10000}(\text{Deposit})$
(C) $\pi_{\text{customer-name}}(\sigma_{\text{balance} >10000}(\text{Borrow})$ **(D)** $\sigma_{\text{customer-name}}(\pi_{\text{balance} >10000}(\text{Borrow})$
Ans: a

Q Which of the following query transformations (i.e., replacing the l.h.s. expression by the r.h.s. expression) is incorrect? R_1 and R_2 are relations. C_1, C_2 are selection conditions and A_1, A_2 are attributes of R_1. **(Gate-1998) (2 Marks)**

a) $\sigma_{C1}(\sigma_{C2} R_1)) \rightarrow \sigma_{C2}(\sigma_{C1}(R_1))$

b) $\sigma_{C1}(\pi_{A1} R_1)) \rightarrow \pi_{A1}(\sigma_{C1}(R_1))$

c) $\sigma_{C1}(R_1 \cup R_2) \rightarrow \sigma_{C1}(R_1) \cup \sigma_{C1}(R_2)$

d) $\pi_{A1}(\sigma_{C1}(R_1)) \rightarrow \sigma_{C1}(\pi_{A1}(R_1))$

Answer: (D)

D) if the selection condition is on attribute A2, then we cannot replace it by RHS as there will not be any attribute A2 due to projection of A1 only.

Q What is the optimized version of the relation algebra expression $\pi_{A1}(\pi_{A2}(\sigma_{F1}(\sigma_{F2}(r))))$, where A_1, A_2 are sets of attributes in r with $A_1 \subset A_2$ and F_1, F_2 are Boolean expressions based on the attributes in r? **(Gate-2014) (2 Marks)**

a) $\pi_{A1}(\sigma_{(F1 \wedge F2)}(r))$ **b)** $\pi_{A1}(\sigma_{(F1 \vee F2)}(r))$ **c)** $\pi_{A2}(\sigma_{(F1 \wedge F2)}(r))$ **d)** $\pi_{A2}(\sigma_{(F1 \vee F2)}(r))$

Ans: a

The Union Operation

- It is a binary operation, denoted, as in set theory, by ∪.
- Written as, $\text{Expression}_1 \cup \text{Expression}_2$, $r \cup s = \{t \mid t \in r \text{ or } t \in s\}$
- For a union operation $r \cup s$ to be valid, we require that two conditions hold:
- The relations r and s must be of the same arity. That is, they must have the same number of attributes, the domains of the ith attribute of r and the ith attribute of s must be the same, for all i.
- Mainly used to fetch data from different relations.

- **Some points to remember**
    - $\text{Deg}(R \cup S) = \text{Deg}(R) = \text{Deg}(S)$
    - $\text{Max}(|R|, |S|) \le |R \cup S| \le (|R| + |S|)$

Q Write a RELATIONAL ALGEBRA query to find all the customer name who have a loan or an account or both?

The Set-Difference Operation

- The set-difference operation, denoted by −, allows us to find tuples that are in one relation but are not in another. It is a binary operator.
- The expression r − s produces a relation containing those tuples in r but not in s.
- For a set-difference operation r − s to be valid, we require that the relations r and s be of the same arity, and that the domains of the ith attribute of r and the ith attribute of s be the same, for all i.
- 0 <= IR - SI <= IRI

Q Write a RELATIONAL ALGEBRA query to find all the customer name who have a loan but do not have an account?

The Cartesian-Product Operation

- The Cartesian-product operation, denoted by a cross (×), allows us to combine information from any two relations.
- It is a binary operator; we write the Cartesian product of relations R_1 and R_2 as $R_1 \times R_2$.
- Cartesian-product operation associates every tuple of R_1 with every tuple of R_2.
    - $R_1 X R_2 = \{rs \mid r \in R_1 \text{ and } s \in R_2\}$, contains one tuple <r, s> (concatenation of tuples r and s) for each pair of tuples $r \in R_1$, $s \in R_2$.
- R_1 X R_2 returns a relational instance whose schema contains all the fields of R_1 (in order as they appear in R_1) and all fields of R_2 (in order as they appear in R_2).
- If R_1 has m tuples and R_2 has n tuples the result will be having = m*n tuples.
- Same attribute name may appear in both R_1 and R_2, we need to devise a naming schema to distinguish between these attributes.

R_1	
A	B
1	P
2	Q
3	R

R_2	
B	C
Q	X
R	Y
S	Z

$R_1 * R_2$			
A	$R_1.B$	$R_2.B$	C
1	P	Q	X
1	P	R	Y
1	P	S	Z
2	Q	Q	X
2	Q	R	Y
2	Q	S	Z
3	R	Q	X
3	R	R	Y
3	R	S	Z

Q Write a RELATIONAL ALGEBRA query to find the name of all the customers along with account balance, who have an account in the bank?

- To solve this query we understand that customer who have an account are available in depositor and balance is available in account, so to answer this query each tuple of the table account must be matched with each tuple with depositor, and they have a common attribute account_no if there is a match then that tuple is valid otherwise redundant, must be eliminated with the conditions.

Q Write a RELATIONAL ALGEBRA query to find the name of all the customers along with account balance, who have an account in the bank?

Q Write a RELATIONAL ALGEBRA query to find the name of all the customers along with loan amount, who have a loan in the bank?

Q Write a RELATIONAL ALGEBRA query to find all loan_no along with amount and branch_name, which is situated in Delhi?

Q Write a RELATIONAL ALGEBRA query to find the name of the customer who have an account in the branch situated in Delhi and balance greater than 1000?

The Rename Operation

- The results of relational algebra are also relations but without any name.
- The rename operation allows us to rename the output relation. It is denoted with small Greek letter **rho** ρ. Where the result of expression **E** is saved with name of **x.**
- $\rho_{x(A1, A2, A3, A4,\ldots..AN)}(E)$

Q Write a RELATIONAL ALGEBRA query to find the account_no along with balance with 8% interest as total amount, with table name as balance sheet?

- even if an attribute name can be derived from the base relations, we may want to change the attribute name in the result.
- One reason to rename a relation is to replace a long relation name with a shortened version that is more convenient to use elsewhere in the query.

Q Write a RELATIONAL ALGEBRA query to find the loan_no with maximum loan amount?

- Another reason to rename a relation is a case where we wish to compare tuples in the same relation. We then need to take the Cartesian product of a relation with itself and, without renaming, it becomes impossible to distinguish one tuple from the other.
- A and b are used to rename a relation is referred to as table alias, correlation variable or tuple variable.

Additional Relational-Algebra Operations

- If we restrict ourselves to just the fundamental operations, certain common queries are lengthy to express. Therefore, we use additional operations.
  - These additional operations do not add any power to the algebra.
  - They are used to simplify the queries.

The Set-Intersection Operation

- We will be using ∩ symbol to denote set intersection.
- $r \cap s = r - (r - s)$
- Set intersection is not a fundamental operation and does not add any power to the relational algebra.
- $r \cap s = \{t \mid t \in r \text{ and } t \in s\}$
- 0 <= I R∩S I <= min (IRI, ISI)

Q Write a RELATIONAL ALGEBRA query to find all the customer name who have both a loan and an account?

The Natural-Join Operation

- The natural join is a binary operation that allows us to combine certain selections and a Cartesian product into one operation.
- The natural join of r and s, denoted by r ⋈ s
- The natural-join operation forms a Cartesian product of its two arguments, performs a selection forcing equality on those attributes that appear in both relation schemas and finally removes duplicate attributes.
- The natural join of r and s is a relation on schema R ∪ S formally defined as follows:

$$r \bowtie s = \Pi_{R \cup S}(\sigma_{r.A_1 = s.A_1 \wedge r.A_2 = s.A_2 \wedge \ldots \wedge r.A_n = s.A_n}(r \times s))$$

Some Points to Remember

- The natural join is **associative** in nature.
- That is if we have a natural join such as:
- $(instructor \bowtie teaches) \bowtie course = instructor \bowtie (teaches \bowtie course)$

R₁	
A	B
1	P
2	Q
3	R

R₂	
B	C
Q	X
R	Y
S	Z

R₁ ⋈ R₂		
A	B	C
2	Q	X
3	R	Y

Q Write a RELATIONAL ALGEBRA query to find the name of all the customers along with account balance, who have an account in the bank?

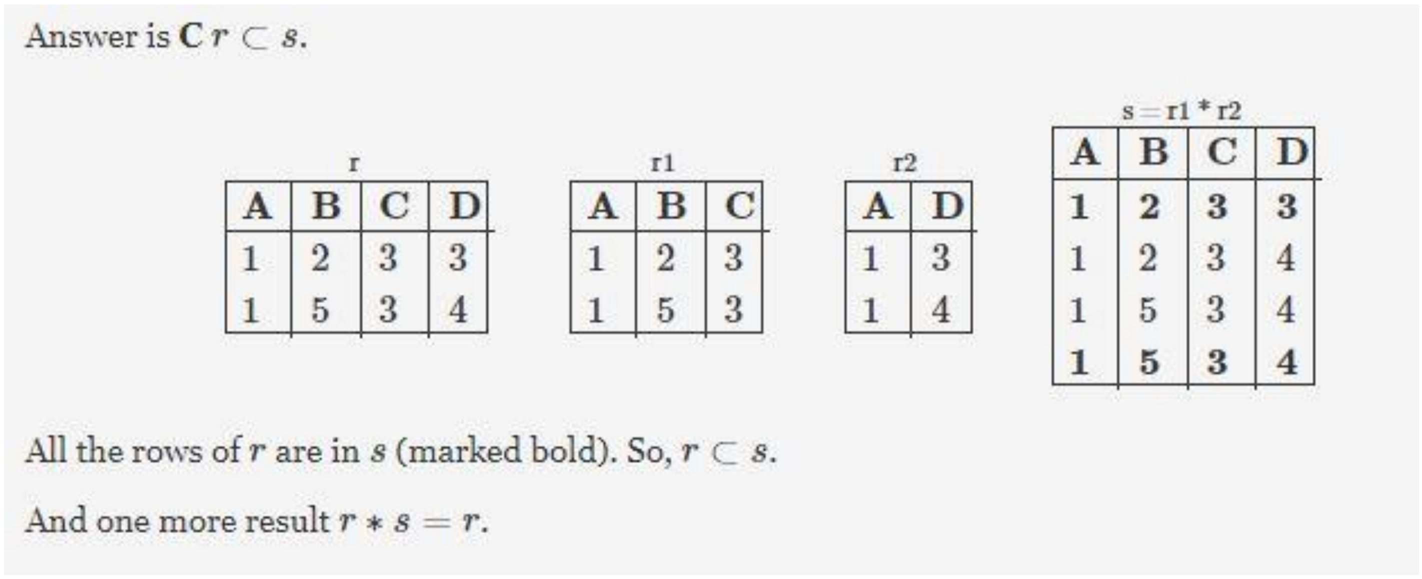
Q Write a RELATIONAL ALGEBRA query to find the name of all the customers along with loan amount, who have a loan in the bank?

Q Write a RELATIONAL ALGEBRA query to find all loan_no along with amount and branch_name, which is situated in Delhi?

Q Write a RELATIONAL ALGEBRA query to find the name of the customer who have an account in the branch situated in Delhi and balance greater than 1000?

Q Let r be a relation instance with schema R = (A, B, C, D). We define $r_1 = \Pi_{A, B, C}(r)$ and $r_2 = \Pi_{A.D}(r)$. Let $s = r_1 * r_2$ where * denotes natural join. Given that the decomposition of r into r_1 and r_2 is lossy, which one of the following is TRUE? **(Gate-2005) (1 Marks)**

(A) $s \subset r$ **(B)** $r \cup s$ **(C)** $r \subset s$ **(D)** $r * s = s$


Answer is **C** $r \subset s$.

r

A	B	C	D
1	2	3	3
1	5	3	4

r1

A	B	C
1	2	3
1	5	3

r2

A	D
1	3
1	4

s = r1 * r2

A	B	C	D
1	**2**	**3**	**3**
1	2	3	4
1	5	3	4
1	**5**	**3**	**4**

All the rows of r are in s (marked bold). So, $r \subset s$.

And one more result $r * s = r$.

Q Let r and s be two relations over the relation schemes R and S respectively, and let A be an attribute in R. The relational algebra expression $\sigma_{A=a}(r \bowtie s)$ is always equal to **(Gate-2001) (1 Marks)**

a) $\sigma_{A=a}(r)$ **b)** r **c)** $\sigma_{A=a}(r) \bowtie s$ **d)** None of the above

Answer is C.

C is just the better form of query, more execution friendly because requires less memory while joining. query, given in question takes more time and memory while joining.

Q Consider the relations R(A, B) and S(B, C) and the following four relational algebra queries over R and S :

I. $\Pi_{A, B}(R \bowtie S)$ **II.** $R \bowtie \Pi_B(S)$

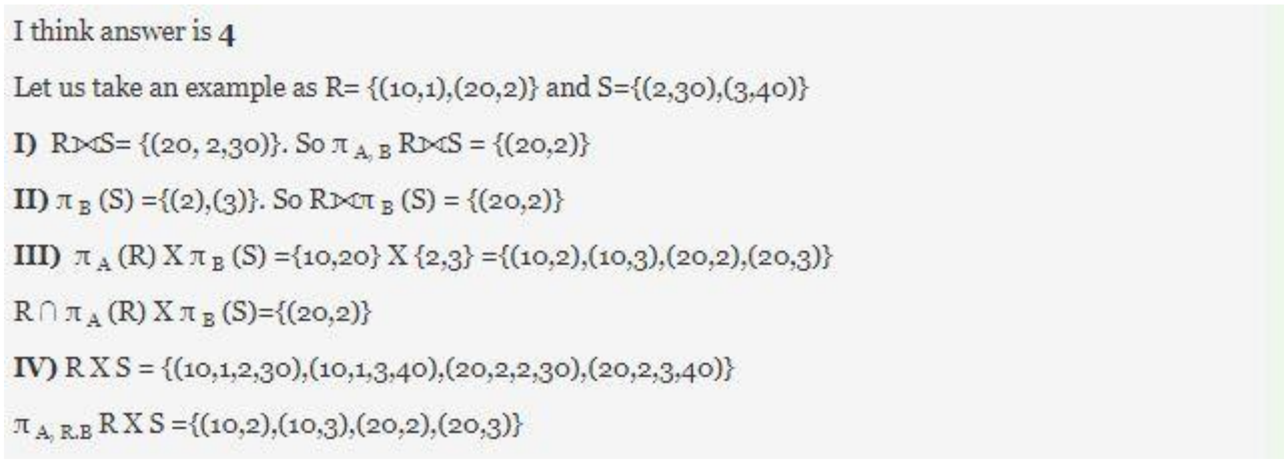
III. $R \cap (\Pi_A(R) \times \Pi_B(S))$ **IV.** $\Pi_{A, R.B}(R \times S)$

where R·B refers to the column B in table R. One can determine that: **(NET-JULY-2016)**

(1) I, III and IV are the same query. **(2)** II, III and IV are the same query.

(3) I, II and IV are the same query. **(4)** I, II and III are the same query

Ans. 4



I think answer is **4**

Let us take an example as R= {(10,1),(20,2)} and S={(2,30),(3,40)}

I) R$\bowtie$S= {(20, 2,30)}. So $\pi_{A, B}$ R$\bowtie$S = {(20,2)}

II) π_B (S) ={(2),(3)}. So R$\bowtie\pi_B$ (S) = {(20,2)}

III) π_A (R) X π_B (S) ={10,20} X {2,3} ={(10,2),(10,3),(20,2),(20,3)}

R ∩ π_A (R) X π_B (S)={(20,2)}

IV) R X S = {(10,1,2,30),(10,1,3,40),(20,2,2,30),(20,2,3,40)}

$\pi_{A, R.B}$ R X S ={(10,2),(10,3),(20,2),(20,3)}

Q Let R and S be two relations with the following schema **(Gate-2008) (2 Marks)**

R (<u>P, Q</u>, R1, R2, R3) S (<u>P, Q</u>, S1, S2)

where {P, Q}is the key for both schemas. Which of the following queries are equivalent?

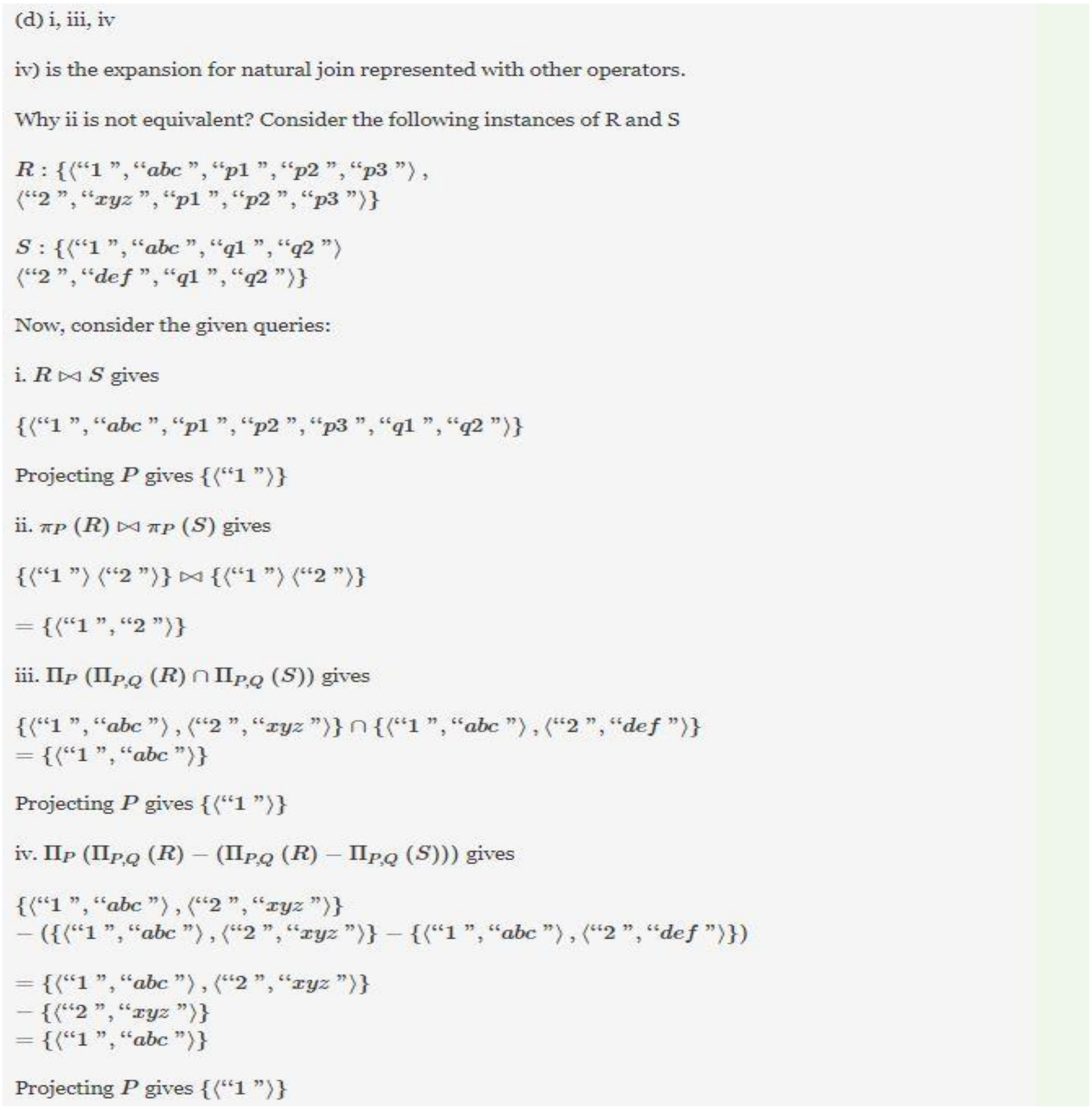
i) $\Pi_P (R \bowtie S)$ **ii)** $\Pi_P (R) \bowtie \Pi_P (S)$

iii) $\Pi_P (\Pi_{P, Q} (R) \cap \Pi_{P, Q} (S))$ **iv)** $\Pi_P (\Pi_{P, Q} (R) - (\Pi_{P, Q} (R) - \Pi_{P, Q} (S)))$

a) Only I and II **b)** Only I and III

c) Only I, II and III **d)** Only I, III and IV

Ans: d



(d) i, iii, iv

iv) is the expansion for natural join represented with other operators.

Why ii is not equivalent? Consider the following instances of R and S

$R : \{\langle \text{"1"}, \text{"}abc\text{"}, \text{"}p1\text{"}, \text{"}p2\text{"}, \text{"}p3\text{"}\rangle,$
$\langle \text{"2"}, \text{"}xyz\text{"}, \text{"}p1\text{"}, \text{"}p2\text{"}, \text{"}p3\text{"}\rangle\}$

$S : \{\langle \text{"1"}, \text{"}abc\text{"}, \text{"}q1\text{"}, \text{"}q2\text{"}\rangle$
$\langle \text{"2"}, \text{"}def\text{"}, \text{"}q1\text{"}, \text{"}q2\text{"}\rangle\}$

Now, consider the given queries:

i. $R \bowtie S$ gives

$\{\langle \text{"1"}, \text{"}abc\text{"}, \text{"}p1\text{"}, \text{"}p2\text{"}, \text{"}p3\text{"}, \text{"}q1\text{"}, \text{"}q2\text{"}\rangle\}$

Projecting P gives $\{\langle \text{"1"}\rangle\}$

ii. $\pi_P(R) \bowtie \pi_P(S)$ gives

$\{\langle \text{"1"}\rangle \langle \text{"2"}\rangle\} \bowtie \{\langle \text{"1"}\rangle \langle \text{"2"}\rangle\}$

$= \{\langle \text{"1"}, \text{"2"}\rangle\}$

iii. $\Pi_P(\Pi_{P,Q}(R) \cap \Pi_{P,Q}(S))$ gives

$\{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}xyz\text{"}\rangle\} \cap \{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}def\text{"}\rangle\}$
$= \{\langle \text{"1"}, \text{"}abc\text{"}\rangle\}$

Projecting P gives $\{\langle \text{"1"}\rangle\}$

iv. $\Pi_P(\Pi_{P,Q}(R) - (\Pi_{P,Q}(R) - \Pi_{P,Q}(S)))$ gives

$\{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}xyz\text{"}\rangle\}$
$- (\{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}xyz\text{"}\rangle\} - \{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}def\text{"}\rangle\})$

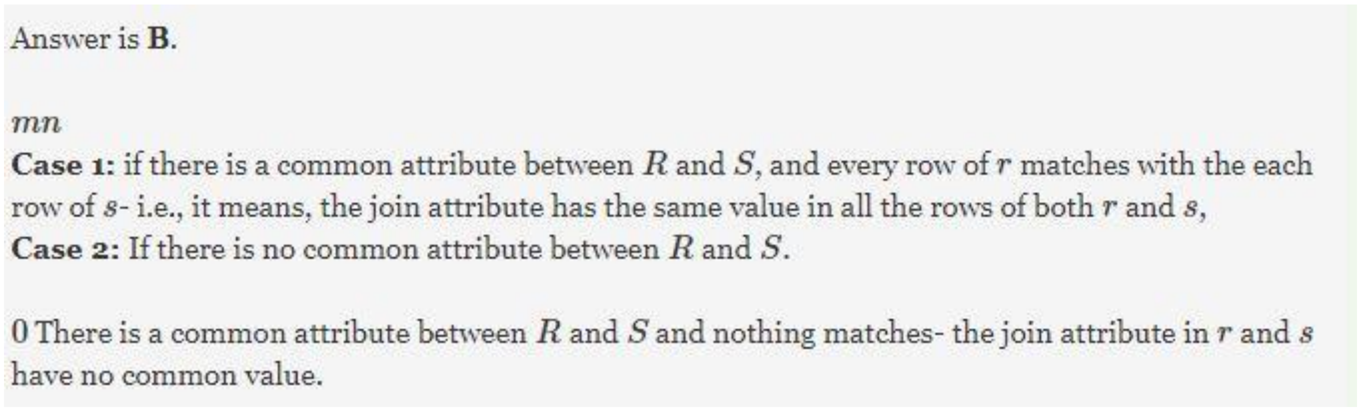
$= \{\langle \text{"1"}, \text{"}abc\text{"}\rangle, \langle \text{"2"}, \text{"}xyz\text{"}\rangle\}$
$- \{\langle \text{"2"}, \text{"}xyz\text{"}\rangle\}$
$= \{\langle \text{"1"}, \text{"}abc\text{"}\rangle\}$

Projecting P gives $\{\langle \text{"1"}\rangle\}$

Q Consider the join of a relation R with a relation S. If K has *m tuples* and S has *n tuples*, then the maximum and minimum sizes of the join respectively are: **(Gate-1999) (1 Marks)**

(A) m+n and 0 **(B)** mn and 0 **(C)** m+n and m-n **(D)** mn and m+n

Answer: (B)


Answer is **B**.

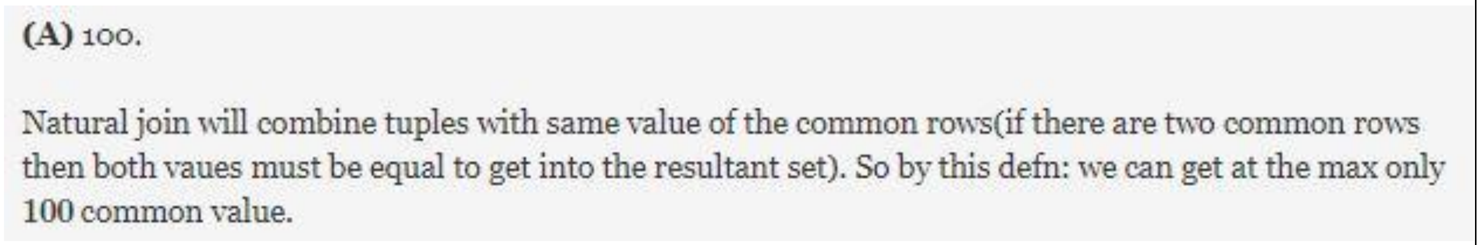
mn
Case 1: if there is a common attribute between R and S, and every row of r matches with the each row of s- i.e., it means, the join attribute has the same value in all the rows of both r and s,
Case 2: If there is no common attribute between R and S.

0 There is a common attribute between R and S and nothing matches- the join attribute in r and s have no common value.

Q The following functional dependencies hold for relations R(A, B, C) and S(B, D, E).
B → A
A → C
The relation R contains 200 tuples and the relation S contains 100 tuples. What is the maximum number of tuples possible in the natural join R ⋈ S? (Gate-2010) (2 Marks)

a) 100 **b)** 200 **c)** 300 **d)** 2000

Ans: a


(A) 100.

Natural join will combine tuples with same value of the common rows(if there are two common rows then both vaues must be equal to get into the resultant set). So by this defn: we can get at the max only 100 common value.

Q (NET-DEC-2015)

Consider the following three tables R, S and T. In this question, all the join operations are natural joins ($\bowtie$). (π) is the projection operation of a relation :

R

A	B
1	2
3	2
5	6
7	8
9	8

S

B	C
6	2
2	4
8	1
8	3
2	5

T

A	C
7	1
1	2
9	3
5	4
3	5

Possible answer tables for this question are also given as below :

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

(a)

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

(b)

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

(c)

A	B	C
3	2	5
7	8	1
9	8	3

(d)

What is the resulting table of $\pi_{A,B}(R \bowtie T) \bowtie \pi_{B,C}(S \bowtie T)$?

(1) (a) (2) (b) (3) (c) (4) (d)

Theta join / Conditional Join

- The theta join/ Conditional join operation is a variant of the natural-join operation that allows us to combine a selection and a Cartesian product into a single operation.
- The theta join operation $r \bowtie_\theta s$ is defined as follows: $r \bowtie_\theta s = \sigma_\theta(r \times s)$

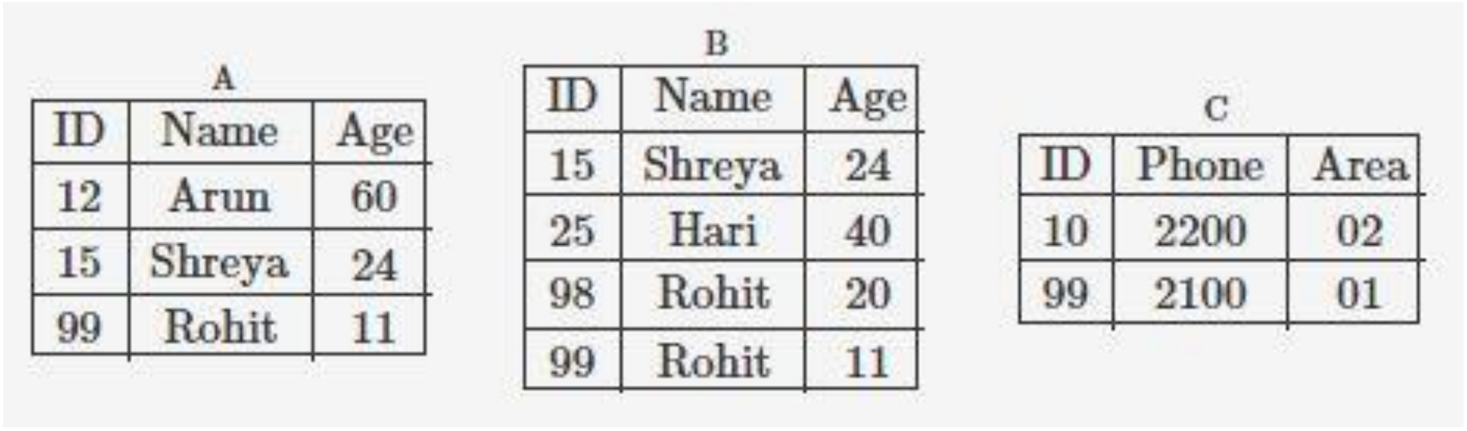
Q Let R1 (<u>A</u>, B, C) and R2 (<u>D</u>, E) be two relation schema, where the primary keys are shown underlined, and let C be a foreign key in R1 referring to R2. Suppose there is no violation of the above referential integrity constraint in the corresponding relation instances r1 and r2. Which one of the following relational algebra expressions would necessarily produce an empty relation? **(Gate-2004) (1 Marks)**

a) $\Pi_D(r_2)-\Pi_C(r_1)$ **b)** $\Pi_C(r_1)-\Pi_D(r_2)$ **c)** $\Pi_D(r_1 \bowtie_{C \neq D} r_2)$ **d)** $\Pi_C(r_1 \bowtie_{C=D} r_2)$

Answer: (B)

C in R1 is a foreign key referring to the primary key D in R2. So, every element of C must come from some D element.

Q Consider the following relations A, B and C:



A

ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11

B

ID	Name	Age
15	Shreya	24
25	Hari	40
98	Rohit	20
99	Rohit	11

C

ID	Phone	Area
10	2200	02
99	2100	01

How many tuples does the result of the following relational algebra expression contain? Assume that the schema of AUB is the same as that of A. **(Gate-2007) (2 Marks)**

$(A \cup B) \bowtie_{A.Id > 40 \vee C.Id < 15} C$

a) 7 **b)** 4 **c)** 5 **d)** 9

Q Suppose database table $T_1(P, R)$ currently has tuples {(10, 5), (15, 8), (25, 6)} and table $T_2(A, C)$ currently has {(10, 6), (25, 3), (10, 5)}. Consider the following three relational algebra queries RA_1, RA_2 and RA_3: **(NET-AUG-2016)**

RA_1: $T_1 \bowtie_{T1.P = T2.A} T_2$ where $\bowtie$ is natural join symbol

RA_2: $T_1 =\bowtie_{T1.P = T2.A} T_2$ where $=\bowtie$ is left outer join symbol

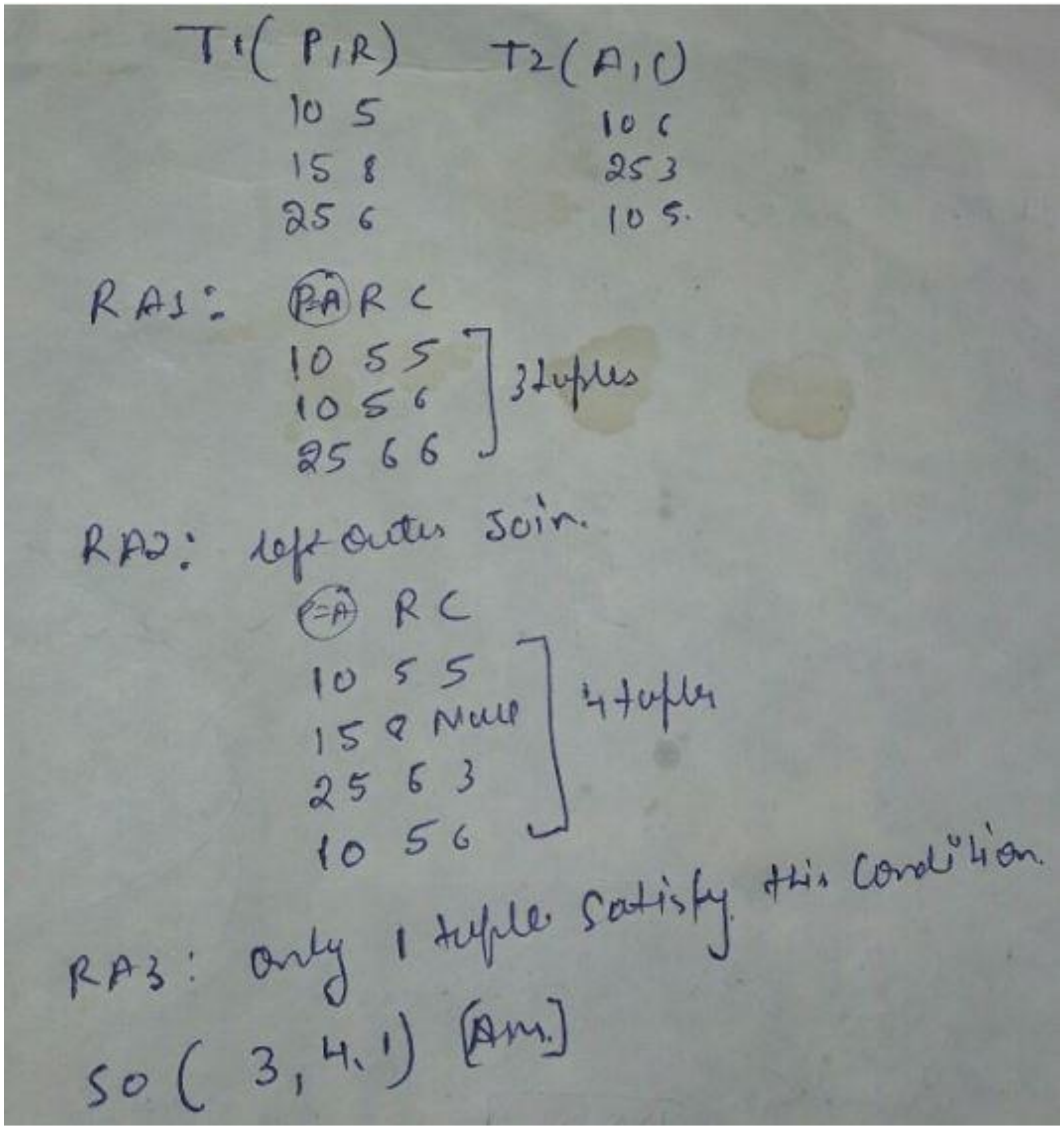
RA_3: $T_1 \bowtie_{T1.P = T2.A \text{ and } T1.R = T2.C} T_2$

The number of tuples in the resulting table of RA_1, RA_2 and RA_3 are given by:

(1) 2, 4, 2 respectively **(2)** 2, 3, 2 respectively

(3) 3, 3, 1 respectively **(4)** 3, 4, 1 respectively

Ans: 4



T1(P,R)

P	R
10	5
15	8
25	6

T2(A,C)

A	C
10	6
25	3
10	5.

RA1:

P=A	R	C
10	5	5
10	5	6
25	6	6

3 tuples

RA2: left outer join.

P=A	R	C
10	5	5
15	8	Null
25	6	3
10	5	6

4 tuples

RA3: only 1 tuple satisfy this condition.

so (3, 4, 1) [Ans.]

Outer join Operations

- The outer-join operation is an extension of the join operation to deal with missing information.
- The outer join operation works in a manner similar to the natural join operation, but preserves those tuples that would be lost in a join by creating tuples in the result containing null values.
- We can use the outer-join operation to avoid this loss of information.
- There are actually three forms of the operation: left outer join, denoted ⟕ ; right outer join, denoted ⟖ ; and full outer join, denoted ⟗ .

Left Outer Join

- The left outer join (⟕) takes all tuples in the left relation that did not match with any tuple in the right relation, pads the tuples with null values for all other attributes from the right relation, and adds them to the result of the natural join.

R_1	
A	B
1	P
2	Q
3	R

R_2	
B	C
Q	X
R	Y
S	Z

R_1 ⟕ R_2		
A	B	C
1	P	NULL
2	Q	X
3	R	Y

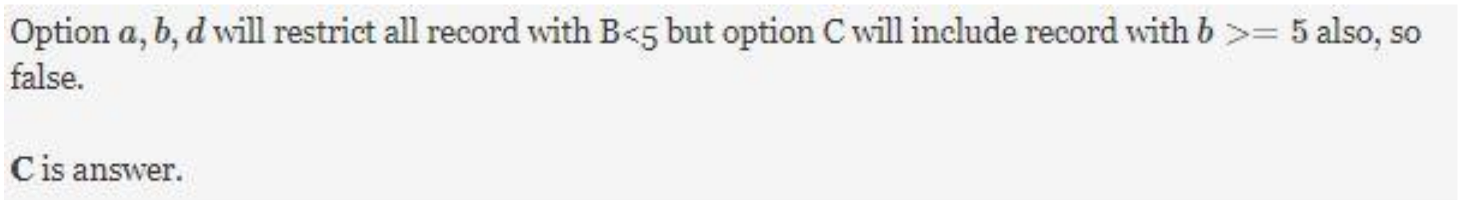
Q Consider the relations r(A, B) and s(B, C), where s.B is a primary key and r.B is a foreign key referencing s.B. Consider the query

Q: $r \bowtie (\sigma_{B<5}(s))$

Let LOJ denote the natural left outer-join operation. Assume that r and s contain no null values. Which of the following is NOT equivalent to Q? **(Gate-2018) (2 Marks)**

a) $\sigma_{B<5}(r \bowtie s)$ **b)** $\sigma_{B<5}$(r LOJ s)

c) r LOJ ($\sigma_{B<5}$(s)) **d)** $\sigma_{B<5}$(r) LOJ s



Option a, b, d will restrict all record with B<5 but option C will include record with $b >= 5$ also, so false.

C is answer.

r

A	B
7	5
7	1
8	1

s

B	C
1	4
2	6
5	6

Right Outer Join

- The right outer join (⟖) is symmetric with the left outer join: It pads tuples from the right relation that did not match any from the left relation with nulls and adds them to the result of the natural join.

R_1	
A	B
1	P
2	Q
3	R

R_2	
B	C
Q	X
R	Y
S	Z

$R_1 ⟖ R_2$		
A	B	C
2	Q	X
3	R	Y
NULL	S	Z

Full Outer Join

- The full outer join(⟗) does both the left and right outer join operations, padding tuples from the left relation that did not match any from the right relation, as well as tuples from the right relation that did not match any from the left relation, and adding them to the result of the join.

R_1	
A	B
1	P
2	Q
3	R

R_2	
B	C
Q	X
R	Y
S	Z

R_1 ⟗ R_2		
A	B	C
1	P	NULL
2	Q	X
3	R	Y
NULL	S	Z

Q Consider two relations R1(A, B) with the tuples (1, 5), (3, 7) and R1(A, C) = (1, 7), (4, 9). Assume that R(A, B, C) is the full natural outer join of R1 and R2. Consider the following tuples of the form (A, B, C)

a = (1, 5, null),

b = (1, null, 7),

c = (3, null, 9),

d = (4, 7, null),

e = (1, 5, 7),

f = (3, 7, null),

g = (4, null, 9).

Which one of the following statements is correct? **(Gate-2015) (1 Marks)**

(A) R contains a, b, e, f, g but not c, d
(B) R contains a, b, c, d, e, f, g
(C) R contains e, f, g but not a, b
(D) R contains e but not f, g

Answer: (C)

Q Consider the following 2 tables

R_1

A	B	C
1	2	3
1	2	4
2	1	3
3	1	3

R_2

A	B	D
1	2	1
2	1	5
4	2	1
3	2	1

The number of rows where null entries are present in the table $R_1 ⟗ R_2$ (R_1 natural full outer join R_2) is ______________

Ans: 3

DIVISION

- Notation – Division is also a binary operator denoted like A ÷ B
- *A* ÷ *B* as the set of all *x* values (in the form of unary tuples) such that for *every y* value in (a tuple of) *B*, there is a tuple *(x, y)* in *A*.
- For each *x* value in (the first column of) *A*, consider the set of *y* values that appear in (the second field of) tuples of *A* with that *x* value. If this set contains (all *y* values in) *B*, the *x* value is in the result of *A* ÷ *B*.
- Division operator is derived using- ***projection, cartisan product, set difference as***

$$A \div B = \prod_x(A) - \prod_x((\prod_x(A \times B) - A))$$

Q Consider the given table R and S find the number of elements retrieved by the query

Table R

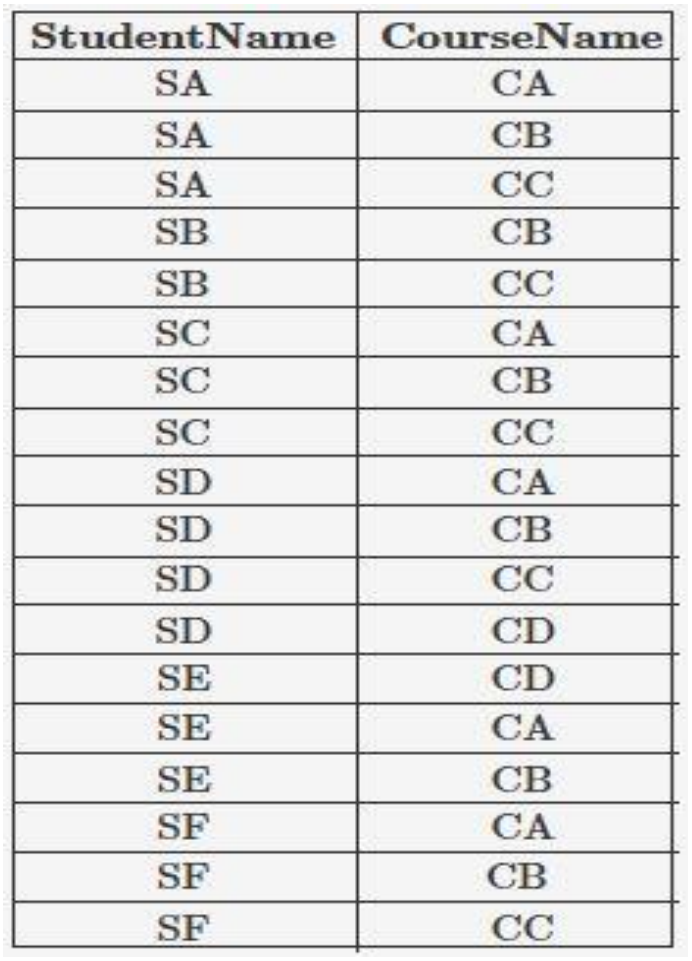
A	B
1	Dog
1	Cat
1	Cow
2	Cat
4	Cat
3	Dog
4	Dog
2	Dog
4	Cow

Table S

B
Cat
Dog

$\Pi_{A,B}(R) \div \Pi_B(S)$____________________

Q Consider a database that has the relation schema CR (StudentName, CourseName). An instance of the schema CR is as given below.



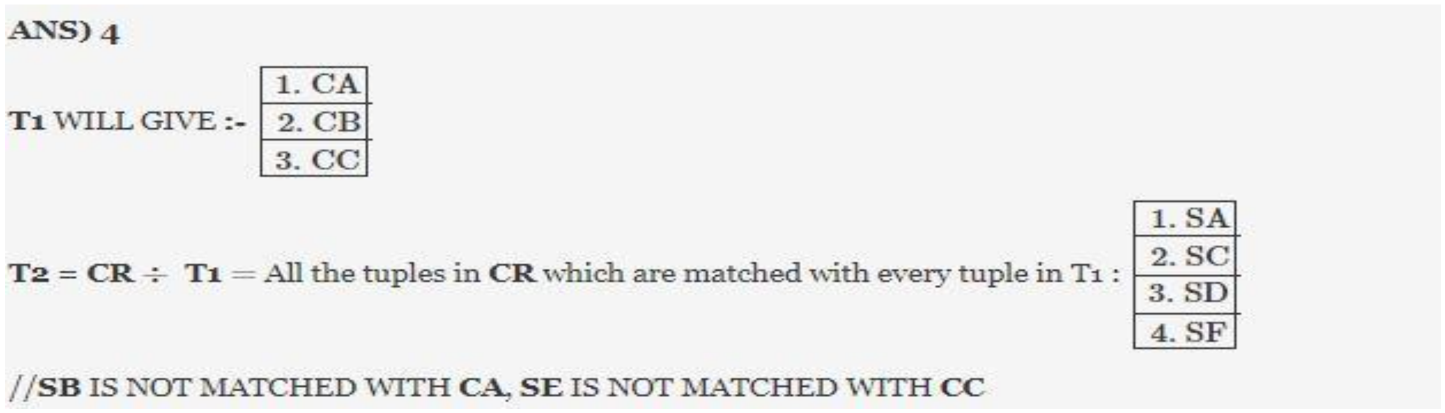
StudentName	CourseName
SA	CA
SA	CB
SA	CC
SB	CB
SB	CC
SC	CA
SC	CB
SC	CC
SD	CA
SD	CB
SD	CC
SD	CD
SE	CD
SE	CA
SE	CB
SF	CA
SF	CB
SF	CC

The following query is made on the database.

$T_1 \leftarrow \pi_{\text{CourseName}} (\sigma_{\text{StudentName=SA}}(CR))$

$T_2 \leftarrow CR \div T_1$

The number of rows in T2 is _______________ . **(Gate-2017) (1 Marks)**



ANS) 4

T1 WILL GIVE :-

1. CA
2. CB
3. CC

T2 = **CR** ÷ **T1** = All the tuples in **CR** which are matched with every tuple in T1 :

1. SA
2. SC
3. SD
4. SF

//**SB** IS NOT MATCHED WITH **CA**, **SE** IS NOT MATCHED WITH **CC**

Q Information about a collection of students is given by the relation *studInfo(studId, name, sex)*. The relation *enroll (studId, courseId)* gives which student has enrolled for (or taken) that course(s). Assume that every course is taken by at least one male and at least one female student. What does the following relational algebra expression represent?(Gate-2007) (2 Marks)

$\pi_{courceId}((\pi_{studId}(\sigma_{sex="female"}(studInfo)) \times \pi_{courseId}(enroll)) - enroll)$

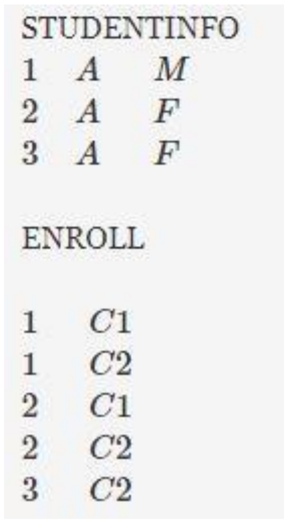
(A) Courses in which all the female students are enrolled.

(B) Courses in which a proper subset of female students are enrolled.

(C) Courses in which only male students are enrolled.

(D) None of the above

Answer: (B)



STUDENTINFO

1	A	M
2	A	F
3	A	F

ENROLL

1	$C1$
1	$C2$
2	$C1$
2	$C2$
3	$C2$

Q Consider the relational schema given below, where eId of the relation dependent is a foreign key referring to empId of the relation employee. Assume that every employee has at least one associated dependent in the dependent relation. (Gate-2014) (2 Marks)

employee (<u>empId</u>, empName, empAge)

dependent(<u>depId</u>, eId, depName, depAge)

Consider the following relational algebra query:

$$\Pi_{empId}(employee) - \Pi_{empId}(employee \bowtie_{(empId=eID)\wedge(empAge\leq depAge)} dependent)$$

The above query evaluates to the set of *empIds* of employees whose age is greater than that of

(A) some dependent.

(B) all dependents.

(C) some of his/her dependents

(D) all of his/her dependents.

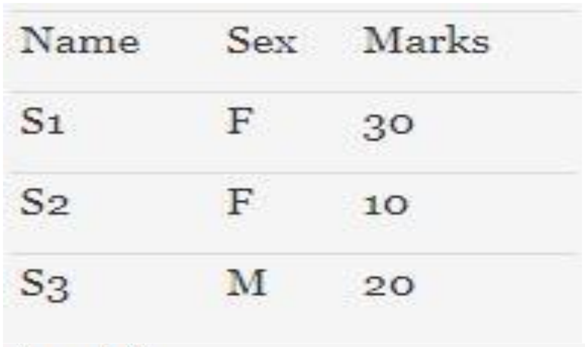
Answer: (D)

(D) all of his/her dependents. The inner query selects the employees whose age is less than or equal to at least one of his dependents. So, subtracting from the set of employees, gives employees whose age is greater than all of his dependents.

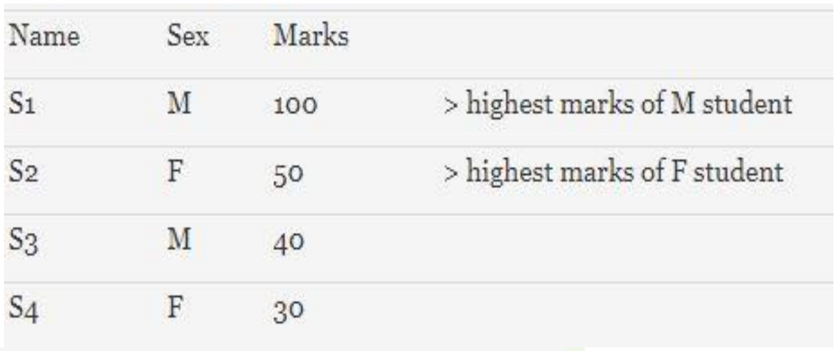
Q Consider the relation Student (name, sex, marks), where the primary key is shown underlined, pertaining to students in a class that has at least one boy and one girl. What does the following relational algebra expression produce? (Note: ρ is the rename operator). **(Gate-2004) (2 Marks)**

$\pi_{name}\{\sigma_{sex=female}(Student)\} - \pi name(Student \bowtie_{(sex=female \wedge x=male \wedge marks \leq m)} \rho_{n,x,m}(Student))$

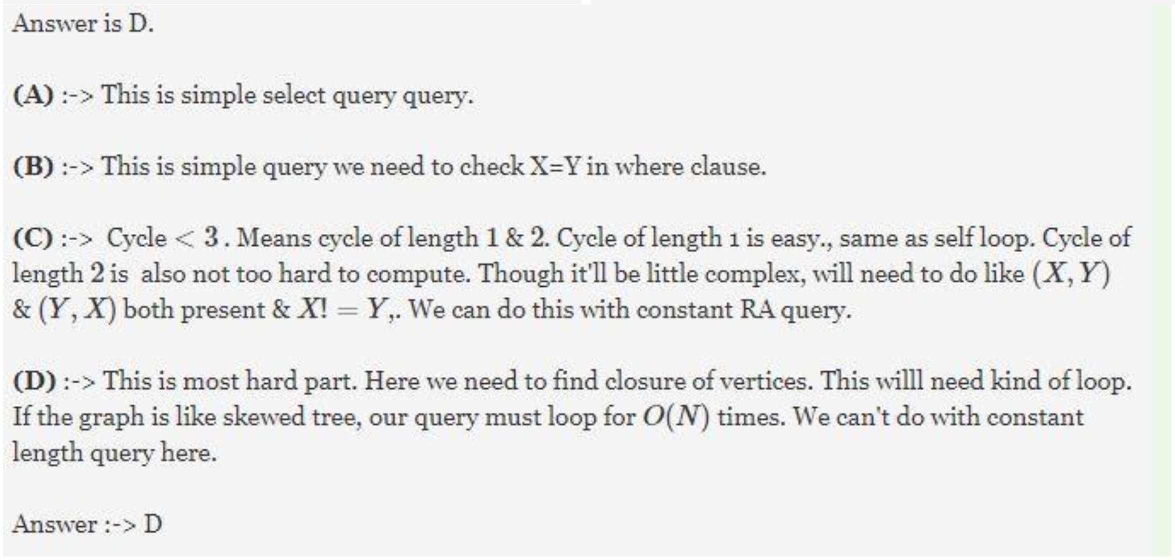
a) names of girl students with the highest marks
b) names of girl students with more marks than some boy student
c) names of girl students with marks not less than some boy student
d) names of girl students with more marks than all the boy students
ans: d



Name	Sex	Marks
S1	F	30
S2	F	10
S3	M	20



Name	Sex	Marks	
S1	M	100	> highest marks of M student
S2	F	50	> highest marks of F student
S3	M	40	
S4	F	30	



Answer is D.

(A) :-> This is simple select query query.

(B) :-> This is simple query we need to check X=Y in where clause.

(C) :-> Cycle < 3 . Means cycle of length 1 & 2. Cycle of length 1 is easy., same as self loop. Cycle of length 2 is also not too hard to compute. Though it'll be little complex, will need to do like (X, Y) & (Y, X) both present & $X! = Y$,. We can do this with constant RA query.

(D) :-> This is most hard part. Here we need to find closure of vertices. This willl need kind of loop. If the graph is like skewed tree, our query must loop for $O(N)$ times. We can't do with constant length query here.

Answer :-> D

Q With respect to relational algebra, which of the following operations are included from mathematical set theory? **(NET-JUNE-2019)**

1) Join **2)** Intersection **3)** Cartesian Product **4)** Project

a) 1 and 4 **b)** 2 and 3 **c)** 3 and 4 **d)** 2 and 4

Ans: b

Q Given the relations **(Gate-2000) (2 Marks)**

employee (name, salary, dept-no), and

department (dept-no, dept-name, address),

Which of the following queries cannot be expressed using the basic relational algebra operations (σ, π, ×, ⋈, ∪, ∩, −)?

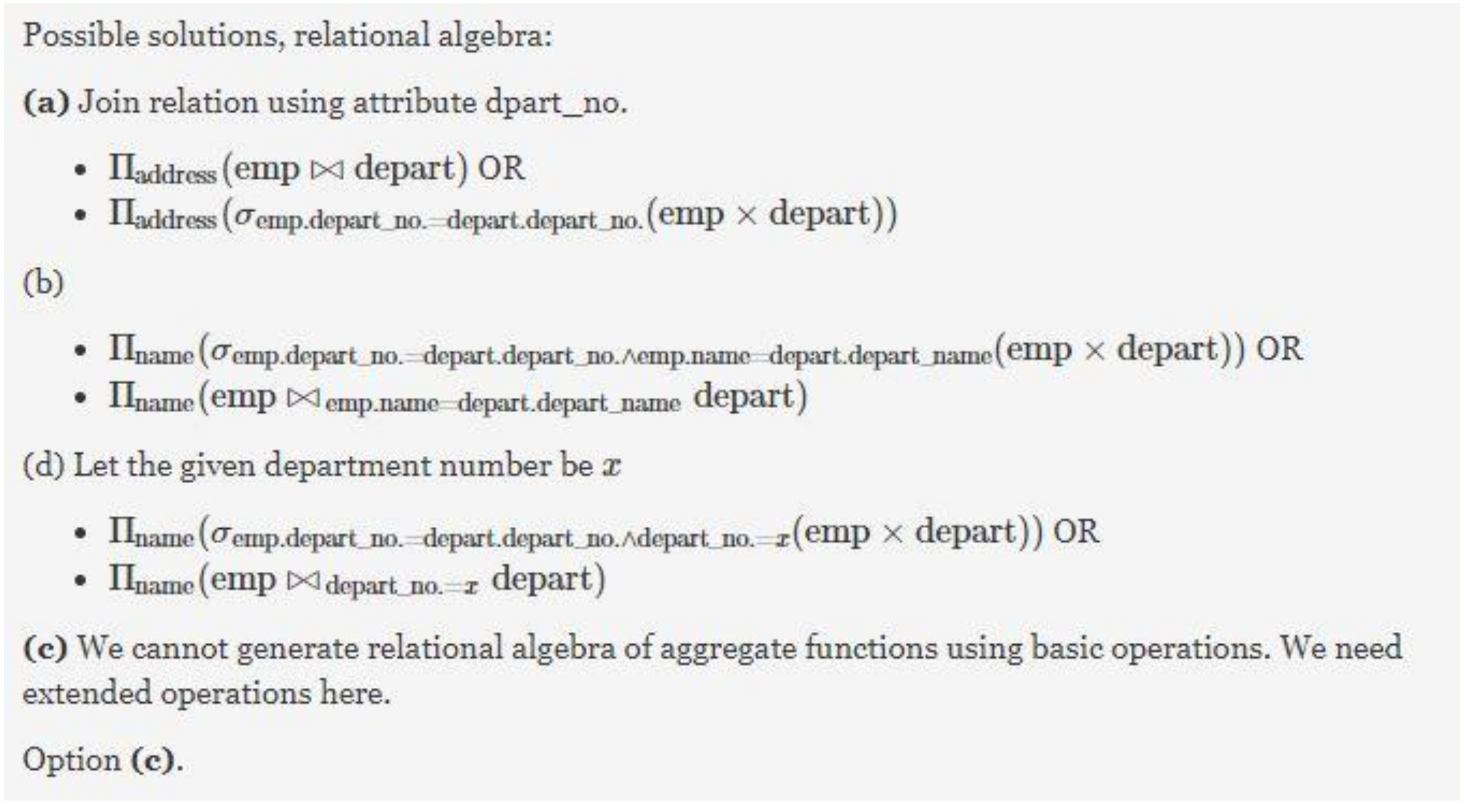
a) Department address of every employee

b) Employees whose name is the same as their department name

c) The sum of all employees' salaries

d) All employees of a given department

Ans: c



Possible solutions, relational algebra:

(a) Join relation using attribute dpart_no.

- $\Pi_{\text{address}}(\text{emp} \bowtie \text{depart})$ OR
- $\Pi_{\text{address}}(\sigma_{\text{emp.depart\_no.}=\text{depart.depart\_no.}}(\text{emp} \times \text{depart}))$

(b)

- $\Pi_{\text{name}}(\sigma_{\text{emp.depart\_no.}=\text{depart.depart\_no.} \wedge \text{emp.name}=\text{depart.depart\_name}}(\text{emp} \times \text{depart}))$ OR
- $\Pi_{\text{name}}(\text{emp} \bowtie_{\text{emp.name}=\text{depart.depart\_name}} \text{depart})$

(d) Let the given department number be x

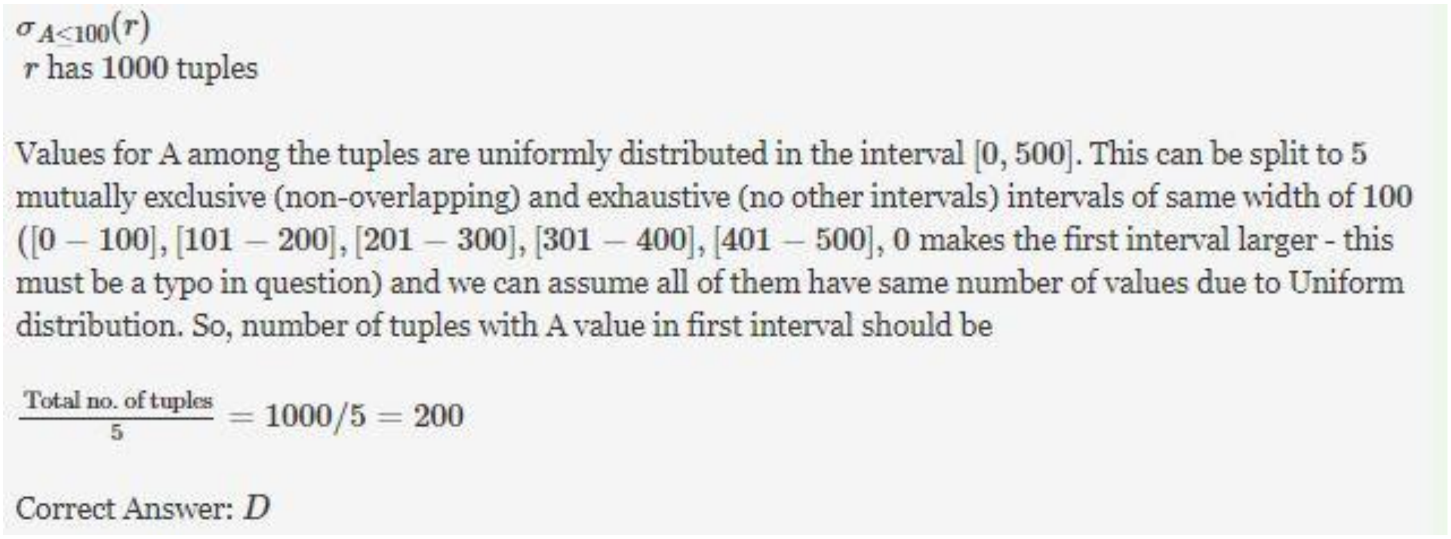
- $\Pi_{\text{name}}(\sigma_{\text{emp.depart\_no.}=\text{depart.depart\_no.} \wedge \text{depart\_no.}=x}(\text{emp} \times \text{depart}))$ OR
- $\Pi_{\text{name}}(\text{emp} \bowtie_{\text{depart\_no.}=x} \text{depart})$

(c) We cannot generate relational algebra of aggregate functions using basic operations. We need extended operations here.

Option **(c)**.

Q Consider a selection of the form $\sigma_{A \leq 100}(r)$, where r is a relation with 1000 tuples. Assume that the attribute values for A among the tuples are uniformly distributed in the interval [0,500]. Which one of the following options is the best estimate of the number of tuples returned by the given selection query? **(Gate-2007) (2 Marks)**

a) 50 **b)** 100 **c)** 150 **d)** 200



$\sigma_{A<100}(r)$
r has 1000 tuples

Values for A among the tuples are uniformly distributed in the interval $[0, 500]$. This can be split to 5 mutually exclusive (non-overlapping) and exhaustive (no other intervals) intervals of same width of 100 ($[0-100], [101-200], [201-300], [301-400], [401-500]$, 0 makes the first interval larger - this must be a typo in question) and we can assume all of them have same number of values due to Uniform distribution. So, number of tuples with A value in first interval should be

$$\frac{\text{Total no. of tuples}}{5} = 1000/5 = 200$$

Correct Answer: D

Q Consider a relational table r with sufficient number of records, having attributes $A_1, A_2\ldots, A_n$ and let $1 \le p \le n$. Two queries Q_1 and Q_2 are given below. **(Gate-2011) (2 Marks)**

Q_1: $\pi_{A1,\ldots,Ap}$ ($\sigma_{Ap=c}$ (r)) where cc is a constant

Q_2: $\pi_{A1,\ldots,Ap}$ ($\sigma_{c1 \le Ap \le c2}$ (r)) where c1 and c2 are constants.

The database can be configured to do ordered indexing on Ap or hashing on Ap. Which of the following statements is TRUE?

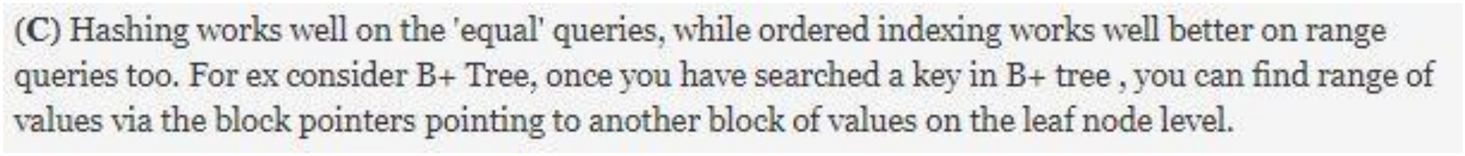
a) Ordered indexing will always outperform hashing for both queries

b) Hashing will always outperform ordered indexing for both queries

c) Hashing will outperform ordered indexing on Q_1, but not on Q_2

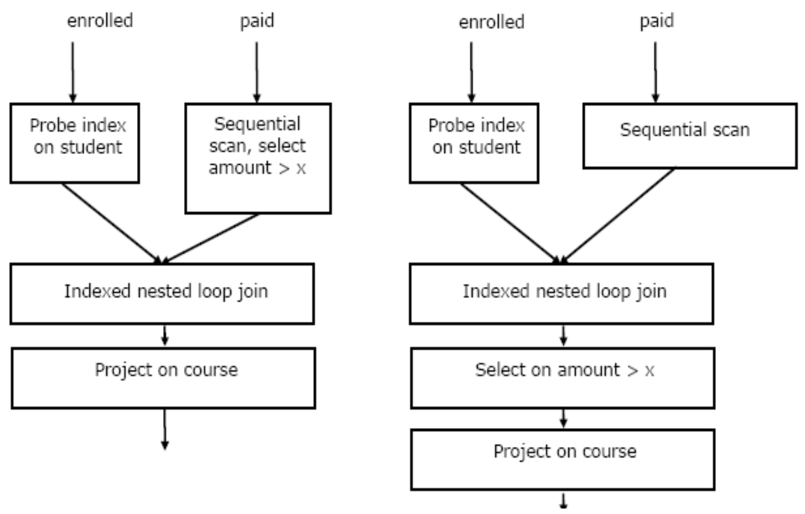
d) Hashing will outperform ordered indexing on Q_2, but not on Q_1

Ans: c



(C) Hashing works well on the 'equal' queries, while ordered indexing works well better on range queries too. For ex consider B+ Tree, once you have searched a key in B+ tree , you can find range of values via the block pointers pointing to another block of values on the leaf node level.

Q Consider the relation enrolled (student, course) in which (student, course) is the primary key, and the relation paid (student, amount), where student is the primary key. Assume no null values and no foreign keys or integrity constraints. Assume that amounts 6000, 7000, 8000, 9000 and 10000 were each paid by 20% of the students. Consider these query plans (Plan 1 on left, Plan 2 on right) to “list all courses taken by students who have paid more than x”. **(Gate-2006) (2 Marks)**



A disk seek takes 4ms, disk data transfer bandwidth is 300 MB/s and checking a tuple to see if amount is greater than x takes 10 micro-seconds. Which of the following statements is correct?

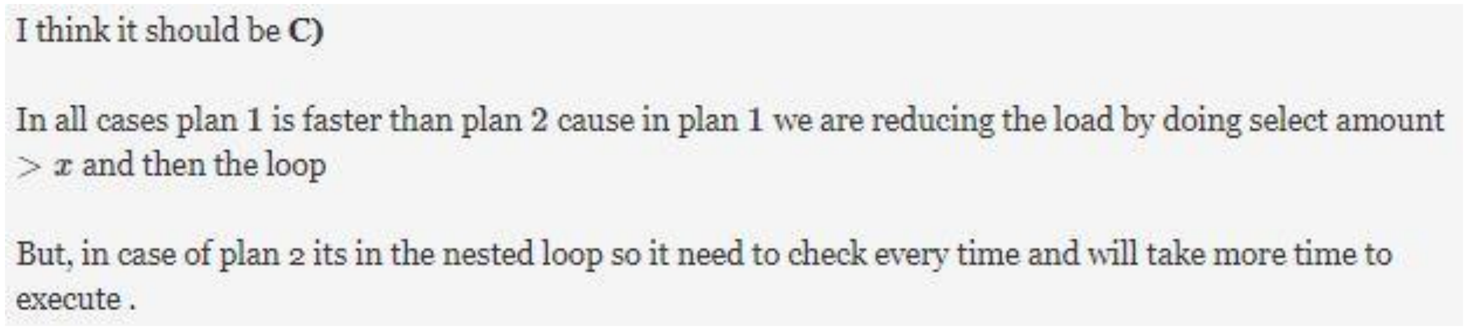
(A) Plan 1 and Plan 2 will not output identical row sets for all databases.

(B) A course may be listed more than once in the output of Plan 1 for some databases

(C) For x = 5000, Plan 1 executes faster than Plan 2 for all databases.

(D) For x = 9000, Plan I executes slower than Plan 2 for all databases.

Answer: (C)



I think it should be **C)**

In all cases plan 1 is faster than plan 2 cause in plan 1 we are reducing the load by doing select amount $> x$ and then the loop

But, in case of plan 2 its in the nested loop so it need to check every time and will take more time to execute .

Q Consider a join (relation algebra) between relations r(R)and s(S) using the nested loop method. There are 3 buffers each of size equal to disk block size, out of which one buffer is reserved for intermediate results. Assuming size(r(R)) < size(s(S)), the join will have fewer number of disk block accesses if **(Gate-2014) (2 Marks)**

a) relation r(R) is in the outer loop.

b) relation s(S) is in the outer loop.

c) join selection factor between r(R) and s(S) is more than 0.5.

d) join selection factor between r(R) and s(S) is less than 0.5.

Ans: a

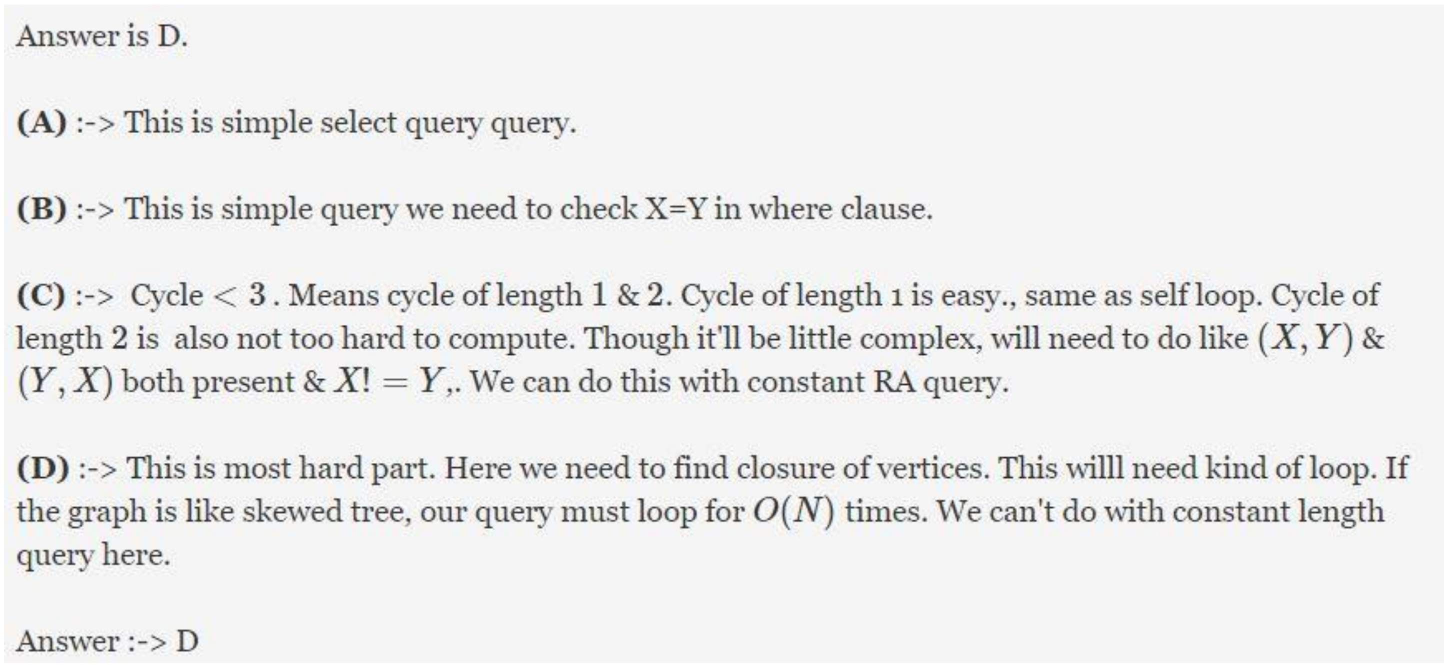
Q Suppose the adjacency relation of vertices in a graph is represented in a table Adj (X, Y). Which of the following queries cannot be expressed by a relational algebra expression of constant length?**(Gate-2001) (1 Marks)**

(A) List of all vertices adjacent to a given vertex

(B) List all vertices which have self-loops

(C) List all vertices which belong to cycles of less than three vertices

(D) List all vertices reachable from a given vertex



Answer is D.

(A) :-> This is simple select query query.

(B) :-> This is simple query we need to check X=Y in where clause.

(C) :-> Cycle < 3. Means cycle of length 1 & 2. Cycle of length 1 is easy., same as self loop. Cycle of length 2 is also not too hard to compute. Though it'll be little complex, will need to do like (X, Y) & (Y, X) both present & $X! = Y$,. We can do this with constant RA query.

(D) :-> This is most hard part. Here we need to find closure of vertices. This willl need kind of loop. If the graph is like skewed tree, our query must loop for $O(N)$ times. We can't do with constant length query here.

Answer :-> D