Minimal set of Canonical cover:

FD set without any redundant attributes.

How to find minimal cover or canonical ~~cov~~ cover:-

1) split the FD's such that RHS contains single attribute

Ex: $A \to BC \Rightarrow A \to B,\ A \to C.$

$AB \to CD \Rightarrow AB \to C,\ AB \to D$

2) Find the redundant FD and delete it from the set

Ex: $A \to B,\ B \to C,\ \cancel{A \to C}$.

$A^+ = ABC$

3) Find the redundant attributes on LHS and delete them

Ex: $AB \rightarrow C$

A^+ contain B. $\rightarrow$ B ✗

$A^+ \rightarrow C$.

$A^+ \rightarrow B$.

Ex: $A \to C \quad AC \to D \quad E \to AD \quad E \to H$

Step 1) RHS should contain only 1 attribute

$A \to C$, $AC \to D$, $E \to A$, $\cancel{E \to D}$, $E \to H$.

Step 2: $A^+ = A$ not contain C.

$AC^+ = AC$ not contain D.

$E^+ = EDH$

$E^+ = EACD$

$E^+ = EACD$ → does not contain 'H'

Step 3 : $A \to C$ $\cancel{AC \to D}$ $E \to A$ $E \to H$.

LHS of FD, check for redundancy.

$AC \to D$

remove it, find $C^+ = C$, it doesnot contain 'A'.

so we ~~can~~ A is required.

$AC \to D$

remove C, find $A^+ = AC$ ✓

$\therefore$ C is not required.

$A \to D$. ✓

Finally: $A \to C \quad A \to D \quad E \to A \quad E \to H$

$A \to CD \qquad E \to AH \qquad F_2.$

Given: $\{A \to C,\ AC \to D,\ E \to AD,\ E \to H\} = F_1$

$F_1 \supseteq F_2, \quad F_2 \supseteq F_1. \quad F_1 = F_2$

F_1 is canonical cover of F_2.

Queries:

Procedural queries

non procedural ~~quaries~~ queries

Procedural queries:

→ Formulations of how to retrieve and what data to retrieve from DB table.

Procedure

Ex: Relational algebra queries.

Non procedural query languages:

formulation of queries such that what data to retrive from DB table. (no how) → not procedural.

Ex: Relational Calculus queries

[uses First order logic, predicate calculus formulas]

Relational Calculus

Tuple Relational Calculus

⇓

SQL is close to TRC.

First version of SQL is developed by IBM

Domain relational Calculus

⇓

QBE (query by example)

close to DRC.

FV of QBC is developed by MS.

Relational algebra:

Basic operators:

pi - π : projection

Rho ← ρ : rename

Sigma ← σ : selection

unary operators
↓
on one table

$\times$: cross product

$\cup$: union

$-$: set difference

Binary operators
on two tables
or two instances of the same table

Derived operators $\rightarrow$ operators derived from Basic operators

natural / conditional

$\bowtie$: Join (using $\times$, σ, π)

$\cap$: intersection (using $-$)

$/$: Division (using $\times$, π, $-$)

Binary operators.

π : projection :

It projects required attributes from 'R'.

R

A	B	C	D

$\pi_{AB}(R)$

A	B

✓

σ: Selection:

$\sigma_p(R)$: Retrieves records of 'R' those satisfy condition (p).

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

In relational algebra tuples can be duplicated.

allowed.

Select doesnt change no of attributes.

$\sigma_{A>5}(R)$

=

A	B	C
8	5	4
7	3	4

no duplicates.

$\pi_{BC}(R)$

B	C
6	8
5	4
3	4

o/p of any RA query does not contain duplicates

Which is true

(i) π is commutative ✗

(ii) $\cap$ is commutative

a) (i) b) (ii) c) (ii), (i) d) none.

ABCD

AB

~~ABC~~ ✗

R: ABCD

$$\underset{AB}{\pi_{list2}}\left(\underset{ABC}{\pi_{list1}}(R)\right) \overset{?}{=} \underset{ABC}{\pi_{list1}}\left(\underset{AB}{\pi_{list2}}(R)\right)$$

A	B	C	D

list

A	B	C

list2

A	B

σ is commutative ✓

$$\sigma_{c_2}(\sigma_{c_1}(R)) \overset{?}{=} \sigma_{c_1}(\sigma_{c_2}(R)) = \underline{\underline{\sigma_{c_1 \text{ and } c_2}(R)}}$$

A R → Database of a university

c_1 → CS Branch = CS

c_2 → marks > 60%

$$\underbrace{\sigma_{\substack{\text{marks}\\>60\%}}(\underbrace{\sigma_{\text{Branch}=CS}(R)}_{\text{CS Student}})}_{\text{CS students whose marks is}>60\%} = \underbrace{\sigma_{\substack{\text{Branch}\\=CS}}(\underbrace{\sigma_{\substack{\text{marks}\\>60\%}}(R)}_{>60\%})}_{\text{CS with 60\%}}$$

Cross product:

R X S all attributes of R followed by all attributes of 'S' and each record of R pairs with each ~~record~~ record of 'S'.

R

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

S

C	D
5	7
7	8

R x S

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

→ attributes are not distinct.

final table attribute need not be distinct.

R

A	B
1	2
3	4

S

C	D
5	6
7	8

R X S

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

→ attri are unique.

X → attri may or maynot be distinct

→ If 'R' has 'x' attributes and S has 'y' attributes then

R×S will have x+y attributes

→ If R has n disple tuples, S has m distinct tuples, then

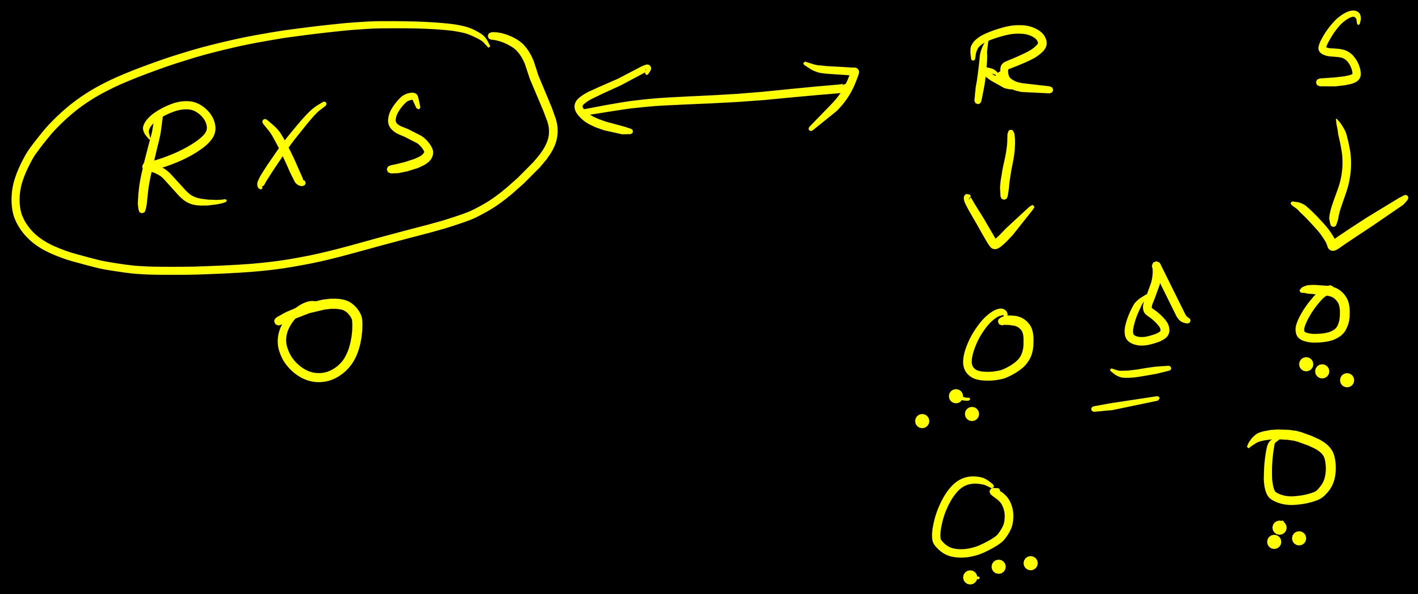
R×S has m×n tuples

→ If R has n tuples and S has 0 tuples then R×S has

0 tuples.

→ If R×S results in Ed Empty resl set, then either

R & S relation should be with 0 tuples.


R x S
O
R
S

Joins: very important:

1) Natural join: $R \bowtie S$

$$R \bowtie S = \pi_{\text{distinct attributes}}\left(\sigma_{\text{equality b/w common attributes of R and S}}(R \times S)\right)$$

R = S

A	B	C
a	1	2

$\times$

B	C	D
1	2	d

=

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

$= \sigma$

A	B	C	B	C	D

π distinct attributes

A	B	C	D

A	B	C	D
a	1	2	d

Ex:

R

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

S

C	D
5	7
7	8

$R \bowtie S$
↓
natural join

① $\times$ product

$R.C = S.C$

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

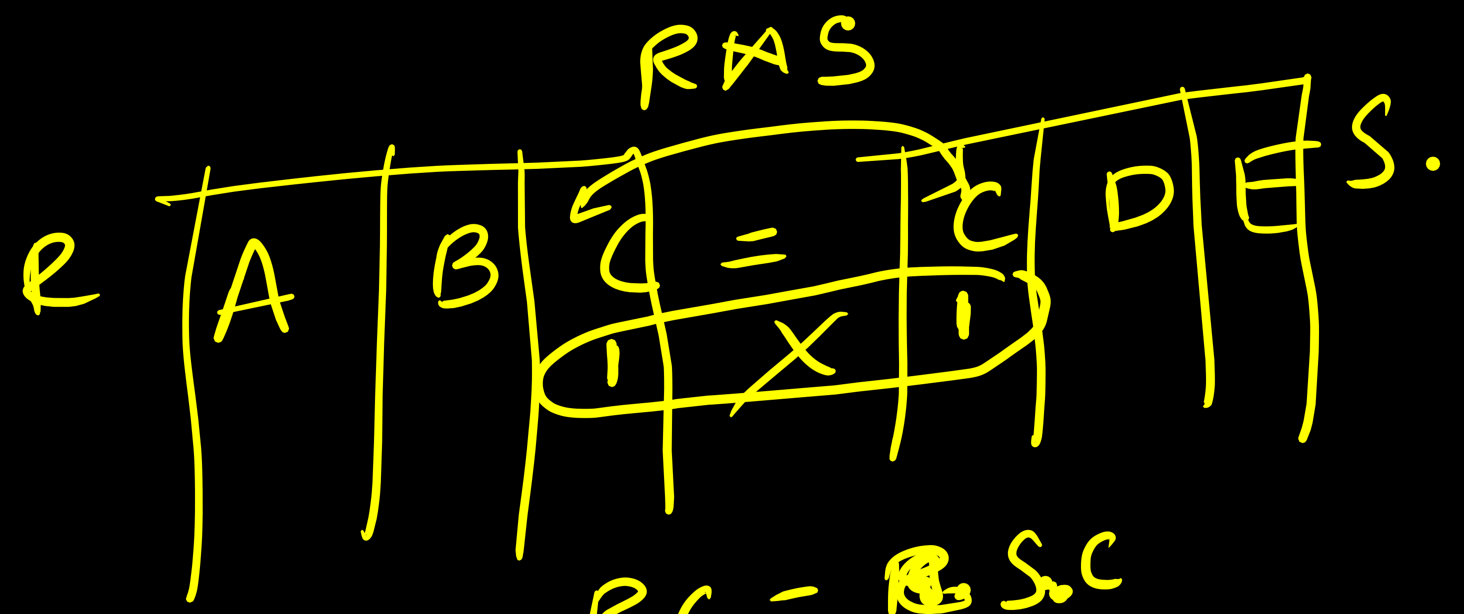
② equality on common attributes.

A	B	C	C	D
4	2	5	5	7
3	4	5	5	7

③ project only distinct attributes.

$R \bowtie S$
↓
natural join

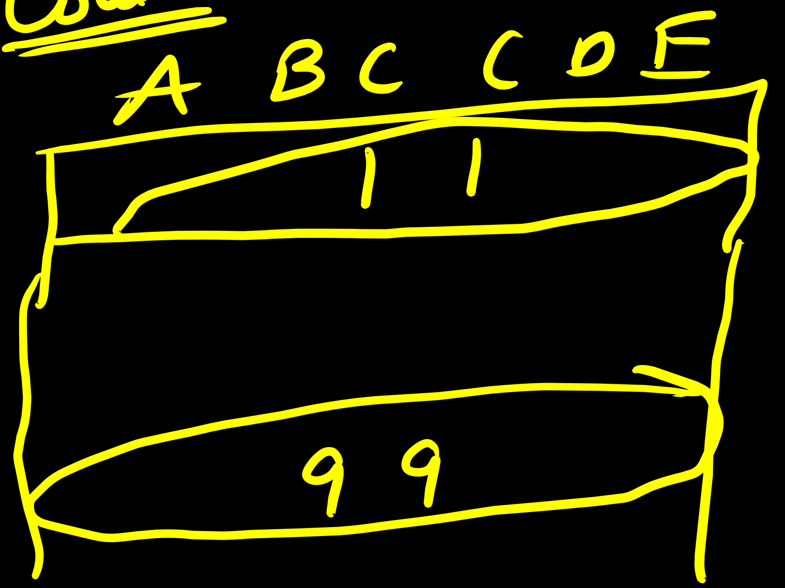
A	B	C	D
4	2	5	7
3	4	5	7



R.C = S.C

attributes → Columns

not data elements:



R.B = S.B and
R.C = S.C.

$R \bowtie S = ?$ if there are no common attributes.

Then $R \bowtie S = R \times S$.

---

if there are common attributes

$$\boxed{R \bowtie S = \pi_{\text{distn attrc.}}\left(\sigma_{\text{comm attr equ}}(R \times S)\right)}$$

$R(ABC) \quad S(BCDE)$

$$R \bowtie S = \pi_{ABCDE}\left(\sigma_{\substack{R.B=S.B \\ R.C=S.C}}(R \times S)\right)$$

multiple subjects will go on very soon.

A	B	C	D	E

6 pm - 7pm

Plan about DSA.

7 - 9 DBMS.