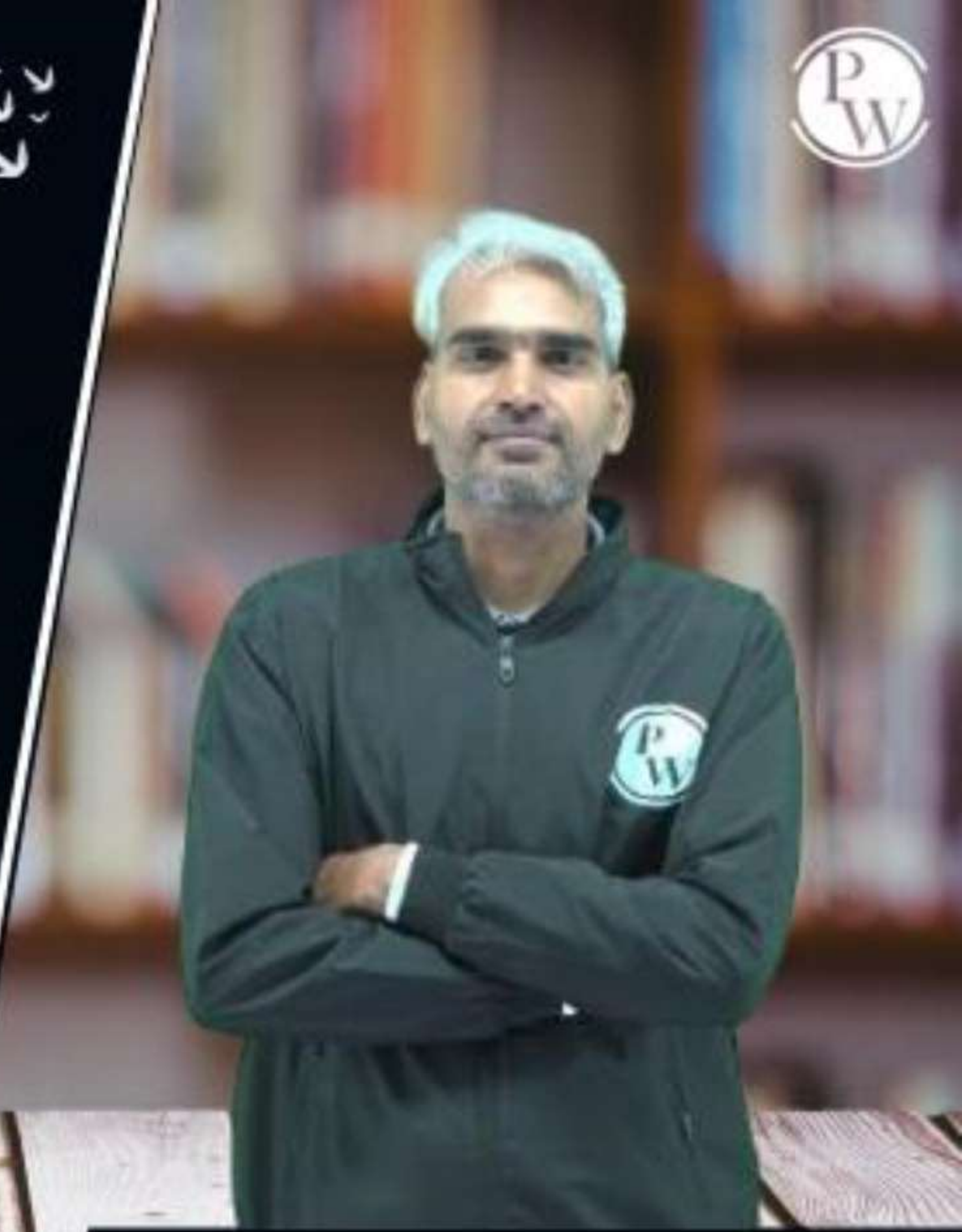
Computer Science & IT

Database Management System

Relational Model & Normal Forms

Lecture No. 11



By- Vishal Sir

Recap of Previous Lecture

Topic: Properties of decomposition

- Dep. preserving
- Lossless Join decomposition

Topic: Dependency preserving decomposition

Slide

Topics to be Covered

$\bowtie$ (Natural Join)

- Topic — Lossless join decomposition
- Topic — Projection, Selection, and Cross product
- Topic — Natural join
- Topic — Lossless Natural Join

Slide

Topic : Lossless Join decomposition

If we decompose a relation R with FD set F into sub-relations R1, R2,.....,Rn with FD sets F1, F2,.....,Fn respectively, then for this decomposition to be called lossless join decomposition following property must hold true.

$$R1 \bowtie R2 \bowtie .... \bowtie Rn = R$$

Topic : Lossless Join decomposition

Let relation R is decomposed into sub-relations R1, R2,.....,Rn

In general, $R1 \bowtie R2 \bowtie .... \bowtie Rn \supseteq R$

if, $R1 \bowtie R2 \bowtie .... \bowtie Rn = R$ then, Lossless join decomposition

if, $R1 \bowtie R2 \bowtie .... \bowtie Rn \supset R$ then, Lossy join decomposition

$R1 \bowtie R2 \bowtie .... \bowtie Rn \subset R$ (not possible)

Topic : Natural Join (⋈)

Natural Join(⋈) is a derived Relational Algebra operation, which is derived using three basic Relational Algebra operation

- Projection (π)
- Selection(σ)
- Cross Product ($\times$)

Topic : Projection (π)

It is used to project the column data from a relation based on the attributes specified with projection operation.

$$\pi_{\underbrace{A_1, A_2, \ldots}_{\text{list of attributes required in the output.}}}(\text{Relation\_name})$$

Enroll

Sid	Cid	Branch
S_1	C_1 ✓	CS
S_2	C_2 ✓	IT
S_3	C_1	CS

$$\underbrace{\pi_{Sid,\,Cid,\,Branch}(Enroll)} \equiv \underbrace{(Enroll)}$$

Output of this query will be Complete Enroll table

$\pi_{Sid}(Enroll) \equiv$

O/P

Sid
S_1
S_2
S_3

$\pi_{Cid}(Enroll) \equiv$

O/P

Cid
C_1
C_2
~~C_1~~

$=$

Cid
C_1
C_2

Relational Algebra query will always produce distinct tuples

$\pi_{sid,\,Cid}(Enroll) \equiv$

O/P

Sid	Cid
S_1	C_1
S_2	C_2
S_3	C_1

Topic : Selection(σ)

It is used to select the tuples(records) from underlying relation based on the predicate condition specified with selection operation.

Syntax :-

$$\sigma_{(\text{Predicate Cond}^n)}(\text{Relation\_name})$$

Enroll

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

Q: Retrieve the Sids of the students who enrolled for Course "C_1"

$$\underbrace{\pi_{Sid}}_{\text{it will project the Sids of the students from the selected tuples}} \left(\underbrace{\sigma_{(Cid = C_1)} (Enroll)}_{\text{it will select the tuples in which } Cid = C_1} \right) \equiv$$

o/p

Sid
S_1
S_3

* Retrieve the records with $Cid = C_1$

$$\sigma_{(Cid = C_1)} (Enroll) \equiv$$

o/p

Sid	Cid	Branch
S_1	C_1	CS
S_3	C_1	CS

Retrieve the Sids of the students who enrolled for Course "C_1" or they are from IT Branch

$$\pi_{Sid} \left(\sigma_{Cid = C_1 \vee Branch = 'IT'} (Enroll) \right) \equiv$$

o/p

Sid
S_1
S_2
S_3

$$\sigma_{Cid = C_3}(Enroll) =$$

Sid	Cid	Branch

$$\pi_{Sid}\left(\sigma_{Cid = C_3}(Enroll)\right) =$$

Sid

Topic : Cross Product (×)

AKA, Cross Join (or) Cartesian Product

two relations as operand

Cross-product is a binary operation. Let R and S are any two relation, then cross product R × S will result in all attributes of R followed by all attribute of S with all possible combinations of tuples from R and S.

Enroll ≡ R (x attributes)

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

m tuples

Course ≡ S (y attributes)

Cid	Cname
C_1	OS
C_2	DBMS
C_3	OS

n tuples

R x S =

Attributes of R — 'x+y' attributes — Attributes of S

R.Sid	R.Cid	R.Branch	S.Cid	S.Cname
S_1	C_1	CS	C_1	OS
S_1	C_1	CS	C_2	DBMS
S_1	C_1	CS	C_3	OS
S_2	C_2	IT	C_1	OS
S_2	C_2	IT	C_2	DBMS
S_2	C_2	IT	C_3	OS
S_3	C_1	CS	C_1	OS
S_3	C_1	CS	C_2	DBMS
S_3	C_1	CS	C_3	OS

m.n tuples

- If relation R has 'X' attributes & relation S has 'y' attributes, then relation 'RxS' will have 'x+y' attributes
- If R has m tuples & S has n tuples then RxS will have 'm.n' tuples.

Topic : Natural Join (⋈)

Natural join(⋈) is a derived relational algebra operation which is derived using $\underbrace{\text{cross product}}$, $\underbrace{\text{selection}}$ and $\underbrace{\text{projection}}$ as follows:

- Let R and S are any two relations then,

$R \bowtie S$ = Step-1: Obtain "R × S"

Step-2: Select the tuples from "R × S" based on the equality condition on all common attributes of R and S.

Step-3: Project distinct attributes from the result of step-2.

Enroll ≡ R

x attributes

m tuples

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_2	IT
S_3	C_1	CS

Course ≡ S

y attributes

n tuples

Cid	Cname
C_1	OS
C_2	DBMS
C_3	OS

$R \times S =$

Attributes 'x+y' (Attributes of R, attributes of S)

m.n tuples

R.Sid	R.Cid	R.Branch	S.Cid	S.Cname
S_1	C_1	CS	C_1	OS
S_1	C_1	CS	C_2	DBMS
S_1	C_1	CS	C_3	OS
S_2	C_2	IT	C_1	OS
S_2	C_2	IT	C_2	DBMS
S_2	C_2	IT	C_3	OS
S_3	C_1	CS	C_1	OS
S_3	C_1	CS	C_2	DBMS
S_3	C_1	CS	C_3	OS

$$R \bowtie S = \underbrace{\pi_{R.Sid,\, R.Cid,\, R.Branch,\, S.Cname}}_{\text{Projection of distinct attributes}} \left(\sigma_{\underbrace{R.Cid = S.Cid}_{\text{Equality Condition on Common attributes of R \& S}}} (R \times S) \right)$$

$R \bowtie S =$ o/p =

Sid	Cid	Branch	Cname
S_1	C_1	CS	OS
S_2	C_2	IT	DBMS
S_3	C_1	CS	OS

eg: $R(A, B, C)$ & $S(C, D, E)$

$$R \bowtie S = \pi_{A,B,C,D,E}\left(\sigma_{R.C=S.C}(R \times S)\right)$$

eg: $R(A, B, C)$ & $S(B, C, D)$

$$R \bowtie S = \pi_{A,B,C,D}\left(\sigma_{R.B=S.B \,\wedge\, R.C=S.C}(R \times S)\right)$$

Equality Condition on all common attributes eg. $R(A,B,C)$ & $S(D,E)$

$$R \bowtie S = \pi_{A,B,C,D,E}\big(\underbrace{\qquad\qquad}_{\uparrow}(R \times S)\big) \equiv R \times S$$

No Common attribute between R & S

∴ No Selection Cond^n

↳ Hence all tuples of $R \times S$ will be Selected

If there is no Common attribute b/w R & S, then Natural join of R & S will degenerate into Crossproduct of R & S

R

A	B
1	1
2	4
3	3

S

B	C
2	1
5	4
7	7

$R \triangleright S$ = Empty Relation

Lossless Natural Join

eg: Consider the following relation R.

R =

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

① Let R is decomposed into two sub-relations $R_1(AB)$ & $R_2(BC)$ Check whether the decomposition is lossy or Lossless join decomposition

R_1

A	B
1	1
2	1
3	2

R_2

B	C
1	1
1	2
2	1

$R_1 \bowtie R_2 =$

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

tuples from R

Extra tuples (Spurious tuples)

Common attribute between R_1 & R_2 is B.

Values of B are neither unique in R_1 nor unique in R_2
{i.e. B is not a Super key of any of R_1 or R_2}
Hence Lossy join decomposition

$R_1 \bowtie R_2 \supset R$ ∴ Lossy join decomposition

eg: Consider the following relation R.

R =

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

② Let R is decomposed into two sub-relations $R_1(AB)$ & $R_2(AC)$ Check whether the decomposition is lossy or Lossless join decomposition

R_1

A	B
1	1
2	1
3	2

R_2

A	C
1	1
2	2
3	1

$R_1 \bowtie R_2$ =

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

$R_1 \bowtie R_2 = R$

∴ Lossless join decomposition.

- Common attribute b/w R_1 & R_2 is A.
- Values of A are unique in R_1 as well as in R_2

i.e. A is Super key of R_1 as well as R_2

Hence Lossless Join decomposition.

eg: Consider the following relation R.

R =

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

③ Let R is decomposed into two sub-relations $R_1(AB)$ & $R_2(BC)$ Check whether the decomposition is lossy or Lossless join decomposition

R_1

A	B
1	1
2	1
3	2

R_2

B	C
1	2
2	1

$R_1 \bowtie R_2 =$

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

Common attribute b/w R_1 & R_2 is B.

Values of B are not unique in R_1 but values of B are unique in R_2

i.e. B is a S.K. of relation R_2.

* Common attribute is a S.K of at least one of the two relations

∴ Lossless Join decomposition

$R_1 \bowtie R_2 = R$

∴ Lossless join decomposition.

eg: Consider the following relation R.

R =

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

(4) Let R is decomposed into two sub-relations $R_1(AB)$ & $R_2(AC)$ Check whether the decomposition is lossy or Lossless join decomposition

R_1

A	B
1	1
2	1
3	2

R_2

A	C
1	2
2	2
3	1

$R_1 \bowtie R_2$ =

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

Common attribute is A.

Values of A are unique in R_1 as well as in R_2

i.e. A is a S.K. of both relation

Hence, Lossless join decomposition.

$R_1 \bowtie R_2 = R$

∴ Lossless join decomposition.

Note :- If relation R is decomposed into two sub-relations R_1 & R_2, then decomposition is lossless join decomposition if and only if, following Conditions are satisfied.

① Attributes of R_1 $\cup$ Attributes of R_2 = Attributes of R

and ② Attributes of R_1 $\cap$ Attributes of R_2 $\neq \emptyset$

{ Because if there is no Common attribute then Natural join will degenerate into Cross Product & it is always lossy.

and ③ (Attributes of R_1 $\cap$ Attributes of R_2) must be a Super key of either R_1 or R_2 or both

Common attributes

#Q. Let R (A, B, C, D, E) be the relational schema with following FD set

$F = \{ AB \to C, C \to D, B \to E \}$

Which of the following decomposition is/are lossless join decomposition?

(i) {R1(ABC), R2(CD)}

Attributes of R_1 $\cup$ Attributes of $R_2 \neq R$ $\therefore$ Lossy

(ii) {R1(ABC), R2(DE)}

Attributes of R_1 $\cup$ Attributes of R_2 = Attributes of R ✓

Attributes of R_1 $\cap$ Attributes of $R_2 = \phi$ $\therefore$ lossy

(iii) {R1(ABC), R2(CDE)}

$R_1 \cup R_2 = R$ ✓

$R_1 \cap R_2 = C \neq \phi$ ✓

$(C)^+ = \{C, D\}$ → 'C' is not a S.K of either relation $\therefore$ Lossy

(iv) {R1(ABCD), R2(BE)} ✓

$R_1 \cup R_2 = R$ ✓

$R_1 \cap R_2 = B \neq \phi$ ✓

$(B)^+ = \{B, E\}$ — $\therefore$ B is a S.K of $R_2(B,E)$ ✓

All three Cond^n are satisfied $\therefore$ lossless

#Q. Let R (A, B, C, D, E, F) be the relational schema with following FD set

$F = \{ AB \rightarrow C, BC \rightarrow A, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, E \rightarrow F)$

Which of the following decomposition is lossless join decomposition.

(i){R1(ABC), R2(ABDE), R3(EF)}

(ii){R1(ABC), R2(ADF), R3(ACDE)}

(iii){R1(AB), R2(BC), R3(ABDE), R4(EF)}

2 mins Summary

Topic Lossless join decomposition

Topic Projection, Selection, and Cross product

Topic Natural join

Topic Lossless Natural Join

THANK - YOU

Slide