CS & IT ENGINEERING

Database Management System

DBMS

Lecture No. 7

By- Ravindrababu Ravula Sir

Recap of Previous Lecture

Topic RA

Topics to be Covered

Topic Minimal Cover

Topic Commands (DDL/DML/DCL),

Topic GROUP BY, HAVING

Topic Nested Queries

Topic: Canonical Cover Minimal Cover

Minimal Cover or Canonical Cover:

Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set.

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, A→C , B→C, A→A, AB → C}

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, A→C , B→C, A→A, AB → C}

Sol: Note that we have A→B and B→C By extension, we already get A→C. So, there is no need for A→C in FD set.

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, ~~A→C~~ , B→C, ~~A→A~~, AB → C}

Sol: Note that we have A→B and B→C By extension, we already get A→C. So, there is no need for A→C in FD set.

⇒ A→A being a Trivial Set/FD is not required.

NOTE:

A → A B → B C → C are all part of Trivial Set ∴ not required in FD set.

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, A→C , B→C, A→A, AB → C}

- Say, we have B→C.
  If we augment A both side, we get AB→AC. This can be dis-integrated into two parts:

AB→A
(Trivial FD)

AB→C
(Can be derived)

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, A→C , B→C, A→A, AB → C}

- Say, we have B→C.
  If we augment A both side, we get AB→AC. This can be dis-integrated into two parts:

AB→A	AB→C
(Trivial FD)	(Can be derived)

Since, AB→A is a Trivial FD, hence will not be in FD set.
AB → C can be derived from B→C, hence we will not put that into FD set.

Topic: Canonical Cover Minimal Cover

#Q. Finding Minimal Possible FD's (Functional Dependency) which are logically equivalent to given FD set. For R(ABC)
{A→B, A→C , B→C, A→A, AB → C}

minimal cover
canonical cover

From this we can say entire FD set is equivalent to {A→B,B→C}, because all other FD's can be derived from these two FD's.

Topic: Canonical Cover Minimal Cover

Just like this, we need to find minimal possible FD, which is equivalent to given FD set, this is called Minimal Cover/Canonical Cover.

- Say F is the Given FD Set

  Let Fm be the Minimal FD set corresponding F.

  Then Fm is the minimum possible FD's which are logically equal to F.

Topic: Canonical Cover Minimal Cover

Just like this, we need to find minimal possible FD, which is equivalent to given FD set, this is called Minimal Cover/Canonical Cover.

- Say F is the Given FD Set

  Let Fm be the Minimal FD set corresponding F.

  Then Fm is the minimum possible FD's which are logically equal to F.

  $Fm \cong F \rightarrow$ Logically equal

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Say we have {AB→C,A→B}, now, we can see that A is determining B from A→B

LHS

$AB \rightarrow C$

$A \rightarrow B$

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Say we have {AB→C,A→B}, now, we can see that A is determining B from A→B since, A is determining B, there is no need of B in Determinants.

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Say we have {AB→C,A→B}, now, we can see that A is determining B from A→B since, A is determining B, there is no need of B in Determinants.

NOTE:

X	→	Y
Determinants		Determined / Determiner

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Say we have {AB→C,A→B}, now, we can see that A is determining B from A→B since, A is determining B, there is no need of B in Determinants.

NOTE:

X	→	Y
Determinants		Determined / Determiner

So, for AB → C

→ B not required, because B is determined by A

⇒ {A→B,A→C}

Topic: Procedure to find Minimal Cover

Assume Given FD Set F

Before going to procedure, let's understand Extraneous Attributes first:

Say we have {AB→C,A→B}, now, we can see that A is determining B from A→B since, A is determining B, there is no need of B in Determinants.

NOTE:

X	→	Y
Determinants		Determined / Determiner

So, for AB → C

→ B not required, because B is determined by A

⇒ {A→B,A→C}

≡ {AB→C,A→B}

B is extraneous attribute.

Topic: Procedure to find Minimal Cover

Similarly,

$\{XYZ \rightarrow W, \quad X^+ = \ldots\ldots\ldots Y\}$

Topic: Procedure to find Minimal Cover

Similarly,

$\{XYZ \rightarrow W, \quad X^{+} = \ldots\ldots\ldots Y\}$

means Closure of X determines Y

Topic: Procedure to find Minimal Cover

Similarly,

$\{XYZ \rightarrow W, \quad X^+ = \ldots\ldots\ldots Y\}$

means Closure of X determines Y

So, there is no need for Y in $XYZ \rightarrow W$ since Y is determined by X

Topic: Procedure to find Minimal Cover

Similarly,

$\{XYZ \rightarrow W, \quad X^+ = \ldots\ldots\ldots Y\}$

means Closure of X determines Y

So, there is no need for Y in $XYZ \rightarrow W$ since Y is determined by X

$\Rightarrow \quad \{XZ \rightarrow W, \qquad X^+ \rightarrow \ldots Y\}$

Y is extraneous Attribute.

Topic: Procedure to find Minimal Cover

Canonical co

STEP 1.Remove extraneous attributes from each determinant of Given FD Set.

{Determinant = LHS of FD set}

LHS

Canonical cover

1)

2)

Topic: Procedure to find Minimal Cover

STEP 1.Remove extraneous attributes from each determinant of Given FD Set.

{Determinant = LHS of FD set}

Extraneous Attribute:

If $XYZ \rightarrow W$, $X^+ = \{\ldots Y \ldots\}$ in FD SET (F)

{X closure}

Then Y extraneous from $XYZ \rightarrow W$

$\{XYZ \rightarrow W, X \rightarrow Y\} \equiv \{XZ \rightarrow W, X \rightarrow Y\}$

extraneous

$\Rightarrow$ Because X closure determines the Y, then Y extraneous from $\{XYZ \rightarrow W\}$ can be removed from the determinant.

$\therefore \quad XYZ \rightarrow W \equiv XZ \rightarrow W$

Topic: Procedure to find Minimal Cover

- If more than one attribute is in the determinant side , then we remove the extraneous attribute.

Topic: Procedure to find Minimal Cover

- If more than one attribute is in the determinant side , then we remove the extraneous attribute.

Example:

What is extraneous attribute of {ABCD→E,BC→F,F→A}?

Topic: Procedure to find Minimal Cover

- If more than one attribute is in the determinant side , then we remove the extraneous attribute.

Example:

What is extraneous attribute of {ABCD→E,BC→F,F→A}?

Now,

Becoz of BC→F & F→A

⇒ BC→A

Topic: Procedure to find Minimal Cover

- If more than one attribute is in the determinant side , then we remove the extraneous attribute.

Example:

What is extraneous attribute of {ABCD→E,BC→F,F→A}?

Now,

Becoz of BC→F & F→A

⇒ BC→A

Now, because BC Closure determining A

∴ A is extraneous.

⇒ ~~A~~B C D→E

Topic: Procedure to find Minimal Cover

- If more than one attribute is in the determinant side , then we remove the extraneous attribute.

Example:

What is extraneous attribute of {ABCD→E,BC→F,F→A}?

Now,

Becoz of BC→F & F→A

⇒ BC→A

Now, because BC Closure determining A

∴ A is extraneous.

⇒ ~~A~~B C D→E

≡{BCD→E,BC→F,F→A}

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

1) Extra remov

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

What are Redundant FD's?

Say ...X→Y...

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

What are Redundant FD's?

Say ...X→Y...

- When we say X→Y is redundant, If we remove X→Y from FD set and find X+, X+ must contain Y

$X \rightarrow Y$

$X^+ = \{ \ldots Y \ldots \}$

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

What are Redundant FD's?

Say ...X→Y...

- When we say X→Y is redundant, If we remove X→Y from FD set and find X+, X+ must contain Y

Ex: {A→B, B→C, A→C}

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

What are Redundant FD's?

Say ...X→Y...

- When we say X→Y is redundant, If we remove X→Y from FD set and find X+, X+ must contain Y

LHS → RHS

$A^+ = ABC$

Ex: {A→B, B→C, A→C}

here A→C is redundant, because we can anyway derive A→C using A→B and B→C

Topic: Procedure to find Minimal Cover

STEP 2.Remove Redundant FD's from result of Step 1.

What are Redundant FD's?

Say ...X→Y...

- When we say X→Y is redundant, If we remove X→Y from FD set and find X+, X+ must contain Y

Ex: {A→B, B→C, A→C}

here A→C is redundant, because we can anyway derive A→C using A→B and B→C

⇒ A FD that can be derived by using the other FD's, such FD's are referred as Redundant FD's.

Topic: Procedure to find Minimal Cover

Ex: {AB→C,AB→D,BC→D}

- AB→C must exist because AB→C cannot be derived through any other FD's.

Topic: Procedure to find Minimal Cover

Ex: {AB→C, AB→D, BC→D}

- AB→C must exist because AB→C cannot be derived through any other FD's.
- If we remove AB→D, what is AB closure from other FD's?

  $(AB)^+ = \{ABCD\}$

$AB^+ = ABCD$

Topic: Procedure to find Minimal Cover

Ex: {AB→C,AB→D,BC→D}

- AB→C must exist because AB→C cannot be derived through any other FD's.
- If we remove AB→D, what is AB closure from other FD's?

  $(AB)^+ = \{ABCD\}$

  ∴ we can derive AB→D even if we remove AB→D, so AB→D is redundant FD.

  So if we remove Redundant FD, the result is Minimal Cover.

Inspiring Stories : Mahabir Pun

Background: **From remote mountain village in Nepal.**

Struggle: **Villages had no internet or even basic facilities.**

Achievements: **Built hydro-powered wifi network to connect remote schools.**

Impact: **Enabled education, jobs, and healthcare in isolated villages.**

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

Gate
↓
simple

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

Sol:

Step-1: Remove Extraneous Attributes

For extraneous, we won't bother right side.

For A→BCDEF here in LHS, one attribute only

⇒ Not extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

Sol:

Step-1: Remove Extraneous Attributes

For extraneous, we won't bother right side.

For A→BCDEF here in LHS, one attribute only

⇒ Not extraneous.

For BC→ADEF

1. If B can determine C, then C extraneous
2. If C can determine B, Then B extraneous

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

1. But B closure is B+ ={B,F} ∉ C doesn't include C. so, C is not extraneous

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

1. But B closure is B+ ={B,F} ∉ C doesn't include C. so, C is not extraneous
2. Also C^+ ={C} ∉B ⇒ doesn't include B then B is not extraneous.

   So, in this case, neither B determines C, nor C determines B.

   Hence, we have no extraneous in BC→ADEF

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

1. But B closure is B+ ={B,F} ∉ C doesn't include C. so, C is not extraneous
2. Also C^+ ={C} ∉B ⇒ doesn't include B then B is not extraneous.

   So, in this case, neither B determines C, nor C determines B.

   Hence, we have no extraneous in BC→ADEF

- Again B→F and D→E have only one attribute in LHS ⇒ neither have extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

- Let's check AE→D

  A closure: A+={ABCDEF}

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

- Let's check AE→D

  A closure: A+={ABCDEF}

⇒ A determines E

  Now since A^+→...E

⇒ E is extraneous ~~AE~~→D

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

- Let's check AE→D

  A closure: A+={ABCDEF}

⇒ A determines E

  Now since A^{+}→...E

⇒ E is extraneous AE→D

After removing extraneous elements { A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover.

{ A→BCDEF, BC→ADEF, B→F, D→E, AE→D}

Step 2: Remove Redundant FD's

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover after STEP 1

{ $A \rightarrow BCDEF$, $BC \rightarrow ADEF$, $B \rightarrow F$, $D \rightarrow E$, $A \rightarrow D$}

Step 2: Remove Redundant FD's

to remove Redundant FD, we will need to split the FD.

SPLIT Rule: say $X \rightarrow YZ \Rightarrow X \rightarrow Y, X \rightarrow Z$

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Why do we need to split?

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Why do we need to split?

A→BCDEF is not a single FD, individually they have 5 FD's, and out of these 5 FD's, any one can be redundant.

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Why do we need to split?

A→BCDEF is not a single FD, individually they have 5 FD's, and out of these 5 FD's, any one can be redundant.

A→BCDEF⇒ A→B A→C
A→D A→E
A→F

Any can be redundant

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Why do we need to split?

A→BCDEF is not a single FD, individually they have 5 FD's, and out of these 5 FD's, any one can be redundant.

A→BCDEF⇒ A→B A→C
A→D A→E
A→F

Any can be redundant

BC→ADEF⇒ BC→A BC→D
BC→E BC→F

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

A→BCDEF⇒ A→B A→C
A→D A→E
A→F

BC→ADEF⇒ BC→A BC→D
BC→E BC→F

B→ F ⇒ {B → F}

D→ E ⇒ {D → E}

A→ D ⇒ {A → D}

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

A→BCDEF⇒ A→B A→C

A→D A→E

A→F

BC→ADEF⇒ BC→A BC→D

BC→E BC→F

B→ F ⇒ {B → F}

D→ E ⇒ {D → E}

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

A→BCDEF⇒ A→B A→C
A→D A→E
A→F

BC→ADEF⇒ BC→A BC→D
BC→E BC→F

B→ F ⇒ {B → F}

D→ E ⇒ {D → E}

Remembering Redundant FD's:

X→Y will be Redundant FD, if even if we remove X→Y, still X closure should determine Y.

So, A→B is not redundant, because there is no other way to determine B; There is no other determiner for B.

If we try to remove A→B, we won't be able to determine B from A.

Topic: Procedure to find Minimal Cover

$A \rightarrow BCDEF \Rightarrow$ $A \rightarrow B$ $A \rightarrow C$
$A \rightarrow D$ $A \rightarrow E$
$A \rightarrow F$

$BC \rightarrow ADEF \Rightarrow$ $BC \rightarrow A$ $BC \rightarrow D$
$BC \rightarrow E$ $BC \rightarrow F$

$B \rightarrow F \Rightarrow \{B \rightarrow F\}$

$D \rightarrow E \Rightarrow \{D \rightarrow E\}$

#Q. FD's after STEP 1

$\{ A \rightarrow BCDEF,\ BC \rightarrow ADEF,\ B \rightarrow F,\ D \rightarrow E,\ A \rightarrow D\}$

So, FD whose determiner is not repeated in the set; such FD is not a redundant FD.

Similarly, $A \rightarrow C$ is not redundant, because no other way to get C from A.

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

So, FD whose determiner is not repeated in the set; such FD is not a redundant FD.

Similarly, A→C is not redundant, because no other way to get C from A.

⇒ {

A→B, BC→A,

A→C, BC→D,

A→D, BC→E,

A→E, BC→F,

A→F, B→F, D→E

}

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

So, FD whose determiner is not repeated in the set; such FD is not a redundant FD.

Similarly, A→C is not redundant, because no other way to get C from A.

⇒ {

A→B, BC→A,

A→C, BC→D

A→D, BC→E,

A→E, BC→F,

A→F, B→F D→E

}

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→D: may or may not redundant (A→D)

  Let's remove A→D and check if A closure still gives D.

  If we remove A→D; A^+ ={ABCEFD} D is included

A→B, BC→A,

A→C, BC→D

A→D, BC→E,

A→E, BC→F,

A→F, B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→D: may or may not redundant (A→D)

  Let's remove A→D and check if A closure still gives D.

  If we remove A→D; A^+ ={ABCEFD} D is included

  ∴ A^+ ={…D…}→A→D is redundant.

A→B,	BC→A,	
A→C,	BC→D	
A→D,	BC→E,	
A→E,	BC→F,	
A→F,	B→F	D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→D: may or may not redundant (A→D)

  Let's remove A→D and check if A closure still gives D.

  If we remove A→D; A^+ ={ABCEFD} D is included

  ∴ A^+ ={…D…}→A→D is redundant.

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

A→E, BC→F,

A→F, B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→D: may or may not redundant (A→D)

  Let's remove A→D and check if A closure still gives D.

  If we remove A→D; A^+ ={ABCEFD} D is included

  ∴ A^+ ={…D…}→A→D is redundant.

A→B,	BC→A,	
A→C,	BC→D	
~~A→D,~~	BC→E,	
A→E,	BC→F,	
A→F,	B→F	D→E

Important Note:

whenever we are doing closure, don't consider the ones that have already being removed; A→D in this case. A→D is already removed.

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→E:

  Let's remove A→E

  A^+ ={ABCEFD}

⇒A→E is redundant.

A→B,	BC→A,	
A→C,	BC→D	
~~A→D,~~	BC→E,	
A→E,	BC→F,	
A→F,	B→F	D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For A→E:

  Let's remove A→E

  A^+ ={ABCEFD}

⇒A→E is redundant.

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

A→F, B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

For A→F:

Let's remove A→F; A^+ ={ABCDEF}

Even if we remove FD, still F can be determined by A

∴ A→F is redundant.

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

A→F, B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

For A→F:

Let's remove A→F; A^+ ={ABCDEF}

Even if we remove FD, still F can be determined by A

∴ A→F is redundant.

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→D:

Let's remove BC→D BC^+ =B C A E F

A→B, BC→A,

A→C, ~~BC→D~~

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→D:

  Let's remove BC→D BC^+ =B C A E F

⇒ BC→D is NOT redundant. ✓

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→E

  Let's remove BC→E BC^+ =BCADEF

A→B, BC→A,

A→C, BC→D

~~A→D,~~ BC→E,

~~A→E,~~ BC→F,

~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→E

Let's remove BC→E BC^+ =BCADEF

NOTE that even if we remove BC→D, we can STILL determine E from BC

⇒ BC→E is redundant.

A→B, BC→A,
A→C, BC→D
~~A→D,~~ BC→E,
~~A→E,~~ BC→F,
~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→E

  Let's remove BC→E BC^+ =BCADEF

  NOTE that even if we remove BC→D, we can STILL determine E from BC

⇒ BC→E is redundant.

A→B, BC→A,

A→C, BC→D

~~A→D,~~ ~~BC→E,~~

~~A→E,~~ BC→F,

~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→F

Let's remove BC→F BC^+ = B C A D E F

A→B, BC→A,
A→C, BC→D
~~A→D,~~ ~~BC→E,~~
~~A→E,~~ BC→F,
~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→F

Let's remove BC→F BC^+ = B C A D E F

again, even when we have removed BC→F, we can still determine BC→F using BC^+

⇒ BC→F is redundant.

A→B, BC→A,
A→C, BC→D
~~A→D,~~ ~~BC→E,~~
~~A→E,~~ BC→F,
~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For BC→F

Let's remove BC→F BC^+ = B C A D E F

again, even when we have removed BC→F, we can still determine BC→F using BC^+

⇒ BC→F is redundant.

A→B, BC→A,
A→C, BC→D
~~A→D,~~ ~~BC→E,~~
~~A→E,~~ ~~BC→F,~~
~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

- For B→F

Let's remove B→F then B^+ will only be {B} ⇒ B→F is NOT redundant.

A→B, BC→A,
A→C, BC→D
~~A→D,~~ ~~BC→E,~~
~~A→E,~~ ~~BC→F,~~
~~A→F,~~ B→F ✗ D→E

Topic: Procedure to find Minimal Cover

#Q. FD's after STEP 1

{ A→BCDEF, BC→ADEF, B→F, D→E, A→D}

Similarly for D→E Let's remove D→E then D + will only be {D}

⇒ D→E is NOT redundant.

A→B, BC→A,
A→C, BC→D
~~A→D,~~ ~~BC→E,~~
~~A→E,~~ ~~BC→F,~~
~~A→F,~~ B→F D→E

Topic: Procedure to find Minimal Cover

$A \rightarrow B$, $BC \rightarrow A$,

$A \rightarrow C$, $BC \rightarrow D$

~~$A \rightarrow D$,~~ ~~$BC \rightarrow E$,~~

~~$A \rightarrow E$,~~ ~~$BC \rightarrow F$,~~

~~$A \rightarrow F$,~~ $B \rightarrow F$ $D \rightarrow E$

To conclude, we have:

1. Removed extraneous attributes
2. Removed Redundant FD's.

The resultant FD will be called Minimal Cover

- **Minimal Cover:** $\{A \rightarrow BC, BC \rightarrow AD, B \rightarrow F, D \rightarrow E\}$

Topic: Procedure to find Minimal Cover

NOTE:

Minimal cover may not be unique.

Topic: Procedure to find Minimal Cover

JNU PHD EE

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH CD→E ABH→BD E→C DH→BC }

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH CD→E ABH→BD E→C DH→BC }

Sol:

1. Remove extraneous attributes:

For A→BC:

one attribute in LHS ⇒ NO chance for extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH CD→E ABH→BD E→C DH→BC }

Sol:

For CD→E:

D ×

C+ = C

therefore, C can't determine D

But for D, D^+ =DAEHBC

⇒ D closure can determine C

⇒ C is extraneous, CD → E implies D → E

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E ABH→BD E→C DH→BC }

For E→C and D→AEH, both have one attribute on LHS

⇒ NO chance of being extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E ABH→BD E→C DH→BC }

- For ABH→BD

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E ABH→BD E→C DH→BC }

- For ABH→BD, A+ = ABC

  Now, note that A itself determines B.

  So, we can say B is extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E ABH→BD E→C DH→BC }

- For ABH→BD A+ = ABC

  Now, note that A itself determines B.

  So, we can say B is extraneous.

⇒ A ~~B~~ H→BD

after removing B, A, H are left.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E A~~B~~H→BD E→C DH→BC }

- For ABH→BD A+ = ABC

  Now, note that A itself determines B.

  So, we can say B is extraneous.

⇒ A~~B~~H→BD

after removing B, A, H are left.

A^+ =ABC but H is not determined by A closure.

⇒ H is NOT extraneous.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E A~~B~~H→BD E→C DH→BC }

- What about H closure? because if H is able to determine A, then A would be extraneous.

  We have to check every possibility.

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH ~~C~~D→E A~~B~~H→BD E→C DH→BC }

- What about H closure? because if H is able to determine A, then A would be extraneous.

  We have to check every possibility. $H^+ = H$

  H cannot determine A

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ $A \rightarrow BC$ $D \rightarrow AEH$ $\cancel{C}D \rightarrow E$ $A\cancel{B}H \rightarrow BD$ $E \rightarrow C$ $DH \rightarrow BC$ }

For $DH \rightarrow BC$ $D^+ = DAEHBC$ where, D determines the H

$\Rightarrow$ H is extraneous

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ $A \rightarrow BC$ $D \rightarrow AEH$ $\cancel{C}D \rightarrow E$ $A\cancel{B}H \rightarrow BD$ $E \rightarrow C$ $D\cancel{H} \rightarrow BC$ }

For $DH \rightarrow BC$ $D^+ = DAEHBC$ where, D determines the H

$\Rightarrow$ H is extraneous

Topic: Procedure to find Minimal Cover

#Q. Find Minimal Cover of this FD Set

{ A→BC D→AEH CD→E ABH→BD E→C DH→BC }

After STEP 1:

{ A→BC D→AEH ~~C~~D→E A~~B~~H→BD E→C D~~H~~→BC }

{ A→BC D→AEH C→E AH→BD E→C D→BC }

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

2. Remove Redundant FD's.

To remove redundant FD, always better to split the FD's so that every determiner is one attribute.

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

2. Remove Redundant FD's.

To remove redundant FD, always better to split the FD's so that every determiner is one attribute.

A→BC

D→AEH

C→E

AH→BD

E→C

D→BC

{ A→B AH→D
A→C E→C
D→A D→B
D→E D→C
D→H
C→E
AH→B }

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

2. Remove Redundant FD's.

To remove redundant FD, always better to split the FD's so that every determiner is one attribute.

A→BC

D→AEH

C→E

AH→BD

E→C

D→BC

{ A→B, A→C, D→A, D→E, D→H, C→E, AH→B, AH→D, E→C, D→B, D→C }

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
D→E	D→C
D→H	
C→E	
AH→B	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For A→B, Remove A →B and check for A+ = ACE

So, it is not redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
D→E	D→C
D→H	
C→E	
AH→B	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For A→C, Remove A →C and check for A+ = AB

So, it is not redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
D→E	D→C
D→H	
C→E	
AH→B	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→E, Remove D →E and check for D+ = DAHBCE

So, it is redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
~~D→E~~	D→C
D→H	
C→E	
AH→B	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→E, Remove D →E and check for D+ = DAHBCE

So, it is redundant

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For C→E , Remove C →E and check for C+ = C

So, it is not redundant

A→B AH→D
A→C E→C
D→A D→B
~~D→E~~ D→C
D→H
C→E
AH→B

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
~~D→E~~	D→C
D→H	
C→E	
AH→B	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For AH→B , Remove AH →B and check for AH+ = AHDBCE

So, it is redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
~~D→E~~	D→C
D→H	
C→E	
~~AH→B~~	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For AH→B , Remove AH →B and check for AH+ = AHDBCE

So, it is redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
~~D→E~~	D→C
D→H	
C→E	
~~AH→B~~	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For E→C , Remove E →C and check for E+ = E

So, it is not redundant

Topic: Procedure to find Minimal Cover

A→B	AH→D
A→C	E→C
D→A	D→B
~~D→E~~	D→C
D→H	
C→E	
~~AH→B~~	

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→B, Remove D →B and check for D+ = DCEHAB

So, it is redundant

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→B, Remove D →B and check for D+ = DCEHAB

So, it is redundant

A→B AH→D
A→C E→C
D→A ~~D→B~~
~~D→E~~ D→C
D→H
C→E
~~AH→B~~

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→C, Remove D →C and check for D+ = DAHBCE

So, it is redundant

A→B AH→D
A→C E→C
D→A ~~D→B~~
~~D→E~~ D→C
D→H
C→E
~~AH→B~~

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

For D→C, Remove D →C and check for D+ = DAHBCE

So, it is redundant

A→B AH→D

A→C E→C

D→A ~~D→B~~

~~D→E~~ ~~D→C~~

D→H

C→E

~~AH→B~~

Topic: Procedure to find Minimal Cover

#Q. After Step1

{A→BC D→AEH C→E AH→BD E→C D→BC}

A→B AH→D
A→C E→C
D→A ~~D→B~~
~~D→E~~ ~~D→C~~
D→H
C→E
~~AH→B~~

minimal

After STEP2:

the Minimal Cover is {A →BC, D →AH, C →E, AH →D, E →C}

Topic: Structured Query Language

SQL (2 marks)

- Data Definition Language (DDL)
- Used to define structure/ definition of DB Table

- Data Manipulation Language (DML)
- Used to modify data records & access required from DB tables

- Data Control Language (DCL)

Algo
DS
SQL

SQL → DS, SD → SQL

Topic: Structured Query Language

SQL

- Data Definition Language (DDL)
- Used to define structure/ definition of DB Table
  - CREATE table
  - DROP table
  - ALTER table (Add/remove attribute)
  - PRIMARY KEY
  - UNIQUE
  - FOREIGN KEY (FK Constraints)
  - NOT NULL
  - CHECK etc.(Used to set range for attribute value)

- Data Manipulation Language (DML)
- Used to modify data records & access required from DB tables

- Data Control Language (DCL)

ER model

SQL

↓

ER model

Topic: Structured Query Language

SQL

- Data Definition Language (DDL)
- Used to define structure/ definition of DB Table
  - CREATE table
  - DROP table
  - ALTER table (Add/remove attribute)
  - PRIMARY KEY
  - UNIQUE
  - FOREIGN KEY (FK Constraints)
  - NOT NULL
  - CHECK etc.(Used to set range for attribute value)

- Data Manipulation Language (DML)
- Used to modify data records & access required from DB tables
  - INSERT into
  - DELETE into
  - UPDATE... SET
  - SELECT * FROM TABLES WHERE Cond;

- Data Control Language (DCL)

Topic: Structured Query Language

SQL

- Data Definition Language (DDL)
- Used to define structure/ definition of DB Table
  - CREATE table
  - DROP table
  - ALTER table (Add/remove attribute)
  - PRIMARY KEY
  - UNIQUE
  - FOREIGN KEY (FK Constraints)
  - NOT NULL
  - CHECK etc.(Used to set range for attribute value)

- Data Manipulation Language (DML)
- Used to modify data records & access required from DB tables
  - INSERT into
  - DELETE into
  - UPDATE... SET
  - SELECT * FROM TABLES WHERE Cond;

- Data Control Language (DCL)

1. Data Control for transaction management [To avoid inconsistencies because of concurrent usage]
   - ROLLBACK
   - COMMIT
   - CHECKPOINT etc,
2. Data Control for user authentication (Security)
   - GRANT ACCESS
   - REVOKE ACCESS etc,

Topic: SQL Vs RA Query

SQL Query Vs RA Query:

SQL:	SELECT DISTINCT A1, A2, .. An	⇒ PROJECTION(π)
	FROM R1, R2, .. Rm	⇒ CROSS PRODUCT(×)
	WHERE p	⇒ SELECTION

Topic: SQL Vs RA Query

SQL Query Vs RA Query:

SQL:	SELECT DISTINCT A1, A2, .. An	⇒ PROJECTION(π)
1.	FROM R1, R2, .. Rm	⇒ CROSS PRODUCT(×)
	WHERE p	⇒ SELECTION

$$\sigma\,(R1 \times R2 \times R3 .. Rm)$$

Topic: SQL Vs RA Query

SQL Query Vs RA Query:

SQL: SELECT DISTINCT A1, A2, .. An ⇒ PROJECTION(π)

1. FROM R1, R2, .. Rm ⇒ CROSS PRODUCT(×)
2. WHERE p ⇒ SELECTION

$$\sigma_p(R1 \times R2 \times R3..Rm)$$

Topic: SQL Vs RA Query

SQL Query Vs RA Query:

SQL:

3. SELECT DISTINCT A1, A2, .. An ⇒ PROJECTION(π)
1. FROM R1, R2, .. Rm ⇒ CROSS PRODUCT(×)
2. WHERE p ⇒ SELECTION

RA: $\pi_{A1,A2,..An}(\sigma_p(R1 \times R2 \times R3..Rm))$

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Sol:

$$\pi_{sid}(\sigma_{Enroll.Cid=Course.Cid}(Enroll \times \sigma_{Instr=KORTH}(Course)))$$

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Sol: By default distinct records

$$\pi_{sid}(\sigma_{Enroll.Cid=Course.Cid}(Enroll \times \sigma_{Instr=KORTH}(Course)))$$

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

×

12

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Sol: By default distinct records

$\pi_{sid}(\sigma_{Enroll.Cid=Course.Cid}(Enroll \times \sigma_{Instr=KORTH}(Course)))$

FROM Enroll, course WHERE Inst = KORTH and enroll cid = course cid;

Sid	Cid	Cid	Inst.
S1	C1	C1	Korth
S1	C2	C2	Korth
S2	C2	C2	Korth

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Sol: By default distinct records

$$\pi_{sid}(\sigma_{Enroll.Cid=Course.Cid}(Enroll \times \sigma_{Instr=KORTH}(Course)))$$

FROM Enroll, course WHERE Inst = KORTH and enroll cid = course cid;

We have to use distinct Explicitly in SQL to get UNIQUE records

Sid	Cid	Cid	Inst.
S1	C1	C1	Korth
S1	C2	C2	Korth
S2	C2	C2	Korth

Topic: SQL Vs RA Query

Enroll

Sid	Cid
S1	C1
S1	C2
S2	C2
S2	C3

Course = Korth

Cid	Inster.
C1	Korth
C2	Korth
C3	Navathe

Retrieve Sid's enrolled some course taught by Korth

Sol: By default distinct records

$\pi_{sid}(\sigma_{Enroll.Cid=Course.Cid}(Enroll \times \sigma_{Instr=KORTH}(Course)))$

FROM Enroll, course WHERE Inst = KORTH and enroll cid = course cid;

SELECT DISTINCT Sid From Enroll, Course WHERE Inst. =KORTH and Enroll.cid = Course.cid

We have to use distinct

Explicitly in SQL to get

UNIQUE records

Sid	Cid	Cid	Inst.
S1	C1	C1	Korth
S1	C2	C2	Korth
S2	C2	C2	Korth

Inspiring Stories : Radio Women

Background: **Dalit women in rural Telangana, no media training.**

Struggle: **Wanted to share their issues, but had no voice.**

Achievements: **Started India's first all-women community radio.**

Impact: **Gives farming tips, rights news, culture to many villages.**

Topic: SQL Vs RA Query

BASIC SQL CLAUSES:

SELECT [DISTINCT] A1, A2,.....An

FROM R1, R2,..... Rm

[WHERE condition]

[GROUP BY (Attributes) [Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAUSES:

SELECT (DISTINCT) A1, A2,.....An

1. FROM R1, R2,..... Rm

[WHERE condition]

[GROUP BY (Attributes) [Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAUSES:

SELECT (DISTINCT) A1, A2,.....An

1. FROM R1, R2,..... Rm
2. [WHERE condition]

[GROUP BY (Attributes) [Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAUSES:

SELECT (DISTINCT) A1, A2,.....An

1. FROM R1, R2,..... Rm
2. [WHERE condition]
3. [GROUP BY (Attributes) [Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAVSES:

SELECT (DISTINCT) A1, A2,.....An

1. FROM R1, R2,..... Rm
2. [WHERE condition]
3. [GROUP BY (Attributes) 4.[Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAVSES:

5. SELECT (DISTINCT) A1, A2,.....An
1. FROM R1, R2,..... Rm
2. [WHERE condition]
3. [GROUP BY (Attributes) 4.[Having condition]

   [ORDER BY (Attributes) [DESC];

   [ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAVSES:

5. SELECT 6. (DISTINCT) A1, A2,.....An
1. FROM R1, R2,..... Rm
2. [WHERE condition]
3. [GROUP BY (Attributes) 4.[Having condition]

   [ORDER BY (Attributes) [DESC];

   [ ] is optional.

Topic: SQL Vs RA Query

BASIC SQL CLAVSES:

5. SELECT 6.(DISTINCT) A1, A2,.....An
1. FROM R1, R2,..... Rm
2. [WHERE condition]
3. [GROUP BY (Attributes) 4.[Having condition]
7. [ORDER BY (Attributes) [DESC];

[ ] is optional.

Execution flow of SQL Query

FROM
↓
WHERE
↓
GROUP BY
↓
HAVING
↓
SELECT
↓
DISTINCT
↓
ORDER BY

Topic: SQL Vs RA Query

FROM → CROSS PRODUCT (×)

↓

WHERE → SELECTION OF (σ)

↓

GROUP BY ✗

↓

HAVING ✗

↓

SELECT

↓

DISTINCT

} PROJECTION OF (π)

↓

ORDER BY ✗

Topic: SQL Vs RA Query

GROUP BY:

- Used to group records based on specific attribute values.

Topic: SQL Vs RA Query

GROUP BY:

- Used to group records based on specific attribute values.

Ex-R(x, y, z)

```
SELECT X AVG(Y), Z
FROM R
GROUP BY X
```

If GROUP BY clause is used then

(i) All Attributes of GROUP BY function clause must be in select clause.

Topic: SQL Vs RA Query

GROUP BY:

- Used to group records based on specific attribute values.

Ex-R(x, y, z)

```
SELECT X, AVG(Y), Z
FROM R
GROUP BY X
```

If GROUP BY clause is used then

(i) All Attributes of GROUP BY function clause must be in select clause.

(ii) can use AGGREGRATE function is SELECT clause.

Topic: SQL Vs RA Query

GROUP BY:

- Used to group records based on specific attribute values.

Ex-R(x, y, z)

```
SELECT X, AVG(Y), Z
FROM R
GROUP BY X
```

If GROUP BY clause is used then

(i) All Attributes of GROUP BY function clause must be in select clause.

(ii) can use AGGREGRATE function is SELECT clause.

(iii) Not allowed to select any other attribute in select clause (other than Attribute in GROUP BY or with aggregate function)

Topic: SQL Vs RA Query

Aggregate function supported by SQL:

- Count ( )
- Sum ( )
- Avg. ( )
- min. ( )
- max ( )

Topic: SQL Vs RA Query

Aggregate function supported by SQL:

Count ( )
Sum ( )
Avg. ( )
min. ( )
max ( )

} Computes aggregation on Non-Null values

Topic: SQL Vs RA Query

Maintained by SQL

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Create table
(A integer,
B integer);
SQL add Row ID by default.

Topic: SQL Vs RA Query

1. **Count:**

   Gives the count of all NOT NULL values.

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

1. Count:

Gives the count of all NOT NULL values.

- Count(*) - No. of records

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

1. **Count:**

   Gives the count of all NOT NULL values.

   - Count(*) - No. of records
   - Count(A) - No. of Non Null Values of Attr A

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

1. Count:

Gives the count of all NOT NULL values.

- Count(*) - No. of records
- Count(A) - No. of Non Null Values of Attr A
- Count(DISTINCT A) - No. of Distinct Non Null Values of Attr A

Topic: SQL Vs RA Query

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

1. Count:

Gives the count of all NOT NULL values.

- Count(*) - No. of records
- Count(A) - No. of Non Null Values of Attr A
- Count(DISTINCT A) - No. of Distinct Non Null Values of Attr A
- We also use renaming, because Count(*), Count(A) doesn't define correct column names.

Topic: SQL Vs RA Query

Ex: Select Count(*) A1,
Count(A) A2,
Count(DISTINCT A) A3
FROM R;

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

Ex: Select Count(*) A1,
Count(A) A2,
Count(DISTINCT A) A3
FROM R;

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

O/P

Count (*)	Count (A)	Count (DISTINCT A)
7	5	4

Topic: SQL Vs RA Query

Ex: Select Count(*) A1,
Count(A) A2,
Count(DISTINCT A) A3
FROM R;

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

O/P

Count (*)	Count (A)	Count (DISTINCT A)
7	5	4

O/P

A1	A2	A3
7	5	4

Topic: SQL Vs RA Query

- Same definitions for SUM, AVG, MIN, MAX.

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

- Same definitions for SUM, AVG, MIN, MAX.

  Ex: SELECT Sum(A) A1, Sum(DISTINCT A) A2, AVG(A) A3, AVG(DISTINCT A) A4, MIN(A) A5, MAX(A) A6.

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

Topic: SQL Vs RA Query

- Same definitions for SUM, AVG, MIN, MAX.

  Ex: SELECT Sum(A) A1, Sum(DISTINCT A) A2, AVG(A) A3, AVG(DISTINCT A) A4, MIN(A) A5, MAX(A) A6.

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	~~20~~	20
5	60	25
6	80	30
7	NULL	NULL

$$AVG(A) = \frac{SUM(A)}{COUNT(A)} = \frac{220}{5} = 44$$

$$AVG(\cancel{A})\ (DISTINCT\ A) = \frac{SUM(DISTINCT\ A)}{COUNT(DISTINCT\ A)} = \frac{200}{4} = 50$$

Topic: SQL Vs RA Query

R

Row ID	A	B
1	20	5
2	40	10
3	NULL	15
4	20	20
5	60	25
6	80	30
7	NULL	NULL

- Same definitions for SUM, AVG, MIN, MAX.

  Ex: SELECT Sum(A) A1, Sum(DISTINCT A) A2, AVG(A) A3, AVG(DISTINCT A) A4, MIN(A) A5, MAX(A) A6.

$$\text{AVG(A)} = \frac{\text{SUM(A)}}{\text{COUNT(A)}} = \frac{220}{5} = 44$$

$$\text{AVG(A) (DISTINCT A)} = \frac{\text{SUM(DISTINCT A)}}{\text{COUNT(DISTINCT A)}} = \frac{200}{4} = 50$$

O/P

A1	A2	A3	A4	A5	A6
220	200	44	50	20	80

Topic: SQL Vs RA Query

Ex- R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

GROUP BY (A)

Topic: SQL Vs RA Query

Ex- R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

GROUP BY (A)

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1 Select A
From R
Group by A;

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1 Select A
From R
Group by A;

A
a1
Null
a2

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1 Select A
From R
Group by A;

A
a1
Null
a2

Ex-2

3. Select A, AVG (C)
1. FROM R
2. GROUP BY A ;

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1 Select A
From R
Group by A;

A
a1
Null
a2

Ex-2

3. Select A, AVG (C)
1. FROM R
2. GROUP BY A ;

A	AVG(C)
A1	85
Null	50
a2	75

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1

Select A
From R
Group by A;

A
a1
Null
a2

Ex-2

3. Select A, AVG (C)
1. FROM R
2. GROUP BY A ;

A	AVG(C)
A1	85
Null	50
a2	75

Ex-3

Select A, B
From R
GROUP BY A;

Select B
From R
GROUP BY A;

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-1

Select A
From R
Group by A;

A
a1
Null
a2

Ex-2

3. Select A, AVG (C)
1. FROM R
2. GROUP BY A ;

A	AVG(C)
A1	85
Null	50
a2	75

Ex-3

Select A, B
From R
GROUP BY A;

Invalid

Select B
From R
GROUP BY A;

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Inspiring Stories : Jaya & Atul Shah

Background: **Couple who retired in Uttarakhand.**

Struggle: **Kids had no access to books or learning.**

Achievements: **Started a library in their home, now kids across villages come daily.**

Impact: **Sparked a reading revolution among hill village children.**

Topic: SQL Vs RA Query

HAVING CLAUSE:

- use to select groups based on condition

Topic: SQL Vs RA Query

HAVING CLAUSE:

- use to select groups based on condition

Ex-1

```
Select A
1. From R
2. GROUP BY A
3. HAVING AVG. (C) >60;
```

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

HAVING CLAUSE:

- use to select groups based on condition

R

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Ex-1

Select A
1. From R
2. GROUP BY A
3. HAVING AVG. (C) >60;

A
a1
a2

A	AVG(C)
a1	85
Null	50
a2	75

Topic: SQL Vs RA Query

Ex-2 Select A

1. From R
2. GROUP BY A
3. HAVING AVG (C) >80;

A	AVG(C)
a1	**85**
Null	50
a2	75

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-2 Select A

1. From R
2. GROUP BY A
3. HAVING AVG (C) >80;

A
a1

A	AVG(C)
a1	**85**
Null	50
a2	75

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-2 Select A

1. From R
2. GROUP BY A
3. HAVING AVG (C) >80;

A
a1

A	AVG(C)
a1	**85**
Null	50
a2	75

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

Ex-3 Select A

1. From R
2. GROUP BY A
3. HAVING C >60;

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Topic: SQL Vs RA Query

Ex-2 Select A

1. From R
2. GROUP BY A
3. HAVING AVG (C) >80;

A
a1

A	AVG(C)
a1	**85**
Null	50
a2	75

A	B	C
a1	b1	80
a2	b4	90
Null	b3	60
a2	b4	60
a1	b1	90
Null	b3	40
a1	a2	Null

R Grouped by A

A	B	C
a1	b1	80
a1	b1	90
a1	b2	Null
Null	b3	60
Null	b3	40
a2	b4	90
a2	b4	60

Ex-3 Select A

1. From R
2. GROUP BY A
3. HAVING C >60; direct Attribute comparison fails

The condition can be one of

aggregate function or other function like SOME, ANY, ALL etc.

Topic: SQL Vs RA Query

Basic SQL clause

SELECT (DISTINCT) A1, A2,.....An

FROM R1, R2,..... Rm

[WHERE condition]

[GROUP BY (Attributes) [Having condition]]

[ORDER BY (Attributes) [DESC];

[ ] is optional.

Topic: SQL Vs RA Query

Basic SQL clause

SELECT (DISTINCT) A1, A2,.....An
FROM R1, R2,..... Rm
[WHERE condition]
[GROUP BY (Attributes) [Having condition]
[ORDER BY (Attributes) [DESC];
[ ] is optional.

Each record

Each group

Sorts in either ascending or descending order

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

Sol: Let's recall RA,

$$\pi_{eid}(Emp) - \pi_{eid}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp))$$

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)
retrieve eid's who get maximum salary

Sol: Let's recall RA,

$\pi_{eid}(Emp) - \pi_{eid}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp))$

```
(select T1.eid
FROM EMP T1, EMP T2
WHERE T1.Sal < T2.Sal;)
```

Employee ids whose salary < some emp salary

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)
retrieve eid's who get maximum salary

Sol: Let's recall RA,

$\pi_{eid}(Emp) - \pi_{eid}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp))$

(select eid from EMP) } all empids

(select T1.eid
FROM EMP T_1, EMP T_2
WHERE T_1.Sal < T_2.Sal;) } Employee ids whose salary < some emp salary

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

Sol: Let's recall RA,

$\pi_{eid}(Emp) - \pi_{eid}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp))$

EXCEPT is used for this.

```
(select eid from EMP)
    EXCEPT
(select T1.eid
FROM EMP T1, EMP T2
WHERE T1.Sal < T2.Sal;)
```

all empids

Employee ids whose salary < some emp salary

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

Sol: Let's recall RA,

$\pi_{eid}(Emp) - \pi_{eid}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp))$

EXCEPT is used for this.

SQL:

Without using Aggregation max()

```
(select eid from EMP)
    EXCEPT
(select T1.eid
FROM EMP T1, EMP T2
WHERE T1.Sal < T2.Sal;)
```

all empids

Employee ids whose salary < some emp salary

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)
retrieve eid's who get maximum salary

- Using max ( );

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)
retrieve eid's who get maximum salary

- Using max ( );

80

select max (sal) from emp

Eid	Sid
e1	70
e2	80
e3	60
e4	80
e5	40

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

- Using max ( );

```
SELECT eid
    FROM EMP
    WHERE Sal = {select max (sal) from emp};
```

Compares with constant

80

Eid	Sid
e1	70
e2	80
e3	60
e4	80
e5	40

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

- Another way:

```
Select eid
FROM Emp, (Select max (sal) maxsal)
          from emp) temp
WHERE emp. Sal = temp. maxsal;
```

table

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)

retrieve eid's who get maximum salary

- Another way:

```
Select eid
1. FROM Emp, (Select max (sal) maxsal)
                from emp) temp
2. WHERE emp. Sal = temp. maxsal;
```

table

Emp

eid	sid
e1	70
e2	80
e3	60
e4	80
e5	40

temp.

Maxsal
80

Topic: SQL Queries Practice

Ex- for relation emp (eid , salary)
retrieve eid's who get maximum salary

- Another way:

```
Select eid
1. FROM Emp, (Select max (sal) maxsal)
                from emp) temp
2. WHERE emp. Sal = temp. maxsal;
```

Emp

	eid	sid
✗	e1	70
✓	e2	80
✗	e3	60
✓	e4	80
✗	e5	40

temp.

Maxsal
80

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve eid's who gets second highest salary

Emp

eid	sid
e1	70
e2	80
e3	60
e4	80
e5	40

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve eid's who gate second highest salary

Sol: $\rho(Temp, \Pi_{eid,sal}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp)))$

All emp's who doesn't get max sal.

Emp

eid	sid
e1	70
e2	50
e3	80
e4	60

Temp

emp	Sal
e1	70
e2	50
e4	60

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve eid's who gate second highest salary

Sol: $\rho(Temp, \Pi_{eid,sal}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp)))$

$\Uparrow$

All emp's who doesn't get max sal.

$\pi_{eid}(Temp) - \pi_{eid}(Temp \bowtie_{Sal<S} \rho_{I,s}(Temp))$

e2
e4

Emp

eid	sid
e1	70
e2	50
e3	80
e4	60

Temp

emp	Sal
e1	70
e2	50
e4	60

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve eid's who gate second highest salary

Sol: $\rho(Temp, \Pi_{eid,sal}(Emp \bowtie_{Sal<S} \rho_{I,s}(Emp)))$

$\Uparrow$

All emp's who doesn't get max sal.

$\pi_{eid}(Temp) - \pi_{eid}(Temp \bowtie_{Sal<S} \rho_{I,s}(Temp))$

e1 e2 e4

e2 e4

Emp

eid	sid
e1	70
e2	50
e3	80
e4	60

Temp

emp	Sal
e1	70
e2	50
e4	60

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve emp's who gate second highest salary

- Using max ( );

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve emp's who gate second highest salary

Emp

eid	sal
e1	70
e2	50
e3	80
e4	60

```
SELECT max(sal)
FROM Emp)
```

80

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve emp's who gate second highest salary

```
Select max (sal)
    FROM Emp
    WHERE Sal< (SELECT max(sal)
                FROM Emp)
```

70

80

Emp

eid	sal
e1	70
e2	50
e3	80
e4	60

Topic: SQL Queries Practice

Ex-2 emp (eid, sal)

Retrieve emp's who gate second highest salary

- Using max ( );

```
Select eid From EMP
WHERE Sal = (Select max (sal)
            FROM Emp
            WHERE Sal< (SELECT max(sal)
                        FROM Emp));
```

70

80

eid
e1

Emp

eid	sal
e1	70
e2	50
e3	80
e4	60

Inspiring Stories : Jose Salvador

Background: **Fisherman lost at sea in the Pacific.**

Struggle: **Alone in a tiny boat for over a year without safety gear.**

Achievements: **Survived 14 months at sea on rainwater, fish, turtles. Landed on a remote island.**

Impact: **One of the longest survival stories ever recorded.**

THANK - YOU