Functional Dependency

Complete Summary

Instructor:

Deepak Poonia

MTech, IISc Bangalore

GATE CSE AIR 53; AIR 67;
AIR 107; AIR 206; AIR 256

DBMS Complete Course:
https://www.goclasses.in/courses/Database-Management-Systems

NOTE :

Complete Discrete Mathematics & Complete Engineering Mathematics Courses, by GO Classes, are **FREE** for ALL learners.

Visit here to watch : https://www.goclasses.in/s/store/

SignUp/Login on Goclasses website for free and start learning.


GO CLASSES
FREE
COMPLETE COURSE
Engineering Mathematics
GATE CS IT
Freely Available Features
FREE
All Video Lectures
Quizzes
Homeworks, Practice Sets
Annotated Notes
Toppers' Hand Written Notes
GATE PYQs Video Solutions
SACHIN MITTAL SIR
www.goclasses.in
/ GoClasses


GO CLASSES
FREE
COMPLETE COURSE
Discrete Mathematics
GATE CS IT
Freely Available Features
FREE
All Video Lectures
Quizzes, Homeworks
Annotated Notes
Toppers' Hand Written Notes
Summary Lectures
GATE PYQs Video Solutions
DEEPAK POONIA SIR
www.goclasses.in
/ GoClasses

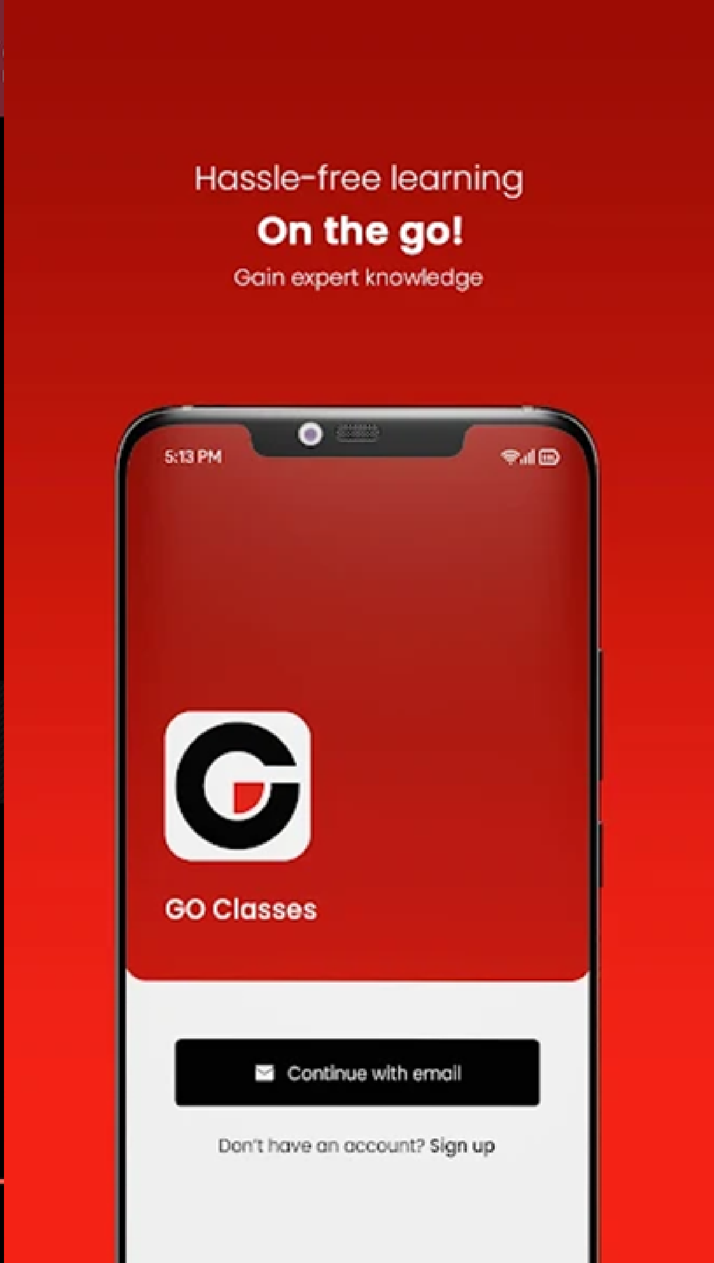
GO CLASSES

GO Classes

Download the GO Classes Android/ iOS App:

https://play.google.com/store/apps/details?id=com.goclasses.courses

Search “GO Classes” on Play Store / App Store.



We are on Telegram. **Contact us** for any help.

Link in the Description!!

Join GO Classes **Doubt Discussion** Telegram Group :

Username:

@GATECSE_GOCLASSES

We are on Telegram. **Contact us** for any help.

Join GO Classes Telegram Channel, Username: **@GOCLASSES_CSE**

Join GO Classes **Doubt Discussion** Telegram Group :

Username: **@GATECSE_Goclasses**

(Any doubt related to Goclasses Courses can also be asked here.)

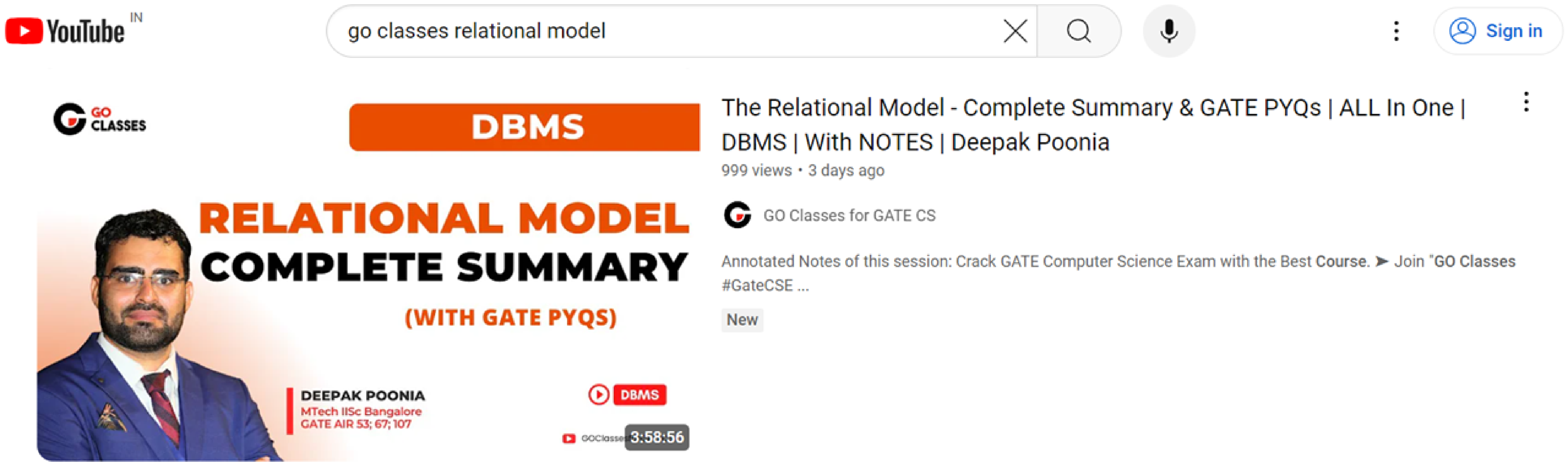
Join GATEOverflow Doubt Discussion Telegram Group :

Username: **@GateOverflow_CSE**

Functional Dependency

Complete Summary

The Relational Model – All in One



The Relational Model

Functional Dependency

Functional Dependency:

Another type of Integrity Constraint, defined on Schema.

Functional Dependency: Another type of Integrity Constraint on Schema.

Example: Consider an Indian School Database:

Students (sid, sname, district, state)

t_1			kota	Rajasthan
t_2			kota	Rajasthan
t_3			Jaipur	Rajasthan
t_4			mumbai	maharastra

Functional Dependency: Another type of Integrity Constraint on Schema.

Example: Consider an Indian School Database:

Students (sid, sname, district, state)

District $\longrightarrow$ State

$\equiv$ District Determines State

$\forall$ tuples t_1, t_2 $\left(t_1.\text{District} = t_2.\text{District} \Rightarrow t_1.\text{State} = t_2.\text{State} \right)$

Functional Dependency:

Example: Students (sid, sname, district, state)

District implies State.

District functionally determines State.

District determines State.

State is determined by District.

State is functionally dependent on District.

State is dependent on District.

$$\boxed{\begin{array}{c}\text{District}\\ \downarrow \\ \text{State}\end{array}}$$

functional Dependency

Functional Dependency:

Let X, Y be set of attributes in a Relation R.

Then we say that $X \rightarrow Y$ iff Whenever two tuples have same X value then those tuples also have same Y value.

Functional Dependency:

Let X, Y be set of attributes in a Relation R.

Then we say that $X \rightarrow Y$ if and only if

for all Tuples t_1, t_2 $\left(\begin{array}{c} t_1.X = t_2.X \\ \Downarrow \\ t_1.Y = t_2.Y \end{array} \right)$

R(A, B, C, D)

$A \rightarrow B$

R(A B C D)

4	5
4	5
7	10
7	10
9	20

✓ A Determines B

✓ A Implies B

✓ A functionally Determines B

R

A	B	C
0	2	4
0	3	4
1	3	5
0	2	4

Instance

- $C \rightarrow A$ In this Instance

$A \rightarrow B$ ✗

✓ A Does NOT Determine B

✓ $A \not\rightarrow B$

$AB \rightarrow C$

$B \not\rightarrow C$

- $A \rightarrow C$ in this instance

R (A B C D)

t_1 $\boxed{\alpha \quad \beta}$ γ

t_2 $\boxed{\alpha \quad \beta}$ γ

$X \rightarrow Y$

X: set of attributes

Y: set of attributes

$\{A, B\} \rightarrow \{C\}$

Easy way to write

$AB \rightarrow C$

Same

Functional Dependency:

Q: Is $AB \rightarrow C$ is Same as $BA \rightarrow C$??

Functional Dependency:

Q: Is $\underbrace{AB \rightarrow C}$ is Same as $\underbrace{BA \rightarrow C}$?? Yes.

$\{A, B\} \rightarrow \{C\}$

$\{A, B\} \rightarrow \{C\}$

Note :

$\{A, B, C\} \rightarrow \{C, D\}$

$ABC \rightarrow CD$

Same

Convention / Easy way to write

Functional Dependency:

Formally:
If column A of a table uniquely identifies the column B of same table then it can represented as A->B (Attribute B is functionally dependent on attribute A)

District $\longrightarrow$ State

An Important NOTE:

Functional Dependencies are defined on Schema...

So, ALL possible Instances must satisfy them.

Example

Assume that the following FDs hold:

$$\text{cid} \rightarrow \text{title}$$
$$\text{title} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{cid, dept}$$

Can the following tuples co-exist?

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c2$	database	2012	ee

Example

Assume that the following FDs hold:

$$\text{cid} \rightarrow \text{title}$$
$$\boxed{\text{title} \rightarrow \text{dept}}$$ Violated
$$\text{cid, year} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{cid, dept}$$

Can the following tuples co-exist? No

Invalid Instance

cid	title	year	dept
$c1$	database	2010	cs
$c1$	database	2011	cs
$c2$	database	2012	ee

Illegal Instance

Functional Dependency "by an Instance":

- Given an instance of $r(R)$, we say that the instance **satisfies** the **functional dependency** $\alpha \rightarrow \beta$ if for all pairs of tuples t_1 and t_2 in the instance such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.

Exercise 1 (2 pts)

Consider the relation instance and functional dependencies below. Which of these FDs hold on the instance and which do not?

A	B	C
0	2	4
0	3	4
1	3	5
0	2	4

$A \rightarrow B$ ✗ $\quad$ $A \rightarrow C$ ✓ $\quad$ $B \rightarrow C$ ✗

$B \rightarrow A$ ✗ $\quad$ $C \rightarrow A$ ✓ $\quad$ $C \rightarrow B$ ✗

Functional Dependency:

- Given an instance of $r(R)$, we say that the instance **satisfies** the **functional dependency** $\alpha \rightarrow \beta$ if for all pairs of tuples t_1 and t_2 in the instance such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.
- We say that the functional dependency $\alpha \rightarrow \beta$ **holds** on schema $r(R)$ if, in every legal instance of $r(R)$ it satisfies the functional dependency.

Q:

If an instance I of relation R satisfies FD A --> B

Then Can we say that R satisfies this FD?

Q:

If an instance I of relation R satisfies FD A --> B

Then Can we say that R satisfies this FD?

No

Can NOT say.

Student (sid, name, state)

monu	up
monu	up
Sonu	mp

→ Instance

In this Instance,
Name → state

On schema

Name $\not\rightarrow$ State

Q:

If an instance I of relation R does Not satisfy FD A --> B Then Can we say that R doesn't satisfy this FD?

Q:

If an instance I of relation R does Not satisfy FD A --> B Then Can we say that R doesn't satisfy this FD? Yes.

Functional Dependency:

- Given an instance of $r(R)$, we say that the instance **satisfies** the **functional dependency** $\alpha \rightarrow \beta$ if for all pairs of tuples t_1 and t_2 in the instance such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.
- We say that the functional dependency $\alpha \rightarrow \beta$ **holds** on schema $r(R)$ if, in every legal instance of $r(R)$ it satisfies the functional dependency.

NOTE:

A functional dependency is a *property of the relation schema R*, not of a particular legal relation state *r* of *R*. Therefore, an FD *cannot* be inferred automatically from a given relation extension *r* but must be defined explicitly by someone who knows the semantics of the attributes of *R*. For example, Figure 14.7 shows a *particular*

NOTE:

Recall that a *legal* instance of a relation must satisfy all specified ICs, including all specified FDs. As noted in Section 3.2, ICs must be identified and specified based on the semantics of the real-world enterprise being modeled. By looking at an instance of a relation, we might be able to tell that a certain FD does *not* hold. However, we can never deduce that an FD *does* hold by looking at one or more instances of the relation because an FD, like other ICs, is a statement about *all* possible legal instances of the relation.

[1] $X \rightarrow Y$ is read as X *functionally determines* Y, or simply as X *determines* Y.

Let us consider the instance of relation r of Figure 8.4, to see which functional dependencies are satisfied. Observe that $A \rightarrow C$ is satisfied. There are two tuples

In this Instance

$A \rightarrow C$

But we can't say it for schema.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

Figure 8.4 Sample instance of relation r.

that have an A value of a_1. These tuples have the same C value—namely, c_1. Similarly, the two tuples with an A value of a_2 have the same C value, c_2. There are no other pairs of distinct tuples that have the same A value. The functional dependency $C \rightarrow A$ is not satisfied, however. To see that it is not, consider the tuples $t_1 = (a_2, b_3, c_2, d_3)$ and $t_2 = (a_3, b_3, c_2, d_4)$. These two tuples have the same C values, c_2, but they have different A values, a_2 and a_3, respectively. Thus, we have found a pair of tuples t_1 and t_2 such that $t_1[C] = t_2[C]$, but $t_1[A] \neq t_2[A]$.

Functional Dependencies

- $X \to Y$ is an assertion about a relation R that whenever two tuples of R agree on all the attributes of X, then they must also agree on all attributes in set Y.
  - Say "$X \to Y$ holds in R."
  - Convention: …, X, Y, Z represent sets of attributes; A, B, C, … represent single attributes.
  - Convention: no set formers in sets of attributes, just ABC, rather than $\{A,B,C\}$.

3.1.1 Definition of Functional Dependency

A *functional dependency* (FD) on a relation R is a statement of the form "If two tuples of R agree on all of the attributes $A_1, A_2, \ldots, A_n$ (i.e., the tuples have the same values in their respective components for each of these attributes), then they must also agree on all of another list of attributes $B_1, B_2, \ldots, B_m$. We write this FD formally as $A_1 A_2 \cdots A_n \to B_1 B_2 \cdots B_m$ and say that

"$A_1, A_2, \ldots, A_n$ functionally determine $B_1, B_2, \ldots, B_m$"

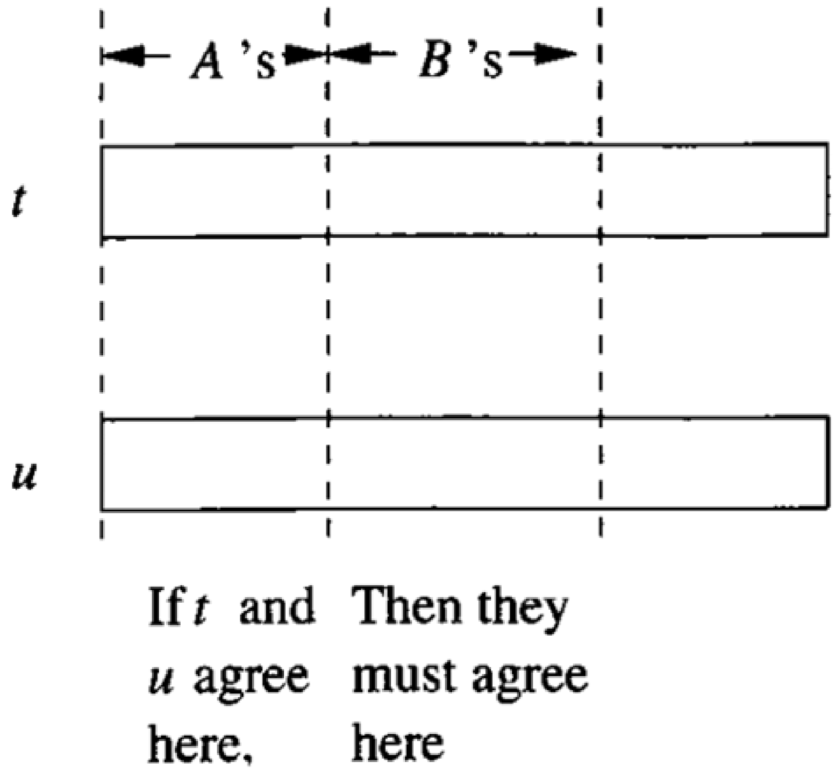


Figure 3.1: The effect of a functional dependency on two tuples.

Definition

A functional dependency (FD) has the form of $X \to Y$ (reads: X implies Y), where X and Y are sets of attributes. It means that whenever two tuples are identical on all the attributes in X, they must also be identical on all the attributes in Y.

Alternatively, you can interpret $X \to Y$ as: each possible value of X can correspond to exactly one value of Y.

1. (15 points) Consider a relation schema R(A,B,C) and its relation instance as follows:

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

Which of the following functional dependencies are satisfied by the above relation instance. If the dependency is not satisfied, specify the tuples that cause the violation.

1). $AB \rightarrow C$

2). $A \rightarrow B$

3). $C \rightarrow A$

4). $BC \rightarrow A$

5). $ABC \rightarrow A$

6). $AB \rightarrow AC$

GO Classes

1. (15 points) Consider a relation schema R(A,B,C) and its relation instance as follows:

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

$\boxed{AB \not\to C}$ In this Instance

So; on schema also $AB \not\to C$

Which of the following functional dependencies are satisfied by the above relation instance. If the dependency is not satisfied, specify the tuples that cause the violation.

1). AB → C — No

2). A → B ✓

3). C → A

4). BC → A

5). ABC → A

6). AB → AC

$A \to B$ in this Instance

But Can't say the Same thing for schema.

1. (15 points) Consider a relation schema R(A,B,C) and its relation instance as follows:

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

Which of the following functional dependencies are satisfied by the above relation instance. If the dependency is not satisfied, specify the tuples that cause the violation.

1). $AB \rightarrow C$

2). $A \rightarrow B$

3). $C \rightarrow A$ NO

4). $BC \rightarrow A$ NO

5). $ABC \rightarrow A$ YES

6). $AB \rightarrow AC$ No

A	B	C
1	2	3
1	2	3

3.5.15 Database Normalization: GATE CSE 2000 | Question: 2.24 top

https://gateoverflow.in/671

Given the following relation instance.

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance?

A. $XY \to Z$ and $Z \to Y$
B. $YZ \to X$ and $Y \to Z$
C. $YZ \to X$ and $X \to Z$
D. $XZ \to Y$ and $Y \to X$

3.5.20 Database Normalization: GATE CSE 2002 | Question: 2.25 top

https://gateoverflow.in/855

From the following instance of a relation schema $R(A, B, C)$, we can conclude that:

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

A. A functionally determines B and B functionally determines C
B. A functionally determines B and B does not functionally determine C
C. B does not functionally determine C
D. A does not functionally determine B and B does not functionally determine C

NOTE:

A functional dependency is a *property of the relation schema R*, not of a particular legal relation state *r* of *R*. Therefore, an FD *cannot* be inferred automatically from a given relation extension *r* but must be defined explicitly by someone who knows the semantics of the attributes of *R*. For example, Figure 14.7 shows a *particular*

NOTE:

Recall that a *legal* instance of a relation must satisfy all specified ICs, including all specified FDs. As noted in Section 3.2, ICs must be identified and specified based on the semantics of the real-world enterprise being modeled. By looking at an instance of a relation, we might be able to tell that a certain FD does *not* hold. However, we can never deduce that an FD *does* hold by looking at one or more instances of the relation because an FD, like other ICs, is a statement about *all* possible legal instances of the relation.

---

[1] $X \rightarrow Y$ is read as X *functionally determines* Y, or simply as X *determines* Y.

Functional Dependency:

- Given an instance of $r(R)$, we say that the instance **satisfies** the **functional dependency** $\alpha \rightarrow \beta$ if for all pairs of tuples t_1 and t_2 in the instance such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.
- We say that the functional dependency $\alpha \rightarrow \beta$ **holds** on schema $r(R)$ if, in every legal instance of $r(R)$ it satisfies the functional dependency.

3.5.15 Database Normalization: GATE CSE 2000 | Question: 2.24 top https://gateoverflow.in/671

Given the following relation instance.

$YZ \rightarrow X$ ✓

$XZ \not\rightarrow Y$

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

$Z \not\rightarrow Y$

$Y \rightarrow Z$

$Y \rightarrow X$

$X \not\rightarrow Z$

Which of the following functional dependencies are satisfied by the instance?

A. $XY \rightarrow Z$ and $Z \rightarrow Y$ ✗

B. $YZ \rightarrow X$ and $Y \rightarrow Z$ ✓

C. $YZ \rightarrow X$ and $X \rightarrow Z$ ✗

D. $XZ \rightarrow Y$ and $Y \rightarrow X$ ✗

3.5.20 Database Normalization: GATE CSE 2002 | Question: 2.25 top https://gateoverflow.in/855

From the following instance of a relation schema $R(A, B, C)$, we can conclude that:

In this Instance $A \to B$

But we can't say the same for schema

$B \not\to C$

Instance as well as schema

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

A. A functionally determines B and B functionally determines C ✗
B. A functionally determines B and B does not functionally determine C ✗
C. B does not functionally determine C ✓
D. A does not functionally determine B and B does not functionally determine C

for Schema

Can't say

Correct

FD hold by the instance

FD hold by the schema

FD holds.

Same

Types of Functional Dependencies:

1. Trivial FD
2. Non-Trivial FD

1. Trivial FD

A	B	C
0	2	4
0	3	4
1	3	5
0	2	4

$AB \rightarrow A$ Yes.

	A	B	A
t_1	5	10	5
t_2	5	10	5

$\underbrace{A \rightarrow A}$

Trivial FD

1. Trivial FD

If $X \supseteq Y$ then $X \rightarrow Y$ Trivial FD

Trivial FDs ALWAYS hold in Every instance of Every Relation.

So, these are of No interest.

X $ABC \longrightarrow BC$ Y

$X \supseteq Y$

Trivial FD

	A	B	C	B	C
t_1	2	2	3	2	3
t_2	2	2	3	2	3

2. Non-trivial FD

$\equiv$ NOT Trivial

$X \rightarrow Y$ is Non-Trivial FD iff $X \not\supseteq Y$

Eg: $AB \rightarrow C$ ✓

$AB \rightarrow A$ ✗

$\underbrace{AB \rightarrow BC}$ ✓

Non-Trivial FD

[redacted] Formally, a functional dependency $X \to Y$ is **trivial** if $X \supseteq Y$; otherwise, it is **nontrivial**. [redacted]

Source: Navathe

Q: R(A,B,C,D);

Which of the following is/are a non-trivial FD ??

1. A -> A
2. A -> B
3. AB -> B
4. A -> AB
5. AB -> AC
6. AB -> CD
7. ABCD -> A

Q: R(A,B,C,D);

Which of the following is/are a non-trivial FD ??

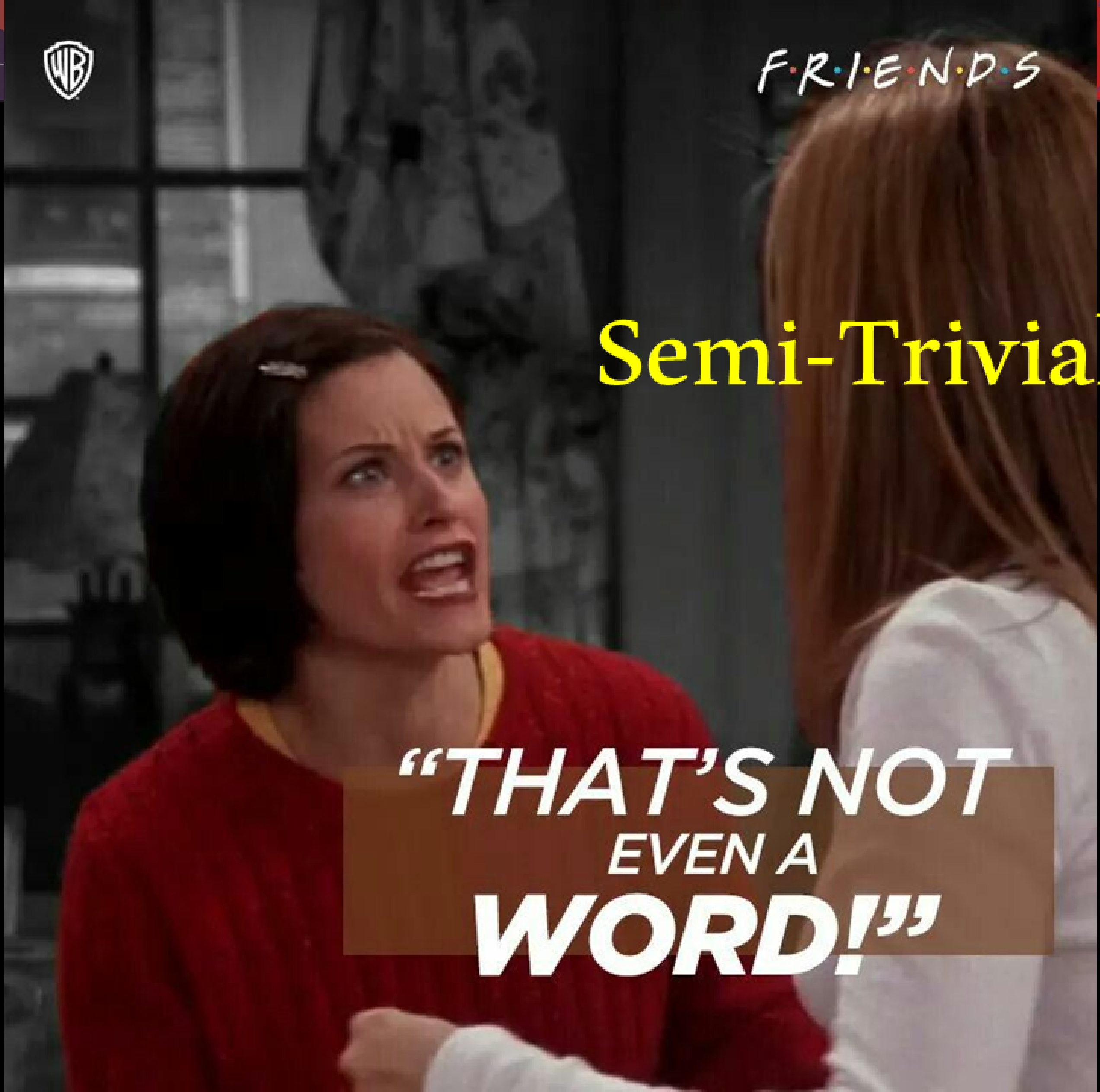
1. A -> A
2. A -> B
3. AB -> B
4. A -> AB
5. AB -> AC
6. AB -> CD
7. ABCD -> A

NOTE (A Misconception):

"Semi Trivial FD ; Semi Nontrivial FD"

No Such Concepts Exist in any Standard Resource.

Semi-Trivial FD



Some functional dependencies are said to be **trivial** because they are satisfied by all relations. For example, $A \rightarrow A$ is satisfied by all relations involving attribute A. Reading the definition of functional dependency literally, we see that, for all tuples t_1 and t_2 such that $t_1[A] = t_2[A]$, it is the case that $t_1[A] = t_2[A]$. Similarly, $AB \rightarrow A$ is satisfied by all relations involving attribute A. In general, a functional dependency of the form $\alpha \rightarrow \beta$ is **trivial** if $\beta \subseteq \alpha$.

3.2.3 Trivial Functional Dependencies

A constraint of any kind on a relation is said to be *trivial* if it holds for every instance of the relation, regardless of what other constraints are assumed. When the constraints are FD's, it is easy to tell whether an FD is trivial. They are the FD's $A_1A_2 \cdots A_n \rightarrow B_1B_2 \cdots B_m$ such that

$$\{B_1, B_2, \ldots, B_m\} \subseteq \{A_1, A_2, \ldots, A_n\}$$

That is, a trivial FD has a right side that is a subset of its left side. For example,

$$\texttt{title year} \rightarrow \texttt{title}$$

is a trivial FD, as is

$$\texttt{title} \rightarrow \texttt{title}$$

Every trivial FD holds in every relation, since it says that "two tuples that agree in all of $A_1, A_2, \ldots, A_n$ agree in a subset of them." Thus, we may assume any trivial FD, without having to justify it on the basis of what FD's are asserted for the relation.

Types of Functional Dependencies:

1. Trivial FD
2. Non-Trivial FD
3. Completely Non-trivial FD (Only Ullman Book Defines it)

FD $X \dashrightarrow Y$

1. Trivial FD : If $X \supseteq Y$

2. Non-Trivial FD : NOT Trivial ;
i.e. If $X \not\supseteq Y$

3. Completely Non-trivial FD (Ullman Book Defines it)
$X \cap Y = \phi$

FD $A_1A_2 \cdots A_n \rightarrow B_1B_2 \cdots B_m$ is

- *Trivial* if the B's are a subset of the A's.
- *Nontrivial* if at least one of the B's is not among the A's.
- *Completely nontrivial* if none of the B's is also one of the A's.

Thus

```
title year → year length
```

is nontrivial, but not completely nontrivial. By eliminating year from the right side we would get a completely nontrivial FD.

Regarding Trivial Functional Dependency, which best describes completely non-trivial?

- None of these
- If a functional dependency (FD) $X \rightarrow Y$ holds, where Y is a subset of X
- If an FD $X \rightarrow Y$ holds, where x intersect $Y = \Phi$
- If an FD $X \rightarrow Y$ holds, where Y is not a subset of X

Regarding Trivial Functional Dependency, which best describes completely non-trivial?

- None of these
- If a functional dependency (FD) $X \rightarrow Y$ holds, where Y is a subset of X

  Trivial FD
- If an FD $X \rightarrow Y$ holds, where x intersect $Y = \Phi$

  Completely Non-Trivial
- If an FD $X \rightarrow Y$ holds, where Y is not a subset of X

  Non-Trivial FD

Another Notation for Functional Dependencies:

$A \rightarrow B$

$R(A \quad B \quad C \quad D)$

$AB \rightarrow C$

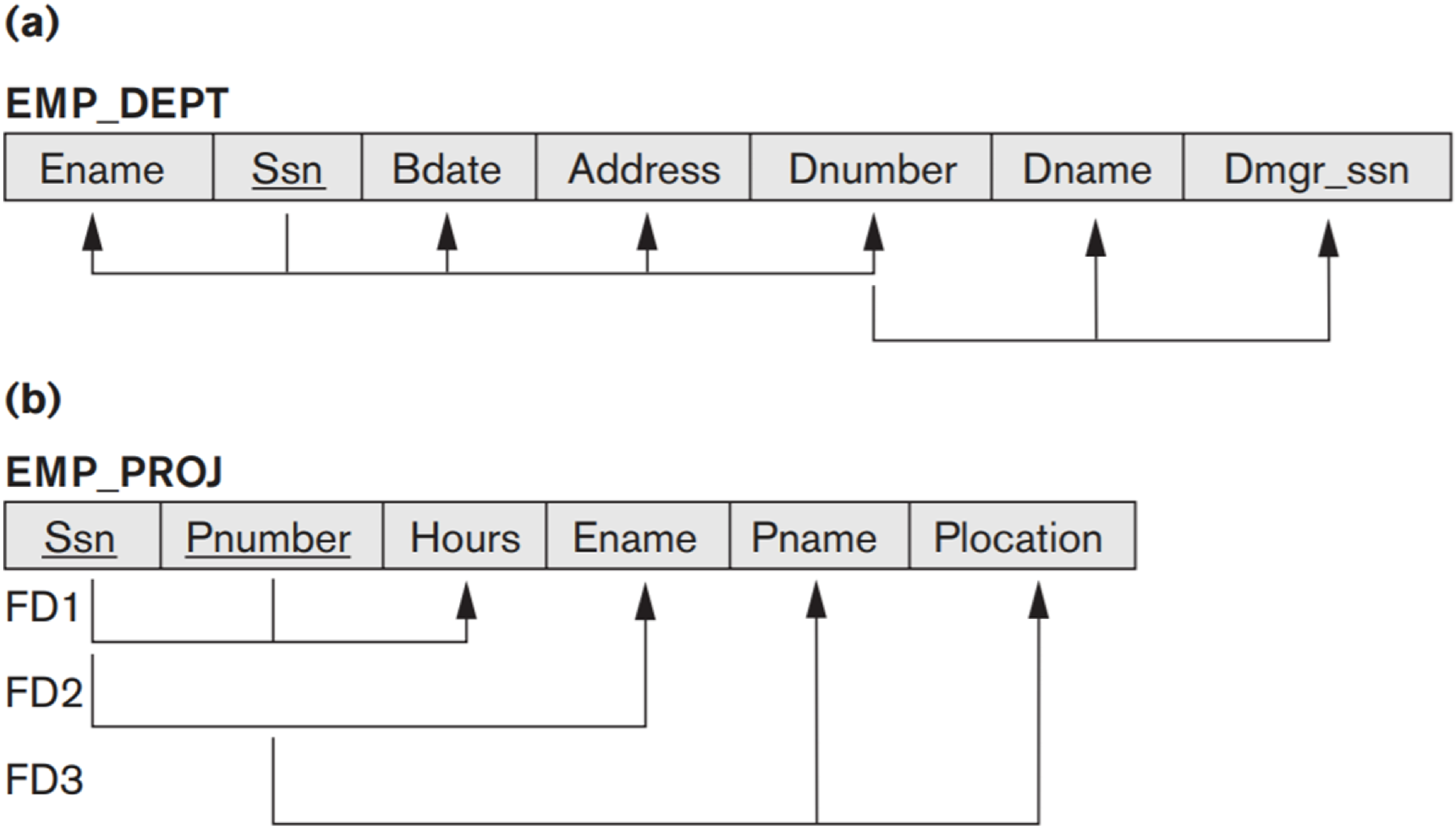
R

A	B	C	D

$AB \rightarrow CD$

R

A	B	C	D



(a)
EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
(b)
EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3

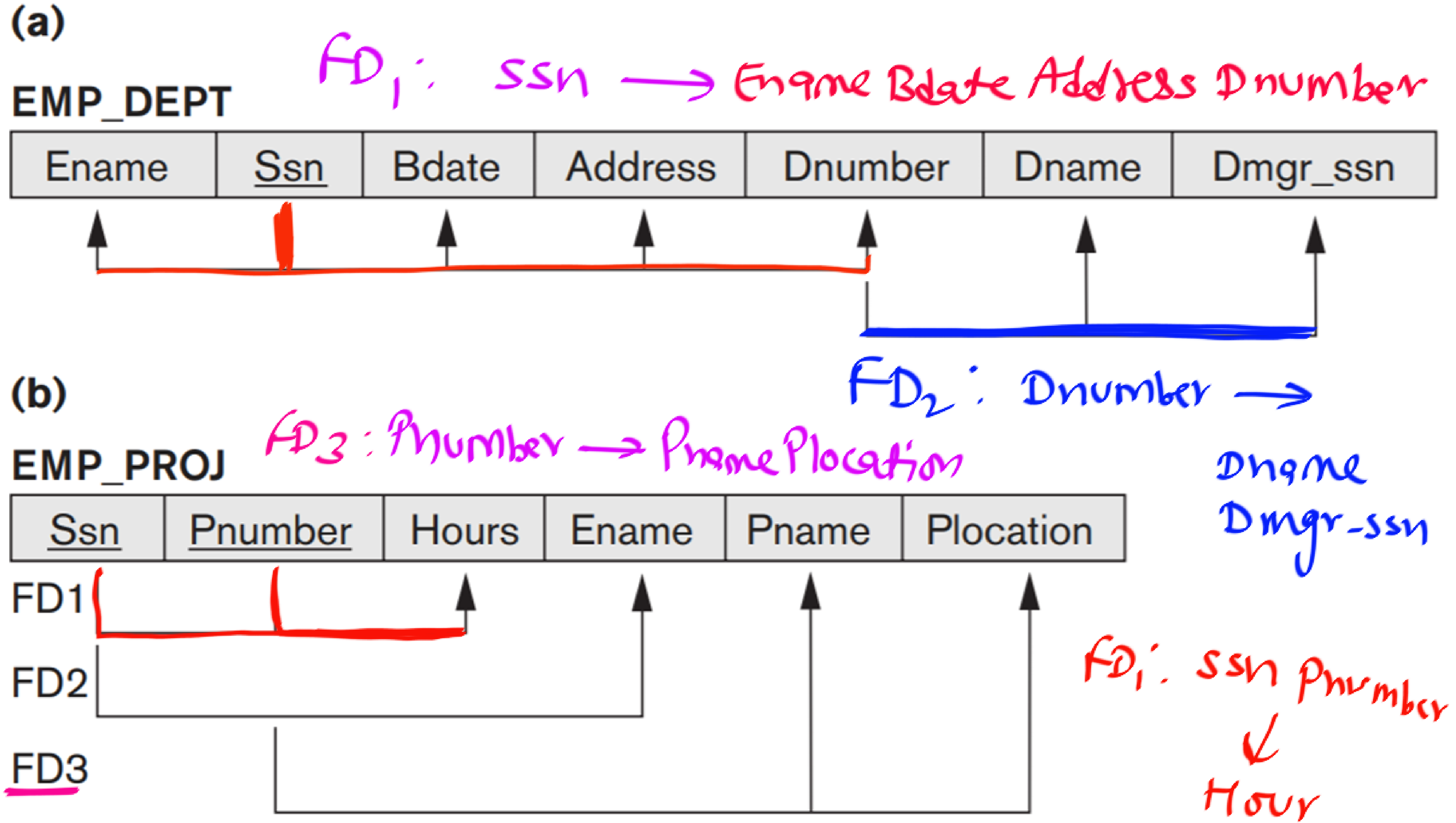
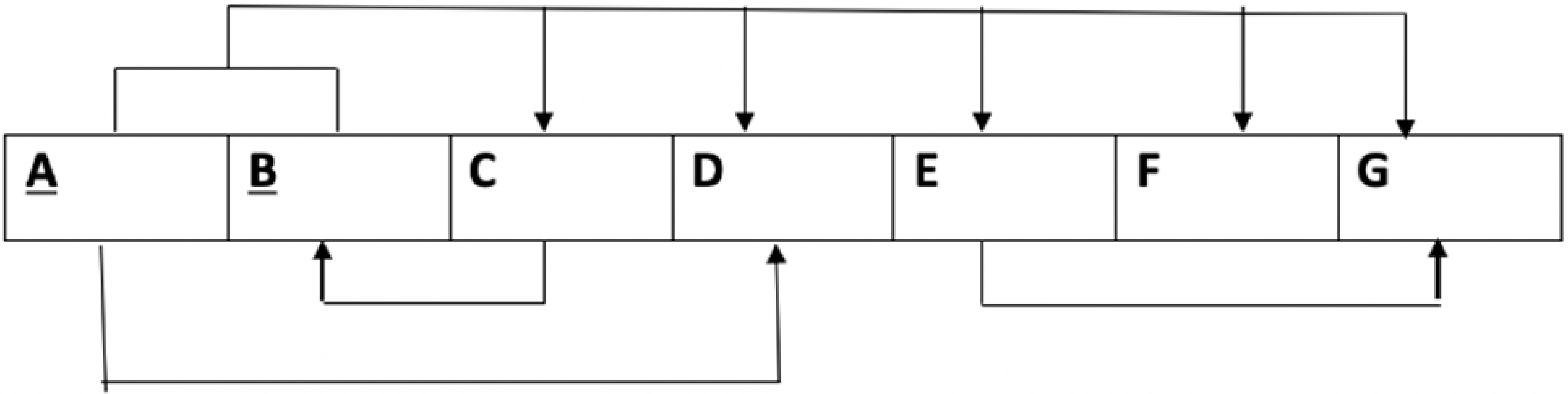

(a)
FD1: ssn → Ename Bdate Address Dnumber
EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
FD2: Dnumber →
Dname
Dmgr-ssn
(b)
FD3: Pnumber → Pname Plocation
EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3
FD1: ssn Pnumber → Hour

Figure 14.3 introduces a **diagrammatic notation** for displaying FDs: Each FD is displayed as a horizontal line. The left-hand-side attributes of the FD are connected by vertical lines to the line representing the FD, whereas the right-hand-side attributes are connected by the lines with arrows pointing toward the attributes.

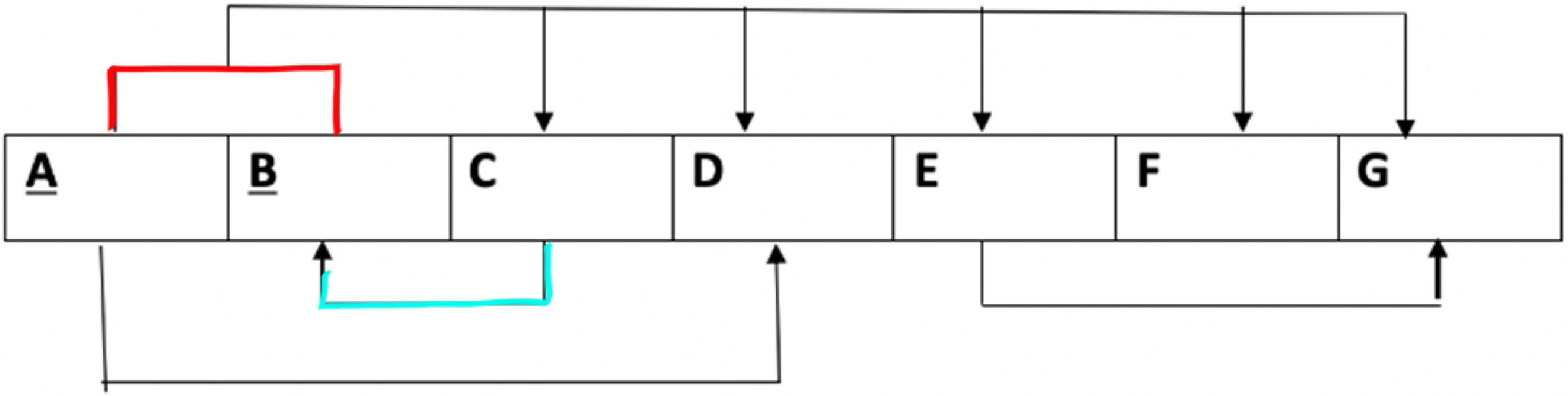
Exercise 2:



FDs :

Exercise 2:

$AB \rightarrow CDEFG$



FDs:

$C \rightarrow B$; $A \rightarrow D$

$E \rightarrow G$

For the relation Apply(SSN,cName,state,date,major), what real-world constraint is captured by SSN,cName → date?

- A student can only apply to one college.
- A student can apply to each college only once.
- A student must apply to all colleges on the same date.
- Every application from a student to a specific college must be on the same date.

Skip

For the relation Apply(SSN,cName,state,date,major), what real-world constraint is captured by SSN,cName → date?

- A student can only apply to one college.
- A student can apply to each college only once.
- A student must apply to all colleges on the same date.
- Every application from a student to a specific college must be on the same date.

Skip

Apply (ssn, cname, date, state, major)

FD: $\boxed{\text{ssn. cname} \longrightarrow \text{date}}$

t_1	S_1	IITD	2 march
t_2	S_1	IITD	2 march
t_2	S_1	IITD	2 march

Consider a relation R(A,B,C,D,E) with functional dependencies:

$A,B \rightarrow C$

$C,D \rightarrow E.$

Suppose there are at most 3 different values for each of A, B, and D. What is the maximum number of different values for E?

- 27
- 9
- 3
- 81

Video Solution Link in the Description!!

Laws (Rules) for Functional Dependencies:

Functional Dependency Laws:

1. Reflexivity: (Law of Trivial FDs)

Reflexivity: If $Y \subseteq X$ then, $X \to Y$. Such FDs are called *trivial* FDs.

1. *Reflexivity.* If $\{B_1, B_2, \ldots, B_m\} \subseteq \{A_1, A_2, \ldots, A_n\}$, then $A_1 A_2 \cdots A_n \to B_1 B_2 \cdots B_m$. These are what we have called trivial FD's.

Functional Dependency Laws:

2. Augmentation:

Augmentation: If $X \to Y$, then $XZ \to YZ$.

2. *Augmentation.* If $A_1A_2 \cdots A_n \to B_1B_2 \cdots B_m$, then

$$A_1A_2 \cdots A_nC_1C_2 \cdots C_k \to B_1B_2 \cdots B_mC_1C_2 \cdots C_k$$

Functional Dependency Laws:

Augmentation: If $X \to Y$, then $XZ \to YZ$.

Proof: Given: $X \to Y$

To Check: $XZ \to YZ$

	X	Z	Y	Z
t_1	a	b	c	b
t_2	a	b	c	b

Hence Proved

Functional Dependency Laws:

3. Transitivity:

Transitivity: If $X \to Y$ and $Y \to Z$, then $X \to Z$.

3. *Transitivity.* If

$$A_1 A_2 \cdots A_n \to B_1 B_2 \cdots B_m \text{ and } B_1 B_2 \cdots B_m \to C_1 C_2 \cdots C_k$$

then $A_1 A_2 \cdots A_n \to C_1 C_2 \cdots C_k$.

Inference rules for FDs (Armstrong's Axioms)

Reflexivity: If $Y \subseteq X$ then, $X \to Y$. Such FDs are called *trivial* FDs.

Augmentation: If $X \to Y$, then $XZ \to YZ$.

Transitivity: If $X \to Y$ and $Y \to Z$, then $X \to Z$.

Some More Functional Dependency Laws:

4. Split on RHS Law:

if $X \to YZ$ then $X \to Y$ and $X \to Z$.

4. Split on RHS Law:

if $X \to YZ$ then $X \to Y$ and $X \to Z$.

Proof: Given: $X \to YZ$

To check: $X \to Y$, $X \to Z$

	X	Y	Z
t_1	a	b	c
t_2	a	b	c

$X \to Y$ ✓

$X \to Z$

- Splitting Rule:

$A_1A_2\ldots A_n \rightarrow B_1B_2\ldots B_m$

Is equivalent to:

$A_1A_2\ldots A_n \rightarrow B_1$

$A_1A_2\ldots A_n \rightarrow B_2$

…

$A_1A_2\ldots A_n \rightarrow B_m$

NOTE:

We Can NOT Split FD on LHS.

i.e. If $XY \rightarrow Z$ then it does Not imply that $X \rightarrow Z$ and $Y \rightarrow Z$??

Ef: $\underbrace{XY}_{} \rightarrow Z$

	X	Y	Z
t_1	a	b	c
t_2	a	b	c
t_3	a	e	f

$XY \rightarrow Z$ ✓

But

$X \not\rightarrow Z$

Some More Functional Dependency Laws:

5. Combine on RHS Law:

Additional Rules of Inference:

Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$.

The Splitting / Combining Rule (cont'd)

- Combining Rule:

Consider the following FD's:

$A_1A_2\ldots A_n \rightarrow B_1$

$A_1A_2\ldots A_n \rightarrow B_2$

…

$A_1A_2\ldots A_n \rightarrow B_m$

We can combine them in one FD as:

$A_1A_2\ldots A_n \rightarrow B_1B_2\ldots B_m$

NOTE:

We can Split as well as Combine FDs on RHS.

① If $A \to BC$ then $A \to B$ & $A \to C$

② If $A \to B$ & $A \to C$ then $A \to BC$

Q: Can we Combine FD on LHS??

i.e. If $X \rightarrow Z$ and $Y \rightarrow Z$ then does it imply that $XY \rightarrow Z$??

Q: Can we Combine FD on LHS?? YES

i.e. If X → Z and Y → Z then does it imply that XY → Z ??

Proof: Given: $X \to Z$, $Y \to Z$

To check: $XY \to Z$

	X	Y	Z
t_1	a	b	c
t_2	a	b	c

proves

NOTE:

We can Split as well as Combine FDs on RHS.

We can only Combine FDs on LHS, Not split.

The Relational Model

Closure of a Set of Attributes

$R(ABCDE)$; set of FDs F

If $X \subseteq R$

Closure of $X = X^+ = X^*$

= set of all attributes which X can Determine.

7 Closure

Given a set of functional dependencies F, and a set of attributes X, the *closure* of X with respect to F, written X_F^+, is the set of all the attributes whose values are determined by the values of X because of F. Often, if F is understood, then the closure of X is written X^+.

Closure of an Attribute Set X = X^+ = X^*

The set of all attributes which X can determine.

Method to find X^+ :

1. Add X in the Result Set.

2. Recursively add the attributes to the result set which can be functionally determined from the attributes already contained in the result set.

For example, given the following set, M, of functional dependencies:

1. $A \to B$
2. $B \to C$
3. $BC \to D$

$A^+ = ABCD = \{A, B, C, D\}$

$B^+ = BCD$

$D^+ = D$

$C^+ = C$

Then we can compute the closure of A with respect to M in the following way:

$(AC)^+ = ACBD$

Can Not Split on LHS.

we can NOT say $C \to D$

Example 3.8: Let us consider a relation with attributes A, B, C, D, E, and F. Suppose that this relation has the FD's $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, and $CF \rightarrow B$. What is the closure of $\{A, B\}$, that is, $\{A, B\}^+$?

Example 3.8: Let us consider a relation with attributes A, B, C, D, E, and F. Suppose that this relation has the FD's $AB \to C$, $BC \to AD$, $D \to E$, and $CF \to B$. What is the closure of $\{A, B\}$, that is, $\{A, B\}^+$?

$(AB)^+ = \{A, B\}^+ = ABCDE$

$AB \longrightarrow AB$

Trivially

$(AB)^+ = \{A, B, C, D, E\}$

$= ABCDE$

Example 3.8: Let us consider a relation with attributes A, B, C, D, E, and F. Suppose that this relation has the FD's $AB \to C$, $BC \to AD$, $D \to E$, and $CF \to B$. What is the closure of $\{A, B\}$, that is, $\{A, B\}^+$?

$A^+ = A$

$B^+ = B$

$D^+ = DE$

$(CF)^+ = \underbrace{CFBADE}$

Example: Let F be the set of following FDs:

$$\text{cid} \to \text{title}$$
$$\text{title} \to \text{dept}$$
$$\text{cid, year} \to \text{dept}$$
$$\text{cid, year} \to \text{cid, dept}$$

What is the closure of $X = \{\text{cid, year}\}$?

Example: Let F be the set of following FDs:

$$\text{cid} \rightarrow \text{title}$$
$$\text{title} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{dept}$$
$$\text{cid, year} \rightarrow \text{cid, dept}$$

What is the closure of $X = \{\text{cid, year}\}$?

$X^+ =$ cid year title dept

3.5.25 Database Normalization: GATE CSE 2006 | Question: 70 top

https://gateoverflow.in/1848

The following functional dependencies are given:

$$AB \rightarrow CD, AF \rightarrow D, DE \rightarrow F, C \rightarrow G, F \rightarrow E, G \rightarrow A$$

Which one of the following options is false?

A. $\{CF\}^* = \{ACDEFG\}$
B. $\{BG\}^* = \{ABCDG\}$
C. $\{AF\}^* = \{ACDEFG\}$
D. $\{AB\}^* = \{ABCDG\}$

Video Solution Link in the Description!!

gate2006-cse databases database-normalization normal

The Relational Model

Defining Keys in terms of FDs

Q: In relation R (A,B,C,D) ;

When will FD $A \rightarrow B$ violate??

Q: In relation R (A,B,C,D) ;

[redacted] FD A → B violates if

There exists an instance such that

	A	B	C	D
t_1	2	5		
t_2	2	7		

Q:

If X is Superkey then what is $(X)^+$??

If X : SK

then in any Instance ;

	X	A	B	C	D
t_1					
t_2					

Different

$X \to A$ Can Never Violate

So, $X \to A$

NOTE:

$$\boxed{X \text{ is Superkey if and only if } (X)^+ = R.}$$

X is Sk iff X ⟶ all attributes

NOTE:

X is Superkey if and only if $(X)^{+} = R$.

X is Candidate Key iff $(X)^{+} = R$ AND X is minimal.

minimal SK

Example. Consider a table $R(A, B, C, D)$, and that $F = \{A \to B, B \to C\}$.

$A^+ = ABC \neq R$; So; A: NOT SK

$(ACD)^+ = ACDB = R$; ACD: SK

$\boxed{(AD)^+ = ADBC = R}$; AD: CK, SK

AD: CK because $(AD)^+ = R$ & AD is minimal

Example. Consider a table $R(A, B, C, D)$, and that $F = \{A \rightarrow B, B \rightarrow C\}$.

- A is not a candidate key, because $A^+ = \{A, B, C\}$ which does not include D.
- ABD is not a candidate key even though $ABD^+ = \{A, B, C, D\}$. This is because $AD^+ = \{A, B, C, D\}$, namely, there is a proper subset AD of ABD such that AD^+ includes all the attributes.
- AD is a candidate key. $\equiv$ minimal SK

Note:

X is a Candidate key iff

(1) $X^+ = R$

&

(2) X is minimal SK i.e. any proper subset of X is NOT a SK.

Note:

X is a Candidate key iff

(1) $X^+ = R$

&

(2) X is minimal SK i.e. any proper subset Y of X then $Y^+ \neq R$

Question :

In relation R; If $(X)^+$ is entire R then what is X??

A. Candidate Key

B. Superkey

Question :

In relation R; If $(X)^+$ is entire R then what is X??

A. Candidate Key

B. Superkey

$X^+ = R$

SK

Q: In relation R; If $(X)^+$ is entire R and for every Y which is a proper subset of X, $(Y)^+$ is Not R.

Then what is X??

A. Candidate Key

B. B. Super key

Q: In relation R; If $(X)^+$ is entire R and for every Y which is a proper subset of X, $(Y)^+$ is Not R.

Then what is X??

A. Candidate Key

B. B. Super key

Candidate key : minimal SK

<u>Note</u>:

$$\boxed{\text{Key} \equiv \text{Candidate Key}}$$

By Default

Keys of Relations

- K is a *superkey* for relation R if K functionally determines all of R.
- K is a *key* for R if K is a superkey, but no proper subset of K is a superkey.

Candidate key

Candidate Key Revisited

Let F be a set of FDs, and R a relation.

Definition

A candidate key is a set X of attributes in R such that

- X^+ includes all the attributes in R.
- There is no proper subset Y of X such that Y^+ includes all the attributes in R.

Note: A proper subset Y is a subset of X such that $Y \neq X$ (i.e., X has at least one element not in Y).

CANDIDATE KEYS

A *candidate key* of a relation schema R is a subset X of the attributes of R with the following two properties:

1. Every attribute is functionally dependent on X, i.e., X^+ = all attributes of R (also denoted as $X^+ = R$).

2. No proper subset of X has the property (1), i.e., X is minimal with respect to the property (1).

Q:

If X is a CK or SK ... & Y ---> X

Then what is Y ??

Q: $X \rightarrow R$

If X is a CK or SK ... & Y ---> X

Then what is Y ??

Y : SK

$Y \longrightarrow X \longrightarrow R$ $\longrightarrow$ $Y \rightarrow R$

so; Y : SK

X : CK or SK

means: $X \to R$

Given: $Y \to X$

So; $Y \to X \to R$, So; $Y \to R$

So; Y : SK

(If Y is Also minimal then Y : CK)

NOTE:

If X is a CK then X can Not be the part of any Bigger CK.

Q:

In relation R (A,B,C,D,E) ; Given set of FDs S.

If E does not appear on RHS of any FD in S,

then who can determine E??

Q:

In relation R (A,B,C,D,E) ; Given set of FDs S.

If E does not appear on RHS of any FD in S,

then who can determine E??

$\boxed{\text{Only E can Determine E}}$

E must be Part of Every CK.

NOTE:

In relation R (A,B,C,D,E) ; Given set of FDs S.

If E does not appear on RHS of any FD in S, then

E must be part of EVERY Candidate Key.

NOTE: To Find Candidate keys:

→ minimal sk

Go step by step.

First find all Single Attribute CKs, then all two Attributes CKs, then all three attributes CKs, & so on...

NOTE: To Find Superkeys

First find ALL candidate keys.. Then every superset of any CK is a SK.

[redacted]
relation schema $r(A, B, C, D, E)$.

$$A \rightarrow BC$$
$$CD \rightarrow E$$
$$B \rightarrow D$$
$$E \rightarrow A$$

List the candidate keys for R.

Source: Korth

Step 1: find single attribute CKs.

[redacted] relation schema $r(A, B, C, D, E)$.

$A \rightarrow BC$
$CD \rightarrow E$
$B \rightarrow D$
$E \rightarrow A$

$B^+ = BD$; B : NOT CK

$C^+ = C$; C : NOT CK

$D^+ = D$; D : NOT CK

$A^+ = ABCDE$

$\underbrace{A^+ = R}$

A : SK & minimal

List the candidate keys for R.

$E^+ = EABCD \Rightarrow E$: SK & minimal

So; A : CK ✓

E : CK

So far; CKs: A, E ⇒ So; No other CK Can Contain A or E.

relation schema $r(A, B, C, D, E)$.

$(BD)^+ = BD$

So; BD: NOT CK

$$A \to BC$$
$$CD \to E$$
$$B \to D$$
$$E \to A$$

So; forget A, E

Now; find Two attributes CKs:

List the candidate keys for R.

$(BC)^+ = BCDEA = R$

BC: minimal SK

$(CD)^+ = CDEAB = R$

CD: SK & minimal ⇒ CD is a CK.

CKs: A, E, BC, CD

relation schema $r(A, B, C, D, E)$.

find three attributes CKs:

$$A \rightarrow BC$$
$$CD \rightarrow E$$
$$B \rightarrow D$$
$$E \rightarrow A$$

Can't Take A or E

List the candidate keys for R.

Can Not be a CK because BC is CK.

BCD : SK but NOT minimal

5.5.34 Database Normalization: GATE CSE 2016 Set 1 | Question: 21

Which of the following is NOT a superkey in a relational schema with attributes V, W, X, Y, Z and primary key $V\ Y$?

A. $VXYZ$ B. $VWXZ$ C. $VWXY$ D. $VWXYZ$

gatecse-2016-set1 databases database-normalization easy

Sk: a superset of some candidate key

5.5.34 Database Normalization: GATE CSE 2016 Set 1 | Question: 21

Which of the following is NOT a superkey in a relational schema with attributes V, W, X, Y, Z and primary key $V\,Y$?

A. $VXYZ$ B. $VWXZ$ C. $VWXY$ D. $VWXYZ$

gatecse-2016-set1 databases database-normalization easy

Sk

only one Ck : Vy

Sk Sk

5.5.52 Database Normalization: GATE IT 2006 | Question: 60

Consider a relation R with five attributes V, W, X, Y, and Z. The following functional dependencies hold: $VY \rightarrow W, WX \rightarrow Z$, and $ZY \rightarrow V$.

Which of the following is a candidate key for R?

A. VXZ B. VXY C. $VWXY$ D. $VWXYZ$

gateit-2006 databases database-normalization normal

X, Y Do NOT appear on RHS of any given FD. So; only XY → XY

So; XY must be part of Every CK.

$(XY)^+ = XY$

option A: VXY Not even a SK.

5.5.52 Database Normalization: GATE IT 2006 | Question: 60

Consider a relation R with five attributes $V, W, X, Y,$ and Z. The following functional dependencies hold: $VY \to W, WX \to Z,$ and $ZY \to V$.

Which of the following is a candidate key for R? → XYV, XYZ, XYW

A. VXZ ✗

B. VXY ✓

C. $VWXY$ ✗ → SK

D. $VWXYZ$ ✗ → SK

gateit-2006 databases database-normalization normal

$XY \to XY$

Trivially

Step by Step : XY : Part of Every CK

(1) $\boxed{XY + \text{one more attribute}}$

$(XYV)^+ = XYVWZ = R \Rightarrow$ XYV : a CK

minimal SK

$(XYW)^+ = XYWZV = R \Rightarrow$ XYW : a CK

$(XYZ)^+ = XYZVW = R \Rightarrow$ (XYZ) : a CK

(2) $\underbrace{XY + \text{Two more attributes}}$: No such CK Possible.

XYVW : NOT a CK because

$\underbrace{XYV, XYW, XYZ}_{}$

CKs

5.5.42 Database Normalization: GATE CSE 2022 | Question: 21

Consider a relation $R(A, B, C, D, E)$ with the following three functional dependencies.

$AB \rightarrow C;\ BC \rightarrow D;\ C \rightarrow E;$

The number of superkeys in the relation R is ______________ .

gatecse-2022 numerical-answers databases database-normalization 1-mark

① Find all CKs.

→ A, B Do not appear on RHS of any given FD

5.5.42 Database Normalization: GATE CSE 2022 | Question: 21

Consider a relation $R(A, B, C, D, E)$ with the following three functional dependencies.

$AB \rightarrow C$; $BC \rightarrow D$; $C \rightarrow E$;

So; A, B must be part of Every CK.

The number of superkeys in the relation R is 8 ________.

gatecse-2022 numerical-answers databases database-normalization 1-mark

$(AB)^+ = ABCDE = R \Rightarrow$ So; AB is the only CK.

R(A B C D E)

Only CK : AB

Superkey : a superset of some CK.

Here; SK : $\boxed{\text{any superset of AB}}$

$2^3 = 8$

R(ABCDE)

Only CK : AB

Supersets of AB

SK = AB( )

any subset of C, D, E

2^3 subsets of CDE

Q: True Or False??

If $A \rightarrow B$ and $BC \rightarrow D$ then $AC \rightarrow D$??

Q: True Or False??

If $A \rightarrow B$ and $BC \rightarrow D$ then $AC \rightarrow D$??

Given

To check

$(AC)^{+} = ACBD$

$AC \rightarrow D$

Q:

Which of the following is/are correct rules in the context of functional dependencies over a database?

A. If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ always holds.
B. If $\alpha \rightarrow \beta$ holds, and γ is a set of attributes, then $\gamma\alpha \rightarrow \gamma\beta$ holds.
C. If $\alpha \rightarrow \beta$ holds, and $\gamma\beta \rightarrow \delta$ holds, then $\alpha\gamma \rightarrow \delta$ holds.
D. If $\alpha \rightarrow \beta$ holds, and $\beta \rightarrow \gamma$ holds, then $\alpha \rightarrow \gamma$ holds.

Q:

Which of the following is/are correct rules in the context of functional dependencies over a database?

A. If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \to \beta$ always holds. Trivial FD (Reflexivity Law)

B. If $\alpha \to \beta$ holds, and γ is a set of attributes, then $\gamma\alpha \to \gamma\beta$ holds. → Augmentation

C. If $\alpha \to \beta$ holds, and $\gamma\beta \to \delta$ holds, then $\alpha\gamma \to \delta$ holds.

[redacted]

Given

To check

$$(\alpha\gamma)^+ = \alpha\gamma\beta\delta \Longrightarrow \alpha\gamma \to \delta$$

Q:

Which of the following is/are correct rules in the context of functional dependencies over a database?

✓ A. If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \to \beta$ always holds. Trivial FD (Reflexivity Law)

✓ B. If $\alpha \to \beta$ holds, and γ is a set of attributes, then $\gamma\alpha \to \gamma\beta$ holds. → Augmentation

✓ C. If $\alpha \to \beta$ holds, and $\gamma\beta \to \delta$ holds, then $\alpha\gamma \to \delta$ holds.

✓ D. If $\alpha \to \beta$ holds, and $\beta \to \gamma$ holds, then $\alpha \to \gamma$ holds. Transitivity Rule

$$\alpha \longrightarrow \beta \longrightarrow \gamma$$

The Relational Model

Functional Dependency:

Inferred FDs (Inference)

Infer == Derive == Imply == Follow

Example: FD Set F = { A -> B ; B -> C }

From F, can we Infer that FD A -> C holds ??

Infer == Derive == Imply == Follow

Example: FD Set F = { A -> B ; B -> C }

From F, can we Infer that FD A -> C holds ??

YES

$A \to C$

Infer == Derive

Example: FD Set F = { A -> B ; B -> C }

From F, can we Infer that FD C -> A holds ??

Infer == Derive

Example: FD Set F = { A -> B ; B -> C }

From F, can we Infer that FD C -> A holds ??

NO

$C^+ = C \Rightarrow$ So; $C \not\rightarrow A$

Inferring FD's

- We are given FD's

  $X_1 \text{->} A_1, X_2 \text{->} A_2, \dots, X_n \text{->} A_n$,

  and we want to know whether an FD $Y \text{->} B$ must hold in any relation that satisfies the given FD's.

  – Example: If $A \text{->} B$ and $B \text{->} C$ hold, surely $A \text{->} C$ holds, even if we don't say so.

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.

We wish to test whether AB→D follows from the set of FD's?

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.

We wish to test whether AB→D follows from the set of FD's? YES

$(AB)^+ = ABCDE \longrightarrow AB \rightarrow D$

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.

We wish to test whether AB→D follows from the set of FD's?

We compute $\{A,B\}^+$ which is {A,B,C,D,E}.

Since D is a member of the closure, we conclude that it follows.

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.
  - We wish to test whether D→A follows from the set of FD's?

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.
  - We wish to test whether D→A follows from the set of FD's? No

$(D)^+ = DE$

$D \not\to A$

- Suppose R(A,B,C,D,E,F) and the
  - FD's AB→C, BC→AD, D→E, and CF→B satisfy.
  - We wish to test whether D→A follows from the set of FD's?

We compute $\{D\}^+$ first.

We start from X={D}.

From D→E, add E to the set. Now X= {D,E}.

We are stuck and no other FD's you can find that the left side is in X.

Since A is not in the list, so, D→A doesn't follow.

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow T$

Which of the following functional dependencies can be inferred from the above functional dependencies?

A. $PS \rightarrow T$
B. $R \rightarrow T$
C. $P \rightarrow R$
D. $PS \rightarrow Q$

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \to QR$
- $RS \to T$

$\Rightarrow (PS)^+ = PSQRT \Rightarrow PS \to T$

Which of the following functional dependencies can be inferred from the above functional dependencies?

A. $PS \to T$

[redacted]

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow T$

$\Rightarrow$ $R^+ = R$ $\Rightarrow$ $R \not\rightarrow T$

Which of the following functional dependencies can be inferred from the above functional dependencies?

[redacted]

B. $R \rightarrow T$ No

[redacted]

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow T$

: $P^+ = PQR \Rightarrow P \rightarrow R$

Which of the following functional dependencies can be inferred from the above functional dependencies?

[redacted]

C. $P \rightarrow R$ ✓

[redacted]

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow T$

: $(PS)^+ = PSQRT \Rightarrow PS \rightarrow Q$

Which of the following functional dependencies can be inferred from the above functional dependencies?

[redacted]

D. $PS \rightarrow Q$ ✓

GATE CSE 2021 Set 2 | Question: 40

asked Feb 18, 2021 • retagged Nov 30, 2022 by **Lakshman Bhaiya**

8,501 views

19

Suppose the following functional dependencies hold on a relation U with attributes P, Q, R, S, and T:

- $P \rightarrow QR$
- $RS \rightarrow T$

Which of the following functional dependencies can be inferred from the above functional dependencies?

A. $PS \rightarrow T$
B. $R \rightarrow T$
C. $P \rightarrow R$
D. $PS \rightarrow Q$

Example

Consider the relation R(A,B,C,D,E) and the set of functional dependencies S1 = {AB → C, AE → D, D → B}.

Which of the following sets S2 of FDs does NOT follow from S1?

- ○ S2 = {AD → C}
- ○ S2 = {AD → C, AE → B}
- ○ S2 = {ABC → D, D → B}
- ○ S2 = {ADE → BC}

Skip

Submit

Video Solution Link in the Description!!

The Relational Model

Functional Dependency:

Closure of a FD set

Q:

Are these FDs SAME?

$A \rightarrow B$; $A \rightarrow AB$

Q:

Are these FDs SAME? NO.

$A \rightarrow B$; $A \rightarrow AB$

$A \rightarrow B$ means "If two tuples have same A value then they have same B value".

$A \rightarrow AB$ means "If two tuples have same A value then they have same AB value".

$$A \to B \xrightarrow{\text{infer}} A \to AB$$

$$A \to AB \xrightarrow{\text{infer}} A \to B$$

But they are "Different FDs" which infer each other.

Closure of a FD Set:

For a given FD set S ; the Set of ALL FDs that can be inferred by S is called the Closure of S.

Definition. Formally, the set of all dependencies that include F as well as all dependencies that can be inferred from F is called the **closure** of F; it is denoted by F^+.

Note: Relation R ; FD set F

F^+ = set of ALL FDs that hold in R

15.2 FUNCTIONAL DEPENDENCIES

A **functional dependency** (FD) is a kind of IC that generalizes the concept of a *key*. Let R be a relation schema and let X and Y be nonempty sets of attributes in R. We say that an instance r of R satisfies the FD $X \rightarrow Y$ [1] if the following holds for every pair of tuples t_1 and t_2 in r:

$X \rightarrow Y$

Nonempty

Nonempty set of attributes

Example. Assume that there are 4 attributes A, B, C, D, and that $F = \{A \to B, B \to C\}$. Then, F^+ includes all the following FDs:

$$F^+ = \left\{ \begin{array}{lll} A \to A & A \to C & BC \to B \\ A \to B & A \to AC & C \to C \\ \cancel{A \to D} & B \to BC & ABD \to BD \\ & B \to C & \\ \_\ \_\ \_\ \_\ \_\ \_\ \_ & & \\ \_\ \_\ \_\ \_\ \_ & & \end{array} \right\}$$

$$\left.\begin{aligned} AB &\rightarrow C \\ BA &\rightarrow C \end{aligned}\right\}$$ Same

$\{A, B\} \rightarrow \{C\}$

$\{B, A\} = \{A, B\}$

$$\left.\begin{aligned} A &\rightarrow B \\ A &\rightarrow AB \end{aligned}\right\}$$ Different FDs

$\{A\} \rightarrow \{B\}$

$\{A\} \rightarrow \{A, B\}$

Example. Assume that there are 4 attributes A, B, C, D, and that $F = \{A \to B, B \to C\}$. Then, F^+ includes all the following FDs:

$A \to A$, $A \to B$, $A \to C$, $B \to B$, $B \to C$, $C \to C$, $D \to D$, $AB \to A$, $AB \to B$, $AB \to C$, $AC \to A$, $AC \to B$, $AC \to C$, $AD \to A$, $AD \to B$, $AD \to C$, $AD \to D$, $BC \to B$, $BC \to C$, $BD \to B$, $BD \to C$, $BD \to D$, $CD \to C$, $CD \to D$, $ABC \to A$, $ABC \to B$, $ABC \to C$, $ABD \to A$, $ABD \to B$, $ABD \to C$, $ABD \to D$, $BCD \to B$, $BCD \to C$, $BCD \to D$, $ABCD \to A$, $ABCD \to B$, $ABCD \to C$, $ABCD \to D$.

$B \to BC$, $A \to ABC$; $A \to BC$, ----

NOTE:

For any FD set F ; F^+ is Unique.

set of ALL FDs that hold in R w.r.t. F.

The Relational Model

Functional Dependency:

Covering

FD set F

FD set G

F covers G iff F infers every FD of G.

$\boxed{\text{F covers G}}$ iff Every FD of G can be inferred from F.

$F \supseteq G$

Notation

15.1.2 Equivalence of Sets of Functional Dependencies

In this section, we discuss the equivalence of two sets of functional dependencies. First, we give some preliminary definitions.

Definition. A set of functional dependencies F is said to **cover** another set of functional dependencies E if every FD in E is also in F^+; that is, if every dependency in E can be inferred from F; alternatively, we can say that E is **covered by** F.

Q: Whom covers whom ??

A. $V \rightarrow W$

$V \rightarrow X$

$Y \rightarrow V$

$Y \rightarrow Z$

B. $V \rightarrow W$

$W \rightarrow X$

$Y \rightarrow V$

$Y \rightarrow Z$

Q: Whom covers whom ??

$$F = \left\{\begin{array}{l} V \to W \\ V \to X \\ Y \to V \\ Y \to Z \end{array}\right\} \qquad G = \left\{\begin{array}{l} V \to W \\ W \to X \\ Y \to V \\ Y \to Z \end{array}\right\}$$

Check: Does G Cover F ? YES

$V^+ = VWX$

Q: Whom covers whom ??

$$F = \left\{ \begin{array}{l} V \to W \\ V \to X \\ Y \to V \\ Y \to Z \end{array} \right\} \qquad G = \left\{ \begin{array}{l} V \to W \\ W \to X \\ Y \to V \\ Y \to Z \end{array} \right\}$$

Check: Does F Cover G ?? NO

In F: $W^+ = W$

Q: Whom covers whom ??

Q.4 Given the following functional dependency sets F and G:

$F = \{A \rightarrow B, AB \rightarrow C, D \rightarrow AC, D \rightarrow E\}$
$G = \{A \rightarrow BC, D \rightarrow AE, E \rightarrow B\}$

(a) [10 pts] Show that all the functional dependencies in F can be inferred from G (i.e., G covers F).

Q: Whom covers whom ??

Q.4 Given the following functional dependency sets F and G:

$$F = \{A \rightarrow B,\ AB \rightarrow C,\ D \rightarrow AC,\ D \rightarrow E\}$$

$$G = \{A \rightarrow BC,\ D \rightarrow AE,\ E \rightarrow B\}$$

✓ ✓ ✗

Check : Does F Cover G ? No

In F :

$A^+ = ABC$; $D^+ = DACE$

$E^+ = E$

Q: Whom covers whom ??

Q.4 Given the following functional dependency sets F and G:

F = {A → B, AB → C, D → AC, D → E}
G = {A → BC, D → AE, E → B}

Check : Does G Cover F ? YES

In G : $(AB)^+ = ABC$

$(D)^+ = DAEBC$

The Relational Model

Functional Dependency:

Equivalence of FD sets

FD Set F , FD Set G

$F \equiv G$ iff F Covers G & G Covers F.

$\boxed{F \equiv G \text{ iff } F^+ = G^+}$

(To solve Questions, we won't use this Definition)

$$F = \{A \to B, B \to C, A \to C, AB \to C, A \to BC\}$$

$$G = \{A \to B, B \to C\}$$

Is $F \equiv G$?

$$F = \{A \to B,\ B \to C,\ A \to C,\ AB \to C,\ A \to BC\}$$

$$G = \{A \to B,\ B \to C\}$$

(1) Does F cover G? Yes

$$F = \{A \to B,\ B \to C,\ A \to C,\ AB \to C,\ A \to BC\}$$

$$G = \{A \to B,\ B \to C\}$$

(2) Does G Cover F? yes

In G: $A^+ = ABC$

$(AB)^+ = ABC$

So; $F \equiv G$

$$F = \{A \to B,\ B \to C,\ A \to C,\ AB \to C,\ A \to BC\}$$

$$G = \{A \to B,\ B \to C\}$$

$$F \equiv G \iff \boxed{F^+ = G^+}$$

15.1.2 Equivalence of Sets of Functional Dependencies

In this section, we discuss the equivalence of two sets of functional dependencies. First, we give some preliminary definitions.

Definition. A set of functional dependencies F is said to **cover** another set of functional dependencies E if every FD in E is also in F^+; that is, if every dependency in E can be inferred from F; alternatively, we can say that E is **covered by** F.

Definition. Two sets of functional dependencies E and F are **equivalent** if $E^+ = F^+$. Therefore, equivalence means that every FD in E can be inferred from F, and every FD in F can be inferred from E; that is, E is equivalent to F if both the conditions—E covers F *and* F covers E—hold.

We can determine whether F covers E by calculating X^+ *with respect to* F for each FD $X \rightarrow Y$ *in* E, and then checking whether this X^+ includes the attributes in Y. If this is the case for *every* FD in E, then F covers E. We determine whether E and F are equivalent by checking that E covers F and F covers E. It is left to the reader as an exercise to show that the following two sets of FDs are equivalent:

$$F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$$
and $G = \{A \rightarrow CD, E \rightarrow AH\}$

exercise to show that the following two sets of FDs are equivalent:

$$F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$$
and $G = \{A \rightarrow CD, E \rightarrow AH\}$

HW

4 Functional Dependencies

A *functional dependency* is a constraint between two sets of attributes in a relation. For a given relation R, if the set of attributes $A_1, \cdots, A_n$, functionally determine the set of attributes $B_1, \cdots, B_n$, then it means that if two tuples that have the same values for all the attributes $A_1, \cdots, A_n$, then they must also have the same values for all the attributes $B_1, \cdots, B_n$. This is written:

$$A_1, \cdots, A_n \rightarrow B_1, \cdots, B_n$$

Note that functional dependencies generalize the concept of a *key*.

The Relational Model

Functional Dependency:

Minimal Cover

Next Topic:

Minimal Cover of FD Set

(minimal cover/ canonical cover/ irreducible FD Set)

For a given set of functional dependencies F:

$F = \{ \, , \, , \, , \, , \, , \, \}$

Expand

$F^+ = \{ \, , \, , \, , \, , \, , \, , \, , \, , \, , \, , \, , \, , \, , \, \}$

$\boxed{F \equiv F^+}$; for any FD set F

F^+ is Unique.

For a given set of functional dependencies F:

$F = \{ \; , \; , \; , \; , \; , \; , \; \}$

Shrink / Reduce

$F_m = \{ \; , \; , \; , \; \}$

minimal Cover of F / Canonical Cover of F / Irreducible FD set

Note:

$\boxed{F \equiv F_m}$; for F, minimal Cover may not be unique.

F can have many minimal Covers

minimal ⟷ Irreducible

Minimal Cover of Functional Dependencies:

For a given set of functional dependencies F:

Minimal Cover F_m of F is an **Irreducible Set of FDs** which is equivalent to F.

i.e. if we remove any FD or any attribute from F_m then it is No Longer equivalent to F.

15.1.3 Minimal Sets of Functional Dependencies

Just as we applied inference rules to expand on a set F of FDs to arrive at F^+, its closure, it is possible to think in the opposite direction to see if we could shrink or reduce the set F to its *minimal form* so that the minimal set is still equivalent to the original set F. Informally, a **minimal cover** of a set of functional dependencies E is a set of functional dependencies F that satisfies the property that every dependency in E is in the closure F^+ of F. In addition, this property is lost if any dependency from the set F is removed; F must have no redundancies in it, and the dependencies in F are in a standard form.

$$\left\{ \begin{array}{l} \bullet\ E \to G \\ \bullet\ E \to S \\ \bullet\ G \to S \end{array} \right\} \quad \left\{ \begin{array}{l} E \to G \\ G \to S \end{array} \right\}$$

NOT minimal

$\equiv$ Reducible

minimal

$\equiv$ Irreducible

$E \to G \to S$

FD set F → F_m

(1) Remove unnecessary attributes from LHS.

(2) Remove Redundant FDs

Next subtopic:

Extraneous Attribute on LHS of a FD

FD set $F = \{ XA \rightarrow Y, \; - - - - - - \}$

A: single attribute

'A' is Extraneous attribute in $XA \rightarrow Y$

iff $X \rightarrow Y$ in F

FD set $F = \{ XA \rightarrow Y, \text{- - - - - -} \}$

A: single attribute

'A' is Extraneous attribute in $XA \rightarrow Y$

iff X^+ Contains Y in F.

FD set $F = \{ XA \rightarrow Y, \ - - - \}$

A : simple attribute

If 'A' is Extraneous in $XA \rightarrow Y$ then We can Remove A.

For example, say we have the following functional dependencies (F):

- $AB \to C$
- $A \to B$

Is 'A' in $AB \to C$ Extraneous?

NO.

$B^+ = B$

For example, say we have the following functional dependencies (F):

- $AB \to C$
- $A \to B$

Is this Extraneous?

YES

$A^+ = ABC$

$\Rightarrow A \to C$

Extraneous

$$\left\{ \begin{array}{l} AB \rightarrow C \\ A \rightarrow B \end{array} \right\} \equiv \left\{ \begin{array}{l} A \rightarrow C \\ A \rightarrow B \end{array} \right\}$$

minimal Cover fm

Exercise:

Suppose a relational schema R(P, Q, R, S), and set of functional dependency as following

$$F : \{ P \rightarrow QR, \\ Q \rightarrow R, \\ P \rightarrow Q, \\ PQ \rightarrow R \}$$

Find Extraneous Attributes on LHS of given FDs.

Exercise:

Suppose a relational schema R(P, Q, R, S), and set of functional dependency as following

Is it Extraneous ?!. Yes

$$F : \{ P \rightarrow QR, \; Q \rightarrow R, \; P \rightarrow Q, \; PQ \rightarrow R \}$$

on LHS, simple attribute; nothing to check

In F :

$Q^+ = QR$

Exercise:

Suppose a relational schema R(P, Q, R, S), and set of functional dependency as following

Is it Extraneous?

YES

F : { P → QR,

Q → R,

P → Q,

PQ → R }

on LHS, single attribute; nothing to check

In F; $P^+ = PQR$

Exercise:

Suppose a relational schema R(P, Q, R, S), and set of functional dependency as following

$P \rightarrow QR$
$Q \rightarrow R$
$P \rightarrow Q$
$P \rightarrow R$

$\equiv$

$$F : \{ P \rightarrow QR,\; Q \rightarrow R,\; P \rightarrow Q,\; PQ \rightarrow R \}$$

$\equiv$

$$\{ P \rightarrow QR,\; Q \rightarrow R,\; P \rightarrow Q,\; Q \rightarrow R \}$$

A canonical cover might not be unique. For instance, consider the set of functional dependencies F , If we apply

Korth

the extraneity test to we find that both B and C are extraneous under F. However, it is incorrect to delete both! The algorithm for finding the canonical cover picks one of the two, and deletes it.

An attribute of a functional dependency is said to be **extraneous** if we can remove it without changing the closure of the set of functional dependencies. The formal definition of **extraneous attributes** is as follows: Consider a set F of functional dependencies and the functional dependency $\alpha \rightarrow \beta$ in F.

- If $A \in \alpha$, to check if A is extraneous, let $\gamma = \alpha - \{A\}$, and check if $\gamma \rightarrow \beta$ can be inferred from F. To do so, compute γ^+ (the closure of γ) under F; if γ^+ includes all attributes in β, then A is extraneous in α.

Testing if an Attribute is Extraneous

- Consider a set F of functional dependencies and the functional dependency $\alpha \rightarrow \beta$ in F.
- To test if attribute $A \in \alpha$ is extraneous in α
  1. compute $(\{\alpha\} - A)^+$ using the dependencies in F
  2. check that $(\{\alpha\} - A)^+$ contains A; if it does, A is extraneous

- $V \to W$
- $VW \to X$
- $Y \to VX$
- $Y \to Z$

Is ‘V’ Extraneous Attribute in VW -> X ??

$$F \left\{ \begin{array}{l} \bullet\ V \rightarrow W \\ \bullet\ VW \rightarrow X \\ \bullet\ Y \rightarrow VX \\ \bullet\ Y \rightarrow Z \end{array} \right\}$$

Is 'V' Extraneous Attribute in VW -> X ?? No

In F; $W^+ = W$

- $V \rightarrow W$
- $VW \rightarrow X$
- $Y \rightarrow VX$
- $Y \rightarrow Z$

Is 'W' Extraneous Attribute in VW -> X ??

$$F = \left\{ \begin{array}{l} \bullet\ V \to W \\ \bullet\ V\underline{W} \to X \\ \bullet\ Y \to VX \\ \bullet\ Y \to Z \end{array} \right\}$$

Is '$\underline{W}$' Extraneous Attribute in VW -> X ?? Yes

In F; $V^+ = VW\underline{X}$

So; we can Remove W.

$R = \{A,B,C,D,E,F,G,H\}$

$F = \{AC \to G,\ D \to EG,\ BC \to D,\ CG \to BD,\ ACD \to B,\ CE \to AG\}$

Is 'D' Extraneous Attribute in ACD -> B ??

$R = \{A,B,C,D,E,F,G,H\}$

$F = \{AC \to G,\ D \to EG,\ BC \to D,\ CG \to BD,\ ACD \to B,\ CE \to AG\}$

Is 'D' Extraneous Attribute in ACD -> B ?? Yes

In f;

$(AC)^+ = ACGBDE$

Next subtopic:

Removing a Redundant FD

Removing a Redundant FD:

For a given set of functional dependencies F:

First Make Sure that RHS of Every FD has Single Attribute.

(This is done to Simplify the process)

$ABC \longrightarrow EFG$

then

$$\left.\begin{array}{l} ABC \rightarrow E \\ ABC \rightarrow F \\ ABC \rightarrow G \end{array}\right\}$$

FD set F = { $X \to A$, - - - - }

A : single attribute

FD $X \to A$ is Redundant

iff

$\boxed{F - \{X \to A\}}$

infer $X \to A$

FD set = { $X \rightarrow A$, - - - - }

infer

A : simple attribute

then $X \rightarrow A$ is Redundant & can be Removed.

Removing a Redundant FD:

For a given set of functional dependencies F:

First Make Sure that RHS of Every FD has Single Attribute.

For each functional dependency $X \to A$ in F

if $\{F - \{X \to A\}\}$ is equivalent to F,

then remove $X \to A$ from F. (*This constitutes removal of a redundant functional dependency $X \to A$ from F when possible*)

$$F1 = \{ AC \rightarrow G, \; CD \rightarrow B, \; BC \rightarrow D, \; CG \rightarrow B, \; CG \rightarrow D, \; CE \rightarrow A, \; CE \rightarrow G, \; D \rightarrow E, \; D \rightarrow G \}$$

Find

Redundant FDs.

F1 = { $\boxed{A\ C \rightarrow G}$

C D → B

B C → D

C G → B

C G → D

C E → A

C E → G

D → E

D → G }

Is this Redundant ?

No

$\Rightarrow (AC)^+ = AC$

$$F1 = \{ AC \to G,\ \boxed{CD \to B},\ BC \to D,\ CG \to B,\ CG \to D,\ CE \to A,\ CE \to G,\ D \to E,\ D \to G \}$$

Is it Redundant ?

YES

find

$(CD)^+ = ?$

$(CD)^+ = CDEGB$

$F1 = \{AC \rightarrow G$

[redacted]

$BC \rightarrow D$

$CG \rightarrow B$

$CG \rightarrow D$

$CE \rightarrow A$

$CE \rightarrow G$

$D \rightarrow E$

$D \rightarrow G\}$

Is it Redundant?

No

$(BC)^+ = ?$

$(BC)^+ = BC$

$F1 = \{ AC \to G$

[redacted]

$BC \to D$

$CG \to B$

$\boxed{CG \to D}$

Redundant

$CE \to A$

$CE \to G$

$D \to E$

$D \to G \}$

Is it Redundant?

YES

$(CG)^+$ in Remaining FDs

$(CG)^+ = CGBD$

Next topic:

Finding Minimal Cover of FDs

Minimal Cover Algo

① Remove all Trivial FDs.

② Make sure All FDs are Completely Non-Trivial.

Eg: $AB \rightarrow BC$

$\boxed{AB \rightarrow B}$ Trivial → Remove

$AB \rightarrow C$

Minimal Cover Algo:

3. Remove Extraneous attributes on LHS of FDs.
4. Now; make RHS of Every FD single attribute & Remove Redundant FDs
5. We got minimal Cover.

Procedure to find the Minimal Cover:

For a given set of functional dependencies:

1. Simplify all RHS (Decomposition)
2. For all FDs on LHS find a redundant (extraneous) attribute
3. Eliminate all redundant FDs

We can formally define a set of functional dependencies F to be **minimal** if it satisfies the following conditions:

1. Every dependency in F has a single attribute for its right-hand side.
2. We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F.
3. We cannot remove any dependency from F and still have a set of dependencies that is equivalent to F.

nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/Alevel_1_DBMS_05May2020_AV.pdf

Alevel_1_DBMS_05May2020_AV.pdf 1 / 3 153%

Canonical Cover - Exercise Practices

Q. Suppose a relational schema R(w x y z), and set of functional dependency as following

$$
\begin{aligned}
F : \{\ & x \rightarrow w, \\
& wz \rightarrow xy, \\
& y \rightarrow wxz\ \}
\end{aligned}
$$

Find the canonical cover F_c (Minimal set of functional dependency).

Canonical Cover - Exercise Practices

Q. Suppose a relational schema R(w x y z), and set of functional dependency as following

Every FD is Completely Non-Trivial.

$$F : \{\, x \rightarrow w,\; wz \rightarrow xy,\; y \rightarrow wxz \,\}$$

w is Extra?!! NO

$z^+ = z$

z is Extra?!! NO

Find the canonical cover F_c ([redacted]

$w^+ = w$

$X \rightarrow W$: Is it Redundant? **No**

$WZ \rightarrow X$

$WZ \rightarrow Y$

$Y \rightarrow W$

$Y \rightarrow X$

$Y \rightarrow Z$

$\Rightarrow X^+ = X$

$X \rightarrow W$

$\boxed{WZ \rightarrow X}$ → Is it Redundant ? Yes

$WZ \rightarrow Y$

$Y \rightarrow W$

$Y \rightarrow X$

$Y \rightarrow Z$

$\boxed{(WZ)^+ = WZYX}$

→ In Remaining FDs find $(WZ)^+$.

$X \rightarrow W$

$WZ \rightarrow Y$ → NOT Redundant

$\boxed{Y \rightarrow W}$ → Redundant $(Y^+ = YXZW)$

$Y \rightarrow X$

$Y \rightarrow Z$

$X \rightarrow W$

$WZ \rightarrow Y$

$\boxed{Y \rightarrow X}$ → NOT Redundant

$\boxed{Y \rightarrow Z}$ → NOT Redundant

→ minimal Cover of F.

Fm :

minimal Cover:

$X \rightarrow W$

$WZ \rightarrow Y$

$Y \rightarrow XZ$

A canonical cover might not be unique. For instance, consider the set of functional dependencies $F = \{A \rightarrow BC,\ B \rightarrow AC,$ and $C \rightarrow AB\}$. If we apply

$$F : \left\{ \begin{array}{l} A \rightarrow BC \\ B \rightarrow AC \\ C \rightarrow AB \end{array} \right\}$$

F : $\boxed{A \rightarrow B}$ → Redundant ?? YES

$A \rightarrow C$

$B \rightarrow A$

$B \rightarrow C$

$C \rightarrow A$

$C \rightarrow B$

$\Rightarrow A^+ = ACB$

F :

$\boxed{A \rightarrow C}$ NOT Redundant

$\boxed{B \rightarrow A}$ Redundant ? YES.

$B \rightarrow C$

$C \rightarrow A$

$C \rightarrow B$

$B^+ = BCA$

F :

$$\left.\begin{array}{l} \checkmark\ A \rightarrow C \\ \checkmark\ B \rightarrow C \\ \checkmark\ C \rightarrow A \\ \checkmark\ C \rightarrow B \end{array}\right\}$$

Minimal Cover $\equiv$ Irreducible Set of FDs

F:

$A \rightarrow B$

$A \rightarrow C$

$B \rightarrow A$

$B \rightarrow C$

$C \rightarrow A$

$C \rightarrow B$

find Another minimal Cover.

F :

$A \to B$

$\boxed{A \to C}$ → Redundant

$B \to A$

$B \to C$

$C \to A$

$C \to B$

In Remaining FDs

$A^+ = ABC$

F :

$A \rightarrow B$

$B \rightarrow A$ → Redundant ? YES

$B \rightarrow C$

$C \rightarrow A$

$C \rightarrow B$

$B^+ = BCA$

F :

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

$\rightarrow C^+ = CAB$

$C \rightarrow B \rightarrow$ Redundant ? YES

F :

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow A$

minimal Cover of F

$\equiv$ Irreducible Set of FDs

minimal $\neq$ minimum

minimal $\equiv$ Irreducible

A canonical cover might not be unique. For instance, consider the set of functional dependencies $F = \{A \rightarrow BC, B \rightarrow AC, \text{and } C \rightarrow AB\}$. If we apply

$$F_c = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$$
$$F_c = \{A \rightarrow B, B \rightarrow AC, C \rightarrow B\}.$$

$$F_c = \{A \rightarrow C, C \rightarrow B, \text{and } B \rightarrow A\}$$
$$F_c = \{A \rightarrow C, B \rightarrow C, \text{and } C \rightarrow AB\}.$$

Source: Korth Book

NOTE:

We can have multiple Minimal Covers, &

different minimal covers may have different

number of FDs & sizes.

Non-Sense Question:

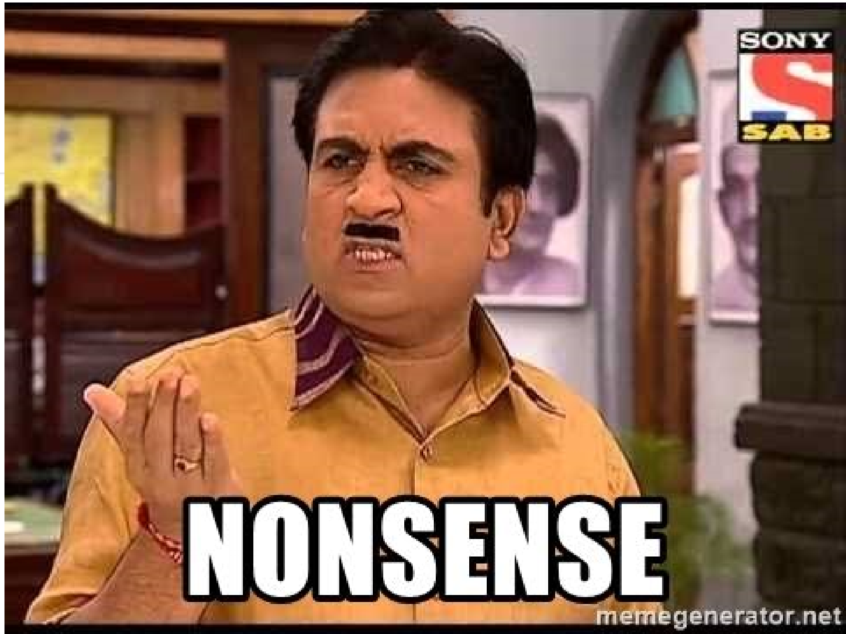
minimal cover

asked in Databases Jan 2, 2016 • retagged Aug 5, 2017 by Arjun

1

Let F = {$PQ \rightarrow R$, $PR \rightarrow Q$, $Q \rightarrow S$, $QR \rightarrow P$, $PQ \rightarrow T$} where F is set of FD's of a relation R(PQRST). The number of functional dependencies are in minimal cover of F is ________.

databases minimal-cover database-normalization



If several sets of FDs qualify as minimal covers of E by the definition above, it is customary to use additional criteria for *minimality*. For example, we can choose the minimal set with the *smallest number of dependencies* or with the smallest *total length* (the total length of a set of dependencies is calculated by concatenating the dependencies and treating them as one long character string).

Source: Navathe

Another Non-Sense Question:

is canonical cover and minimal cover the same thing?

asked in **Databases** Jan 7, 2015 • edited Jul 27, 2017 by **Arjun** 5,723 views

9

Is canonical cover and minimal cover the same thing?

edit flag close hide answer comment Follow

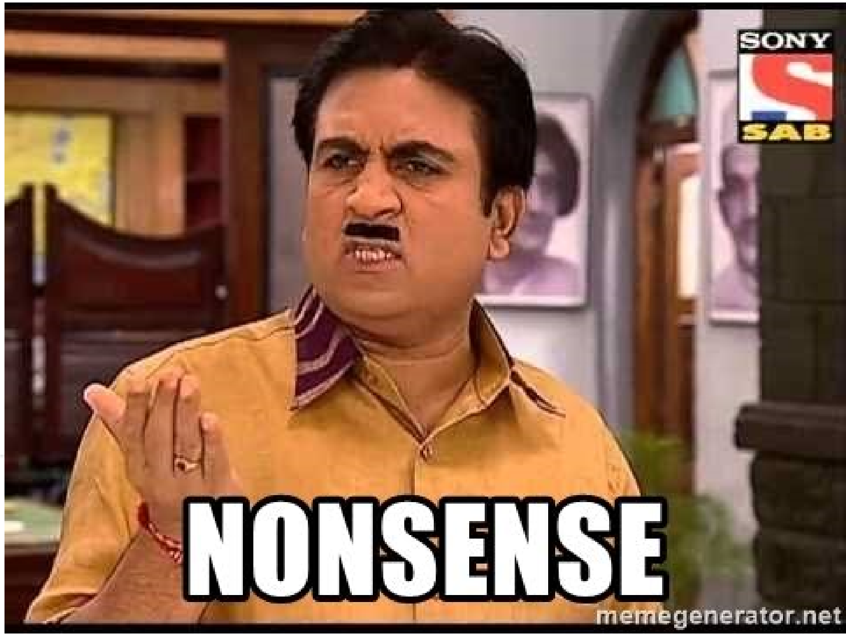
Pip Box Lists Mark Read Feature Hide Answers

Delete with Reason Bad

share this

4 Comments

Show 5 previous comments



Definition. A **minimal cover** of a set of functional dependencies E is a minimal set of dependencies (in the standard canonical form[5] and without redundancy) that is equivalent to E. We can always find *at least one* minimal cover F for any set of dependencies E using Algorithm 15.2.

---

[4] This is a standard form to simplify the conditions and algorithms that ensure no redundancy exists in F. By using the inference rule IR4, we can convert a single dependency with multiple attributes on the right-hand side into a set of dependencies with single attributes on the right-hand side.

[5] It is possible to use the inference rule IR5 and combine the FDs with the same left-hand side into a single FD in the minimum cover in a nonstandard form. The resulting set is still a minimum cover, as illustrated in the example.

Source: Navathe

minimal Cover:

$A \rightarrow C$
$B \rightarrow A$
$B \rightarrow D$

$\equiv$

$A \rightarrow C$
$B \rightarrow AD$

fm

minimal Cover / Canonical Cover

Definition. A **minimal cover** of a set of functional dependencies E is a minimal set of dependencies (in the standard canonical form[5] and without redundancy) that is equivalent to E. We can always find *at least one* minimal cover F for any set of dependencies E using Algorithm 15.2.

---

[4]This is a standard form to simplify the conditions and algorithms that ensure no redundancy exists in F. By using the inference rule IR4, we can convert a single dependency with multiple attributes on the right-hand side into a set of dependencies with single attributes on the right-hand side.

[5]It is possible to use the inference rule IR5 and combine the FDs with the same left-hand side into a single FD in the minimum cover in a nonstandard form. The resulting set is still a minimum cover, as illustrated in the example.

Source: Navathe

5.5.36 Database Normalization: GATE CSE 2017 Set 1 | Question: 16

The following functional dependencies hold true for the relational schema $R\{V, W, X, Y, Z\}$:

- $V \rightarrow W$
- $VW \rightarrow X$
- $Y \rightarrow VX$
- $Y \rightarrow Z$

Which of the following is irreducible equivalent for this set of functional dependencies?

A. $V \rightarrow W$
$V \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow Z$

B. $V \rightarrow W$
$W \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow Z$

C. $V \rightarrow W$
$V \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow X$
$Y \rightarrow Z$

D. $V \rightarrow W$
$W \rightarrow X$
$Y \rightarrow V$
$Y \rightarrow X$
$Y \rightarrow Z$

5.5.36 Database Normalization: GATE CSE 2017 Set 1 | Question: 16

The following functional dependencies hold true for the relational schema $R\{V, W, X, Y, Z\}$:

- $V \to W$
- $VW \to X$
- $Y \to VX$
- $Y \to Z$

$\underline{Y \to X}$

$\underline{\text{Valid}}$ But $\underline{\underline{\text{Redundant}}}$

Which of the following is irreducible equivalent for this set of functional dependencies?

A. $V \to W$
$V \to X$
$Y \to V$
$Y \to Z$

B. $V \to W$
$W \to X$
$Y \to V$
$Y \to Z$

C. $V \to W$
$V \to X$
$Y \to V$
$\boxed{Y \to X}$
$Y \to Z$

D. $V \to W$
$W \to X$
$Y \to V$
$Y \to X$
$Y \to Z$

5.5.36 Database Normalization: GATE CSE 2017 Set 1 | Question: 16

The following functional dependencies hold true for the relational schema $R\{V,W,X,Y,Z\}$:

F

- $V \to W$
- $VW \to X$
- $Y \to VX$
- $Y \to Z$

$\Rightarrow$ $W^+ = W$

$F_m \equiv F$

$\equiv$ minimal cover

Which of the following is irreducible equivalent for this set of functional dependencies?

A. $V \to W$
$V \to X$
$Y \to V$
$Y \to Z$

B. $V \to W$
$\boxed{W \to X}$
$Y \to V$
$Y \to Z$

C. $V \to W$
$V \to X$
$Y \to V$
$Y \to X$
$Y \to Z$

D. $V \to W$
$\boxed{W \to X}$
$Y \to V$
$Y \to X$
$Y \to Z$

$V \rightarrow W$

$VW \rightarrow X$

$Y \rightarrow VX$

$Y \rightarrow Z$

Is V Extraneous? **No**

$W^+ = W$

Is W Extraneous? **Yes**

$V^+ = VWX$

$V \rightarrow W$ → Is it Redundant? No

$V \rightarrow X$

$Y \rightarrow V$

$Y \rightarrow Z$

$Y \rightarrow X$

$\Rightarrow V^+ = VX$

$V \rightarrow W$

$V \rightarrow X$ Redundant? No

$Y \rightarrow V$

$Y \rightarrow Z$

$Y \rightarrow X$

$V^{+} = VW$

$V \rightarrow W$

$V \rightarrow X$

$\boxed{Y \rightarrow V} \rightarrow$ Redundant ?

$Y^+ = YZX$

No

$Y \rightarrow Z$

$Y \rightarrow X$

$V \to W$

$V \to X$

$Y \to V$

$\boxed{Y \to Z}$ → Redundant ? No

$\boxed{Y \to X}$ → Redundant ? YES

$Y^+ = YZVXW$

$V \rightarrow W$

$V \rightarrow X$

$Y \rightarrow V$

$Y \rightarrow Z$

minimal Cover

5.5.33 Database Normalization: GATE CSE 2015 Set 3 | Question: 20

Consider the relation $X(P, Q, R, S, T, U)$ with the following set of functional dependencies

$$F = \{$$
$$\{P, R\} \rightarrow \{S, T\},$$
$$\{P, S, U\} \rightarrow \{Q, R\}$$
$$\}$$

Which of the following is the trivial functional dependency in F^+, where F^+ is closure to F?

A. $\{P, R\} \rightarrow \{S, T\}$

B. $\{P, R\} \rightarrow \{R, T\}$

C. $\{P, S\} \rightarrow \{S\}$

D. $\{P, S, U\} \rightarrow \{Q\}$

gatecse-2015-set3 databases database-normalization easy

5.5.33 Database Normalization: GATE CSE 2015 Set 3 | Question: 20

Consider the relation $X(P, Q, R, S, T, U)$ with the following set of functional dependencies

$$F = \{ \\ \{P, R\} \rightarrow \{S, T\}, \\ \{P, S, U\} \rightarrow \{Q, R\} \\ \}$$

option c : Trivial

Remaining options : NonTrivial

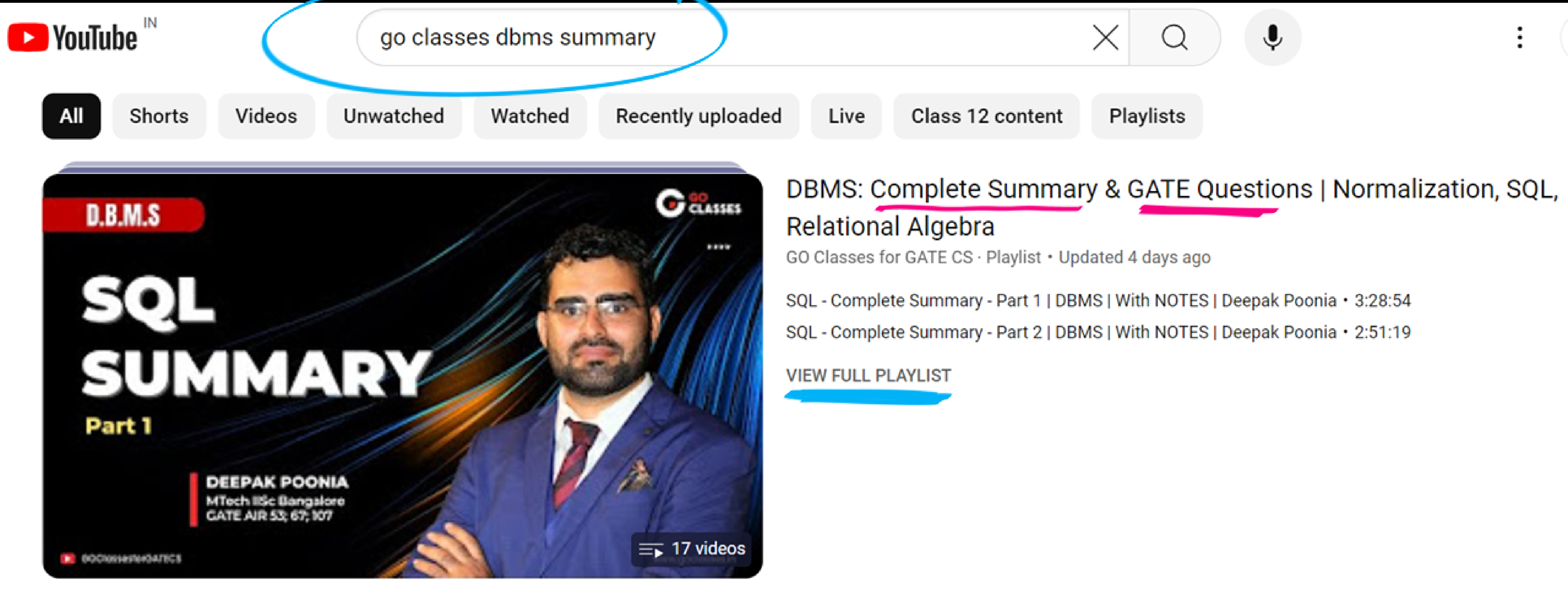
Which of the following is the trivial functional dependency in F^+, where F^+ is closure to F?

A. $\{P, R\} \rightarrow \{S, T\}$
B. $\{P, R\} \rightarrow \{R, T\}$
C. $\{P, S\} \rightarrow \{S\}$ ✓
D. $\{P, S, U\} \rightarrow \{Q\}$

gatecse-2015-set3 databases database-normalization easy

$X \to Y$ is Trivial iff $X \supseteq Y$

$X \to Y$ is NonTrivial if it is NOT Trivial.


YouTube IN
go classes dbms summary
All
Shorts
Videos
Unwatched
Watched
Recently uploaded
Live
Class 12 content
Playlists
D.B.M.S
GO CLASSES
SQL SUMMARY
Part 1
DEEPAK POONIA
MTech IISc Bangalore
GATE AIR 53; 67; 107
17 videos
DBMS: Complete Summary & GATE Questions | Normalization, SQL, Relational Algebra
GO Classes for GATE CS · Playlist • Updated 4 days ago
SQL - Complete Summary - Part 1 | DBMS | With NOTES | Deepak Poonia • 3:28:54
SQL - Complete Summary - Part 2 | DBMS | With NOTES | Deepak Poonia • 2:51:19
VIEW FULL PLAYLIST