Design Fundamentals

Dr. Sambit Bakshi

NIT Rourkela

August 28, 2023

Outline

Relational Database Design

Redundant data lead to following anomalies in database:

- Insert Anamolies
- Update Anamolies
- Deletion Anamolies

Redundancy is often caused by a functional dependency present in the relation.

Functional Dependencies

Functional Dependency:
A functional dependency, denoted by $X \longrightarrow Y$, between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation state r of R. The constraint is that, for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, they must also have $t_1[Y] = t_2[Y]$.

Armstrong's axioms:

- **Reflexivity rule**: If X is a set of attributes and $Y \subseteq X$,then $X \longrightarrow Y$ holds.
- **Augmentation rule**: If $X \longrightarrow Y$ holds and Z is a set of attributes, then $ZX \longrightarrow ZY$ holds.
- **Transitivity rule**: If $X \longrightarrow Y$ holds and $Y \longrightarrow Z$ holds, then $X \longrightarrow Z$ holds.

A functional dependency $X \longrightarrow Y$ is termed as **trivial** if $X \supseteq Y$; otherwise,it is **nontrivial**.

Functional Dependencies

More inference axioms:

- **Union rule**. If $X \longrightarrow Y$ and $X \longrightarrow Z$, then $X \longrightarrow YZ$ holds.
- **Pseudotransitive rule**. If $X \longrightarrow Y$ and $YW \longrightarrow Z$, then $XW \longrightarrow Z$ holds.
- **Decomposition rule**. If $X \longrightarrow YZ$, then $X \longrightarrow Y$ and $X \longrightarrow Z$ hold.

Armstrong's axioms are **sound** and **complete**. These inference axioms can be derived from Armstrong's axioms.

Functional Dependencies

Proving Union rule from Armstrong's axioms:

Given: $\mathtt{X} \longrightarrow \mathtt{Y}$; $\mathtt{X} \longrightarrow \mathtt{Z}$

$\Longrightarrow \mathtt{XX} \longrightarrow \mathtt{XY}$; $\mathtt{XY} \longrightarrow \mathtt{YZ}$ (using augmentation of $\mathtt{X}$ in $\mathtt{X} \longrightarrow \mathtt{Y}$ and $\mathtt{Y}$ in $\mathtt{X} \longrightarrow \mathtt{Z}$)

$\Longrightarrow \mathtt{X} \longrightarrow \mathtt{XY}$; $\mathtt{XY} \longrightarrow \mathtt{YZ} \Longrightarrow \mathtt{X} \longrightarrow \mathtt{YZ}$ (using transitivity rule)

Functional Dependencies

Proving Pseudotransitive rule from Armstrong's axioms:

Given: $\texttt{X} \longrightarrow \texttt{Y}$; $\texttt{YW} \longrightarrow \texttt{Z}$

$\implies \texttt{XW} \longrightarrow \texttt{YW}$; $\texttt{YW} \longrightarrow \texttt{Z}$ (using augmentation of $\texttt{W}$ in $\texttt{X} \longrightarrow \texttt{Y}$)

$\implies \texttt{XW} \longrightarrow \texttt{Z}$ (using transitivity rule)

Functional Dependencies

Proving Decomposition rule from Armstrong's axioms:

Given: $\mathtt{X} \longrightarrow \mathtt{YZ}$

We know $\mathtt{YZ} \longrightarrow \mathtt{Y}$; $\mathtt{YZ} \longrightarrow \mathtt{Z}$ (using reflexive rule)

From $\mathtt{X} \longrightarrow \mathtt{YZ}$; $\mathtt{YZ} \longrightarrow \mathtt{Y}$

$\Longrightarrow \mathtt{X} \longrightarrow \mathtt{Y}$ (using transitivity rule)

From $\mathtt{X} \longrightarrow \mathtt{YZ}$; $\mathtt{YZ} \longrightarrow \mathtt{Z}$

$\Longrightarrow \mathtt{X} \longrightarrow \mathtt{Z}$ (using transitivity rule)

Implication of Functional Dependencies and Closure

Let functional dependency set $FD = \{AB \longrightarrow CD,\ B \longrightarrow DE,\ C \longrightarrow F,\ E \longrightarrow G,\ A \longrightarrow B\}$**. Use Armstrong's axioms to derive that** $A \longrightarrow FG$ **is logically implied by** FD

Step#	Inference	Justification
1	$A \longrightarrow B$	Given
2	$A \longrightarrow AB$	Augmentation of A on step 1
3	$AB \longrightarrow CD$	Given
4	$A \longrightarrow CD$	Transitivity on steps 2,3
5	$B \longrightarrow DE$	Given
6	$A \longrightarrow DE$	Transitivity on steps 1,5
7	$A \longrightarrow ACD$	Augmentation of A on step 4
8	$ACD \longrightarrow CDE$	Augmentation of C,D on step 6
9	$A \longrightarrow CDE$	Transitivity on steps 7,8
10	$A \longrightarrow CE$	Trivial dependency from step 9
11	$C \longrightarrow F$	Given
12	$CE \longrightarrow EF$	Augmentation of E on step 11
13	$E \longrightarrow G$	Given
14	$FE \longrightarrow FG$	Augmentation of F on step 13
15	$CE \longrightarrow FG$	Transitivity on steps 12,14
16	$A \longrightarrow FG$	Transitivity on steps 10,15

Implication of Functional Dependencies and Closure

The set of **ALL** FDs implied by a given set $\mathtt{F}$ of FDs is called the **closure** of $\mathtt{F}$, and denoted as $\mathtt{F}^+$.

Armstrong Axioms can be applied repeatedly to infer all FDs implied by a set $\mathtt{F}$ of FDs.

We already read that Armstrong axioms are **sound** and **complete**. The exact meaning is:

Sound: The axioms generate ONLY FDs in $\mathtt{F}^+$ when applied to a given set of FDs $\mathtt{F}$.

Complete: The axioms, when repeatedly applied to a given set of FDs $\mathtt{F}$, will generate ALL FDs in $\mathtt{F}^+$.

Implication of Functional Dependencies and Closure

Attribute Closure:
For a given FD set, **closure of an attribute** is the set of all the attributes in the relation that the input attribute can determine by using inference axioms and given FD set. Closure of an attribute $\mathtt{A}$ is denoted by $\{\mathtt{A}\}^+$ or $(\mathtt{A})^+$.

Closure of $\mathtt{AB} = (\mathtt{AB})^+ = \{\mathtt{A}^+ \cup \mathtt{B}^+ \cup$ (Any FD in $\mathtt{F}$ where $\mathtt{AB}$ is the determinant)$\}$

Given the following FD set $\mathtt{F}=\{\mathtt{X} \longrightarrow \mathtt{YZ},\ \mathtt{ZW} \longrightarrow \mathtt{P},\ \mathtt{P} \longrightarrow \mathtt{Z},\ \mathtt{W} \longrightarrow \mathtt{XPQ},\ \mathtt{XYQ} \longrightarrow \mathtt{YW},\ \mathtt{WQ} \longrightarrow \mathtt{YZ}\}$, find the closure of all the single attributes.

Implication of Functional Dependencies and Closure

Systematically computing Closure of an FD set:

Step 1. Compute S, which is the set all attributes in the FD set

Step 2. Compute P(S), which is the power set of S except null element

Step 3. Compute closure of each element of P(S)

Step 4. If the closure of an element of P(S) is of the form $\{X\}^+ = \{Y\}$, then $(2^{|Y|} - 1)$ number of FDs will be found from this. The FDs will be of the form $X \longrightarrow Z$ where Z is any element in P(Y) (power set of Y) except null.

Implication of Functional Dependencies and Closure

Systematically computing Closure of an FD set:

Find closure of $F = \{A \longrightarrow B,\ A \longrightarrow C,\ B \longrightarrow C\}$

Attribute Closure	**Derived FDs**
$A^+ = \{ABC\}$	$A \longrightarrow A$, $A \longrightarrow B$, $A \longrightarrow C$, $A \longrightarrow AB$, $A \longrightarrow BC$, $A \longrightarrow AC$, $A \longrightarrow ABC$
$B^+ = \{BC\}$	$B \longrightarrow B$, $B \longrightarrow C$, $B \longrightarrow BC$
$C^+ = \{C\}$	$C \longrightarrow C$
$(AB)^+ = \{ABC\}$	$AB \longrightarrow A$, $AB \longrightarrow B$, $AB \longrightarrow C$, $AB \longrightarrow AB$, $AB \longrightarrow BC$, $AB \longrightarrow AC$, $AB \longrightarrow ABC$
$(BC)^+ = \{BC\}$	$BC \longrightarrow B$, $BC \longrightarrow C$, $BC \longrightarrow BC$
$(AC)^+ = \{ABC\}$	$AC \longrightarrow A$, $AC \longrightarrow B$, $AC \longrightarrow C$, $AC \longrightarrow AB$, $AC \longrightarrow BC$, $AC \longrightarrow AC$, $AC \longrightarrow ABC$
$(ABC)^+ = \{ABC\}$	$ABC \longrightarrow A$, $ABC \longrightarrow B$, $ABC \longrightarrow C$, $ABC \longrightarrow AB$, $ABC \longrightarrow BC$, $ABC \longrightarrow AC$, $ABC \longrightarrow ABC$
Closure of F	All the FDs above in this column

Implication of Functional Dependencies and Closure

Q. In a schema with attributes `A, B, C, D, E`, following set of functional dependencies are given: `A -> B; A -> C; CD -> E; B -> D; E -> A`
Which of the following functional dependencies is NOT implied by the above set?
(a) `CD -> AC`
(b) `BD -> CD`
(c) `BC -> CD`
(d) `AC -> BC`
[GATE2005]

Implication of Functional Dependencies and Closure

Q. In a schema with attributes `A, B, C, D, E`, following set of functional dependencies are given: `A -> B; A -> C; CD -> E; B -> D; E -> A`

Which of the following functional dependencies is NOT implied by the above set?

(a) `CD -> AC`

(b) `BD -> CD`

(c) `BC -> CD`

(d) `AC -> BC`

[GATE2005]

ANSWER: (b)

Implication of Functional Dependencies and Closure

Extraneous Attribute:
For a given FD set $\mathtt{F}$, an attribute $\mathtt{A}$ is **extraneous** in $\mathtt{X} \longrightarrow \mathtt{Y}$ if $\mathtt{A}$ can be removed from the left side or right side of $\mathtt{X} \longrightarrow \mathtt{Y}$ without altering the closure of $\mathtt{F}$.

Let $\mathtt{G} = \{\mathtt{A} \longrightarrow \mathtt{BC},\ \mathtt{B} \longrightarrow \mathtt{C},\ \mathtt{AB} \longrightarrow \mathtt{D}\ \}$

Attribute $\mathtt{C}$ is extraneous in the right side of $\mathtt{A} \longrightarrow \mathtt{BC}$
i.e., $\mathtt{G}' = \{\mathtt{A} \longrightarrow \mathtt{B},\ \mathtt{B} \longrightarrow \mathtt{C},\ \mathtt{AB} \longrightarrow \mathtt{D}\ \}$ has same closure as $\mathtt{G}$

Attribute $\mathtt{B}$ is extraneous in the left side of $\mathtt{AB} \longrightarrow \mathtt{D}$
i.e., $\mathtt{G}'' = \{\mathtt{A} \longrightarrow \mathtt{BC},\ \mathtt{B} \longrightarrow \mathtt{C},\ \mathtt{A} \longrightarrow \mathtt{D}\ \}$ has same closure as $\mathtt{G}$

Implication of Functional Dependencies and Closure

The Satisfies Algorithm

Used to determine if a relation R satisfies or doesn't satisfy a given FD: $A \longrightarrow B$

- **Input:** Relation R and an FD: $A \longrightarrow B$
- **Output:** TRUE if R satisfies $A \longrightarrow B$, otherwise FALSE
- **Step 1:** Sort the tuples of the relation R on the attribute(s) A (determinant) so that tuples with equal values under A are next to each other
- **Step 2:** Check that tuples with equal values under A also have equal values under attribute(s) B
- **Step 3:** If any two tuples of R have equal values under A but different values under attribute(s) B, output of the algorithm is FALSE
- **Step 4:** If every two tuples of R having equal values under A also have same values under attribute(s) B, output of the algorithm is TRUE

Implication of Functional Dependencies and Closure

Consider the relation TABLE_PURCHASE_DETAIL(Customer_ID, Store_ ID, Purchase_Location)

TABLE_PURCHASE_DETAIL

Customer ID	Store ID	Purchase Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

Check if the following functional dependencies are satisfied in the above relation:

Q1. Customer_ID $\longrightarrow$ Purchase_Location

Q2. Store_ID $\longrightarrow$ Purchase_Location

Q3. {Customer_ID, Store_ID} $\longrightarrow$ Purchase_Location

Q4. Customer_ID $\longrightarrow$ Store_ID

Implication of Functional Dependencies and Closure

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

Q. Which of the following functional dependencies are satisfied by the instance?
(a) `XY -> Z` and `Z -> Y`
(b) `YZ -> X` and `Y -> Z`
(c) `YZ -> X` and `X -> Z`
(d) `XZ -> Y` and `Y -> X`
[GATE 2000]

Implication of Functional Dependencies and Closure

Redundancy in functional dependency:
Given a set F of FDs, a FD $A \longrightarrow B$ in F is said to be **redundant** with respect to the FDs of F if and only if $A \longrightarrow B$ is implied and can be derived from a subset F' of F such that $F' \equiv F - \{A \longrightarrow B\}$.

Eliminating Redundant FDs allows us to minimize the set of FDs.

Implication of Functional Dependencies and Closure

The Membership Algorithm

Used to determine if there exists a redundant FD $A \longrightarrow B$ in a given set of functional dependencies F

- **Input:** F and a FD $A \longrightarrow B$ belonging to F
- **Output:** TRUE if $A \longrightarrow B$ is redundant in F, otherwise FALSE
- **Step 1:** Remove temporarily $A \longrightarrow B$ from F. Set $G = F - \{ A \longrightarrow B \}$. If $G \neq \phi$, proceed to Step 2; otherwise halt with output FALSE
- **Step 2:** Initialize the set of attributes T_i with $i=1$ with the set of attribute(s) A, i.e., Set $T_i = T_1 = \{A\}$.
- **Step 3:** Search in G for FDs $X \longrightarrow Y$ such that $X \subseteq T_i$.
- **Step 3a:** If such FD $X \longrightarrow Y$ is found from Step 3, form $T_{i+1} \longleftarrow Y \cup T_i$ and assign $i \longleftarrow i+1$.
- **Step 3aa:** If all the attributes of B belongs to T_i, declare the FD $A \longrightarrow B$ to be redundant, halt with output TRUE.
- **Step 3ab:** If all attributes of B are not members of T_i, assign $G \longleftarrow G - \{X \longrightarrow Y\}$ and repeat Step 3.
- **Step 3b:** If $G = \phi$ or there is no such FD is found from Step 3, then halt with output FALSE.

Implication of Functional Dependencies and Closure

Given the set F=$\{X \longrightarrow YW,\ XW \longrightarrow Z,\ Z \longrightarrow Y,\ XY \longrightarrow Z\}$. Using membership algorithm, determine if the FD $XY \longrightarrow Z$ is redundant in F.

Step#	G	**Is** G = ϕ	i	T_i	**Does** Z $\in T_i$ **?**
1	$\{X \longrightarrow YW,\ XW \longrightarrow Z,\ Z \longrightarrow Y\}$	No	1	$\{XY\}$	No
2	$\{XW \longrightarrow Z,\ Z \longrightarrow Y\}$	No	2	$\{XYW\}$	No
3	$\{Z \longrightarrow Y\}$	No	3	$\{XYWZ\}$	Yes

The algorithm halts as $Z \in T_i$ at i=3.

$XY \longrightarrow Z$ is redundant in F.

Implication of Functional Dependencies and Closure

Verification of the membership algorithm by iteratively applying Armstrong's axioms and derived axioms

Step#	Inference	Justification
1	X $\longrightarrow$ YW	Given
2	XY $\longrightarrow$ YW	Augmentation of Y on Step 1
3	XY $\longrightarrow$ XYW	Augmentation of X on Step 2
4	XW $\longrightarrow$ Z	Given
5	XYW $\longrightarrow$ YZ	Augmentation of Y on Step 4
6	XY $\longrightarrow$ YZ	Transitivity on Steps 3,5
7	YZ $\longrightarrow$ Z	Trivial
8	XY $\longrightarrow$ Z	Transitivity on Steps 6,7

Implication of Functional Dependencies and Closure

Given the set F={X $\longrightarrow$ Z, ZW $\longrightarrow$ X, Z $\longrightarrow$ Y, Z $\longrightarrow$ W}. Using membership algorithm, determine if the FD Z $\longrightarrow$ Y is redundant in F.

Step#	G	**Is** G = ϕ	i	T_i	**Does** Y $\in$ T_i **?**
1	{X $\longrightarrow$ Z, ZW $\longrightarrow$ X, Z $\longrightarrow$ W}	No	1	{Z}	No
2	{X $\longrightarrow$ Z, ZW $\longrightarrow$ X}	No	2	{ZW}	No
3	{X $\longrightarrow$ Z}	No	3	{XZW}	No
4	ϕ	Yes	4	{XZW}	No

The algorithm halts as G = ϕ but Y $\notin$ T_4.

Z $\longrightarrow$ Y is NOT redundant in F.

Implication of Functional Dependencies and Closure

Given the set $F=\{X \longrightarrow YZ,\ YW \longrightarrow Z,\ Z \longrightarrow X,\ X \longrightarrow W\}$. Using membership algorithm, determine if the FD $X \longrightarrow W$ is redundant in F.

Step#	G	**Is** $G = \phi$	i	T_i	**Does** $W \in T_i$ **?**
1	$\{X \longrightarrow YZ,\ YW \longrightarrow Z,\ Z \longrightarrow X\}$	No	1	$\{X\}$	No
2	$\{YW \longrightarrow Z,\ Z \longrightarrow X\}$	No	2	$\{XYZ\}$	No
3	$\{YW \longrightarrow Z\}$	No	3	$\{XYZ\}$	No

The algorithm halts as G will not reduce further but $W \notin T_3$.
$X \longrightarrow W$ is NOT redundant in F.

Implication of Functional Dependencies and Closure

Check if $\mathtt{BD} \longrightarrow \mathtt{E}$ is a redundant FD in $\mathtt{F} = \{\mathtt{A} \longrightarrow \mathtt{B},\ \mathtt{C} \longrightarrow \mathtt{D},\ \mathtt{BD} \longrightarrow \mathtt{E},\ \mathtt{AC} \longrightarrow \mathtt{E}\}$

Check if $\mathtt{AC} \longrightarrow \mathtt{E}$ is a redundant FD in $\mathtt{F} = \{\mathtt{A} \longrightarrow \mathtt{B},\ \mathtt{C} \longrightarrow \mathtt{D},\ \mathtt{BD} \longrightarrow \mathtt{E},\ \mathtt{AC} \longrightarrow \mathtt{E}\}$

Eliminate redundant FDs from $\mathtt{F}=\{\mathtt{X} \longrightarrow \mathtt{Y},\ \mathtt{Y} \longrightarrow \mathtt{X},\ \mathtt{Y} \longrightarrow \mathtt{Z},\ \mathtt{Z} \longrightarrow \mathtt{Y},\ \mathtt{X} \longrightarrow \mathtt{Z},\ \mathtt{Z} \longrightarrow \mathtt{X}\}$ using the Membership algorithm.

Find the redundant FDs in the set $\mathtt{F} = \{\mathtt{X} \longrightarrow \mathtt{YZ},\ \mathtt{ZW} \longrightarrow \mathtt{P},\ \mathtt{P} \longrightarrow \mathtt{Z},\ \mathtt{W} \longrightarrow \mathtt{XPQ},\ \mathtt{XYQ} \longrightarrow \mathtt{YW},\ \mathtt{WQ} \longrightarrow \mathtt{YZ}\}$. Apply Membership algorithm $|\mathtt{F}|$ times to validate non-redundancy of every member FDs.

The set $\mathtt{G}$ found after removing **ALL** redundant FDs from $\mathtt{F}$ is called **non-redundant cover** of $\mathtt{F}$.

Implication of Functional Dependencies and Closure

Two sets of FDs F and G defined over same relation schema are **equivalent** iff
i. every FD in F can be inferred from G
AND
ii. every FD in G can be inferred from F

G **covers** F if every FD in F can be inferred from G (i.e., if F^+ is subset of G^+)

Two sets of FDs F and G defined over same relation schema are equivalent if F covers G and G covers F

G is a **non-redundant cover** of F if G covers F and no proper subset H of G exist such that $H^+ = G^+$.

Implication of Functional Dependencies and Closure

A **superkey** is a unique set of attribute(s) that determine the set of other attributes in a relation. In a relation `R(A,B,C,D,E,F)`, we define set of attributes as `P=`$\{$`A,B,C,D,E,F`$\}$. A superkey `SK` is a subset of `P` (`SK` $\subseteq$ `P`) that determines all other attributes,
i.e.,`(SK)`$^+$ `= P` or `SK -> P - SK`.

There can be many superkeys in a relation. A superkey is a set of attributes that has the uniqueness property, but is not necessarily minimal.

candidate key is a minimal superkey, i.e. removing any attribute from a candidate key will not retain its ability to uniquely determine other attributes.

Two properties of candidate key, or called just **key**, are `unique` and `minimal`.

Implication of Functional Dependencies and Closure

If a relation has multiple keys, database designer specifies one of them to be the used as a key while others won't be. This specially selected key is called **primary key**.

The candidate keys which do not get elected as primary key are called **alternate keys** or **secondary keys**.

Convention: in a relational schema, underline the attributes of the primary key.

Surrogate key, also called a *synthetic* primary key, is generated when a new record is inserted into a table automatically by a database that can be declared as the primary key of that table . It is the sequential number outside of the database that is made available to the user and the application or it acts as an object that is present in the database but is not visible to the user or application.

Implication of Functional Dependencies and Closure

How to find superkeys / candidate keys in a given relation:
Superkeys are those sets of attributes whose closure is the set of all attributes.

Find all superkeys and candidate keys in $F = \{A \longrightarrow B, A \longrightarrow C, B \longrightarrow C\}$
Let us first find the attribute closure of all subsets of the attribute set.

Attribute set	Closure	Closure contains all attributes?	Superkey?
A	$A^+ = \{ABC\}$	Y	Y
B	$B^+ = \{BC\}$	N	N
C	$C^+ = \{C\}$	N	N
AB	$(AB)^+ = \{ABC\}$	Y	Y
BC	$(BC)^+ = \{BC\}$	N	N
AC	$(AC)^+ = \{ABC\}$	Y	Y
ABC	$(ABC)^+ = \{ABC\}$	Y	Y

Superkeys: A; $\{AB\}$; $\{AC\}$; $\{ABC\}$.
Candidate keys are minimal superkeys, i.e., those superkeys, whose proper subsets are not a superkey, are candidate keys.
Candidate keys: A. In this case, there is only one candidate key!

Implication of Functional Dependencies and Closure

Q. An instance of relational schema `R(A,B,C)` has distinct values for attribute `A`. Can you conclude that `A` is a candidate key for `R`?
[GATE1994]

Implication of Functional Dependencies and Closure

Q. Relation `R` has eight attributes `A,B,C,D,E,F,G,H`. Fields of `R` contain only atomic values. `F = {CH -> G, A -> BC, B -> CFH, E -> A, F -> EG}` is a set of functional dependencies (FDs) so that F^+ is exactly the set of FDs that hold for `R`. How many candidate keys does the relation `R` have?
(a) 3
(b) 4
(c) 5
(d) 6
[GATE2013]

Implication of Functional Dependencies and Closure

Q. Consider a relation scheme `R (A, B, C, D, E, H)` on which the following functional dependencies hold: {`A -> B, BC -> D, E -> C, D -> A`}. What are the candidate keys of `R`?
(a) `AE, BE`
(b) `AE, BE, DE`
(c) `AEH, BEH, BCH`
(d) `AEH, BEH, DEH`
[GATE2005]

Implication of Functional Dependencies and Closure

Q. A Relation `R` with FD set {`A -> BC, B -> A, A -> C, A -> D, D -> A`}. How many candidate keys will be there in `R`?
(a) 1
(b) 2
(c) 3
(d) 4

Implication of Functional Dependencies and Closure

Q. The maximum number of superkeys for the relation schema `R(E,F,G,H)` with `E` as the key is:
(a) 5
(b) 6
(c) 7
(d) 8
[GATE2014]

Implication of Functional Dependencies and Closure

Q. Which of the following is NOT a superkey in a relational schema with attributes `V, W, X, Y, Z` and primary key `VY` ?
(a) `VXYZ`
(b) `VWXZ`
(c) `VWXY`
(d) `VWXYZ`
[GATE2016]

Implication of Functional Dependencies and Closure

Q. Consider the relation scheme `R = {E, F, G, H, I, J, K, L, M}` and the set of functional dependencies `{ {E,F} -> {G}, {F} -> {I,J}, {E,H} -> {K,L}, {K} -> {M}, {L} -> {N}}` on `R`. What is the key for `R`?
(a) `{E,F}`
(b) `{E,F,H}`
(c) `{E,F,H,K,L}`
(d) `{E}`
[GATE2014]

Implication of Functional Dependencies and Closure

Q. The following functional dependencies hold true for the relational schema `R{V, W, X, Y, Z}`:
`V -> W; VW -> X; Y -> VX; Y -> Z`
Which of the following is irreducible equivalent for this set of functional dependencies?
(a) `{V -> W; V -> X; Y -> V; Y -> Z}`
(b) `{V -> W; W -> X; Y -> V; Y -> Z}`
(c) `{V -> W; V -> X; Y -> V; Y-> X; Y -> Z}`
(d) `{V -> W; W -> X; Y -> V; Y -> X; Y -> Z}`
[GATE2017]