UNIT-3(Data Base Design & Normalization)

Functional dependencies:

A functional dependency X->Y in a relation holds if two tuples having same value of attribute A also have same value for attribute B.

Where x ->determinant

Y->dependent

X	Y
a	1
b	2

X->Y

X	Y
a	1
a	2

here ,
y is not functionaly dependent on x

---

a->1

b->2

a->1

a->2

i.e Attribute Y is functionally dependent upon attribute X if a value of X determines a single value of attribute Y at any one time.

If t1(x)=t2(x) then t1(y)=t2(y)

	X	Y
t1	a	1
	b	2
t2	a	1
	b	2

Here in the above relation: t1(x)=a , t2(x)=a and t1(y)=1,t2(y)=1

so ,here X→Y

Example:

Employee number	Employee Name	Salary	City
1	Dana	50000	San Francisco
2	Francis	38000	London
3	Andrew	25000	Tokyo

In this example, if we know the value of Employee number, we can obtain Employee Name, city, salary, etc. By this, we can say that the city, Employee Name, and salary are functionally depended on Employee number.

Employee number→Employee Name

Employee number->Salary

Employee number→City

Example2:

eid	ename	department	Designation	Age
1	Amit	Production	Manager	40
2	Sumit	Marketing	Asst.Manager	35
3	Rahul	Production	Team Leader	30
1	Amit	Markting	Asst.manager	40
5	Ronit	Finance	Team Leader	32
6	Mohit	R & D	Manager	45

In the above table:

Functional Dependencies are:

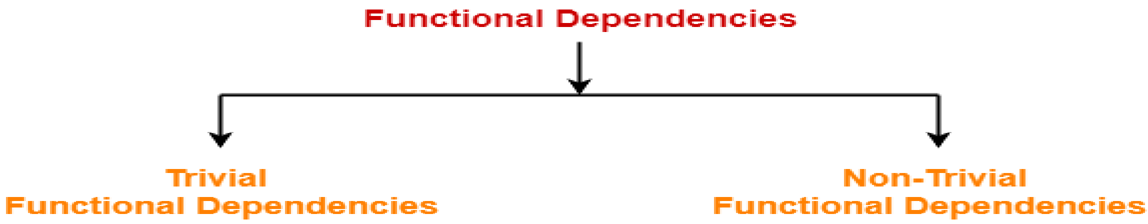
1.eid→ename
2.eid→age
3.ename,age→eid
4.emane→age

Invalid functional Dependencies are

1.eid->department
2.eid→Designation
3.ename→department
4.ename→designation

Types Of Functional Dependencies-

There are two types of functional dependencies-



1. Trivial Functional Dependencies
2. Non-trivial Functional Dependencies

1. Trivial Functional Dependencies-

=>A functional dependency X → Y is said to be trivial if and only if Y ⊆ X.
=>Thus, if RHS of a functional dependency is a subset of LHS, then it is called as a trivial functional dependency.

Examples-

The examples of trivial functional dependencies are-
AB → A
AB → B
AB → AB

2. Non-Trivial Functional Dependencies-

- A functional dependency X → Y is said to be non-trivial if and only if Y ⊄ X.
- Thus, if there exists at least one attribute in the RHS of a functional dependency that is not a part of LHS, then it is called as a non-trivial functional dependency.

Examples-

The examples of non-trivial functional dependencies are-

- AB → BC
- AB → CD

Armstrong Axioms or Inference Rules-

1. Reflexivity-

If B is a subset of A, then A → B always holds.

2.Transitivity-

If A → B and B → C, then A → C always holds.

3.Augmentation-

If A → B, then AC → BC always holds.

4.Decomposition-

If A → BC, then A → B and A → C always holds.

5.Composition-

If A → B and C → D, then AC → BD always holds.

6.Additive-

If A → B and A → C, then A → BC always holds.

Rules for Functional Dependency-

Rule-01:

A functional dependency X → Y will always hold if all the values of X are unique (different) irrespective of the values of Y.

Example-

Consider the following table-

A	B	C	D	E
5	4	3	2	2
8	5	3	2	1
1	9	3	3	5
4	7	3	3	8

The following functional dependencies will always hold since all the values of attribute ‘A’ are unique-

- A → B
- A → BC
- A → CD
- A → BCD
- A → DE
- A → BCDE

In general, we can say following functional dependency will always hold-

A → Any combination of attributes A, B, C, D, E

Similar will be the case for attributes B and E.

Rule-02:

A functional dependency X → Y will always hold if all the values of Y are same irrespective of the values of X.

Example-

Consider the following table-

A	B	C	D	E
5	4	3	2	2
8	5	3	2	1
1	9	3	3	5
4	7	3	3	8

The following functional dependencies will always hold since all the values of attribute ‘C’ are same-

- $A \rightarrow C$
- $AB \rightarrow C$
- $ABDE \rightarrow C$
- $DE \rightarrow C$
- $AE \rightarrow C$

In general, we can say following functional dependency will always hold true-

Any combination of attributes A, B, C, D, E $\rightarrow$ C

Closure of an Attribute Set-

- The set of all those attributes which can be functionally determined from an attribute set is called as a closure of that attribute set.
- Closure of attribute set {X} is denoted as $\{X\}^+$.

Example-

Consider a relation R (A , B , C , D , E , F , G) with the functional dependencies-

$$A \rightarrow BC$$

$$BC \rightarrow DE$$

$$D \rightarrow F$$

$$CF \rightarrow G$$

Now, let us find the closure of some attributes and attribute sets-

Closure of attribute A-

$A^+ = \{ A \}$

$= \{ A , B , C \}$ (Using $A \rightarrow BC$)

$= \{ A , B , C , D , E \}$ (Using $BC \rightarrow DE$)

$= \{ A , B , C , D , E , F \}$ (Using $D \rightarrow F$)

$= \{ A , B , C , D , E , F , G \}$ (Using $CF \rightarrow G$)

Thus,

$$A^+ = \{ A , B , C , D , E , F , G \}$$

Closure of attribute B-

$B^+=\{B\}$

Closure of attribute D-

$D^+ = \{ D \}$

$= \{ D , F \}$ (Using $D \rightarrow F$)

We can not determine any other attribute using attributes D and F contained in the result set.

Thus,

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

Closure of attribute set {B, C}-

$\{ B , C \}^+ = \{ B , C \}$

$= \{ B , C , D , E \}$ (Using $BC \rightarrow DE$)

$= \{ B , C , D , E , F \}$ (Using $D \rightarrow F$)

$= \{ B , C , D , E , F , G \}$ (Using $CF \rightarrow G$)

Thus,

$$\{ B , C \}^+ = \{ B , C , D , E , F , G \}$$

Example- Consider a relation R (A , B , C , D , E , F , G) with the functional dependencies-

$$AB \rightarrow CD$$

$$AF \rightarrow D$$

$$DE \rightarrow F$$

$$C \rightarrow G$$

$$F \rightarrow E$$

$$G \rightarrow A$$

Find the closure of the following

(a) $\{ CF \}^+$

(b) $\{ BG \}^+$

(c) $\{ AF \}^+$

(d) $\{ AB \}^+$

Solution-

(a):

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

$= \{ C , F , G \}$ (Using $C \rightarrow G$)

$= \{ C , E , F , G \}$ (Using $F \rightarrow E$)

$= \{ A , C , E , E , F \}$ (Using $G \rightarrow A$)

$= \{ A , C , D , E , F , G \}$ (Using $AF \rightarrow D$)

(b):

$\{ BG \}^+ = \{ B , G \}$

$= \{ A , B , G \}$ (Using $G \rightarrow A$)

$= \{ A , B , C , D , G \}$ (Using $AB \rightarrow CD$)

(c):

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

$= \{ A , D , F \}$ (Using $AF \rightarrow D$)

$= \{ A , D , E , F \}$ (Using $F \rightarrow E$)

(d):

$\{ AB \}^+ = \{ A , B \}$

$= \{ A , B , C , D \}$ (Using $AB \rightarrow CD$)

$= \{ A , B , C , D , G \}$ (Using $C \rightarrow G$)

Equivalence of Two Sets of Functional Dependencies-

In DBMS,

- Two different sets of functional dependencies for a given relation may or may not be equivalent.
- If F and G are the two sets of functional dependencies, then following 3 cases are possible-

Case-01: F covers G (F ⊇ G)

Case-02: G covers F (G ⊇ F)

Case-03: Both F and G cover each other (F = G)

Case-01: Determining Whether F Covers G-

Following steps are followed to determine whether F covers G or not-

Step-01:

=>Take the functional dependencies of set G into consideration.

=>For each functional dependency X → Y, find the closure of X using the functional dependencies of set G.

Step-02:

=>Take the functional dependencies of set G into consideration.

=>For each functional dependency X → Y, find the closure of X using the functional dependencies of set F.

Step-03:

=>Compare the results of Step-01 and Step-02.

=>If the functional dependencies of set F has determined all those attributes that were determined by the functional dependencies of set G, then it means F covers G.

- Thus, we conclude F covers G (F ⊇ G) otherwise not.

Case-02: Determining Whether G Covers F-

Following steps are followed to determine whether G covers F or not-

Step-01:

Take the functional dependencies of set F into consideration.

- For each functional dependency X → Y, find the closure of X using the functional dependencies of set F.

Step-02:

Take the functional dependencies of set F into consideration.

- For each functional dependency X → Y, find the closure of X using the functional dependencies of set G.

Step-03:

Compare the results of Step-01 and Step-02.

- If the functional dependencies of set G has determined all those attributes that were determined by the functional dependencies of set F, then it means G covers F.
- Thus, we conclude G covers F (G ⊇ F) otherwise not.

Case-03: Determining Whether Both F and G Cover Each Other-

- If F covers G and G covers F, then both F and G cover each other.
- Thus, if both the above cases hold true, we conclude both F and G cover each other

  (F = G).

PRACTICE PROBLEM BASED ON EQUIVALENCE OF FUNCTIONAL DEPENDENCIES-

Problem-

A relation R (A , C , D , E , H) is having two functional dependencies sets F and G as shown-

Set F-

A → C

AC → D

$$E \rightarrow AD$$

$$E \rightarrow H$$

Set G-

$$A \rightarrow CD$$

$$E \rightarrow AH$$

Which of the following holds true?

(A) $G \supseteq F$

(B) $F \supseteq G$

(C) $F = G$

(D) All of the above

OR

Ques: Consider the following two sets of FDs, F={A→C, AC→D, E→AD,E→H} and

G={A→CD,E→AH} check whether they are equivalent. (AKTU MCA-2009-10)

Solution-

Determining whether F covers G-

Step-01:

$(A)^+ = \{A\}$

$=\{A,C,D\}$(using$A \rightarrow CD$)

$(E)^+ = \{E\}$

$=\{E,A,H\}$

$=\{E,A,H,C,D\}$

$=\{ A , C , D , E , H \}$ // closure of left side of $E \rightarrow AH$ using set G

Step-02:

- $(A)^+ = \{ A\}$

={A,C,D}// closure of left side of A → CD using set F

$(E)^+$ = {E, A, D, C, H}

= { A , C , D , E , H } // closure of left side of E → AH using set F

<u>Step-03:</u>

Comparing the results of Step-01 and Step-02, we find-

- Functional dependencies of set F can determine all the attributes which have been determined by the functional dependencies of set G.
- Thus, we conclude F covers G i.e. F ⊇ G.

<u>Determining whether G covers F-</u>

<u>Step-01:</u> for FDs F={A→C, AC→D, E→AD,E→H}

$(A)^+$ = {A,C,D}

={ A , C , D } // closure of left side of A → C using set F

$(AC)^+$ = {A,C,D}

={ A , C , D } // closure of left side of AC → D using set F

$(E)^+$ = {E,A,D,H,C}

={ A , C , D , E , H } // closure of left side of E → AD and E → H using set F

<u>Step-02:</u> for FDs G={A→CD,E→AH}

- $(A)^+$ = {A,C,D}

  ={ A , C , D } // closure of left side of A → C using set G

- $(AC)^+$ ={A,C,D}

  ={ A , C , D } // closure of left side of AC → D using set G

- $(E)^+$ ={E,A,H,C,D}

  = { A , C , D , E , H } // closure of left side of E → AD and E → H using set G

<u>Step-03:</u>

Comparing the results of Step-01 and Step-02, we find-

- Functional dependencies of set G can determine all the attributes which have been determined by the functional dependencies of set F.
- Thus, we conclude G covers F i.e. G ⊇ F.

Determining whether both F and G cover each other-

From Step-01, we conclude F covers G.

From Step-02, we conclude G covers F.

Thus, we conclude both F and G cover each other i.e. F = G.

Thus, Option (D) is correct.

Finding the Keys Using Closure-

Super Key- super key is a set of attribute with the help of which you can uniquely identify a row or tuple in a table or you can uniquely identify all the row in a table.

Finding super key using closure:

- If the closure result of an attribute set contains all the attributes of the relation, then that attribute set is called as a super key of that relation.

Example- R(ABCD)

ABC→D

AB→CD

A→BCD

$(ABC)^+$=ABCD

$(AB)^+$=ABCD

$(A)^+$=ABCD

In the above example,

- The closure of attribute ABC,AB and A is the entire relation schema.
- Thus, attribute ABC, AB, A is a super key for that relation.

<u>Candidate Key-</u> A minimal super key is called candidate key.

Finding candidate key using closure:

If there exists no subset of an attribute set whose closure contains all the attributes of the relation, then that attribute set is called as a candidate key of that relation.

Example-

In the above example,

- No subset of attribute A contains all the attributes of the relation.
- Thus, attribute A is also a candidate key for that relation.

Primary Key: A primary key is that candidate key which is selected by database administrator as primary mean to identify all tuples.

Example: Consider the relation

R(A,B,C,D) With FDs

$B \rightarrow ACD$

$ACD \rightarrow B$

Find the super key, Candidate key and Primary key.

Sol: $B^+ = ABCD$

$ACD^+ = ABCD$

Super key: B, ACD

Candidate key: B, ACD

Primary key: B