Functional dependency

Relationship between one attribute with another in a table.

Denoted as $\longrightarrow$.

If X and Y are the attributes

Functional dependency is $X \longrightarrow Y$

$X \longrightarrow Y$ is X determines Y

(or)

Y is Functionally dependent on X.

Consider Example

X	Y
10	5
20	6
30	7
10	5

If X = 20 then

Y = 6

Here X is known as determinant

Y is Dependent

X, Y → set of Attributes

X, Y can be single attribute

(or)

It can can be set of attribute

- $X \rightarrow Y$ is FD is it satisfies a constraint

X	Y
10	5
20	6
30	7
10	5

If $t_1 \cdot X = t_2 \cdot X$

then

$t_1 \cdot Y = t_2 \cdot Y$

$t_1 \cdot X = 10$ $\quad$ $t_1 \cdot Y = 5$

$t_2 \cdot X = 10$ $\quad$ $t_2 \cdot Y = 5$

If

$t_1 \cdot X$ & $t_2 \cdot X$ are equal

then

$t_1 \cdot Y$ & $t_2 \cdot Y$ must be equal

- If this condition is satisfied then only X & Y are called Functional denpendent.

X	Y
10	5
20	6
30	7
10	4

Here it is not satisfing the condition

So X & Y are not Functionally dependent

$t_1 \cdot X = 10$ $\quad$ $t_1 \cdot Y = 5$

$t_2 \cdot X = 10$ $\quad$ $t_2 \cdot Y = 4$

Full functional Dependent

$X \rightarrow Y$ is full F.D

if we remove any attribute of X

voilates the Functionaldependency Rule

$(A, B) \rightarrow C$

if B is removed

$A \rightarrow C$ is not Functional dependency

(or)

if A is removed.

$A \rightarrow B$ is not Functional dependency

Then it is called Full Functional Dependent

Partial Functional Dependent

$X \rightarrow Y$ is Partial Functional Dependent

if we remove any attribute of X doesn't

voilates the Functional Dependency Rule

if B is removed from $(A, B) \rightarrow C$

$A \rightarrow C$ is Functional dependency

then it is called Partial Functional dependency

Example

Sid	Sname	Address	Course
101	Vinutha	Banglore	Python
101	Vinutha	Banglore	Java
102	Vidhya	Doddaballapura	Python
103	Vidhya	Yelahanka	C
104	Tasrin	Coorg	Java
105	Deeksha	Hasana	Python

FD's

Sid → Sname ✓

$t_1.X = 101$ → $t_1.Y = Vinutha$

$t_2.X = 101$ $t_2.Y = Vinutha$

no more repeatation

Sid → Address ✓

$t_1.X = 101$ → $t_1.Y = Banglore$

$t_2.X = 101$ $t_2.Y = Banglore$

CamScanner

Sid → Course [X]

t_1.X = 101 → t_1.Y = Python

t_2.X = 101 t_2.Y = Java

Course → SName [X]

t_1.X = Python → t_1.Y = Vinitha

t_2.X = Python t_2.Y = Rakshitha

Sname → Sid X

t_1.X = Vidhya t_1.Y = 102

t_2.X = Vidhya t_2.Y = 103

(Sid, Sname) → Course X

t_1.X = 101, Vinitha → t_1.Y = Python

t_2.X = 101, Vinitha t_2.Y = Java

(Sid, Course) → Sname ✓

All are unique

Full Functional Dependent

&

Partical Functional Dependent

Sid, Course → Sname

Sid → Sname ✓

(Sid, Course) → Sname

⇓

Partial Functional Dependent

(Sname, Course) → Sid ✓

no repeation

If course is removed

Sname → Sid ✗

Remove Same

Course → Sid ✗

(Sname, Course) → Sid

⇓

Full Functional Dependency

Properties of Functional Dependency

↓

Armstrong Axioms

Condition of Function Dependency

$X \longrightarrow Y$

$t_1 \cdot X = t_2 \cdot X$

then

$t_1 \cdot Y = t_2 \cdot y$

① Reflexivity

If A is a set of Attributes

B is subset of A

then

$A \longrightarrow B$ is Functional Dependent

Dependent is Subset of Determinant

② Augmentation

If $A \longrightarrow B$ is Functional Dependent

If C is a attribute or set of attribute

added to both Determinant & Dependent

then $AC \longrightarrow CB$ is Functional Dependent

③ Transitivity

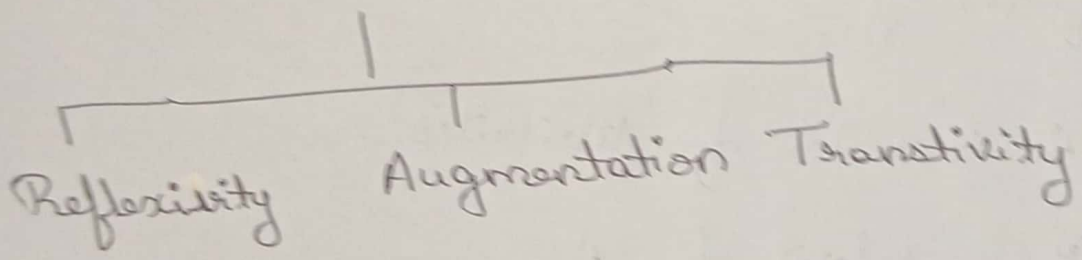
If $A \longrightarrow B$ is Functional Dependent

$B \longrightarrow C$ is also Functional Dependent

then

$A \longrightarrow C$ is also Functional Dependent

Armstrong Axioms



Inference Rules Derivied from Armstrong Axioms

Inference Rules or Secondary Rules

① Union

If $A \longrightarrow B$ is Functional Dependent

$A \longrightarrow C$ is Functional Dependent

then

$A \longrightarrow BC$ is also Functional Dependent

② Composition

If $A \rightarrow B$ is Functional Dependent

$C \rightarrow D$ is Functional Dependent

then

$AC \rightarrow BD$ is also F.D

③ Decomposition

If $A \rightarrow BC$ is Functional Dependent

then

$A \rightarrow B$

$A \rightarrow C$ both are Functional Dependent

④ Psuedo Transitivity

If $A \rightarrow B$ is F.D

$BC \rightarrow D$ is F.D

then $AC \rightarrow D$ is also F.D

Transitivity

$A \rightarrow B$

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

Inference Rules or Secondary Rules

- Union
- Composition
- Decomposition
- Psuedo Transitivity

NORMALIZATION

optimizing Relations

Finding the most efficient and effective way to connect or relate data, Aiming for improved performance, accuracy or clarity.

- In order to Avoid • Redundancy
- Anomalies

In order to Avoid this 2 problem we should optimize table which is done by Decomposition this process is know as Normalization.

This process is Done in different phases.

1NF — First Normal Form

2NF - Second Normal Form

3NF - Third Normal Form

BCNF - Boyce Codd Normal Form

4NF - Fourth Normal Form

5NF - Fifth Normal Form

- using all this we can achive normalization
- If we achive 1NF 2NF 3NF BCNF we can say schema is good design
  - If the relation is in 2NF first it should satisfy 1NF.
  - If the relation is in 3NF first it should satisfy 2NF.
  - If the relation is in BCNF first it should satisfy 3NF.

Still if there might be the chance of Redendency then we will use 4NF

- If the relation is in 4NF first it should satisfy BCNF
- If the relation is in 5NF first it should satisfy 4NF

Consider unnormalized data - one table with all attributes