Functional dependency (FD)

A func. dep. (FD) is denoted as $A \rightarrow B$, in a relation R holds, if two tuples having same value of A also have same value for B

$A \rightarrow B$

Accno → Year

A001	2015
A001	2015
~~A001~~	~~2016~~ ✗

Student (Roll, Name, Phone, Add, State, country)

FDs : {Roll → Name, Roll → Add, State → country}

Bihar → India
WB → India
Bihar → India

Country → State ✗
India → Bihar
India → WB

Proceeding towards Normalization

Name → Address ✗

Roll	Name	Phone	Add	State	Country
R001	XYZ	9831..	37/11A -- B.S...	WB	India
R002	ABC	8910...	1/1 SLe	Bihar	India
R003	Amal	700..	35 B.K..	Bihar	India

Roll → Name ✓

Roll → Add ✓

State → Country ✓

A → B

Country → State ✗

B is func. dep. on A

R001 → XYZ

Constraint: 2 students may have the same name, but their addresses are different

Address → Name ?

State is not a S. key
but
this one is FD

FDs in a relation are dependent on the domain of the rel.

State → Country

Left side of an FD not necessarily a super key

(Country is functionally dependent on State)

FD set of a relation is the set of all FDs present in the relation

Acc No $\quad$ Year

A001 $\rightarrow$ 2015

A001 $\rightarrow$ 2016

Case - I

✗

Acc No

A001 $\rightarrow$ 2015

A001 $\rightarrow$ 2015

Case - II

✓

A001 $\rightarrow$ 2015

A002 $\rightarrow$ 2016

Case - III

✓

FD

one - to - one mapping ? $\rightarrow$ think

one - to many " ?

many to one " ?

Armstrongs Axioms in FD

1) Axiom of reflexivity: If A is a set of attr. and B is subset of A, i.e.

$B \subseteq A$, then $A \rightarrow B$

this is a trivial property

e.g. $\{\underbrace{\text{Roll, Name}}_{A}\} \rightarrow \underbrace{\text{Name}}_{B}$

2) <u>Axiom of Augmentation</u>: If $A \rightarrow B$ holds and C is an attr. set, then $AC \rightarrow BC$ also holds

This is adding attributes in dependencies, does not change the basis dependencies.

if, Roll $\longrightarrow$ Name

then $\{Roll, State\} \longrightarrow \{Name, State\}$

3) Axiom of transitivity

if $A \rightarrow B$ holds, and $B \rightarrow C$ holds

then $A \rightarrow C$ will also hold

$$\text{Roll} \rightarrow \text{aadhar} , \quad \text{aadhar} \rightarrow \text{Name}$$

$$\text{Roll} \rightarrow \text{State} , \quad \text{State} \rightarrow \text{Country}$$

4) Union: if $A \to B$ and $A \to C$ hold
then $A \to BC$ holds

5) Composition: if $A \to B$ and $C \to D$ hold,
then $AC \to BD$ holds

6) Decomposition: if $A \to BC$ holds, then
$A \to B$ and $A \to C$ hold

7) Pseudo transitivity: if $A \to B$ and $BC \to D$ hold,
then $AC \to D$ hold.

$AC \to BC$ (augm.)
$BC \to D \implies AC \to D$ (trans.)

Attribute closure

$A \rightarrow A$

Atribute closure of an attribute set can be defined as the set of attributes which can be functionally determined from it. A

$r(A, B, C, D, E, F, G)$

FDs : $\{A \rightarrow B, A \rightarrow C, CD \rightarrow E, B \rightarrow D, E \rightarrow A, F \rightarrow G\}$

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

closure of $\{A, F\} = (AF)^+ = \cancel{AF}^+ = \{A, F, B, C, D, E, G\}$

$(EF)^+ = \{E, F, A, B, C, D, G\}$ | $F^+ = \{F, G\}$

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

How to fiind attribute closure of an attr. set?

.) Add elements of attr. set to the result set
..) Recursively add elements to the result set which can be func. determined from the elements of the result set

Find candidate and Super keys using attribure closure

1.) If attr. closure of an attr. set contains ALL attr. of a relation, then the attr. set is called super key (SK) of the rel.

2..) if no subset of this attr. set can functionally determine all attr. of the rel., then the set is called candidate key (CK)

from prev. example, $(AF)^+ = \{A, B, C, D, E, F, G\}$

$\Rightarrow \{A, F\}$ is a S.K.

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

$(F)^+ = \{F, G\}$

$\Rightarrow \{AF\}$ is C.K.

$\Rightarrow \{E, F\}$ is a S.K.

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

$(F)^+ = \{F, G\}$

$\Rightarrow \{E, F\}$ is a C.K.

other CK check

$\Rightarrow CKs = \{AF, EF\}$

$r(Id, Name, City, State)$

H.W.

FDs : $\{ Id \rightarrow Name,$
$Id \rightarrow City$
$Id \rightarrow State$
$City \rightarrow State \}$

find the set of candidate keys