Chapter 7: Normalized Database Design Part 2

Abu Raihan Mostofa Kamal

Professor, CSE Department
Islamic University of Technology (IUT)

October 30, 2023

[1]This is based on Textbook, its companion slide and other sources

Chapter Outline

Closure of Attribute Sets: X+

Attribute Closure X+

Attribute closure **X+** of an attribute set **X** can be defined as set of attributes which can be functionally determined from it.

Purpose:

- **Given a Relation and a set of FDs**, it is **used to determine** if a given attribute set is a **Candidate Key** or not.
- Determination of **all possible Candidate Keys** are needed for **testing of almost all Normal Forms** (will be discussed later).

Keys (Revisited) in terms of X+

Superkey

A superkey is a set of attribute (X) whose closure(X+) contains **all the attributes** of the given relation.

This definition is exactly identical to our previous definition: it can uniquely identify **each record**, but here it is presented in terms of attribute closure.

Candidate Key

A candidate key is a superkey whose **proper subset** is **not a superkey**. Proper subset means: **at least one element less**. (i.e. i.e.There shall not be any other superkey whose length is less than the present one)

This definition is exactly identical to our previous definition: it is the **minimum superkey**, but here it is presented in terms of attribute closure.

Closure of Attribute Sets(X+): How to compute (Meth1)

Given Relation $R(A, B, C, D, E)$ and **FD:** $A \longrightarrow B$, $B \longrightarrow C$,$C \longrightarrow D$,$D \longrightarrow E$

Method 1: Using Armstrong's Axioms without any optimization. **Simple trick:** Compute all $\alpha \longrightarrow \beta_i$ for all possible i. And finally **union** them to get the α^+

We can reduce:

- $A \longrightarrow C$ (transitivity)
- $A \longrightarrow A$ (reflexivity)
- $A \longrightarrow D$ (transitivity)
- $A \longrightarrow E$ (transitivity)
- $A \longrightarrow A, B, C, D, E$ (union)
- So, $A^+ = \{A, , B, C, D, E\}$

- $B \longrightarrow D$ (tran)
- $B \longrightarrow E$ (tran)
- $B \longrightarrow B$
- $B \longrightarrow B, C, D, E$ (U.)
- So, $B^+ = \{B, C, D, E\}$

- $C \longrightarrow E$
- $C \longrightarrow C$
- $C \longrightarrow C, D, E$
- So, $C^+ = \{C, D, E\}$

Process goes like that...**But** it is a long, time-consuming and very naive approach specially for a larger set of FDs and larger number of attributes.

Closure of Attribute Sets(X+): How to compute (Method 2)

Given Relation **R(A,B,C,D,E)** and **FD:** $A \longrightarrow B$, $B \longrightarrow C$,$C \longrightarrow D$,$D \longrightarrow E$

Method 2: Simplified approach

Basic Principle

First use **Reflexivity Rule** for a given attribute set, then iterate through the **given FD**s to add element in the set.When you finish the last FD, re-iterate if new element can be added. **Simple trick:** if $\alpha \longrightarrow \beta$ holds, and α is already in the Set, then add β (as α can determine β).

Closure of Attribute Sets(X+): How to compute (Method 2)

Given Relation **R(A,B,C,D,E)** and **FD:** $A \longrightarrow B, B \longrightarrow C, C \longrightarrow D, D \longrightarrow E$

Method 2: Simplified approach. We will compute **AD+** first:

- $AD^+ = \{A, D, B, E, C\}$
- $AB^+ = \{A, B, C, D, E\}$
- $B^+ = \{B, C, D, E\}$
- $CD^+ = \{C, D, E\}$
- $C^+ = \{C, D, E\}$

Determination of Superkey based on X+

(Recall the definition:) A superkey is a set of attribute (X) whose closure(X+) contains **all the attributes** of the given relation.
So far we have computed X+ as follows:

- $A^+ = \{A, B, C, D, E\}$ Superkey ✔
- $AD^+ = \{A, B, C, D, E\}$ Superkey ✔
- $B^+ = \{B, C, D, E\}$ Superkey ✘(A is absent)
- $AB^+ = \{A, B, C, D, E\}$ Superkey ✔
- $C^+ = \{C, D, E\}$ Superkey ✘ (A,B are absent)
- $CD^+ = \{C, D, E\}$ Superkey ✘ (A,B are absent)

Determination of Candidate Key

(Recall the definition:) A candidate key is a superkey(SK) whose **proper subset** is **not a superkey**. Proper subset means: at least one element less.
So far we have computed X+ as follows:

- $A^+ = \{A, B, C, D, E\}$ Superkey ✔ Candidate key ✔
- $AD^+ = \{A, B, C, D, E\}$ Superkey ✔ Not a Candidate key ✘(A is a proper subset of AD and A is already a SK)
- $B^+ = \{B, C, D, E\}$ Superkey ✘(A is absent)
- $AB^+ = \{A, B, C, D, E\}$ Superkey ✔ Not a Candidate key ✘(A is a proper subset of AD and A is already a SK)
- $C^+ = \{C, D, E\}$ Superkey ✘ (A,B are absent)
- $CD^+ = \{C, D, E\}$ Superkey ✘ (A,B are absent)

So, A is the only Candiate Key

Recap of Method 1 and Method 2

Method 1 : Just use all possible A. Axioms to determine X^+.

But, it is hard to compute for a large X.

Method 2 : Axioms with a simple technique.

Basic Principle : For a given X

i. First use Reflexivity Rule

ii. Look at the given FDs to include elements (if possible)

Recap of Method 1 and Method 2 (Cont.)

Example (same): Given $R(A, B, C, D, E)$

FD: $A \to B$, $B \to C$, $C \to D$, $D \to E$

Need to compute: X^+ (X is a set of attributes)

Big Picture: X^+ ↓ SK ↓ CK ↓ Normal Forms

Solution:

$A^+ = \{A\}$ Ref. Rule (RR)

$= A, B$ (FD: $A \to B$)

$= A, B, C$ (FD: $B \to C$)

$= A, B, C, D$ (FD: $C \to D$)

$= A, B, C, D, E$ (FD: $D \to E$)

$= \{A, B, C, D, E\}$ (all attributes)

So, A is a Superkey (SK)

Recap of Method 1 and Method 2 (Cont.2)

Method 2 : Another Example

Given : $R(A, B, C, D, E)$

$FD = \{A \to B,\ D \to E\}$

SK = Superkey

CK = Candidate key

Solution : Lets start with A

So, $A^+ = \{A\}$ (RR)

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

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

[$\alpha \to \beta$ means α can determine β]

Now Lets check,

1. $ABCDE^+ = \{A, B, C, D, E\}$ (so is a SK)
2. $ABDE^+ = \{A, B, D, E\}$ only RR, no FD helps here (NOT SK)
3. $ACDE^+ = \{A, C, D, E, B\}$ RR + FD : $A \to B$ (SK)
4. $ACD^+ = \{A, C, D, B, E\}$ RR + FD : $A \to B$, $D \to E$ (SK) (ACD)

Method 1 and Method 2: Problem

Both methods are **hard to implement** in real perspective especially when the R and FDs are large as the methods basically follow the brute-force approach. We are in need of an **optimized method**. We term it **Method 3**. In practice Method 3 is used.

Acronym used:
SK : Superkey
CK : Candidate Key
PSS : Proper Subset
RS : Reduced Set
PA : Prime Attributes
DA : Dependent Attributes
RP : Replacement Process

Method 3: Algorithm Overview (High-level)

Input: A relation **R(A,B,C...)** and a set of FDs: $A \longrightarrow B$,$C \longrightarrow D$...
Output: Possible sets of Candidate Keys
Step 1: Compute **Closure of All** Attributes+
then **discard** using given and implied FDs to get a Reduced Set (RS)
Now, check all PSS of the RS, if No PSS is a SK
then RS is the CK

Step 2: (Check for more CKs)
Form **Dependent Attributes (DA)** (static) from the given FDs
and **Prime Attribute (PA)** (all attributes of the CK discovered so far) (dynamic)
Now Compute $\alpha = DA \cap PA$
if α is NULL (empty) then no other CK is possible
else go for **Replacement Process (RP)** for each element of α
For each RP go to Step 1 to check if it a CK.

Method 3

Method 3   Basic Principle –

Step 1 : i) Start with all attributes and then (ii) eliminate attribute using FD's

e.g. If ABC in (i) and FD: $B \rightarrow C$

Here C can be removed as B can determine C and B is already in the set.

$AB\cancel{C} \Rightarrow AB$

{ other step will be discussed once we finish this step by example }

Benifit of step 1 : No of proper subsets will be significantly reduced.

Method 3: Simple Example

Method 3 (example) (step 1) ← discard step

Given: $R(A, B, C, D, E)$

FD $\{A \to B,\ D \to E\}$

Lets start with A, B, C, D, E (all)

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

A is already there.

Now, FD, $A \to B$ so, B can be removed from Left side and still will be able to reduce all attributes.

$A\cancel{B}CDE^+ = \{\underbrace{A, C, D, E}_{RR}, B\}$ $A \to B$

Right part

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

apply $D \to E$ (both are there so one can be removed)

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

$ACD^+ = \{A, B, C, D, E\}$ (SK) ✓

Method 3: Simple Example Cont.

So, $ACD^+ = \{A, B, C, D, E\}$ (SK) ✓

Need to test all proper sub sets are SK or not to finally say ACD is a C.K.

$\left\{\begin{array}{l} A \to B \\ D \to E \end{array}\right\}$ FD

ACD all proper subsets are:

$2^3 - 2 = 6$

$(2^x - 2)$ No.

AC	need to check	$AC^+ = \{A, C, B\}$ × SK
CD	"	$CD^+ = \{C, D, E\}$ × SK
AD	"	$AD^+ = \{A, D, E, B\}$ × SK
A	"	$A^+ = \{A, B\}$ × SK
C	"	$C^+ = \{C\}$ × SK
D	"	$D^+ = \{D, E\}$ × SK

Note: Here we have 6 proper subsets to check in contrast of earlier version $ABCDE = 2^5 - 2 = 30$

"step) motivation is to reduce the no. of checks."

Method 3: Simple Example Cont.

Method 3 : step 2

A New Term - Prime Attributes (PAs) : Collection of attributes of all candidate Keys of a given relation.

In step 1 we got ACD as CK (by testing ACD^+)

So, upto now PAs = {A, C, D}

Principle : For a given CK, form another set called Dependent Set DS/DA

also called Dep. Attributes (DA)

e.g. $A \to B$, $D \to E$ $\Rightarrow$ B E

IF PA $\cap$ DA = φ then CK is the only CK

no other possible CK is there.

present example

Else go for the Replacement Process (RP) (will later be explained)

Now example will be given

END IF;

Method 3: Simple Example (End)

Given $R(A, B, C, D, E)$ FD $(A \to B, D \to E)$

So, $DA = \{B, E\}$ (fixed) and $PA = \{A, C, D\}$ (as upto now CK = ACD)

ACD was CK in step 1

So, $\boxed{PA \cap DA} = \{A, C, D\} \cap \{B, E\}$

$= \varphi$

So, ACD is the only CK

Proved

Method 3: Standard Example (Begin)

New Example for the case $\boxed{CK_i \cap DS \neq \phi}$

Given : $R(A, B, C, D)$ & $\boxed{FD: A \to B,\ B \to C,\ C \to A}$

Step1 $ABCD^+ = \{A, B, C, D\}$ (since both A, B are there and B can be deduced from A so, B is redundant)

discard use: $A \to B \Rightarrow A\cancel{B}CD$

discard on T.R. $A \to B$ & $B \to C \Rightarrow A \to C$

$ACD = A\cancel{C}D$, so, we get AD

No more discard possible.

so, $AD^+ = A, B, C, D$ [we dont need to check now as discard method ensures it]

Now check all proper subsets of AD (i.e A, D)

Method 3: Standard Example Cont.

$A^+ = A, B, C$ NOT a SK

$D^+ = D$ NOT a SK

FD $A \to B$ $B \to C$ $C \to A$

So, AD is a candidate key (CK).

New step: Now we need to check if some CK exist.

Upto now - $PAs = A, D$

(fixed) $DA = B, C, A$

$A \to B$, $B \to C$, $C \to A$ $\Rightarrow DA$

Now, $DA \cap PAs = A$

(Now we look for $\boxed{\alpha \to A}$ then replace A with α, α will replace A)

Here, FD: $C \to A$

Method 3: Standard Example Cont.

So, RP AD ⇒ CD

RH will be replaced by LH.

$C \to A$

AD → A (C) → CD

FD: $A \to B$

$B \to C$

$C \to A$

(we start with existing CK (AD))

Now check if CD is SK.

obviously $CD^+ = (\text{all} --)$ (RP ensures it)

Proper Subset C, D

$C^+ = \{C, A, B\}$ ✗ $D^+ = \{D\}$ ✗

So, CD is a CK.

Now $PAs = \{A, D, C\}$ CD (DA= B, C, A)

Check $PAs \cap DA = A, C \Rightarrow C$ since A has already been considered.

Method 3: Standard Example Cont.

By same process: FD: $B \to C$ (already a CK)

FD $A \to B, B \to C, C \to A$

RP CD

$(B \to C)$ C (B) BD

$\Rightarrow BD \Rightarrow BD^+ = (\text{all})$

PSS

$B^+ = \{B, C, A\}$ ✗ SK (NOT)

$D^+ = \{D\}$ ✗ SK (NOT)

So, BD is a CK.

DA = B, C, A

Now, $PAs = \{A, D, C, B\}$ BD

$PAs \cap DA = A, B, C \Rightarrow B$

So, we look for FDs. $A \to B$

RP

$BD \to AD \Rightarrow AD$ is repeated

So, process ends.

Method 3: Standard Example (End)

So, Finally we get

set of CKs = {AD, CD, BD}

End of example.

Summary of Method 3 (which is used)

1. Start with all attributes $(A_1 A_2 \cdots A_n)^+$
2. Start discard using given or implied FD

$|(A_i \cdots A_j)| \leq |A_1 \cdots A_n|$

3. Compute $(A_i \cdots A_j)^+ \Rightarrow$ check all proper subsets
4. If it is a CK then form P.A also form DA = { } fixed

updatable

Method 3: Algorithm Review

5. Find $PA \cap DA = x_1 x_2 \cdots x_k$ $\quad k = 1, 2, \ldots$
and we replacing by $\alpha \to X$
and go to step 3.

Method 3: More Example A

2 More examples — Example A

Given $R(A, B, C, D)$

FD: $AB \rightarrow CD \quad D \rightarrow B \quad C \rightarrow A$

Solution

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

discard $AB\cancel{C}\cancel{D}^+ =$ "

$(AB \rightarrow CD)$

$AB^+ =$ "

Proper Subsets (PSS)

PSS of $AB \Rightarrow A, B$

so, $A^+ = \{A\}$ NOT SK

$B^+ = \{B\}$ NOT SK

So, AB is a CK. (i)

Method 3: More Example A (Cont.)

step 2

RP

AB is a CK

$PA = \{A, B\}$ upto now

FD: $AB \to CD$, $D \to B$, $C \to A$

$DA = \{C, D, B, A\}$

so, $PA \cap DA = A, B$

There will be 2 PR for 2 elements (i.e A, B)

AB

$C \to A$ A (C)   BC

B (D) $D \to B$ AD

$\to$ PSS   B, C $\Rightarrow$ $B^+ = \{B\}$ NOT SK

$C^+ = \{C, A\}$ NOT SK

So, BC is a CK $\to$ (ii)

Method 3: More Example A (Cont.2)

FD: $AB \rightarrow CD$, $D \rightarrow B$
$C \rightarrow A$

Now, $AD^+ = \{all\}$

PSS
$A^+ = A$
$D^+ = D, B$

So, AD is a CK — iii

Now, updated $PA = \{A, B, C, D\}$ | BC AD $DA = \{C, D, B, A\}$

$PA \cap DA = A, B, C, D = C, D$

None of the FD has C on D in RHS.

So, no RP here.

Method 3: More Example A (End)

But here one more discard can be done that will bring another (SK → CK).

FD: $AB \to CD$, $D \to B$, $C \to A$

$\cancel{AB}CD^+ =$ all

$CD^+ =$ "

PSS ⇒ $C^+ = \{C\}$ NOT SK

$D^+ = \{D, B\}$ NOT SK

So, CD is a CK —— (iv)

Now, PA = {A, B, C, D} No change, Process ends here.

Ans: {AB, BC, AD, CD} CK

Method 3: More Example B (Begin)

Example B (Adv.) Given, $R(A, B, C, D, E, F)$

FD: $AB \to C$ $C \to DE$ $E \to F$ $D \to A$ $C \to B$

Here discard is a tricky one as A. Axioms will be used.

Sol^n: — $ABCDEF^+ =$ all

We use TR $AB \to C$ $C \to DE \Rightarrow AB \to DE$

$ABC\cancel{D}\cancel{E}F^+ =$ all

$ABCF^+ =$

We use $C \to DE \Rightarrow C \to D$ and $C \to E$ (decomposition)

Again given $E \to F$ and $AB \to C$

$\Rightarrow AB \to F$ (TR)

$AB\cancel{C}\cancel{F}^+ =$

$AB \to C$

$AB^+ =$

Method 3: More Example B (Cont.)

PSS $A^+ = \{A, \}$ NOT SK

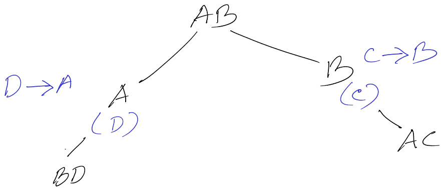
$B^+ = \{B, \}$ NOT SK

FD:
$AB \rightarrow C$
$C \rightarrow DE$
$E \rightarrow F$
$D \rightarrow A$
$C \rightarrow B$

So, AB is a CK.

Now, $PA = \{A, B\}$ $DA = \{C, D, E, F, A, B\}$

$PA \cap DA = A, B$ (Needs 2 PR)



Method 3: More Example B (End)

FD: $AB \to C$, $C \to DE$, $E \to F$, $D \to A$, $C \to B$

- $BD^+ =$ — all
  PSS: $B^+ =$ ✓ NOT SK
  $D^+ =$ ✓ NOT SK
  So, BD is a CK

- $AC^+ =$ all
  PSS: $A^+ = \{A, \}$
  $C^+ = \{C, B, D, E, F, A\}$ = is a SK also a CK because PSS is has $\emptyset$.

So, C is a CK.

Ans. {AB, BD, C}

Normalization: Motivation (Recap)

The **goal** of relational database design is to generate a set of relation schemas that will meet the following 2 goals:

- allows us to store information **without unnecessary redundancy**
- but allows us to **retrieve information easily**
- Hence, we need a standard method to evaluate a design called Normal Forms

Normalization:Motivation Example (Recap)

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

3 Problems:

1. **Redundancy**: Department info is repeated.
2. **Inconsistency**: Update of department info should be propagated properly.
3. **Introduces Bad Business Logic**: You can not enter data for a new department unless there is a teacher of that department.

Atomic Domains and First Normal Form

- A domain is atomic if elements of the domain are considered to be **indivisible units**.
- **Examples**: Hight, weight, country information in general.
- We say that a relation schema **R** is in first normal form (**1NF**) if:
  - The domains of **all attributes of R are atomic**.
  - And the value of each attribute contains **only a single value** from that domain.

First Normal Form: Example

ID	First Name	FamilyName	PhoneNumber
101	a	b	0176543, 0176548
102	c	d	0165438
103	e	f	0176654, 0189756

ID	First Name	FamilyName	PhoneNumber
101	a	b	0176543, 0176548
102	c	d	0165438
103	e	f	0176654, 0189756

- **Multiple values** (even each one is atomic) exist in phone number.

Apparent Solution: Add more attribute

ID	First Name	FamilyName	Phone1	Phone2
101	a	b	0176543	0176548
102	c	d	0165438	
103	e	f	0176654	0189756

- Now it is in 1NF
- But it has **problems**: space wastage (many entries are null).
- Even a larger **design problem**: what if you want to add another phone number? And then another !!!!
- What is the ideal solution?

An Ideal Solution

ID	FirstName	FamilyName
101	a	b
102	c	d
103	e	f

ID	Phone No
101	0176543
101	0176548
102	0165438
103	0176654
103	0189756

- Now it is in 1NF.
- Multiple values can be added efficiently.

1NF: Problem with Integer Values

- Integer is Considered as **atomic** as long as it **has no sub-parts**
- Integer with sub-parts creates problems (e.g. ID= CSE001)
- Extra programming effort is needed to extract dept information and serial number from a given ID.
- Since we need to split the ID to get other information, **the ID is no longer called atomic**.
- So, the schema is NOT in 1NF.
- But if the schema **contains separate attributes** for each of the participating attribute of the ID (in this case: dept) then it is in 1NF. This incurs **redundancy** but is intentional and useful by the end-users.

Second Formal Forms (2NF)

Definition

A relation R is in 2nd Normal Form (2NF) if the following 2 conditions hold:

1. The relation R is in First Normal Form (1NF) (**i.e. all attributes are atomic**)
2. **No Partial Dependency** (P.Dep) exists in the relation R.

What is Partial Dependency (P.Dep)?

A Relation R is said to have Partial Dependency if

Proper Subset (PSS) of $CK_i \longrightarrow NonPrimeAttribute(NonPA)$

$(NonPA = R - PrimeAttribute)$

Second Formal Forms (2NF) By Example 1

Example 1 for 2NF Test. Given: R(A, B, C, D, E, F)

FD: A→B, B→C, C→D, D→E

(Here we assume each R is already in 1NF so, we need to test the 2ND condition only)

Sol^n: First we need to find all CKs (using method 3)

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

Now use FDs to ⇒ discard

use TR: A→B B→C ⇒ A→C

C→D ⇒ A→D

D→E ⇒ A→E

$A\cancel{BCDE}F^+ = $ all

$AF^+ = $ all

→ PSS ⇒ $A^+ = ABCD$ ✗ SK

$F^+ = F$ ✗ SK

So, AF is a CK.

Second Formal Forms (2NF) By Example 1 (Cont.)

Now need to check if more CK exist (i.e $DA \cap PA \neq \phi$)

dependent attributes

Prime "

$A \to B$ $B \to C$ $C \to D$

$D \to E$

$R(A, B, C, D, E, F)$

Here, $DA = B, C, D, E$

Upto now $PA = \{A, F\}$

(AF CK)

So, $DA \cap PA = \phi$

So, No other CK exists here.

Only CK is AF [upto this was required for condition 2]

$\boxed{\text{Condition 2}}$:

Now $CK = AF$

Non $PA = B, C, D, E$

i.e ($R - PA$)

Second Formal Forms (2NF) By Example 1 (End)

CK = AF

Non PA = B, C, D, E

PSS (CKi) → NonPA (any)

2ND condition

Now, we have A → B

PSS of (AF) : A, F

And B is in NonPA

So, PSS (CKi) → NonPA holds

A → B

Thus given R is NOT in 2NF. Proved

Second Formal Forms (2NF) By Example 2 (Begin)

2NF Example 2 | Given $R(A, B, C, D)$

FD: $AB \to CD$, $C \to A$, $D \to B$

Sol^n:

Step1 Find CK (take all) $ABCD^+ = \{A, B, C, D\}$ (RR)

$AB\cancel{C}\cancel{D}^+ =$ " use FD $\boxed{AB \to CD}$ to discard/eliminate

$AB^+ =$ "

PSS of AB: $A^+ = \{A, \}$ SK NO

$B^+ = \{B\}$ SK NO

So, $\boxed{AB}$ is a CK

Second Formal Forms (2NF) By Example 2 (Cont.)

Next step: check if more CK exist (use, DA ∩ PA)

Upto now, $CK_1 = AB$ so, $PA = \{A, B\}$

$DA = C, D, A, B$

Right side of FD

FD: $AB \to CD$, $C \to A$
$D \to B$

Replacement Process (RP) ⇒

So, $PA \cap DA = \{A, B\}$ (2 RP will occur.)

AB (already a CK)

- A $(C \to A)$
  - (C)
    - BC
- B $(D \to B)$
  - (D)
    - AD

AB (CK_1)

- BC
- AD

PSS: $B^+ = B$ ✗
$C^+ = C, A$ ✗

PSS:
$A^+: A$ ✗
$D^+: D, B$ ✗

So, $\boxed{BC}$ and $\boxed{AD}$ are also CK.

Second Formal Forms (2NF) By Example 2 (Cont.2)

FD $AB \to CD$
$C \to A$
$D \to B$

$CKs = \{AB, BC, AD\}$

So, PA (updatable each time a CK is found) $= \{A, B, C, D\}$

(Fixed) $DA = \{C, D, A, B\}$

So, $DA \cap PA = A, B, C, D = \{C, D\}$

(A, B have been used already)

Note steps

CKi
✓ AB (RP done)
? BC (RP pending)
? AD (")

FD $AB \to CD$
$\Rightarrow$ $AB \to C$
$AB \to D$

BC → C (AB) (as C is common in BC, D is not in BC) → AB

AD → D (AB) → AB

$AB \to D$

AB is already a CK so, process ends.

Second Formal Forms (2NF) By Example 2 (End)

Step: (upto now we computed all CKs)
Now need to check 2ND condition of 2NF

$\boxed{\text{PSS of } CK_i \rightarrow NonPA_j}$

CKs = AB, BC, AD | PA = {A, B, C, D}
So, NonPA = R − PA
= {A, B, C, D} − {A, B, C, D}
= ϕ

Since there is no NonPA the 2ND condition will not hold. So, the Relation is in 2NF.
(Done)

Third Normal Form (3NF)

Motivation

- 2NF states **"No Partial Dependency (P.Dep) exists in the relation R"** to get rid of **Redundancy** (and other associated problems).
- But still anomaly may appear in 2NF due to **NonPrime Transitivity Dependency**. (We will discuss the idea using an example).

Third Normal Form (3NF): One Motivating Example

ID	Name	BG	Division	Country	Weather
101	a	-	Dhaka	BNG	Mild
102	b	-	Dhaka	BNG	Mild
103	c	-	Dhaka	BNG	Mild
104	d	-	Rajshahi	BNG	Hot

Here ID is the SK and CK as well as PK. This Relation is in 2NF (you can check it). But here we observe the following NonPrime Transitivity Dependency:

$Weather \longrightarrow Division, Country$ This **FD introduces Redundancy and update anomaly** in the Relation.

Here, both LHS(Weather) and RHS(Division,Country) are Non Prime Attribute. **NonPrime Transitivity Dependency** will be explained soon.

Third Normal Form (3NF):Conditions

A Relation R is in 3NF if the following **2 conditions** are met:

1. R must be in 2NF (i.e. No Partial Dependency, **PSS of** $(CK_i) \longrightarrow NonPA$)
2. It must **not** contain any **Transitivity Dependency for Non Prime Attributes**, in other words there must not be any $FD : \alpha \longrightarrow \beta$ where both α and β are Non Prime Attributes.

Note 1: See the conditions are changing gradually from 2NF to 3NF:
in 2NF we should not have any $PA \longrightarrow NonPA$
while in 3NF we should not have any $NonPA \longrightarrow NonPA$. It is getting stricter.

Note 2: Third Normal Form (3NF) is considered adequate for normal relational database design because most of the 3NF tables are free of insertion, update, and deletion anomalies.

3NF Second Condition: Transitivity Non Prime Attribute Dep.

□ 2ND condition : explained (example)

SID	Name	Dept	DeptEst.
1	a	CSE	1998
2	b	EE	1997
3	a	CSE	1998

Transitive NonPA in both sides

Here SID is the only CK.

We observe, FD : SID → Dept (since SID CK so, SID → any combination)

and $\boxed{\text{Dept} \rightarrow \text{DeptEst}}$

⇒ SID → DeptEst (TR)

⇒ Here, both Dept and DeptEst are Non PA.

So, if we get any $\boxed{\alpha \rightarrow \beta}$ where α, β are both Non PA

⇒ $X \rightarrow \alpha$ and $\alpha \rightarrow \beta$

where X is the CK. [it introduces redundancy]

3NF: Example 1 (Begin))

Example 1 for 3NF:

$R(A, B, C, D)$ FD: $A \to B$, $B \to C$, $C \to D$

Sol^n: (First Find all CKs) i.e. CKs are needed to find PA that is used to find NonPA

Condition

we don't find any FD $\alpha \to \beta$ where both α, β are Non Prime Attribute (NonPA)

$ABCD^+ = $ all

discard $A\cancel{B}\cancel{C}\cancel{D}^+ = $ all (use TR: $A \to B, C, D$)

So, A is a SK

and also a SK (why? because there is no PSS of A)

simple

$\boxed{A \text{ is a SK}}$

3NF: Example 1 (Cont.)

- Next: is there any more CK?

FD:-
$A \to B$
$B \to C$
$C \to D$

Upto now, we got,

$A = CK$

So, $PA = \{A\}$

and $DA = \{B, C, D\}$

$\cap = \varphi$

So, No other CK possible.

3NF: Example 1 (End))

FD R(A, B, C, D)

A → B

B → C

C → D

So, we have
CK₁ = A

PA = {A}

Non PA = {B, C, D}

Now check against each FD :—

× A → B   A is PA, no need to check RHS.

✓ B → C   Both B, C are NonPA

✓ C → D   " C, D " "

So, R is NOT IN 3NF

Done

3NF: Example 2 (Begin))

Example 2 (3NF): Given $R(A, B, C, D, E, F)$

FD: $AB \to CDEF$, $BD \to F$

Sol^n:

step 1: Find all CKs.

$ABCDEF^+ = (all)$

(discard) $AB\cancel{C}\cancel{D}\cancel{E}\cancel{F}^+ =$ " $(AB \to CDEF)$

$AB^+ = (all)$

PSS (AB): $A^+ = A$ ✗ SK

$B^+ = B$ ✗ SK

so, AB is CK

Step 1.2 Any other CK?

CK = AB

PA = A, B

NonPA = C, D, E, F

3NF: Example 2 (End))

upto now, CK = AB, PA = {A, B} DA = CDEF (Right side)

$\boxed{PA \cap DA = \phi}$

So, no CK possible.

Now, apply 2ND condition - CK = AB, PA = A, B

$\boxed{\text{Non PA} = C, D, E, F}$

check each FD:

(NO) AB → CDEF (Non PA → Non PA exists ?)

Exists → BD → F (here, since D is a NonPrime BD as a whole is also NonPrime)

So, NonPA → Non PA

(i.e. AB → BD & BD → F

⇒ AB → F Transitivity NonPA)

$\boxed{\text{So, R is NOT in 3NF.}}$ Done

Boyce - Codd Normal Form (BCNF)

Motivation

- BCNF is the advance version of 3NF. It is **stricter** than 3NF and also called **3.5 NF**. BCNF was developed in 1974 by Raymond F. Boyce and Edgar F. Codd to **address certain types of anomalies** not dealt with by 3NF as originally defined.
- **Anomalies** occur in 3NF when we have **Multiple Overlapping Candidate Keys**.

Boyce - Codd Normal Form: Conditions

A Relation R is in BCNF if the following **2 conditions** are met:

1. R must be in 3NF (i.e. No Transitivity Dependency on NonPA (both sides))
2. For **each** Non-trivial Functional Dependency DF:$\alpha \longrightarrow \beta$, α **must be a Superkey (SK)**.

   Note: It is relatively easy to check, for a smaller R you can readily check the RHS and see if it is a SK or not. But here we will use our conventional process we used earlier.

BCNF By Example (Begin)

$\boxed{\text{BCNF Example 1}}$

R(A, B, C)

FD: A → B   B → C   C → A

Task: To test if R is in BCNF.

Solⁿ:

Note: BCNF removed the problem for multiple overlapping CKs

such as AB
BC
CD

Review conditions —

1) Must be in 3NF

2) For all non trivial dep. x → y
x must be a SK

Note: We'll use same method (3) to get all CKs.
It gives other info needed for BCNF (i.e SK)

BCNF By Example (Cont.)

$R(ABC)$
$A \to B \quad B \to C$
$C \to A$

$\cancel{A}\cancel{B}C^+ = \{A, B, C\}$
$A \to B$
$C \to A$

$\Rightarrow C^+ = \{all\}$ —— is a SK and CK

(i.e No need to check as No PSS exists)

Again $A\cancel{B}\cancel{C}^+ = all$
$B \to C$
$A \to B$

$\Rightarrow A^+ = \{all\}$ —— is a SK and CK

So, $PA = \{A, C\}$ $DA = A, B, C$
$\alpha = PA \cap DA = A, C$

So, for $CK_1 = C$ $CK_2 = A$

BCNF By Example (End)

A

$C \to A$

A

(C)

C (already done)

C

$B \to C$

C

(B)

B (new) → is a SK and CK

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

RHS

So, CKs are {A, B, C} also SK

Now check RHS = {A, B, C} and all of them are SK, $\boxed{\text{so, it is in BCNF.}}$

2NF: Real Example

Real Example $\boxed{2NF}$

2NF condition:-

No such dependency-

$\boxed{PSS(CK_i) \rightarrow NonPA}$

Because such FD introduces Redundancy.

Student-Course

SID	CID	Credit
1	CSE101	3
2	CSE102	4
1	CSE103	3
3	CSE101	3
2	CSE101	3

Here, CK = (SID, CID) (composite)

Now, we observe, $\boxed{CID \rightarrow Credit}$

And CID is a PSS of CK.

So, Right decomposition -

StudentsCourse

(CK) SID	CID (F.K)
1	CSE101
2	CSE102

Courses

CID	Credit
CSE 101	3
CSE102	4
CSE103	3

3NF: Real Example

Real Example $\boxed{3NF}$

3NF condition

No such Transitivity dependency (TD)

NonPA → NonPA

Students

SID	Dept	DeptBudget
1	CSE	110
2	EEE	120
3	CSE	110

Here we observe - CK = SID

and Dept → DeptBudget

∴ NonPA = Dept, Dept Budget

→ Both Dept and DeptBudget are NonPA.

Explanation : (TD)

SID → Dept and Dept → DeptBudget

⇒ SID → DeptBudget

It means if you have SID and dept then you can deduce deptbudget, so deptbudget in this R is Redundant.

BCNF: Real Example

Real Example

$\boxed{\text{BCNF}}$ Suppose, students can take courses. One course may be conducted by multiple teachers, But one teacher can take only one course.

BCNF conditions

For each Non Trivial FD $\boxed{\alpha \rightarrow \beta}$

α is a SK

Students

SID	CID	TID
1	Java	KH
1	C	SH
2	DB	ARMK
3	Java	HK
4	DB	ANS
4	DB	ARMK
5	C	SH

Here, CK = (SID, CID)

observe —

$\boxed{TID \rightarrow CID}$

But TID is not a SK.

So, it is not in BCNF

So, we decompose —

SID	TID
1	KH
1	SH
2	ARMK
3	HK
4	ANS
4	ARMK
5	SH

TID	CID
ARMK	DB
KH	Java
SH	C
HK	Java

Thank You