

Chapter 7: Normalized Database Design Part 2

Abu Raihan Mostofa Kamal

Professor, CSE Department
Islamic University of Technology (IUT)

October 30, 2023

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

Closure of Attributes

oooooooooooooooooooo

Normalization

oo

Fist Normal Form and Atomic Domain

ooooo

Second Normal Form

oooooooo

Third Normal Form

oooooooooo

Boyce - Codd Normal Form (BCNF)

oooooooooo

Chapter Outline

Closure of Attributes

Normalization

Fist Normal Form and Atomic Domain

Second Normal Form

Third Normal Form

Boyce - Codd Normal Form (BCNF)



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: $A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E$

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

We can reduce:

- $A \rightarrow C$ (transitivity)
 - $A \rightarrow A$ (reflexivity)
 - $A \rightarrow D$ (transitivity)
 - $A \rightarrow E$ (transitivity)
 - $A \rightarrow A, B, C, D, E$ (union)
 - So, $A^+ = \{A, B, C, D, E\}$
- $B \rightarrow D$ (tran)
 - $B \rightarrow E$ (tran)
 - $B \rightarrow B$
 - $B \rightarrow B, C, D, E$ (U.)
 - So, $B^+ = \{B, C, D, E\}$
- $C \rightarrow E$
 - $C \rightarrow C$
 - $C \rightarrow C, D, E$
 - 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: $A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E$**

Method 2: Simplified approach

Basic Principle

First use **Reflexivity Rule** for a given attribute set, then iterate through the **given FDs** to add element in the set. When you finish the last FD, re-iterate if new element can be added. **Simple trick:** if $\alpha \rightarrow \beta$ 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 $FD: A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow E$

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 Candidate 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)



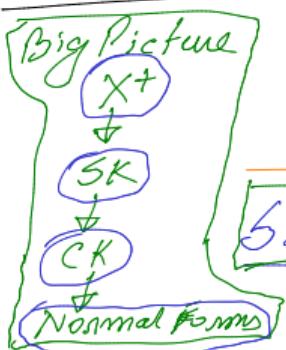
Recap of Method 1 and Method 2 (Cont.)

Example (same):

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

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

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



Solution:

$$\begin{aligned}
 A^+ &= \{ A \} \quad \text{Ref. Rule (RR)} \\
 &= A, B \quad (\text{FD: } A \rightarrow B) \\
 &= A, B, C \quad (\text{FD: } B \rightarrow C) \\
 &= A, B, C, D \quad (\text{FD: } C \rightarrow D) \\
 &= A, B, C, D, E \quad (\text{FD: } D \rightarrow E) \\
 &= \{ A, B, C, D, E \} \quad (\text{all attributes})
 \end{aligned}$$

So, A is a Superkey (SK)



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

Method 2: Another Example Given is $R(A, B, C, D, E)$

Solution: Lets start with A

$SK = \text{SuperKey}$
 $CK = \text{Candidate Key}$ so, $A^+ = \{A\}$ (RR)
 $= \{A, B\}$ $FD: A \rightarrow B$

Thus, $BC^+ = \{B, C\}$ $\left[\begin{array}{l} \alpha \rightarrow \beta \text{ means} \\ \alpha \text{ can determine } \beta \end{array} \right]$

Now Lets check

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

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

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

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.

$$AB \not\Rightarrow AB$$

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



Method 3: Simple Example

Method 3 (example)
 (Step) ~~discard step~~

Given: $R(A, B, C, D, E)$
 FD $\{A \rightarrow B, D \rightarrow E\}$

Let's start with A, B, C, D, E (a1)

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

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

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

$$\Rightarrow ACDE^+ = \{A, B, C, D, \cancel{E}\} \quad \text{right part}$$

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

$$\Rightarrow ACD^+ = \{A, B, C, D, E\} \quad (SK) \checkmark$$



Method 3: Simple Example Cont.

$$50, \quad ACDT = \{A, B, C, D, E\} \quad (5k) \checkmark$$

Need to test all proper subsets are 5K or not
 finally say ACD is a C-K: $\xleftarrow{\{A \rightarrow B\}, \{D \rightarrow E\}}$

ACD all proper subsets are

$$(2^2 - 2) = 6$$

1c need to check

CD

AD

A
-

C
D

1

$$AC^+ = \{A, C, B\} \cdot \times SK$$

$$CD^T = \{C, D, E\} \times SK$$

$$ADT = \{A, D, E, B\} \times SK$$

$$AT = \{A, B\} \times SK$$

$$C^+ = \{c, 3\} \times SK$$

$$D = \{D, E\}$$

Note: Here we have 6 proper subsets to check

Here we have 6 proper subsets.
In contrast of earlier version $A B C D E \simeq 2^5 - 2 = 30$

"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⁺)

$$\text{So, upto now } PAs = \{A, C, D\}$$

also called Dep. Attribute (DA)

Principle : For a given CK, form another set called Dependent Set (DS) / DA [e.g. $A \rightarrow B \rightarrow E$, $D \rightarrow E$]

IF $PAs \cap DA = \emptyset$ then CK is the only CK

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 $(A \rightarrow B, D \rightarrow E)$

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

(fixed)
ACD was CK
in step 1

$$\text{so, } PA \cap DA = \{A, C, D\} \cap \{B, E\} = \emptyset$$

$$= \emptyset$$

(Proved)

So, ACD is the only CK



Method 3: Standard Example (Begin)

New Example for the case

$$CK \cap DS \neq \emptyset$$

Given: $R(A, B, C, D)$ & $FD: A \rightarrow B, B \rightarrow C, C \rightarrow A$

Step 1

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

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

discard on FR: $A \rightarrow B \& B \rightarrow C \Rightarrow A \rightarrow C$

(since both A, B are there and
 B can be deduced from A
so, B is redundant)

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

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

$$D^+ = D \text{ NOT a SK}$$

So, AD is a candidate key (CK).

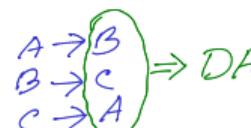


New step: Now we need to check if some CK exist.

Upto now - PAs = A, D

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

Now, $DA \cap PAs = A$ (Now we look for

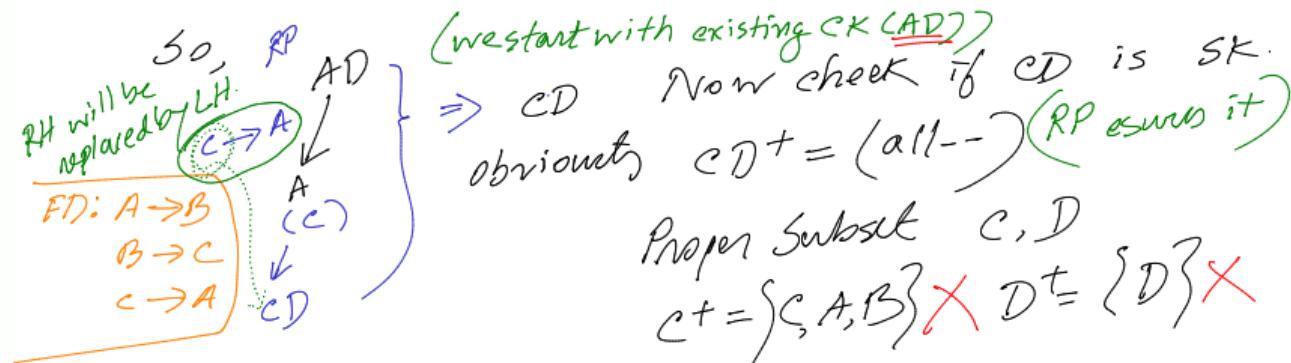


Here, FD: $C \rightarrow A$

$\alpha \rightarrow A$ them
replace A with α ,
 α will replace A



Method 3: Standard Example Cont.



Proper Subset C, D

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

so, CD is a CK.

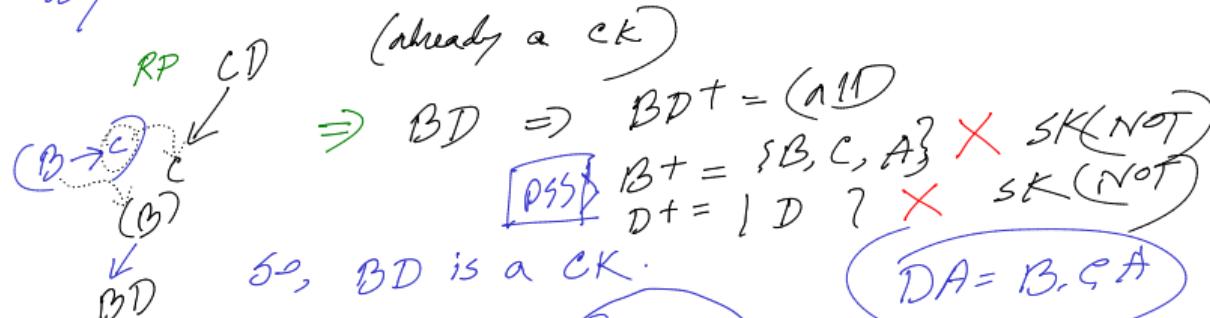
Now $PAs = \{A, D, C\}$
Check $PAs \cap DA = A, C \Rightarrow C$ since A has already been considered

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



Method 3: Standard Example Cont.

By same prcn: FD: $B \rightarrow C$ FD $A \rightarrow B, B \rightarrow C, C \rightarrow A$

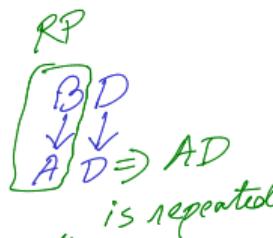


Now, $PAs = \{A, D, C, B\}$

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

so, we look for FDs.

$A \rightarrow B$



so, process ends.



Method 3: Standard Example (End)

So, Finally we get

$$\text{set of CKs} = \{AD, CD, BD\}$$

Summary of Method 3 End of example.
 (which is used)

1. Start with all attributes $(A_1 A_2 \dots A_n)^+$
2. Start discard using given or implied FD
 $|(A_i \dots A_j)| \leq |A_1 \dots A_n|$
3. compute $\underline{(A_i \dots A_j)}^+ \Rightarrow$ check all proper subsets
4. IF it is a CK then form P.A $\xrightarrow{\text{also form DA = ?}}$ ^{fixed}
 $\xrightarrow{\text{updateable}}$



Closure of Attributes

oooooooooooooooooooo●ooooooo

Normalization

oo

First Normal Form and Atomic Domain

ooooo

Second Normal Form

oooooooo

Third Normal Form

oooooooo

Boyce - Codd Normal Form (BCNF)

oooooooo

Method 3: Algorithm Review

5. Find $PA \cap DA = x_1, x_2 \dots x_k \quad k=1, 2, \dots$

and we replacing by $x \rightarrow x$
and go to step 3.



Method 3: More Example A

2 More examples —

Example A

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

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

Solution

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

discard

$AB \cancel{CD}^+ = " \quad (AB \rightarrow CD)$

$AB^+ = "$

Proper Subsets (PSS)

PSS of $AB \Rightarrow A, B$

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

so, AB is a CK. ————— (i)



Method 3: More Example A (Cont.)

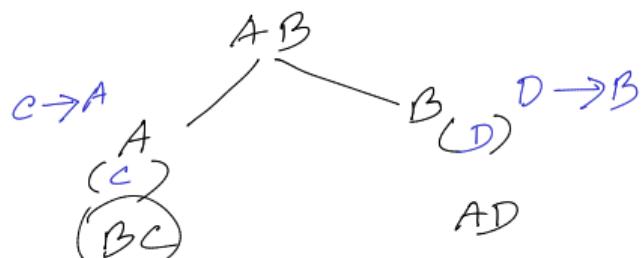
~~Step 2 RP~~
~~AB is a eA~~

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

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

$$\text{so, } PA \cap DA = A, B$$

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



\rightarrow PSS $B, C \Rightarrow BT = \{B\}$ NOT SK
 $c^+ = \{C, A\}$ NOT SK

so, BC is a CK ————— (ii)



Method 3: More Example A (Cont.2)

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

FD: $AB \rightarrow cD, D \rightarrow B$
 $c \rightarrow A$

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

so, AD is a CK

iii

Now, updated PA = $\{A, B, C, D\}$

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

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

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

$$\cancel{ABCD^+} = "u"$$

$$CD^+ = "$$

$$\begin{aligned} PSS \Rightarrow C^+ &= \{C\} ? && \text{NOT SK} \\ D^+ &= \{D, B\} && \text{NOT SK} \end{aligned}$$

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 (Advo)

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

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

Stone discord is a
tricky one as A. Axioms
will be used.

Solⁿ: — $ABCDEF^+ = \text{all}$

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

~~$ABCDEF^+$~~ = all

$ABC^+ =$

We use $C \rightarrow DE \Rightarrow C \rightarrow D$ and $C \rightarrow E$
(decomposition)
 Again given $E \rightarrow F$ and $AB \rightarrow C$

$\Rightarrow AB \rightarrow F$ (TR)

~~ABC^+~~ =
 $AB^+ =$



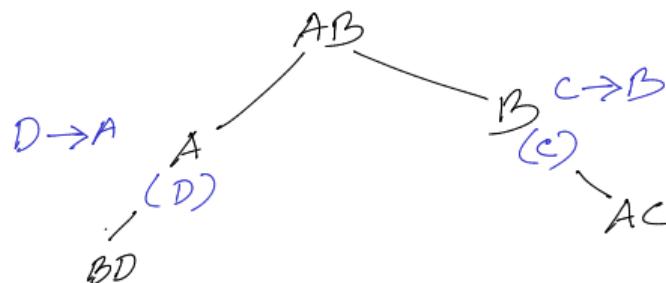
Method 3: More Example B (Cont.)

PSS $A^+ = \{A\}$, $B^+ = \{B\}$ } NOT SK
 } NOT SK
 so, AB is a CK.

$$\begin{array}{l} AB \rightarrow C \\ C \rightarrow DE \\ E \rightarrow F \\ D \rightarrow A \\ C \rightarrow B \end{array}$$

FD:

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)

□ $BD^+ = \text{--- all}$

PSS: $B^+ = \checkmark$ NOT SK
 $D^+ = \checkmark$ NOT SK

so, BD is a CK

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

□ $AC^+ = \text{all}$

PSS: $A^+ = \{A\}$
 $C^+ = \{C, B, D, E, F, A\}$ is a SK
 also a CK because
 PSS is here \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:** Height, 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 → NonPrimeAttribute(NonPA)

(NonPA = R – PrimeAttribute)



Second Formal Forms (2NF) By Example 1

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

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

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

So $\stackrel{n}{\exists} CKs$ First we need to find all CKs (using method 3)

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

Now use FDs to \Rightarrow use TR $A \rightarrow B$ $B \rightarrow C \Rightarrow A \rightarrow C$
discard $C \rightarrow D \Rightarrow A \rightarrow D$
 $D \rightarrow E \Rightarrow A \rightarrow E$

$$\begin{aligned} ABCDEF^+ &= \text{all} \\ AFT^+ &= \text{all} \end{aligned} \quad \left. \begin{array}{l} \text{PSS} \Rightarrow A^+ = ABCD^+ \\ \qquad \qquad \qquad F^+ = FXSK \end{array} \right\}$$

so, AF is a CK.



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

Now need to check if more CK exist (i.e $DA \cap PA \neq \emptyset$)
 $A \rightarrow B$ $B \rightarrow C$ $C \rightarrow D$ $D \rightarrow E$
 $R(A-B, C, D, E, F)$

Here, $DA = B, C, D, E$ dependent attributes
 $PA = \{A, F\}$ prime w.r.t DA

Upto now $PA = \{A, F\}$
(AF CK) so, $DA \cap PA = \emptyset$

so, No other CK exists here.

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

Condition 2 :

Now CK = AF
 $NonPA = B, C, D, E$

(R - PA)
i.e



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

$$CK = \underline{AF}$$

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

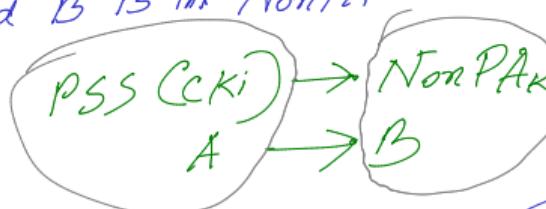
Now we have

$$A \rightarrow B$$

$$\text{PSS of } \underline{AF} : A$$

And B is in Non PA

so,



holds

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

Solⁿ:

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

$ABCPD^+ = " "$

$AB^+ = " "$

use FD $AB \rightarrow CD$
to discard/eliminate

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, DAN PA)

Upto now, CK₁ = AB so, PA = {A, B}

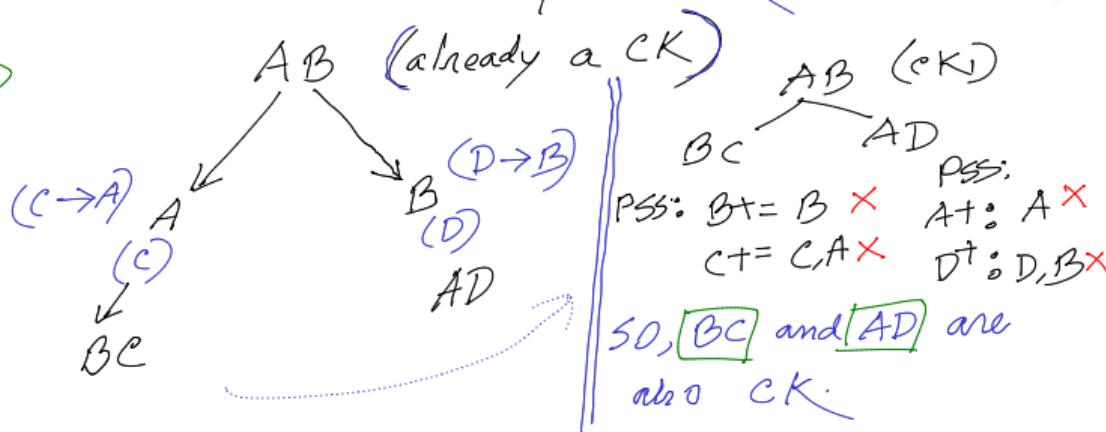
DA = C, D, A, B

Right side of FD

FD: AB → CD, C → A
D → B

Replacement Process
(RP) ⇒

so, PA ∩ DA = {A, B} (2 RP will occur.)



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

$$\begin{array}{l} \text{FD } AB \rightarrow CD \\ C \rightarrow A \\ D \rightarrow B \end{array}$$

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

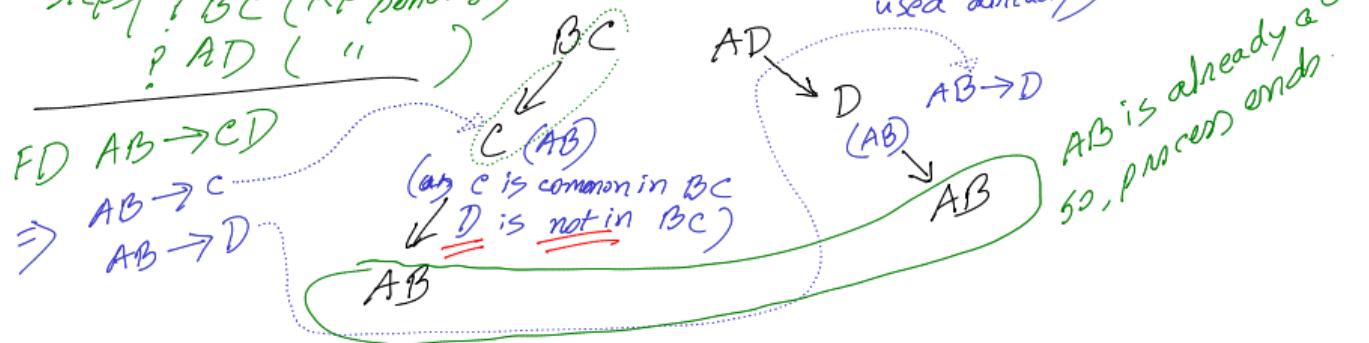
so, PA (updateable each time a CK is found) = $\{A, B, C, D\}$

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

Note | CK_i ✓ AB (RP done)
 steps | ? BC (RP pending)
 ? AD ("")

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

(A, B have been used already)



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

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

PSS of CK_i → NonPA_j

$$\text{CKs} = AB, BC, AD \quad | \quad PA = \{A, B, C, D\}$$

$$\text{so, NonPA} = R - PA$$

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

$$= \emptyset$$

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 \rightarrow *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** \rightarrow **NonPA**)
2. It must **not** contain any **Transitivity Dependency for Non Prime Attributes**, in other words there must not be any $FD : \alpha \rightarrow \beta$ 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** \rightarrow **NonPA**

while in 3NF we should not have any **NonPA** \rightarrow **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 :

$SID \rightarrow Dept$ (since $SID \rightarrow CK$)
and $Dept \rightarrow DeptEst$ (so, $SID \rightarrow any combination$)

$\Rightarrow SID \rightarrow DeptEst$ (TR)

Here, both Dept and DeptEst are Non PA.
so, if we get any $\alpha \rightarrow \beta$ where α, β are both Non PA

$\Rightarrow X \rightarrow \alpha$ and $\alpha \rightarrow \beta$

where X is the CK. [it introduces redundancy]



3NF: Example 1 (Begin))

Example 1 for 3NF^o

Solⁿ: (First Find all CKs)

R(A, B, C, D) $\stackrel{FD: A \rightarrow B, B \rightarrow C}{=} C \rightarrow D$

i.e.
CKs are needed
to find PA that
is used to find
NonPA

Condition

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

$ABCD^+ = \text{all}$

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

so, A is a SK

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

A is a SK

simple



3NF: Example 1 (Cont.)

□ Next: is there any more CK?

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

Upto now, we got,

$A = CK$

so, $PA = \{A\}$
 and $DA = \{B, C, D\}$

$\overline{\cap = \emptyset}$

So, No ^{other} CK possible.



3NF: Example 1 (End))

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

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow D$

so, we have
 $CK = A$

Now check against
each FD :-

$PA = \{A\}$

$NonPA = \{B, C, D\}$

$\times A \rightarrow B$ A is PA , no need to check RHS.

$\checkmark B \rightarrow C$
 $\checkmark C \rightarrow D$

Both B, C are $NonPA$
" " "

so, R is NOT IN 3NF.

Done



3NF: Example 2 (Begin))

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

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

Solⁿ step 1: find all CKs.

$$ABCDEF^+ = (\text{all})$$

(discard) $AB \cancel{CDEF}^+ = " \quad (AB \rightarrow CDEF)$

$$AB^+ = (\text{all})$$

PSS (AB): $A^+ = A \quad \times \text{SK}$
 $B^+ = B \quad \times \text{SK}$

so, AB is CK

Step 1.2 Any other CK?

$$\begin{aligned} CK &= AB \\ PA &= A, B \\ \text{NonPA} &= C, D, E, F \end{aligned}$$



3NF: Example 2 (End))

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

$$PA \cap DA = \emptyset$$

so, no CK possible.

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

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

check each FD:

$$(NO) AB \rightarrow CDEF \quad (\text{Non PA} \rightarrow \text{Non PA})$$

$$\text{Exists } BD \rightarrow F \quad (\text{here, since } D \text{ is a NonPrime})$$

$$\text{so, Non PA} \rightarrow \text{Non PA}$$

exists?
BD as a whole is also NonPrime

$$(i.e. AB \rightarrow BD \& BD \rightarrow F \\ \Rightarrow AB \rightarrow F \text{ Transitivity NonPA})$$

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 $DF: \alpha \rightarrow \beta$, 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)

BCNF Example 1

$R(A, B, C)$

FD: $A \rightarrow B$ $B \rightarrow C$ $C \rightarrow A$

Task: To test if R is in BCNF.

Solⁿ: Note: BCNF removed the problem for multiple overlapping CKs

Review conditions—
 1) Must be in 3NF

such as



2) For all non trivial dep. $X \rightarrow Y$

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.)

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

$$A \rightarrow B$$

$$C \rightarrow A$$

$$\Rightarrow C^+ = \{\alpha\} \quad \text{--- is a SK and CK}$$

Again $\cancel{ABC}^+ = \alpha\}$

$$B \rightarrow C$$

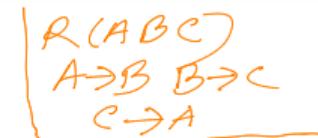
$$A \rightarrow B$$

$$\Rightarrow A^+ = \{\alpha\} \quad \text{--- is a SK and CK}$$

$$\text{so, } PA = \{A, C\} \quad DA = A, B, C$$

$$\alpha = PA \cap DA = A, C$$

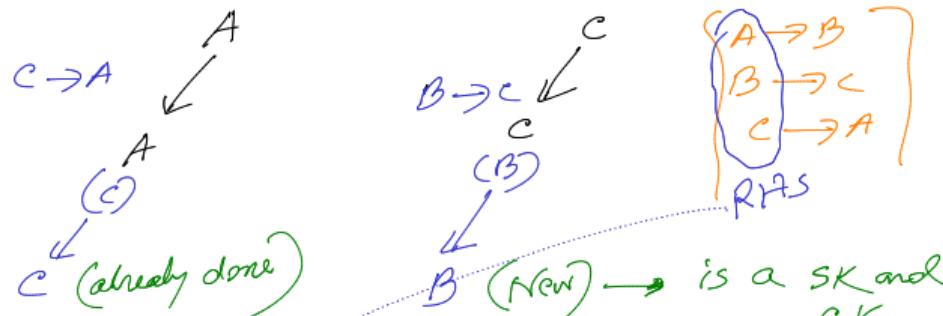
so, for $CK_1 = C$ $CK_2 = A$



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



BCNF By Example (End)



so, CKs are $\{A, B, C\}$ also SK

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



2NF: Real Example

Real Example

2NF

2NF condition:-

No such dependency.

$PSS(CR) \rightarrow NonPA$

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

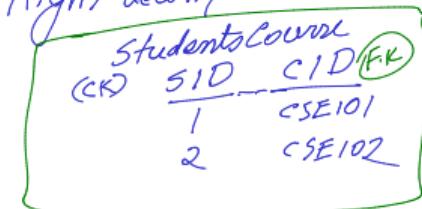
Here, CK = (SID, CID) (composite)

Now, we observe,

$CID \rightarrow credit$

And CID is a PSS of CK.

So, Right decomposition -



Courses

CID	credit
CSE101	3
CSE102	4
CSE103	3



3NF: Real Example

few Example (3NF)

3NF condition

No such Transitivity dependency (TD)

$\text{NonPA} \rightarrow \text{NonPA}$

Explanation of (TD)

$SID \rightarrow Dept$ and $Dept \rightarrow DeptBudget$

$\Rightarrow SID \rightarrow DeptBudget$

It means if you have SID and $Dept$

then you can deduce $DeptBudget$, so $DeptBudget$ in this R is Redundant.

Students

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

Here we observe -

$CK = SID$
and $Dept \rightarrow DeptBudget$

$\therefore \text{NonPA} = \text{Dept}, \text{Dept Budget}$

\Rightarrow Both $Dept$ and $DeptBudget$ are NonPA.



BCNF: Real Example

Real Example

BCNF

Suppose, students can take courses. One course may be conducted by multiple teachers, But one teacher can take only one course.

Students

BCNF conditions

For each Non Trivial FD
 $\alpha \rightarrow \beta$

$\underline{\underline{\alpha}}$ is a $\underline{\underline{SK}}$

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—

$TID \rightarrow CID$

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



Closure of Attributes

oooooooooooooooooooo

Normalization

oo

Fist Normal Form and Atomic Domain

ooooo

Second Normal Form

oooooooo

Third Normal Form

oooooooo

Boyce - Codd Normal Form (BCNF)

oooooooo●

Thank You

