

# فصل هفتم: نرمال سازی

## (Normalization)

(مطالب امتحان تا انتهای زیر بخش ۶ از فصل ۷ می باشد.)



درس پایگاه داده  
دانشگاه صنعتی نوشیروانی باپل  
مهندی عمادی  
[m.emadi@nit.ac.ir](mailto:m.emadi@nit.ac.ir)



# Outline

- **Features of Good Relational Design**
- **Functional Dependencies**
- **Decomposition Using Functional Dependencies**
- **Normal Forms**
- **Functional Dependency Theory**
- **Algorithms for Decomposition using Functional Dependencies**
- **Decomposition Using Multivalued Dependencies**





# Overview of Normalization





# Features of Good Relational Designs

- Suppose we combine *instructor* and *department* into *in\_dep*, which represents the natural join on the relations *instructor* and *department*

<i>ID</i>	<i>name</i>	<i>salary</i>	<i>dept_name</i>	<i>building</i>	<i>budget</i>
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

- There is repetition of information
- Need to use null values (if we add a new department with no instructors)





# A Combined Schema Without Repetition

- Not all combined schemas result in repetition of information

- Consider combining relations

- ▶  $\text{sec\_class(sec\_id, building, room\_number)}$  and

- ▶  $\text{section(course\_id, sec\_id, semester, year)}$

- into one relation

- ▶  $\text{section(course\_id, sec\_id, semester, year, building, room\_number)}$

- No repetition in this case





# Decomposition

- The only way to avoid the repetition-of-information problem in the *in\_dep* schema is to decompose it into two schemas – *instructor* and *department* schemas.

- Not all decompositions are good. Suppose we decompose

*employee*(*ID*, *name*, *street*, *city*, *salary*)

into

*employee1* (*ID*, *name*)

*employee2* (*name*, *street*, *city*, *salary*)

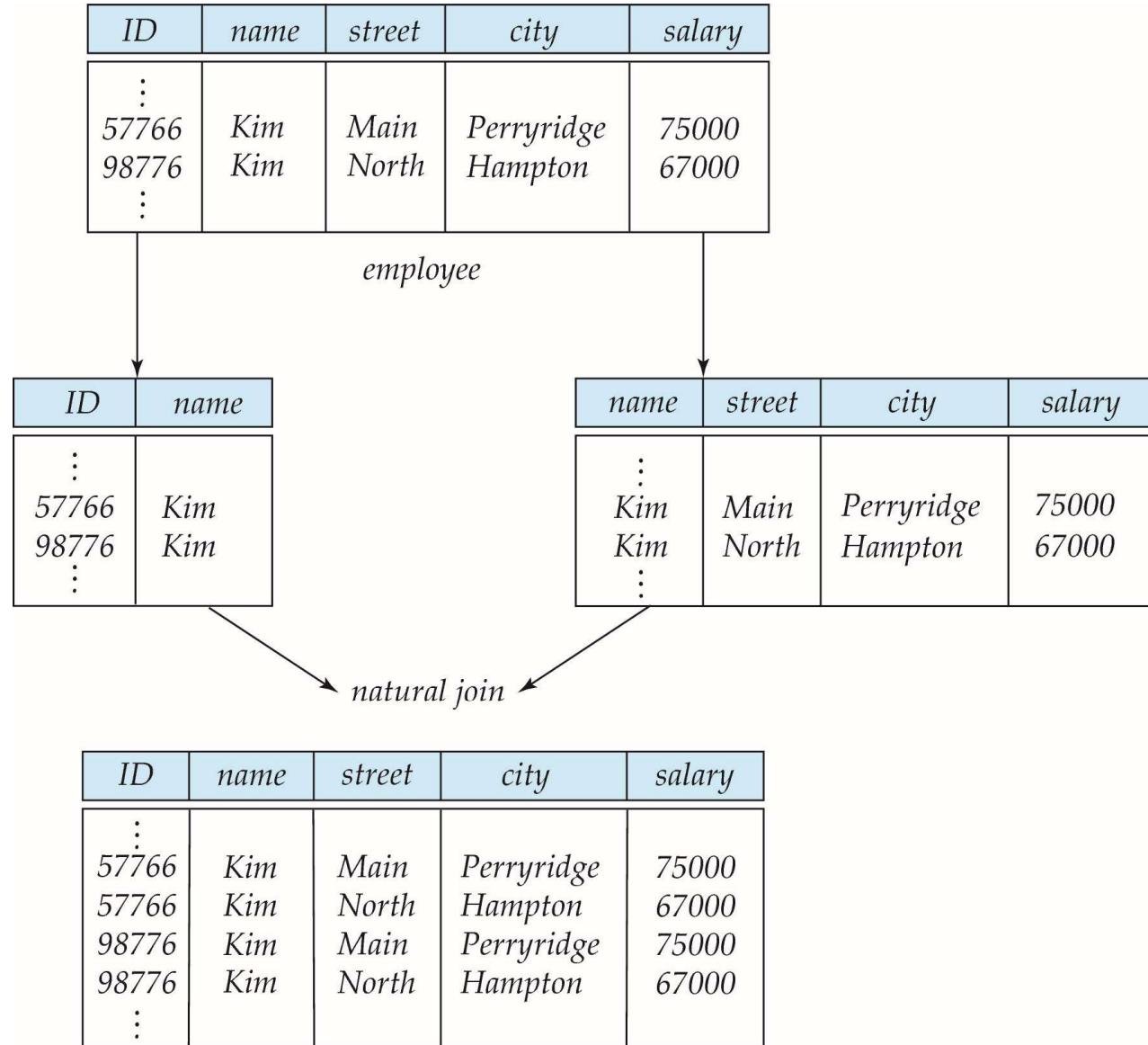
The problem arises when we have two employees with the same name

- The next slide shows how we lose information -- we cannot reconstruct the original *employee* relation -- and so, this is a lossy decomposition.





# A Lossy Decomposition





# Lossless Decomposition

- Let  $R$  be a relation schema and let  $R_1$  and  $R_2$  form a decomposition of  $R$ . That is  $R = R_1 \cup R_2$
- We say that the decomposition is a **lossless decomposition** if there is no loss of information by replacing  $R$  with the two relation schemas  $R_1 \cup R_2$
- Formally,

$$\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$$

- And, conversely a decomposition is lossy if

$$r \subset \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) = r$$





# Example of Lossless Decomposition

- Decomposition of  $R = (A, B, C)$

$$R_1 = (A, B) \quad R_2 = (B, C)$$

$r$	$\Pi_{A,B}(r)$	$\Pi_{B,C}(r)$
$\begin{array}{ c c c } \hline A & B & C \\ \hline \alpha & 1 & A \\ \beta & 2 & B \\ \hline \end{array}$	$\begin{array}{ c c } \hline A & B \\ \hline \alpha & 1 \\ \beta & 2 \\ \hline \end{array}$	$\begin{array}{ c c } \hline B & C \\ \hline 1 & A \\ 2 & B \\ \hline \end{array}$

$\Pi_A(r) \bowtie \Pi_B(r)$
$\begin{array}{ c c c } \hline A & B & C \\ \hline \alpha & 1 & A \\ \beta & 2 & B \\ \hline \end{array}$





# مشکلات Redundancy

- ذخیره تکراری
- آنومالی به روز رسانی (Update)
- آنومالی درج (Insert)
- آنومالی حذف (Delete)

- آنومالی در عملیات درج به هریک از سه وضع زیر گفته می‌شود:
  - عدم امکان انجام یک عمل (که منطقاً باید قابل انجام باشد)
  - بروز پیامد بد پس از انجام یک عمل
  - بروز فزونکاری در سیستم در انجام یک عمل





# صورت‌های نرمال (Normal Form)

- صورت نخست نرمال (**1NF**)
- صورت دوم نرمال (**2NF**)
- صورت سوم نرمال (**3NF**)
- صورت نرمال بایس-کاد (**BCNF**)
- صورت چهارم نرمال (**4NF**)
- صورت پنجم نرمال (**5NF**)
- صورت نرمال میدان-کلیدی (**DKNF**)
- صورت نرمال تحدید-اجتماع (**RUNF**)





## وابستگی تابعی

### Functional Dependency

تعريف - فرض کنید که  $R$  یک متغیر رابطه‌ای و  $A$  و  $B$  دو زیرمجموعه دلخواه از عنوان  $R$  باشند. می‌گوییم  $B$  با  $A$  وابستگی تابعی دارد و چنین نمایش می‌دهیم:

$$A \rightarrow B$$

اگر و فقط اگر در هر میدان ممکن از متغیر رابطه‌ای  $R$ , به هر مقدار  $A$  فقط یک مقدار  $B$  متناظر باشد.

به ازای یک مقدار خاص از  $A$  حتماً یک مقدار مشخصی از  $B$  خواهیم داشت.

**A**: دترمینان  
**B**: وابسته





## وابستگی تابعی بدیهی (نامهم)

### Trivial FD

اگر در  $R(C_1, C_2, C_3, \dots)$  داشته باشیم:

$$B = \{C_1\} \text{ و } A = \{C_1, C_2\}$$

در این صورت:  $A \rightarrow B$  یک وابستگی تابعی  $B \subseteq A$  نامهم است. به بیان دیگر اگر  $B$  زیرمجموعه‌ای از  $A$  باشد، در این صورت  $A \rightarrow B$  یک وابستگی بدیهی است.





# قواعد استنتاج آزمترانگ

فرض:  $A, B, C$  و  $D$  زیرمجموعه‌هایی از صفات رابطه  $R$  باشند. قواعد زیر برقرارند:

1. قاعده انعکاس: اگر  $B \subseteq A$  آنگاه:  $A \rightarrow B$

2. قاعده تعدد (تراگذری): اگر  $A \rightarrow C$  و  $B \rightarrow C$  آنگاه  $A \rightarrow B$

3. قاعده افزایش: اگر  $(A, C) \rightarrow (B, C)$  آنگاه  $A \rightarrow B$

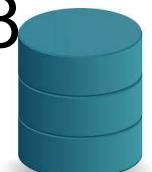
4. قاعده تجزیه: اگر  $A \rightarrow B$  و  $A \rightarrow C$  آنگاه  $A \rightarrow (B, C)$

5. قاعده اجتماع: اگر  $A \rightarrow B$  و  $A \rightarrow C$  آنگاه  $A \rightarrow (B, C)$

6. قاعده ترکیب: اگر  $(A, C) \rightarrow (B, D)$  آنگاه  $A \rightarrow B$  و  $C \rightarrow D$

7. قاعده شبه تعدد: اگر  $(A, C) \rightarrow D$  آنگاه  $A \rightarrow B$  و  $(C, B) \rightarrow D$

8. قاعده یگانگی عمومی: اگر  $A \cup (C - B) \rightarrow (B, D)$  آنگاه  $A \rightarrow B$  و  $C \rightarrow D$





- **Reflexivity** (انعکاس)
- **Transitivity** (تعدی یا تراگذری)
- **Augmentation** (افزایش)
- **Decomposition** (تجزیه)
- **Union** (ترکیب)





## مجموعه کاهش ناپذیر وابستگی های تابعی

مجموعه ای از وابستگی های تابعی  $R$ , به نام  $F$  را کاهش ناپذیر گوییم اگر:

1. در  $F$  وابستگی تابعی افزونه نباشد.
2. در سمت راست هر  $FD$  از  $F$  صفت ساده وجود داشته باشد.
3. هیچ صفتی در سمت چپ  $FD$  های  $F$  افزونه نباشد.





# Functional Dependencies

R	A	B	C	D	E	F
a1	b1	c1	d1	e1	f1	
a1	b1	c2	d1	e2	f3	
a2	b1	c2	d3	e2	f3	
a3	b2	c3	d4	e3	f2	
a2	b1	c3	d3	e4	f4	
a4	b1	c1	d5	e1	f1	

Dependencies for  
this relation:

$A \rightarrow B$  •

$A \rightarrow D$  •

$B,C \rightarrow E,F$  •

Do they all hold  
in this instance of  
the relation R?

- Functional dependencies are specified by the database programmer based on the intended meaning of the attributes.





## وابستگی تابعی قام (کامل)

اگر  $X$  و  $Y$  دو زیرمجموعه از مجموعه عنوان رابطه  $R$  باشند، می‌گوییم  $Y$  با  $X$  وابستگی تابعی **قام** دارد و چنین نشان می‌دهیم:  $X \Rightarrow Y$

اگر و فقط اگر  $Y$  با  $X$  وابستگی داشته باشد. ولی با هیچ زیرمجموعه‌ای از  $X$  وابستگی تابعی نداشته باشد.





# Normal Forms





## رابطه 1NF

تعریف- رابطه ای **1NF** است اگر هر صفت خاصه آن در هر تاپل، تک مقداری باشد، به بیان دیگر، صفت چند مقداری نداشته باشد.

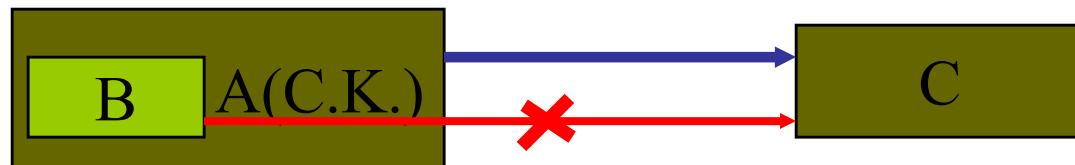
در حالت کلی با مدل منطقی رابطه ای همیشه رابطه ها در **1NF** هستند.





## رابطه 2NF

تعریف - رابطه‌ای 2NF است اگر اولاً 1NF باشد و ثانیاً تمام صفات غیرکلید با کلید کاندید وابستگی تابعی تام داشته باشند. به عبارت دیگر هر صفت غیرکلید با کلید کاندید بطور کاهش ناپذیر وابسته باشد.





## رابطه 2NF

تعریف بیان شده برای رابطه دارای کلید ترکیبی آمده در  
حالتی که رابطه ما کلید ترکیبی نداشته باشد حتما 2NF  
است





## تبدیل ۲NF به ۱NF

<b><u>StudentId</u></b>	<b><u>Courseld</u></b>	<b>Dept</b>	<b>Grade</b>
12	CS101	IE	B-
12	CS102	IE	C+
12	IE201	IE	C-
7	CS102	CS	A
7	CS352	CS	D

<b><u>StudentId</u></b>	<b><u>Courseld</u></b>	<b>Grade</b>
12	CS101	B-
12	CS102	C+
12	IE201	C-
7	CS102	A
7	CS352	D

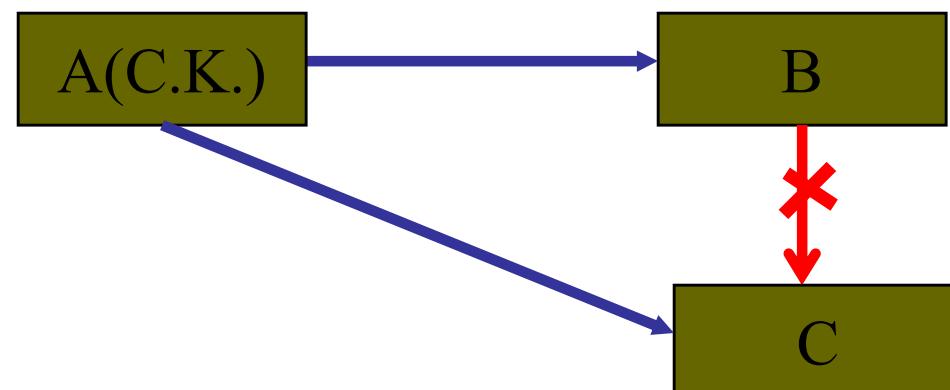
<b><u>StudentId</u></b>	<b>Dept</b>
12	IE
7	CS





## رابطه 3NF

تعريف- رابطه‌ای 3NF است اگر 2NF باشد و هر صفت غیر کلید با کلید کاندید، وابستگی تابعی بی‌واسطه داشته باشد.





## 3NF

<u>StudentId</u>	<u>S_Name</u>	Dept	DeptID
12	Ali	IE	1
13	Reza	IE	1
14	Behrooz	IE	1
7	Reza	CS	2
8	Zahra	CS	2

$\text{DeptID} \rightarrow \text{Dept}$

<u>StudentId</u>	<u>S_Name</u>	DeptID
12	Ali	1
13	Reza	1
14	Behrooz	1
7	Reza	2
8	Zahra	2

<u>DeptID</u>	Dept
1	IE
2	CS





## صورت نرمال بایس-کاد

### Boyce-Codd Normal Form

رابطه “R“ BCNF است اگر برای همه وابستگی‌های تابعی آن که به صورت زیر بیان می‌شود:

$$\alpha \rightarrow \beta \quad \text{where } \alpha \subseteq R \text{ and } \beta \subseteq R$$

حداقل یکی از دو شرط زیر برقرار باشد:

$\alpha \rightarrow \beta$  یک وابستگی بدیهی باشد ( $\beta \subseteq \alpha$ ) •

$\alpha$  یک ابر کلید (super key) برای رابطه R باشد •

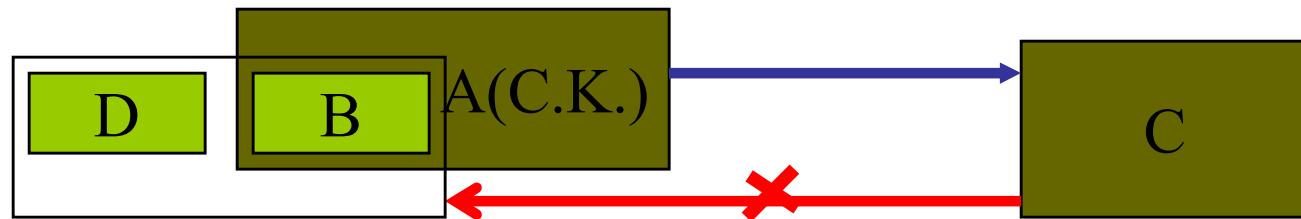




## صورت نرمال بایس-کاد

### Boyce-Codd Normal Form

تعريف اول: رابطه‌ای BCNF است اولاً ۳NF باشد و اگر در آن هر دترمینان، کلید کاندید باشد. این تعريف ساده‌شده و غیر صوری است.





## رابطه 3NF

رابطه “R“ 3NF است اگر برای همه وابستگی‌های تابعی آن که به صورت زیر بیان می‌شود:

$$\alpha \rightarrow \beta \quad \text{where } \alpha \subseteq R \text{ and } \beta \subseteq R$$

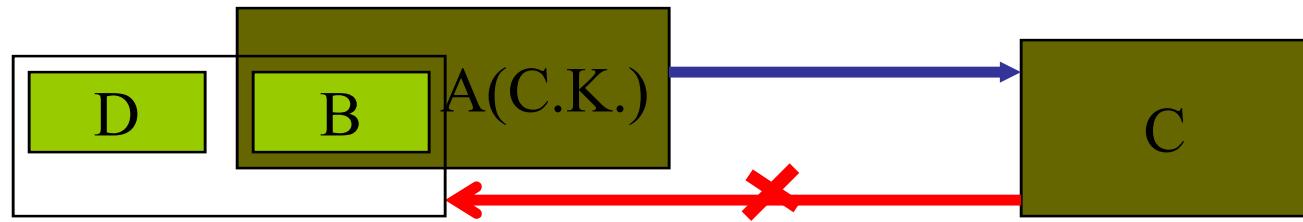
حداقل یکی از سه شرط زیر برقرار باشد:

- $(\beta \subseteq \alpha) \rightarrow \beta$  یک وابستگی بدیهی باشد
- یک ابر کلید (super key) برای رابطه R باشد
- صفت‌های  $\alpha$ - $\beta$ - قسمتی از یک کلید کاندید باشند

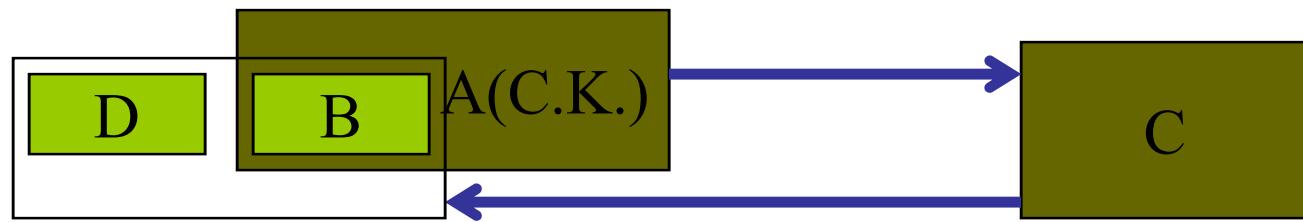




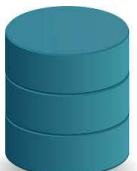
# تفاوت صورت نرمال بایس-کاد و 3NF



BCNF ■



3NF ■





## صورت نرمال بایس-کاد

تعريف چهارم: رابطه‌ای BCNF است اگر و فقط اگر سمت چپ هر FD مهم (مطرح) و کاهش‌ناپذیر، کلید کاندید رابطه باشد.





## صورت نرمال بایس-کاد

### Boyce-Codd Normal Form

اگر یک رابطه “ $R$ ” BCNF نباشد و وابستگی‌های تابعی مربوطه که  $(\beta \subseteq R)$  نبودن را سبب شده است  $\alpha \rightarrow \beta$  باشد (  $\alpha \subseteq R$  ) و می‌توان با جداسازی این رابطه به دو رابطه زیر دو رابطه BCNF داشت:

$$(\alpha \cup \beta) \bullet$$

$$(R - (\beta - \alpha)) \bullet$$

\*البته توضیح بیشتر در این مورد خارج از حوزه این درس می‌باشد)





## Boyce-Codd Normal Form

- A relation schema  $R$  is in BCNF with respect to a set  $F$  of functional dependencies if for all functional dependencies in  $F^+$  of the form

$$\alpha \rightarrow \beta$$

where  $\alpha \subseteq R$  and  $\beta \subseteq R$ , at least one of the following holds:

- $\alpha \rightarrow \beta$  is trivial (i.e.,  $\beta \subseteq \alpha$ )
- $\alpha$  is a superkey for  $R$





## Boyce-Codd Normal Form (Cont.)

- Example schema that is *not* in BCNF:

*in\_dep (ID, name, salary, dept\_name, building, budget )*

because :

- $\text{dept\_name} \rightarrow \text{building}, \text{budget}$ 
  - ▶ holds on *in\_dep*
  - ▶ but
- *dept\_name* is not a superkey

- When decompose *in\_dept* into *instructor* and *department*

- *instructor* is in BCNF
- *department* is in BCNF





# Decomposing a Schema into BCNF

- Let  $R$  be a schema  $R$  that is not in BCNF. Let  $\alpha \rightarrow \beta$  be the FD that causes a violation of BCNF.
- We decompose  $R$  into:
  - $(\alpha \cup \beta)$
  - $(R - (\beta - \alpha))$
- In our example of  $in\_dep$ ,
  - $\alpha = dept\_name$
  - $\beta = building, budget$and  $in\_dep$  is replaced by
  - $(\alpha \cup \beta) = (dept\_name, building, budget)$
  - $(R - (\beta - \alpha)) = (ID, name, dept\_name, salary)$





# Example

- $R = (A, B, C)$   
 $F = \{A \rightarrow B, B \rightarrow C\}$
- $R_1 = (A, B), R_2 = (B, C)$ 
  - Lossless-join decomposition:

$$R_1 \cap R_2 = \{B\} \text{ and } B \rightarrow BC$$

- Dependency preserving
- $R_1 = (A, B), R_2 = (A, C)$ 
  - Lossless-join decomposition:

$$R_1 \cap R_2 = \{A\} \text{ and } A \rightarrow AB$$

- Not dependency preserving  $\bowtie$   
(cannot check  $B \rightarrow C$  without computing  $R_1 \times R_2$ )





## Third Normal Form

- A relation schema  $R$  is in **third normal form (3NF)** if for all:

$$\alpha \rightarrow \beta \text{ in } F^+$$

at least one of the following holds:

- $\alpha \rightarrow \beta$  is trivial (i.e.,  $\beta \in \alpha$ )
- $\alpha$  is a superkey for  $R$
- Each attribute  $A$  in  $\beta - \alpha$  is contained in a candidate key for  $R$ .

(NOTE: each attribute may be in a different candidate key)

- If a relation is in BCNF it is in 3NF (since in BCNF one of the first two conditions above must hold).
- Third condition is a minimal relaxation of BCNF to ensure dependency preservation (will see why later).





# 3NF Example

- Consider a schema:

$\text{dept\_advisor}(s\_ID, i\_ID, \text{dept\_name})$

- With function dependencies:

$i\_ID \rightarrow \text{dept\_name}$

$s\_ID, \text{dept\_name} \rightarrow i\_ID$

- Two candidate keys =  $\{s\_ID, \text{dept\_name}\}, \{s\_ID, i\_ID\}$

- We have seen before that  $\text{dept\_advisor}$  is not in BCNF

- $R$ , however, is in 3NF

- $s\_ID, \text{dept\_name}$  is a superkey
- $i\_ID \rightarrow \text{dept\_name}$  and  $i\_ID$  is NOT a superkey, but:
  - ▶  $\{ \text{dept\_name} \} - \{ i\_ID \} = \{ \text{dept\_name} \}$  and
  - ▶  $\text{dept\_name}$  is contained in a candidate key





# Redundancy in 3NF

Consider the schema R below, which is in 3NF ■

- $R = (J, K, L)$
- $F = \{JK \rightarrow L, L \rightarrow K\}$
- And an instance table:

J	L	K
$j_1$	$l_1$	$k_1$
$j_2$	$l_1$	$k_1$
$j_3$	$l_1$	$k_1$
null	$l_2$	$k_2$

- What is wrong with the table?
  - Repetition of information
  - Need to use null values (e.g., to represent the relationship  $l_2, k_2$  where there is no corresponding value for J)





# Comparison of BCNF and 3NF

- Advantages to 3NF over BCNF. It is always possible to obtain a 3NF design without sacrificing losslessness or dependency preservation.
- Disadvantages to 3NF.
  - We may have to use null values to represent some of the possible meaningful relationships among data items.
  - There is the problem of repetition of information.





# Higher Normal Forms

- It is better to decompose *inst\_info* into:

- *inst\_child*:

<i>ID</i>	<i>child_name</i>
99999	David
99999	William

- *inst\_phone*:

<i>ID</i>	<i>phone</i>
99999	512-555-1234
99999	512-555-4321

- This suggests the need for higher normal forms, such as Fourth Normal Form (4NF), which we shall see later





# Multivalued Dependencies





# وابستگی تابعی چندمقداری

## Multi Valued Dependency

تعریف اول- در رابطه  $R(X, Y, Z)$  با صفات ساده یا مرکب  $X$ ،  $Y$  و  $Z$  می‌گوییم که  $Y$  با  $X$  وابستگی تابعی چندمقداری دارد و چنین نمایش می‌دهیم:

$$X \rightarrow\!\!\! \rightarrow Y$$

اگر به یک مقدار  $X$ ، مجموعه‌ای از مقادیر  $Y$  متناظر باشند.





## وابستگی تابعی چندمقداری

تعريف دوم - در رابطه  $R$ ، صفت  $Y$  با  $X$  وابستگی تابعی چندمقداری دارد اگر و فقط اگر مجموعه مقادیر  $Y$  متناظر با یک مقدار از جفت  $(X, Z)$  در  $R$  فقط به مقدار  $X$  بستگی داشته باشد و وابسته به مقدار  $Z$  نباشد.





## BCNF است ولی 4NF نیست

زبان برنامه نویسی	استاد	درس
Java	احمدی	برنامه نویسی پیشرفته
C++	احمدی	برنامه نویسی پیشرفته
Java	سلیمانی	برنامه نویسی پیشرفته
C++	سلیمانی	برنامه نویسی پیشرفته
Java	احمدی	مبانی برنامه نویسی
C	احمدی	مبانی برنامه نویسی
Pascal	احمدی	مبانی برنامه نویسی
Java	ادیب	مبانی برنامه نویسی
C	ادیب	مبانی برنامه نویسی
Pascal	ادیب	مبانی برنامه نویسی





# جداول 4NF معادل اسلاید قبلی

استاد	درس
احمدی	برنامه نویسی پیشرفته
سلیمانی	برنامه نویسی پیشرفته
احمدی	مبانی برنامه نویسی
ادیب	مبانی برنامه نویسی

زبان برنامه نویسی	درس
Java	برنامه نویسی پیشرفته
C++	برنامه نویسی پیشرفته
Java	مبانی برنامه نویسی
C	مبانی برنامه نویسی
Pascal	مبانی برنامه نویسی





## رباطه 4NF

تعریف - رابطه‌ای 4NF است اگر BCNF باشد و در آن  
وابستگی تابعی چند مقداری مهم وجود نداشته باشد.





# Multivalued Dependencies (MVDs)

- Suppose we record names of children, and phone numbers for instructors:
  - $inst\_child(ID, child\_name)$
  - $inst\_phone(ID, phone\_number)$
- If we were to combine these schemas to get
  - $inst\_info(ID, child\_name, phone\_number)$
  - Example data:
    - (99999, David, 512-555-1234)
    - (99999, David, 512-555-4321)
    - (99999, William, 512-555-1234)
    - (99999, William, 512-555-4321)
- This relation is in BCNF
  - Why?





# Multivalued Dependencies

- Let  $R$  be a relation schema and let  $\alpha \subseteq R$  and  $\beta \subseteq R$ . The multivalued dependency

$$\alpha \rightarrow\!\!\!\rightarrow \beta$$

holds on  $R$  if in any legal relation  $r(R)$ , for all pairs for tuples  $t_1$  and  $t_2$  in  $r$  such that  $t_1[\alpha] = t_2[\alpha]$ , there exist tuples  $t_3$  and  $t_4$  in  $r$  such that:

$$\begin{aligned}t_1[\alpha] &= t_2[\alpha] = t_3[\alpha] = t_4[\alpha] \\t_3[\beta] &= t_1[\beta] \\t_3[R - \beta] &= t_2[R - \beta] \\t_4[\beta] &= t_2[\beta] \\t_4[R - \beta] &= t_1[R - \beta]\end{aligned}$$





# MVD -- Tabular representation

## ■ Tabular representation of $\alpha \rightarrow\rightarrow \beta$

	$\alpha$	$\beta$	$R - \alpha - \beta$
$t_1$	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$a_{j+1} \dots a_n$
$t_2$	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$b_{j+1} \dots b_n$
$t_3$	$a_1 \dots a_i$	$a_{i+1} \dots a_j$	$b_{j+1} \dots b_n$
$t_4$	$a_1 \dots a_i$	$b_{i+1} \dots b_j$	$a_{j+1} \dots a_n$





## MVD (Cont.)

- Let  $R$  be a relation schema with a set of attributes that are partitioned into 3 nonempty subsets.

$Y, Z, W$

- We say that  $Y \rightarrow\rightarrow Z$  ( $Y$  multidetermines  $Z$ ) if and only if for all possible relations  $r(R)$

$\langle y_1, z_1, w_1 \rangle \in r$  and  $\langle y_1, z_2, w_2 \rangle \in r$

then

$\langle y_1, z_1, w_2 \rangle \in r$  and  $\langle y_1, z_2, w_1 \rangle \in r$

- Note that since the behavior of  $Z$  and  $W$  are identical it follows that

$Y \rightarrow\rightarrow Z$  if  $Y \rightarrow\rightarrow W$





# Example

- In our example:

$$\begin{aligned} ID &\rightarrow\!\!\!\rightarrow \text{child\_name} \\ ID &\rightarrow\!\!\!\rightarrow \text{phone\_number} \end{aligned}$$

- The above formal definition is supposed to formalize the notion that given a particular value of  $Y$  ( $ID$ ) it has associated with it a set of values of  $Z$  ( $\text{child\_name}$ ) and a set of values of  $W$  ( $\text{phone\_number}$ ), and these two sets are in some sense independent of each other.
- Note:
  - If  $Y \rightarrow Z$  then  $Y \rightarrow\!\!\!\rightarrow Z$
  - Indeed we have (in above notation)  $Z_1 = Z_2$   
The claim follows.





# Use of Multivalued Dependencies

- We use multivalued dependencies in two ways:
  1. To test relations to **determine** whether they are legal under a given set of functional and multivalued dependencies
  2. To specify **constraints** on the set of legal relations. We shall concern ourselves *only* with relations that satisfy a given set of functional and multivalued dependencies.
- If a relation  $r$  fails to satisfy a given multivalued dependency, we can construct a relations  $r'$  that does satisfy the multivalued dependency by adding tuples to  $r$ .





# Theory of MVDs

- From the definition of multivalued dependency, we can derive the following rule:
  - If  $\alpha \rightarrow \beta$ , then  $\alpha \rightarrow\rightarrow \beta$
- That is, every functional dependency is also a multivalued dependency
- The closure  $D^+$  of  $D$  is the set of all functional and multivalued dependencies logically implied by  $D$ .
  - We can compute  $D^+$  from  $D$ , using the formal definitions of functional dependencies and multivalued dependencies.
  - We can manage with such reasoning for very simple multivalued dependencies, which seem to be most common in practice
  - For complex dependencies, it is better to reason about sets of dependencies using a system of inference rules (Appendix C).





## Fourth Normal Form

- A relation schema  $R$  is in **4NF** with respect to a set  $D$  of functional and multivalued dependencies if for all multivalued dependencies in  $D^+$  of the form  $\alpha \twoheadrightarrow \beta$ , where  $\alpha \subseteq R$  and  $\beta \subseteq R$ , at least one of the following hold:
  - $\alpha \twoheadrightarrow \beta$  is trivial (i.e.,  $\beta \subseteq \alpha$  or  $\alpha \cup \beta = R$ )
  - $\alpha$  is a superkey for schema  $R$
- If a relation is in 4NF it is in BCNF





# Restriction of Multivalued Dependencies

- The restriction of  $D$  to  $R_i$  is the set  $D_i$  consisting of
  - All functional dependencies in  $D^+$  that include only attributes of  $R_i$
  - All multivalued dependencies of the form
$$\alpha \rightarrow\rightarrow (\beta \cap R_i)$$
where  $\alpha \subseteq R_i$  and  $\alpha \rightarrow\rightarrow \beta$  is in  $D^+$





# 4NF Decomposition Algorithm

*result* := { $R$ };

*done* := false;

*compute*  $D^+$ ;

Let  $D_i$  denote the restriction of  $D^+$  to  $R_i$

while (not *done*)

if (there is a schema  $R_i$  in *result* that is not in 4NF) then

begin

let  $\alpha \rightarrow\!\!\rightarrow \beta$  be a nontrivial multivalued dependency that holds

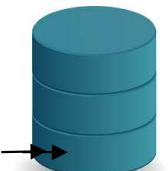
on  $R_i$  such that  $\alpha \rightarrow R_i$  is not in  $D_i$ , and  $\alpha \cap \beta = \emptyset$ ;

*result* := (*result* -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ );

end

else *done* := true;

Note: each  $R_i$  is in 4NF, and decomposition is lossless-join





# Example

- $R = (A, B, C, G, H, I)$   
 $F = \{ A \rightarrow\!\!\!\rightarrow B$   
 $\quad B \rightarrow\!\!\!\rightarrow HI$   
 $\quad CG \rightarrow\!\!\!\rightarrow H \}$
- $R$  is not in 4NF since  $A \rightarrow\!\!\!\rightarrow B$  and  $A$  is not a superkey for  $R$
- Decomposition
  - a)  $R_1 = (A, B)$   $(R_1$  is in 4NF)
  - b)  $R_2 = (A, C, G, H, I)$   $(R_2$  is not in 4NF, decompose into  $R_3$  and  $R_4$ )
  - c)  $R_3 = (C, G, H)$   $(R_3$  is in 4NF)
  - d)  $R_4 = (A, C, G, I)$   $(R_4$  is not in 4NF, decompose into  $R_5$  and  $R_6$ )
    - $A \rightarrow\!\!\!\rightarrow B$  and  $B \rightarrow\!\!\!\rightarrow HI \Rightarrow A \rightarrow\!\!\!\rightarrow HI$ , (MVD transitivity), and
    - and hence  $A \rightarrow\!\!\!\rightarrow I$  (*MVD restriction to  $R_4$* )
  - e)  $R_5 = (A, I)$   $(R_5$  is in 4NF)
  - f)  $R_6 = (A, C, G)$   $(R_6$  is in 4NF)





## مزایای نرمالترسازی

- ارائه یک طراحی بهتر و واضح‌تر با کمترین اختلاط اطلاعات
- کاهش بعضی انواع افزونگی
- کاهش بعضی آنومالیها
- تسهیل اعمال بعضی قواعد جامعیت





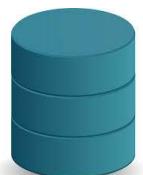
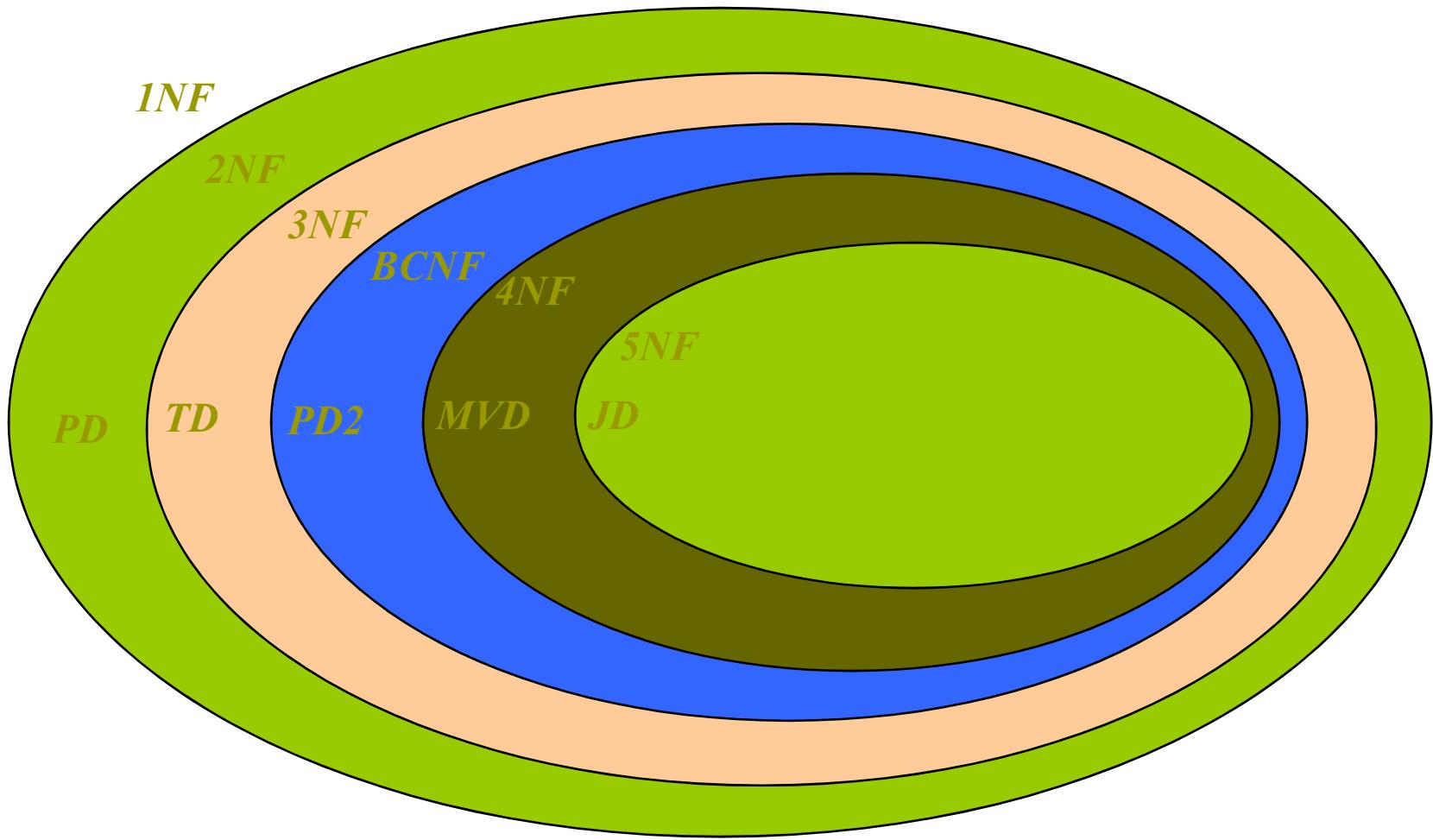
## معایب روش نرمالترسازی

- 1- بروز فزونکاری در سیستم در عمل بازیابی
- 2- ایجاد نوعی افزونگی از نوع افزونگی در سطح ادراکی
- 3- زمانگیر بودن فرآیند نرمالترسازی به ویژه اگر محیط عملیاتی بزرگ و تعداد رابطه‌ها زیاد باشد.
- 4- تصمیم‌گیری دشوار در اثر تعدد تجزیه‌ها
- 5- امکان وابستگی بین مجموعه صفات یک خردجهان، وابستگیهای به جز وابستگی تابعی و پیوندی باشد و بنابراین سبب مطرح شدن قواعد جامعیت دیگر و نیز ضوابط دیگری برای تجزیه رابطه‌ها شود و . . .





# مرور صورت های نرمال





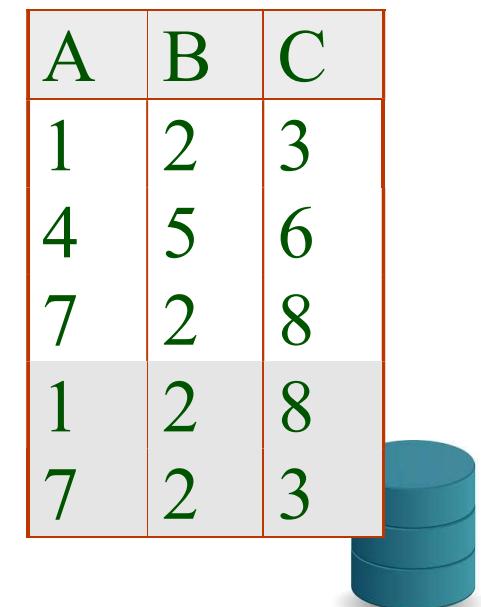
# مشکلات تجزیه

A	B	C
1	2	3
4	5	6
7	2	8



A	B
1	2
4	5
7	2

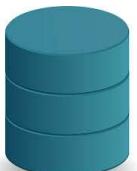
B	C
2	3
5	6
2	8





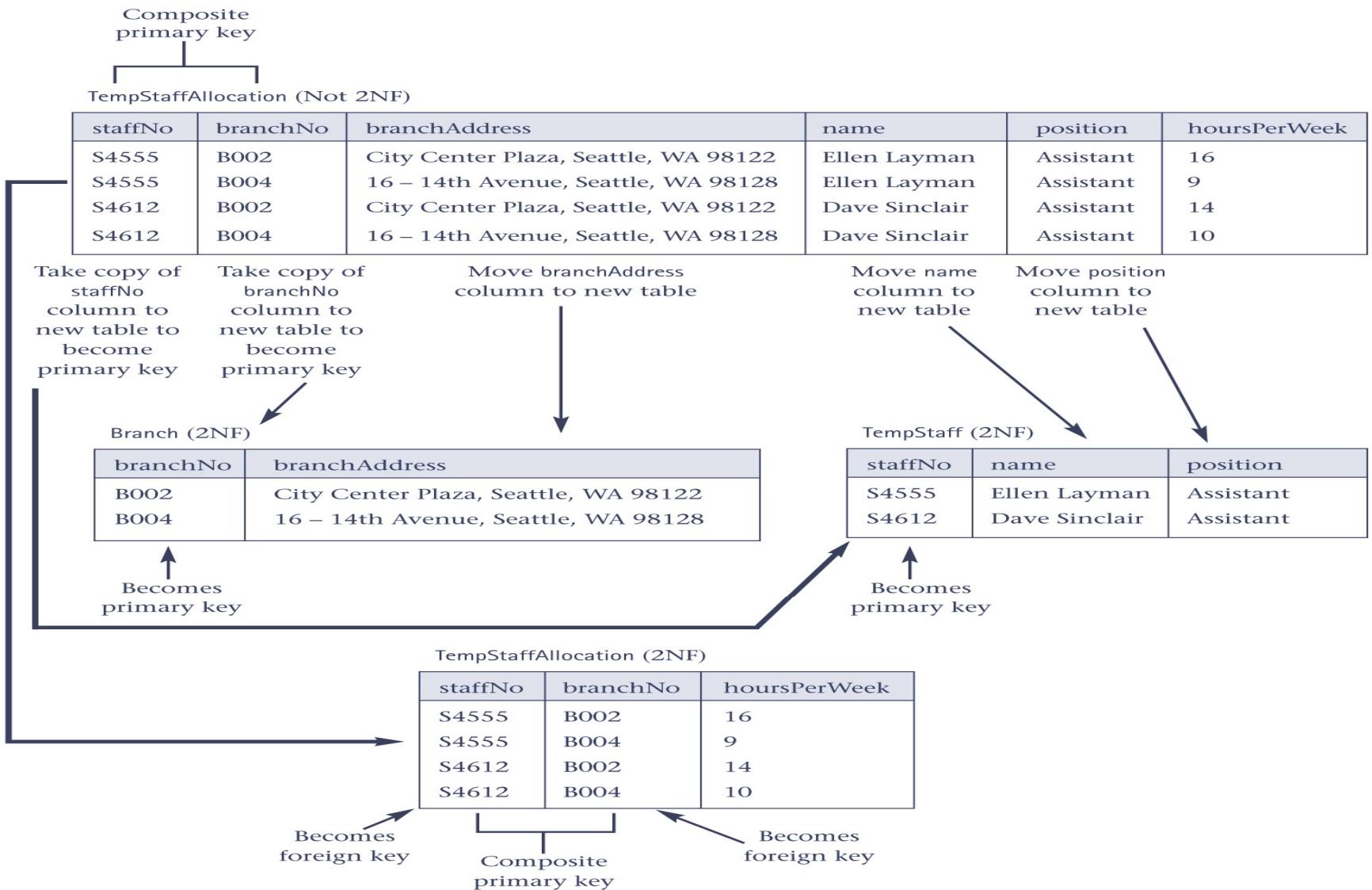
## Employee → Skill | Language

Employee	Skill	Language
ali	Cook	Turkish
Ali	Cook	French
Ali	Cook	English
ali	Driver	Turkish
ali	Driver	French
ali	Driver	English





# Converting TempStaffAllocation table to 2NF

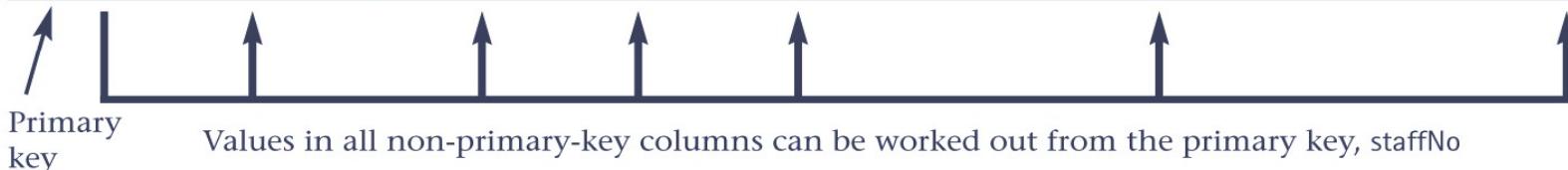




# StaffBranch table is not in 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000



Values in branchAddress and telNo columns can be worked out from branchNo, so table *not* in 3NF



Values in branchNo and telNo columns can be worked out from branchAddress, so table *not* in 3NF



Values in branchNo and branchAddress columns can be worked out from telNo, so table *not* in 3NF





# Converting the StaffBranch table to 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	BO01	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	BO01	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	BO02	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	BO02	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	BO04	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
SO415	Art Peters	Manager	41000	BO03	14 – 8th Avenue, New York, NY 10012	212-371-3000

Take copy of  
branchNo column to  
new table to  
become primary key

Move branchAddress  
column to new table

Move telNo  
column to new  
table

Branch (3NF)

branchNo	branchAddress	telNo
BO01	8 Jefferson Way, Portland, OR 97201	503-555-3618
BO02	City Center Plaza, Seattle, WA 98122	206-555-6756
BO03	14 – 8th Avenue, New York, NY 10012	212-371-3000
BO04	16 – 14th Avenue, Seattle, WA 98128	206-555-3131

Becomes  
primary key

Becomes  
candidate key

Becomes  
candidate key

Staff (3NF)

staffNo	name	position	salary	branchNo
S1500	Tom Daniels	Manager	46000	BO01
S0003	Sally Adams	Assistant	30000	BO01
S0010	Mary Martinez	Manager	50000	BO02
S3250	Robert Chin	Supervisor	32000	BO02
S2250	Sally Stern	Manager	48000	BO04
SO415	Art Peters	Manager	41000	BO03

Primary key

Becomes  
foreign key





# Normalization

## BCNF to 4NF Relations

Branch\_Staff\_Client relation

<i>Branch_No</i>	<i>SName</i>	<i>CName</i>
B3	Ann Beech	Aline Stewart
B3	David Ford	Aline Stewart
B3	Ann Beech	Mike Richie
B3	David Ford	Mike Richie



Branch\_Staff relation

<i>Branch_No</i>	<i>SName</i>
B3	Ann Beech
B3	David Ford

Branch\_Client relation

<i>Branch_No</i>	<i>CName</i>
B3	Aline Stewart
B3	Mike Richie



# پایان فصل هفتم

