

# CSE460/560 DATA MODELS AND QUERY LANGUAGES

Decomposition

Cheng-En Chuang

(Slides Adopted from Jan Chomicki and Ning Deng)



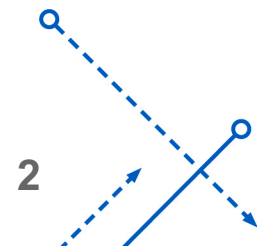
University at Buffalo

Department of Computer Science  
and Engineering

School of Engineering and Applied Sciences

# Outline

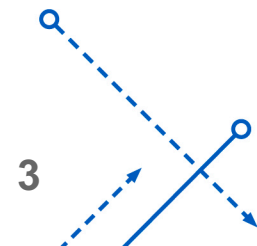
1. Decomposition
  1. Decomposition into 3NF
  2. Decomposition into BCNF



# Outline

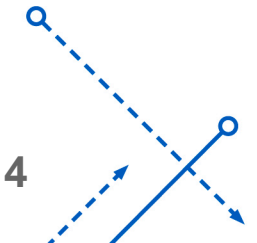
## 1. Decomposition

1. Decomposition into 3NF
2. Decomposition into BCNF



# Decomposition

- Decomposition
  - Replace a relation schema  $R$  by two relation schema  $R_1$  and  $R_2$
- Goals of Decomposition
  - Eliminate redundancy by decomposing a relation into several relations
    - In a higher normal form
  - It is important to check that a decomposition does not lead to bad design



# Decomposition

- Key: title, year

<i>title</i>	<i>year</i>	<i>length</i>	<i>genre</i>	<i>studioName</i>	<i>starName</i>
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

<i>title</i>	<i>year</i>	<i>starName</i>
Star Wars	1977	Carrie Fisher
Star Wars	1977	Mark Hamill
Star Wars	1977	Harrison Ford
Gone With the Wind	1939	Vivien Leigh
Wayne's World	1992	Dana Carvey
Wayne's World	1992	Mike Meyers

<i>title</i>	<i>year</i>	<i>length</i>	<i>genre</i>	<i>studioName</i>
Star Wars	1977	124	sciFi	Fox
Gone With the Wind	1939	231	drama	MGM
Wayne's World	1992	95	comedy	Paramount

# Problem with Decomposition

- FDs
  - Employee  $\rightarrow$  Branch
  - Project  $\rightarrow$  Branch

Employee	Project	Branch
Brown	Mars	L.A.
Green	Jupiter	San Jose
Green	Venus	San Jose
Hoskins	Saturn	San Jose
Hoskins	Venus	San Jose

# Problem with Decomposition

- A decomposition  $(R_1, R_2)$

Employee	Branch
Brown	L.A.
Green	San Jose
Hoskins	San Jose

Project	Branch
Mars	L.A.
Jupiter	San Jose
Saturn	San Jose
Venus	San Jose

# Problem with Decomposition

- A decomposition  $(R_1, R_2)$  with bogus tuples

**R**

Employee	Project	Branch
Brown	Mars	L.A.
Green	Jupiter	San Jose
Green	Venus	San Jose
Hoskins	Saturn	San Jose
Hoskins	Venus	San Jose

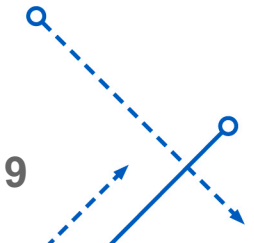
**$R_1$  Join  $R_2$**

Employee	Project	Branch
Brown	Mars	L.A.
Green	Jupiter	San Jose
Green	Venus	San Jose
Hoskins	Saturn	San Jose
Hoskins	Venus	San Jose
Green	Saturn	San Jose
Hoskins	Jupiter	San Jose



# Lossless-Join Decomposition

- The information was NOT recovered precisely (bogus tuples)
  - We need a lossless-join decomposition
- $(R_1, R_2)$  is a lossless-join decomposition of  $R$  with respect to a set of FDs  $F$  if for every instance  $r \in R$  that satisfies  $F$ :
  - $\pi_{R_1}(R) \bowtie \pi_{R_2}(R) = R$
- Use Chase test to see if a decomposition is lossless-join



# Lossless-Join Decomposition (Chase Test)

- $R(A, B, C, D)$
- FDs
  - $A \rightarrow B$
  - $A \rightarrow C$
  - $CD \rightarrow A$
- Decomposition
  - $S_1(A, D)$
  - $S_2(A, C)$
  - $S_3(B, C, D)$
- If there is a  $(a,b,c,d)$ 
  - It's a lossless-join decomposition

A	B	C	D
a	b1	c1	d
a	b2	c	d2
a3	b	c	d

$A \rightarrow B$

A	B	C	D
a	b1	c1	d
a	b1	c	d2
a3	b	c	d

$A \rightarrow C$

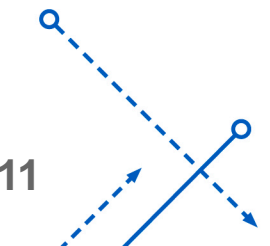
A	B	C	D
a	b1	c	d
a	b1	c	d2
a3	b	c	d

$CD \rightarrow A$

A	B	C	D
a	b1	c	d
a	b1	c	d2
a	b	c	d

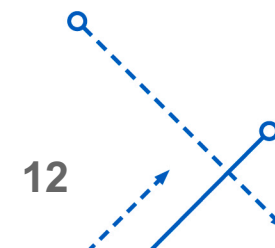
# Dependency Preservation

- Dependencies associated with relation schema  $R_1$  and  $R_2$  in a decomposition  $(R_1, R_2)$ 
  - $F_{R_1} = \{ X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq R_1 \}$
  - $F_{R_2} = \{ X \rightarrow Y \mid X \rightarrow Y \in F^+ \wedge XY \subseteq R_2 \}$
- $(R_1, R_2)$  preserves a dependency  $f$  iff  $f \in (F_{R_1} \cup F_{R_2})^+$



# Example: Dependency Preservation

- Address(street, city, zipcode)
- FD
  - street, city  $\rightarrow$  zipcode
  - zipcode  $\rightarrow$  city
- Candidate keys
  - {street, city}
  - {street, zipcode}
- Decomposition
  - SC(street, zipcode), CZ(city, zipcode)
  - Does NOT preserve street, city  $\rightarrow$  zipcode



# Outline

1. Decomposition
  1. **Decomposition into 3NF**
  2. Decomposition into BCNF



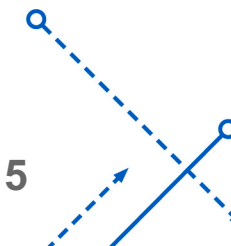
# Decomposition into 3NF

- Basis  $F'$  of  $F$ 
  - $(F')^+ = F^+$
- Minimal basis  $F'$  for  $F$ 
  1. All the FDs in  $F'$  have singleton right sides  
e.g.  $X \rightarrow A$  where  $A$  is a single attribute
  2. If any FD is removed from  $F'$ , the result is no longer a basis
    1. No trivial FD in  $F'$
  3. If for any FD in  $F'$  we remove one or more attributes from the left side of  $F$ , the result is no longer a basis



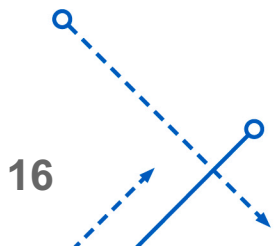
# Decomposition into 3NF

- Example of Minimal basis  $F'$ 
  - $R(A, B, C)$
  - FDs  $F$ 
    1.  $A \rightarrow B$
    2.  $A \rightarrow C$
    3.  $B \rightarrow A$
    4.  $B \rightarrow C$
    5.  $C \rightarrow A$
    6.  $C \rightarrow B$
    7.  $AB \rightarrow C$
    8.  $AC \rightarrow B$
    9.  $BC \rightarrow A$
    10.  $A \rightarrow BC$
    11.  $A \rightarrow A$
  - 1.  $F' = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$ 
    1. FD1 to FD6 (Transitivity)
    2. FD7
      1.  $A \rightarrow B \Rightarrow AB \rightarrow BB \Rightarrow AB \rightarrow B$   
(Augmentation)
      2.  $AB \rightarrow B, B \rightarrow C \Rightarrow AB \rightarrow C$  (Transitivity)
    3. FD8, FD9 same as FD7
    4. FD10 by combining FD1 and FD2
    5. FD11 (Trivial)
  - 2. There are some other minimal basis



# Decomposition into 3NF

1. 3NF Synthesis Algorithm
2. Input: A relation  $R$  and a set FD  $F$  that hold for  $R$
3. Output: A decomposition of  $R$  into a collection of relations
  1. Property of the collection: 3NF, Lossless-join, Dependency Preservation
4. Method:
  1. Create a minimal basis  $F'$
  2. For each FD  $X \rightarrow A \in F'$ 
    1. Create a relation with attributes  $XA$
  3. If none of relation from step 2 is a superkey of  $R$ 
    1. Add another relation whose schema is a key for  $R$



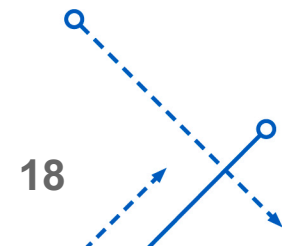


# Decomposition into 3NF

- Example of 3NF Synthesis Algorithm
  - $R(A, B, C, D, E)$
  - FDs  $F$ 
    1.  $AB \rightarrow C$
    2.  $C \rightarrow B$
    3.  $A \rightarrow D$
  - Candidate Keys
    - $\{A, B, E\}$
    - $\{A, C, E\}$
- 1. Create a minimal basis  $F'$ 
    1.  $F$  is already a minimal basis
  2. For each FD  $X \rightarrow A \in F'$ 
    1. Create a relation with attributes  $XA$ 
      1.  $S_1(A, B, C); S_2(B, C); S_3(A, D)$
      2. No need for  $S_2$  due to  $S_2 \subset S_1$
  3. If none of relation from step 2 is a superkey of  $R$ 
    1. Add another relation whose schema is a key for  $R$ 
      1.  $S_4(A, B, E)$
  4. Output:  $S_1(A, B, C); S_3(A, D); S_4(A, B, E)$

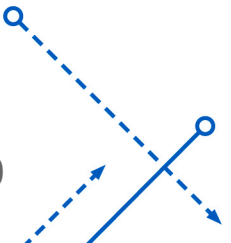
# Outline

1. Decomposition
  1. Decomposition into 3NF
  - 2. Decomposition into BCNF**



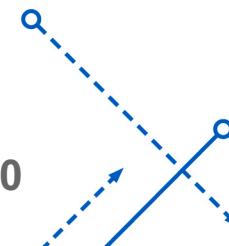
# Algorithm: Decomposition of schema $R$ into BCNF

- For some nontrivial non-key dependency  $X \rightarrow A \in F^+$ 
  - Create a relation  $R_1$  with the set of attributes  $XA$  and FDs  $F_{R_1}$
  - Create a relation  $R_2$  with the set of attributes  $R - \{A\}$  and FDs  $F_{R_2}$
- Decompose further  $R_1, R_2$  if they are not in BCNF
- This algorithm produces a lossless-join decomposition into BCNF



# Example-Decomposition into BCNF

- Movies(title, year, studioName, president, presAddr)
  - FDs
    1. title, year  $\rightarrow$  studioName
    2. studioName  $\rightarrow$  president
    3. president  $\rightarrow$  presAddr
  - Given FD8
    - Movie1(studioName, president, presAddr)
      - Given FD3
        - Movie3(studioName, president);
        - Movie4(president, presAddr)
    - Movie2(title, year, studioName)
  - Output:
    - Movie2(title, year, studioName);
    - Movie3(studioName, president);
    - Movie4(president, presAddr)
- $F^+$  contains FD1, FD2, FD3 and following new FDs
  4. title, year  $\rightarrow$  president
  5. title, year  $\rightarrow$  presAddr
  6. title, year  $\rightarrow$  studioName, president, presAddr
  7. studioName  $\rightarrow$  presAddr
  8. studioName  $\rightarrow$  president, presAddr



# Recommended Reading

Database Systems: The Complete Book  
Chapter 3.1 – 3.6