

In the Lecture Series Introduction to Database Systems

# Normalization

# Learning Objectives

- Understand the rationale (anomalies) and definition of the main normal forms based on functional dependencies (BCNF, 3NF)
- Be able to decompose and synthesize a schema into a lossless and dependency preserving BCNF and 3NF.

# Anomalies

- Redundant storage
- Update anomalies
- Insertion anomalies
- Deletion anomalies

# Anomalies: Example

Assume that the position determines the salary:  
company  
 $\text{position} \rightarrow \text{salary}$

eNumber	firstName	lastName	address	department	position	salary
1XU3	Dewi	Srijaya	12a Jln Lempeng	Toys	Clerk	2000
4W3E	Izabel	Leong	10 Outram Park	Sports	Trainee	1200
3XXE	John	Smith	107 Clementi Rd	Toys	Clerk	2000
5SD2	Axel	Bayer	55 Cuscaden Rd	Sports	Trainee	1200
6RG5	Winnie	Lee	10 West Coast Rd	Sports	Manager	2500
755Y	Sylvia	Tok	22 East Coast Ln	Toys	Manager	2600
2SD3	Eric	Wei	100 Jurong drive	Toys	Assistant manager	2200
?	?	?	?	?	Security guard	1500

Redundant storage

Update anomaly

key

key

Insertion anomaly

Potential deletion anomaly

# Decomposition

- A decomposition of a relation scheme  $R$  is a set of relation scheme  $R_i$  such that:

$$\bigcup_i R_i = R$$

Namely, we have all the attributes.

The tables  $R_i$  are called 'fragments'

# Decomposition: Example

employee

eNumber	firstName	lastName	address	depart- ment	position
1XU3	Dewi	Srijaya	12a Jln Lempeng	Toys	Clerk
4W3E	Izabel	Leong	10 Outram Park	Sports	Trainee
3XXE	John	Smith	107 Clementi Rd	Toys	Clerk
5SD2	Axel	Bayer	55 Cuscaden Rd	Sports	Trainee
6RG5	Winnie	Lee	10 West Coast Rd	Sports	Manager
755Y	Sylvia	Tok	22 East Coast Ln	Toys	Manager
2SD3	Eric	Wei	100 Jurong drive	Toys	Assistant manager

key

key

salary

position	salary
Clerk	2000
Trainee	1200
Manager	2500
Assistant manager	2200
Security guard	1500

key

# Lossless Decomposition: Example

- The decomposition is **lossless** if we can recover the initial table:

```
SELECT firstName, lastName, address,  
       department, e.position, salary  
FROM employee AS e, salary AS s  
WHERE e.position = s.position
```

- Some attributes must be repeated: the position appears in both fragments.

# Lossless Decomposition: Example

employee

eNumber	firstName	lastName	address	depart- ment	position
1XU3	Dewi	Srijaya	12a Jln Lempeng	Toys	Clerk
4W3E	Izabel	Leong	10 Outram Park	Sports	Trainee
3XXE	John	Smith	107 Clementi Rd	Toys	Clerk
5SD2	Axel	Bayer	55 Cuscaden Rd	Sports	Trainee
6RG5	Winnie	Lee	10 West Coast Rd	Sports	Manager
755Y	Sylvia	Tok	22 East Coast Ln	Toys	Manager
2SD3	Eric	Wei	100 Jurong drive	Toys	Assistant manager

{ } key

{ } key

salary

position	salary
Clerk	2000
Trainee	1200
Manager	2500
Assistant manager	2200
Security guard	1500

{ } key

# Lossless Decomposition: Counter Example

Flight Number	Departure time	Arrival time	Origin	Destination
SG12	12h00	13h00+	SIN	CDG
TG414	15h50	16h30	SIN	JKT
TG415	12h00	14h20	BKK	SIN

key

# Lossless Decomposition: Counter Example

Flight Number	Departure time	Arrival time	Origin	Destination
SG12	12h00	13h00+	SIN	CDG
TG414	15h50	16h30	SIN	JKT
TG415	12h00	14h20	BKK	SIN

Flight Number	Departure time	Origin
SG12	12h00	SIN
TG414	15h50	SIN
TG415	12h00	BKK

?

Departure time	Arrival time	Destination
12h00	13h00+	CDG
15h50	16h30	JKT
12h00	14h20	SIN

# Lossless, Dependency Preserving

- The decomposition is lossy
- And we lost functional dependencies:

$\{\text{Flight Number}\} \rightarrow \{\text{Arrival time, Destination}\}$

# Decomposition: Example

- Consider the relation scheme  
 $\{C, S, J, D, P, Q, V\}$

with FDs

$$\{C\} \rightarrow \{S, J, D, P, Q, V\}$$

$$\{J, P\} \rightarrow \{C\}$$

$$\{S, D\} \rightarrow \{P\}$$

Contracts(Contractid, Supplierid, Projectid, Deptid, Partid, Qty, Value)

# Decomposition: Example

- Consider the decomposition into  $\{C, S, J, D, Q, V\}$ ,  $\{S, D, P\}$
- The decomposition is lossless
  - We can recover the initial relation thanks to S,D
- However the functional dependency  $\{J, P\} \rightarrow \{C\}$  is **lost** across the two relations...
- This decomposition is NOT **dependency preserving**

# Dependency Preserving Decomposition

- The decomposition of a relation scheme
  - $R$  with FDs  $F$ 
    - Is a set of relation schemes  $R_i$  with FDs  $F_i$
  - The decomposition is dependency preserving if and only if
$$(\cup_i F_i)^+ = F^+$$

# Too Much Decomposition

- It might be tempting to decompose to the extreme
- Evaluation of queries may be **inefficient** since it will involve combining several relations

# Too Much Decomposition: Example

Flight Number	Departure time	Origin
SG12	12h00	SIN
TG414	15h50	SIN
TG415	12h00	BKK

Flight Number	Arrival time	Destination
SG12	13h00+	CDG
TG414	16h30	JKT
TG415	14h20	SIN

# Looking for a “Good” Design

- The designer needs guidelines:
  - Normalization theory
    - Minimal redundancy and no anomalies
    - Lossless decompositions
    - Dependency preserving decompositions
- But ultimately the designer needs to look at the **workload** (the queries and their efficiency requirement)

# Boyce-Codd Normal Form (BCNF)

- R is a relation schema, with the set F of FDs
- R is in BCNF if and only if
  - For all X:  $X \subset R$
  - And, for all A  $\in R$
  - such that there exists a FD:  $X \rightarrow \{A\}$  in  $F^+$
- Then
  - $A \in X$  ( $X \rightarrow \{A\}$  is **trivial**), or
  - X is a **superkey** for R

## Second Normal Form (2NF)

- R is a relation schema, with the set F of FDs
- R is in 2NF if and only if
  - For all  $X: X \subset R$
  - and, for all  $A \in R$
  - such that there exists a FD:  $X \rightarrow \{A\}$  in  $F^+$
- Then
  - $A \in X$  ( $X \rightarrow \{A\}$  is **trivial**), or
  - X is NOT a proper subset of a **candidate key** for R, or
  - A is **part of some candidate key** for R

# Third Normal Form (3NF)

- R is a relation schema, with the set F of FDs
- R is in 3NF if and only if
  - For all  $X: X \subset R$
  - And, for all  $A \in R$
  - such that there exists a FD:  $X \rightarrow \{A\}$  in  $F^+$
- Then
  - $A \in X$  ( $X \rightarrow \{A\}$  is **trivial**), or
  - X is a **superkey** for R, or
  - A is **part of some candidate key** for R  
*(A is called a prime attribute)*

- BCNF:
  - Trivial, or
  - $X$  is a superkey for  $R$
- 3NF:
  - Trivial, or
  - $X$  is a superkey for  $R$ , or
  - $A$  is part of some candidate key for  $R$
- 2NF:
  - Trivial, or
  - $X$  is not a proper subset of a candidate key for  $R$ , or
  - $A$  is part of some candidate key for  $R$

# Decomposition into BCNF

Let  $S$  be the initial set of schemes with FDs  $F$

**Until** all relation schemes in  $S$  are in BCNF

**for** each  $R$  in  $S$

**if** FD  $X \rightarrow Y$  in  $F^+$  violates BCNF for  $R$

*( $X \rightarrow R$  not hold, not superkey)*

*$X \cap Y = \emptyset$ , not trivial)*

**then**

use  $X \rightarrow X^+$

let  $S$  be  $(S - \{R\}) \cup \{(R - X^+) \cup X, X^+\}$

**endfor**

**endumtil**

# Decomposition into BCNF

$X \rightarrow Y$

$X = \{C, D\}$

$X+ = \{C, D, E, F\}$

A	B	C	D	E	F

A	B	C	D

C	D	E	F

# Decomposition into BCNF

- The different possible orders\* in which we may consider the dependencies violating BCNF in the algorithm application may lead to different decompositions

*\*orders in which we consider the constraints violating the BCNF condition*

# Decomposition into BCNF

- Let us consider the relation scheme  $R=\{A,B,C,D,E\}$  and the FDs:
  - $\{A\} \rightarrow \{B\}$
  - $\{A\} \rightarrow \{E\}$
  - $\{C\} \rightarrow \{D\}$
- Candidate key:  $\{A, C\}$

# Decomposition into BCNF

- R is not in BCNF
- Because (for instance):
  - $\{A\} \rightarrow \{B\}$  holds
  - It is NOT trivial
  - $\{A\}$  is NOT a superkey

# Decomposition into BCNF

- Pick  $\{A\} \rightarrow \{B\}$  for decomposition
- Expand into  $\{A\} \rightarrow \{A,B,E\}$
- $\{A,B,C,D,E\}$  becomes
  - $\{A,C,D\}$  and  $\{A,B,E\}$
- With FDs: (we need projected FDs)
  - $\{A\} \rightarrow \{B\}$ , on  $\{A,B,E\}$
  - $\{A\} \rightarrow \{E\}$ , on  $\{A,B,E\}$
  - $\{C\} \rightarrow \{D\}$ , on  $\{A,C,D\}$

## Remark: Projecting FDs

- If  $S$  is a fragment after decomposition of a relation  $R$  with FDs  $F$
- The set of projected FDs on  $S$  is the set  $G$  of FDs
  - If  $X \rightarrow Y$  is in  $G$ 
    - Then  $X$  and  $Y$  are subsets of  $S$
    - $X \rightarrow Y$  is in  $F^+$
  - If  $X \rightarrow Y$  is in  $F^+$  and  $X$  and  $Y$  are subsets of  $S$ 
    - Then  $X \rightarrow Y$  is in  $G^+$

# Decomposition into BCNF

- Pick  $\{C\} \rightarrow \{D\}$  for decomposition
- Expand into  $\{C\} \rightarrow \{C,D\}$
- $\{A,C,D\}$  and  $\{A,B,E\}$  become
  - $\{A,C\}$ ,  $\{C,D\}$  and  $\{A,B,E\}$
- With FDs: (*we need projected FDs*)
  - $\{A\} \rightarrow \{B\}$ , *on*  $\{A,B,E\}$
  - $\{A\} \rightarrow \{E\}$ , *on*  $\{A,B,E\}$
  - $\{C\} \rightarrow \{D\}$ , *on*  $\{C,D\}$

# Decomposition into BCNF

- Finally the BCNF decomposition of  $R=\{A,B,C,D,E\}$  with the FDs:

$\{A\} \rightarrow \{B\}$ ,

$\{A\} \rightarrow \{E\}$ ,

$\{C\} \rightarrow \{D\}$

- is:  $R1=\{A,C\}$ ,  $R2=\{C,D\}$  and  $R3=\{A,B,E\}$

# Decomposition into BCNF

- BCNF decomposition may not be dependency preserving
- Example: {A,B,C} with FDs  
 $\{A,B\} \rightarrow \{C\}$   
 $\{C\} \rightarrow \{A\}$
- The first FD (the key!) is not preserved.

# Decomposition into 3NF (Synthesis)

Let S be the set of relation scheme R with FDs F

**for** each  $X \subset R$  such that  $X \rightarrow \{A_i\}$  is in the **minimal cover** of F

**if** no scheme contains  $X \cup_i \{A_i\}$

**then** create relation with scheme  $X \cup_i \{A_i\}$

**endfor**

**if** no scheme contains a key for R

**then** create a relation with scheme with key for R

# Decomposition into 3NF (Synthesis)

- Let us consider the relation scheme  $R=\{A,B,C,D,E\}$  and the FDs:
  - $\{A\} \rightarrow \{B\}$
  - $\{A\} \rightarrow \{E\}$
  - $\{C\} \rightarrow \{D\}$
- Candidate key:  $\{A, C\}$

# Decomposition into 3NF (Synthesis)

- Minimal cover:

$$\{A\} \rightarrow \{B\}$$

$$\{A\} \rightarrow \{E\}$$

$$\{C\} \rightarrow \{D\}$$

- Decomposition:

- $R1 = \{A, B\}$

- $R2 = \{A, E\}$

- $R3 = \{C, D\}$

- $R4 = \{A, C\}$

# Decomposition into 3NF

- (Extended) Minimal cover:

$$\{A\} \rightarrow \{B, E\}$$

$$\{C\} \rightarrow \{D\}$$

- Decomposition:

- $R1 = \{A, B, E\}$
- $R2 = \{C, D\}$
- $R3 = \{A, C\}$

# Dependency preserving

- The 3NF synthesis algorithm always finds a lossless dependency preserving decomposition

## Credits

The content of this lecture is based  
on chapter 8 of the book  
“Introduction to database  
Systems”

By  
S. Bressan and B. Catania,  
McGraw Hill publisher

Clipart and media are licensed from  
Microsoft Office Online Clipart  
and Media

Copyright © 2016 by Stéphane Bressan



# Các dạng chuẩn – Dạng chuẩn 2 (2NF)

## - Kiểm tra lđqh đạt 2NF (tt)

- Ví dụ:
  - Cho R(ABCD)
  - F = { $AB \rightarrow C$ ,     $A \rightarrow D$ ,               $BD \rightarrow C$ }
- Hỏi: Lược đồ R có ở dạng chuẩn 2 không?
- Cách làm:
  - Khóa là: K = AB
  - Tập thuộc tính khóa là {A, B}
  - Tập thuộc tính không khóa là {C, D}
  - Xét fd:  $A \rightarrow D$ 
    - A là tập con của khóa, D là thuộc tính không khóa
- rõ ràng thuộc tính không khóa d phụ thuộc không đầy đủ vào khóa → R không đạt dạng chuẩn 2 (không là 2NF)

# Các dạng chuẩn – Dạng chuẩn 2 (2NF)

## - Kiểm tra lđqh đạt 2NF (tt)

- Ví dụ:

Cho R(CSZ)       $F = \{CS \rightarrow Z, Z \rightarrow C\}$

Hỏi: Lược đồ R có ở dạng chuẩn 2 không?

- Cách làm:

- Khóa là: K1 = CS, K2 = SZ

- Tập thuộc tính khóa: CSZ

- Không có thuộc tính không khóa

- không thẻ nào có trường hợp thuộc tính không khóa phụ thuộc không đầy đủ vào khóa → lược đồ R ở dạng chuẩn 2.

- Nhận xét:

- **Nếu tất cả các thuộc tính của lược đồ quan hệ đều là thuộc tính khóa thì lược đồ là 2NF.**

# Các dạng chuẩn – Dạng chuẩn 2 (2NF)

## - Kiểm tra lđqh đạt 2NF (tt)

- Ví dụ:
  - Cho R(ABCD)
  - $F = \{A \rightarrow BC, B \rightarrow D\}$

Hỏi: Lược đồ R có ở dạng chuẩn 2 không?
- Cách làm:
  - Lược đồ quan hệ trên có một khóa duy nhất là: K = A
  - →Lược đồ R là 2NF vì không tồn tại tập con thực sự của khóa nên không tồn tại phụ thuộc hàm không đầy đủ của thuộc tính không khóa vào khóa.
- Nhận xét:
  - **Nếu tất cả các khóa của lược đồ quan hệ chỉ có một thuộc tính thì lược đồ quan hệ là 2NF**

## Các dạng chuẩn – Dạng chuẩn 3 (3NF)

### - Kiểm tra lđqh đạt 3NF (tt)

- **Ví dụ:**

- $R = (C, S, Z)$ ,  $F = \{CS \rightarrow Z, Z \rightarrow C\}$
- Khóa dự tuyển: CS và SZ
- Tất cả các thuộc tính đều là thuộc tính khóa  
 $\rightarrow R$  là 3NF

- **Ví dụ:**

- $R = (A, S, I, P)$ ,  $F = \{SI \rightarrow P, S \rightarrow A\}$ ,
- Khóa: SI
- $S \rightarrow A$  nhưng S là tập con của khóa và A là thuộc tính không khóa  $\rightarrow R$  không là 2NF  $\rightarrow R$  không là 3NF