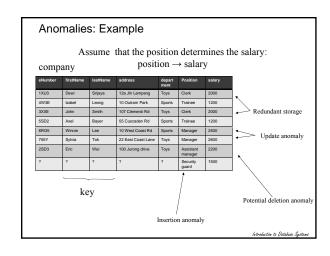
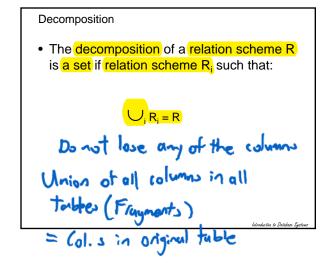
Normalization

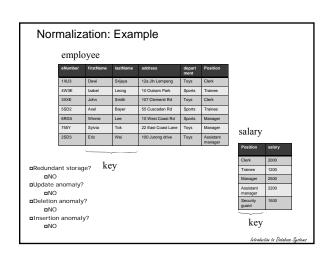
Presented by Stéphane Bressan

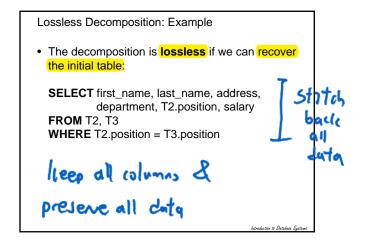
# Understand the rationale (anomalies) and definition of the main normal forms based on functional dependencies (2NF, 3NF and BCNF) Be able to decompose (or synthesize) a schema into a dependency preserving BCNF or 3NF.

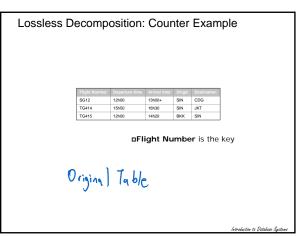
# Anomalies Redundant storage Update anomalies Insertion anomalies Deletion anomalies

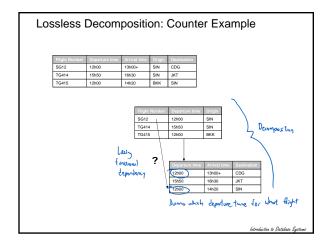


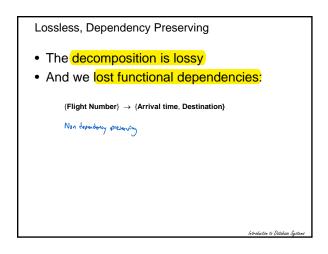










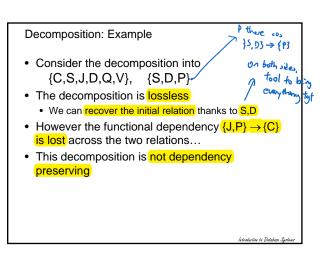


Decomposition: Example

• Consider the relation scheme
{C,S,J,D,P,Q,V}

• with FDs
{C} → {S,J,D,P,Q,V}
{J,P} → {C}
{S,D} → {P}

(ordinate) (continuate) , Supplier i.d., project i.d., dept i.d., partial , gfy we have )



# Dependency Preserving Decomposition

- The decomposition of a relation scheme
  - R with FDs F
  - Is a set of relation schemes R; with FDs F;
- 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 y join cause inefficiency

### 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

can actually decompose anto 4 tables

# 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)

# 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 ⊂ R
  - and, for all A ∈ R
  - such that there exists a FD:  $X \rightarrow \{A\}$  in F+
- Then
  - A ∈ X (the FD 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 (A is called a prime attribute)





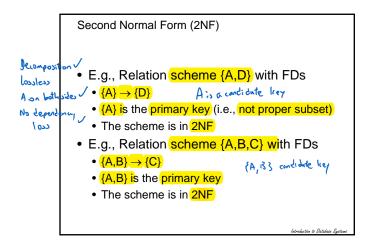


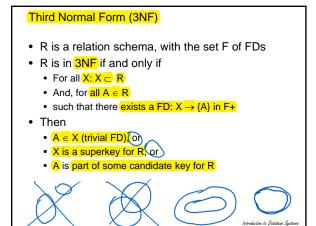
Not a proper Jubset

candidate key: minimal superker

# Second Normal Form (2NF)

- Consider the relation scheme {A,B,C,D} with the FDs:
  - $\{A,B\} \rightarrow \{C,D\}$  and  $\{A,B\} \rightarrow \{A,B,c,0\}$
- $\{A\} \rightarrow \{D\}$ • {A,B} is a primary key (it is not a proper
- subset) • {A} is a proper subset of a primary key ∨ 6 late
- {D} is not part of some candidate key (1) to not on prime attribute
- This scheme is not in 2NF





3NF is always in 2NF 3NF strater cos of 2nd cond.

Third Normal Form (3NF)

- Rationale:
  - If X is not a superkey for R then this dependency is partial and may cause redundancy
  - If A is not part of some candidate key for R then there is a transitive dependency: candidate key → X → A
  - Insertion/deletion/update anomalies

BCNF = 3NF = 2NF = 1NF

Introduction to Database Systems

# 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 ⊆ R • And, for all A ∈ R • such that there exists a FD: X → {A} in F+ • Then • A ∈ X, or • X is a superkey for R No lost condition Strider removing a possibility BCNF alwaps in 3NF

```
2NF:

A ∈ X, or
X is not a proper subset of a candidate key for R, or
A is part of some candidate key for R

3NF:

A ∈ X, or
X is a superkey for R, or
A is part of some candidate key for R

BCNF:

A ∈ X, or
X is a superkey for R

BCNF:

A ∈ X, or
X is a superkey for R

A ∈ X, or
X is a superkey for R

All W mut yee All column in Ft !!
```

```
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 → Y in F+ violates BCNF for R

(X → R not hold, not superkey)

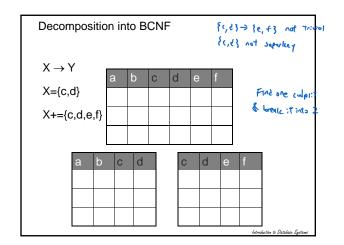
X ∩ Y = Ø, not trivial)

then

use X → X+

let S be (S - {R}) ∪ {(R-X+) ∪ X, X+}

endfor
enduntil
```



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

There may be problems which may lose functional dependencies May get sth

non-teperdeny preserving

Introduction to Database Sustems

### 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+

troduction to Database Systems

# Decomposition into BCNF

Let us consider the relation scheme R={A,B,C,D,E} and the FDs:

$$\{A\} \rightarrow \{B\}$$
And trivial, non-superliesy
 $\{A\} \rightarrow \{E\}$ 
 $\{C\} \rightarrow \{D\}$ 

{A,c} → {A,B,C,D,E}

Candidate key: {A, C}

{A, C}

BCOF

Minimal Cover

L>{1} ≥ 3 B, E3

\* 3NF algo: Call minimal cover. Ly This case, already a minimal cover.

 $\{(3 \Rightarrow j D)\}$ Decomp:  $k|: \{A, B, E\}$ 

Decomposition: RI = (A,B)

RZ = (A,E)

R3 = (C,D)

Rs: (C, D)
Rs: (A, C)

R4 = (A,C)

It BUF synthesis, usually can find BUNF (NJT ALWAX)

# Decomposition into BCNF

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

Introduction to Database Systems

# Decomposition into BCNF

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

{A} → {B}, on {A,B,E} ✓ \ (A3 is supplied

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

Introduction to Database System

# Decomposition into BCNF

- Pick  $\{C\} \rightarrow \{D\}$  for decomposition
- Expand into  $\{C\} \rightarrow \{D\}$
- {A,C,D} and {A,B,E} becomes
  - {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: {A,C}, {C,D} and {A,B,E}

La Depundency preserving

# Decomposition into BCNF

- BCNF decomposition may not be dependency preserving
- Example: {A,B,C} with FDs

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

Decompose to CA, BC.  $\{C\} \rightarrow \{A\}$ 

The first FD is not preserved.

Darle & Conquer Alyo

## Decomposition into 3NF

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

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

**if** no scheme contains  $X \cup_{i} \{A_{i}\}$ 

then create relation with scheme X U; {A;}

endfor

if no scheme contains a key for R

then create a relation with scheme with key for R

# Dependency preserving

• The 3NF synthesis algorithm always finds a loosless dependency preserving decomposition

Copyright © 2013 by Stéphane Bressa

