AALBORG UNIVERSITY

Database System (SW5)

7. Relational Database Design

Tiantian Liu
Department of Computer Science
Aalborg University
Fall 2025

Motivation

AALBORG UNIVERSITY

- A good design prevents problems that occur later during usage
- Changing a database schema before it is in use is much cheaper
- Redundancy is the source of many problems and should be avoided by meeting certain design criteria

Learning Goals

AALBORG UNIVERSITY

- Understand concepts of functional dependencies and normal forms
- Describe the quality of a design by using normal forms
- Improve a database design by decomposition of relations

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Features of Good Relational Designs

AALBORG UNIVERSITY

- *instructor*(*i_id, name, salary, dept_name*)
- *department*(*dept_name, building, budget*)
- Suppose we combine *instructor* and *department* into *in_dep,* which represents the natural join on the relations *instructor* and *department*

i_id	*name*	*salary*	*dept_name*	*building*	*budget*
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
12121	Wu	90000	Finance	Painter	120000
45565	Katz	75000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
76543	Singh	80000	Finance	Painter	120000

Any problem?

Data Anomalies

AALBORG UNIVERSITY

- There are several anomalies in the table
- First, redundancy:
  - *building* and *budget* for *Finance* are duplicated
- Second, update anomalies:
  - We may accidentally update one of *Finance* department's building, leaving the other unchanged

ID	*name*	*salary*	*dept_name*	*building*	*budget*
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
12121	Wu	90000	Finance	Painter	120000
45565	Katz	75000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
76543	Singh	80000	Finance	Painter	120000

Decomposition

AALBORG UNIVERSITY

- 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

ID	*name*	*salary*	*dept_name*
10101	Srinivasan	65000	Comp. Sci.
12121	Wu	90000	Finance
45565	Katz	75000	Comp. Sci.
58583	Califieri	62000	History
76543	Singh	80000	Finance

instructor

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
History	Painter	50000
Finance	Painter	120000

department

Decomposition

AALBORG UNIVERSITY

- Not all decompositions are good.
- Suppose we decompose instructor as follows

ID	*name*	*salary*	*dept_name*
10101	Srinivasan	65000	Comp. Sci.
12121	Wu	90000	Finance
45565	Katz	75000	Comp. Sci.
58583	Califieri	62000	History
67131	Califieri	70000	Physics
76543	Singh	80000	Finance

instructor

ID	*name*
10101	Srinivasan
12121	Wu
45565	Katz
58583	Califieri
67131	Califieri
76543	Singh

name	*salary*	*dept_name*
Srinivasan	65000	Comp. Sci.
Wu	90000	Finance
Katz	75000	Comp. Sci.
Califieri	62000	History
Califieri	70000	Physics
Singh	80000	Finance

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

Normalization Theory

AALBORG UNIVERSITY

- Decide whether a particular relation *R* is in "good" form.
- In the case that a relation *R* is not in "good" form, decompose it into set of relations $\{R_1, R_2, ..., R_n\}$ such that
  - Each relation is in good form
  - The decomposition is a lossless decomposition
- Our theory is based on:
  - Functional dependencies

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Functional Dependencies

AALBORG UNIVERSITY

- An instance of a relation that satisfies all such real-world constraints is called a legal instance of the relation
- A legal instance of a database is one where all the relation instances are legal instances
- Constraints on the set of legal relations
  - Require that the value for a certain set of attributes determines uniquely the value for another set of attributes

Functional Dependencies Definition

AALBORG UNIVERSITY

- Let *R* be a relation schema

$$\alpha \subseteq R \ \ and \ \ \beta \subseteq R$$

- The **functional dependency**

$$\alpha \rightarrow \beta$$

holds on *R* if and only if for any legal relations *r*(*R*), whenever any two tuples t_1 and t_2 of *r* agree on the attributes α, they also agree on the attributes β. That is,

$$t_1[\alpha] = t_2[\alpha] \ \Rightarrow \ t_1[\beta] = t_2[\beta]$$

- The α values uniquely identify the β values, α functionally determines β
- If $\beta \subseteq \alpha$, $\alpha \rightarrow \beta$ is called **trivial**

Functional Dependencies--Example

AALBORG UNIVERSITY

- Example: Consider $r(A, B, C)$ with the following instance of r.

A	B	C
a	4	5
a	4	6
b	1	2
c	8	3
d	1	5

- On this instance
- $A \rightarrow B$ hold;
- $B \rightarrow A$ does **NOT** hold

Functional Dependencies--Example

AALBORG UNIVERSITY

ID	*name*	*salary*	*dept_name*
10101	Srinivasan	65000	Comp. Sci.
12121	Wu	90000	Finance
45565	Katz	75000	Comp. Sci.
58583	Califieri	65000	History
67131	Califieri	70000	Physics
76543	Singh	80000	Finance

Which FDs are fulfilled?

- $\{ID\} \rightarrow \{name\}$
- $\{name\} \rightarrow \{salary\}$
- $\{ID, name\} \rightarrow \{salary, dept\_name\}$
- $\{name\} \rightarrow \{ID\}$
- $\{ID, salary\} \rightarrow \{name\}$

Functional Dependencies--Example

AALBORG UNIVERSITY

ID	*name*	*salary*	*dept_name*
10101	Srinivasan	65000	Comp. Sci.
12121	Wu	90000	Finance
45565	Katz	75000	Comp. Sci.
58583	Califieri	65000	History
67131	Califieri	70000	Physics
76543	Singh	80000	Finance

Which FDs are trivial?

- $\{ID, name\} \rightarrow \{name\}$
- $\{ID, name, salary\} \rightarrow \{salary\}$
- $\{ID, name\} \rightarrow \{salary, dept\_name\}$
- $\{ID, salary\} \rightarrow \{name, salary\}$
- $\{ID, name, salary\} \rightarrow \{salary, dept\_name\}$

If $\beta \subseteq \alpha$, $\alpha \rightarrow \beta$ is called **trivial**

Functional Dependencies--Exercise

AALBORG UNIVERSITY

A	*B*	*C*	*D*
a1	b1	c1	d1
a2	b2	c2	d2
a2	b2	c2	d3
a3	b1	c3	d3

Which FDs are fulfilled?

- $A \rightarrow A$
- $A \rightarrow B$
- $A \rightarrow C$
- $A \rightarrow D$
- $A \rightarrow AB$
- $A \rightarrow BC$
- $A \rightarrow CD$
- $A \rightarrow ABC$

Note: *AB* is a shorthand notation for {*A, B*}

Armstrong's Axioms

AALBORG UNIVERSITY

- Reflexive rule
  - A set of attributes $\rightarrow$ A subset of the attributes
  - if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$
- Example
  - {*ID, name, salary*} $\rightarrow$ {*name*}
  - {*ID, name, salary*} $\rightarrow$ {*name, salary*}
  - {*ID, name, salary*} $\rightarrow$ {*ID, name*}
  - ABCD $\rightarrow$ ABC
  - ABCD $\rightarrow$ BCD
  - ABCD $\rightarrow$ ACD
  - ABCD $\rightarrow$ BC
  - ABCD $\rightarrow$ A
  - ABCD $\rightarrow$ C

Armstrong's Axioms

AALBORG UNIVERSITY

- Augmentation rule
  - If $\alpha \rightarrow \beta$, then $\gamma\alpha \rightarrow \gamma\beta$
- Example
  - Given {*ID*} $\rightarrow$ {*name*}
    - {*ID, salary*} $\rightarrow$ {*name, salary*}
    - { {*ID, dept_name*} $\rightarrow$ {*name, dept_name*}
  - Given A $\rightarrow$ B
    - AC $\rightarrow$ BC
    - AD $\rightarrow$ BD
    - ACD $\rightarrow$ BCD
    - ACD $\rightarrow$ ABCD

Armstrong's Axioms

AALBORG UNIVERSITY

- Transitivity rule
  - If $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$
- Example
  - Given {*ID*} $\rightarrow$ {*dept_name*}, {*dept_name*} $\rightarrow$ {*building*}
    - {*ID*} $\rightarrow$ {*building*}
  - Given A $\rightarrow$ B, B $\rightarrow$ C, C $\rightarrow$ D
    - A $\rightarrow$ C
    - A $\rightarrow$ D
    - B $\rightarrow$ D

Reasoning with FDs--Example

AALBORG UNIVERSITY

- Given $A \rightarrow B$, $BC \rightarrow D$
- Prove that $AC \rightarrow D$
- Proof
  - Given $A \rightarrow B$, we have $AC \rightarrow BC$ (Augmentation)
  - Given $AC \rightarrow BC$ and $BC \rightarrow D$, we have $AC \rightarrow D$ (Transitivity)

Exercise 1

AALBORG UNIVERSITY

- 1. Given $A \rightarrow B$, $D \rightarrow C$, prove that $AD \rightarrow BC$

- 2. Given $A \rightarrow C$, $AC \rightarrow D$, $AD \rightarrow B$, prove that $A \rightarrow B$

Exercise 1--Solution

AALBORG UNIVERSITY

- Given A $\rightarrow$ B, D $\rightarrow$ C
- Prove that AD $\rightarrow$ BC
- Proof
  - Given A $\rightarrow$ B, we have AD $\rightarrow$ BD (Augmentation)
  - Given D $\rightarrow$ C, we have BD $\rightarrow$ BC (Augmentation)
  - Given AD $\rightarrow$ BD and BD $\rightarrow$ BC, we have AD $\rightarrow$ BC (Transitivity)

Exercise 1--Solution

AALBORG UNIVERSITY

- Given A $\rightarrow$ C, AC $\rightarrow$ D, AD $\rightarrow$ B
- Prove that A $\rightarrow$ B
- Proof
  - Given A $\rightarrow$ C, we have A $\rightarrow$ AC (Augmentation)
  - Given A $\rightarrow$ AC and AC $\rightarrow$ D, we have A $\rightarrow$ D (Transitivity)
  - Given A $\rightarrow$ D, we have A $\rightarrow$ AD (Augmentation)
  - Given A $\rightarrow$ AD and AD $\rightarrow$ B, we have A $\rightarrow$ B (Transitivity)

Additional Rules

AALBORG UNIVERSITY

- Union rule
  - If $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$, then $\alpha \rightarrow \beta\gamma$ holds
- Example
  - Given {*ID*} $\rightarrow$ {*name*}, {*ID*} $\rightarrow$ {*salary*}
    - {*ID*} $\rightarrow$ {*name, salary*}
- Proof
  - Given $\alpha \rightarrow \beta$, we have $\alpha \rightarrow \alpha\beta$ (Augmentation)
  - Given $\alpha \rightarrow \gamma$, we have $\alpha\ \beta \rightarrow \gamma\beta$ (Augmentation)
  - Given $\alpha \rightarrow \alpha\beta$ and $\alpha\ \beta \rightarrow \gamma\beta$, we have $\alpha \rightarrow \beta\gamma$ (Transitivity)

Additional Rules

AALBORG UNIVERSITY

- Decomposition rule
  - If $\alpha \rightarrow \beta\gamma$ holds, then $\alpha \rightarrow \beta$ holds and $\alpha \rightarrow \gamma$ holds
- Example
  - Given $\{ID\} \rightarrow \{name, salary\}$
    - $\{ID\} \rightarrow \{name\}$ and $\{ID\} \rightarrow \{salary\}$
- Proof
  - Given $\beta\gamma$, we have $\beta\gamma \rightarrow \beta$ and $\beta\gamma \rightarrow \gamma$ (reflexivity)
  - Given $\alpha \rightarrow \beta\gamma$ and $\beta\gamma \rightarrow \beta$, we have $\alpha \rightarrow \beta$ (Transitivity)
  - Given $\alpha \rightarrow \beta\gamma$ and $\beta\gamma \rightarrow \gamma$, we have $\alpha \rightarrow \gamma$ (Transitivity)

Additional Rules

AALBORG UNIVERSITY

- Pseudotransitivity rule
  - If $\alpha \rightarrow \beta$ holds and $\gamma\beta \rightarrow \delta$ holds, then $\alpha\ \gamma \rightarrow \delta$
- Proof
  - Given $\alpha \rightarrow \beta$, we have $\alpha\gamma \rightarrow \gamma\beta$ (Augmentation)
  - Given $\alpha\gamma \rightarrow \gamma\beta$ and $\gamma\beta \rightarrow \delta$, we have $\alpha\gamma \rightarrow \delta$ (Transitivity)

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Closure of Attribute Sets

AALBORG UNIVERSITY

- Given a set of attributes α, define the closure of α under *F* (denoted by α+) as the set of attributes that are functionally determined by α under *F*
- *R = (A, B, C, G, H, I)*
- *F* = {$A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H$}
- $(AG)^+ = ABCGHI$

Closure of Attribute Sets

AALBORG UNIVERSITY

- Algorithm to compute α+, the closure of a under *F*

result := α;
while (changes to *result*) **do**
    for each $\beta \rightarrow \gamma$ **in** *F* **do**
        begin
            if $\beta \subseteq$ *result* **then** *result* := *result* $\cup \gamma$
        end

Attribute Sets Closure--Example

AALBORG UNIVERSITY

- $R = (A, B, C, G, H, I)$
- $F = \{A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H\}$
- $(AG)^+$
  - 1. $result = AG$
  - 2. $result = ABG$ ($A \rightarrow B$ and $A \subseteq AG$)
  - 3. $result = ABCG$ ($A \rightarrow C$ and $A \subseteq ABG$)
  - 4. $result = ABCGH$ ($CG \rightarrow H$ and $CG \subseteq ABCG$)
  - 5. $result = ABCGHI$ ($CG \rightarrow I$ and $CG \subseteq ABCGH$)

Attribute Sets Closure--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D, E, F)$
- $F = \{AB \rightarrow C$
  $AD \rightarrow E$
  $B \rightarrow D$
  $AF \rightarrow B\}$
- $(BC)^+$
  - 1. *result* = BC
  - 2. *result* = BCD ($B \rightarrow D$ and $B \subseteq BC$)

Attribute Sets Closure--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D, E, F)$
- $F = \{AB \rightarrow C$
  $AD \rightarrow E$
  $B \rightarrow D$
  $AF \rightarrow B\}$
- $(AB)^+$
  - 1. $result = AB$
  - 2. $result = ABC$ ($AB \rightarrow C$ and $AB \subseteq AB$)
  - 3. $result = ABCD$ ($B \rightarrow D$ and $B \subseteq ABC$)
  - 4. $result = ABCDE$ ($AD \rightarrow E$ and $AD \subseteq ABCD$)

Exercise 2

AALBORG UNIVERSITY

- $R = (A, B, C, D, E, F)$
- $F = \{AB \rightarrow C$
  $AD \rightarrow E$
  $B \rightarrow D$
  $AF \rightarrow B\}$
- $(AF)^+ = ???$

Exercise 2--Solution

AALBORG UNIVERSITY

- $R = (A, B, C, D, E, F)$
- $F = \{AB \rightarrow C$
  $AD \rightarrow E$
  $B \rightarrow D$
  $AF \rightarrow B\}$
- $(AF)^+$
  - 1. $result = AF$
  - 2. $result = ABF$ ($AF \rightarrow B$)
  - 3. $result = ABCF$ ($AB \rightarrow C$ and $AB \subseteq ABF$)
  - 4. $result = ABCDF$ ($B \rightarrow D$ and $B \subseteq ABCF$)
  - 5. $result = ABCDEF$ ($AD \rightarrow E$ and $AD \subseteq ABCDF$)

Closure & FD

AALBORG UNIVERSITY

- To prove that $X \rightarrow Y$ holds, we only need to show that $(X)^+$ contains Y.
- $R = (A, B, C, D, E, F)$
- $F = \{AB \rightarrow C$
  $AD \rightarrow E$
  $B \rightarrow D$
  $AF \rightarrow B\}$
- Prove that $AF \rightarrow D$
- $(AF)^+ = ABCDEF$, which contains D
- Therefore, $AF \rightarrow D$ holds

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Keys & FD

AALBORG UNIVERSITY

- *K* is a superkey for relation schema *R* if and only if $K \rightarrow R$
  - A set of attributes in a table that decides all other attributes
- *K* is a candidate key for *R* if and only if
  - $K \rightarrow R$, and
  - for no $\alpha \subset K$, $\alpha \rightarrow R$
  - if we remove any attribute from the candidate key, it will not decide all other attributes
  - A candidate key must be a superkey
- Example
  - *ID* is a candidate key
  - {*ID, name*} is a superkey
  - {*ID, salary*} is a superkey
  - {*ID, name, salary, dept_name*} is a superkey

ID	*name*	*salary*	*dept_name*
10101	Srinivasan	65000	Comp. Sci.
12121	Wu	90000	Finance
45565	Katz	75000	Comp. Sci.
58583	Califieri	62000	History
67131	Califieri	70000	Physics
76543	Singh	80000	Finance

Keys & FD--Example

AALBORG UNIVERSITY

- $R = (A, B, C, G, H, I)$
- $F = \{A \rightarrow B$
  $A \rightarrow C$
  $CG \rightarrow H$
  $CG \rightarrow I$
  $B \rightarrow H\}$
- $(AG)^+ = ABCGHI$
- Is AG a superkey? YES
- Is AG a candidate key? YES
  - AG is a superkey
  - No subset of AG is a superkey ($(A)^+ = ABCH$, $(G)^+ = G$)

Finding the Candidate Keys--Example

AALBORG UNIVERSITY

- Candidate Key: A minimal set of attributes that decides all other attributes
- $R = (A, B, C)$
- $F = \{A \rightarrow B, B \rightarrow C\}$
  - Is A a candidate key?
  - $(A)^+ = ABC$
  - Is B a candidate key?
  - $(B)^+ = BC$
  - Is C a candidate key?
  - $(C)^+ = C$
  - Is AB or BC or AC or ABC a candidate key?
  - $(AB)^+ = ABC$, $(BC)^+ = BC$, $(AC)^+ = ABC$, $(ABC)^+ = ABC$

Finding the Candidate Keys--Algorithm

AALBORG UNIVERSITY

- 1. Check all possible combinations of attributes in the table
  - Example: *A, B, C, AB, BC, AC, ABC*
- 2. For each combination, compute its closure
  - Example: $(A)^+ = \ldots$, $(B)^+ = \ldots$, $(C)^+ = \ldots$, …
- 3. If a closure contains all attributes, then the combination is a superkey, and might be a candidate key
  - Example: $(A)^+ = ABC$
- 4. Make sure that you select candidate keys
  - Example: $(A)^+ = ABC$, $(AB)^+ = ABC$, don't select *AB*

Algorithm--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D)$
- $F = \{AB \rightarrow C, AD \rightarrow B, B \rightarrow D\}$
- First, enumerate all attribute combinations:
  - *A, B, C, D*
  - *AB, AC, AD, BC, BD, CD*
  - *ABC, ABD, ACD, BCD*
  - *ABCD*

Algorithm--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D)$
- $F = \{AB \rightarrow C, AD \rightarrow B, B \rightarrow D\}$
- Second, compute the closures:
  - $(A)^+ = A$, $(B)^+ = BD$, $(C)^+ = C$, $(D)^+ = D$
  - $(AB)^+ = ABCD$, $(AC)^+ = AC$, $(AD)^+ = ABCD$, $(BC)^+ = BCD$, $(BD)^+ = BD$, $(CD)^+ = CD$
  - $(ABC)^+ = ABCD$, $(ABD)^+ = ABCD$, $(ACD)^+ = ABCD$, $(BCD)^+ = BCD$
  - $(ABCD)^+ = ABCD$

Algorithm--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D)$
- $F = \{AB \rightarrow C, AD \rightarrow B, B \rightarrow D\}$
- Third, find all superkeys:
  - $(A)^+ = A$, $(B)^+ = BD$, $(C)^+ = C$, $(D)^+ = D$
  - $(AB)^+ = ABCD$, $(AC)^+ = AC$, $(AD)^+ = ABCD$, $(BC)^+ = BCD$, $(BD)^+ = BD$, $(CD)^+ = CD$
  - $(ABC)^+ = ABCD$, $(ABD)^+ = ABCD$, $(ACD)^+ = ABCD$, $(BCD)^+ = BCD$
  - $(ABCD)^+ = ABCD$

Algorithm--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D)$
- $F = \{AB \rightarrow C, AD \rightarrow B, B \rightarrow D\}$
- Finally, select candidate keys:
  - $(A)^+ = A$, $(B)^+ = BD$, $(C)^+ = C$, $(D)^+ = D$
  - $(AB)^+ = ABCD$, $(AC)^+ = AC$, $(AD)^+ = ABCD$, $(BC)^+ = BCD$, $(BD)^+ = BD$, $(CD)^+ = CD$
  - $(ABC)^+ = ABCD$, $(ABD)^+ = ABCD$, $(ACD)^+ = ABCD$, $(BCD)^+ = BCD$
  - $(ABCD)^+ = ABCD$

Small Trick

AALBORG UNIVERSITY

- Always check small combinations first
- $R = (A, B, C, D)$
- $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A\}$
- Compute the closures:
  - $(A)^+ = ABCD$, $(B)^+ = ABCD$, $(C)^+ = ABCD$, $(D)^+ = ABCD$
  - No need to check others
  - The others are all superkeys but not candidate keys
- Candidate keys: *A, B, C, D*

Small Trick

AALBORG UNIVERSITY

- If an attribute that does not appear in the right side of any *FD*, then it must be in every key
- $R = (A, B, C, D)$
- $F = \{AB \rightarrow C, AD \rightarrow B, B \rightarrow D\}$
- Notice that *A* does not appear in the right side of any functional dependencies
- In that case, *A* must be in every key
- Candidate keys of *R*: *AB, AD* (From the previous slide)

Finding the Candidate Keys--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D)$
- $F = \{A \rightarrow B, A \rightarrow C, C \rightarrow D\}$
- A must be in every key
- Compute the closures:
  - $(A)^+ = ABCD$
  - No need to check others
- Keys: A

Finding the Candidate Keys--Example

AALBORG UNIVERSITY

- $R = (A, B, C, D, E)$
- $F = \{AB \rightarrow C, C \rightarrow B, BC \rightarrow D, CD \rightarrow E\}$
- A must be in every key
- Compute the closures:
  - $(A)^+ = A$
  - $(AB)^+ = ABCDE$
  - $(AC)^+ = ACBDE$
  - $(AD)^+ = AD$, $(AE)^+ = AE$
  - $(ADE)^+ = ADE$
  - No need to check others
- Keys: AB, AC

Exercise 3

AALBORG UNIVERSITY

- $R = (A, B, C)$
- $F = \{A \rightarrow B\}$
- Find candidate keys for R

Exercise 3--Solution

AALBORG UNIVERSITY

- $R = (A, B, C)$
- $F = \{A \rightarrow B\}$
  - Is A a candidate key?
  - $(A)^+ = AB$
  - Is B a candidate key?
  - $(B)^+ = B$
  - Is C a candidate key?
  - $(C)^+ = C$
  - Is AB or BC or AC or ABC a candidate key?
  - $(AB)^+ = AB$, $(BC)^+ = BC$, $(AC)^+ = ABC$, $(ABC)^+ = ABC$

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Lossless Decomposition

AALBORG UNIVERSITY

- 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,

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

- And, conversely a decomposition is lossy if

$$r \subset \prod_{R_1}(r) \bowtie \prod_{R_2}(r)$$

Example of Lossless Decomposition

AALBORG UNIVERSITY

- Decomposition of *R = (A, B, C)*
- $R_1 = (A, B)$, $R_2 = (B, C)$

A	B	C
α	1	A
β	2	B

r

A	B
α	1
β	2

$\Pi_{A,B}(r)$

B	C
1	A
2	B

$\Pi_{B,C}(r)$

$\Pi_A (r) \bowtie \Pi_B (r)$

A	B	C
α	1	A
β	2	B

Example of Lossy Decomposition

AALBORG UNIVERSITY

ID	*name*	*salary*	*dept_name*
…			
58583	Califieri	62000	History
67131	Califieri	70000	Physics
…			

instructor

ID	*name*
…	
58583	Califieri
67131	Califieri
…	

name	*salary*	*dept_name*
…		
Califieri	62000	History
Califieri	70000	Physics
…		

natural join

ID	*name*	*salary*	*dept_name*
…			
58583	Califieri	62000	History
58583	Califieri	70000	Physics
67131	Califieri	62000	History
67131	Califieri	70000	Physics
…			

Lossless Decomposition

AALBORG UNIVERSITY

- We can use functional dependencies to show when certain decomposition are lossless.
- Let R be a relation schema and let R_1 and R_2 form a decomposition of R. That is $R = R_1 \cup R_2$
- R_1 and R_2 is a lossless decomposition if we can derive at least one of the following FDs:
  - $(R_1 \cap R_2) \rightarrow R_1$, i.e., common attributes are superkey in R_1 or
  - $(R_1 \cap R_2) \rightarrow R_2$, i.e., common attributes are superkey in R_2

Lossless Decomposition--Example

AALBORG UNIVERSITY

- $R = (A, B, C)$
- $F = \{A \rightarrow B, B \rightarrow C\}$
- $R_1 = (A, B), \quad R_2 = (B, C)$
  - Lossless decomposition:
  - $R_1 \cap R_2 = \{B\}$ and $B \rightarrow BC$
- $R_1 = (A, B), \quad R_2 = (A, C)$
  - Lossless decomposition:
  - $R_1 \cap R_2 = \{A\}$ and $A \rightarrow AB$

Exercise 4

AALBORG UNIVERSITY

- Is this decomposition lossless?
- Department(*name, street, number, city, postcode*)
- FDs
  - {*name*} $\rightarrow$ {*street, number, city, postcode*}
  - {*street, number, city*} $\rightarrow$ {*postcode*}
  - {*street, number, city*} $\rightarrow$ {*name*}
- Decomposition
  - dept1(*name, street, number, city*)
  - dept2(*name, postcode*)

Exercise 4--Solution

AALBORG UNIVERSITY

- Decomposition
  - dept1(*name, street, number, city*)
  - dept2(*name, postcode*)
- FDs
  - {*name*} → {*street, number, city, postcode*}
  - {*street, number, city*} → {*postcode*}
  - {*street, number, city*} → {*name*}
- Decomposition
- Is dept1 ∩ dept2 a superkey of dept 1 or dept 2? Yes
  - {*name*} → {*name, street, number, city*}
  - {*name*} → {*name, postcode*}

Dependency Preserving

AALBORG UNIVERSITY

- All functional dependencies that hold for R must be verifiable in the new schemas $R_1, ..., R_n$
- We can check all dependencies locally on $R_1, ..., R_n$
- We avoid the alternative: computing the join $R_1 \bowtie \cdots \bowtie R_n$ to test if an FD is violated
- Let F_R, F_{Ri} be the set of FDs in R and R_i, the decomposition of R into $R_1, ..., R_n$ is dependency preserving if
  - $F^+_R = (F_{R1} \cup \ldots \cup F_{Rn})^+$
- Computing F^+_R and $(F_{R1} \cup \ldots \cup F_{Rn})^+$ is expensive

Dependency Preserving--Example

AALBORG UNIVERSITY

- Is this decomposition dependency preserving?
- *R = (A, B, C)*
- $F = \{A \rightarrow B, B \rightarrow C\}$
- $R_1 = (A, B), \quad R_2 = (B, C)$

- yes!
- $A \rightarrow B$ can be checked locally in R_1
- $B \rightarrow C$ can be checked locally in R_2

Relation Design and Decompositions

AALBORG UNIVERSITY

- A good relation design has no redundancy, no update anomalies, ability to represent all the information
- A bad design can be converted to a good design by decomposing large relational schemas into smaller ones
- A good decomposition is lossless and dependency preserving

Agenda

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Normal Forms

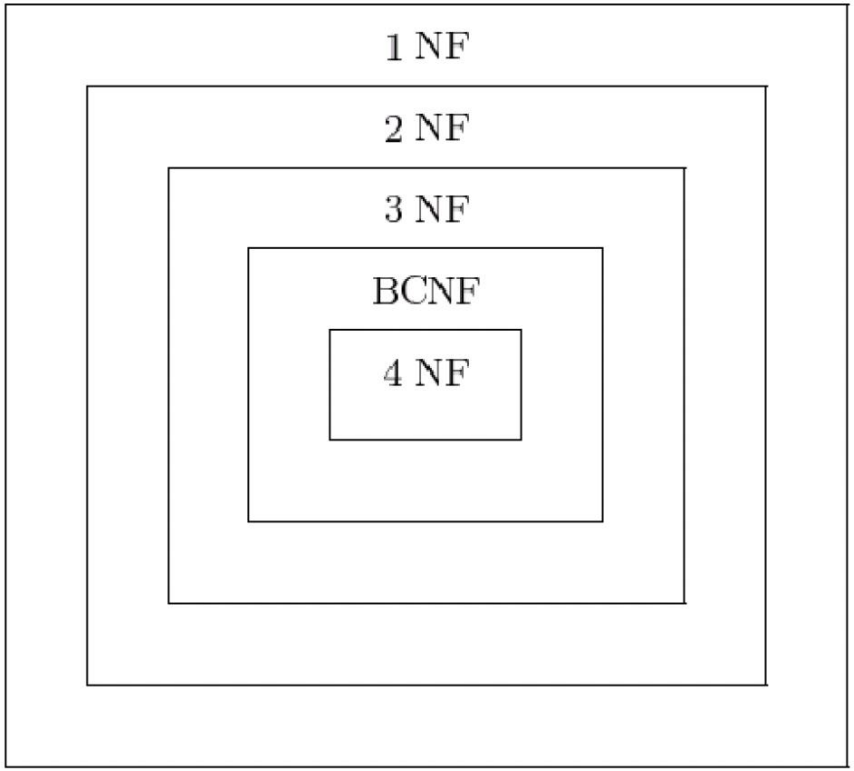
AALBORG UNIVERSITY

- Normal Forms define characteristics of relational schemas
- NFs forbid certain combinations of FDs in a relation
- NFs avoid redundancies and anomalies
- NFs are guidelines to obtain good decompositions
- Examples: 1NF, 2NF, **3NF, BCNF**, 4NF

Normal Forms

AALBORG UNIVERSITY

- Examples: 1NF, 2NF, 3NF, BCNF, 4NF



First Normal Form (1NF)

AALBORG UNIVERSITY

- *R* is in 1NF if the domains of all its attributes are atomic (no composite or set-valued domains)

i_id	*name*	*course_id*
10101	Srinivasan	CS-101, CS-120, CS-131
12121	Wu	FIN-101
15151	Mozart	MUS-101, MUS-145
83821	Brandt	CS-190
98345	Kim	EE-190

Not in 1NF

i_id	*name*	*course_id*
10101	Srinivasan	CS-101
10101	Srinivasan	CS-120
10101	Srinivasan	CS-131
12121	Wu	FIN-101
15151	Mozart	MUS-101
15151	Mozart	MUS-145
83821	Brandt	CS-190
98345	Kim	EE-190

In 1NF

Second Normal Form (2NF)

AALBORG UNIVERSITY

- *R* is in 2NF if it is in 1NF and every non-key attribute is functionally dependent on the whole of every candidate key.

i_id	*name*	***course_id***
10101	Srinivasan	CS-101
10101	Srinivasan	CS-120
10101	Srinivasan	CS-131
12121	Wu	FIN-101
15151	Mozart	MUS-101
15151	Mozart	MUS-145
83821	Brandt	CS-190
98345	Kim	EE-190

$i\_id \rightarrow name$

Not in 2NF

Third Normal Form (3NF)

AALBORG UNIVERSITY

- 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., $\beta \in \alpha$)
  - α 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)

3NF--Example

AALBORG UNIVERSITY

- Candidate key: {*ID*}
- F = {

  $\{ID\} \rightarrow \{name, dept\_name, building\}$

  $\{dept\_name\} \rightarrow \{building\}$

  }
- Is this relation in 3NF?
  - True
  - False

ID	*name*	*dept_name*	*building*
10101	Srinivasan	Comp. Sci.	Taylor
12121	Wu	Finance	Painter
15151	Mozart	Music	Packard
83821	Brandt	Comp. Sci.	Taylor
98345	Kim	Elec. Eng.	Taylor

3NF--Example

AALBORG UNIVERSITY

- Candidate key: {*ID*}
- F = {

  $\{ID\} \rightarrow \{name, dept\_name, building\}$

  $\{dept\_name\} \rightarrow \{building\}$

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

ID	*name*	*dept_name*	*building*
10101	Srinivasan	Comp. Sci.	Taylor
12121	Wu	Finance	Painter
15151	Mozart	Music	Packard
83821	Brandt	Comp. Sci.	Taylor
98345	Kim	Elec. Eng.	Taylor

3NF--Example

AALBORG UNIVERSITY

- Candidate key: {*ID*}
- F = {

  $\{ID\} \rightarrow \{name, dept\_name, building\}$

  $\{dept\_name\} \rightarrow \{building\}$

  }

ID	*name*	*dept_name*	*building*
10101	Srinivasan	Comp. Sci.	Taylor
12121	Wu	Finance	Painter
15151	Mozart	Music	Packard
83821	Brandt	Comp. Sci.	Taylor
98345	Kim	Elec. Eng.	Taylor

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

Boyce Codd Normal Form (BCNF)

AALBORG UNIVERSITY

- A relation schema *R* is in **Boyce Codd Normal Form (BCNF)** if for all:

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

- at least one of the following holds:
  - $\alpha \rightarrow \beta$ is trivial (i.e., $\beta \in \alpha$)
  - α is a superkey for *R*
  - ~~Each attribute *A* in $\beta - \alpha$ is contained in a candidate key for *R*.~~
- Difference to 3NF: no third option
- BCNF is a stricter form of 3NF ("includes" 3NF)

Summary

AALBORG UNIVERSITY

- Introduction of Normalization Theory
- Functional Dependency
- Attribute Sets Closure
- Keys
- Decomposition
- Normal Forms

Next Lecture

AALBORG UNIVERSITY

- Physical Design
  - Understand how tables are stored in files
  - Understand basic indexing techniques
  - Understand the effects of storage and indexes on the performance of basic SQL queries