Week 9 Lecture Notes: 30.03.2015 – 03.04.2015

Note Takers: Sarper Aydın – Semih Kaldırım

CHAPTER 19: Functional Dependencies

- The functional dependency X →Y says either of the following : i) X determines Y and ii) Y is dependent on X. In a relationship, these functional dependencies should be **TRUE** for **ALL** instances of a relation
- Functional dependencies of a relation can also be taken as integrity constraints for that relation (i.e correctness requirements).
- Functional Dependencies can be used to identify problems (as they are a source of for the problems) in relational schemas.
- These problems are update, addition and deletion anomalies.

Relational Decomposition (Schema Refinement)

- Relational decomposition is a method to get rid of the problems arising from the functional dependencies.
- In order to do this there exists certain normal forms for relations which is known to reduce these problems.
- These normal forms are: First Normal Form (1NF), 2NF, 3NF and Boyce-Codd Normal Form.

Consider the following example: Employee (eno, ename, city , status)

eno	ename	city	status
e1	Ali	34	1
e2	Veli	34	1
e3	Oya	35	1
e4	Hasan	35	1
e5	Merve	26	2

Problems regarding this relation:

i) Redundancy: The entries entered in red (color) are a source for redundancy as status of the city 34 is repeated twice (or already entered in first tuple).

ii) How to add a new city? A person working in there is needed so that a new city can be added into relation → This leads to the addition anomaly.

iii) To change the status of city 34, we need to change several (more than one) tuples → This leads to update anomaly.

iv) Delete 'Merve' → We lose the info that city='26' has status='2' → This leads to the deletion anomaly.

The method behind the normalizations is the following:

One Large Table → Normalization is Done → Smaller Tables

By this methodology, the problems arising from the functional dependencies can be avoided.

For the mentioned problems, consider the following relational decomposition:

Employee → Emp (eno, ename, ecity) & CityStatus (city, status)

eno	ename	city
e1	Ali	34
e2	Veli	34
e3	Oya	35
e4	Hasan	35
e5	Merve	26

city	status
34	1
35	1
26	2

What did we gain by this decomposition? Consider the following query in relational algebra: Display the status of e1

i) Before relational decomposition: $\pi_{status} (\sigma_{eno=e1} \text{ (Employee)})$

ii) After relational decomposition: $\pi_{status} (\sigma_{eno=e1} \text{ (Emp)} \bowtie \text{CityStatus})$

iii) Emp ⋈ CityStatus gives the whole Employee relation. Such a decomposition is called Lossless Decomposition. If it's otherwise, it's called Lossy Decomposition.

First Normal Form (1NF)

- Each attribute or column must have an atomic value (not a set value but a singular value)

Consider the following relation: students (stno, stname, dept, hobbies)

stno	stname	dept	hobbies
s1	Ali	IE	reading, music , hiking

We decompose it in the following way: students1 (stno, stname, dept) & students2 (stno, hobby)

- t1, t2 are tuples. If X→Y, then t1 . X → t1 . Y or t2 . X → t2 . Y

stno	stname	dept
s1	Ali	IE

stno	hobby
s1	reading
s1	music
s1	hiking

- As seen from the decomposition, all attributes have an atomic value. Therefore, this is a decomposition into First Normal Form.

Hierarchy of Normal Forms:

- BCNF is a subset of 3NF; 3NF is a subset of 2NF; 2NF is a subset of 1NF

Example regarding relational decomposition and functional dependencies: R (A, B, C, D, E) and ABC→E and A= {a1, a2, a3} and E = {e1, e2, e3}

A	B	C	D	E
a1	b1	c1	d1	e1
1	b1	c2	d2	e2
2	b1	c1	d3	e3
a3	b2	c1	d4	3
a1	b1	c1	d5	4
5	b1	c1	d6	e1

We are trying to deduce the entries in red (1, 2, 3, 4):

- 5 = a1 (From the functional dependency we see that D has no effect in determining E. So from the first tuple, we can deduce that a1,b1,c1 determines the E value as e1 so A=a1).
- 4 = e1 (From the sixth tuple and functional dependency we see that D has no effect in determining E. So from the first tuple, we can deduce that a1,b1,c1 determines the E value as e1).
- 2= a3 or a2 (Because B and C values are same as first tuple, the difference in E comes from the difference in the A entry but we can not know for sure if it is a3 or a2).
- 1= a1 or a2 or a3 (Because the difference in E can either come from different A or C values)
- 3 = e1 or e2 or e3 (Different A and B values may result in either of the E values)

Armstrong Axioms

1) **Reflexivity**

   $X \supseteq Y$ then $X \rightarrow Y$

   {Name, Last Name} → {Name}

   Trivial functional dependecy

2) **Augmentation**

   $X \rightarrow Y$ then $XZ \rightarrow YZ$

3) **Transivity**

   $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Derived Axioms

Union

$X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

Derivatives using by Augmentation and Transivity

$X \rightarrow Y$ augment it with Z then $XZ \rightarrow YZ$

$X \rightarrow Z$ augment it with X then $XX \rightarrow XZ$

From transivity

$XX \rightarrow XZ \rightarrow YZ \Rightarrow X \rightarrow YZ$

Decomposition

$X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

Use Reflexivity and Transivity

YZ→ Y

YZ→ Z

From Transivity,

X→ YZ→ Y then X→ Y

Y→ YZ→ Z then X→ Z

Pseudo Transivity

X→ Y and YZ→ W then XZ→ W

Augmentate XZ→ YZ

XZ→ YZ

YZ→ W

Then XZ→ W

Accumulation

X→ Y and Y→ Z then X→ YZ

Proving validty of a Functional Dependency

EX: X→ Y and W→ Z

Then, does XW→YZ hold?

X→ Y

By Augmentation ,

XZ→ YZ

W→ Z

By Augmentation,

WX→ ZX

XW→XA

From Transivity,

XW→YZ

Implications (Results) and FD Clousure

The set of all FDs implied by a set of given FDs is called the clouse F^+.

F: { X→ Y, Y→ Z }

F^+ : { X→ Y, Y→ Z, X→ Z}

Armstrong Axioms are sound and complete in terms of F^+.

- Soundness: All FDs are found by applying Armstrong Axioms are in F^+. Nothing extra.

- Completeness: All FDs in F^+ can be generated by the application of the axioms of the original set of F.

Important Note: Any FD generated by appliying R(reflexivity) is called trivial FD.

F^+ Example

	A	B	C	AB	AC	BC	ABC	
A	R		G					
B		R						
C			R					
AB	R	R	(1)R,T	R	(4)R		R,A,T	
AC	R		R		R			
BC	G	R	R	(2) A	A	R	(3)A	
ABC	R	R	R	R	R	R	R	

R: Reflexivity

A: Augmentation

T: Transitive

G: Given

1) R,T

$AB \rightarrow C$

$AB \rightarrow A$

$A \rightarrow C$

2) $BC \rightarrow AC$

$BC \rightarrow A$

$BBC \rightarrow AB$

$BC \rightarrow AB$

3) $BC \rightarrow ABC$

$BC \rightarrow A$

$ABC \rightarrow AA$

$ABC \rightarrow A$

$BC \rightarrow ABC$

$BC \rightarrow A$

$BBCC \rightarrow ABC$

$BC \rightarrow ABC$

4) $AB \rightarrow A$

$A \rightarrow C$

$AAB \rightarrow AC$

$AB \rightarrow AC$