Lecture 7: Normal Forms, Relational Algebra

Monday, 10/9/2000

Outline

- BCNF, 3NF (3.7)
- Multivalued Dependencies, 4NF (3.8)
- Relational Algebra (4.1)

Boyce-Codd Normal Form

A simple condition for removing anomalies from relations:

A relation R is in BCNF if and only if:

Whenever there is a nontrivial dependency $A_1, A_2, \ldots A_n \rightarrow B$ for R , it is the case that $\{ A_1, A_2, \ldots A_n \}$ a super-key for R.

In English (though a bit vague):

Whenever a set of attributes of *R* is determining another attribute, should determine ***all*** the attributes of *R*.

Example

Name	SSN	Phone Number
Fred	123-321-99	(201) 555-1234
Fred	123-321-99	(206) 572-4312
Joe	909-438-44	(908) 464-0028
Joe	909-438-44	(212) 555-4000

What are the dependencies?
SSN → Name
What are the keys?
Name, SSN, PhoneNumber
Is it in BCNF?

Decompose it into BCNF

SSN	Name
123-321-99	Fred
909-438-44	Joe

SSN → Name

SSN	Phone Number
123-321-99	(201) 555-1234
123-321-99	(206) 572-4312
909-438-44	(908) 464-0028
909-438-44	(212) 555-4000

SSN → PhoneNumber

What About This?

Name	Price	Category
Gizmo	$19.99	gadgets
OneClick	$24.99	camera

Name → Price, Category

It is already in BCNF

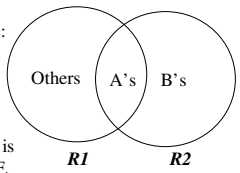
BCNF Decomposition

Find a dependency that violates the BCNF condition:

$A_1, A_2, \ldots A_n \longrightarrow B_1, B_2, \ldots B_m$

Heuristics: choose $B_1, B_2, \ldots B_m$ "as large as possible"

Decompose:



R1 ***R2***

Find a 2-attribute relation that is not in BCNF.

Continue until there are no BCNF violations left.

Example Decomposition

Person:

Name	SSN	Age	EyeColor	PhoneNumber

Functional dependencies:

SSN ⟶ Name, Age, Eye Color

BNCF: Person1(SSN, Name, Age, EyeColor),
Person2(SSN, PhoneNumber)

What if we also had an attribute Draft-worthy, and the FD:

Age ⟶ Draft-worthy

Other Example

- R(A,B,C,D) A ⟶ B, B ⟶ C
- Key: A, D
- Violations of BCNF: A ⟶ B, A ⟶ C, A ⟶ BC
- Pick A ⟶ BC: split into R1(A,BC) R2(A,D)
- What happens if we pick A ⟶ B first ?

Correct Decompositions

A decomposition is *lossless* if we can recover:

R(A,B,C)

↓

R1(A,B) R2(A,C)

↓

R'(A,B,C) = R(A,B,C)

R' is in general larger than R. Must ensure R' = R

Decomposition Based on BCNF is Necessarily Lossless

Attributes A, B, C. FD: A ⟶ C

Relations R1(A,B) R2(A,C)

Tuple in R: (a,b,c)
Tuples in R1: (a,b), (a,b')
Tuples in R2: (a,c), (a,c')

Tuples in the join of R1 and R2: (a,b,c), (a,b,c'), (a,b',c), (a,b',c')

Can (a,b,c') be a bogus tuple? What about (a,b',c') ?

3NF: A Problem with BCNF

Unit	Company	Product

FD's: Unit -> Company; Company, Product -> Unit
So, there is a BCNF violation, and we decompose.

Unit	Company

Unit ⟶ Company

Unit	Product

No FDs

So What's the Problem?

Unit	Company
Galaga99	UW
Bingo	UW

Unit	Product
Galaga99	databases
Bingo	databases

No problem so far. All *local* FD's are satisfied.
Let's put all the data back into a single table again:

Unit	Company	Product
Galaga99	UW	databases
Bingo	UW	databases

Violates the dependency: company, product -> unit!

Solution: 3rd Normal Form (3NF)

A simple condition for removing anomalies from relations:

A relation R is in 3rd normal form if and only if:

Whenever there is a nontrivial dependency $A_1, A_2, \ldots A_n \rightarrow B$ for R , it is the case that $\{ A_1, A_2, \ldots A_n \}$ a super-key for R, or B is part of a key.

What happened to first and second normal forms?

Will we have more normal forms?

Multi-valued Dependencies

SSN	Phone Number	Course
123-321-99	(206) 572-4312	CSE-444
123-321-99	(206) 572-4312	CSE-341
123-321-99	(206) 432-8954	CSE-444
123-321-99	(206) 432-8954	CSE-341

The multi-valued dependencies are:

SSN $\twoheadrightarrow$ Phone Number
SSN $\twoheadrightarrow$ Course

Definition of Multi-valued Dependecy

Given *R(A1,…,An,B1,…,Bm,C1,…,Cp)*

the MVD $A1,\ldots,An \twoheadrightarrow B1,\ldots,Bm$ holds if:

for any values of *A1,…,An* the "set of values" of *B1,…,Bm* is "independent" of those of *C1,…Cp*

Definition of MVDs Continued

Equivalently: the decomposition into

R1(A1,…,An,B1,…,Bm), R2(A1,…,An,C1,…,Cp)

is lossless

Note: an MVD $A1,\ldots,An \twoheadrightarrow B1,\ldots,Bm$
Implicitly talks about "the other" attributes *C1,…Cp*

Rules for MVDs

If $A1,\ldots An \rightarrow B1,\ldots,Bm$

then $A1,\ldots,An \twoheadrightarrow B1,\ldots,Bm$

Other rules in the book

4th Normal Form (4NF)

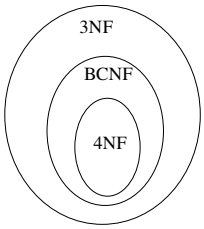
R is in 4NF if whenever:

A1,…,An B1,…,Bm

is a nontrivial MVD, then *A1,…,An* is a superkey

Same as BCNF with FDs replaced by MVDs

Confused by Normal Forms ?



In practice: (1) 3NF is enough, (2) don't overdo it !

More Normal Forms

- 1st normal form (1NF):
  - just make all attributes atomic
- 2nd normal form (2NF):
  - weaker than 3NF
  - no longer used

Querying the Database

- How do we specify what we want from our database?

  Find all the employees who earn more than $50,000 and pay taxes in New Jersey.
- We design *high-level query languages:*
  - SQL (used everywhere)
  - Datalog (used by database theoreticians, their students, friends and family)
- **Relational algebra:** a basic set of operations on relations that provide the *basic principles*.

Relational Algebra at a Glance

- Operators: relations as input, new relation as output
- Five basic RA operators:
  - Basic Set Operators
    - union, difference (no intersection, no complement)
  - Selection: σ
  - Projection: π
  - Cartesian Product: X
- Derived operators:
  - Intersection, complement
  - Joins (natural,equi-join, theta join, semi-join)
- When our relations have attribute names:
  - Renaming: ρ

Set Operations

- Binary operations
- Union: all tuples in R1 or R2
  - R1 U R2
  - Example:
    - ActiveEmployees U RetiredEmployees
- Difference: all tuples in R1 and not in R2
  - R1 – R2
  - Example
    - AllEmployees - RetiredEmployees

Selection

- Unary operation: returns a subset of the tuples which satisfy some condition
- Notation: σ_c (R)
- *c* is a condition:
  - =, <, >, and, or, not
- Find all employees with salary more than $40,000:
  - $\sigma_{Salary > 40000}$ (Employee)

Selection Example

Employee

SSN	Name	DepartmentID	Salary
999999999	John	1	30,000
777777777	Tony	1	32,000
888888888	Alice	2	45,000

Find all employees with salary more than $40,000.

SSN	Name	DepartmentID	Salary
888888888	Alice	2	45,000

Projection

- Unary operation: returns certain columns
- Eliminates duplicate tuples !
- Notation: $\Pi_{A1,\ldots,An}$ (R)
- Example: project social-security number and names:
  - $\Pi_{SSN, Name}$ (Employee)

Projection Example

Employee

SSN	Name	DepartmentID	Salary
999999999	John	1	30,000
777777777	Tony	1	32,000
888888888	Alice	2	45,000

SSN	Name
999999999	John
777777777	Tony
888888888	Alice

Cartesian Product

- Binary Operation
- Result is tuples combining any element of R1 with any element of R2, for R1 X R2
- Schema is union of Schema(R1) & Schema(R2)
- Notation: R1 x R2
- Example: Employee x Dependents
- Very rare in practice; but joins are very often

Cartesion Product Example

Employee

Name	SSN
John	999999999
Tony	777777777

Dependents

EmployeeSSN	Dname
999999999	Emily
777777777	Joe

Employee_Dependents

Name	SSN	EmployeeSSN	Dname
John	999999999	999999999	Emily
John	999999999	777777777	Joe
Tony	777777777	999999999	Emily
Tony	777777777	777777777	Joe

Join (Natural)

- Most important, expensive and exciting.
- Combines two relations, selecting only related tuples
- Equivalent to a cross product followed by selection
- Resulting schema has all attributes of the two relations, but one copy of join condition attributes

Join Example

Employee

Name	SSN
John	999999999
Tony	777777777

Dependents

EmployeeSSN	Dname
999999999	Emily
777777777	Joe

Employee_Dependents

Name	SSN	Dname
John	999999999	Emily
Tony	777777777	Joe