Lecture 10: Database Design and Relational Algebra

Monday, October 21, 2002

1

Outline

- Design of a Relational schema (3.6)
- Relational Algebra (5.2)
- Operations on bags (5.3, 5.4)
  - Reading assignment 5.3 and 5.4 (won't have time to cover in class)

2

Boyce-Codd Normal Form

A simple condition for removing anomalies from relations:

> A relation R is in BCNF if:
>
> If $A_1, ..., A_n \rightarrow B$ is a non-trivial dependency in R , then $\{A_1, ..., A_n\}$ is a key for R

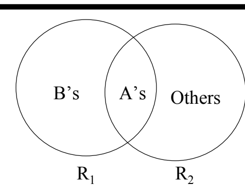
In English (though a bit vague):

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

3

BCNF Decomposition Algorithm

> Repeat
> choose $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ that violates the BNCF condition
> split R into $R_1(A_1, \ldots, A_m, B_1, \ldots, B_n)$ and $R_2(A_1, \ldots, A_m$, [others])
> continue with both R_1 and R_2
> Until no more violations



Is there a 2-attribute relation that is not in BCNF ?

4

Example

Name	SSN	PhoneNumber	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield
Joe	987-65-4321	908-555-1234	Westfield

What are the dependencies?

SSN → Name, City

What are the keys?

{SSN, PhoneNumber}

Is it in BCNF?

5

Decompose it into BCNF

Name	SSN	City
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

SSN → Name, City

SSN	PhoneNumber
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121
987-65-4321	908-555-1234

6

Example Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

SSN → name, age

age → hairColor

Decompose in BCNF (in class):

Step 1: find all keys (How ? Compute S^+, for various sets S)

Step 2: now decompose

7

Other Example

- R(A,B,C,D) A → B, B → C
- Key: AD
- 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 ?

8

Lossless Decompositions

A decomposition is *lossless* if we can recover:

R(A,B,C)

↓ Decompose

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

↓ Recover

R'(A,B,C) should be the same as R(A,B,C)

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

9

Lossless Decompositions

- Given R(A,B,C) s.t. A→B, the decomposition into R1(A,B), R2(A,C) is lossless

10

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

Notice: we loose the FD: Company, Product → Unit

11

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!

12

Solution: 3rd Normal Form (3NF)

A simple condition for removing anomalies from relations:

> A relation R is in 3rd normal form if :
>
> Whenever there is a nontrivial dependency $A_1, A_2, ..., A_n \rightarrow B$ for R , then $\{A_1, A_2, ..., A_n\}$ a super-key for R, or B is part of a key.

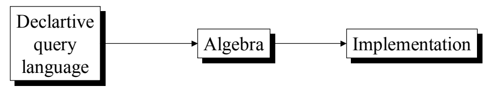
Tradeoff:

BCNF = no anomalies, but may lose some FDs

3NF = keeps all FDs, but may have some anomalies

Relational Algebra

- Formalism for creating new relations from existing ones
- Its place in the big picture:



SQL, relational calculus

Relational algebra
Relational bag algebra

Relational Algebra

- Five operators:
  - Union: $\cup$
  - Difference: -
  - Selection: σ
  - Projection: Π
  - Cartesian Product: $\times$
- Derived or auxiliary operators:
  - Intersection, complement
  - Joins (natural,equi-join, theta join, semi-join)
  - Renaming: ρ

1. Union and 2. Difference

- $R1 \cup R2$
- Example:
  - ActiveEmployees $\cup$ RetiredEmployees

- R1 – R2
- Example:
  - AllEmployees -- RetiredEmployees

What about Intersection ?

- It is a derived operator
- $R1 \cap R2 = R1 – (R1 – R2)$
- Also expressed as a join (will see later)
- Example
  - UnionizedEmployees $\cap$ RetiredEmployees

3. Selection

- Returns all tuples which satisfy a condition
- Notation: $\sigma_c(R)$
- Examples
  - $\sigma_{Salary > 40000}$(Employee)
  - $\sigma_{name = \text{"Smithh"}}$(Employee)
- The condition c can be =, <, ≤, >, ≥, <>

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.

$\sigma_{\text{Salary > 40000}}$ (Employee)

SSN	Name	DepartmentID	Salary
888888888	Alice	2	45,000

19

4. Projection

- Eliminates columns, then removes duplicates
- Notation: $\Pi_{A1,\ldots,An}(R)$
- Example: project social-security number and names:
  - $\Pi_{\text{SSN, Name}}$ (Employee)
  - Output schema: Answer(SSN, Name)

20

Projection Example

Employee

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

$\Pi_{\text{SSN, Name}}$ (Employee)

SSN	Name
999999999	John
777777777	Tony
888888888	Alice

21

5. Cartesian Product

- Each tuple in R1 with each tuple in R2
- Notation: R1 × R2
- Example:
  - Employee × Dependents
- Very rare in practice; mainly used to express joins

22

Cartesian Product Example

Employee

Name	SSN
John	999999999
Tony	777777777

Dependents

EmployeeSSN	Dname
999999999	Emily
777777777	Joe

Employee x Dependents

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

23

Relational Algebra

- Five operators:
  - Union: ∪
  - Difference: -
  - Selection: σ
  - Projection: Π
  - Cartesian Product: ×
- Derived or auxiliary operators:
  - Intersection, complement
  - Joins (natural,equi-join, theta join, semi-join)
  - Renaming: ρ

24

Renaming

- Changes the schema, not the instance
- Notation: $\rho_{B1,\ldots,Bn}$ (R)
- Example:
  - $\rho_{LastName,\ SocSocNo}$ (Employee)
  - Output schema: Answer(LastName, SocSocNo)

Renaming Example

Employee

Name	SSN
John	999999999
Tony	777777777

$\rho_{LastName,\ SocSocNo}$ (**Employee**)

LastName	SocSocNo
John	999999999
Tony	777777777

Natural Join

- Notation: R1 ⋈ R2
- Meaning: R1 ⋈ R2 = $\Pi_A(\sigma_C(R1 \times R2))$
- Where:
  - The selection σ_C checks equality of all common attributes
  - The projection eliminates the duplicate common attributes

Natural Join Example

Employee

Name	SSN
John	999999999
Tony	777777777

Dependents

SSN	Dname
999999999	Emily
777777777	Joe

Employee ⋈ Dependents =

$\Pi_{Name,\ SSN,\ Dname}(\sigma_{SSN=SSN2}(\text{Employee x } \rho_{SSN2,\ Dname}(\text{Dependents}))$

Name	SSN	Dname
John	999999999	Emily
Tony	777777777	Joe

Natural Join

- R=

A	B
X	Y
X	Z
Y	Z
Z	V

S=

B	C
Z	U
V	W
Z	V

- R ⋈ S=

A	B	C
X	Z	U
X	Z	V
Y	Z	U
Y	Z	V
Z	V	W

Natural Join

- Given the schemas R(A, B, C, D), S(A, C, E), what is the schema of R ⋈ S ?
- Given R(A, B, C), S(D, E), what is R ⋈ S ?
- Given R(A, B), S(A, B), what is R ⋈ S ?

Theta Join

- A join that involves a predicate
- R1 $\bowtie_\theta$ R2 = σ_θ (R1 $\times$ R2)
- Here θ can be any condition

31

Eq-join

- A theta join where θ is an equality
- R1 $\bowtie_{A=B}$ R2 = $\sigma_{A=B}$ (R1 $\times$ R2)
- Example:
  - Employee $\bowtie_{SSN=SSN}$ Dependents
- Most useful join in practice

32

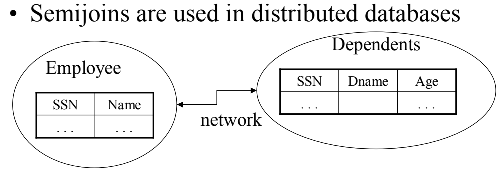
Semijoin

- R $\ltimes$ S = $\Pi_{A1,\ldots,An}$ (R $\bowtie$ S)
- Where $A_1, \ldots, A_n$ are the attributes in R
- Example:
  - Employee $\ltimes$ Dependents

33

Semijoins in Distributed Databases

- Semijoins are used in distributed databases



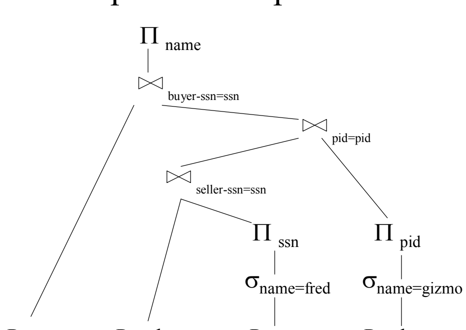
Employee $\bowtie_{ssn=ssn}$ ($\sigma_{age>71}$ (Dependents))

T = Π_{SSN} $\sigma_{age>71}$ (Dependents)

R = Employee $\ltimes$ T

Answer = R $\bowtie$ Dependents

Complex RA Expressions



35

Operations on Bags

A **bag** = a set with repeated elements

All operations need to be defined carefully on bags

- {a,b,b,c}$\cup${a,b,b,b,e,f,f}={a,a,b,b,b,b,b,c,e,f,f}
- {a,b,b,b,c,c} – {b,c,c,c,d} = {a,b,b,d}
- σ_C(R): preserve the number of occurrences
- Π_A(R): no duplicate elimination
- Cartesian product, join: no duplicate elimination

Important ! Relational Engines work on bags, not sets !

Reading assignment: 5.3 – 5.4

36

Finally: RA has Limitations !

- Cannot compute "transitive closure"

Name1	Name2	Relationship
Fred	Mary	Father
Mary	Joe	Cousin
Mary	Bill	Spouse
Nancy	Lou	Sister

- Find all direct and indirect relatives of Fred
- Cannot express in RA !!! Need to write C program

37