Relational Database Design

- Finding database schemes with good properties
- Example:
  - Database for information on suppliers, parts supplied, and shipments
  - Information consists of:
    - S#: supplier number
    - SNAME: supplier name
    - SCITY: supplier city
    - P#: part number
    - PNAME: part name
    - PCITY: city where part is stored
    - QTY: quantity of shipment

Relational Database Design

- One possible database schema
  - BAD[S#, SNAME, SCITY, P#, PNAME, PCITY, QTY]
- Why BAD is bad:
  - redundancy
    - SCITY is determined by S#

    However, the same SCITY could appear many times with the same S# in this relation
    - Functional dependency: S# $\rightarrow$ SCITY
  - update anomalies
    - SCITY could be changed in one place but not in another, for the same S# $\rightarrow$ inconsistency

Relational Database Design

- Why BAD is bad:
  - insertion anomalies
    - cannot record supplier info unless that supplier supplies a part
  - deletion anomalies
    - by deleting parts we can lose supplier info
- Solution
  - New schema:

S[S#, SNAME, SCITY]	Key: S#
P[P#, PNAME, PCITY]	Key: P#
SP[S#, P#, QTY]	Key: S# P#

    - No other functional dependencies besides key dependencies
    - Normal forms: “nice” forms for schemas

Decomposing Relation Schemes

- Let R be a relation scheme
- A decomposition of R is a set $\rho=\{R_1, \ldots, R_k\}$ of relation schemes such that:
  - 1. $att(R_i) \subseteq att(R)$, $1 \le i \le k$
  - 2. $att(R) = \cup^k_{i=1} att(R_i)$
- Example
  - {S, P, SP} is a decomposition of the relation schema BAD (see earlier example)

Conditions for a "Reasonable" Decomposition

- Do S, P, SP contain the same info as BAD?
  - $BAD = \pi_S(BAD) \bowtie \pi_P(BAD) \bowtie \pi_{SP}(BAD)$?
- Can the dependencies for BAD be enforced by "local" dependencies on S, P, SP?
  - Let F be the set of fd's for BAD and
  - $\pi_X(F^+) = \{V \rightarrow W \in F^+ \mid VW \subset X\}$

Is $\pi_S(F^+) \cup \pi_P(F^+) \cup \pi_{SP}(F^+)$ equivalent (cover) to F ?

Lossless Join

- Let R be a relation schema and D a set of dependencies over R. A decomposition $\rho=\{R_1, \ldots, R_k\}$ of R has lossless join wrt D iff, for every relation R satisfying D,

$$R = \pi_{R1}(R) \bowtie \pi_{R2}(R) \bowtie \ldots \bowtie \pi_{Rk}(R)$$

Denote by m_ρ the query

$$\pi_{R1}(R) \bowtie \pi_{R2}(R) \bowtie \ldots \bowtie \pi_{Rk}(R)$$

Testing Lossless Join

1. Compute tableau for m_ρ: <s, t>
2. Compute $CHASE_D$<s, t>
3. Minimized $CHASE_D$ <s, t> must be R

- Example: R = BAD ρ = {S, P, SP}
  - <s, t>: (matrix notation)

S#	SNAME	SCITY	P#	PNAME	PCITY	QTY
a_1	a_2	a_3	b_{14}	b_{15}	b_{16}	b_{17}
b_{21}	b_{22}	b_{23}	a_4	a_5	a_6	b_{27}
a_1	$\cancel{b_{32}}$ a_2	$\cancel{b_{33}}$ a_3	a_4	$\cancel{b_{35}}$ a_5	$\cancel{b_{36}}$ a_6	a_7
a_1	a_2	a_3	a_4	a_5	a_6	a_7

  - F = {S#→SCITY SNAME, P#→PNAME PCITY, P#S# →QTY}
  - $CHASE_F$<s, t> makes third row equal to <a_1, …, a_7> so the minimized tableau is equivalent to BAD (lossless join)

57

Technical Problem

- Checking if X→A is implied by a set F of fds (F |= X→A)
  - <u>Notation:</u> F^+ = {X→Y | F |= X→Y}

- Needed to compute keys and check satisfaction of BCNF and other normal forms.

- Solution: "closure" of a set of attributes wrt F: X^+
  - X^+ = {A | F |= X→A} (all attributes determined by X)

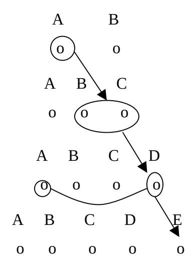
- So, X→A $\in$ F^+ iff A $\in$ X^+

58

Example of Closure Computation

- R = ABCDEF
- F = {A→C, BC→D AD→E}
- X = AB

- $X^{(0)}$ = AB
- $X^{(1)}$ = ABC
- $X^{(2)}$ = ABCD
- $X^{(3)}$ = ABCDE
- $X^{(4)} = X^{(3)}$
- X^+ = ABCDE
- To check if X is key in R: $X^+ = R$



Computing the closure of a set of attributes wrt a set of fd's

- Let F be a set of fd's over R, and $x \subseteq R$. The following computes the closure x^+ of x wrt FL

1. $X^{(0)} \leftarrow X$
2. $X^{(i+1)} \leftarrow X^{(i)} \cup \{Z \mid V \rightarrow Z \in F, V \subseteq X^{(i)}\}$

We get: $X^{(0)} \subseteq X^{(1)} \subseteq \ldots \subseteq X^{(i)} \subseteq X^{(i+1)} \subseteq \ldots \subseteq att(R)$

Since att(R) is finite, there must exist a $k \geq 0$ such that $X^{(k)} = X^{(k+1)}$ (and thus, $X^{(j)} = X^{(k)}\ \forall j \geq k$).

Then, $X^+ = X^{(k)}$

Dependency Preserving Decompositions

- Notation:
  - if F is a set of fd's over R and $X \subseteq R$ then

    $\pi_X(F) = \{V \rightarrow W \in F^+ \mid VW \subseteq X\}$.

- Definition:
  - Let $\rho = (R_1, \ldots, R_k)$ be a decomposition for R and F a set of fd's over R. Then ρ preserves F iff
  - $G = \cup^k_{i=1} \pi_{R_i}(F^+)$ is a cover for F (i.e., $G^+ = F^+$).

Testing Preservation of Dependencies

- Naïve method:
  1. Compute F^+
  2. Compute $G = \cup^k_{i=1} \pi_{R_i}(F^+)$
  3. Check that $F \subseteq G^+$

  Drawback: impractical, since the size of F^+ can be exponential in the size of F.

- Improved method:

```
for each X → Y in F do
   Z := X
   while changes occur in Z do
     for i := 1 to k do
     Z: = Z ∪ ((Z ∩ Ri)+ ∩ Ri)
   if Y ⊄ Z output "no" and stop
output "yes"
```

This computes the closure of Z wrt G

(+ is wrt F)

Example

$R = \{A\ B\ C\ D\}$ $\rho = \{AB, BC, CD\}$

$F = \{A \to B, B \to C, C \to D, D \to A\}$

Does ρ preserve F?

Clearly, ρ preserves $A \to B$, $B \to C$, $C \to D$.

Does ρ preserve $D \to A$?

	$\{D\}$	
	$\{CD\}$	$\{D\} \cup ((\{D\} \cap \{CD\})^+ \cap \{CD\})$
	$\{BCD\}$	$\{CD\} \cup ((\{CD\} \cap \{BC\})^+ \cap \{BC\})$
!	$\{ABCD\}$	$\{BCD\} \cup ((\{BCD\} \cap \{AB\})^+ \cap \{AB\})$
	$D \subseteq \{ABCD\}$	
	output yes.	

63

Normal Forms

- Terminology:
  - Let R be a relation schema and F a set of fd's over R.
  - Key: $X \subseteq att(R)$ such that $X \to att(R) \in F^+$.
  - Minimal key: $X \subseteq att(R)$ s.t. $X \to att(R) \in F^+$ and there is no $Y \subset\neq X$ such that $Y \to att(R) \in F^+$.
  - $A \in att(R)$ is prime: $A \in X$ where X is a minimal key
  - A is non-prime: A is not a member of any minimal key.

64

Purpose of normal forms

- Eliminate problems of redundancy and anomalies.
- Normal forms:
  - first, second, third, Boyce-Codd
- Boyce-Codd Normal Form (BCNF)
  - A relation scheme R is in BCNF wrt a set of fd's F over R, iff whenever $X \rightarrow A \in F^+$ and $A \not\in X$, X is a key for R
  - Only nontrivial dependencies are those induced by keys

65

Example

- BAD(S#, P#, SNAME, PNAME, SCITY, PCITY, QTY)
  - not in BCNF wrt

F = S#→SNAME SCITY
P#→PNAME PCITY
S# P#→QTY

  - S(S#, SCITY, SNAME) is in BCNF wrt S#→SNAME SCITY
  - P(P#, PCITY, PNAME) is in BCNF wrt P#→PNAME PCITY
  - SP(S# P# QTY) is in BCNF wrt S# P# → QTY

66

Decomposition of a relation schema into BCNF relation schemas, with lossless join

- Any relation scheme has a decomposition into BCNF relation schemes, with lossless join
  - not always dependency preserving
- Example

  R = CITY ZIP ST

  F = CITY ST $\rightarrow$ ZIP, ZIP $\rightarrow$ CITY

  - R has no decomposition into BCNF schemas, which preserves F
    - (CITY ST $\rightarrow$ ZIP is never preserved)

67

Algorithm to obtain a lossless join decomposition of a given R wrt F

Start with $\rho = \{R\}$

Apply recursively the following procedure:

for each S in ρ not in BCNF,

let $X \rightarrow A \in F^+$ be such that $XA \subseteq S$, $A \notin X$, X is not a minimal key of S

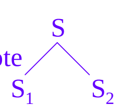
replace S by S_1 and S_2, where

$S_1 = XA$

$S_2 = S - A$

until all relation schemes are in BCNF

Note



lossless join:

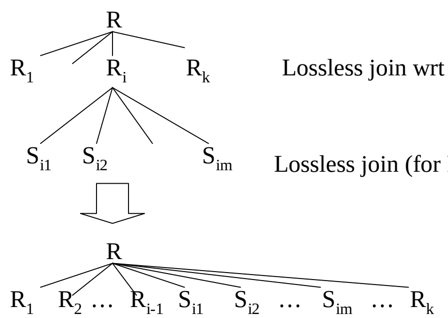
$X \rightarrow A \in F^+$

X is a key for S_1

68

Note

- a)



Lossless join wrt F

Lossless join (for R_i) wrt $\pi_{R_i}(F+)$

Lossless join wrt F

- b) If ρ is a lossless join decomposition of R wrt F and $\rho \subseteq \rho'$ then ρ' is a lossless join decomposition as well.

Example

R = C T H R S G

C = course
T = teacher
H = hour
R = room
S = student
G = grade

F: $C \rightarrow T$
$HR \rightarrow C$
$HT \rightarrow R$
$CS \rightarrow G$
$HS \rightarrow R$ only key: HS

Example

CTHRSG

Key = HS, C→T CS →G, HR →C, HS →R, TH →R

CSG

key = CS   CS →G

CTHRS

key = SH   C→T TH→R
HR →C HS →R

CT

key = C C→T

CHRS

key = SH CH →R HR →C HS →R

CHR

key = CH, CH →R
HR HR →C

CHS

key = SH SH →C

71

Remarks (drawbacks)

- Decomposition is not unique
  - CHRS could be decomposed into CHR and CHS <u>or</u> CHR and RHS
- The decomposition does not preserve TH→att(R)
  - local fds:

    G = {CS→G C→T CH →R HR →C SH →C}

    which does not imply TH → att(R)

72

- Is there an efficient algorithm for BCNF decomposition?
  - Most likely NO:

It is NP-complete to decide whether a relation scheme R is in BCNF wrt a set of fd's.

$\Downarrow$

Any algorithm for BCNF is most likely exponential

73

- Problem with BCNF:
  - Not every relation schema can be decomposed into BCNF relation schemas which preserve the dependencies and have lossless join.

$\Downarrow$

- Third Normal Form
  - A relation scheme R is in Third Normal Form wrt a set F of fd's over R, if whenever $X \rightarrow A$ holds in R and $A \notin X$ then either X is a minimal key or A is prime

Weaker than BCNF

74

Computing a 3NF decomposition

- Minimal covers
  - A set F of fd's is minimal iff

    1) $X \rightarrow Y \in F \Rightarrow Y$ is a single attribute
    2) for no $X \rightarrow A \in F$ is $F - \{X \rightarrow A\}$ equivalent to F
       (no fd can be removed from F)
    3) if $X \rightarrow A \in F$ then there is no proper subset Z of X such that $F - \{X \rightarrow A\} \cup \{Z \rightarrow A\}$ is equivalent to F
       (attributes cannot be removed from fd's in F)

  - Fact:. Every set of fd's is equivalent to some minimal set of fds (called a minimal cover).

75

Dependency preserving decompositions into Third Normal Form

R a schema

F minimal set of fd's over R

$\rho = \{XA_1 \ldots A_m \mid X \rightarrow A_i \in F\} \cup \{B \in R \mid B \text{ does not occur in } F\}$

- Example
  - R = CTHRSG (see earlier example)
  - F = $C \rightarrow T$ $CS \rightarrow G$
    $HR \rightarrow C$ $HS \rightarrow R$ (minimal)
    $HT \rightarrow R$
  - Then ρ = {CT, HRC, HTR, CSG, HRS}

76

- Theorem. ρ preserves F and each R_i in ρ is in 3NF wrt $\pi_{R_i}(F^+)$.

- To obtain decomposition that also has lossless join:
  - add to ρ a set K where K is a minimal key for R

- Theorem $\rho \cup \{K\}$ is a 3NF decomposition which is dependency preserving and has lossless join.

77

A (simplified) overview of schema design

1. Choose attributes in R
2. Specify dependencies F

   (tool: "Armstrong relations")
3. Find a lossless-join, dependency preserving decomposition into Normal Form schemas
   - BCNF, if possible
   - 3NF, if not BCNF

- Using the normalized, decomposed database
  - updates
  - queries

78

Updates

- Every time an update is performed, satisfaction of F by the new database must be checked: can be checked locally
- dangling tuples
  - if they are allowed, updates can be made locally
  - if they are not allowed, updates should be made using the original schema R as an interface

    universal relation interface

Queries

- The user may be allowed to ask queries against
  - the decomposition of R
  - R (universal relation interface)

    translate the query against R to a query against the decomposition
    - complicated, if dangling tuples are allowed (can't just take the join of the decompositions)

Multi-valued dependency (mvd)

- Example

loan-number	cust-name	street city

  – no fd (so BCNF)
  – redundancy still exists

loans and addresses are independent info for same customer

cust-name $\rightarrow\rightarrow$ loan-number

81

- Lossless join wrt

cust-name, loan-number
cust-name, street, city

cust-name	loan-number	street	city
c	l	-	-
c	-	s	d
c	l	s	d

symmetry: same as
cust-name $\rightarrow\rightarrow$ street city

82

More generally,

An mvd over R is an expression $V \rightarrow\rightarrow W$ where $VW \subseteq att(R)$

Meaning:

lossless join wrt VW, V(R–W)

Note: $V \rightarrow\rightarrow W$ if

$V \rightarrow\rightarrow (R-W)$

83

Implication: harder

- R = ABCD

  $A \rightarrow\rightarrow B$ and $B \rightarrow\rightarrow C$

  $\Downarrow$

  $A \rightarrow\rightarrow C$

"lossless join wrt AB, ACD

and

lossless join wrt BC, ABD

imply

lossless join wrt AC, ABD"

84

Need to prove:

A	B	C	D
a	b_1	c	d_1
a	b	c_1	d

$\Downarrow$

A	B	C	D
a	b	c	d

Suppose there is a pattern

A	B	C	D
a	b_1	c	d_1
a	b	c_1	d

Lossless join wrt AB, ACD implies

A	B	C	D	
a	b_1	c	d_1	
a	b	c_1	d	
a	b_1	c_1	d	new
a	b	c	d_1	

and lossless join wrt BC, ABD yields

A	B	C	D
a	b_1	c	d_1
a	b	c_1	d
a	b_1	c_1	d
a	b	c	d_1
a	b	c	d
.	.	.	.
.	.	.	.

Interaction of mvds and fds

- R = ABC

  $A \rightarrow B$ implies $A \rightarrow\rightarrow B$

A	B	C
a	b	c_1
a	b_1	c

$A \rightarrow B$ implies $b = b_1$

Same as chase:

$\pi_{AB}(R) \bowtie \pi_{AC}(R) \equiv_{A \rightarrow B} R$

87

- R = ABC

  $A \rightarrow\rightarrow B$, $C \rightarrow B$ imply

  $A \rightarrow B$

Proof by contradiction:

A	B	C
a	b_1	c_1
a	b_2	c_2

$b_1 \neq b_2$

+ $A \rightarrow\rightarrow B$

imply

A	B	C
a	b_1	c_1
a	b_2	c_2
a	b_1	c_2
a	b_2	c_1

Violation of $C \rightarrow B$

88

Use in query optimization

- Extension of chase

  idea: tableau T is equivalent to q(T)

  on all databases R satisfying a lossless join $q(R) \subseteq R$

Similar to earlier technique:

$A \rightarrow\rightarrow B$, $B \rightarrow\rightarrow C$ imply $A \rightarrow\rightarrow C$

Same as

$\pi_{AC}(R) \bowtie \pi_{ABD}(R) \equiv R$

for all R satisfying $A \rightarrow\rightarrow B$, $B \rightarrow\rightarrow C$

89

Minimization of rspj queries on databases satisfying fds and mvds:

- build tableau
- chase wrt fds and mvds
- minimize chased tableau
- construct query with minimum number of joins

90

Example

R = ABCD

q = $\pi_{ABD}(R) \bowtie \pi_{AC}(R)$,

database satisfies

$A \rightarrow B$, $B \rightarrow\rightarrow C$

- tableau for q:

A	B	C	D
a	b	c_1	d
a	b_1	c	d_1
a	b	c	d

91

- Chase wrt $A \rightarrow B$ yields

A	B	C	D
a	b	c_1	d
a	b	c	d_1
a	b	c	d

- Chase wrt $B \rightarrow\rightarrow C$ yields

A	B	C	D
a	b	c_1	d
a	b	c	d_1
a	b	c_1	d_1
a	b	c	d
a	b	c	d

92

- The minimized tableau is

A	B	C	D
a	b	c	d
a	b	c	d

- The minimized query is R

93

Extension of BCNF: 4NF

A relation schema R with set F of fds + mvds is in 4NF iff whenever $X \to\to Y$ holds (nontrivial), X is a key

- Note: 4NF $\Rightarrow$ BCNF
  converse is false

- Decomposition algorithm:
  same, replace fd by mvd.

94

Inclusion dependency (ind)

- Arises naturally in the context of class hierarchies

  "Every TA is a student"

  TA[name] $\subseteq$ student[name]

- General form

  P[X] $\subseteq$ Q[Y]

  Meaning

  $\pi_X(P) \subseteq \pi_Y(Q)$

95

- Inds + fds are intractable:

  implication problem is undecidable!

- Special cases:
  - acyclic incds + fds
  - unary incds + fds

  decidable

- Implication for incds alone:

  PSPACE — complete

96