CS 589 Principles of Databases, Unit 2, Part 5

Normal Form Review

Key
- FDs F on schema R, $K \subseteq R$
1. $K^+ = R$ $K \rightarrow R$
2. No proper subset of K has this property

Can be multiple keys

An attribute A is prime if it is in some key K

Superkey: Just Part 1. above

Example

Q D C N R P T (Quarter, Dept, Course #, Course Name, Room, Prof, Time)

{DC → N, DN → C, QDCT → RP, P → D, QRT → DC}

$QRT^+ = QRTDCNP$
- $QR^+ = QR$
- $QT^+ = QT$
- $RT^+ = RT$
- $Q^+ = Q$
- $R^+ = R$
- $T^+ = T$

CS 589 Principles of Databases, Unit 2, Part 5

More Examples

{DC → N, DN → C, QDCT → RP, P → D,
QRT → DC}

$QCPT^+$ = QCPTDNR key

CPT^+ CPTDN

QPT^+ QPTD

QCT^+ QCT

QCP^+ QCPDN

What about QRPT? — not a key — not minimal

$QRPT^+$ = QRPTCND

QRT^+ =

Different Example

FLight Date Pilot Gate

{LD → PG, L → G} LD^+ = LDPG L^+ = LG D^+ = D

LD is a key, so L, D are prime

P, G are not prime –

no key with P

no key with G

Third Normal Form (3NF)

Schema R with FDs F is in third normal form if for any nontrivial FD X → A, $A \notin X$, in F^+

X is a superkey of R or A is prime

Last example was not in 3NF

Consider L → G

Another Example

FLight Date PilotID PilotName

{LD → IN, I → N, N → I}

LD key

I, N not prime

I → N

N → I

} 3NF violations

Aside: Boyce-Codd Normal Form (BCNF)

Schema R with FDs F is in Boyce-Codd normal form if for any nontrivial FD $X \rightarrow A$, X is superkey

BCNF Example

Airport **C**ompany **B**illTo

{A C → B, B → C}

This schema is in 3NF

B → C, C → prime, part of key AC

This schema is not in BCNF

B → C

B not a superkey

CS 589 Principles of Databases, Unit 2, Part 5

Normalization through Decomposition

If R not in 3NF for FDs F, find an FD $Y \rightarrow Z$ in F^+ where Y is not a key; split into

R-Z , YZ

FLight Date PilotID PilotName

{LD → IN, I → N, N → I}

LDI , IN

Problems with Decomposition

1. Figuring out non-prime attributes can be hard NP-complete
2. Might end up with more relations than you need
3. Might not be able to enforce FDs

Decomposition Example

R = A B C D E

{AB → CDE, AC → BDE, B → C, C → B, C → D, B →E}

AB, AC are keys, so C → D is a problem

Decompose on C → D

R3 = ABC, R4 = BDE } 3NF

R1 = ABCE, R2 = CD

B → E is a problem, decompose again

R11 = ABC R12 = BE R2 = CD

But …

Enforcing FDs Example

R = A B C D E

{A → BCDE, CD → E, CE → B}

A is the only key, so CD → E is a problem

Decompose on CD → E

ABCD, CDE

How do you enforce CE → B?

Normalization through Synthesis

Start with schema R and FDs F

Generate database schema R1, R2, …, Rn such that

- Every relation schema is in 3NF
- Can enforce F
- Avoid extra relation schemas
- Have a lossless decomposition from R onto R1, R2, …, Rn.

Depends on a cover for F of a certain form

What's a Cover

A set of FDs G is a cover for a set of FDs F if $G^+ = F^+$ $G \equiv F$

G is a non-redundant cover if you cannot eliminate FDs and still be a cover

{A → B, B → C, ~~A → C~~}

Can test FDs one by one to see if any can be left out

$\{A \to B, B \to C\} \models A \to C$ yes, so omit

$\{A \to B\} \models B \to C$ no, so keep $B \to C$

Minimum Cover

A minimum cover is one with fewest possible FDs

Non-redundant, but not minimum:

{A → BC, B → A, AD → E, BD → I}

{A → BC, B → A, AD → EI}

Finding a Minimum Cover

Replace every FD X → Y in F with X → X+, then find a non-redundant cover

{A → BC, B → A, AD → E, BD → I}

{A → ABC, B → ABC, AD → ABCDEI
BD → ABCDEI}

non-redundant

{A → ABC, B → ABC, AD → ABCDEI}

Reduced Covers

Are there extraneous attributes?

Can appear on the right side or the left

{AB → C, A → B}
{A → BC, B → C}

$A \to B \models \{A \to AB, AB \to C\} \models A \to C$

How to remove?

For XA → Y, see if
$F \models X \to Y$ if so, replace $XA \to Y$ with $X \to Y$

For X → YB, see if
$F - \{X \to YB\} \cup \{X \to Y\} \models X \to YB$

Reduce Left Sides First

G = {A → C, AB → DE,
AB → CDI, AC → J}

Reduce left sides

G′ = $\{A \to C, AB \to DE, AB \to CDI, A \to J\}$

Reduce right sides

G″ = $\{A \to C, AB \to DE, AB \to I, A \to J\}$

Synthesis Algorithm

Start with FDs F
Find G = minimum cover for F
Find H = reduced cover for G
For each X → Y in H, add a schema XY

Synthesis Example

F = {A → BC, B → A,
AD → E, BD → I}

G = {A → ABC, B → ABC, AD → ABCDEI}

H = {A → BC, B → A, AD → EI}

ABC AB ADEI