CS 589 Principles of Databases, Unit 2, Part 5

Normal Form Review

Key

- FDs F on schema R, $K \subseteq R$
  1. $K^+ = R$
  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

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

QRT^+ = QRTDCNP

- QR^+ =
- QT^+ =
- RT^+ =
- Q^+ =
- R^+ =
- T^+ =

More Examples

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

QCPT⁺ =

CPT⁺ =

QPT⁺ =

QCT⁺ =

QCP⁺ =

What about QRPT?

Different Example

F**L**ight **D**ate **P**ilot **G**ate

{LD → PG, L → G}

LD is a key, so L, D are prime

P, G are not prime –

Third Normal Form (3NF)

Schema R with FDs F is in <u>third normal form</u> if for any nontrivial FD X → A,

Last example was not in 3NF

Consider L → G

Another Example

F**L**ight **D**ate Pilot**I**D Pilot**N**ame

{LD → IN, I → N, N → I}

Aside: Boyce-Codd Normal Form (BCNF)

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

BCNF Example

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

{A C → B, B → C}

This schema is in 3NF

This schema is not in BCNF

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

FL**L**ight **D**ate Pilot**I**D Pilot**N**ame

{LD → IN, I → N, N → I}

Problems with Decomposition

1. Figuring out non-prime attributes can be hard
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

B → E is a problem, decompose again

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

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 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

Minimum Cover

A minimum cover is one with fewest possible FDs

Non-redundant, but not minimum:

{A → BC, B → A, AD → E, BD → I}

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}

Reduced Covers

Are there extraneous attributes?

Can appear on the right side or the left

{AB → C, A → B}
{A → BC, B → C}

How to remove?

For XA → Y, see if

For X → YB, see if

Reduce Left Sides First

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

Reduce left sides

G′ =

Reduce right sides

G″ =

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}
```

Lossless Decomposition

If none of the FDs represents a key for all the attributes, need to add one more relation to get a lossless decomposition

R = ABCDE

F = {A → B, C → D}