# Relational Normalization Theory (supplemental material)

CSE 532, Theory of Database Systems

Stony Brook University

http://www.cs.stonybrook.edu/~cse532

# Quiz 8

- Consider a schema S with functional dependencies:
  - {A⟶BC, C⟶FG, E⟶HG, G⟶A}.

(a) Compute the attribute closure of A with respect to S.

- ABCFG

(b) Suppose we decompose S so that one of the subrelations, say called R, contains attributes AFE only. Find a projection of the functional dependencies in S on R (i.e., find the functional dependencies between attributes AFE).

- {A→F, E→AF} or {A→F, E→A, E→ F} or {A→F, E→A}

(c) Is R in BCNF?

- No, A is not a superkey.

Use one cycle of the BCNF decomposition algorithm to obtain two subrelations, and check BCNF cond.

- R1={AF,A→F} and R2={EA,E→A}. They are in BCNF.

# Quiz 8 (cont.)

- {A⟶BC, C⟶FG, E⟶HG, G⟶A}.
- Decomposition: R1={AF,A→F} and R2={EA,E→A}.

(d) Show that your decomposition is lossless.

- {AF} ∩ {EA} = {A}, which is a key of R1.

(e) Is your decomposition dependency preserving?

- Yes, because each of the FDs of R is entailed by FDs of R1 and R2.

(f) Is R in 3NF? Explain.

- No, F is not part of a key.

Use 3NF decomposition algorithm to obtain subrelations, and answer whether they are in 3NF.

- Obtain same R1={AF,A→F} and R2={EA,E→A}. They are in 3NF.

# Quiz 9

- Consider the following functional dependencies over the attribute set BCGHMVWY:

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

  - Find a minimal cover.
  - Decompose into lossless 3NF.
  - Check if all the resulting relations are in BCNF.
    - If not, decompose it into a lossless BCNF.

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

- **Minimal cover:**
  1. Split WY → BV into WY → B and WY → V.

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

- **Minimal cover:**
  1. Split WY → BV into WY → B and WY → V.
  2. Since W → V and W → B are implied by the given set of FDs, we can replace WY → B, WY → V, and WC → V in the original set with W → B (W → V already exists in the original set, so we discard the duplicate).

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

- **Minimal cover:**
  1. Split WY → BV into WY → B and WY → V.
  2. Since W → V and W → B are implied by the given set of FDs, we can replace WY → B, WY → V, and WC → V in the original set with W → B (W → V already exists in the original set, so we discard the duplicate).
  3. V → Y and BY → V can be derived from the original set so we can replace BV → Y and BYH → V with V → Y and BY → V.

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

- **Minimal cover:**
  1. Split WY → BV into WY → B and WY → V.
  2. Since W → V and W → B are implied by the given set of FDs, we can replace WY → B, WY → V, and WC → V in the original set with W → B (W → V already exists in the original set, so we discard the duplicate).
  3. V → Y and BY → V can be derived from the original set so we can replace BV → Y and BYH → V with V → Y and BY → V.
  4. We remove redundant FDs: W → B.

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | WY → BV | WC → V | V → B |
| BG → M | BV → Y | BYH → V | M → W |
| Y → H | CY → W | | |

- **Minimal cover:**

1. Split WY → BV into WY → B and WY → V.
2. Since W → V and W → B are implied by the given set of FDs, we can replace WY → B, WY → V, and WC → V in the original set with W → B (W → V already exists in the original set, so we discard the duplicate).
3. V → Y and BY → V can be derived from the original set so we can replace BV → Y and BYH → V with V → Y and BY → V.
4. We remove redundant FDs: W → B.
5. The final result is:

| | | | |
|---|---|---|---|
| W → V | V → B | BG → M | V → Y |
| BY → V | M → W | Y → H | CY → W |

# Quiz 9

| | | | |
|---|---|---|---|
| W → V | V → B | BG → M | V → Y |
| BY → V | M → W | Y → H | CY → W |

- **Lossless 3NF:**

  (WV; {W → V})

  (VBY; {BY → V; V → Y; V → B})

  (BGM; {BG → M})

  (MW; {M → W})

  (YH; {Y → H})

  (CYW; {CY → W})

- Since BGM is a superkey of the original schema, we don't need to add anything to this decomposition.

# Quiz 9

- **BCNF:**
  - The schema (BGM; {BG → M}) is not in BCNF because M→B is entailed by the original set of FDs.
    - Decompose BGM with respect to M → B into (BM; {M → B}) and (GM; {}), loosing the FD BG → M.
  - The schema (CYW; {CY → W}) is not in BCNF because W→Y is entailed by the original set of FDs.
    - Decompose CYW using W → Y into (WY; {W → Y}) and (WC; {}), loosing CY → W.

# Extra problem 1

- Prove that the following sets of FDs are equivalent:

F: A → B, C → A, AB → C

G: A → C, C → B, CB → A

F ≡ G ⇔ F entails G and G entails F

1. F entails G:

   $A^+_F = ABC$, so F entails A → C

   $C^+_F = ABC$, so F entails C → B

   $CB^+_F = ABC$, so F entails CB → A

2. G entails F:

   $A^+_G = ABC$, so G entails A → B

   $C^+_G = ABC$, so G entails C → A

   $AB^+_G = ABC$, so G entails AB → C

# Extra problem 2

- Consider the schema R over the attributes ABCDEFG with the following functional dependencies:

  $AB \rightarrow C$, $C \rightarrow B$, $BC \rightarrow DE$, $E \rightarrow FG$

and the following multivalued dependencies:

$$R = BC \bowtie ABDEFG$$

$$R = EF \bowtie FGABCD$$

  - Decompose this schema into 4NF while trying to preserve as many functional dependencies as possible: first use the 3NF synthesis algorithm, then the BCNF algorithm, and finally the 4NF algorithm.
    - Is the resulting schema after decomposition dependency-preserving?

# Extra problem 2

AB → C, C → B, BC → DE, E → FG

**1.** 3NF: split the right-hand sides of the FDs and minimize the left-hand sides:

AB → C, C → B, C → D, C → E, E → F, E → G

Synthesize the 3NF schemas: R1 = (ABC; {AB → C}),

R2 = (CBDE; {C → BDE}), R3 = (EFG; {E → FG})

**2.** BCNF:

R1 is not in BCNF, because C → B is implied by our original set of FDs (and C is not a superkey of R1): we decompose R1 further using C → B: R11 = (AC; {}) and R12 = (BC; {C → B}).

# Extra problem 2

R11 = (AC; {}), R12 = (BC; {C → B}),

R2 = (CBDE; {C → BDE}), R3 = (EFG; {E → FG})

R = BC ⋈ ABDEFG, R = EF ⋈ FGABCD

4NF: R = BC ⋈ ABDEFG projects onto R2 as R2 = BC⋈BDE and it violates 4NF because B = BC∩BDE is not a superkey.

Therefore, we decompose R2 into (BC; {C→B})and(BDE; {}).

MVD R = EF⋈FGABCD projects onto R3 as R3 = EF⋈FG and it violates 4NF because F = EF∩FG is not a superkey.

Therefore, we decompose R3 into (EF; {E→F}) and (FG; {}).

# Extra problem 2

AB → C, C → B, C → D, C → E, E → F, E → G

R = BC ⋈ ABDEFG, R = EF ⋈ FGABCD

R11 = (AC; {}), R12 = (BC; {C → B})

R21 = (BC; {C→B}), R22 = (BDE; {})

R31 = (EF; {E→F}), R32 = (FG; {})

The decomposition is not dependency preserving:

- A → B, which was present in the original schema, is now not derivable from the FDs attached to the schemas in the decomposition.

# Extra problem 3

- Consider the schema R over the attributes ABCDEFG:

  DE → F, BC → AD, FD → G, F → DE, D → E

  R = ABCFG ⋈ BCDE

- Compute 4NF:

1. 3NF: minimal cover:

***Split the LHS:***

DE → F, BC → A, BC → D, FD → G, F → D, F → E, D → E

***Reduce the RHS:***

Since D → E, we can delete E in the LHS of DE → F: D→ F

Since F → D, we can delete D in the LHS of FD → G: F→ G

***Remove redundant FDs:***

D → F, BC → A, BC → D, F → G, F → D, D → E

# Extra problem 3

D → F, BC → A, BC → D, F → G, F → D, D → E

R = ABCFG ⋈ BCDE

**3NF:** (DEF; {D → EF})

(BCAD; {BC → AD})

(FGD; {F → GD})

This decomposition is ***lossless*** since $BCAD^+$ = BCADEFG, i.e., the attribute set of the second schema is a superkey of the original schema.

**BCNF: yes**

**4NF:** We project the MVD on BCAD: BCAD = ABC ⋈ BCD.

ABC ∩ BCD = BC is a superkey of the schema (BC→AD), so, this MVD does not violate the 4NF.

# Extra problem 4

- Consider R=BCEGVWY with the MVDs:

  MVD1: R = WBCY $\bowtie$ YEVG

  MVD2: R =WBCE $\bowtie$ WBYVG

  MVD3: R =WBY $\bowtie$ CYEVG

**4NF 1:** Use MVD1 to obtain the following decomposition: WBCY, YEVG.

Use MVD2 to WBCY to yield WBC, WBY.

Use MVD3 cannot be used to decompose WBC because the join attribute, Y, is not in this attribute set.

MVD3 cannot be used to decompose WBY or YEVG because it projects as a trivial dependency in these cases: WBY=WBY $\bowtie$ Y and YEVG =Y $\bowtie$ YEVG.

**4NF 2:** the decomposition **4NF 1** is not unique: we can apply MVD3 and then MVD1 then will obtain: WBY, CY, YEVG.