CS34800 - PSO Solutions – Week 12

1. Find all candidate keys for R.

   R(A, B, C, D, E, F)
   F:
   DF -> C
   BC -> F
   E -> A
   ABC -> E

   Step 1. Attributes only on the left-hand side of functional dependencies: B, D
   Step 2. Attributes only on the right-hand side of functional dependencies: -
   Step 3. Attributes not appearing in the functional dependencies: -
   Step 1 + Step 3: {B, D} should be part of every candidate key.

   $\{BD\}^+$: BD

   1. Adding A: $\{ABD\}^+$: ABD
   2. Adding C: $\{BCD\}^+$ = {BCDF}
   3. Adding E: $\{BDE\}^+$ = {ABDE}

   1.1. Adding C: $\{ABCD\}^+$: {ABCDEF} → ABCD is a CK
   1.2. Adding F: $\{ABDF\}^+$: {ABCDEF} → ABDF is a CK
   2.1. Adding E: $\{BCDE\}^+$: {ABCDEF} → BCDE is a CK
   3.1. Adding F: $\{BDEF\}^+$: {ABCDEF} → BDEF is a CK

2. What is the highest normal form of R (1NF, 3NF or BCNF)?

   R(C, O, L, D, P, S)
   F:
   C -> D
   O -> L
   CO -> P
   P -> S

   We first find the set of all candidate keys for R:
   Attributes only on the left: CO (part of every CK)
   Attributes only on the right: DLS (not part of any CK)
   Attributes not in FDs: -
   $\{CO\}^+$: {CODLPS} → CO is the only CK

   C→ D violates 2NF, therefore the highest normal form of R is 1NF.

3. State the highest normal form (1NF, 3NF or BCNF) for each of the relations in the decomposition.

   R1(C, O, P, S); R2(C, O, L); R3(C, D)

   Dependencies that apply to R1: CO → P, P→S
   The only CK for R1 is CO and both dependencies pass 2NF, but P→S fails 3NF, therefore the highest normal form of R1 is 2NF.

   Dependencies that apply to R2: O → L
   The only CK for R2 is CO, and O→L fails 2NF, therefore the highest normal form of R2 is 1NF.

   Dependencies that apply to R3: C→D
   The only CK for R3 is C, and C→D satisfies BCNF as C is a superkey, therefore the highest normal form of R3 is BCNF.

4. Find a decomposition of R into 3NF relations that is lossless-join and dependency-preserving.

   R(A, B, C, D, E)
   F:
   A -> B
   A -> C
   C -> A
   BD -> E

   Step 1: F'={A→BC, C→A, BD→E}
   Step 2: R1(A, B, C); R2(A, C); R3(B, D, E)
   Step 3: We remove R2, as it is a subset of R1 -> R1(A, B, C); R3(B, D, E)
   Step 4: R has two candidate keys: AD and CD, none of them is contained in R1 or R3, so we form R4(A, D)

   The final decomposition is: R1(A, B, C); R3(B, D, E); R4(A, D)