**BCNF Decomposition** (refer Lecture 7: Pg. 7 – 9)

|  |  |
| --- | --- |
| 1. Identify keys   (minimum, maybe > 1) | Remove right-hand side, combination of remaining are keys.  Double-check if correct  C -> B Key: {AC} |
| 1. If stuck, use closure to identify **HIDDEN FDs** | E.g. {BC}+ = {BC~~DE~~A} BC -> A |
| 1. Identify all FD that violates BCNF   (choose the simplest) | Use closure: (*More but not all*)   * Right-hand side does not contain all attributes * Right > Left   {D}+ = {D} OK!  (AC)+ = {ACD} Violates BCNF |
| 1. Compute the closure of left-hand side |  |
| 1. Decompose into 2 tables | * Contain all attributes in the closure * Contain left + attributes NOT in closure |
| 1. Repeat until no violations | Condition of no violations:   * Only 2 attributes * Match with FDs * Match with keys identified in Step 1 |

BCNF VS 3NF

|  |  |
| --- | --- |
| **BCNF** | **3NF** |
| For every non-trivial X-> Y  Either X contains a key  Or attribute in Y does not appear in X  **Example:**  R(A, B)  Given FDs: A🡪B  Key: A  All FDs on R: A🡪B, AB🡪A, AB🡪B, AB🡪AB  AB🡪A, AB🡪B, AB🡪AB: trivial  A🡪B: The left hand side contains a key  Therefore, R is in BCNF  **Another Example:**  R(A, B, C)  Given FDs: A🡪B  Key: AC  All FDs on R: A🡪B, AB🡪B, AC🡪C, …  The left hand side of A🡪B does not contain a key  Therefore, R is NOT in BCNF | Either X contains a key  Or each attribute in Y is either contained in a Key or in X  **Example:**  Given FDs: C🡪B, AB🡪C, BC🡪C  Keys: {AB}, {AC}  AB🡪C is OK, since AB is a key of R  C🡪B is OK, since B is in a key of R  BC🡪C is OK, since C is in BC  So R is in 3NF  **Another Example:**  Given FDs: A🡪B, B🡪C  Keys: {A}  A🡪B is OK, since A is a key of R  B🡪C is not OK, since C is NOT in a key of R, and it is NOT in the left hand side  So R is NOT in 3NF |