LET'S START WITH DBMS :)

How to normalise table

In normalisation we generally break/decompose the table into 2 or more tables.

Steps to normalize a table

1.Write down all the attributes of table, CK, Prime and non-prime attributes and start analyzing with the FD.
2. For table to be in 1NF : Table should have atomic (indivisible) values and a primary key
3.For table to be in 2NF : No Partial dependency(LHS proper subset of CK and RHS non-prime attribute should be false)
4. For table to be in 3NF : No transitive dependency(LHS must be a CK or RHS a prime attribute should be true)
5. For table to be in BCNF : LHS must be a CK or SK
6. If it fails at any of these steps decompose the table on a common attribute which is CK (lossless)

LET'S START WITH DBMS :)

How to normalise table

1.Write down all the attributes of table, CK, PA,NPA and start analyzing with the FD.
2. For table to be in 1NF : Table should have atomic (indivisible) values and a primary key
3.For table to be in 2NF : No Partial dependency(LHS proper subset of CK and RHS non-prime attribute should be false)
4. For table to be in 3NF : No transitive dependency(LHS must be a CK or RHS a prime attribute should be true)
5. For table to be in BCNF : LHS must be a CK or SK
6. If it fails at any of these steps decompose the table on a common attribute which is CK (lossless)

R(A,B,C,D) and assume we have the following functional dependencies:
A→B, B→C, C->D

Step 1 : ABCD , CK-> A , Prime Attribute={A} , Non-Prime Attribute ={B,C,D}

Step 2: ABCDE, Since we are assuming our relation R is in a standard relational model, it is already in 1NF

Step 3 : Check for 2NF
A→B=(no pd as A is not a proper subset of CK and B is non prime(False and True=false))
B→C=(no pd as B is not a proper subset of CK and C is non prime (False and True=false))
C->D=(no pd as C is not a proper subset of CK and D is non prime (False and True=false))

LET'S START WITH DBMS :).

How to normalise table

1.Write down all the attributes of table, CK, PA,NPA and start analyzing with the FD.
2. For table to be in 1NF : Table should have atomic (indivisible) values and a primary key
3.For table to be in 2NF : No Partial dependency(LHS proper subset of CK and RHS non-prime attribute should be false)
4. For table to be in 3NF : No transitive dependency(LHS must be a CK or RHS a prime attribute should be true)
5. For table to be in BCNF : LHS must be a CK or SK
6. If it fails at any of these steps decompose the table on a common attribute which is CK (lossless)

ABCD , CK-> A , Prime Attribute={A} , Non-Prime Attribute ={B,C,D}

Step 4 : Check for 3NF
A→B=(no td as LHS is a CK)
B→C=(td is there as LHS is not CK and RHS non-prime)
C->D=(td is there as LHS is not CK and RHS non-prime)

So lets decompose the table
R1(A,B), R2(B,C), R3(C,D)

LET'S START WITH DBMS :)

How to normalise table

1.Write down all the attributes of table, CK, PA,NPA and start analyzing with the FD.
2. For table to be in 1NF : Table should have atomic (indivisible) values and a primary key
3.For table to be in 2NF : No Partial dependency(LHS proper subset of CK and RHS non-prime attribute should be false)
4. For table to be in 3NF : No transitive dependency(LHS must be a CK or RHS a prime attribute should be true)
5. For table to be in BCNF : LHS must be a CK or SK
6. If it fails at any of these steps decompose the table on a common attribute which is CK (lossless)

ABCD , CK-> A , Prime Attribute={A} , Non-Prime Attribute ={B,C,D}

Step 4 : Check for BCNF

R1(AB) A→B=(A is a candidate key OR a super key, so R1 is in BCNF)

R2(BC) B→C=(B is a candidate key OR a super key, so R2 is in BCNF)

R3(CD) C->D=(C is a candidate key OR a super key, so R3 is in BCNF)

Now, all decomposed relations R1, R2, and R3 are in BCNF