Normalization

1. Given the relation shown below
   a. List all functional dependencies satisfied by the relation shown below.
   b. Give all the candidate keys of this relation.

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c3
a2	b1	c4

Answer:
   The relation satisfies the following functional dependencies and the ones can be derived from them: A→ B, C→ B, C→A

   There is only one candidate key of this relation: {C}

2. Consider the following relation:
   CAR_SALE(Car#, Date_sold, Salesman#, Commission%, Discount_amt)
   Assume that a car may be sold by multiple salesmen, and hence {Car#, Salesman#} is the primary key. Additional dependencies are
       Date_sold→ Discount_amt
       Car# → Data_sold
       Salesman# → Commission%
   a) Based on the given primary key, is this relation in 1NF, 2NF, or 3NF? Why or why not?
   b) How would you successfully normalize it completely?

Answer:
   a) It is in 1NF since D1: Data# →Data_sold and D2: Salesman# → Commission% are both partial dependencies.
   b) To convert it to 2NF, we first decompose the relation to two relations to deal with D1. So we have
       R1=(Car#, Data_sold, Discount_amt)
       R2=(Car#,Salesman#,Comission%)
   R1 is in 2NF. R2 is not due to D2. So we decompose R2 to deal with D2. So we have:
       R1=(Car#, Data_sold, Discount_amt)
       R21=(Salesman#, Comission%)
       R22==(Car#,Salesman#)
   Now R1, R21, R22 are all in 2NF.

R21 and R22 are both in 3NF and BCNF. R1 is not in 3NF since D3: Data_sold→ Discount_amt is a transitive dependency. So to convert it to 3NF, we decompose R1:

        R11=(Data_sold, Discount_amt)
        R12=(Car#, Data_sold)
        R21=(Salesman#, Comission%)
        R22==(Car#,Salesman#)

3. Consider the relation R (A,B,C,D,E, F, G, H, I, J) with the following FDs:
        AB→ C, BD→ EF, AD→ GH, A→I, H→ J
a) What is the candidate key of R?
b) Explain why the relation is not in 2NF.
c) Normalize it to 2NF, 3NF, and BCNF.

Answer:
a) The candidate key of R is {ABD}
b) It is not in 2NF since D1: AB→C, D2: BD→EF, D3: AD→GH, D4: A→I are all partial dependencies.
c) To convert to 2NF, we first decompose the relation to deal with D1. So we have:
        R1=(A,B,C,I)
        R2=(A,B,D,E,F,G,H,J)
Both R1 and R2 have partial dependencies, including D4 in R1, D2 and D3 in R2. We decompose R1 to deal with D4. So we have:
        R11={A, I}
        R12=(A,B,C)
        R2=(A,B,D,E,F,G,H,J)
Now R11 and R12 are in 2NF. R2 is not due to D2 and D3. We decompose R2 to deal with D2 first. So we have:
        R11={A, I}
        R12=(A,B,C)
        R21=(B,D,E,F)
        R22=(A,B,D,G,H,J)
Now R11, R12, R21 are in 2NF. R22 is not due to D3. We decompose R22 to deal with D3. So we have:
        R11={A, I}
        R12=(A,B,C)
        R21=(B,D,E,F)
        R221=(A,D,G,H,J)
        R222=(A,B,D)
Now all the relations are in 2NF.
R11, R12, R21, R222 are all in 3NF and BCNF. R221 is not in 3NF since D5: H→J is a transitive dependency. We decompose R221 to deal with D5. So we have:
        R11={A, I}
        R12=(A,B,C)
        R21=(B,D,E,F)
        R2211=(H, J)

R2212=(<u>A,D</u>,G,H)
R222=(<u>A,B</u>,D)
Now all the relations are in 3NF and BCNF.

4. Consider the following relation for published books:
   BOOK (Book_title, Author_name, Book_type, List_price, Author_affil, Publisher).
   Suppose the following dependencies exist:
   Book_title→ Publisher, Book_type
   Book_type → List_price
   Author_name → Author_affil

   a) What normal form is the relation in? Explain your answer.
   b) Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

Answer:
a) The key of the relation is (Book_title, Author_name). Therefore, the relation is in 1NF but not in 2NF since D1: Book_title→ Publisher, Book_type and D2: Author_name → Author_affil are partial dependencies.
b) To convert the relation to 2NF, we first decompose the relation to deal with D1. So we have:
Book1=(<u>Book_title</u>, Publisher, Book_type, List_price)
Book2=(<u>Book_title, Author_name</u>, Author_affil)
Now Book1 is in 2NF but Book2 is not due to D2. We decompose the Book2 to deal with D2 So we have:
Book1=(<u>Book_title</u>, Publisher, Book_type, List_price)
Book21=(<u>Author_name</u>, Author_affil)
Book22=(<u>Book_title, Author_name</u>)
Now all the relations are in 2NF.

Book21 and Book22 are in 3NF and BCNF. Book1 is not in 3NF since D3: Book_type → List_price is a transitive dependency. We decompose R1 to deal with D3. So we have:
Book11=(<u>Book_type</u>, List_price)
Book12=(<u>Book_title</u>, Publisher, Book_type)
Book21=(<u>Author_name</u>, Author_affil)
Book22=(<u>Book_title, Author_name</u>)
Now all the relations are in 3NF and BCNF.