LET'S START WITH DBMS :)

Lossy and Lossless decomposition

Lossy Decomposition

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

Consider there is a relation R,
Lossy decomposition occurs when a relation R is decomposed or broken into two or more relations, but data is lost, and the original relation R can't be reconstructed by joining these decomposed relations.

There are 2 rules for a decomposition to be lossy

1. Some data from the original relation R is lost after decomposition
2. Join of the decomposed relations(R1, R2..Rn) is not equal to the original relation R

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

A	B	C
1	2	3
4	2	6

R

Lossy decomposition

There is a relation R(A,B,C)

Step 1: Lets decompose the relation based on any attribute and keep that attribute as common, for now lets use B as common attribute
Decomposed relations: R1(A,B) and R2(B,C)

A	B
1	2
4	2

R1

B	C
2	3
2	6

R2

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

A	B	C
1	2	3
4	5	6

R

Lossy decomposition

Step 2: Lets perform a natural join between R1 and R2
When we do R1 natural join R2 we won't get the original relation back(lossy)

A	B	C
1	2	3
1	2	6
4	2	3
4	2	6

We can see some additional tuples that were not in the original relation R (lossy decomposition)

R1 natural join R2

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

Lossy decomposition

How to ensure a decomposition is lossless

1. Divide or decompose the table on basis of CK or SK present in the relation so that there is no duplicacy
2. For a decomposition to be lossless
    a. R1 U R2= R
    b. R1$\cap$R2 = common attribute
3. To ensure that a decomposition is lossless, a common approach is to use the dependency preservation property

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

Lossless Decomposition

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

Consider there is a relation R,
Lossless decomposition ensures that when a relation R is decomposed/breaked into two or more relations, no data is lost, and the original relation R can be again reconstructed by joining these decomposed relations.

There are 2 rules for a decomposition to be lossless

1. All data in the original relation R should be preserved after decomposition
2. Join of the decomposed relations(R1, R2..Rn)= original relation R

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

Lossless Decomposition

So if the table is decomposed and we want to query the attributes present in both the tables we will use the join operation.

Natural Join :

- The natural join operation combines tuples(rows) from two relations based on common attributes.
- It only includes those combinations of tuples that have the same values for the common attributes.

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

A	B	C
1	2	3
4	5	6

R

Lossless decomposition

There is a relation R(A,B,C) with CK as A

Step 1: Lets decompose the relation based on the CK or SK of the given R
Decomposed relations: R1(A,B) and R2(A,C)

A	B
1	2
4	5

R1

A	C
1	3
4	6

R2

LET'S START WITH DBMS :)

Lossy and Lossless decomposition

A	B	C
1	2	3
4	5	6

R

Lossless decomposition

There is a relation R(A,B,C) with CK as A

Step 2: Lets perform a natural join between R1 and R2
When we do R1 natural join R2 we get the original relation back(lossless)

A	B	C
1	2	3
4	5	6

R1 natural join R2