- Decomposition of a relation in relational model is done to convert it into appropriate normal form
- A relation R is decomposed into two or more only if the decomposition is both lossless join and dependency preserving.

1. **Lossless join decomposition**

There are two possibilities when a relation R is decomposed into R1 and R2.They are

- Lossy decomposition i.e., $R1 \bowtie R2 \supset R$
- Lossless decomposition i.e., $R1 \bowtie R2 = R$

For a decomposition to be lossless, it should hold the following conditions

- Union of attributes of R1 and R2 must be equal to attribute R. each attribute of R must be either in R1 or in R2 i.e., $Att(R1) \cup Att(R2) = Att(R)$
- Intersection of attributes of R1 and R2 must not be null i.e., $Att(R1) \cap Att(R2) \neq \emptyset$
- Common attribute must be a key for atleast one relation(R1 or R2) i.e., $Att(R1) \cap Att(R2)$ -> Att(R1) or $Att(R1) \cap Att(R2)$->Att(R2)

Example
A relation R(A,B,C,D) with FD set {A->BC} is decomposed into R1(ABC) and R2(AD). This is lossless join decomposition because

- First rule holds true as $Att(R1) \cup Att(R2)=(ABC) \cup (AD)= (ABCD) = Att(R)$
- Second rule holds true as $Att(R1) \cap Att(R2) = (ABC) \cap (AD) \neq \emptyset$
- Third rule holds true as $Att(R1) \cap Att(R2) = A$ is a key of R1(ABC) because A->BC is given

2. **Dependency Preserving Decomposition**

- If we decompose a relation R into relations R1 and R2, all dependencies of R must be part of either R1 or R2 or must be derivable from combination of functional dependencies(FD) of R1 and R2
- Suppose a relation R(A,B,C,D) with FD set {A->BC} is decomposed into R1(ABC) and R2(AD) which is dependency preserving because FD A->BC is a part of R1(ABC)

Example
Consider a schema R(A,B,C,D) and functional dependencies A->B and C->D which is decomposed into R1(AB) and R2(CD)

This decomposition is dependency preserving decompostion because

- A->B can be ensured in R1(AB)
- C->D can be ensured in R2(CD)