

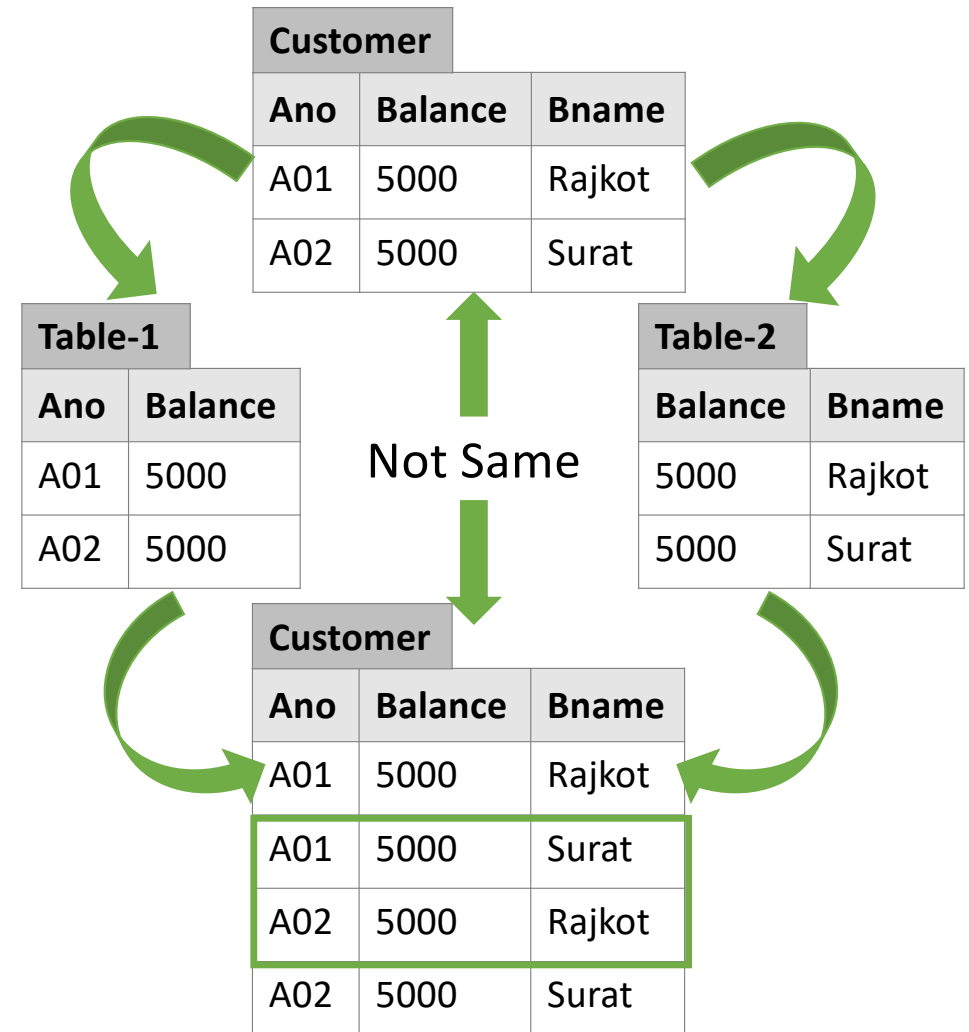
# Decomposition

# What is decomposition?

- Decomposition is the **process of breaking down given relation into two or more relations**.
- Relation R is replaced by two or more relations in such a way that:
  - Each new relation contains a **subset** of the **attributes of R**
  - Together, they all **include all tuples** and **attributes of R**
- Types of decomposition
  - Lossy decomposition
  - Lossless decomposition (non-loss decomposition)
- Properties of Decomposition:
  - Lossless (Mandatory)
  - Dependency Preserving (Optional)

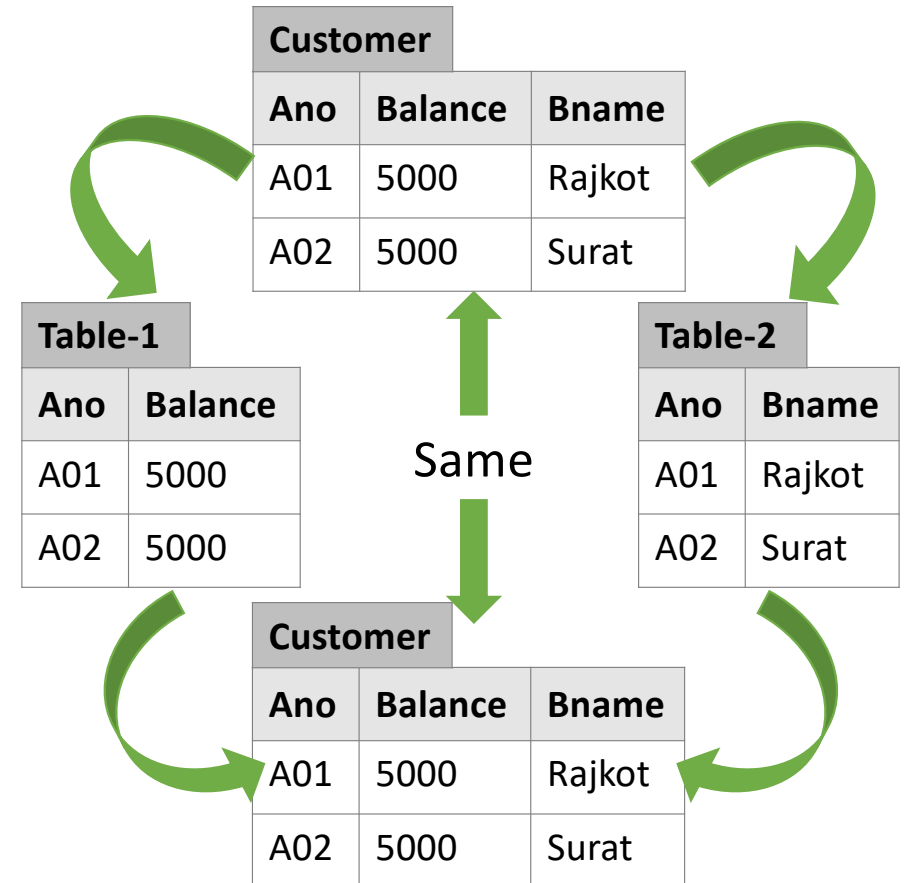
# Lossy decomposition

- The decomposition of relation R into R1 and R2 is lossy when the natural join of R1 and R2 does not yield the same relation as in R.
- This is also referred as lossy-join decomposition.
- The disadvantage of such kind of decomposition is that some information is lost during retrieval of original relation.
- From practical point of view, decomposition should not be lossy decomposition.



# Lossless decomposition

- The decomposition of relation R into R1 and R2 is lossless when the natural join of R1 and R2 produces the same relation as in R.
- This is also referred as a non-additive (non-loss) decomposition.
- All decompositions must be lossless.



# Determine Whether Decomposition Is Lossless Or Lossy?

Consider a relation R is decomposed into two sub relations  $R_1$  and  $R_2$ .

Then,

- If all the following conditions satisfy, then the decomposition is lossless.
- If any of these conditions fail, then the decomposition is lossy.

**Condition-01:** Union of both the sub relations must contain all the attributes that are present in the original relation R.

$$R_1 \cup R_2 = R$$

**Condition-02:**

- Intersection of both the sub relations must not be null.
- In other words, there must be some common attribute which is present in both the sub relations.

$$R_1 \cap R_2 \neq \emptyset$$

**Condition-03:** Intersection of both the sub relations must be a candidate key of either  $R_1$  or  $R_2$  or both.

$$R_1 \cap R_2 = \text{candidate key of } R_1 \text{ or } R_2$$

# Determine Whether Decomposition Is Dependency Preserving?

Consider a relation  $R$  having a set of FDs is decomposed into two sub-relations  $R_1$  and  $R_2$ .

Then,

- Find out all the valid non-trivial FDs for each sub relations.
  - If all the FDs of  $R$  are proved to be the member of non-trivial FD set of sub-relations, it is said to be dependency preserving decomposition.
- 
- First write down all the possible non-trivial FDs for all sub-relations.
  - Now, check which are valid and invalid FDs out of them.
    - Determine the closure of each non-trivial FD with the help of FD set of  $R$  and if it satisfies the non-trivial FD, it is said to be valid non-trivial FD.

# Exercise

R(ABCD)

✓  
FD = {A → B, B → C, C → D, D → A}

Decomposition R1(AB), R2(BC), and R3(CD)

- Is Lossless?

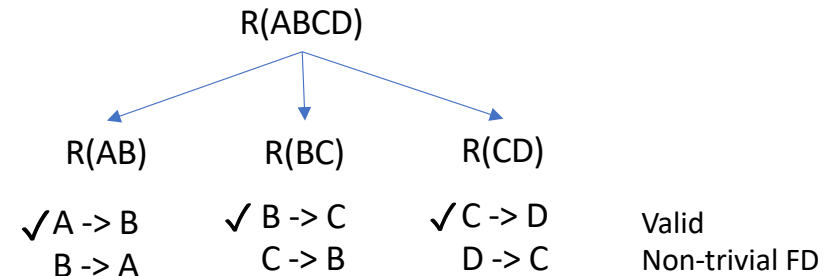
- Is Dependency Preserving?

For Lossless Decomposition:

- Take the closure of common attribute between R1 and R2, i.e., B.  $B^+ = BCDA$ .
- It is the CK of both R1 and R2 because it is determining all the attributes of both the table.
- Take the closure of common attribute between R2(BC) and R3(CD), i.e., C.  $C^+ = CDAB$ .
- It is the CK of both R2 and R3 because it is determining all the attributes of both the table.
- It means it is lossless decomposition.

For Dependency Preserving Decomposition:

- Find out all the valid non-trivial FDs of all the decomposed tables.
- If we can derive all the FD of R with the help of F1 and F2, then decomposition will be called dependency preserving.



- A → B, B → C, and C → D are directly exist in the set of valid non-trivial FDs.
- The membership of D → A will be checked by taking the closure of D with the help of valid non-trivial FD set.  
 $D^+ = DCBA$ , hence D → A exists.  
✓

# Exercise

R(ABC)

FD = {A  $\rightarrow$  B, B  $\rightarrow$  C, C  $\rightarrow$  A}

Decomposition R1(AB) and R2(BC)

- Is Lossless?

- Is Dependency Preserving?



# Exercise

R(ABCD)

FD = {AB  $\rightarrow$  CD, D  $\rightarrow$  A}

Decomposition R1(AD) and R2(BCD)

- Is Lossless?

- Is Dependency Preserving?

# Exercise

R(ABCDEFG)

FD = {AB  $\rightarrow$  C, AC  $\rightarrow$  B, AD  $\rightarrow$  E, B  $\rightarrow$  D, BC  $\rightarrow$  A, E  $\rightarrow$  G}

Decomposition R1(ABC), R2(ABDE) and R3(EG)

- Is Lossless?

- Is Dependency Preserving?

# Exercise

R(ABCDE)

FD = {A  $\rightarrow$  BC, C  $\rightarrow$  DE, D  $\rightarrow$  E}

Decomposition R1(ABCD) and R2(DE)

- Is Lossless?

- Is Dependency Preserving?