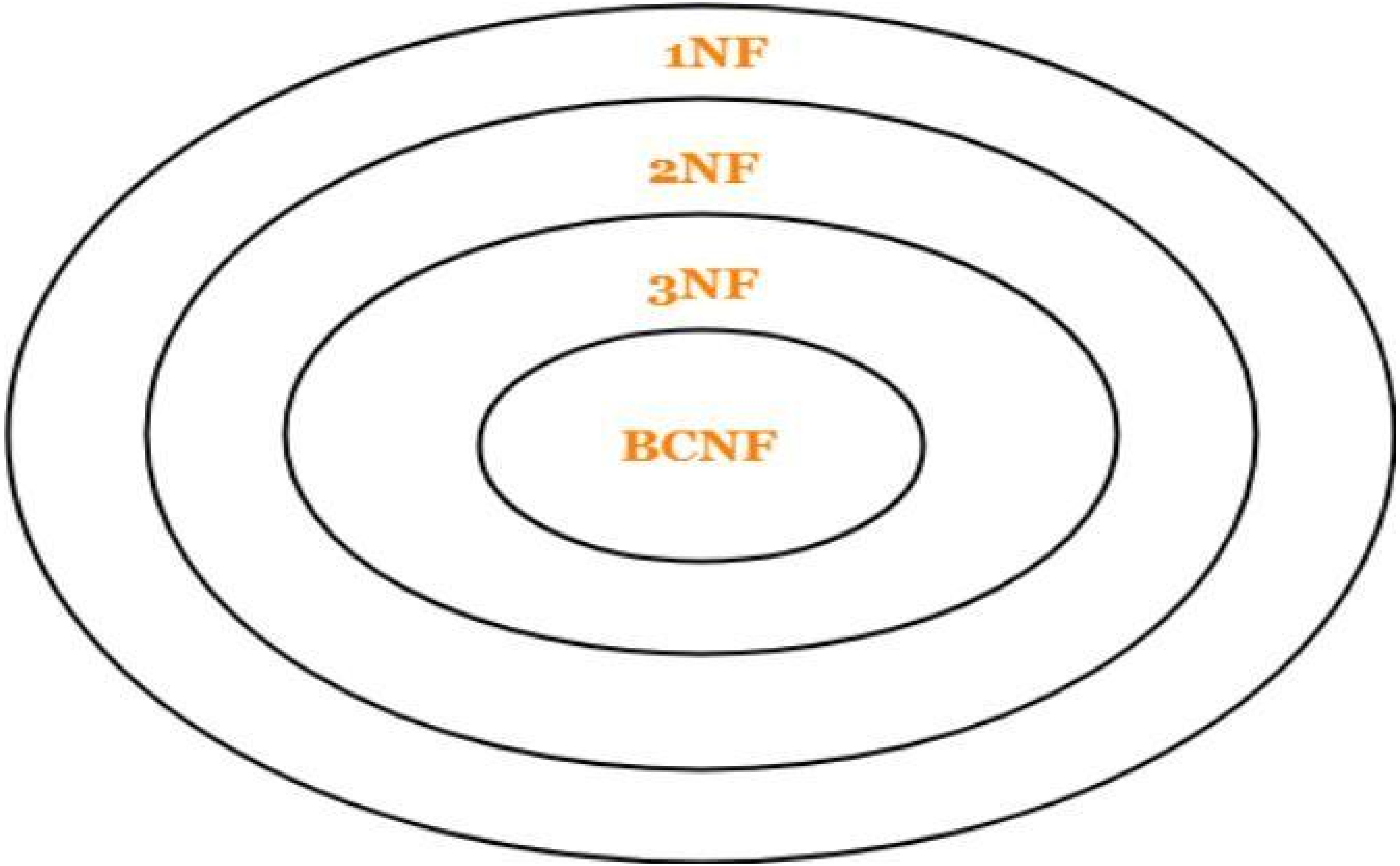
Important Points of Normalization


1NF
2NF
3NF
BCNF

- A relation in BCNF will surely be in all other normal forms.
- A relation in 3NF will surely be in 2NF and 1NF.
- A relation in 2NF will surely be in 1NF.

- BCNF is stricter than 3NF.
- 3NF is stricter than 2NF.
- 2NF is stricter than 1NF.

- In a relational database, a relation is always in First Normal Form (1NF) at least.

- If all the attributes of a relation are prime attributes, then it will always be in 2NF at least.
- This is because there will be no chances of existing any partial dependency.
- Since there are no non-prime attributes, there will be no Functional Dependency which determines a non-prime attribute.

While determining the normal form of any given relation:

- Start checking from BCNF.
- This is because if it is found to be in BCNF, then it will surely be in all other normal forms.
- If relation is not in BCNF, start moving towards the outer circles and check for other normal forms in order they appear.

- If all the attributes of a relation are prime attributes, then it will always be in 3NF at least.
- This is because there will be no chances of existing any transitive dependency for non-prime attributes.

- Third Normal Form (3NF) is considered adequate for normal relational database design.

- Every binary relation (a relation with only two attributes) is always in BCNF.

- BCNF is free from redundancies arising out of functional dependencies (zero redundancy).

- A relation with only trivial functional dependencies is always in BCNF.
- In other words, a relation with no non-trivial functional dependencies is always in BCNF.

- BCNF decomposition is always lossless but not always dependency preserving.

- Sometimes, going for BCNF may not preserve functional dependencies.
- So, go for BCNF only if the lost functional dependencies are not required else normalize till 3NF only.

- There exist many more normal forms even after BCNF like 4NF and more.
- But in the real world database systems, it is generally not required to go beyond BCNF.

- Unlike BCNF, Lossless and dependency preserving decomposition into 3NF and 2NF is always possible.

- If a relation consists of only singleton candidate keys and it is in 3NF, then it must also be in BCNF.

- If a relation consists of only one candidate key and it is in 3NF, then the relation must also be in BCNF.

How to find the highest Normal form of a Relation?

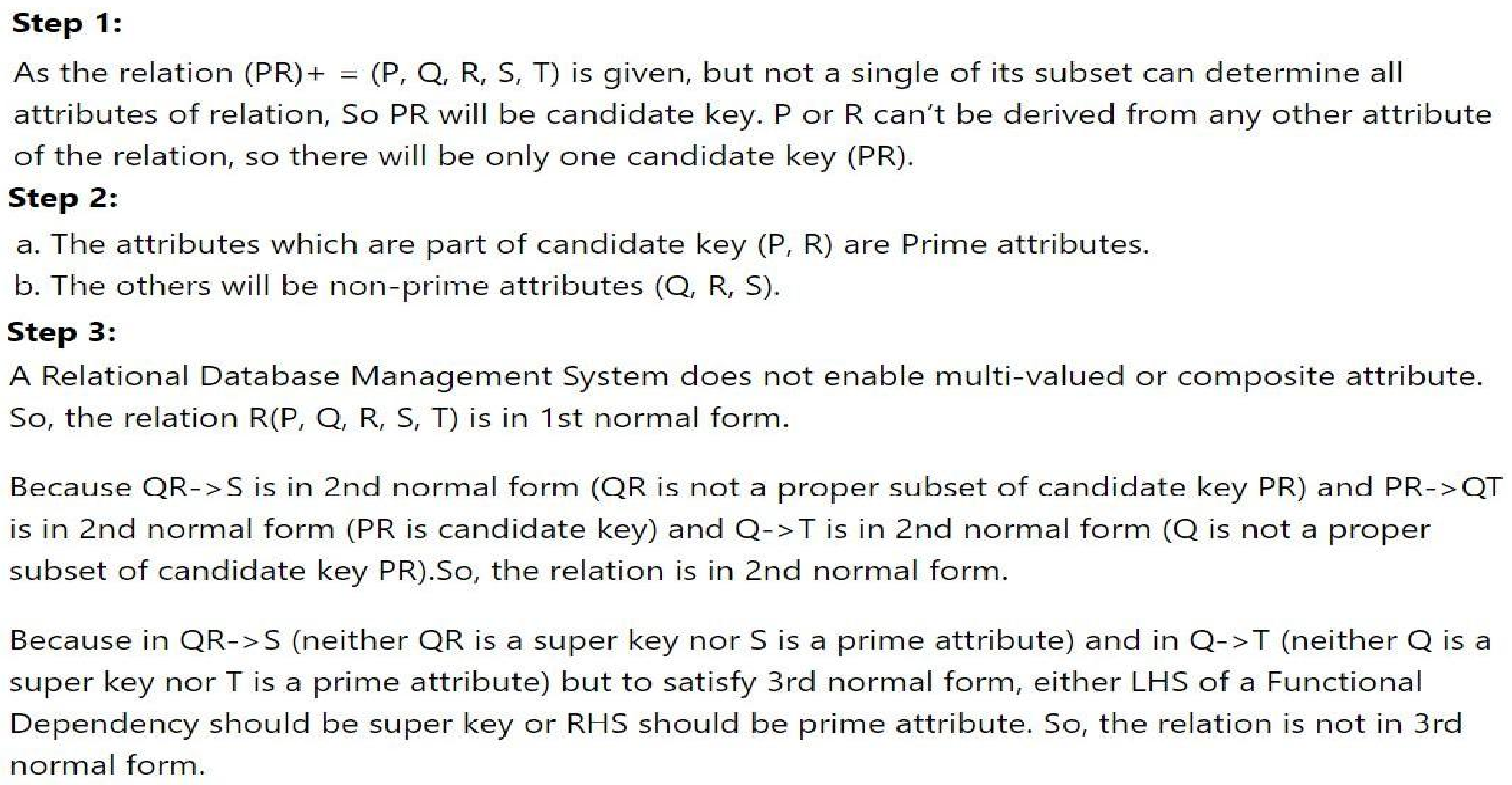
Steps to follow to find the highest normal form of a relation

- The first step is to find all feasible candidate keys of the relation and its attributes.
- The second step is to organize into two categories all the attributes of the relation:

  Prime attributes

  Non-prime attributes
- Third step is to examine for 1st normal form and then 2nd and so on. If the process is unsuccessful in satisfying nth normal form condition, then the highest normal form will be n-1.

Problem 1) Find the highest normal form of a relation R(P, Q, R, S, T) with Functional dependency set as (QR->S, PR->QT, Q->T).



Step 1:

As the relation (PR)+ = (P, Q, R, S, T) is given, but not a single of its subset can determine all attributes of relation, So PR will be candidate key. P or R can’t be derived from any other attribute of the relation, so there will be only one candidate key (PR).

Step 2:

a. The attributes which are part of candidate key (P, R) are Prime attributes.
b. The others will be non-prime attributes (Q, R, S).

Step 3:

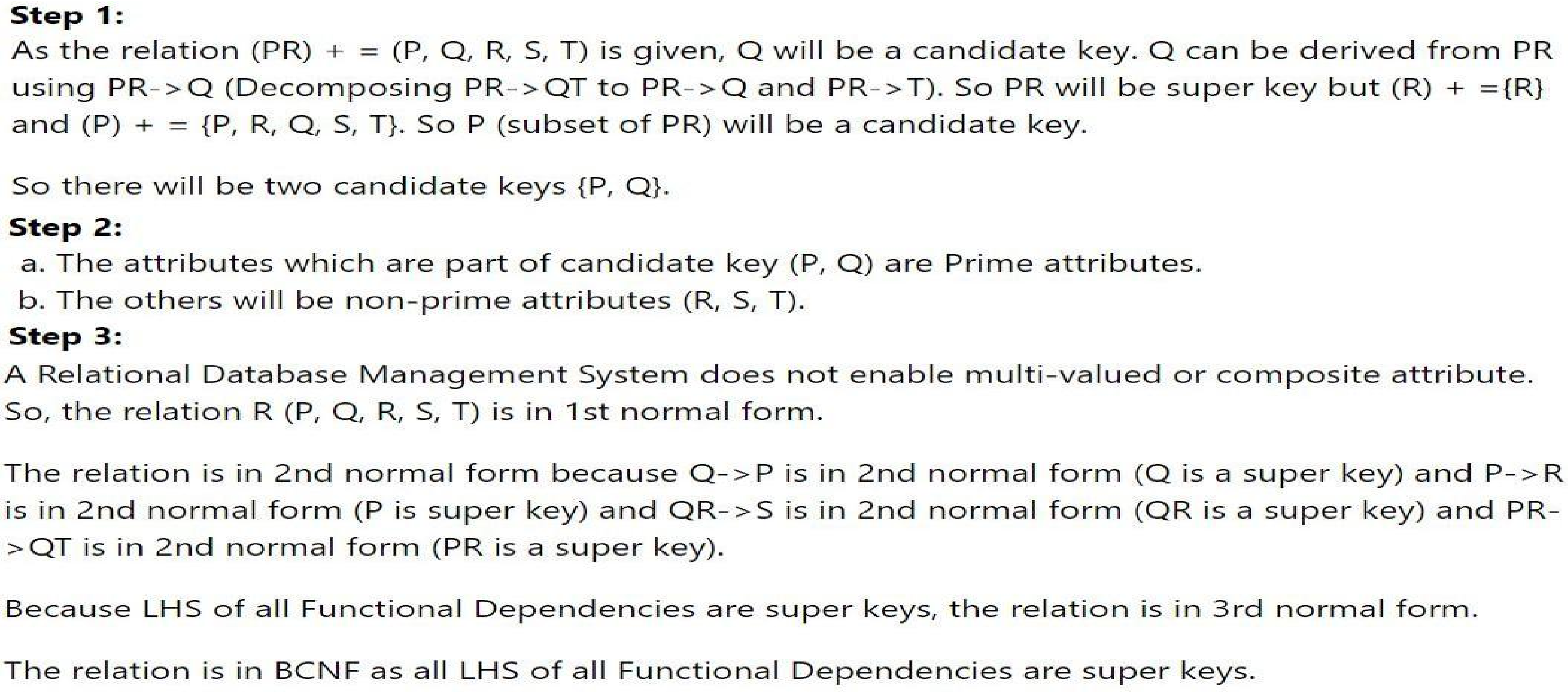
A Relational Database Management System does not enable multi-valued or composite attribute. So, the relation R(P, Q, R, S, T) is in 1st normal form.

Because QR->S is in 2nd normal form (QR is not a proper subset of candidate key PR) and PR->QT is in 2nd normal form (PR is candidate key) and Q->T is in 2nd normal form (Q is not a proper subset of candidate key PR).So, the relation is in 2nd normal form.

Because in QR->S (neither QR is a super key nor S is a prime attribute) and in Q->T (neither Q is a super key nor T is a prime attribute) but to satisfy 3rd normal form, either LHS of a Functional Dependency should be super key or RHS should be prime attribute. So, the relation is not in 3rd normal form.

So, the highest normal form of relation will be 2nd Normal form.

Problem 2) Find the highest normal form of a relation R (P, Q, R, S, T) with Functional Dependency set (Q->P, P->R, QR->S, PR->QT).



Step 1:

As the relation (PR) + = (P, Q, R, S, T) is given, Q will be a candidate key. Q can be derived from PR using PR->Q (Decomposing PR->QT to PR->Q and PR->T). So PR will be super key but (R) + ={R} and (P) + = {P, R, Q, S, T}. So P (subset of PR) will be a candidate key.

So there will be two candidate keys {P, Q}.

Step 2:

a. The attributes which are part of candidate key (P, Q) are Prime attributes.
b. The others will be non-prime attributes (R, S, T).

Step 3:

A Relational Database Management System does not enable multi-valued or composite attribute. So, the relation R (P, Q, R, S, T) is in 1st normal form.

The relation is in 2nd normal form because Q->P is in 2nd normal form (Q is a super key) and P->R is in 2nd normal form (P is super key) and QR->S is in 2nd normal form (QR is a super key) and PR->QT is in 2nd normal form (PR is a super key).

Because LHS of all Functional Dependencies are super keys, the relation is in 3rd normal form.

The relation is in BCNF as all LHS of all Functional Dependencies are super keys.

So, the highest normal form is BCNF.

Problem 3) Find the highest normal form of a relation R (P, Q, R, S, T) with Functional Dependency set (P->S, Q->P, QR->S, PR->QT).

Step 1:

As the relation (PR) + = (P, Q, R, S, T) is given, but not a single of its subset can determine all attributes of relation, so PR will be candidate key. P can be derived from Q, so we can replace P in PR by Q. So QR will also be a candidate key.

So the two candidate keys will be (PR, QR).

Step 2:

a. The attributes which are part of candidate key (P, Q, R) are Prime attributes.

b. The others will be non-prime attributes (S, T).

Step 3:

A Relational Database Management System does not enable multi-valued or composite attribute. So, the relation R (P, Q, R, S, T) is in 1st normal form.

Because P->S is partial dependency (P which is a subset of candidate key PR is determining non-prime attribute S), the relation is not in 2nd Normal form because the 2nd normal form does not enable partial dependency.

So, the highest normal form will be 1st Normal Form.

The End