depends on a key attribute or on a composite key. Non-key attributes that do not meet this criteria are split into simpler entities.

To apply 2NF, the table must be 1NF. So if we consider the Salesperson table which is in 1NF and apply 2NF, we find that each non key attributes (Employe name, Storebranch, Department) depends on primary key employeenumber, so the table is already in 2NF.

In the Sales file, the non key attribute salesprice depends on composite key (employeenumber & itemnumber). And the non key attribute itemdescription depends on itemnumber and is not related in any way to employeenumber, so when we apply 2NF to Sales file, it splits into tables

i) Salespersonitem file

Employeenumber*
itemnumber*
Salesprice

ii) Item file

Itemnumber*
Itemdescription

Therefore after applying 2NF, the resultant relations that we have are

a) Salesperson
   i) Employeenumber*
   ii) Employeename
   iii) StoreBranch
   iv) Department

b) Salespersonitem file
   i) Employeenumber*
   ii) itemnumber*
   iii) Salesprice

c) Item file
   i) Itemnumber*
   ii) Item description.

## Second Normalization

Third Normal Form (TNF) — Third normal form is based on transitive dependency. A relation is said to be in TNF if it is in 3NF and no non-prime attribute is functionally dependent on other non-prime attribute.

When we consider the salesperson file which is already in 2NF, we find that it is not in 3NF. This is because the nonkey attribute employername is dependent on prime attribute employernumber. Similarly another nonkey attribute storebranch is also dependent on prime attribute employernumber. But Department which is a non key attribute depends on another nonkey attribute which is storebranch. Therefore we apply 3NF to salesperson relation and get the following relations:

i) Employee file
Employernumber*
employername
Storebranch

ii) Branch file
*StoreBranch
Department.

We find Salespersonitem file and Itemfile are already in 3NF. So finally after applying 3NF, the 4 relations that we have are

i) Employeefile
Employernumber*
Employername
StoreBranch

ii) Branchfile
*StoreBranch
Department

iii) Salesperson itemfile
Employernumber*
itemnumber*
Salesprice

iv) Item file
Itemnumber*
Item description

Third Normalization

Boyce Codd Normal Form (BCNF) – BCNF was proposed as a simple form of 3NF but it is much more strict than 3NF, meaning that every relation in BCNF is also in 3NF. But a relation in 3NF not necessarily be in BCNF.

A relation which is in 3NF, may have no. of overlapping composite candidate keys. Let us assume that a relation has more than one composite keys. Further assume that composite keys have a common attribute. If an attribute of a composite key is dependant on an attribute of another composite key, then a normalization called BCNF is required.

We consider the relation Professor (Prof-code, dept, HOD, percent-time). It is assumed that

1) A professor can work in more than one department.
2) Percentage of time he spent on each department is given by percent-time.
3) Each department has only one HOD.

Functional dependency for Professor relations are

dept → HOD

Prof-code, dept → percent-time, HOD

Prof-code, HOD → percent-time, dept.

Two possible composite keys are (Prof-code, dept) & (Prof-code, HOD). and dept Prof-code is common attribute and dept, HOD are part of composite keys. Let the relation Professor has following information

Professor

| Prof-code | dept | HOD | percent-time |
|---|---|---|---|
| P1 | physics | X42 | 50 |
| P1 | Chemistry | AAA | 50 |
| P2 | Maths | ABC | 40 |
| P2 | Physics | X42 | 60 |

The above relations has following Problems:

1) Names of HOD are being repeated (Redundancy Problem)
2) Insertion anamolies - If a new Professor joins, unless he has been assigned for some classes we cannot insert his record.
3) Deletion anamolies - If professor P2 resigns, we have to delete all the records pertaining to Professor P2. When we delete the record of P2 that contains information about Maths dept, we accidentally delete complete information about Maths dept.
4) Updation anamolies - If there is a change in HOD of Physics dept, we have to make multiple updation.

Therefore normalization of the above relation is done by splitting it into two new relations as

Professor
i) Prof_code *
ii) Dept
iii) percent_time

department
i) Dept *
ii) HOD

Functional dependencies can be shown as

Prof_code, dept → percent_time

dept → HOD.

Two relations Professor and Department becomes

Professor

| Prof_code | Dept | Percent_time |
|---|---|---|
| P1 | Physics | 50 |
| P1 | Chemistry | 50 |
| P2 | Maths | 40 |
| P2 | Physics | 60 |

Department

| Dept | HOD |
|---|---|
| Physics | XYZ |
| Chemistry | AAA |
| Maths | ABC |