THE NORMAL FORMS (NF):

The normal forms (abbrev. NF) of relational database theory provide criteria for determining a table's degree of vulnerability to logical inconsistencies and anomalies.

The main normal forms are summarized below. (*MCS-43 syllabus starts from 4 NF*)

Normal form	Brief definition
First normal form (1NF)	There are no duplicated rows in the table. All underlying domains contain atomic values only
Second normal form (2NF)	No non-prime attribute in the table is functionally dependent on a proper subset of a candidate key
Third normal form (3NF)	Every non-prime attribute is non-transitively dependent on every candidate key in the table
Boyce–Codd normal form (BCNF)	Every non-trivial functional dependency in the table is a dependency on a superkey
Fourth normal form (4NF)	Every non-trivial multivalued dependency in the table is a dependency on a superkey
Fifth normal form (5NF)	Every non-trivial join dependency in the table is implied by the superkeys of the table
Domain/key normal form (DKNF)	Every constraint on the table is a logical consequence of the table's domain constraints and key constraints

FUNCTIONAL DEPENDENCY:

A functional dependency is denoted by X → Y, between two sets of attributes X and Y. X→Y means that the values of the Y component of a tuple in "A" depend on or is determined by the values of X component. In other words, the value of Y component is uniquely determined by the value of X component. This is functional dependency from X to Y (but not Y to X)

Example:
STUDENT (enrolno, sname, cname, classlocation, hours)

In the above relation, the following F.D. exists:

enrolno →sname (the enrolment number of a student uniquely determines the student names, so sname is functionally dependent on enrolment number).

Unnormalized Table: The relation below in not normalized as the values are not atomic.

Client	Orders	Dated
1	1 BANANA, 3 ALU, 4 GOBI	20-AUG-01
1	1 BANANA, 3 ALU, 4 GOBI	20-AUG-11
2	3 TAMATAR, 4 ALU	17-AUG-11
1	1 VINDI, 3 TAMATAR	16-AUG-11

1NF: The table below is in 1NF, as there are no duplicated rows and columns are atomic.

CLIENT	ORDERNO	PRODUCTNO	PRODUCTDETAILS	QUANTITY	DATED
1	1	1	BANANA	1	20-AUG-11
1	1	2	ALU	3	20-AUG-11
1	1	3	GOBI	4	20-AUG-11
2	2	4	TAMATAR	3	17-AUG-11
2	2	2	ALU	4	17-AUG-11
1	3	5	VINDI	4	16-AUG-11
1	3	4	TAMATAR	3	16-AUG-11

2NF: Partial dependency means dependency of certain attributes to a subset of candidate key. In the above Structure, the candidate key is (client, orderno, productno). But there is a dependency (orderno →client). As client is not fully dependent on the candidate key, we decompose it to remove this partial dependency. Again there is another partial dependency (orderno →dated). So we also remove dated table and put it in another table.

ORDERNO	CLIENT	DATED
1	1	20-AUG-11
2	2	17-AUG-11
3	1	16-AUG-11

ORDERNO	PRODUCTNO	PRODUCTDETAILS	QUANTITY
1	1	BANANA	1
1	2	ALU	3
1	3	GOBI	4
2	4	TAMATAR	3
2	2	ALU	4
3	5	VINDI	4
3	4	TAMATAR	3

3NF: In the above table,candidate key is (orderno,productno). But there exists an F.D. (productno →productdetails). Productno is not the candidate key of the table. So the column productdetails is partially dependent on candidate key. To remove this partial dependency, we decompose the structure as follows:

PRODUCTNO	PRODUCTDETAILS
1	BANANA
2	ALU
3	GOBI
4	TAMATAR
5	VINDI

ORDERNO	PRODUCTNO	QUANTITY
1	1	1
1	2	3
1	3	4
2	4	3
2	2	4
3	5	4
3	4	3

ORDERNO	CLIENT	DATED
1	1	20-AUG-11
2	2	17-AUG-11
3	1	16-AUG-11

BCNF: Mostly all tables that are in 3NF are also in BCNF. For BCNF, All determinants should be candidate key. Determinant means, the key which determines an attribute i.e. left hand side of a functional dependency. In above relations, we have already made all determinants candidate key.

To make a relation in BCNF, we should define all primary key and foreign keys of the table properly.