**Unit-I**

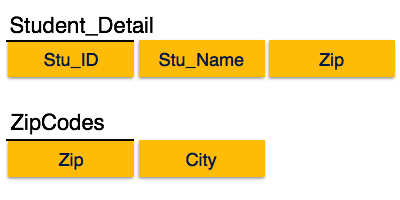
**Chapter1.4**

**(BCNF and 4NF)**

**Boyce-Codd Normal Form**

Boyce-Codd Normal Form (BCNF) is an extension of Third Normal Form on strict terms. BCNF states that −

* **For any non-trivial functional dependency, X → A, X must be a super-key**



In the above image, Stu\_ID is the super-key in the relation Student\_Detail and Zip is the super-key in the relation ZipCodes. So,

**Stu\_ID → Stu\_Name, Zip**

**and**

**Zip → City**

Which confirms that both the relations are in BCNF.

The Boyce Codd Normal form of the **database normalization** is an extended version of the 3NF. It is referred to as the 3.5NF. Any table or relation complies with the BCNF if it holds the following conditions:

* **The relation must be in the third normal form (3NF).**
* **For any functional dependency A -> B in the relation, A should be the superkey.**

You can clearly understand the BCNF concept through an example of employees working in multiple companies’ departments. Consider an Employee relation having Emp\_ID, Emp\_Nationality, Emp\_Department, Dept\_Type, No\_of\_Employees as attributes. The relation holds the following values:

**Employee**



The candidate for the above relation is {Emp\_ID, Emp\_Dept}. Neither the single Emp\_ID attribute can provide the department information, nor the Emp\_Dept can determine the employee information. So, the above relation does not comply with the BCNF. To make the above table in BCNF, divide it into three tables as follows:

**Employee\_Nationality**



Here, Emp\_ID is the candidate key. The functional dependency is Emp\_ID -> Emp\_Nationality. Hence, it is in the BCNF.

**Employee\_Department**



Here, Emp\_Dept is the candidate key, and the functional dependency is {Emp\_Dept -> Dept\_Type, No\_of\_Employees}. Therefore, the above relation also complies with BCNF.

**Employee\_ID\_Dept**



Boyce-Codd Normal Form says that if there is a functional dependency A → B, then either A is a superkey or it is a trivial functional dependency. A trivial functional dependency means that all columns of B are contained in the columns of A. For instance, (course code, professor name) → (course code) is a trivial functional dependency because when we know the value of course code and professor name, we do know the value of course code and so, the dependency becomes trivial.

**Let us understand what’s going on:**

**A is a superkey:** this means that only and only on a superkey column should it be the case that there is a dependency of other columns. Basically, if a set of columns (B) can be determined knowing some other set of columns (A), then A should be a superkey. Superkey basically determines each row uniquely.

**It is a trivial functional dependency:** this means that there should be no non-trivial dependency. For instance, we saw how the professor’s department was dependent on the professor’s name. This may create integrity issues since someone may edit the professor’s name without changing the department. This may lead to an inconsistent database.

**Fourth Normal Form (4NF):**

We have seen the multi-valued dependency in the above section. The table is said to be in the fourth normal form if it holds all the below conditions:

* The relation should comply with the Boyce Codd Normal Form.
* There should be no multi-valued dependencies between the table’s attributes.

We shall talk about the fourth normal form using Students relation. The Students relation has three attributes. Stud\_ID, Stud\_Course, and Stud\_Hobby. The table values are as below:

**Students**



The above relation is not in the fourth normal form (4NF), as it has multi-valued dependencies in it. The attributes Stud\_Course and Stud\_Hobby are dependent on the Stud\_ID attribute, which ends in multi-valued dependency. Therefore, to make the above relation in 4NF, we need to break the relation into two different relations as follows:

**Students\_Course**



**Students\_Hobby**



The above two relations are in the fourth normal form.

**Rules for 4th Normal Form**

For a table to satisfy the Fourth Normal Form, it should satisfy the following two conditions:

1. It should be in the **Boyce-Codd Normal Form**.
2. And, the table should not have any **Multi-valued Dependency**.

Let's try to understand what multi-valued dependency is in the next section.

**What is Multi-valued Dependency?**

A table is said to have multi-valued dependency, if the following conditions are true,

1. For a dependency A → B, if for a single value of A, multiple value of B exists, then the table may have multi-valued dependency.
2. Also, a table should have at-least 3 columns for it to have a multi-valued dependency.
3. And, for a relation R(A,B,C), if there is a multi-valued dependency between, A and B, then B and C should be independent of each other.

If all these conditions are true for any relation(table), it is said to have multi-valued dependency.

In the fourth normal form,

* It should meet all the requirement of 3NF
* Attribute of one or more rows in the table should not result in more than one rows of the same table leading to multi-valued dependencies

To understand it clearly, consider a table with Subject, Lecturer who teaches each subject and recommended Books for each subject.



If we observe the data in the table above it satisfies 3NF. But LECTURER and BOOKS are two independent entities here. There is no relationship between Lecturer and Books. In the above example, either Alex or Bosco can teach Mathematics. For Mathematics subject , student can refer either ‘Maths Book1’ or ‘Maths Book2’. i.e.;

**SUBJECT –> LECTURER**

**SUBJECT–>BOOKS**

This is a multivalued dependency on SUBJECT. If we need to select both lecturer and books recommended for any of the subject, it will show up (lecturer, books) combination, which implies lecturer who recommends which book. This is not correct.

SELECT c.LECTURER, c.BOOKS FROM COURSE c WHERE SUBJECT = 'Mathematics';

To eliminate this dependency, we divide the table into two as below:



Now if we want to know the lecturer names and books recommended for any of the subject, we will fire two independent queries. Hence it removes the multi-valued dependency and confusion around the data. Thus the table is in 4NF.

--**Select the lecturer namesSELECT c.SUBJECT , c.LECTURER FROM COURSE c WHERE c.SUBJECT = 'Mathematics'; --Select the recommended book namesSELECT c.SUBJECT , c.BOOKS FROM COURSE c WHERE c.SUBJECT = 'Mathematics';**

**Other References**

[Fourth Normal Form (4NF) - 4NF in DBMS - 4NF in Database (tutorialcup.com)](https://www.tutorialcup.com/dbms/fourth-normal-form.htm)

[DBMS - Joins - Tutorialspoint](https://www.tutorialspoint.com/dbms/database_joins.htm)

[Introduction to Database Normalization (softwaretesttips.com)](https://www.softwaretesttips.com/database-normalization/)

**Suggested Book References**

1. J. Date, “An Introduction to Database Systems”,Addison Wesley.
2. Navathe,“Fundamentals of Database System”, The Benjamin / Cummings Publishing Co.