**BCNF**

Boyce- Codd Normal Form is one of the forms of the normalization. A database table is in BCNF if and only if there are no non-trivial functional dependencies of attributes on anything other than a superset of a candidate key. A candidate key is a column or combination of columns in a table that forms a unique key in the database. The combination of attributes can be used to identify a database record without referring to any other data. Each table can contain multiple candidate keys, any one of which can qualify as the primary key. A table contains only one primary key.

**Rules for BCNF**

1. It should be in third Normal Form
2. And for any dependency A- B , A should be super key.

A relation is in BCNF if every determinant is a candidate key. Consider a database table that stores employee information and has the attributes *<Student\_Id*>, *<Subject>* and <Professor>.

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Subject** | **Professor** |
| 1013 | Software Engineering | Dhairya |
| 1014 | Operating Systems | Smith |
| 1015 | CPP | Sonia |

This table satisfies the 1st Normal form because all the values are atomic , column names are unique and all the values stored in a particular column are of same domain. This table also satisfies the 2nd Normal form as there is no partial dependency and there is no transitive dependency hence the table also satisfies the 3rd normal form but in order for the table to satisfy BCNF , we will decompose this table into two tables.

**Student Table**

|  |  |
| --- | --- |
| **Student ID** | **Professor ID** |
| 1013 | 101 |
| 1014 | 102 |
| 1015 | 103 |

**Professor Table**

|  |  |  |
| --- | --- | --- |
| **Professor ID** | **Professor** | **Subject** |
| 101 | Dhairya | Software Engineering |
| 102 | Smith | Operating System |
| 103 | Sonia | CPP |

**The above relation satisfy Boyce-Codd Normal Form.**