DATABASE NORMALIZATION - BCNF

Boyce Codd Normal Form (BCNF) was created as an extension to the third normal form, or 3NF, in 1974 by Raymond Boyce and Edgar Codd. The men were working to create database schemas that minimalize redundancies with the goal of reducing computational time. The third normal form removes columns that are not dependent on the primary key in addition to meeting the guidelines in the first and second normal forms. BCNF, which is sometimes referred to as 3.5NF, meets all the requirements of 3NF and requires that candidate keys not have any dependency on other attributes in a table.

BCNF goal is to increase data-integrity by organizing the columns and tables of a relational database in order to achieve database normalization

If a relation is in BCNF then all the functional dependencies will have a super key on its left side. you can consider this as an introduction to a composite key. If there is a functional dependency x->y then x must be a Superkey in that relation.

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 identifies 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.

Super Key:

A *super key* is a group of single or multiple keys which identifies rows in a table. A Super key may have additional attributes that are not needed for unique identification.

Composite Key:

A *Composite key* in SQL can be defined as a combination of multiple columns, and these columns are used to identify all the rows that are involved uniquely.

Example:

Let's assume there is a company where employees work in more than one department.

EMPLOYEE table:

EMP_ID	EMP_COUNTRY	EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
264	India	Designing	D394	283
264	India	Testing	D394	300
364	UK	Stores	D283	232
364	UK	Developing	D283	549

In the above table Functional dependencies are as follows:

1. **EMP_ID** → EMP_COUNTRY
2. **EMP_DEPT** → **{**DEPT_TYPE, EMP_DEPT_NO**}**

The table is not in BCNF because neither EMP_DEPT nor EMP_ID alone are keys.

To convert the given table into BCNF, we decompose it into three tables:

EMP_COUNTRY table:

EMP_ID	EMP_COUNTRY
264	India
264	India

EMP_DEPT table:

EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
Designing	D394	283
Testing	D394	300
Stores	D283	232
Developing	D283	549

EMP_DEPT_MAPPING table:

EMP_ID	EMP_DEPT
D394	283
D394	300
D283	232
D283	549

Candidate keys:

For the first table: EMP_ID
For the second table: EMP_DEPT
For the third table: {EMP_ID, EMP_DEPT}

Conclusion

The above example demonstrates BCNF. In layman terms a table is said to be in BCNF if the left side of the table is represented as a unique identifier for the entire table. This unique identifier is called a Super Key. It can be one column or more than one column grouped together. Which is why we use the term Super Key to represent the unique identifier that determines the entire form.