To determine whether the given relation is in BCNF (Boyce-Codd Normal Form), we need to check if it satisfies the following condition:

For every non-trivial functional dependency (X -> Y) in the relation, X must be a superkey.

Functional dependencies:

*emp\_name, team\_name -> leader\_name*

*leader\_name -> team\_name*

The candidate key for this relation is (emp\_name, team\_name), which consists of both attributes. Now let's check if each functional dependency satisfies the BCNF condition:

**emp\_name, team\_name -> leader\_name:**

Both emp\_name and team\_name are part of the candidate key.

Therefore, this functional dependency satisfies the BCNF condition.

**leader\_name -> team\_name:**

leader\_name is not a superkey since it is not part of the candidate key.

**Hence, this functional dependency violates the BCNF condition.**

Since the relation has a functional dependency (leader\_name -> team\_name) that violates BCNF, we need to decompose the relation into smaller tables to achieve BCNF.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| emp\_name | team\_name |  | leader\_name | team\_name |
| John | Amex |  | John | Amex |
| Michal | Fox |  | Patrick | Fox |
| Trawinski | Leopards |  | Klaudia | Amex |
| Smith | Leopards |  | Klaudia | Leopards |
|  |  |  | Christian | Leopards |

In the new decomposition, each table satisfies the BCNF condition:

**Team\_Employees:**

The primary key is (emp\_name, team\_name).

There are no non-trivial functional dependencies, so it also satisfies BCNF.

**Team\_Leaders:**

The primary key is leader\_name.

There are no non-trivial functional dependencies, so it automatically satisfies BCNF.