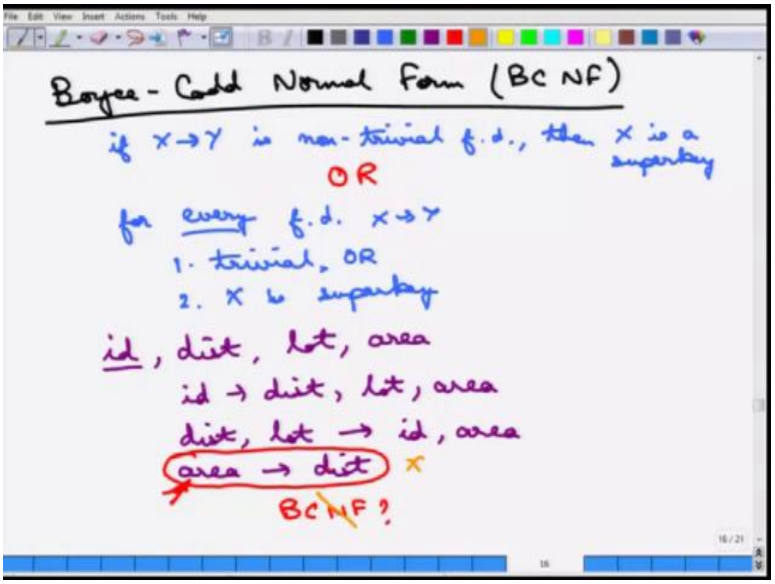
Fundamentals of Database Systems
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 15
Normalization Theory: BCNF

The next form is called Boyce Codd Normal or BCNF.

(Refer Slide Time: 00:13)



This is the Boyce codd normal form, a relation is in Boyce codd normal form if X Y is non-trivial f d, then X is a super key. So, for every non trivial f d the left side must be a super key there is an alternative definition, which is for every f d X to Y, either it is trivial OR X is super key. Having the equivalence is very easy; this is essentially just breaking this down to that non trivial thing down.

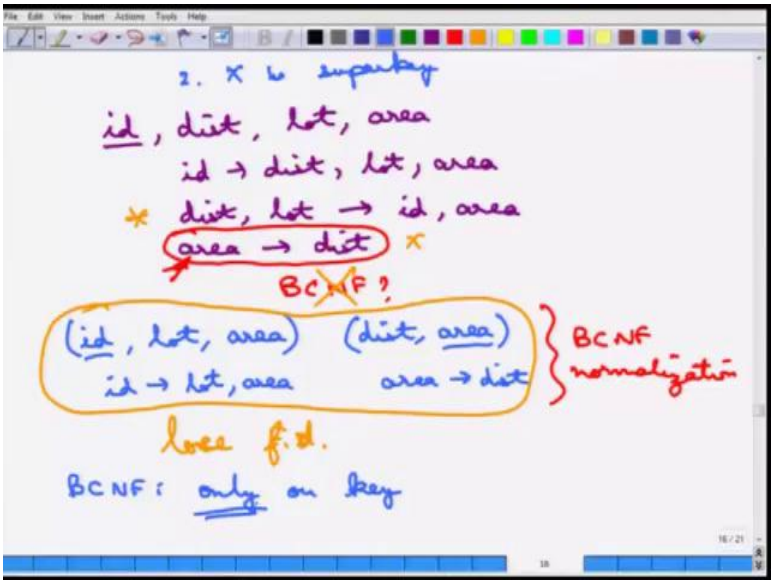
So, what does BCNF tries to do is to BCNF tries to go to a higher form, then 3 NF it tries to say that whenever there is a functional dependency, it is non-trivial then the left side must be on the super key. So, it tries to say that every relation, the functional dependencies must be only on the keys, it cannot be on anything else. So, they are... So, if there is a functional dependency X to Y, the X must be a super key, so that is what it tries to do.

Now, the process of BCNF normalization may be actually a problem. So, before that let us consider the example that we were looking at earlier. So, this was the relation after that L 1 1 that we looked about earlier. So, this is just to recollect, this was the

dependency with id determining everything dist, lot and area; however, there was another dependency which was saying distance and lot together determines id and area and there was let us say this is another thing area determines district.

So, the question is, is this in BCNF the answer is no, because of this functional dependency. The other two functional dependencies are fine, because of this functional dependency area determines district, but area by itself is not a super key. So, this is where this fails, so this not this relation is not in BCNF.

(Refer Slide Time: 02:58)

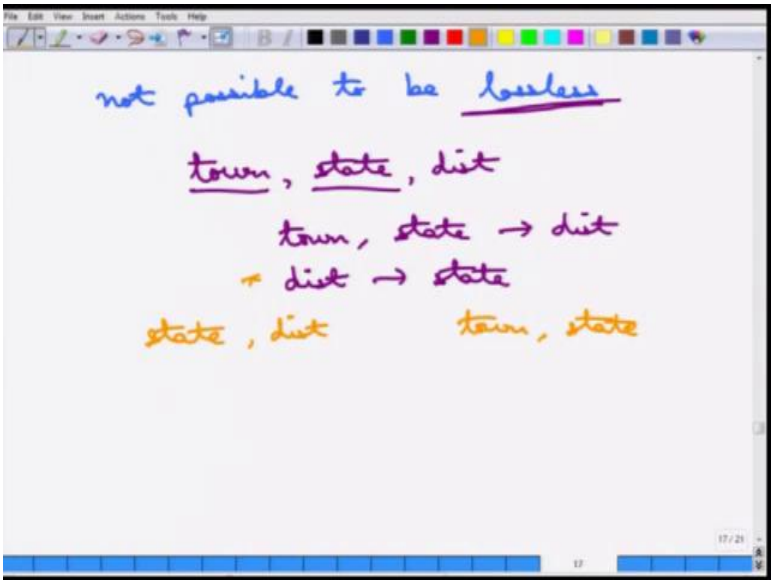


So, if one wants to normalize this BCNF what one can break this down is that id, lot, area, then it can be district to area that is it correct. Now, what are the functional dependencies in this? The first, here the functional dependency is id determines lot, area and area determines district. So, very importantly after BCNF normalization, this is the process of BCNF normalization one can see something important here is that it can lose functional dependencies, so there are certain functional dependencies that are lost.

So, what are the functional dependency that is lost? This functional dependency is lost, so after the BCNF normalization is done, it is not always that one can preserve all the functional dependencies. So, this is can be problematic may not or whatever it, may or may not be problematic, but that is the thing. So, functional dependencies may be lost and informally BCNF is can be summarized as everything depends on only the key, so that is the way of saying.

So, why is BCNF normalization important is that, if the area to district this actually introduces a problem. So, if one knows the area, one knows the district, but area is not a key. So, in it may happen the changing area, area is changed due to something else and the district needs to be also changed. So, again the problem comes with those anomalies that one attribute is touched and other attributes are touched unnecessarily when it is not useful to touch it. So, the other problem with BCNF decomposition is that it may not be always possible.

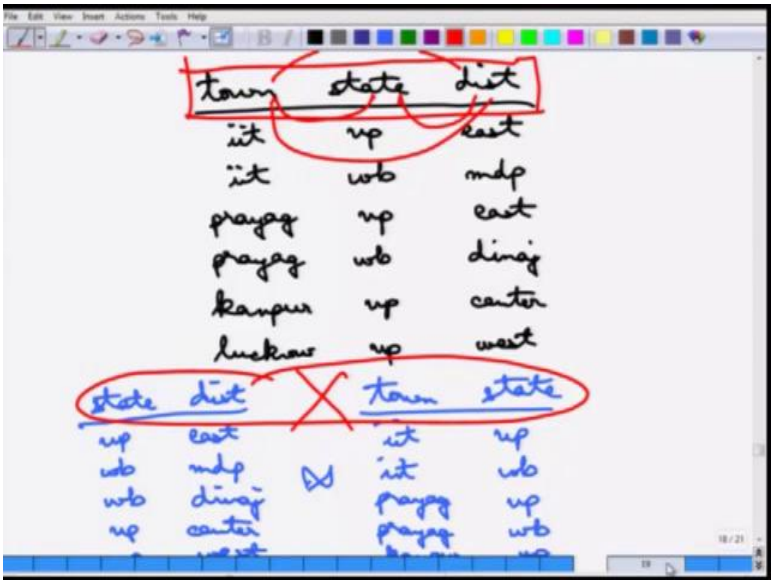
(Refer Slide Time: 04:57)



So, not possible in the sense that it may not be always possible to be lossless, so it can lose certain, sometimes it can lose not just functional dependency, the decomposition may not be lossless. So, for example, let us consider a relationship in the following manner a town, state and district. So, which where... So, the functional dependencies are town and state together determines the district and the district name determines the state.

So, suppose these are the two functional dependencies now of course, this is not in BCNF, so to break it down into BCNF the rules says that the problem is with this district to state. So, let us break it down into this two relationships state with district and town and state. So, let us isolate district and state together and this is also town and state.

(Refer Slide Time: 06:08)

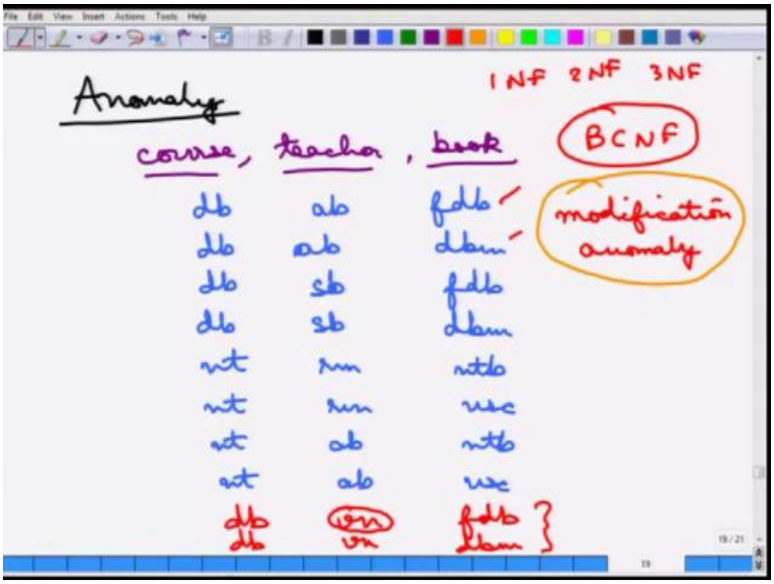


But, here is an example where this will fail, so town state and district this is an example. So, IIT is a town in UP and suppose it is in the district east and so on and so forth, IIT is a town also in the state West Bengal, then Prayag is in UP and WB as well and Kanpur and Lucknow all those things. Now, if one breaks it down to this state district and town state, so if one breaks it down to state, district and town, state let us see what does one get.

So, state, district if one breaks it down, so you get UP, East, West Bengal, MDP then West Bengal, Dinaj, UP, center, UP, West and town and state if one breaks it down, this is and well. So, this is all the other six is there, the point is one can check that this is definition this decomposition is not lossless, because if one joins this then you do not get back the original relationship. So, this is glossy and this cannot be allowed under any circumstance.

So, this decomposition is wrong, this is not allowed and one can also say just to argue that may be the BCNF decomposition rule was wrong that there is no way to break this relationship down into any other way. For example, if one tries to break it down into town state and town district or state district and town district, none of this will actually give back the original form. So, this is this problem with BCNF decomposition, so it is not always possible to ensure that a relationship is in BCNF, it may happen it may be possible it may not be possible.

(Refer Slide Time: 08:11)



There is another problem with BCNF it is the anomaly, there is a particular anomaly that can happen with BCNF. So, for example, consider the relationship of a course and teacher and books, so what essentially means is that, so to all these three together is the primary key. So, what does it mean is that suppose you consider a course database, it can be taught by many teachers in a particular institute and each teacher can take up many of the books.

So, a particular invocation of this thing may be in the following manner. So, let us say there is the database course which can be taught by a teacher say a b and whatever these are the name of the books and this can be taught by whoever and so on so forth. So, let us just see what happens, so essentially what one means is that the course can be taught by any teacher and the teacher can adopt any book and similarly this can happen with lots of there can be other courses as well, which this where this can happen.

Now, the problem that will happen is that the functional dependencies, there are essentially no functional dependency. So, this is determined to be in BCNF, so what I mean to say is this relation is in BCNF, because there is only one functional dependency and nothing is valid. So, this is all seems to be fine, but modification anomalies are still there, so there is a modification anomaly there in this thing.

So, although this relation is in seems to be all fine from the point of view of normalization. So, it is in 1 NF, 2 NF, 3 NF it is in 1 NF, 2 NF, 3 NF and BCNF all of these things, so it seems to be all fine there is still a modification anomaly. Why is there

a modification anomaly? One can consider that if there is a new teacher that comes who can teach database. So, suppose introduce the name of a teacher v n who can teach database.

Now, there can be two books adopted by it, so f d b and d b m, so you require both the tuples to be inserted. So, even though there is only one piece of new information that is inserted into the database two tuples are added. And similarly, if a b today leaves from this institution and so there is only one teacher which deleted there are two tuples are needs to be deleted and so on so forth. So, you can see that there is a problem with this relationship parse.

So, what it essentially means is that the functional dependencies and the keys can normalize a relation only up to a certain extent. So, it can make the design of a database good up to a certain extent, it cannot make it completely bereft of any modification anomaly. So, there can still be modification anomalies even if one covers all the normal forms that is possible with a functional dependencies and keys. So, that will be the start of our next topic which is on multi value dependencies.