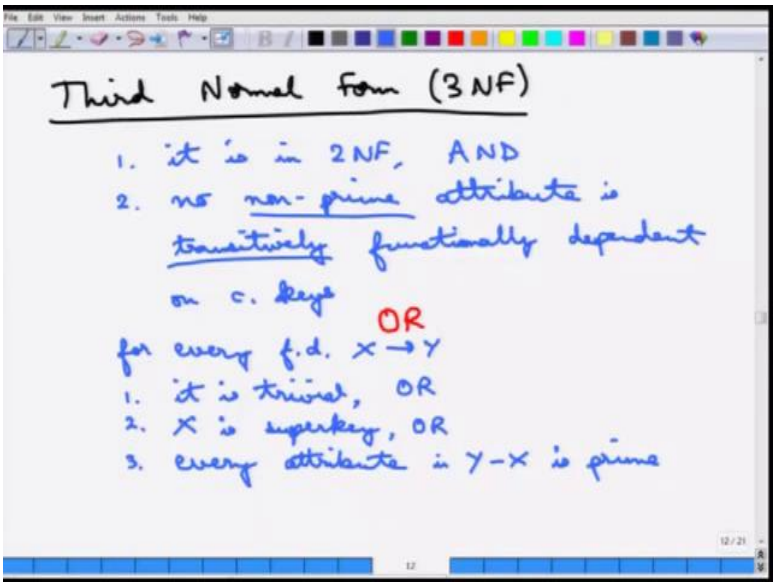
Fundamentals of Database Systems
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 14
Normalization Theory: 3NF

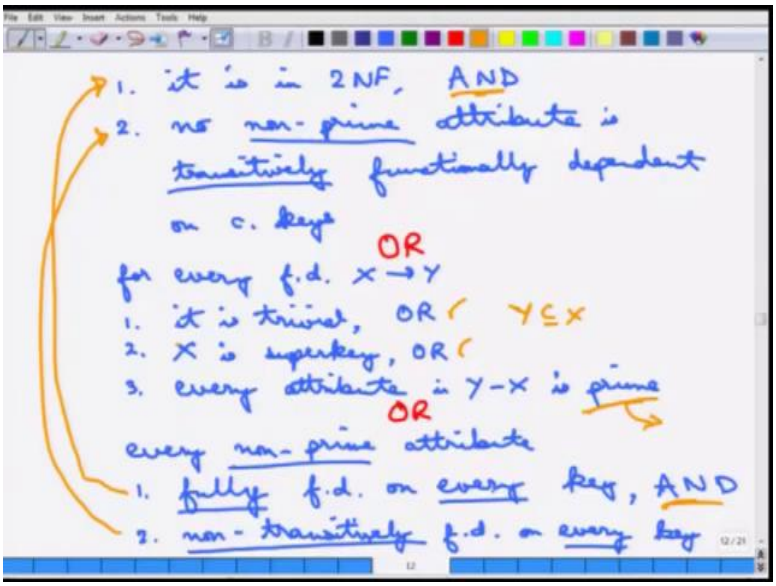
So, we will next go over the 3rd normal form or the 3NF, more popularly known as the 3NF.

(Refer Slide Time: 00:11)



So, a relation is in 3rd normal form, if first of all it is in 2NF and no non prime attribute is transitively, functionally dependent on candidate keys. This is one way of defining it. There is an alternative definition ...or … let me write down that alternative definition. For every functional dependency X determines Y, either it is trivial OR X is super key. So, that means, X functionally determines every Y. Or, every attribute in Y minus X is prime. There is another way of alternatively defining this.

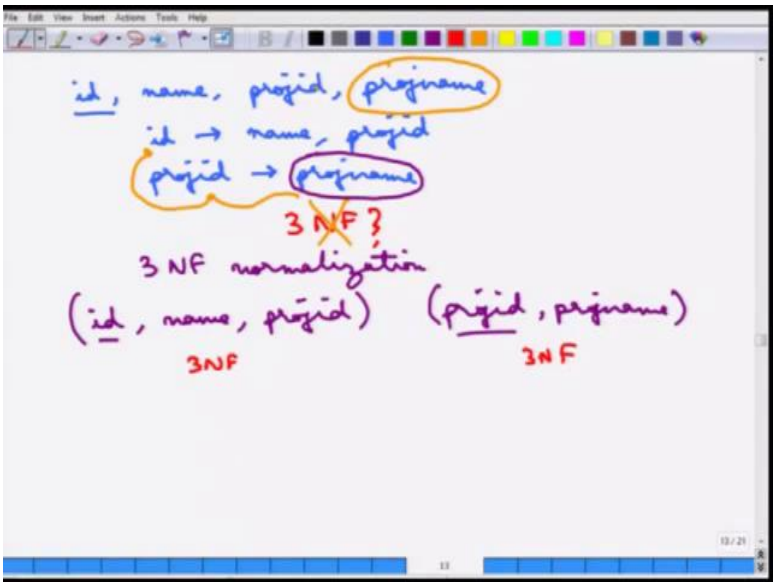
(Refer Slide Time: 02:01)



So, or, every non prime attribute is fully functionally dependent on every key and non transitively functionally dependent on every key. So, let us start from the third definition, probably that is easiest to understand. So, we take every non prime attribute and it must be both fully functionally dependent and non-transitively functionally dependent on every key. Fine. So, how is this equivalent, for example, to the first definition. This is equivalent to this. So, this is essentially the definition of 2NF, and this is equivalent to this definition.

So, every non prime attribute is functionally transitively dependent on this. So, at least the first and third definitions are equivalent and let us see now the second definition. How is it equivalent to the other definition? So, see for every functionally dependent thing, if it is trivial so, that means, Y is a subset of X, that means, Y is not non prime etc., or X is a super key, that means X is part of this prime attribute. So, the things is ... this part of the prime attribute or every attribute in Y minus X is prime, that means, every attribute in Y minus X also functionally determines everything else. This, remember that these are all or and this is and, this is also and. So, these are the three equivalent definitions of 3rd normal form and let us now take an example.

(Refer Slide Time: 04:03)



So, again we will consider this following kind of definition. So, it's id, name, project id, project name and here we are saying id is the key. So, the functional dependencies are essentially id, of course, determines everything else, but it also determines name and project id and then project id essentially determines project name. Is this in 3NF? So, it is not in 3NF, because of the following reason. So, let us identify one at a time. So, id is a prime attribute, so nothing to be done. Name, name is fully determined on ... by the id.

So, and it is not transitively dependent on any other thing. So, name is also fine. So, is project id. However, project name has a problem. Project name, it is a non prime attribute and it transitively determines on id. So, because project name is determined by project id which in turn is determined by id. So, project name, this fails and the entire question of whether this is 3NF or not is answered in the negative. This is not in 3NF. Now, we would like to again do this process of 3NF normalization and once we do that, so this is 3NF normalization.

And once we do that, we will essentially try to isolate the offending attribute. Offending attribute here is the project name. So, we will try to isolate it. So, we will break this relation into two parts. First is the id, name and project id. So, this is the first relation. And, the second relation is project id with project name. So, this is both are in 3NF. So, let me write down this is the keys and this is the only functional dependencies that are available.

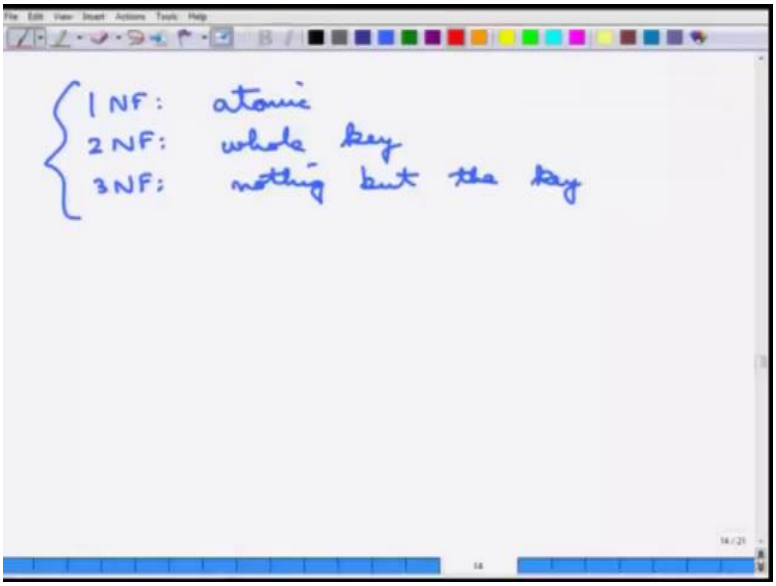
So, this is again in 3NF one can determine. So, this is in 3NF normalization. So, what again is the problem with 3NF? Why it is important to break a relation into 3NF? So, let us consider

this example … and the project name is determined on project id which in turn is determined by id. So, once more if the project id changes or if the id changes, project name is affected unnecessarily. So, if project name affects this id unnecessarily, it does not make sense.

So, if project name is changed, the effect is carried in a ripple manner to the id. So, corresponding to every name who works in that … so every employee who works in that project name gets affected. So, all those tuples now need to be modified. On the other hand, when it is broken down into this 3NF normalized forms, the project name is isolated from this.

So, the first table is not touched at all and in the second table there is only one change that needs to be done, which is just the project name is changed. So, this is a much better design as we can see. So, we have looked at these three normalized forms and one can informally summarize them in the following manner.

(Refer Slide Time: 07:15)

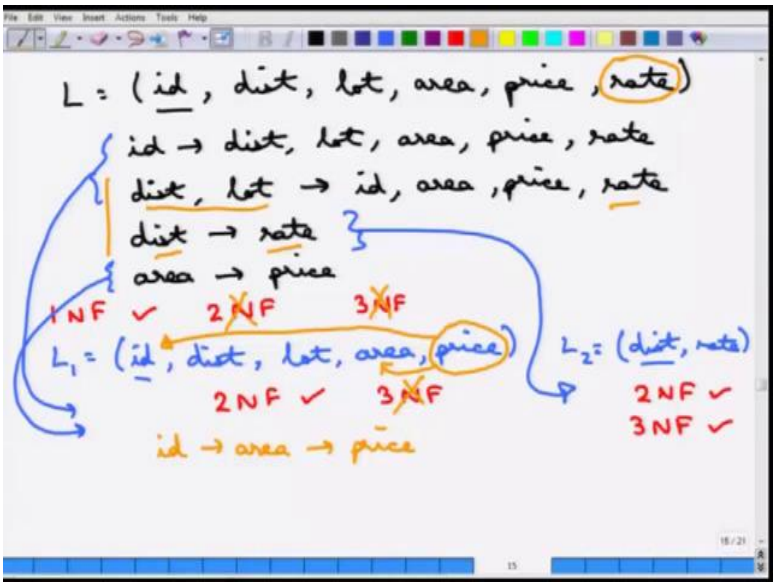


So, 1NF is that all attributes are atomic, correct? So, that is all that can say. In 2NF, all attributes must depend on the whole keys. So, that is why it's called a fully functionally dependent. And 3 NF, all attributes must depend on nothing but, the key or nothing. So, it is of course, in 2NF and nothing but the key. So, that means, there is no transitive dependency it must depend only on that key. So, this is of course … I mean ... an informal way of remembering what the 1NF, 2NF and 3NF does.

So, how do we determine if a relation is in 1NF? There should not be any multi-valued attributes or nested relations etc. etc. And in the 2NF, it should fully functionally depend and in 3NF it should transitively depend. So, that is the way to do it. And how does one remedy? So, if a particular relation is not in 1NF then what one needs to be … done? Is that, offending attribute, the multi-valued attribute, the nested attribute needs to be broken down into atomic attribute.

So, what does one do when a relation is not in 2NF? The offending key which does not fully depend on one of the candidate keys is isolated. The same thing is done for the 3 NF. The offending key which is transitively dependent on a key. So, those two functional dependencies are broken into two relations. So, that is the way for the normal forms. And let us do a little bit of example to ensure that the understanding has gone through.

(Refer Slide Time: 08:45)



So, here is an example that we will do. So, this is L, we will just try to say what is the thing. Here is an id, district, lot number, area, price and rate. And this is the candidate key. Okay, so, what are the functional dependencies in this? Of course, id determines everything else because id is the key. So, id determines ... district, lot, area and price and rate. Now, there are some other functional dependencies as well. So, district and lot if one knows the district and lot together, it can determine which id it is in, which area it is in, and the price and rate of the part of land that one talks about.

So, if one knows the district then one can know the rate of the land that is being there. And if one knows the area in which where the land is located, one can know the price of the... know, because this essentially area time is the unique price, etc. So the first question is, is this in 1NF? The answer is yes, because we are just assuming that every attribute is atomic. So, and the next question is, is it in 2NF? The answer is no. Tthe reason why the answer is no is that consider this attribute, let us say, rate.

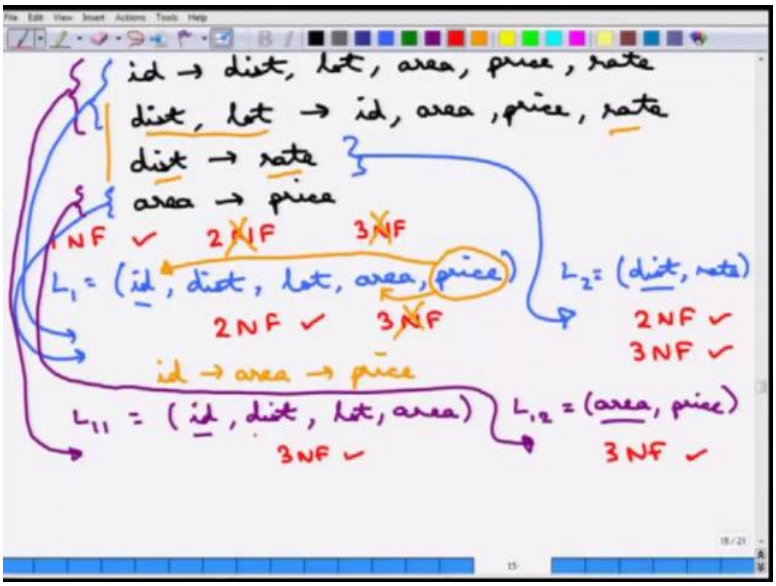
So, rate depends on district and lot and rate also depends on district. So, it is not fully dependent on district and lot. So, it violates this 2NF, so this is not in 2NF. The question then comes is it in 3 NF? Now of course, not without any testing one can say this is not in 3NF, because if something is not in 2NF it cannot be in 3NF. So, if this goes progressively. So, now, let us try to see how to do this 2NF normalization.

So, if one attempts to break it up into 2NF normalization, the following relations can be produced. So, the L_1 ... so, we essentially need to get rid of all the offending keys in the 2NF testing method. So, what one does is the following this L_1 is produced with the following attributes: area and price. This is id and L_2 is simply district to rate. So, that is what one tries to identify and here the key is district. And the same functional dependencies flow through. So, these two flows here, this again flows here and this flows there. These are the functional dependencies.

Now, one has to test whether this is in 2NF or 3NF, etc, so L_1, is L_1 is in 2NF? By the way, it is probably it's easier to argue about L_2. Is L_2 in 2 NF? It is yes, I mean this is nothing to be done. This is very simple. It is in 2NF. is L_1 in 2NF? It is in 2NF, because one can check that these all go through. The question is, now is L_2 in 3NF? L_2 is 3NF, because one very simple rule is that if there are only two attributes and one of them is the key and that is the only key, it is in any normal form that one can think of this. This rate is determined only by district, there are no other functional dependencies to start with.

What about L_1 is it in 3NF? It is not, the reason why it is not is that, this attribute price is determined ... it depends on id. Because, id is of course the key. But it is also dependent on area. So, that means, id determines … so, this is a problem, id determines area which determines price.

(Refer Slide Time: 12:50)



So, this is the offending key. So, this is not in 3NF. Once more, if one wants to break it up into 3NF, then what does one need to do, is to break this up into two more parts and the first thing is L_{11}, let us say, which is simply id, district, lot and area. And third one is L_{12} is simply area and price. So, the offending price is broken down into this thing. So, this is area and this again remains as id. So, these two functional dependencies are satisfied here. And this functional dependency is satisfied here.

So, then the test comes is this is in 3NF? Of course, as I said, this is easy. Is this L_{11} is in 3NF? Yes, it is in 3NF as one can test, because there is only one key so far. And the district and lot together determines everything else, but that doesn't matter right now. So, this is all about 3NF, we will start on the next form later.