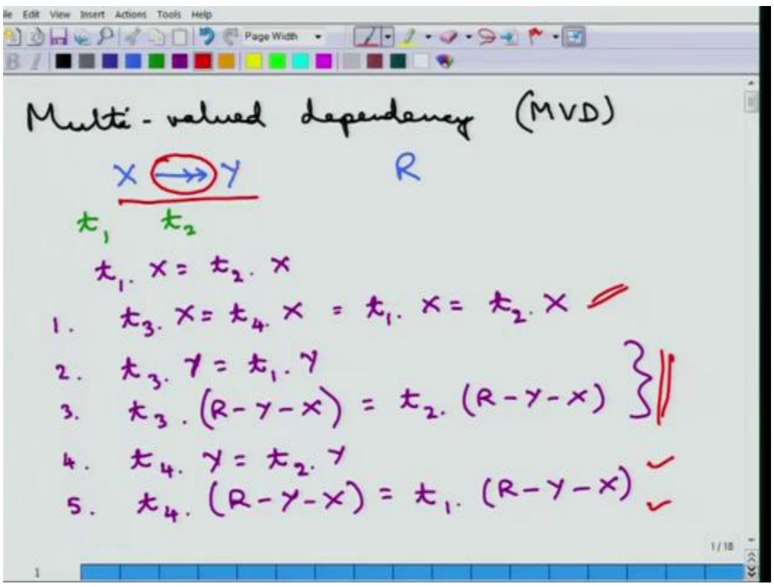
Fundamentals of Database Systems
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 16
Normalization Theory: MVD

As we saw up to 3 NF or BCNF is practical, but it may still have certain problems. For example, in the database course example that we saw if a teacher teaches the course on database, the books must be common to all the teachers. Now, to handle such anomalies and to go to higher normal forms, we will recover the concept of what is called a multi valued dependency.

(Refer Slide Time: 00:34)

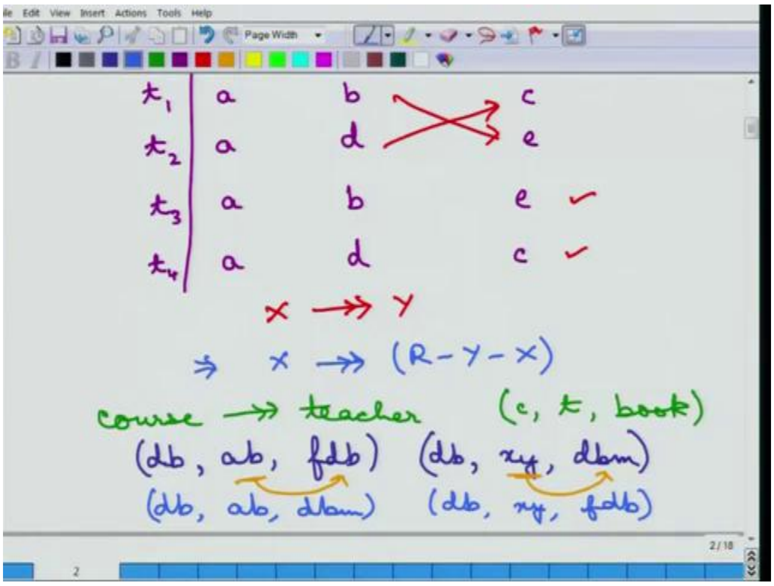


So, a Multi Valued Dependency it is mostly written as MVD instead of a functional dependency, which is sometimes written as FD. So, it is a Multi Valued Dependency MVD, it is defined in the following way. So, suppose Y depends multi value depends on X, so this is the symbol first of all note the symbol, this is a different from the FD symbol, because there are two arrows. And now this holds for all value or for a relation schema R, this holds if for all legal instances of R.

Now, if there is a pair of tuples t 1 and t 2 such that the following happens, now suppose these are the conditions. So, suppose t 1 dot X is equal to t 2 dot X, then there are two more other tuples, such that t 3 dot X is equal to t 4 dot X and those are then equal to t 1 dot X which means these are also equal to t 2 dot X. So, all these four tuples of the same X part and if t 3 dot Y is equal to t 1 dot Y and t 3 dot R minus Y minus X. So, R minus Y minus X is essentially all the attributes that are in the relation R, but not in Y and not in X, if these two conditions happen then it must be that the following two conditions happen.

So, t 4 dot Y must be equal to t 2 dot Y and t 4 dot this R minis Y minus X, so the attribute values for R minus Y minus X for t 4 should be equal to that for t 1. So, this is the formal definition of this, so if these two conditions happen when this is the case, then these two must occur. So, this is the definition of the multi valued dependency X, so multi value determines Y. Now, an example will be the better way of understanding it, so here is what I am trying to do.

(Refer Slide Time: 03:00)



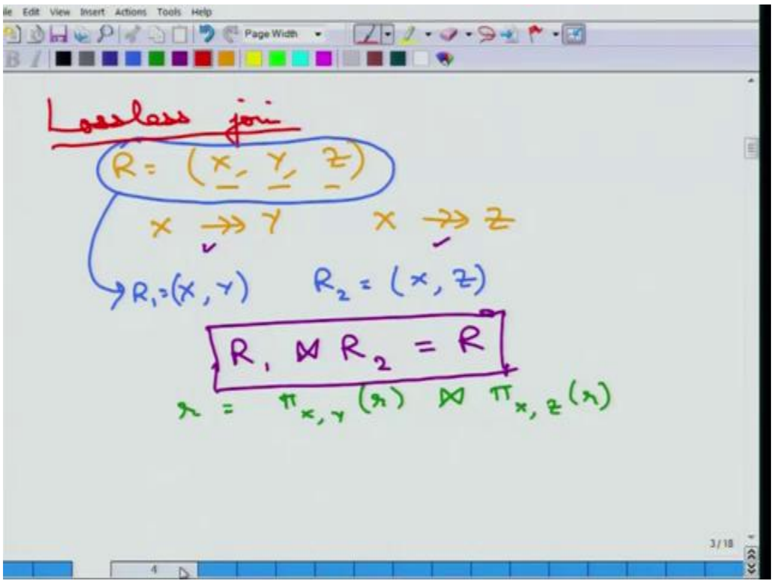
So, suppose this is X this is Y and this is all the attributes that are not in X and Y. So, this is essentially R minus Y minus X. And let us now consider first tuple t 1 which has value a and this is b, this is c these are the values. Now, the t 2 also has the same X values which is a and it has got different Y and R minus Y. So, let us say those are d and e, now if t 3 has e and t 4 has a then it must happen that this combination b c and d e must again take place.

So, it is essentially b e and d c this they must be happening this or they must be happening. So, because this is the combination and this is the combination, both these combinations must be there in the relation and then it is called that X multi value determines Y. Now, from the symmetry of this one can say that if X multi value determines Y, so if there is an MVD X goes to Y then there is also the MVD that X multi value determines to R minus Y minus X and that is obvious from the example and the definition of this.

And in the database course that we saw the example of the database course that we saw earlier, essentially what is happening is that the course MVD to from course to teacher exists when the relation is this course teacher and book. Now, what does that mean is that suppose there is the course is this database. So; that means, that if there is a course database which is the course and if there is a teacher let us say a b and if that teacher teaches the book f d b, if this happens and if there is another tuple for the same course database and then there is some other person x y which teaches some other books d b m book, if these tuples happen.

So, if the teacher a b teaches the book f d b and if the teacher x y teaches the book d b m, then it must happen that in that relation the two other tuples must also exists. So, a b must be able to teach with the book d b m and x y must be able to teach with the book f d b. Now, if this why is this required, intuitively if this does not happen then there is something special about the book and the teacher. It is not about the course, then the teacher probably prefers the book more much more and refuses to use the other books and similar things may happen. So, that is not a desirable situation and this is why this multi value dependency is useful. So, now, if we use this then we can move ahead and see how multi valued dependency is actually useful for a lossless join.

(Refer Slide Time: 06:00)

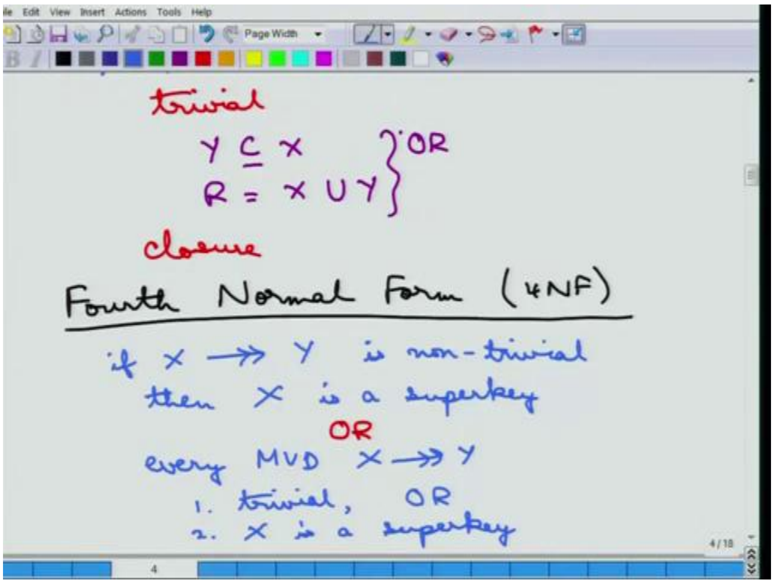


We define this concept of a lossless join and here is a thing, so suppose there is this relation R with X, Y and Z and then if X MVD Y then also X MVD Z determines. Now, if this relation is now broken up into the following manner that this is X, Y, so R 1 is equal to your X, Y and R 2 is equal to X, Z let us say now and then we take the join of R 1 natural join of R 1 with R 2 which is of course, on x this must give us back the relation R, so that is the whole point.

So, essentially we can say write it in a different manner that R is equal to your... So, if you project it only on X and Y then this is simply equal to if you project it on X and Z on R. So, that is the same as saying R 1 also, so this must hold then this satisfies your MVD and then if this MVD and if this lossless join condition holds, this because this MVD is hold then and R can be broken down into X, Y and X, Z which is a much better design as we saw.

Because, we do not need to store all the tuples of this teacher and course, we just need to store the d b is can be taught by this teacher and d b can use this book and that is good. So, this is essentially why MVD is useful, because it defines the concept of lossless join and now similar to all the other things is this MVD can be used to define certain things.

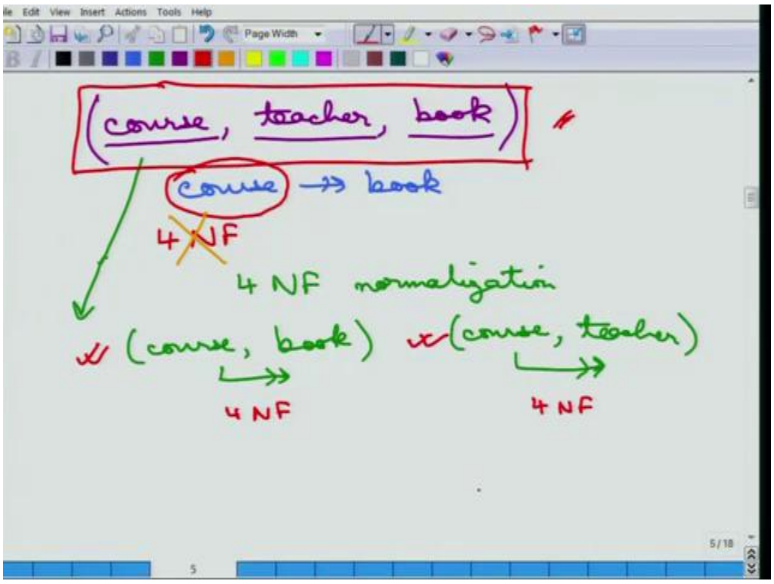
(Refer Slide Time: 07:55)



So, before that just simple things is that X MVD Y is called trivial, if the following things happen, either Y is part of X or simply R is equal to X union Y. So, I mean this is in R condition, this there is either this or this happens then this is called a trivial and any other MVD is nontrivial. So, then analogously like the functional dependencies the closure of a set of MVD is can be defined etcetera and then there are different inference rules for MVD is such as the complementation, augmentation, transitive, replication and equivalence.

But, using that let us define what is called the fourth normal form, so this is the fourth normal form or the 4 NF and 4 NF essentially, a relation is said to be in 4 NF, if there is a functional dependency if X this is nontrivial then X is a super key. So, for every non trivial thing the left side is a super key, now there is an alternative definition just as we saw the definition for other things, there is an alternative definition and the alternative definition says that, for every MVD X to Y the two conditions happen, either it is trivial OR X is a super key is a same definition and this is what the definition of the fourth normal form is. So, that is how to define the fourth normal form and the example we have already seen just to recap it.

(Refer Slide Time: 10:03)

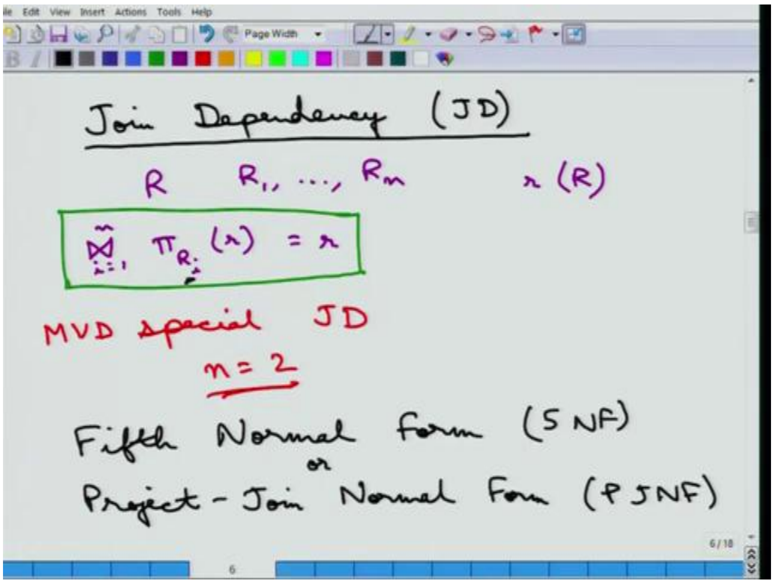


So, this course teacher book and the primary key is all of this together is the primary key and these if the MVD course to book exists there, if course to book exists then this relation can be broken down into this course teacher and course book. Now, what happens in this case is, so if this is the only MVD that is there, the first thing we need to test is whether this is 4 NF or not. The question is whether this is 4 NF or not, now the thing is course is not a super key, so this relation is not in 4 NF.

So, that is the problem with this and therefore, once we do the 4 NF normalization, suppose we do this what is called a 4 NF normalization then the following things happen is that this is broken down into the following two relations, which is course with book and course with teacher and the only functional the MVD is are from course to book and from course to teacher. So, which are both trivial, because X goes to Y and thus right side and the R minus Y minus X is empty, so this is a trivial, so this is by definition in 4 NF.

So, this is a better design than using this entire thing, because as we argued earlier this has got modification anomalies, in the sense that if a new book is introduced multiple updates require or if a new teacher is introduced multiple updates are required in this relation, but not in these two. So, these two are in a better design that is the whole point of the MVD and the 4 NF the forth normal form.

(Refer Slide Time: 12:07)



Now, the MVD can be extended to a general concept which is called the Join Dependency or JD, this is essentially as I have been saying that from MVD onwards it goes into the realm of more theoretical than practical things and JD is an even more higher form of theoretical dependency, it essentially says that if there is a relation R for it is suppose the relation R consists of this it can be broken up into these things R 1 to R n, then if you take the any R i from this relation r belongs to this R is a relation instance for the relation schema capital R.

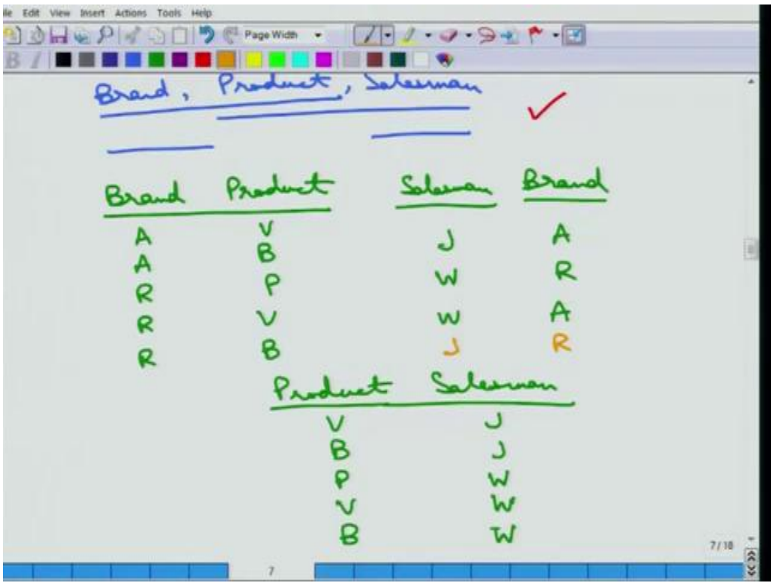
And if you join all of them together then you get what is known as the original relation R this is of course, from R equal to 1 to n. So, if this condition holds then there says to be join dependency. So, essentially the idea of join dependency is the following in the MVD what we did is that we broke up the relation R into two parts R 1 and R 2 such that the join of R 1 and r 2 the natural join of course, gives back the relation the complete relation R.

Now, here what we are doing is that we are breaking up the schema capital R into different set of attributes R 1 to R n such that the join of all of these together gives back the actual the original relation. So, that is one it happens for any for all possible instances or all legal instances of the relation of this small r. So, then this is called the join dependency and one can see that multi valued dependency can be now defined as a special case of join dependency.

So, MVD is a special join special case of join dependency, where the n equal to 2 that is it essentially. So, it is just broken up into two parts and if you break it up into multiple parts then this is called join dependency. Now, just as we could define a fourth normal form using the multi valued dependency, we can define what is known as the fifth normal form using this join dependency, this is called the fifth normal form or 5 NF in a similar manner.

The fifth normal form is also sometimes called project join normal form, because essentially it says that once you joined, once you project it out on this particular attributes and join it you get back the original relation. So, that is called a Project Join Normal Form or PJNF. So, again these are mostly theoretical things and example can be the following.

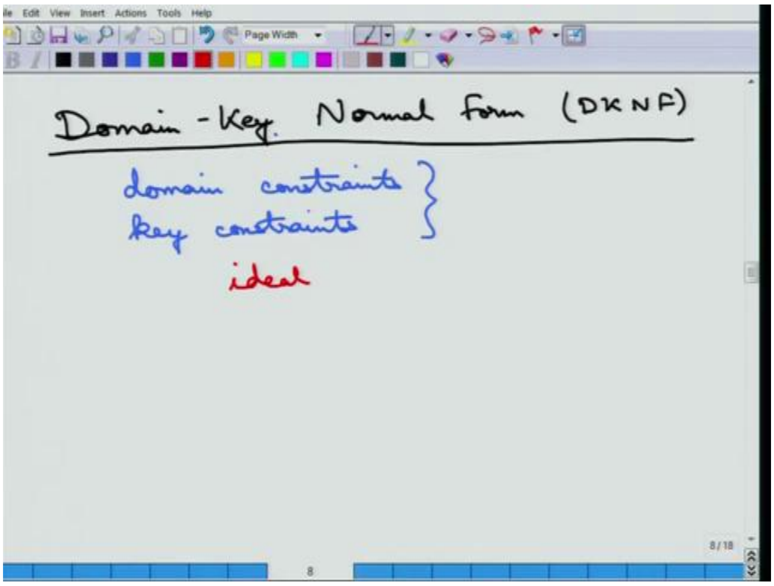
(Refer Slide Time: 14:51)



So, suppose there is a brand and product and a sales man, then it can be probably broken up into two parts at a time. So, this can be taken or this can be taken or this along with this can be taken and then example can be found out and a very quick example may be the following. So, here in this example if this brand product sales man has been broken up into these three relations, brand products, sales man brand and product sales man then one can see that the natural join of all of these tables together gives back the original relation brand product sales man, such that the join dependency is satisfied.

So, this is a much better design than keeping this original design of brand product sales man. Because, for example, the insertion the modification anomalies are minimal here, because now insertion of a particular sales man promotes a particular brand that can be done by using inserting in only one case. For example, if J sales man does it in brand a it is just needs to be inserted here instead of multiple insertions in the original table, where it is contains all the three attributes.

(Refer Slide Time: 16:29)



So, that is the project join normal form, the final form of this is called the Domain Key Normal Form or the DKNF, this is considered the highest form of normalization and this is essentially just theoretical concept. And here it says that a relation schema is said to be in DKNF, if all the constraints and relations that should hold can be enforced simply by the domain constraints and the key constraints. So, everything that should hold can be enforced by simply the domain constraints and the key constraints that is it.

So, we do not require any other constraints or any other conditions etcetera. so these two essentially define everything that this relation should hold that is all. So, this is the ideal normal form and this is mostly theoretical and everything after 3 NF or BCNF is mostly theoretical and database practitioner do not attempt to do it, so this is in that is it. So, once a relation is DKNF, because this is ideal there is no anomaly, there is absolutely no anomaly and all the functional dependencies and multi valued dependencies need not be checked anymore. So, this is the ideal case and that ends the topic on normalization. So, essentially given a relation you want to try to normalize it up to 3 NF or BCNF.

Thank you.