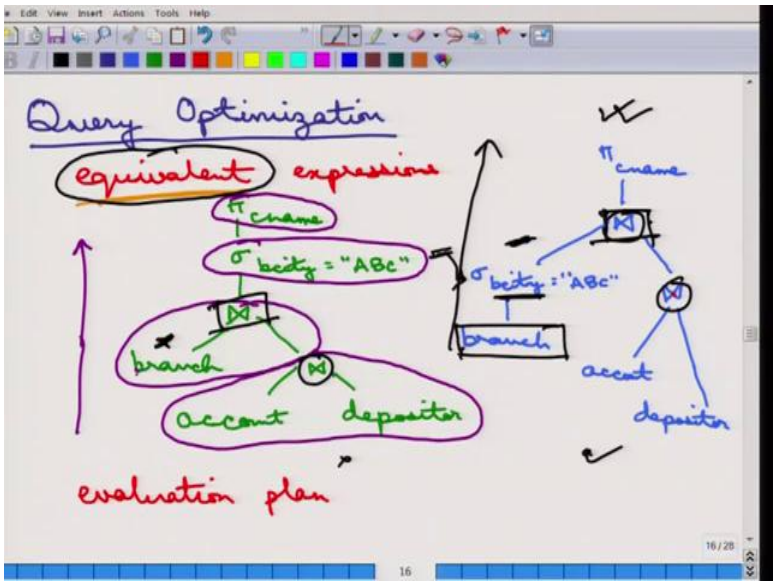
Fundamentals of Database System
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 24
Query Optimization: Equivalent Expressions and Simple Equivalence Rules

(Refer Slide Time: 00:15)



Welcome, we will start on the next topic which is on Query Optimization. So, what do you mean by query optimization is that, we have previously seen the different query processing algorithms and for a particular query, different such algorithms can be applied. So, the question is, which algorithm to apply and that is the topic of discussion for this query optimization. Now, first of all we must understand why there can be different ways of evaluating a query, so that is called equivalent expressions.

So, there can be multiple expressions in duration of algebra, which are equivalent. And I am going to give an example right away, but equivalent, the word equivalent means that the moment what the input is, the two expressions that are considered to be equivalently each other. For the same input they must result in the same output, no matter what the input is, it is not for only a specific particular type of input, but for all possible inputs and here is an example of two equivalent expressions, let us see.

So, first of all suppose there is the account; that is joined, a natural joint with depositor. Now, this the result of this is then, the natural joint with branch, the result of that is then

applied through a sigma function, which is say branch city is equal to your A B C it does not some expressions and finally, this is being by divide all the customer name. So, what does this expression wants to, what does this expression tries to find, it tries to find out the customer names, names of all customers, who has an account in a branch, which is located in the city A B C.

So, that is what the idea of doing this, now this is one way of evaluating it. Now, what does the equivalent expression mean is that when such an expression is being shown, this is called an expression tree. So, when this is being done, what it says is that first of all an account and depositor, the natural joint of an account and depositor is are formed. The result of that is then natural joint with branch, from the result of that thus the selection of the branch city is being done and finally, this projection is being done.

So, the competition proceeds or the order proceeds from the leaf to the top; that is what is being done. Now, there is an equivalent expression which can be done in the following manner, now let us try to write it down. So, suppose there is the branch, on which the sigma B city is done and that is then, joined with your natural join of an account and depositor and finally, the customer name is being projected out. So, first of all, why have this equivalent the reason is, So what have it done here?

So, first of all is that this, the branch city expression the selection on the branch city expression that has been brought earlier than the joint. So, this is mean, what earlier than the joint. So, this evaluation plan goes from here to here, so this is join before the join is join, why is the equivalent is that, the branch city attribute is present only in the branch. So, it does not matter whether the join is joined before or afterwards as far as the correctly is constant.

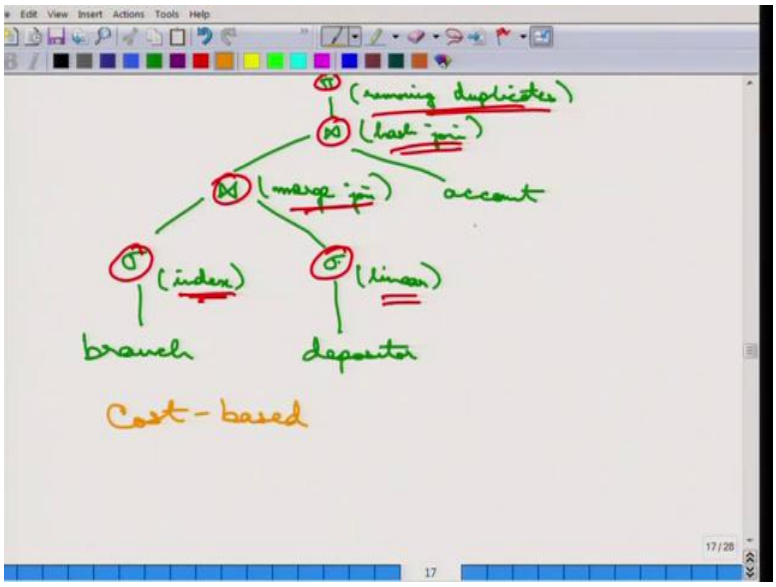
But, one can see, what is the effect of doing this, the output at the end of this selection the output; that is produced here is much lesser, than the output that is produced here. So, this joint is the much faster joint much more efficient joint than this joint, because this is joined with inter branch table this part of course, the same as this, this is joined with the inter branch table while as this is joined with only the branches; that is in the city of A B C, so this is going to be much faster.

So, the whole idea of this query optimization is to figure out all this equivalent expressions and when out of those equivalent expressions once choose the one; that is

suppose ((Refer Time: 04:52)) faster; that is suppose ((Refer Time: 04:54)) waiter. Now, between these two one can clearly see that this is going to be a faster or at least as far as this think. So, this is going to be the database engine will actually evaluate the query in that this manner and not in this manner, so this is the one that is false.

Now, this is the equivalent expression, then there is something called an evaluation plan the evaluation plan in addition to this equivalent expression also says, which algorithm is going to be use for each other operation. So, for example, this is the natural joint and there has been, so there are five join algorithms, that can be applied to do each of these, it can be applied to do this joint. So, which algorithm is going to be done, so; that is what is called an evaluation plan. So, I am expression of the evaluation plan we will be the followings.

(Refer Slide Time: 05:51)



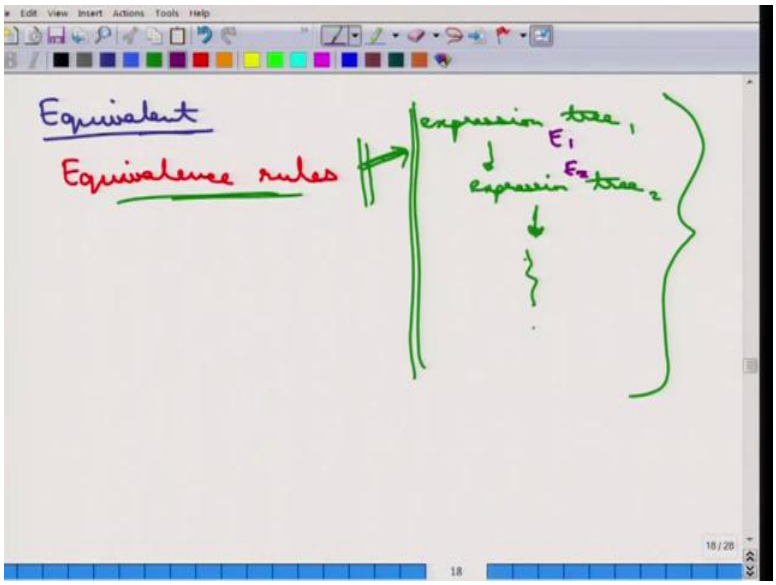
So, let we again write down one example, so suppose this is branch and then, you say this is some sigma, which is being done this sigma is done using an index. Then; that is joint and this is done with let us say there is another relation depositor from, which some another sigma is done this sigma is let us say done using a linear scan. Then, this joint is let us say the merge joint and then, there is another joint; that is being done, which is let us say the hash joint, because this is done with let us say the account or something.

And finally, there is the projection taken, which is simply the moving duplicates etcetera, because this new duplicates can be done. So, this is the complete evaluation plan,

because in each of these operations the corresponding algorithm is also mentioned. So, that is, what an evaluation plan is it is a little more elaborate than just an equivalent expression. So, the optimization plan is to do a cost based optimization plan, so each of this evaluation plans the estimate of the cost is being first made and the evaluation plan that results in the least cost estimate is being execute.

Now, note that this is just an estimate, so what may actually happen is there many another evaluation plan, which will result in a better runtime for a particular query, but the estimates were unable to say that and the database engine just follows, what the best estimate is being given.

(Refer Slide Time: 07:43)

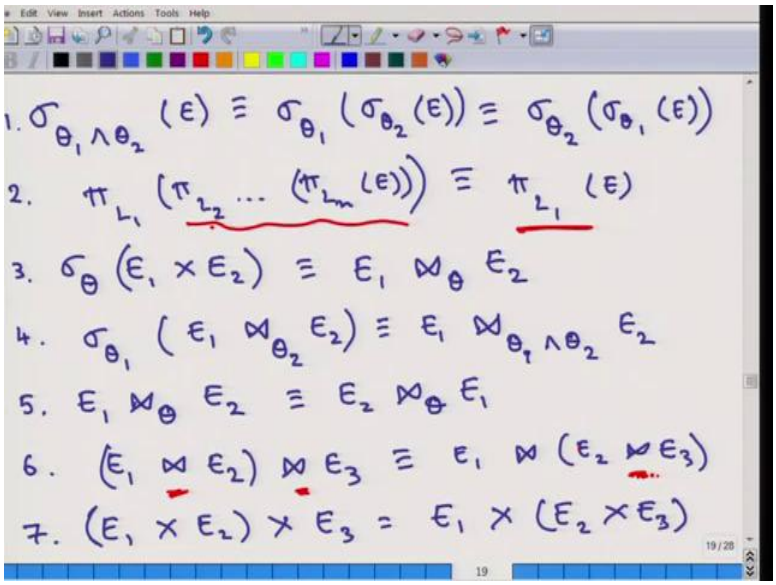


So, coming back to the point of equivalent expressions, let us go over this when can two expressions, we said equivalent as I said two expressions to relationally algebra expressions are equivalent, if they generate same set of output for same set of input. And this is been specified by a set of equivalence rules, so we will specify many such rules their and this specifies, which expressions are equivalent. And then, once expression tree is being generated, so from one expression tree, let us say this is expression tree one. each of these equivalence rules is applied.

So, there is some expression inside this expression an equivalence expression rule is being applied to generate another expression tree and this keeps on going till all the expressions and all the equivalent expressions have been exhaust. So, this is the total set

of all the expression trees that are generated. And by the generation mechanism these are all going to be equivalent expression tree is, because from one tree to the another only one expression is changed and the equivalence rules says that the expression one in this tree is equivalent to the expression two in these trees, so; that means these two trees are equivalent, so let us, now go over, what the equivalence rules are.

(Refer Slide Time: 09:12)



The first equivalence rule is the following is that, if this is being done. So, if the sigma contains two conditions theta 1 and theta 2; that is same as applying the conditions that same as doing the two sigma's one after another, this is easy to understand and this is also the same as I mean changing the order of doing these two sigma's, because and it does not matter, so this is the first equivalence rule. The second equivalence rule is, if there is the series of projections suppose there is series of projections finally, there is an L n of E, this is same as just doing the last projection.

Now, of course, for these two happen it must be that L 1 is the subset of L 2 and L 2 is the subset of L n all those things. But, when all these projections it does not make any sense, so one can simply do the final projection. That makes equivalent rule is suppose there is the E 1 Cartesian product of E 2 and then, there is the sigma place a condition predicate apply to it.

Now, this the essentially the definition of the joint, so this is simply the join of E 1 with E 2 and after the Cartesian product is done, if the sigma if the selection is done; that is

the same the joint, using this join something more can be done. So, suppose there is the and there is the joint condition is theta 2 and then, there is the selection of theta 1 is being done and; that is same as doing the join itself on theta 1 and theta 2.

Now, whenever I am writing down these equivalent expressions all of them can be proved, but you can at least intuitively follow that these are going to be correct and the proofs if you want you can try. And next one is that the how do you write the joint whether even in the left or even it is in the left it does not matter, so it is simply a, that is a easier one to understand. Then, there is something interesting, so this is the joint and that is finally, joined with E 3 this is equivalent to first doing the join of E 2 and E 3 and then, doing the join and even.

Now, there are some assumption inside this is that E 2 and E 3 can be, so these are natural joints by the way, so there is the there is the assumption E 2 and E 3 do share attributes, so that the natural join can be handled. So, and similarly E 1 and E 2 can also E 2 is essentially the central realization that here attributes with both E 1 and E 3 . So, it does not matter, which natural join is first done.

So, E 2 can go with even one first and then, with E 3 or it can go with E 3 first and then, with 1. In a very similar rule is for the Cartesian product, which is provably easier to understand, but let me just write it down for completeness, again it is the same idea that it does not matter, which Cartesian product is done first.