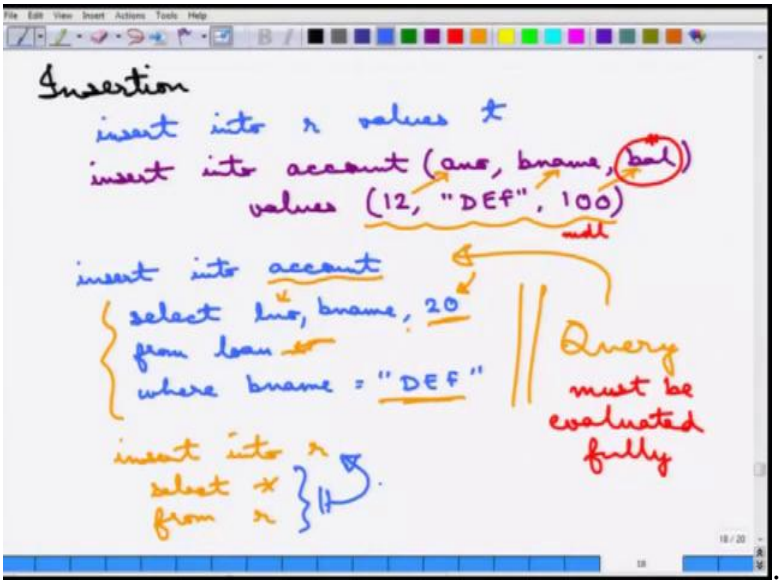
Fundamentals of Database Systems
Prof. Arnab Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 11
SQL: Updates, Joins, Views and Triggers

(Refer Slide Time: 00:13)



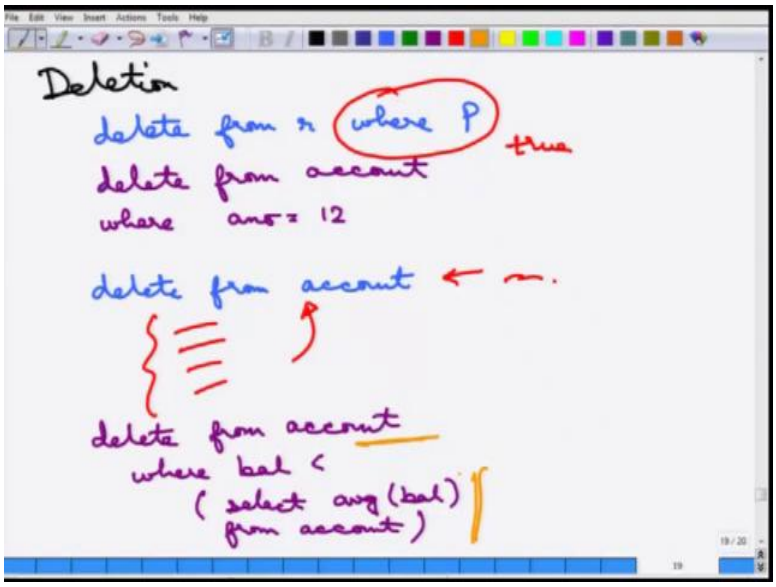
The first thing we will cover is the insertion. The insertion, the syntax is simply insert into the particular relation name, let us say r values, which forms of the values, the tuples that is formed. So, an example is we can say essentially this means 12 is an account number, DEF is the branch name and 100 is the balance, so this tuple gets inserted into the account table.

If the schema is obvious; that means, if the schema of the account is actually a number, bname and balance, then this can be omitted, but it can be ((Refer Time: 00:55)). And instead of 100 etcetera, a particular value is not known and null can be also inserted, provided balance is null able. I mean, if the condition you remember that while we were setting up the table name, we can define certain attributes as whether they can allow null values or not, if they allow null values, then this is fine otherwise they will be erased. Now, one thing is that insertion can also be, so the value of a query can also be used to insert into a table.

So, for example, this can be done, what essentially does is that it creates a new account with the balance 20, so the balance is always set to 20 at this branch DEF. For every loan, wherever there is a loan it creates an account and the account number is said to the same as the loan number. So, this is way of inserting as part of that, so this is essentially a query, which is first solved, then all the values are inserted into the account.

Now, the one thing is that the query must be evaluated fully before the insertion starts, as otherwise there can happen infinite insertions can happen. So, very easy example is the following, so suppose you want to duplicate a relation, first of all whatever it is, this is not very clear why, but suppose you want to do that, so you can write as queries simply insert into r select star from r. Now, imagine what will happen, if this is not being done, so this must be evaluated first, so and then all the results are inserted into it. So, this is only returns 1, so this essentially duplicates the relation that it is, I mean duplicates the tuples in the relation nothing.

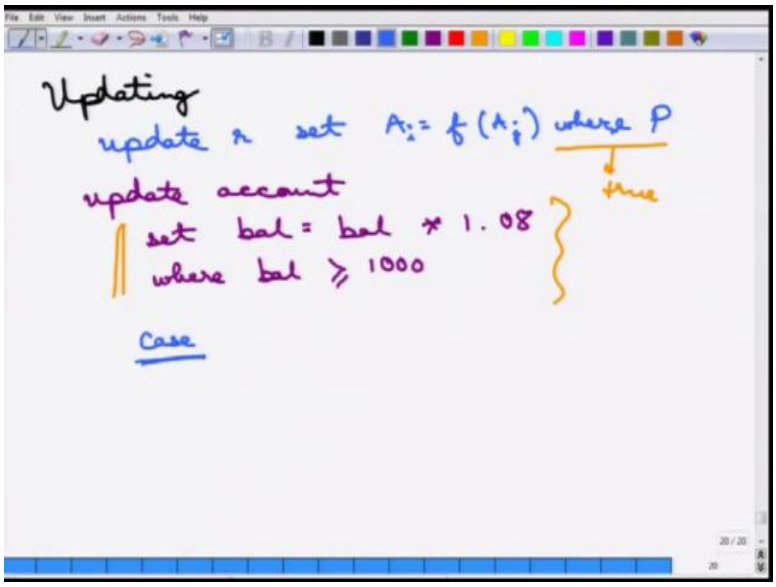
(Refer Slide Time: 02:39)



So, then, we can move on to deletion; the syntax is delete from relation, where there is a predicate, so you delete from the relation r, where the predicate is correct. So, example is, so essentially the account number 12 is deleted and if the where is empty, so if we simply say delete from account, the same rule is applied if the where part is empty, then it means true, which means every tuple will be deleted from account. So, it is at the end of this, it returns an empty relation nothing else is.

And just like insertion, delete can also you can use a query as part of this delete thing. So, only where the query is evaluated and only those tuples that satisfy the query can be r delete and that interesting example in this space is the following. Once more, so what it tries to do is, it deletes all accounts with the balance is less than the average, now again once more the balance must the average must be completed first and then, the deletion happens, otherwise it will keep on deleting; you can see what will happen.

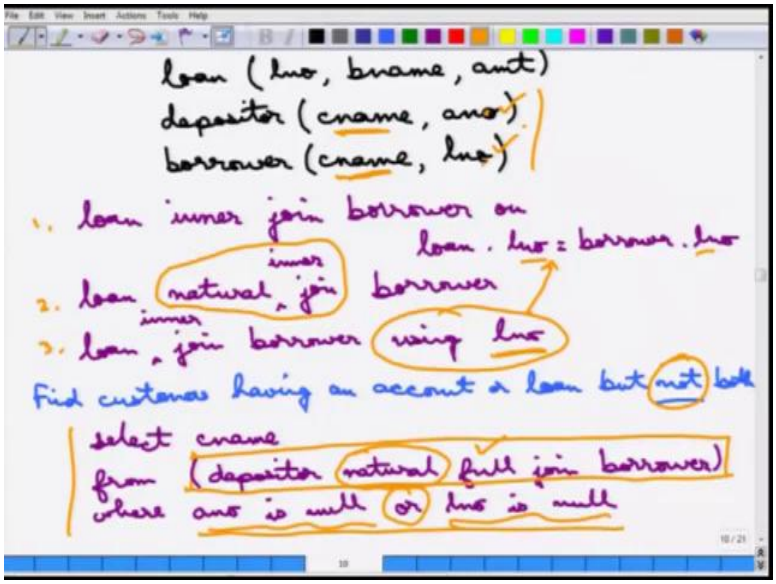
(Refer Slide Time: 03:54)



Finally, there is this updating; the syntax is update relation r set certain values, set attributes least etcetera. So, set attribute least equal to sum function of whatever, some other attribute, so set A i is equal to function of A j, where p, so the predicate. An example is, so what this tries to do is every account, where the balance is greater than some amount 1000 or 8 percent interest is given, so the balance is essentially incremented by 8 percent, so this can be done.

Once more, where if the where is empty then it evaluates to true so; that means, everything will be done and so on and so forth, there is the same issue as follow. And again it depends on how you are doing, so again this can be a query of which is first evaluated and then, this thing is done and there can be case for, that there is a case clause that can be which is essentially equal to the switch case kind of thing and there we can see the syntax later on. So, this does the basic queries and the database modifications are complete, we will next do another very important topic next.

(Refer Slide Time: 05:21)



So, we will next cover the join in SQL, so join as we saw there are different types of join inner join, then there is a left join, which essentially means it is a left outer join, right join, full join, natural join, this fall in one group. The natural join falls in another group, a particular join can also we set can be set on a particular predicate and then, it can be also set using which attribute. So, the on is on the predicate and using is the attribute, so these are the different ways one can join.

So, for example, we can say loan, inner join and borrower on you can specify the condition, which is loan dot loan number is equal to borrower dot loan number. Now this is equivalent to saying this is loan, natural join, borrower, this is again equivalent to saying loan, join, borrower. I am sorry, this should be natural inner join and this is loan inner join using l number. So, all these three queries are equivalent, that there are different ways of stating.

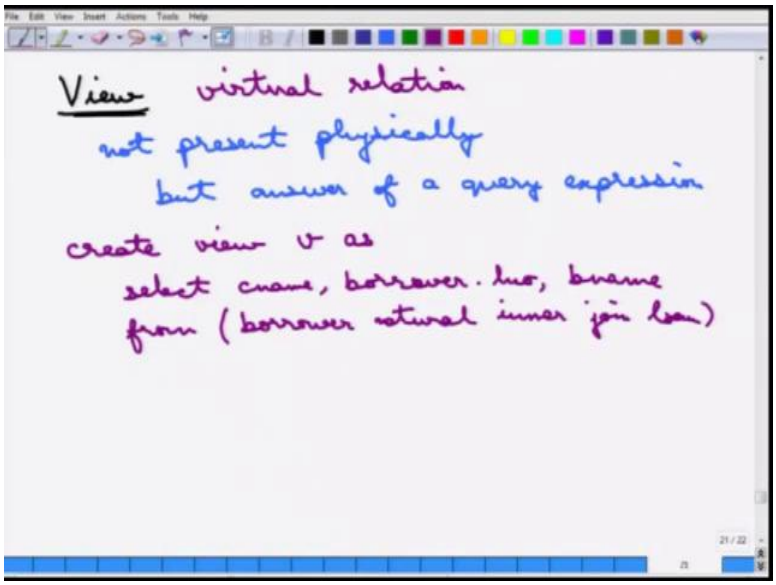
So, the natural inner join is the same as the inner join on that number, because it is the same attribute value, so it is essentially, if you say using real number this essentially transfers to this condition and it reaches the same and it is writing it down explicitly on and this is the natural join as well. So, an example of join may be, so following can be one way of solving it. Now, do remember there are different ways in which a particular query can be solved or there are different equivalent SQL statements that can be written that will solve the same query.

So, here is one way of doing it from this is, what I am writing it down. So, depositor natural full join borrower, where a number is null or l number is null, now this requires a little bit of thought. So, what it is being done is the following is that, first of all this depositor natural full join borrower this will create every possible combination of depositor and borrower, so it will find out all customers, now this is a natural full join.

So, when you say natural join the depositor and the borrower table, they agree on the customer name. So, it must have that when you joining the depositor and borrower and they must be the same customer name, so it will find out all for each customer name the account number and the l number. Now, so even if there is an account number, but no corresponding l number, because it is a full join it will output them.

Now, what do we want is that we want those kind of things, where there is either an account number or an l number, so at least one of them is null. So, that is why this is an or clause, so at least one of them. So, if a customer has both account number and l number, then both will be valid, so this will be false, so that is not going to be output. So, this is the way to solve find customers having an account or loan, but not both. So, this completes the part about SQL etcetera and, so there are some more constructs in SQL some little bit of constructs that we will cover next.

(Refer Slide Time: 09:55)

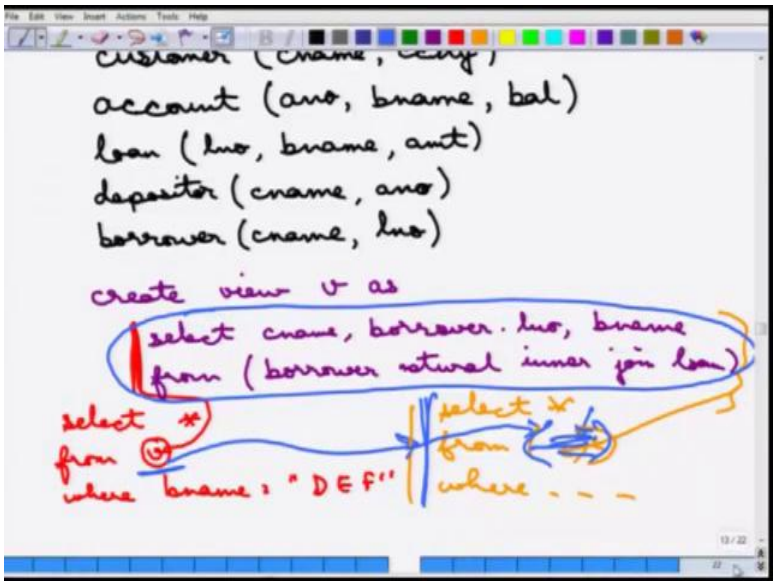


So, some important other constructs of SQL the first one is the view. So, of view is the answer of the query; that is not present physically, so this is not present physically, but

answer of a query expression. So, for example, essentially a view is a view can be considered as a virtual relation an example. So, why a view is needed view helps in query processing an example may be the following is that, suppose this is being done.

So, create view, so this is the syntax to create a view v as if you say. So, what does this view tries to do is find all the loans of the customers, but, but it will does not bother about the loan amount. So, if we go to this example, so so this is the view, so this view can be later used anywhere, where this is use useful, so if we do the following things.

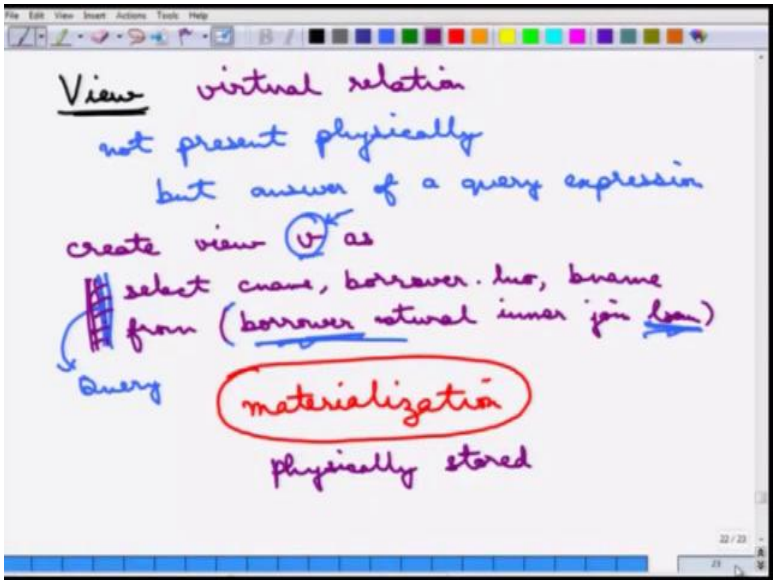
(Refer Slide Time: 11:16)



So, suppose select star from v, where branch name is equal to D E F. So, you want the names of all customers who have an account at the D E F branch well that is it. So, this view v is essentially this the part of this query, so this is essentially the same as writing select star from, then this entire thing will go here this is here, where the same clause is being done. So, important thing to notice, that how a view stored the view as I said is not stored physically.

So, what is being stored is the query expression is being stored, so what is stored is this following query expression. So, very well v is used it is actually replaced by the query equivalent query expression, so this is the query expression. So, it is actually being replaced by the query expression and at run time, this query is evaluated and the entire answer is resulting. So, why is such a case done.

(Refer Slide Time: 12:34)

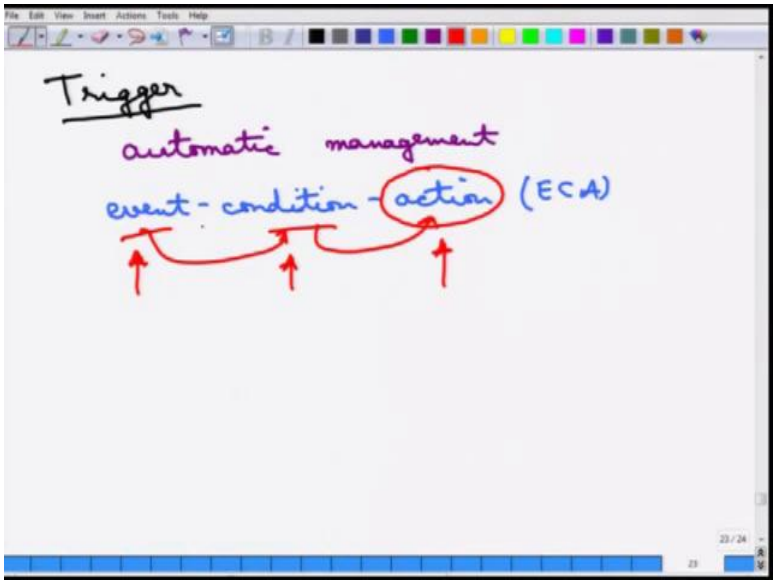


The reason is when the view is used the relation that it uses the view from we have changed. So, for example, if we go back to the query that we see this create view v uses the following tables borrower and loan. Now, if you store this as a table if borrower and loan has changed, there is no way the view v will know that change, on the other end, if only the query expression is stored, then it is fine. Because, if borrower and loan changes it does not matter this is evaluated again at query time at run time, so nothing else is problem.

So, that is why there are restrictions to, what can be done, which kind of views can be done whether a view can be updated etcetera. A view may not be updated because updating a view essentially means updating the borrower and loan tables right, which is not clear how to update or it may get into the problems of null tuples, null values, and all those things.

For some views there is a term called materialization, so some views are materialized. So, this depends on the query engine etcetera. The database engine, when some views are materialized it essentially means the view is stored physically is physically stored. Now, why will that be done when the view is simple enough etcetera, when the view is used in multiple queries, then it makes sense to materialize a view, because then, this following query is going to be evaluated at run time etcetera. So, that is the view materialization issue and otherwise, a view is not very much updatable etcetera fine.

(Refer Slide Time: 14:20)



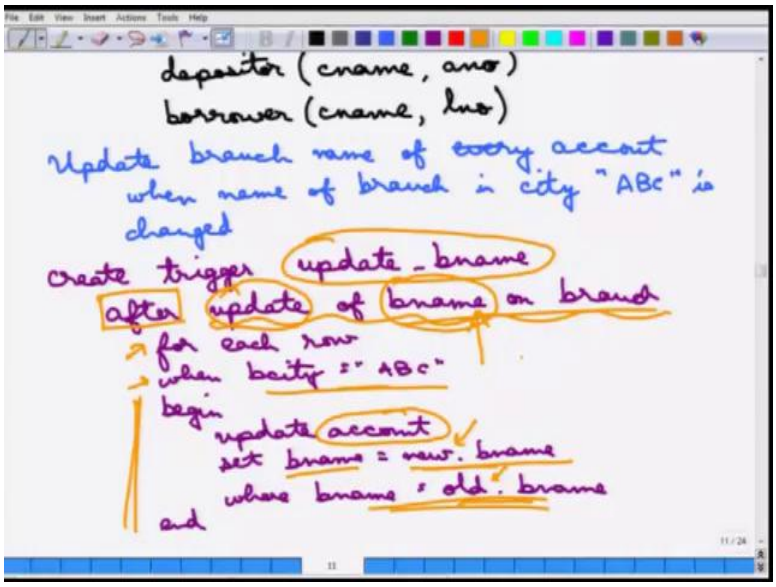
So, the last topic, that we will cover in this SQL is something called a trigger. So, at trigger, so let me write it down what a trigger is at trigger statement allows automatic management of database stuffs. So, there is a, so it is a automatic, so whenever some action takes place in one relation automatically some other statements take place and it triggers essentially it triggers a couple of things.

For example whenever a grade is submitted for example, in the in the example of student databases whenever a grade for a particular course is submitted the cgpi of the student is automatically recalculated and; that is being stored, so that can be written as a trigger. So, whenever a new grade is inserted, there is a trigger, that will automatically recomputed the cgpi. So, it essentially follows, what is called and event condition action models, so this is a easier models.

So, whenever an event happens it checks for certain condition, if that is true the corresponding action is being taken, so the event is essentially a database modification. So, such as an insertion updates or deletion etcetera, the condition is a predicate and the action is any other database action or some even external programs can be done etcetera. And the action can be done either before or after the event, so this is a modification event.

So, the action can be specified as a before action or a after action. So, generally this is a after action, but it can be also specified as a before action for example, we can think of the following query.

(Refer Slide Time: 16:01)



So, create trigger, so let us say it creates a trigger you can say update the name of the trigger you can give something let us say update b name this is a trigger after. So, this says that after that this is a after model, so when the event has taken place after that after update of branch name on branch. So, then you can say how this update will be done for, so what is the action that can be taken.

So, for each row when b city is equal to ABC you say you have the begin, there is a begin and end statement update account in the following manner set b name is equal to new dot b name, where b name is equal to old dot b name this is an end. So, this example covers a lot of issues, so let me go over them 1 by 1. So, it first says create a trigger the name of trigger is update b name fine this says after so; that means, an after event after, what is the event update of branch name on branch.

So, whenever there is an update of branch name on branch this trigger is essentially invoked. Now, how which is done, so that the trigger is applied for each row only when the branch city is ABC this is the condition on the trigger. So, it is not for everything on the branch, then what is the action that is being taken is that the account table the account

relation is updated in the following manner. The branch name is set to the new branch name for those things where it was the old branch name.

So, this new and old are essentially special marker, so which says in the, because it is an update event, so this the new branch name is got from here and the old branch name is got from whatever was ready in the database. So, this is the example of how to work with the trigger, so this completes a part of SQL. So, we have covered about all the basic operators we have covered nested sub queries we have covered, the updates the deletion insertion etcetera and some special issues in SQL, which are the views and the triggers.