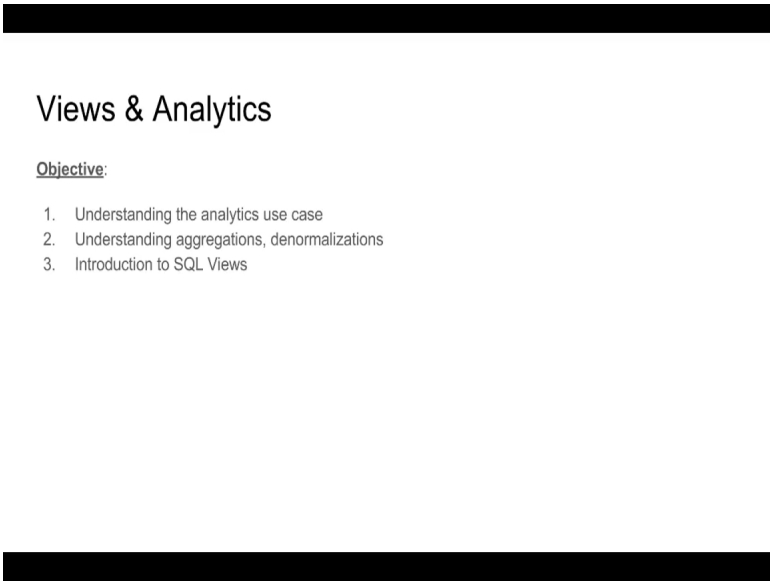
Introduction to Morden Application Development

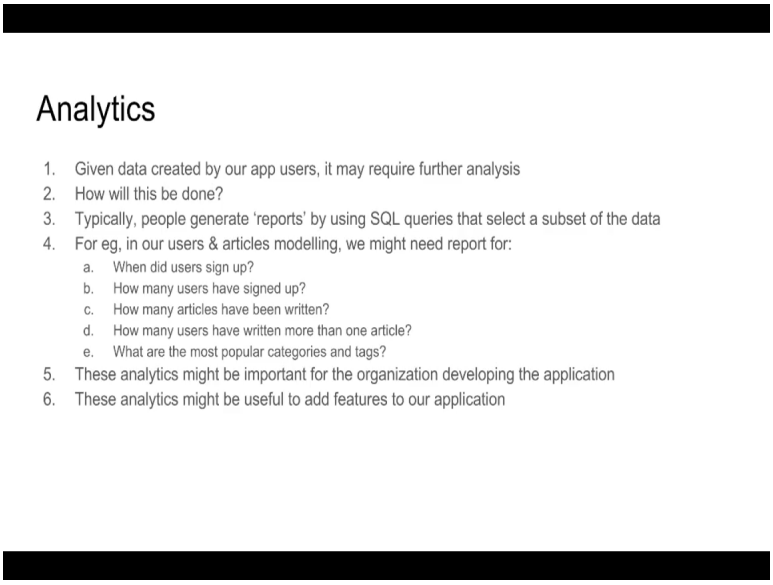
Dr. Gaurav Raina
Prof. Tanmai Gopal
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 11
Lecture – 20
Analytics and Views

(Refer Slide Time: 00:06)



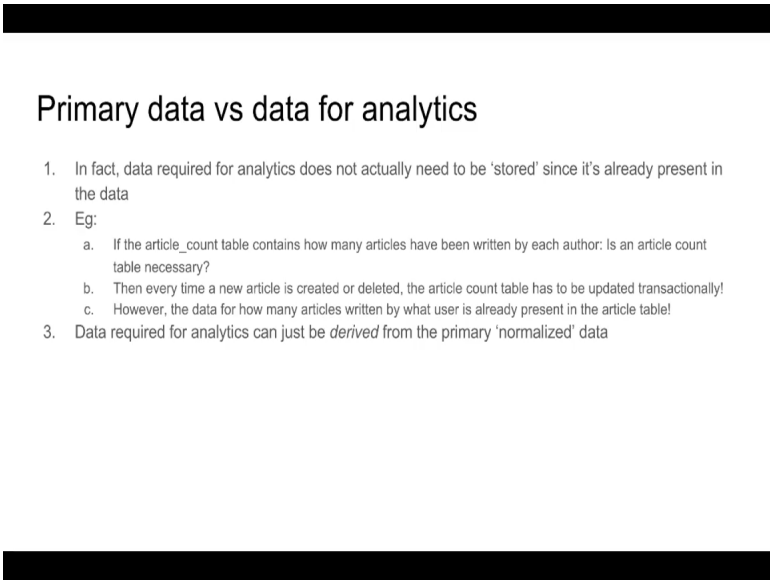
(Refer Slide Time: 00:08)



Hi guys, welcome to module 11, in this module we are going to talk about analytics and the concept of views. So, let us try to understand what you mean by analytics. Data is created as our application is used by an application or by the uses of our application and this data might require further analysis. So, just to give you an example in the application that we created where you are looking at users and articles we might need reports for how many uses are signed up when did this user signup, how many articles have been written, what the most popular categories and tags are.

So, this kind of information on obtaining this kind of information is falls under the purview of what we called analytics. Analytics might be important for the organisation for doing for the development and the application to understand what features are being used and what features are being desired by users it might be important for the organisation from a business point of view analytics is also useful when we need to add some features for applications. For example, if you want to show a list of the popular categories and tags on the homepage doing the analytics to find out for the popular categories and tags would be very useful.

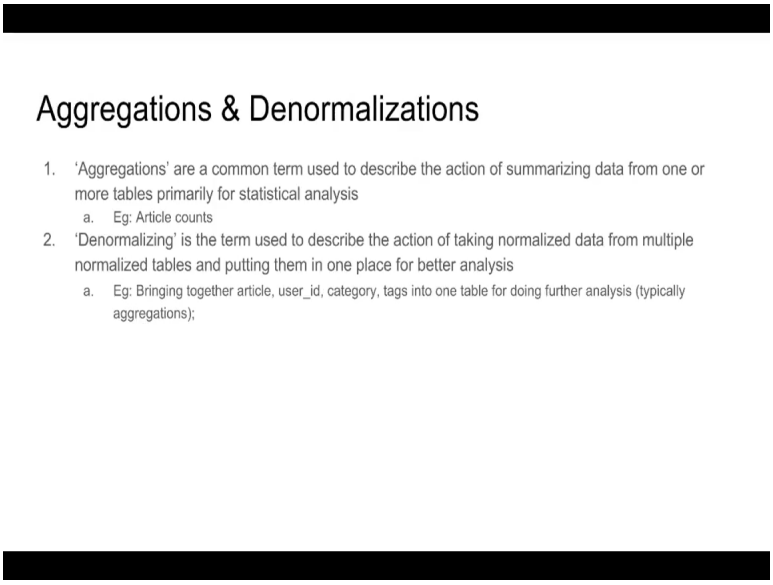
(Refer Slide Time: 01:03)



So, far the data models that we had can store the user information and the article information, but there are no data models that we have in our database that allows to fetch analytics information for example, if you want to get the information for how many articles have been written by each author we do not really have a table in a database that is something like the article count table that contains 2 columns called author id and article count. If he had that table then you will be able to query that table we would be able to insert data into that table to how many articles have been written by an author and we will be able to query that table whenever we wanted analytics for seeing how many articles have been written by an author. But if we think about it this is a redundant information right the article count table is actually just a kind of a duplicate or a summary of the information that is already there in the article table.

And that is what we are going to understand in this module the fact that the data that is required for analytics can just be derived from the primary normalised data that we were storing in our data models.

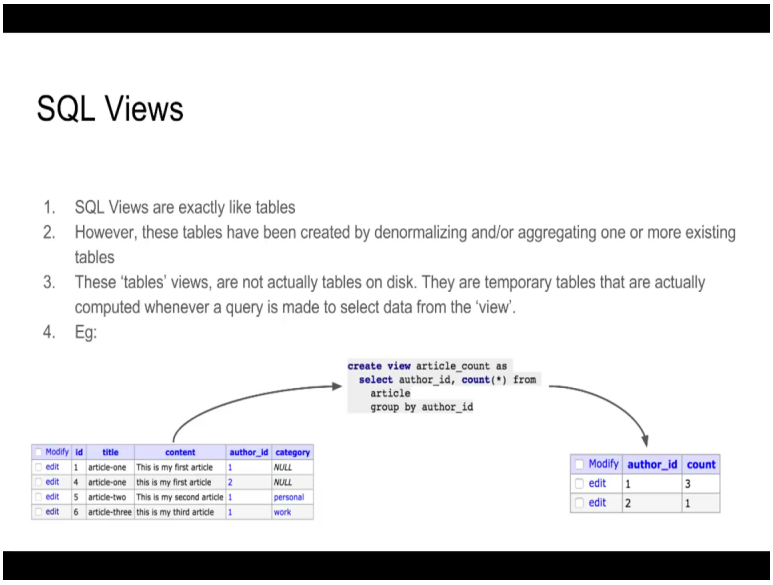
(Refer Slide Time: 02:02)



So, let us look at 2 ways of obtaining analytics information from the primary data models that we already have a 2 class actions that fall into creating information that is ready for analytics is called aggregation and de-normalization. So, an aggregation is when we take existing data and we summarise it in a way that is ideal for further statistical analysis. So, for example, we can aggregate the article table to create the article under article underscore count table right because the article table contains the primary information and doing some kind of a summarisation or aggregating the data from the table can give us exactly the information that we need. De-normalizing is the process of taking information that spans multiple tables putting them together into one table this act of putting them together in one table is not necessary useful by itself, but it is useful especially when further aggregations need to be done.

So, for example, we can take the article category and tags which were three separate tables we can merge it into one table and then after merging it into one table we can apply an aggregation on it to find out let us say what the most popular tags that an author uses are. So, this process of bringing data together into one table is called de-normalizing.

(Refer Slide Time: 03:23)



So, let us look at an example of what we have been talking about. Here we have the article table which contains the id title content author id and category column what we want out of this table is that you want the information for which author has written how many articles. So, this is target information that we want on the right. We want the author id and we want to know count of the articles that each author id has written. So, we can see that we want author id one first three articles author id 2 was one article. This process of getting to this data is called an aggregation. So, we are summarising the content of this table and we are getting this information we have already explored that we can get this information by doing a select query and the select query we have tried out in one of the previous modules.

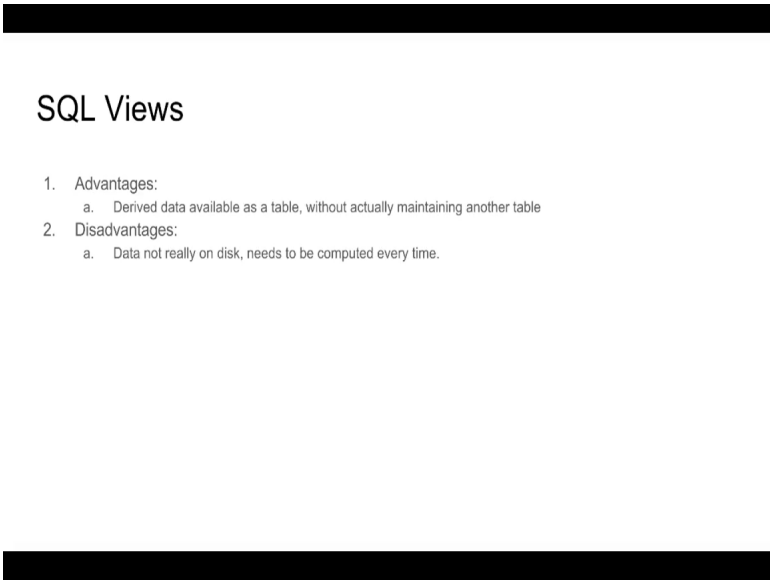
What we have now done is that we have created a view out of the select query so; that means, not now there is suddenly a new kind of table that has been created which is an aggregation or a summary of the table that it came from. So, SQL views are table like entities that have been created by performing aggregations or de-normalizations or may be just simple select queries from a table or from a set of existing tables. It is important to note that this table like entity that has been created by a view is not is not actually a table this data is not actually stored on disk if we everyone to query rows from this view. The entire select query will actually be computed to get the rows from this table like

entity and because this table like entity is actually temporary it is not actually stored on disk this is called a view why do we want to create another view and not another table because we want this table to derive all of it is data from this table; that means, we would always like to write update insert into this table right and as soon as we update this table we would want to read from this view to get the information that is required for our analytics right.

(Refer Slide Time: 05:16)

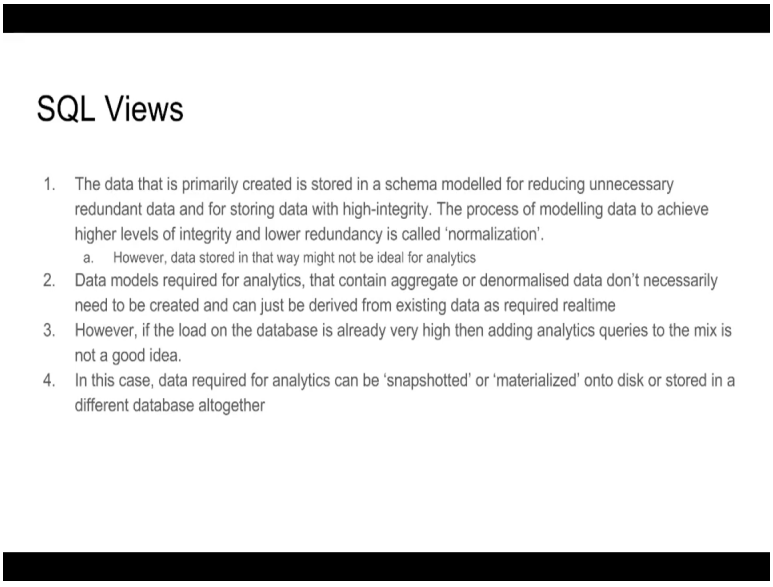
Which is why this is not actually a new table if you make an update to our table right where we added a new row for example, article id 10 has been added which is written by the author id 2. Instantly the view is also modified and so now, if we query data from this view you will notice that the author id 2 now has a count of 2. So, to summarise an SQL view is exactly like the table is technically implemented by the DBMS as just another select query.

(Refer Slide Time: 05:40)



So, the advantages of having uses are quite clear the disadvantage is that because the data is not really on disk it needs to be computed every time and; that means, that if we have very large tables converting them into views to fetch the information that we want might become expensive.

(Refer Slide Time: 05:55)

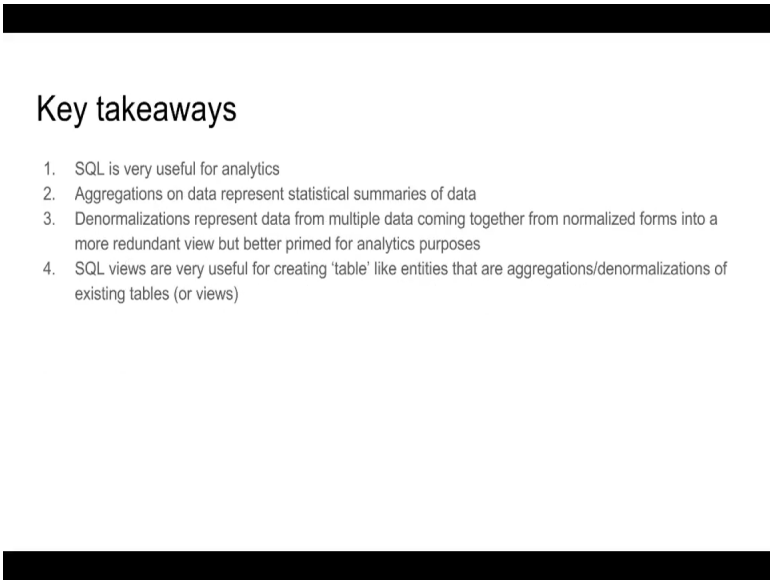


If you think about this in a little more detail when we did the data modelling for actually creating and storing our data our bias was to create a schema or a set of data models that reduce redundancy and it increase the amount of integrity that we have.

This process of trying to achieve higher level of integrity and low amount of redundancy is called normalisation. However, we understand that we can always derive the data that we have from our normalised tables into views that can help us that will make analytics applications much easier; however, if the load in the database is already very high then adding extra analytics query to the mix might not be a very good idea. Example if you have a few tons of millions rows in our article table computing this view every single time might not be very efficient in which case some databases give us the option of snapshotting the view and actually storing it disks that it can be queried. So, this process is called materializing the view and so, we are materializing a portion of our data and storing that on to disk.

The advantage of materializing querying the materialized view becomes very efficient because the data does not have to be computed every single time especially if the computation is expensive, but the disadvantage is that the data is not live anymore; that means, for primary tables are updated then we have to refresh our materialized view to be able to get the latest when we materialize data from an existing set of tables we do not have to store it in the same database we can even store it in a completely different database. So, for example, a very common pattern is to use an SQL database or relational database like post SQL as a primary database because of the support that it has for transactions and for acid properties, but to materialize information from that database into another database like MongoDB or Elasticsearch to enable analytics or search like features.

(Refer Slide Time: 07:41)



The important things to take away from this module are that SQL is a very useful tool for analytics because we can make all sorts of queries to compute exactly the data that we want. We understood that aggregations are a way to represent statistical summaries of the data, the third thing we understood is that de-normalization is the process of getting data from multiple tables and putting that into one table and we did not look at an example of de-normalization, but we understand how de-normalization can help especially in doing further aggregation. We also look at the concept of SQL views and how SQL views can be used to create table like entities that are very useful for analytics. We understood that we can use aggregations and de-normalizations to create SQL views we also touched upon the concept of materialized views.

It is important to understand that several concepts that we have covered in this module are not exclusive to the analytics domain alone for example, aggregations or de-normalizations and materialized views are also concepts that become useful when we talk about read performance. So, the purpose of this module was to just touch upon few basic concepts that will come in handy when we take our first steps with analytics.