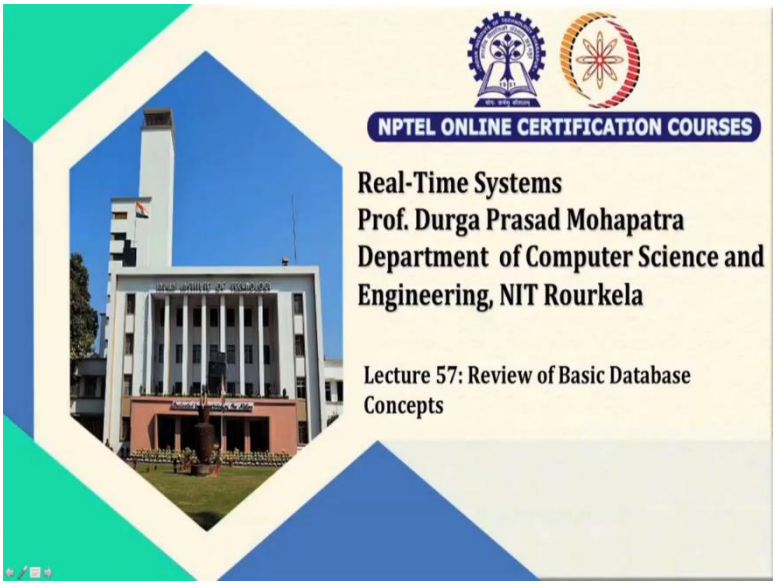
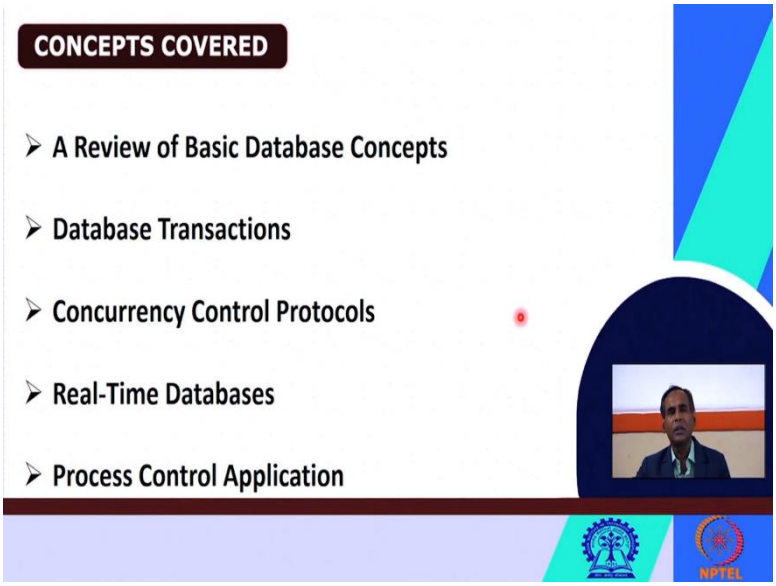
**Real-Time Systems**
**Professor Durga Prasad Mohapatra**
**Department of Computer Science and Engineering**
**National Institute of Technology, Rourkela**
**Lecture 57**
**Review of Basic Database Concepts**

(Refer Slide time: 00:14)




Good afternoon to all of you. Today we will discuss about a new chapter on Real Time Databases. In this lecture, we will discuss the basic concepts or the review of the basic database concepts.

(Refer Slide Time: 00:32)

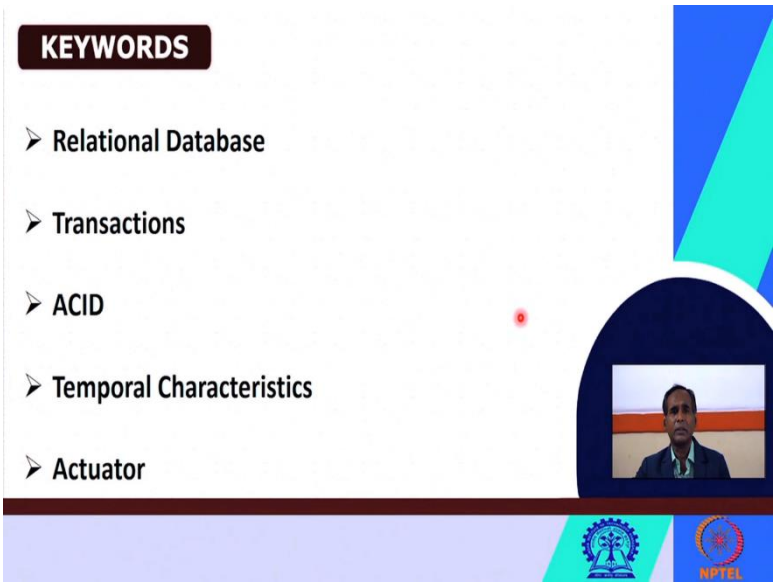



We will first discuss a brief review of basic database concepts, then we will discuss a little bit about database transactions, then concurrency control protocols, what do you mean by real time

databases, how do they differ from the conventional databases and we will take at least one example today that is on process control application.

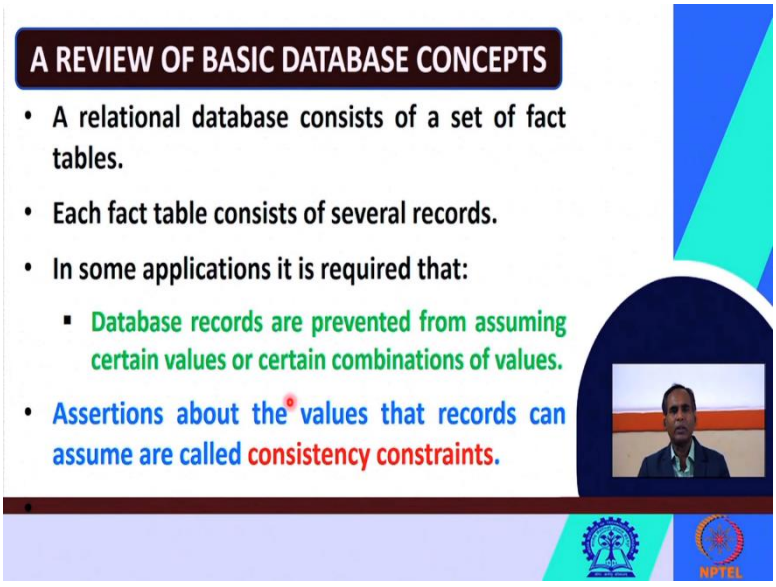
So, for this chapter you require the fundamental concepts of a database. I hope you must have already read that paper in some of the previous semesters. So, those who have forgotten please have a look on the fundamental concepts of database. I will also discuss few of the basic concepts. So, then you will feel comfortable to understand what is real time database.

(Refer Slide Time: 01:13)




These keywords we will use such as relational database, transactions, ACID property, temporal characteristics, actuator, sensor, etc. Those keywords we will use in this lecture.

(Refer Slide Time: 01:26)

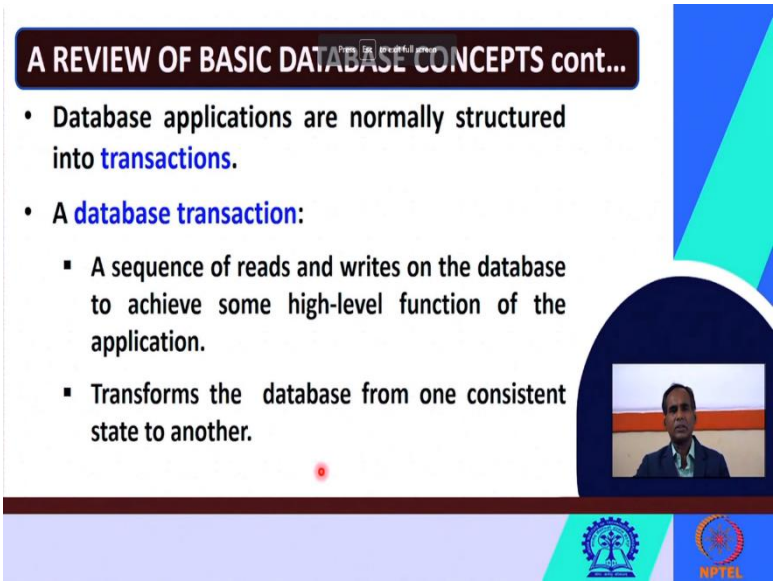


So, let us first start with a brief preview of the basic database concepts. A relational database you already have seen different types of databases such as the relational database and then your network database and these graphical databases those kinds of things. What is the relational database that you have already known?

Here I want to say that a relational database normally it consists of a set of fact tables is it not, a relational database consists of a set of some fact tables where each table consists of several records, the table consists of several records. You might have known that in some of the applications it is required that the database records are prevented from assuming some values or certain values or some or certain combination of values.

We will see some of the particular applications are there where it is required that the database records, they are prevented from assuming certain values they cannot take those values, they cannot assume certain values or certain range of values or certain combination of values. The assertions about which the values that the records in the database, they can assume are called as consistency constraints. So, the assertions are the constraints about the values that the records can assume or can take in a database, we call them as consistency constraints.

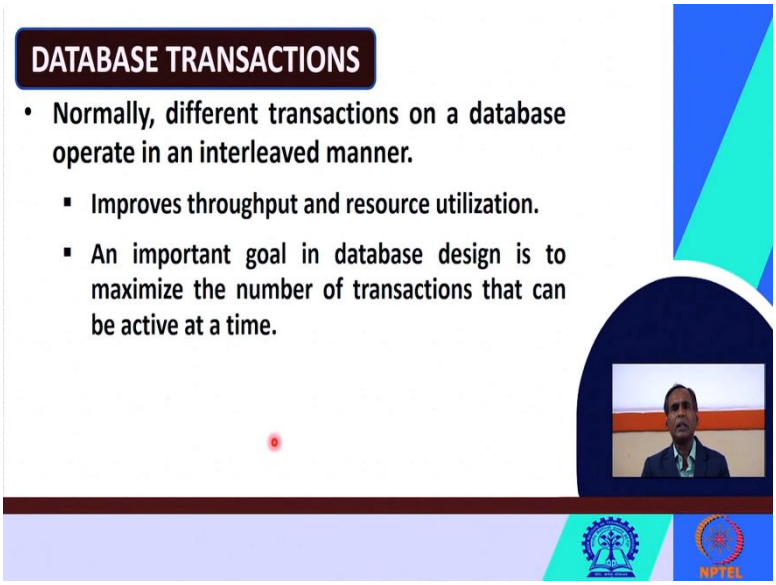
(Refer Slide Time: 02:57)




You know that they that normally the database applications are structured into some transactions, what is a transaction? A database transaction is a sequence of reads and write operations. A database transaction is a sequence of read and write operations on the database in order to achieve some high-level function of the application.

So, database transaction means it is a sequence of read and write operations on the database to achieve some high-level function of the application. So, this database transaction what does it do? The database transaction, it transforms the database from one consistent state to another consistent state. So, a database transaction, it transforms the database from one consistent state to another consistent state.

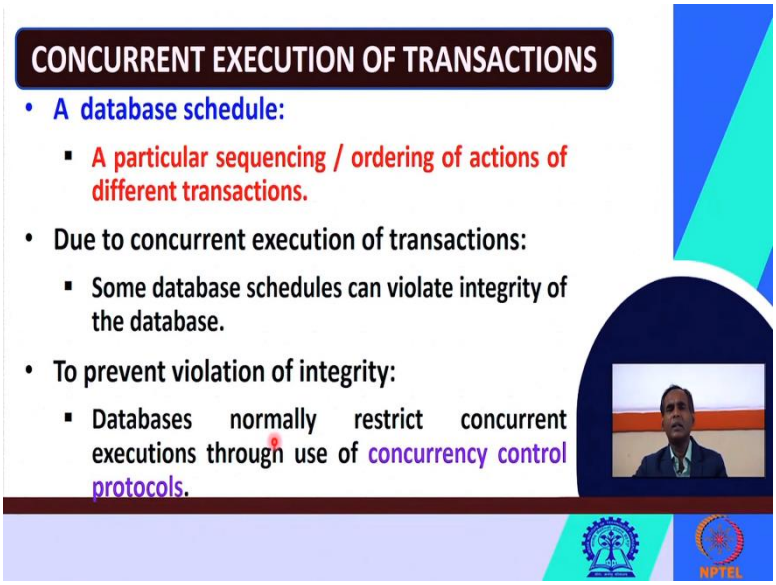
(Refer Slide Time: 03:47)




Normally, the different transactions on a database, they operate in an interleaved manner not in a sequential manner. So normally, these different transactions in a database they operate or they execute in an interleaved manner instead of running in a sequential manner. So, this mechanism improves the throughput on the resource utilization. So, why these data transactions they operate on interleaved manner?

So, since they are operating in an interleaved manner, this improves the throughput and the resource utilization. So, this is an important goal in database design. An important goal in database design is to maximize the number of transactions that can be active at any time. So, what is the number of transactions that has to be maximized which can be active at time, this is an important goal in designing databases.

(Refer Slide Time: 04:41)




Now, let us see about this concurrent execution of transactions, how the transactions can be executed concurrently and what are the problems associated with it. So, before going to the concurrent execution of data transactions, let us see what is your database schedule. I have already told you what is a database transaction. Now let us see what is a database schedule. So, database schedule means it is a particular sequencing of actions or it is a particular ordering of actions of different transaction.

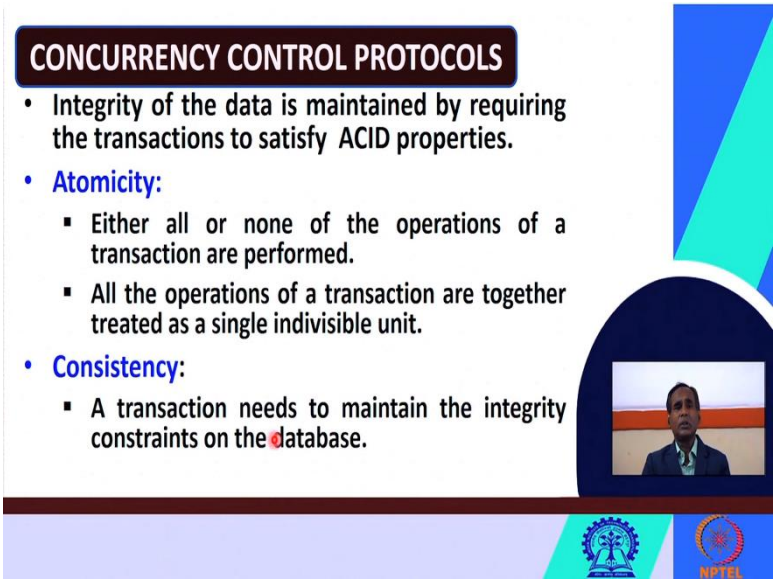
So, what are the different transactions if you can find out a particular sequence or particular order of the different actions of the different transactions, we call it a database schedule. So, database schedule is a particular sequencing or ordering of actions of different transactions. Then let us see what can happen due to concurrent execution of transactions. What can the consequence? Due to the concurrent execution of transactions, some database schedules they can violate the integrity of the database.

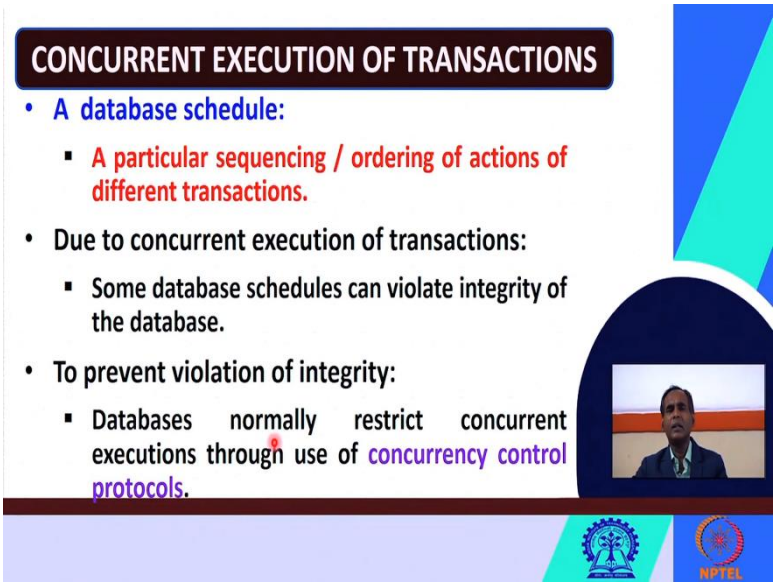
So, due to the concurrent execution of the transactions, the following might happen, some of the database's schedules, they may violate, they can violate the integrity of the database. So, then how can prevent from the violation of integrity? To prevent the violation of integrity, the database normally restricts the concurrent execution through use of some protocols called as large concurrency control protocols. So, to prevent the violation of integrity, the databases they normally restrict, they normally limit the concurrent executions by using some protocols known as concurrency control protocols.

So, concurrency control protocols can be used to restrict the concurrent executions in order to prevent from violation of integrity. So, to prevent the violation of integrity, the database

normally restricts the concurrent executions through use of some protocols known as concurrency control protocols. Towards the latter part of this module, we will discuss the different types of the concurrency control protocols.

(Refer Slide Time: 06:48)



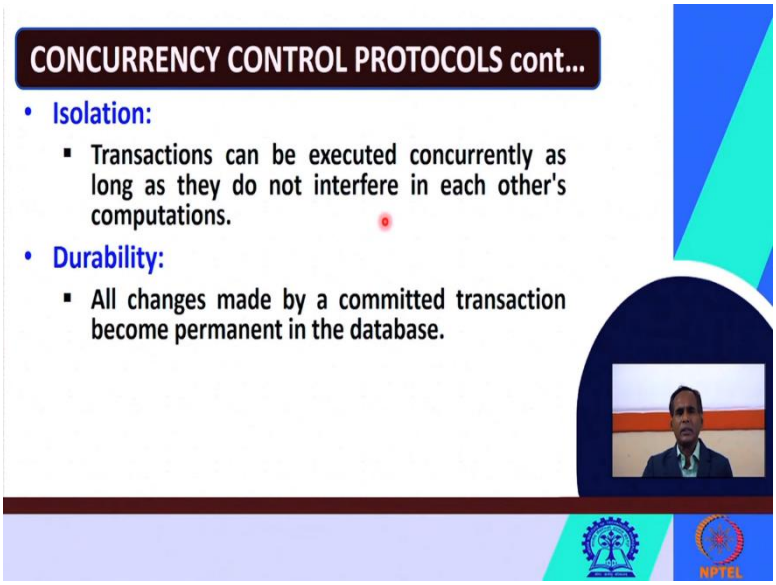

So, the integrity of the data, how it can be maintained? So, the integrity I have already told you, if what, due to concurrent execution of transactions, some of the database schedules, they may violate the integrity of the database. Now, the question is how the integrity of the data can be maintained. So, integrity of the data can be maintained by declaring the transactions to satisfy the ACID properties. So, if the transactions they will satisfy the ACID properties, then the integrity of the data can be maintained. What is ACID property or what are the different, what are the ACID properties that you have already known in your database paper earlier.

So, in ACID property, A stands for atomicity, C stands for consistency, I stands for isolation, D stands for durability. So, let us first see atomicity. Atomicity means what? Atomicity means here either all or none of the operations of the transaction are performed. So, if there is a transaction there are said three operations, either you have to execute all the three operations or you will not execute any of the operations. It cannot be in-between you cannot just execute one or two out of the three operations.

So, atomicity means either all of the operations or none of the operations of the transactions are performed not partially. All the operations of transaction they are together treated as a single individual unit. So, how many operations are there? Maybe three operations or four operations or five operations all the operations of a transaction they are combiningly treated as a single indivisible unit. That is why they name is your atomicity. Next let us see consistency.

What do you mean by consistency? A transaction needs to maintain the integrity constraints on the database. So, what constraints are there? So, transaction it has to maintain those constraints on the database. We call it consistency, a transaction it requires to maintain the integrity constraints on the database, this is known as consistency property.

(Refer Slide Time: 08:45)

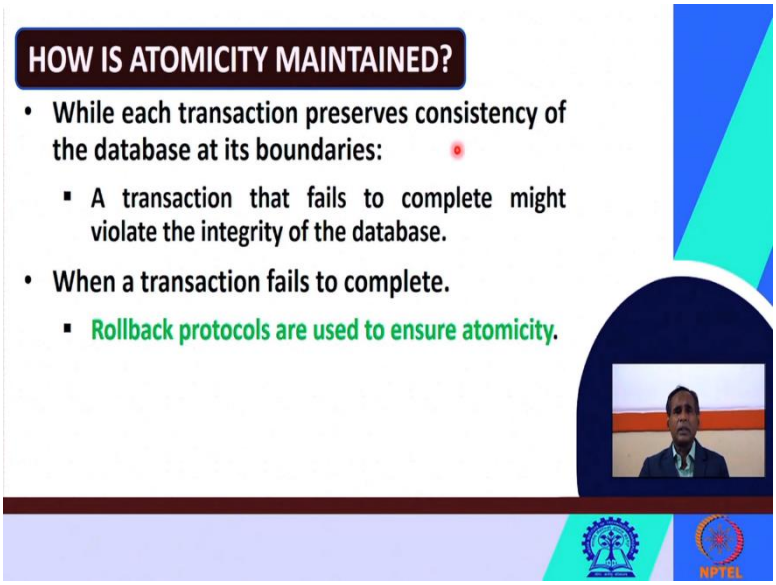


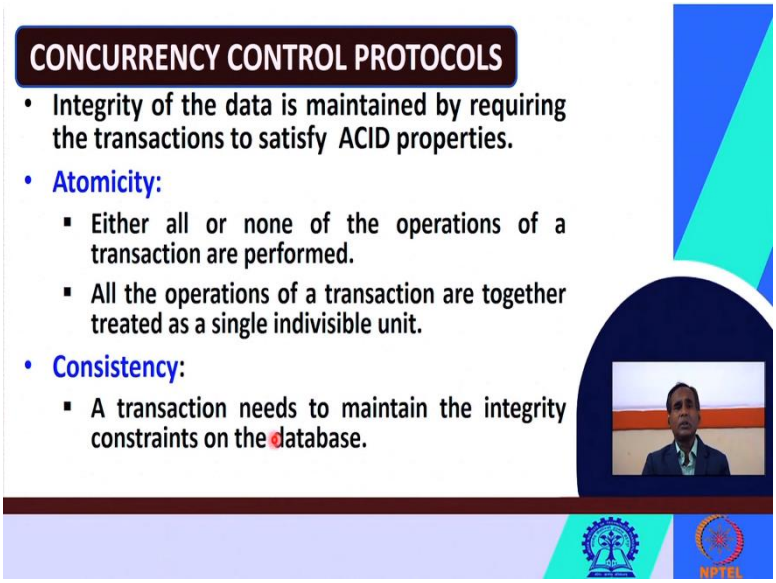

What do you mean under isolation property? The transactions can be executed concurrently as long as they do not interfere in each other's computation. So, if there are say two number of transactions, say a and b, we call them a and b can be adjusted concurrently as long as they do not interfere in each other's computations. Transaction a does not interfere in the computation of transaction b or transaction b does not interpret in each other's computation.

So, as long as the transactions they do not interfere in each other's computations, we say that those transactions are isolation. There they can be executed concurrently. So, the transactions that can be executed concurrently as long as they do not interpret in each other's computations, then we say that the transactions since they do not interfere in each other's computations, they are known as we said that they are in isolation.

Then the last property is durability. So, durability means all changes made by a committed transaction, they become permanent in the database. Durability means the duration. For how much duration they are valid, those kinds of things. So, all the changes that we have made by a committed transaction, those changes they become permanent in a database. So, once you are making the changes by a committed transaction, not about a transaction but a committed transaction then they become permanent in the database. This is known as durability. So, durability represents the fact that the all changes made by a committed transaction, they become permanent in the database.

(Refer Slide Time: 10:21)



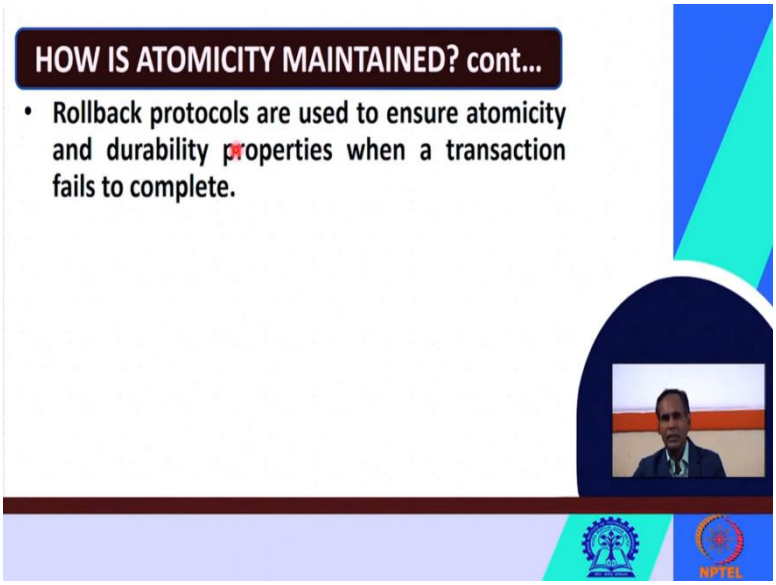



Now let us go one by one. How atomicity is maintained? How isolation is maintained? Those things let us see. So, let us see how atomicity can be maintained. While each transaction preserves consistency of the database at its boundaries, a transaction that fails to complete might violate the integrity of the database. So, while each transaction it preserves consistency of the database at its boundaries. The transactions or each and every transaction, it preserves the consistency of the database at its boundaries.

Consistency, I have already told you that the transaction it has to or it needs to maintain the integrity constraints on the database. So, normally each transaction it preserves the consistency or the database at its boundaries, no problem. At the boundaries it maintains the consistency but a transaction that fails to complete might violate the integrity of the database. So, we know that each transaction preserves the consistency of the database at its boundaries, but a transaction which fails to complete might violate the integrity of the database.

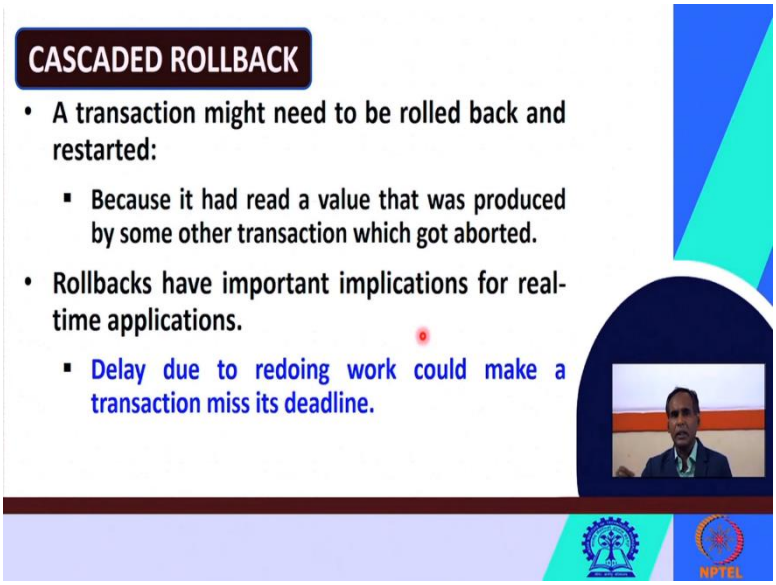
Now when a transaction fails to complete, how can ensure atomicity? When a transaction fails to complete, you can use the rollback protocols. So, rollback protocols are used to ensure atomicity when a transaction fails to complete. Rollback just you what rollback or go back, undo. Rollback means just you undo perform some undo operations. So, when a transaction fails to complete, the rollback protocols they may be used to ensure atomicity.

(Refer Slide Time: 11:55)




So, I have already told you that rollback protocols, they can be used to ensure atomicity not only atomicity, they can also be used to ensure the durability properties when a transaction fails to complete. So, rollback protocols they are used to ensure atomicity as well as durability properties when a transaction fails to complete. So, anytime whenever a transaction fails to complete, these rollback protocols they may be used to ensure atomicity and durability.

(Refer Slide Time: 12:26)




Now, let us see what I mean by cascaded rollback. A transaction might need to be rollback and restarted because it had a read value which was produced by some other transaction which got aborted. So, now let us see why a transaction might need to be rolled back. A transaction might

need to be rolled back and it has to be restarted because this transaction might have read a value and that value was produced by some other transaction which has already got aborted.

So, since this previous transaction, it has already got aborted so due to this, this first transaction has read a value and that value was produced by some other transaction which has already been got aborted, so in this case, this transaction might need to be rolled back and it has to be restarted.

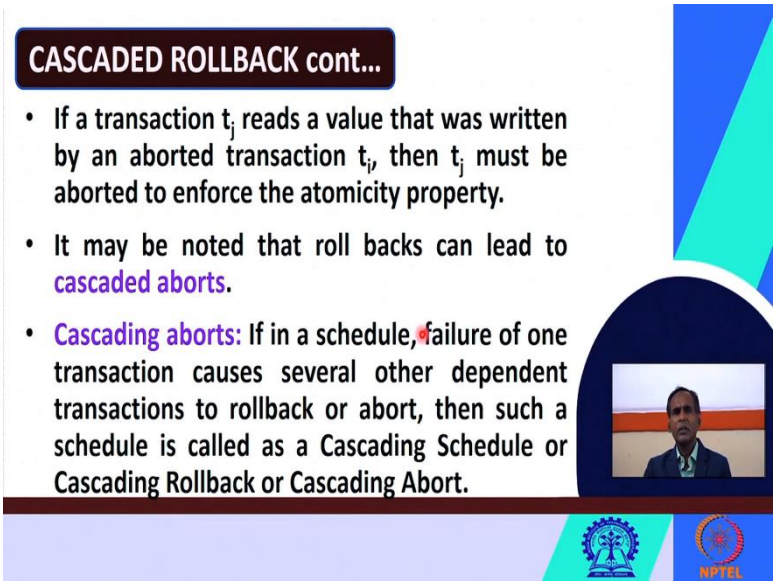
So, a transaction it might need to be rolled back and it has to be restarted because this transaction had read a value that was produced by some other previous transaction which has already got aborted. So that is why this recent or current transaction, it needs to be rolled back.

These rollbacks have important implications for real time applications. I have already told you rollback means what, you have to what, first you have to undo and again you have to redo some of the work. So, while you will redo those some of the works, what will happen, it may the transaction may miss its deadline.

So, rollbacks have very important implications for real time applications. For normal applications, deadline or delay that does not affect much. In that case, rollbacks are okay, no problem, they are fine. But for real time applications, when you are performing rollback then after this undoing action, you have to redo some of the work and due to and for redoing, it will take some time. And by this time, your deadline might have passed. The transaction may miss the deadline.

So, that is why we want to conclude like that these rollbacks even if they can be used for ensuring atomicity and durability, but the rollbacks they have important implications for real time applications because the delay that is caused due to the redoing work, it could make a transaction miss its deadline. By this time, the transaction may miss its deadline. That is that the rollbacks they have important implications on the real time applications.

(Refer Slide Time: 14:41)




Now let us take a small example actually, how does this what rollback might happen? Suppose, there is a transaction. If a transaction tj it has read a value which was written by an aborted transaction tj. That means tj reads a value that was written by a transaction ti and the ti has been aborted. Then what I have to do? I must have to aborted also tj. So, if a transaction tj reads a value which was written by another transaction ti which has been aborted, then this transaction tj also must be aborted to enforce the atomic property.

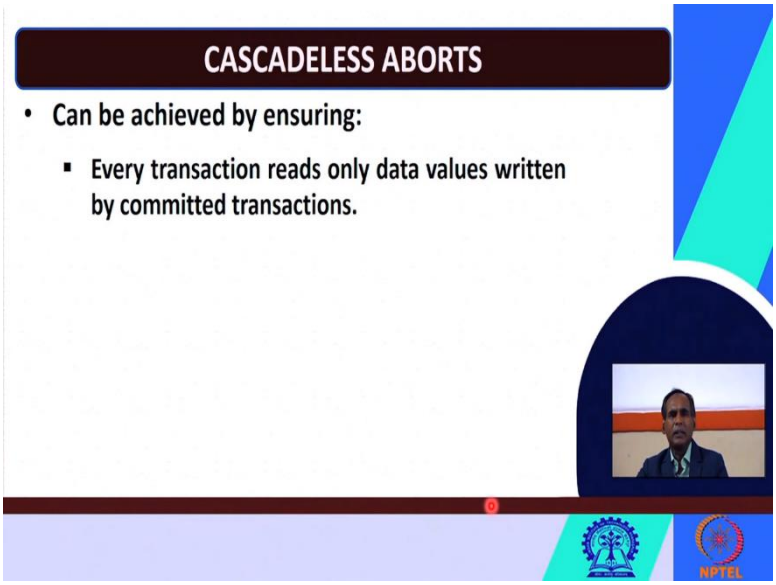
So, since tj has read a value by or which was written by an aborted transaction ti, then tj must have to be also aborted in order to enforce the atomicity property. Then let us say that you might have already known that it may be noted that the rollbacks can lead to cascaded aborts when you are performing these rollback operations, it may be noted that the rollbacks can lead to cascaded aborts. So, what do you mean by cascading aborts?

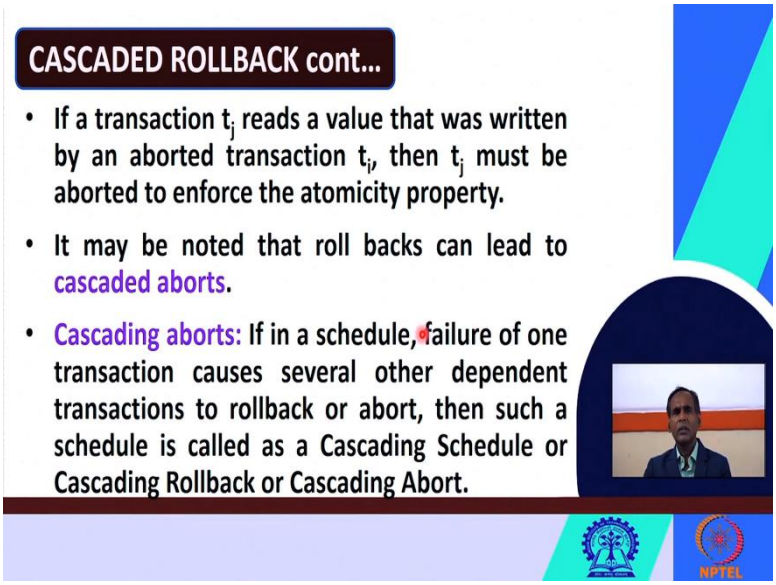
So, if in a schedule, failure of one transaction causes several other dependent transactions to rollback or abort, then such types of schedules are called a cascading schedule or cascading rollback or cascading abort. I am repeating again what do you mean by cascading aborts? Suppose, you are having a schedule, there are suppose many three number of transactions A, B, C.

Failure of one transaction suppose transaction A has failed, if it causes several other dependent transactions because this transaction A was dependent on B and B was dependent on C so transaction if C has failed, since B is dependent on C, so B has to abort it and on B, A is dependent. So, since B has to be aborted, so also A has to be aborted.

So, in that way you can generalize in this way. If in a schedule, the failure of one transaction it may cause it may lead to the fact that several other dependent transactions on it they have to abort, then such a schedule is called as a cascading schedule or cascading rollback or cascading abort. This you have already read in database paper.

(Refer Slide Time: 17:10)



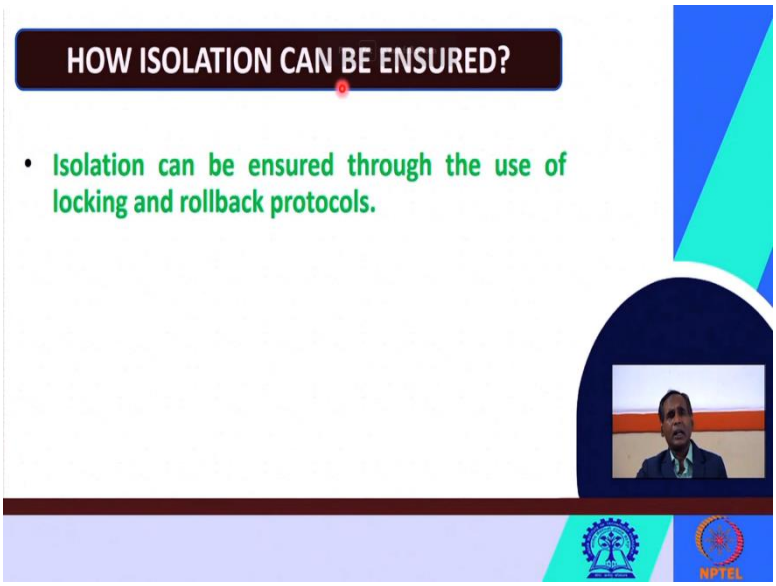




Then how can achieve a cascadeless abort? I have already told you; rollbacks may lead to cascaded aborts. What is a cascade abort I have already told you? Then how can achieve cascadeless aborts? So, cascadeless aborts can be achieved by ensuring that every transaction reads only data values written by the committed transactions not by aborted transactions.

So, cascadeless aborts, they can be achieved by ensuring that each transaction it reads only data values written by committed transactions not by aborted transaction. So, if what every

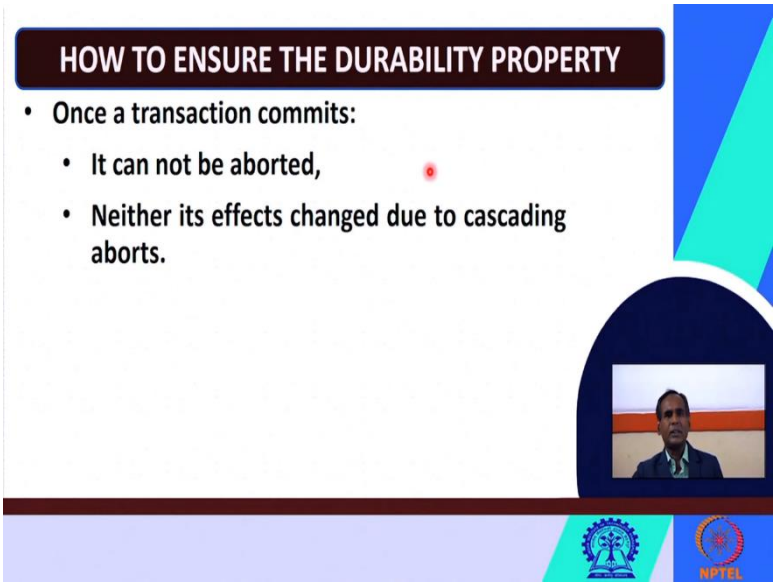
transaction reads only data values written by committed transactions, then we can achieve cascadeless aborts.

(Refer Slide Time: 17:57)




Now, we have already seen how atomicity can be ensured. Let us see how isolation can be ensured. So, isolation can be ensured through the use of locking and rollback protocol. So, you have already known locking and unlocking in case of normal databases. So, isolation can be ensured by using or through the use of locking and rollback protocols.

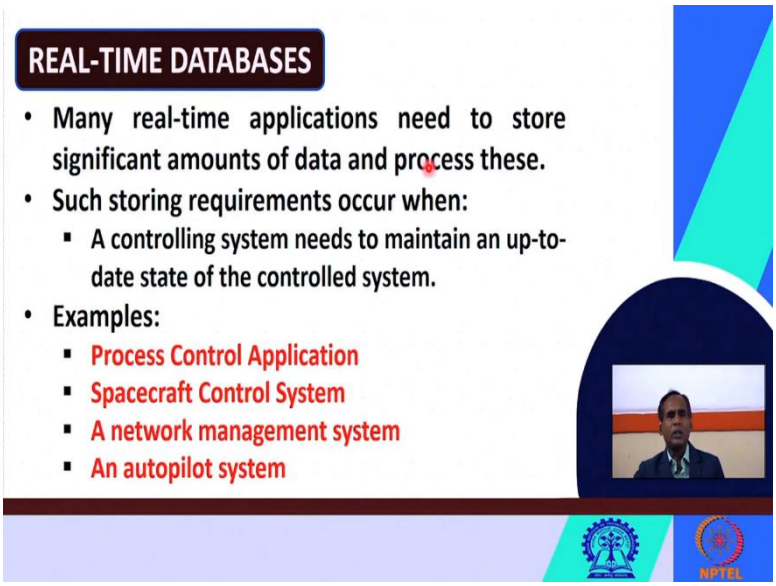
(Refer Slide Time: 18:19)




Now let us see how we can ensure durability property. So, you know once a transaction commits, it cannot be aborted. If already if a transaction commits, then it cannot be aborted now. So, once a transaction commits it cannot be aborted, neither its effects change due to

cascading effect. So, in this way you can share that once your transaction is committed, once a transaction commits, you can ensure the durability property.

(Refer Slide Time: 18:44)



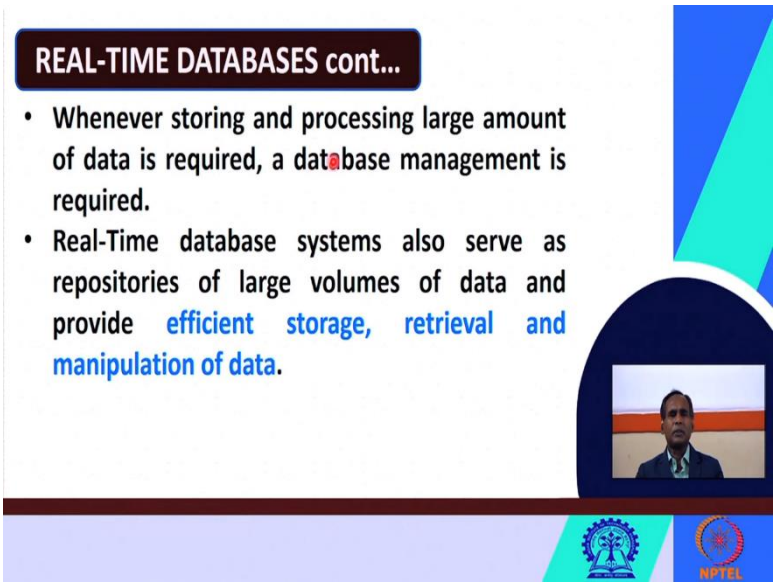

Now let us go to about the real-time databases. What do you mean by exactly there we have already seen the conventional databases, some basic concepts we have seen? A basic review of the conventional databases we have seen. Now let us see about the real-time databases, many real-time applications need to store significant amounts of data and process these. So, many of the real-time applications they require to store a significant volume of data, significant amounts of data or significant volume of data and process these data.

So, storing these data may require some additional thing. Let us say such storing requirements occur when a controlling system needs to maintain an up-to-date state of the control system. So, what I am saying here, many real time applications, they need to store significant amounts of data or a large volume of data and process these such storing requirements they occur when a controlling system, it needs to maintain an up-to-date state of the control system. So, these storing requirements they may occur when the controlling system needs to maintain an up-to-date state of the control system.

So, there should be a controlling system which needs to maintain an up-to-date state of the control system. You can see some of the applications where real-time databases are used. So, I have listed here some applications where the real time databases are used such as process control application, spacecraft control system, network management system, then internet service system, autopilot system. These are all examples of the applications where real time

databases are used. So, today we will discuss only this process control application. Some other examples we will discuss in the next class.

(Refer Slide Time: 20:33)

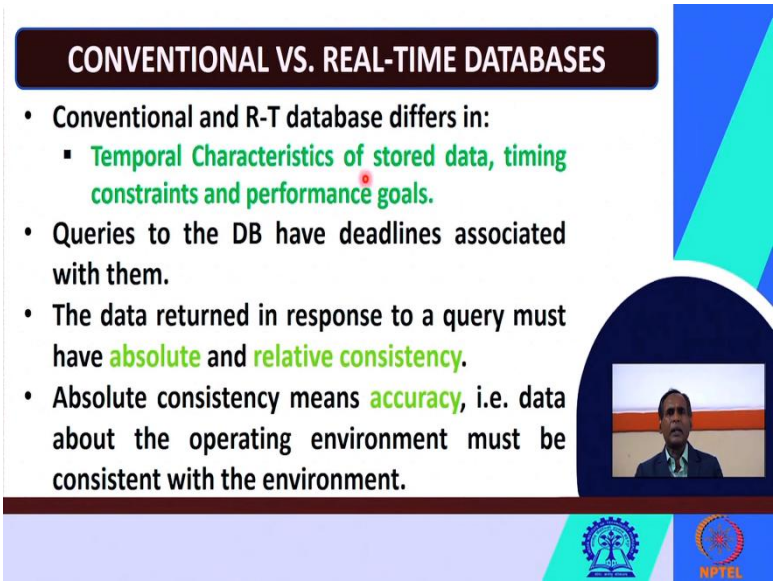



Now let us say, we are going to that example let us see, again, some of the fundamental concepts. Like whenever storing and processing large amounts of data is required, a database management system is required. So, when you require to store, process and update a large amount of data, then you have to use a database management system or collage we have the DBMS database management system.

The real-time database systems, they also serve as repositories of large volumes of data. The real-time system database systems, they also serve as repositories of what, large volumes of data and these real-time database systems, they also provide efficient storage of data, retrieval and manipulation of data.

So, by using real time database management systems, we can efficiently store the large volume of data, we can retrieve the data from this database and we can manipulate the data stored in the database. So real time database systems, they also serve as a repository of large volumes of data. And they can provide efficient storage, efficient retrieval and efficient manipulation of the data.

(Refer Slide Time: 21:52)




So now, we will quickly look at the conventional versus the real time databases, the conventional and the real time databases, how do they differ? Normally, the conventional and real-time databases, they differ in several characteristics. So, three of the important characteristics I have represented here by which the conventional and real-time databases may differ. Number one, the temporal characteristics of stored data, temporal means here, we will see the time aspects.

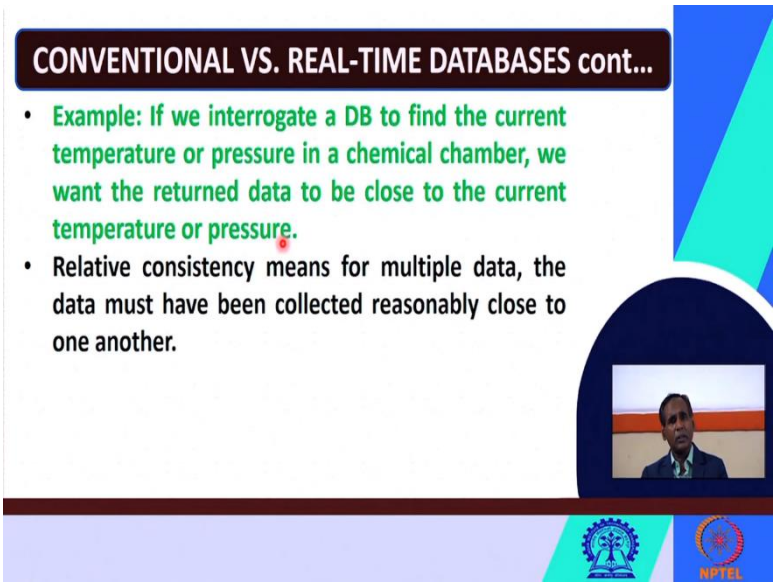
So, the conventional and real time databases, they differ in the temporal characteristics of the stored data, then they also differ in the different timing constraints. And also, they differ in their performance goals. So, mainly in these three ways conventional and real-time databases differ. So, another important aspect or another one more difference, you can see the queries to the database, they have deadlines associated with them in real-time database. In conventional databases, normally, no such deadlines are associated or that is flexible.

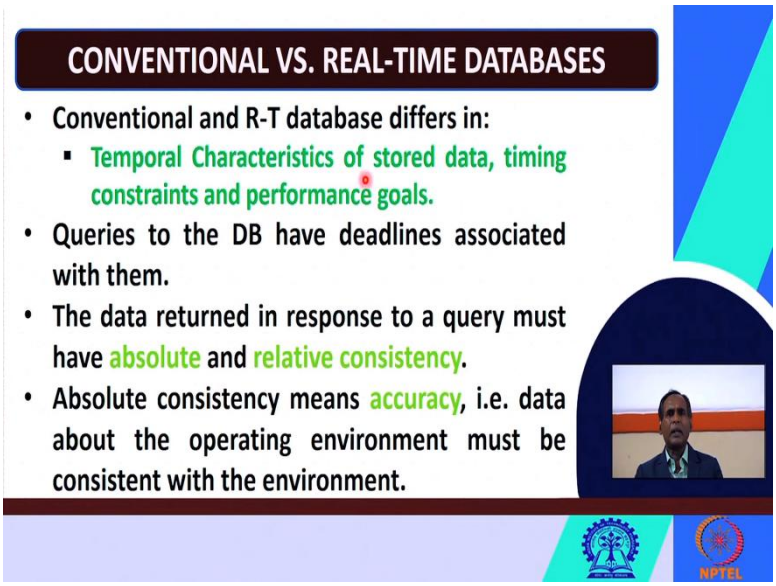
But in real-time databases, the queries to the database, they have some deadlines associated with them and another difference, you can see the data returned in response to a query in real-time database must have some absolute and relative consistency. Listen to what I am saying. Suppose you have fired a query, then the data returned in response to that query must have two types of consistencies, absolute consistency and relative consistency. So, this might not be required in the conventional database.

Now let us see what is absolute consistency and what is relative consistency. Absolute consistency means some accuracy. That is the data about the operating environment must be consistent with the environment, okay. So, when we are saying absolute, absolute consistency,

it means some accuracy, that is, the data about the operating environment must be consistent with the environment.

(Refer Slide Time: 23:54)




We can take a small example here. So, if we interrogate a database, suppose we are interrogating a database to find the current temperature or the pressure in a chemical chamber, then we want the returned data to be close to the current temperature and pressure. What I am saying again, I am discussing an example of what absolute consistency.

So, if we interrogate a database to find out a query such as what is the current temperature or what is the current pressure in a chemical chamber, then we want the returned data, they should be close to the current temperature or pressure, because I have already told you, it is means to absolute means what, the data about the operating environment must be consistent to the

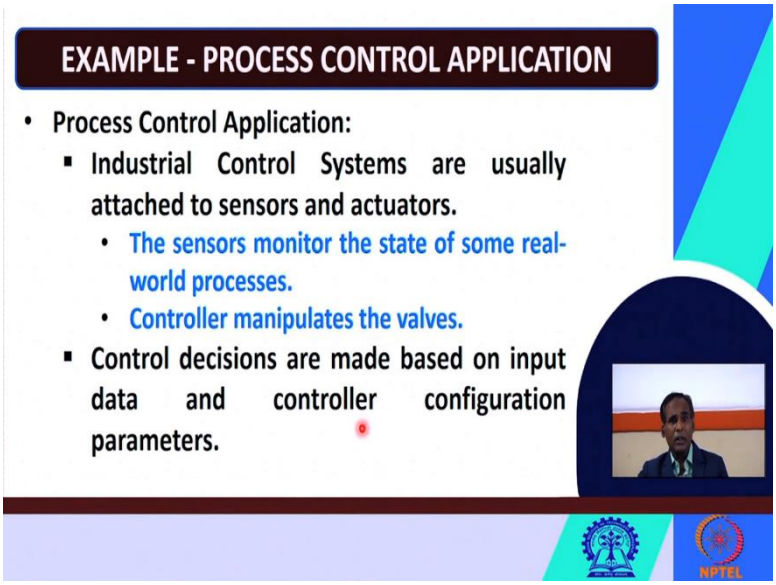
environment. So, if you interrogating a database to find the current temperature or pressure in a chemical chamber, then we want the returned data, what data we are getting, this should be close to the current temperature and pressure. There should not be much variation.

Then we will see relative consistency. What do you mean by relative consistency? Relative consistency means for multiple data, the data must have been collected reasonably close to one another. So, if you are getting or asking for two or three or many data, then what will happen, the data that you are collecting they must be collected reasonably close to one another. Suppose, again let us take the same example, chemical chamber.

Two data you want to get like the temperature and the pressure and if you are collecting the temperature at time t is equal to 100 and pressure you are collecting our t is equal to 300 then we say that they are not consistent, they are inconsistent. When you are collecting the temperature, if it is at time t is equal to 100, the pressure should be also collected at the same time or close to that t is equal to 100.

If you are collecting the temperature at t equal to 100 and collecting the data of the pressure at t = 200 or 300, we say that the data are inconsistent. So, relative consistency means for multiple data, the data must have been collected reasonably close to one another.

(Refer Slide Time: 26:16)



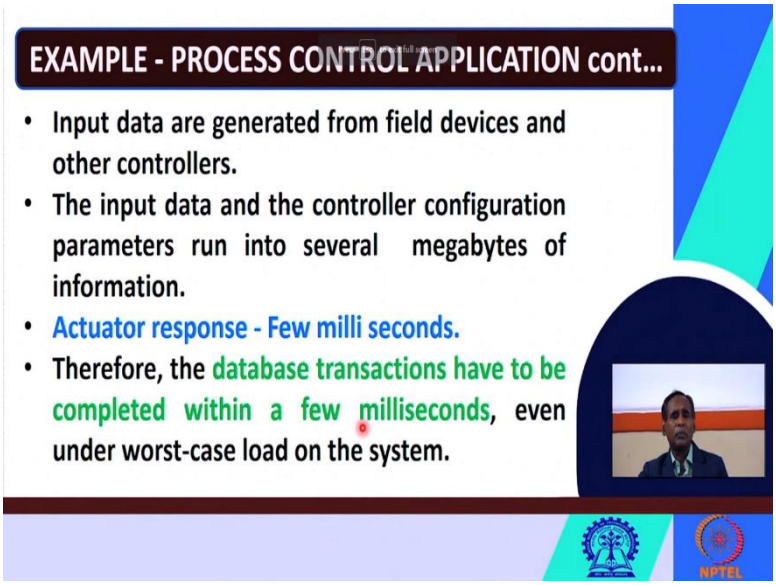

Now let us quickly take up a small example today. So, we will take up an example where real-time databases are required. So, today we will discuss this process control application. This is one of the example where this is one of the application where real-time database is required.

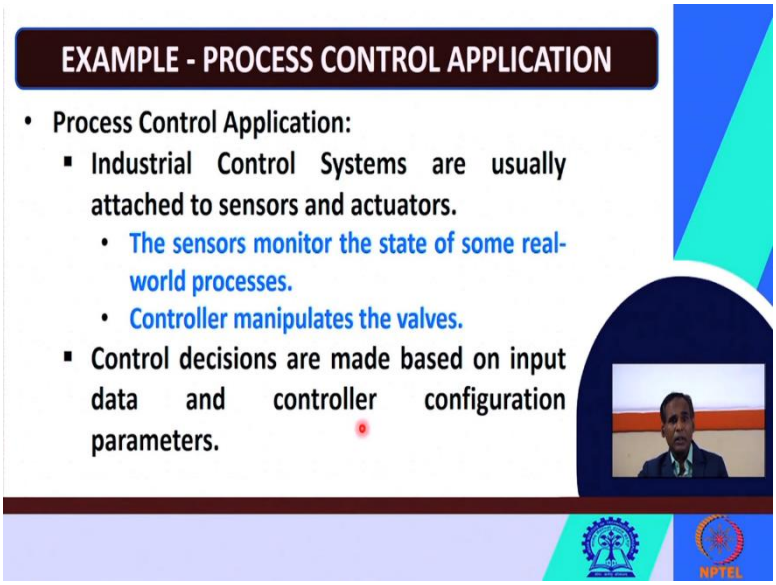
So, normally the process control application what happens here, the industrial control systems are usually attached to sensor and actuators. Whenever you are talking about a process control application, so, these are you can see examples you can see that the industrial control systems they are usually attached to sensors and actuators. So, any real time systems you see in most of the real-time systems of the sensors and actuators are connected. So, industrial control systems which are examples of real-time systems, they are usually attached to a number of sensors and actuators.

These sensors what do they do, the sensors they monitor the state of the some real-world processes. They will sense the what environment they will collect some data, the sensors they monitor the state of some real-world processes and what do the controller do? The controller manipulates the valves. So, here we are discussing a industrial control system or a process control system. Here the controller manipulates the impulse, the control decisions are made based on input data and controller configuration parameters.

The control decisions, so for the of the controller, it manipulates the valves so, the control decisions are made based on what? The control decisions are made based on what the input data and the controller configuration parameters. So, in a process control application or in a industrial control system, the control decisions are made based on the input data and controller configuration parameters.

(Refer Slide Time: 28:28)






The input data those are generated from the field services, input data is generated from where, from the field, the devices and other controllers. So, while you are discussing about the process control application, the input data they are generated from the field devices and from the other controllers, the input data and the controller input configuration parameters they run into several megabytes of information. See the size is quite a little bit large.

So, the input data and the controller configuration parameters in case of a process control application, they may vary or they may take what up to the size of some megabytes. So, the input data and the control and permission parameters they may run into several megabytes of information and but see I have already told you this industrial control systems are connected into they are attached to some sensors and actuators, the actuator response it may takes or each of the order of some few milliseconds you see, they may require a huge space may be of the order of several megabytes of information.

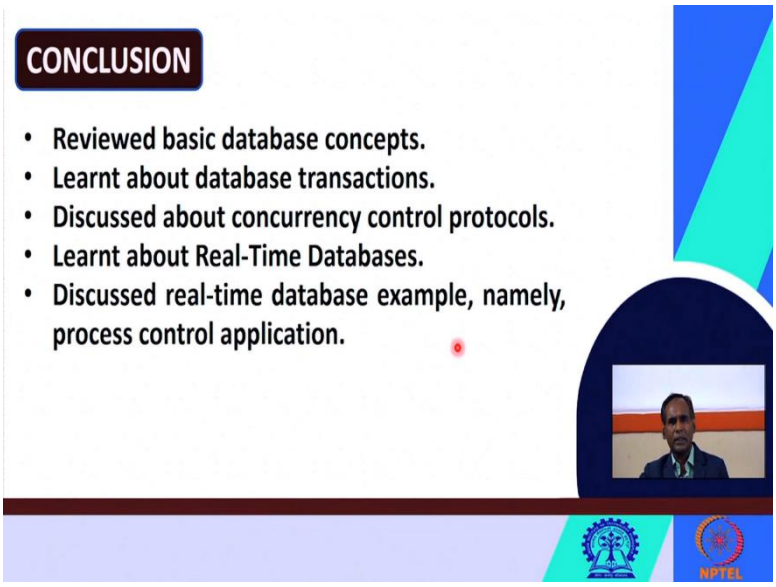
The actuator may respond or the actuator response time maybe order of some few milliseconds. So therefore, the database transactions have to be completed within a what, few milliseconds. I am discussing about the real-time database. Real-time database means here some timing constraints are associated with the database when you are asking some query or some transaction has to be completed that has to be completed within some few milliseconds otherwise, say some deadline will be missed or it will be crossed.

The transaction might miss the deadline etc. So, therefore, the database transactions have to be completed within a few milliseconds. Therefore, the database transactions have to be completed within a few milliseconds even under worst case load on the system, even if the system is in full load, even under worst case load on the system, the database transactions have

to be completed within a few milliseconds otherwise, what will happen the transaction may miss the deadline.

In this way you have seen in this application, process control application how it uses real-time databases and the transactions how they should be completed within a few milliseconds even if or even under worst case load on the system otherwise, the transactions may miss the deadlines. This is how we have seen process control application uses real-time databases.

(Refer Slide Time: 30:34)




So, today we have reviewed some basic database concepts. We have learnt about the database transactions, we have also seen about the ACID property of the transactions, we have discussed about the little bit concurrency control mechanism and we have learnt about what is real-time databases, how do they differ from the conventional databases. We have discussed a small application for a small application which uses real-time database, that application is process control application.

(Refer Slide Time: 31:12)


So, we have taken these things from these two books.

(Refer Slide Time: 31:15)


Thank you very much.