GEOGRAPHIC INFORMATION SYSTEMS

Prof. Bharath H Aithal
Ranbir and Chitra Gupta School of Infrastructure Design and Management Indian Institute of Technology - Kharagpur

Module No # 07
Lecture No # 35
Normalization Forms

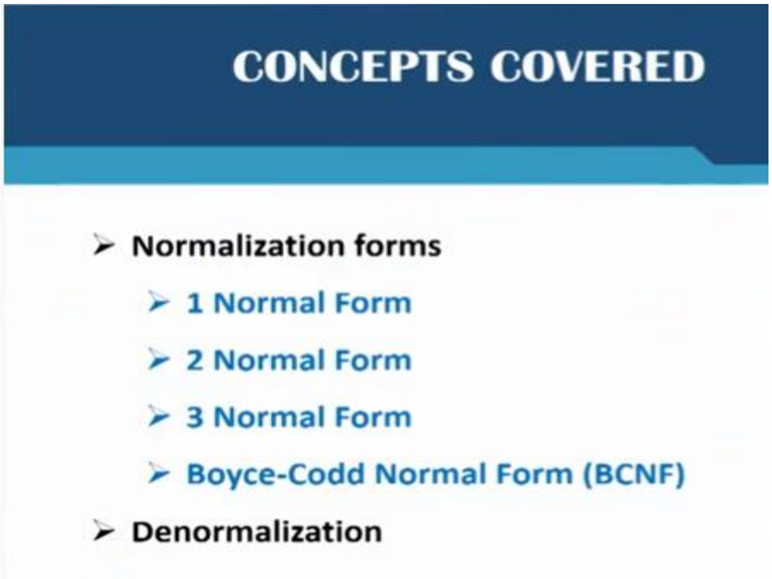
Hello Namaste welcome back to the course on geographic information system today's we have in the previous this entire week we spoke about what is the database what are the different types of database query language. Then we have also looked at what is what are the 2 types of database management system languages then we looked at what is the database management system basically we also looked at different types of database manage we looked at hierarchal database we looked at relational database object oriented databases.

Now once we have looked at all of these now we have learnt what is the database and what are different database management systems and database system by default. So once you have learnt it now you have data you have data in the database. Now if you have to really query something or understand how best the data is and how best the data can be assessed or how fast the data can be accessed for that you need certain ways of organizing your database.

It is just like if your house has to look really neat and you have to access some of the things very fast. So you organize your house in such a way that a certain things are easily accessible or if you have one particular thing you can find out the other. So similarly in a database you will have to do certain kinds of probably a manipulations in terms of normalization that is what is called as data normalization.

We have learnt the in our previous class today we would learn what a different normal forms okay we have 5 different normal forms we look at it and how if your data is normalized how do you de-normalize you need it for certain operations when you have to de-normalization or you are moving data out of a database.

(Refer Slide Time: 02:21)

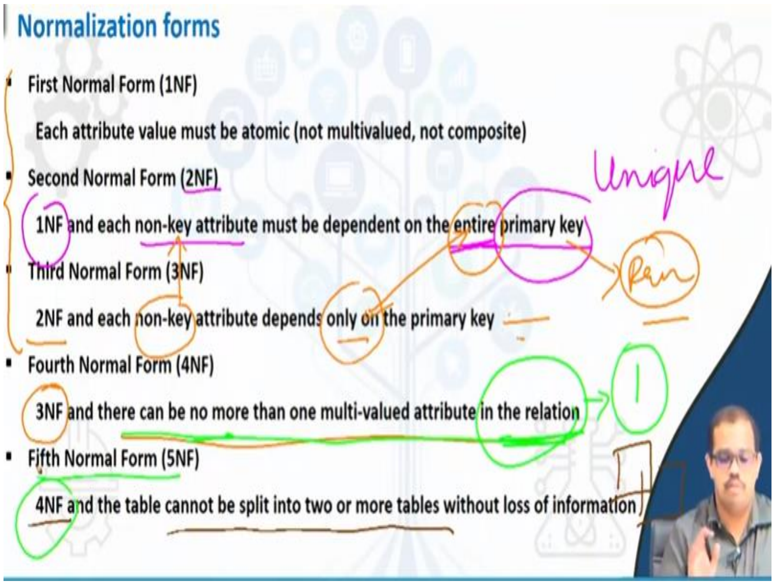


CONCEPTS COVERED

- Normalization forms
  - 1 Normal Form
  - 2 Normal Form
  - 3 Normal Form
  - Boyce-Codd Normal Form (BCNF)
- Denormalization

So now in this class first we would look at certain concepts first as I said I we would look at normalization forms we will look at first normal form, second normal form, third normal form and we look at third normal form we will also look at Boyce-Codd normal form or BCNF and we will also look at forth and fifth normal forms though I have not listed it here and finally we would also look at de-normalization okay. So how do we get out of normalization and how are we do and de-normalization of a data that is in the normal form.

(Refer Slide Time: 02:51)



Normalization forms

- First Normal Form (1NF)
  Each attribute value must be atomic (not multivalued, not composite)
- Second Normal Form (2NF)
  1NF and each non-key attribute must be dependent on the entire primary key
- Third Normal Form (3NF)
  2NF and each non-key attribute depends only on the primary key
- Fourth Normal Form (4NF)
  3NF and there can be no more than one multi-valued attribute in the relation
- Fifth Normal Form (5NF)
  4NF and the table cannot be split into two or more tables without loss of information

Now let us understand normal now when we look at normalizations as I said is to put a data in order in a database in order to see how well your data can be organized. So that retrieval is fast quicker and you have a efficiency in the database now when I say the data is in the first normal form it means to say that each attribute data must be atomic. When I say atomic it should not have multi value which means every attribute data say cannot have 2 values at an instant okay and it is not composite data.

And once you have attained the first normal form if you have to look at a second normal form which means you have arranged your house in particular order so which mean that some only some of those are quite accessible retrievably fast but if you want to improve that so next step is the second normal form. When is say second normal form the main condition is that your database first should be in the first normal form.

Once your database in the first normal form each key non key attribute so it is not about key it is all the non-key attributes must be dependent on the primary key. Now please understand the usage of primary key and a non-key here okay so when I say primary key interest and unique key where which is uniquely identified in the entire database correct. So now when I am looking at the second normal form it is that it has to be in the first normal form and any non-key attribute which means which are those which are not considered as any kind of keys whether it is primary key or whether it is secondary key or whether it is foreign key.

So none of these should be a non-key attributes none of these forms so it is a non-key attributes that must be dependent entirely on the primary key okay. So this is what means the second normal form now so once you have attained this the second normal form now you try to see that you attain the third normal form which means you are trying to iterate it more, So that your house looks more cleaner or more organized okay.

So when I say third normal form it is that basic thing is it has to be in the second normal form right and each non-key again it is the same non-key attribute that depends only on here at least look at it here it is entirely on the primary key but here it is only on the primary key which means it can even depend on primary key plus any other key but it has to be entirely dependent on primary key for this retrieval.

But in the third normal form it has to be dependent on the primary key only without dependency on the primary key it cannot be got into the third normal form okay. So this is about the third normal form now the next stage is the fourth normal form which means extremely organized okay. So now if you want to make it extremely organized it means that you should have your data already the third normal form for your fourth normal form and there can be no more than one multivalued attributes in the relation.

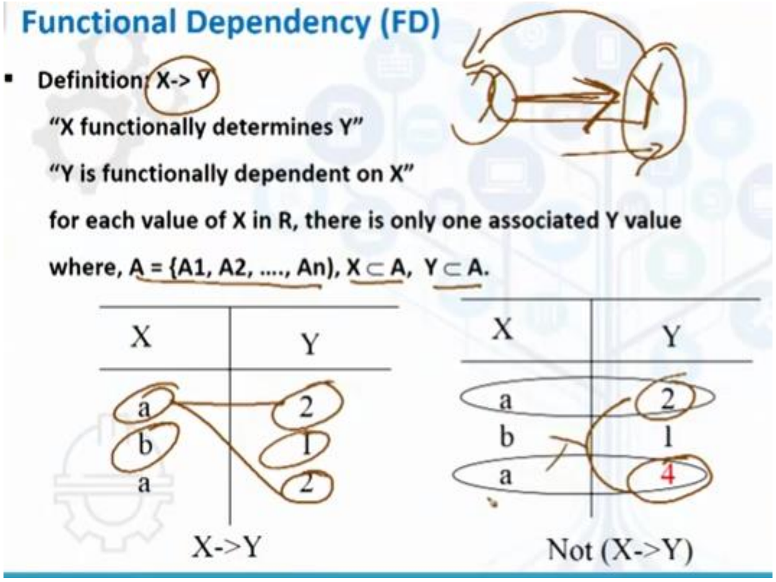
Till here you did not have any of these 3 forms we did not have any kind of multivalued functions but in the third normal form you there can be no more than 1 multivalued which means any attributes can have only one multivalued attribute in the relation that is what is the fourth normal form. So now we are in the stage where first we cut down it that you cannot have and you should have always have an atomic form.

Now once you have attained the atomic form that is it is not the multivalued then in the second normal form we use a first normal form and a non key attribute okay which is entirely dependent on the primary key. In the third normal form we use a second normal form and a non-key attribute that is dependent only on the primary key. But it is a fourth normal form we have third normal form there can be certain attribute values which are multi valued in terms of it is relation.

So it is only in terms of it is relation so please understand this point so multivalued in terms of it relation. The final thing is the fifth normal form again has to be in the fourth your data as to be in fifth normal form then the first thing is satisfies the fourth normal form conditions. Then table cannot be split in 2 or more tables without a loss of information okay. So if you are trying to look at the fourth normal form after fourth normal form there are many attributes which are multivalued which means that there are more than 1 multivalued form attributes.

So you try to see that the loss you can duplicate into the number of rather than duplicate you can multiply the number of tables but that have multiplication of into tables will lead into certain loss of information in the fifth normal form. So normally the fourth normal form as considered to be extensively best okay but the fifth normal form has certain issues in terms of loss of data okay. So that is why it is normally not used.

(Refer Slide Time: 09:06)



Functional Dependency (FD)

- Definition: X-> Y

"X functionally determines Y"

"Y is functionally dependent on X"

for each value of X in R, there is only one associated Y value

where, A = {A1, A2, …., An), X ⊂ A, Y ⊂ A.

X	Y
a	2
b	1
a	2

X->Y

X	Y
a	2
b	1
a	4

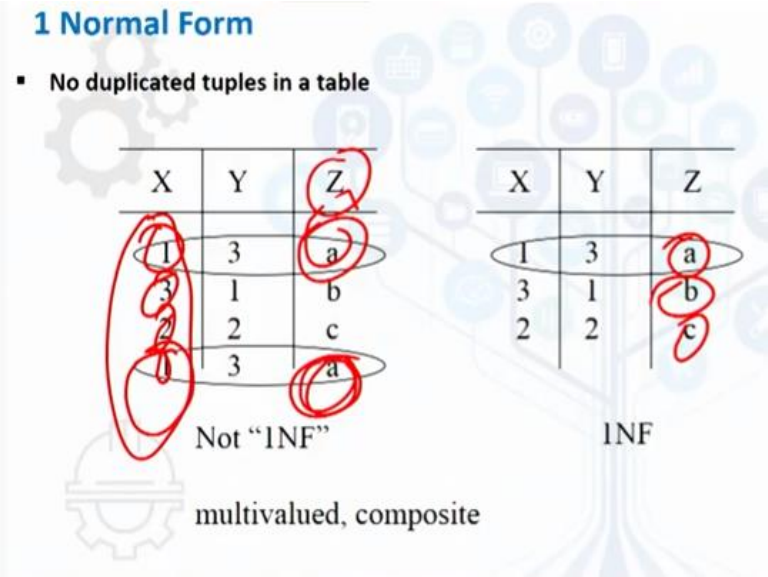
Not (X->Y)

Now when we are looking at yeah when we are looking at the normal forms the first thing we have to understand is the concept of functional dependency when I say when I think about a value X is value Y is functionally dependent on X which means you are trying to write that value Y survives because of X the value of X says if the this arrow is positioning like this it means to say that this particular variable is functionally dependent on this or X functionally determines Y okay.

So without X Y cannot be found out that is exactly the meaning okay so Y is functionally dependent on X the arrow shows which variable has functionally dependent on which particular variable. So now for each value of X in R which means say in that particular set there is one associated Y values always when it is functionally dependent okay. So let us consider this as a particular set which means X belongs to A Y also belongs to A okay.

Now if X belongs to A okay now if it as to be functionally dependent it means to say X and Y if you see here this value **this value** so Y is functionally dependent on X okay. But when you look at this particular value is 4 and this particular value is 2 you cannot have such dependency that exists for a single particular attribute. So it is functionally not dependent on X it cannot be determined okay basically. So this is about functional dependency so once you have understood about functional dependency let us look at normal form.

(Refer Slide Time: 11:12)



1 Normal Form

- No duplicated tuples in a table

X	Y	Z
1	3	a
3	1	b
2	2	c
1	3	a

Not “1NF”

X	Y	Z
1	3	a
3	1	b
2	2	c

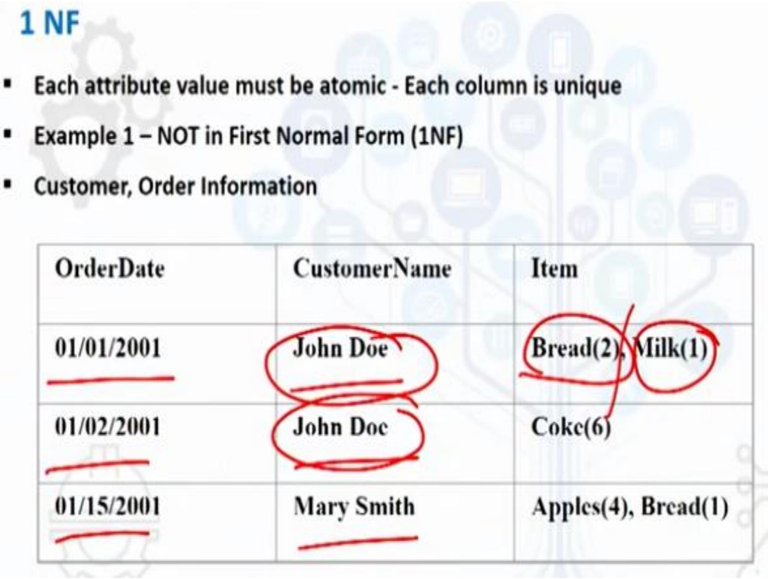
1NF

multivalued, composite

Now what does first norm forms says? It the first normal forms says the value should be atomic so what does atomic mean? Atomic basically means every column is unique so when you are looking at it let us consider this particular example XYZ when you are looking at each of these columns okay XSXS 1, 3, 2, 1. Now this particular column itself is not unique okay if we consider X as a primary key but and Z as attribute data so you both attribute data repeats and it is nor unique.

So it is not in a first normal form it is multivalued function okay so now if you look at a same table that has been put in the first normal form the value is unique it does not repeat. So it is not a multivalued nor composite function. Here it is more of a composite function so this what is called a first normal form how you translate very simple example of a firs normal form. Your database would be too huge it will not be as simple as what i am showing here but this is just an example of giving you an idea of what does first normal form mean.

(Refer Slide Time: 12:40)



1 NF

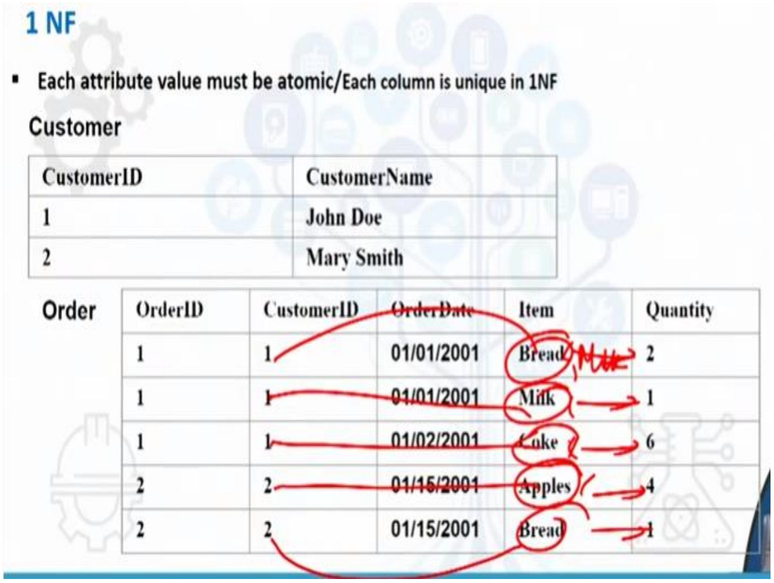
- Each attribute value must be atomic - Each column is unique
- Example 1 – NOT in First Normal Form (1NF)
- Customer, Order Information

OrderDate	CustomerName	Item
01/01/2001	John Doe	Bread(2), Milk(1)
01/02/2001	John Doe	Coke(6)
01/15/2001	Mary Smith	Apples(4), Bread(1)

Now again let me take another example here you have a customer order information okay. When I looking at a customer order information there is an order date look at every details very carefully. You have an order date this are 3 different order dates all 3 order dates are independent they are not on the same day okay. So you have a customer name so when you look at the customer name you have the same name that is repeating and when you look at it even if this customer name we consider it as a primary key if you item as an attribute data.

Attribute data has to be if you look at here it has not atomic each column is not unique it has 2 different value okay.

(Refer Slide Time: 13:41)



1 NF

- Each attribute value must be atomic/Each column is unique in 1NF

Customer

CustomerID	CustomerName
1	John Doe
2	Mary Smith

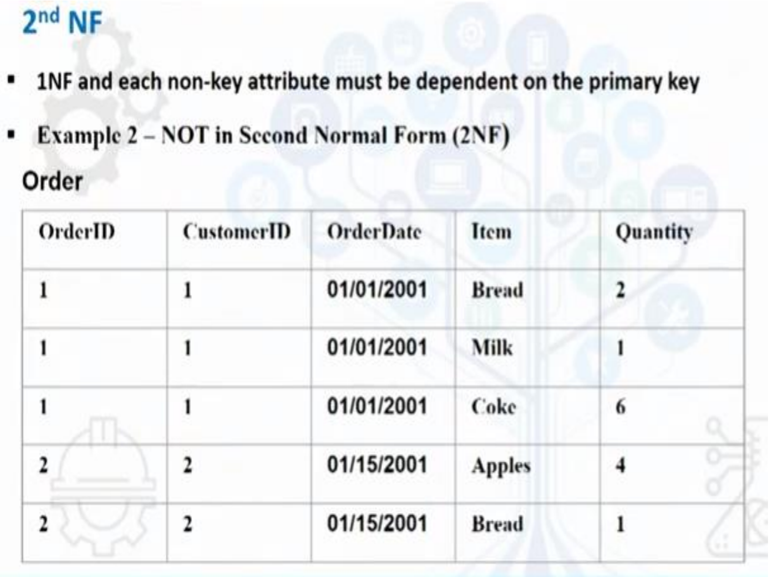
Order

OrderID	CustomerID	OrderDate	Item	Quantity
1	1	01/01/2001	Bread	2
1	1	01/01/2001	Milk	1
1	1	01/02/2001	Coke	6
2	2	01/15/2001	Apples	4
2	2	01/15/2001	Bread	1

Now if this particular thing can be translate if we want to try and translate this particular thing in the first normal form it is as simple as this I create a table which just a customer ID so this would be my primary key in this case I would create a customer ID and a customer name okay. Now this particular customer ID is linked to another table which as order ID order date and the item. Now you can see any of these tables no data is all data are atomic if you see here bread, milk, coke, apples and bread.

If you look at this is if bread is focus and if you look at this is for first customer and if you look at milk is related to the first customer coke is related the first customer apple is related to first customer and the quantity is related here okay. So most of these values are unique to this particular customer only. We do not have anything like milk etc., so this makes this particular database and first normal form the one that is translated to the first normal form database okay. So I hope everyone has understood what is first normal form this is a very basic form of how database works okay.

(Refer Slide Time: 15:10)



2nd NF

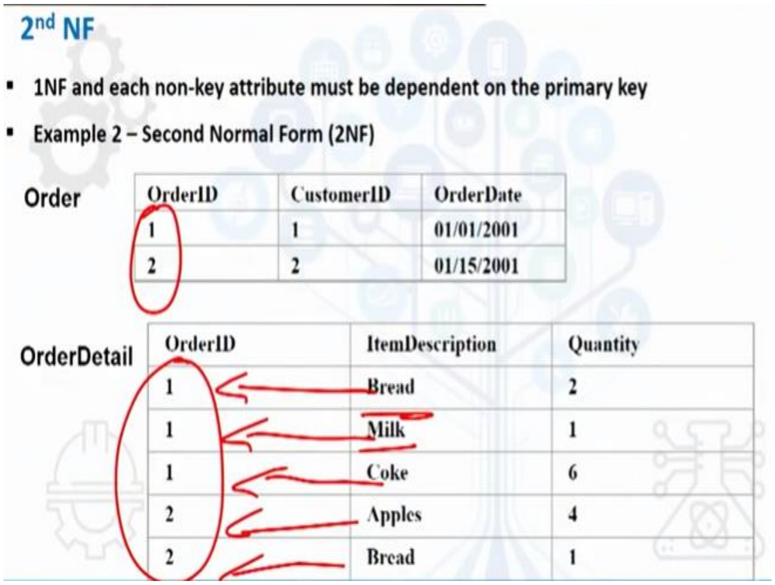
- 1NF and each non-key attribute must be dependent on the primary key
- Example 2 – NOT in Second Normal Form (2NF)

Order

OrderID	CustomerID	OrderDate	Item	Quantity
1	1	01/01/2001	Bread	2
1	1	01/01/2001	Milk	1
1	1	01/01/2001	Coke	6
2	2	01/15/2001	Apples	4
2	2	01/15/2001	Bread	1

So if we are looking at the second normal form I did tell you that it has to be in the first normal form first which means it has to be atomic. Now and each non key attribute must be dependent on the primary key now i said let us take customer ID as a primary key. Now we have all of these are non-key attributes whether it is quantity item or order date. When you are looking at let us say bread. Bread is not completely dependent on the primary key here okay because it is determined by the quantity it is determined also by the order ID okay.

(Refer Slide Time: 15:57)



2^{nd} NF

- 1NF and each non-key attribute must be dependent on the primary key
- Example 2 – Second Normal Form (2NF)

Order

OrderID	CustomerID	OrderDate
1	1	01/01/2001
2	2	01/15/2001

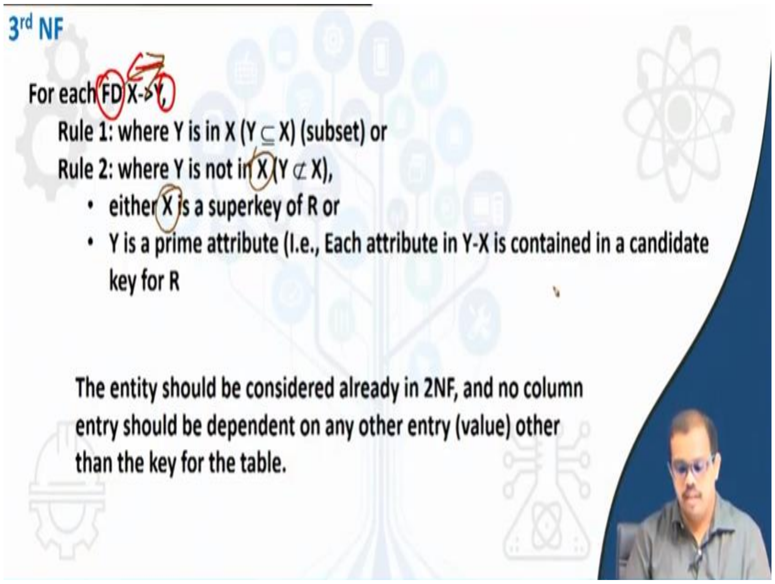
OrderDetail

OrderID	ItemDescription	Quantity
1	Bread	2
1	Milk	1
1	Coke	6
2	Apples	4
2	Bread	1

So if a particular for example here now I am if I consider order ID as a primary key in this case okay now what do I do is that I break up into 2 different forms, 2 different tables. So I consider here order ID as a primary key okay again order ID is here primary key. Now each item here is dependent on the primary key for its existence this is a non-key attribute there is no key associated with it so it is dependent on the primary key.

Primary keys are 1 and 2 okay so that gives you more information about a quantity so now if you have arranged it in a form of primary key you have arranged it in form of a second normal form even looking at atomic values and all these values are atomic there is no second value attached to any of the attributes that are linked here. So it mean it are atomic and also non-key attributes are defined only by the dependent entirely on the primary key right. So this means to say that this particular thing is in the second normal form.

(Refer Slide Time: 17:13)



3rd NF

For each FD X->Y,

Rule 1: where Y is in X ($Y \subseteq X$) (subset) or

Rule 2: where Y is not in X ($Y \not\subset X$),

- either X is a superkey of R or
- Y is a prime attribute (I.e., Each attribute in Y-X is contained in a candidate key for R

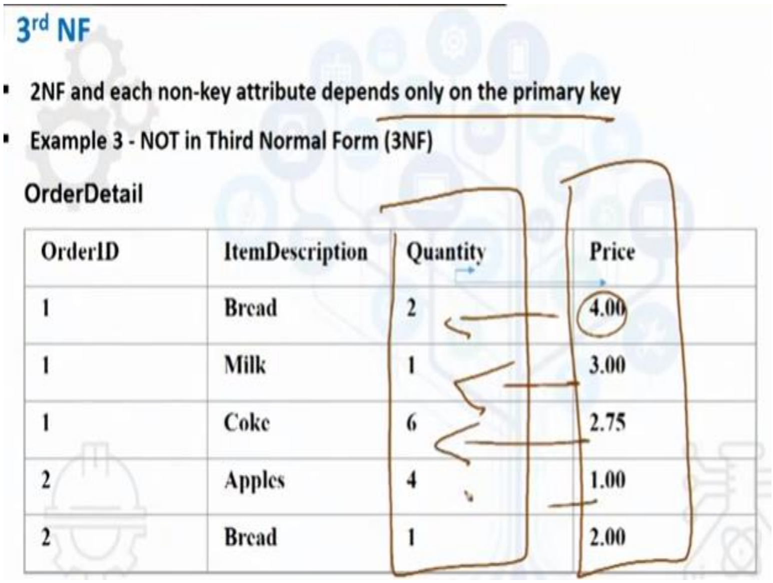
The entity should be considered already in 2NF, and no column entry should be dependent on any other entry (value) other than the key for the table.

Now if it has to be a third normal form the entity should be considered already in the second normal form which mean you should already have the entire table in the second normal form and no to column entries should be dependent on any other entry other than the key for the table okay. So if you understand it other than the primary key for the table so when we look at this if you are going by the mathematical relationship for every functional dependency of Y with X okay.

Now when we are looking at this it is functional dependency something like this so functional dependency of Y to X. So first rule states that where Y is in X which means Y is when you are looking at the subset okay Y is a subset of X or when you are looking at the rule number 2 when Y has not the subset of X okay which means to say that these are 2 different rules where in if you have these 2 rules satisfy for a functional dependence of Y towards X then that particular table is in the third normal form which means in the rule number of 2 if X is a super key of R.

If you are considering this set X is a super key of R is the relationship then entity relationship value so if your R is considered to be the major set and if you are trying to consider X is a super Key of R then Y is a prime attribute which means to say Y dependent on X is contained in the candidate key. When I say candidate key every attribute information can be a candidate key okay. So why that is belongs to X is a contained in the candidate key of for R. So let me explain it in a more simpler way.

(Refer Slide Time: 19:25)



3rd NF

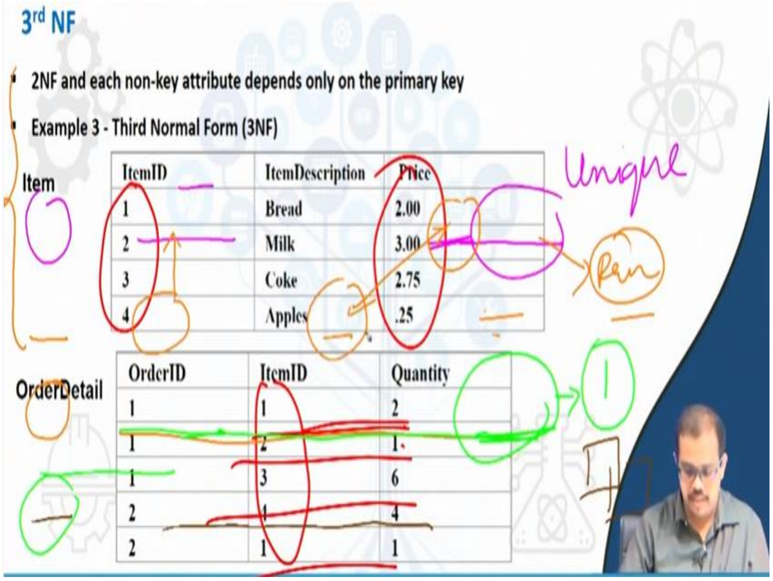
- 2NF and each non-key attribute depends only on the primary key
- Example 3 - NOT in Third Normal Form (3NF)

OrderDetail

OrderID	ItemDescription	Quantity	Price
1	Bread	2	4.00
1	Milk	1	3.00
1	Coke	6	2.75
2	Apples	4	1.00
2	Bread	1	2.00

So if we are considering this example here okay let us say the first thing we have consider is it has to be in a second normal form when we are looking at it here at this particular table we can also see that non-key attributes depend may not depend only on the primary key in this case. When you look at this particular order key it very clearly says that third normal form says that it has to be dependent only on the primary key okay.

But when you see this is one non key attribute when you looking at here price is dependent on the quantity again the price is dependent on the quantity. Now which means to say that this is one of non-key attribute which is dependent on the other and not directly on the primary key that we are trying to use. So this means to say that this particular thing is not in the third normal form. **(Refer Slide Time: 20:32)**



3rd NF

- 2NF and each non-key attribute depends only on the primary key
- Example 3 - Third Normal Form (3NF)

Item

ItemID	ItemDescription	Price
1	Bread	2.00
2	Milk	3.00
3	Coke	2.75
4	Apples	.25

OrderDetail

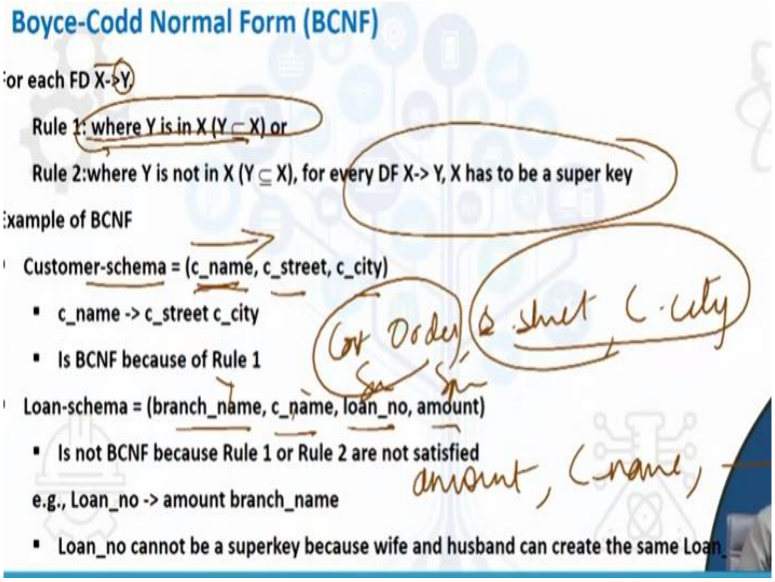
OrderID	ItemID	Quantity
1	1	2
1	2	1
1	3	6
2	4	4
2	1	1

So how so what we would do it is has simple as this again first thing is see that whether it is in the second normal form when I say second normal form it has to be atomic and most importantly it should be dependent on the primary key. Now let us say now I have broken down into 2 groups when I am looking at the third normal form the same item ID is considered to be the primary key here okay. Now I have the same primary keys here okay first thing is it has to be atomic.

Now if you see it is atomic no value is repeated the second thing is it has to be in dependent on the primary key. Now everything is dependent only on the primary key for example number of quantity is dependent item ID quantities again dependent on the item ID again okay. So when you look at the item ID here it says this item description it is price is 3 okay. So it again this is dependent on the item ID so it is in the second normal form.

Now the third normal form says it is only dependent on the primary key yes it is only dependent on the primary key yes both of these are only dependent and not dependent on anything else. So now this table has been split into 2 and it is in the third normal form so this is how we start with normal database first normal form the second normal form and third normal form. So please be careful the way you split up each of these three into different normal forms okay and okay. So now let us also look at what do okay let us look at the next set wherein we will be discussing about the fourth normal form okay.

(Refer Slide Time: 22:32)



Boyce-Codd Normal Form (BCNF)

or each FD X->Y

Rule 1: where Y is in X (Y ⊂ X) or

Rule 2:where Y is not in X (Y ⊆ X), for every DF X-> Y, X has to be a super key

xample of BCNF

- Customer-schema = (c_name, c_street, c_city)
  - c_name -> c_street c_city
  - Is BCNF because of Rule 1
- Loan-schema = (branch_name, c_name, loan_no, amount)
  - Is not BCNF because Rule 1 or Rule 2 are not satisfied

  e.g., Loan_no -> amount branch_name
  - Loan_no cannot be a superkey because wife and husband can create the same Loan

Now before we go into fourth normal form we will also look at BCNF as i said it is Boyce-Codd normal form when we are looking are functional dependency that is why being dependent on X okay. It means when I have a rule Y is in X and Y is not in X it means to say the subset of X are not in X. We would define in the second rule we have already defined that X has to be a super key this we have already seen previously.

How let me give you an example of this we have a customer's schema or customer database we have customer name customer where he stays which street and which city okay. Let us say we are trying to look at the database of Amazon. So Amazon may have huge number of customers all throughout or Filpkart may have huge number of customer's all throughout. So now I have a customer name who has ordered on a Filpkart and his street or her street and the city okay where he or she stays.

Now which means to say that customer as per this protocol customer name is dependent on customer street and city which is not in the BCNF or because it does not satisfy rule number 1 which means Y is in X it is not a subset of X okay. Customer name is not a subset of X okay but if you have something like this. For example order ID of customer order okay customer street and customer city okay what does it mean?

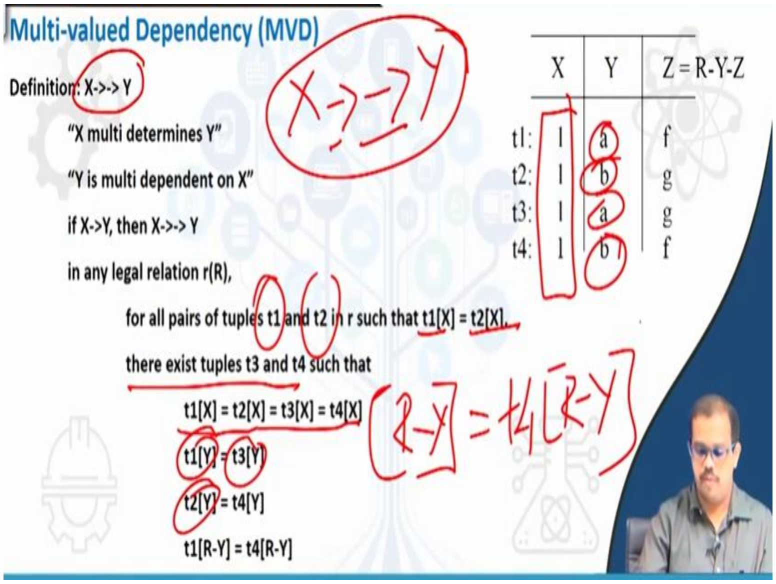
Now the order of a particular item is dependent on the street and the city where it is being delivered. There are so many items which may not be delivered right. So this satisfy the rule number 1 which means to say that Y is in X it is subset of these functions okay. So this is an example of being a Boyce- Codd normal form okay now another example very good example that can be taken is a loan schema okay.

If you have a branch name, customer name, loan number and amount this also cannot be an in the BCNF or in the third normalize form why? Basically does not satisfy both the rules for example the rule. Roll number cannot be any super key here for example what are you trying to define here you are trying to define Y, X a super key 1 and super key 2. Super key 2 can be an 1 example of where it can be used but when you are looking at super key 1 which is an loan number this particular loan number can be for both wife and husband.

When the super key is for both wife and husband it means to say that it is duplicating. So if it is duplicating your entire card number forms fails. So this cannot be in the third normal form if it was just a branch name, customer name and amount and it could have been in a third normal form because the branch name is dependent on I mean if you look at Y rather than branch name if I say amount is Y, customer name, let us say some other attributes which has a super kiosk.

So then it would have been in the second roll which would have been satisfied the roll number 2 okay. So that would be in the BSNF form these are some examples that I am trying to tell you in the real life how you put it in the normalized form okay.

(Refer Slide Time: 26:40)



Multi-valued Dependency (MVD)

Definition: X->-> Y

"X multi determines Y"

"Y is multi dependent on X"

if X->Y, then X->-> Y

in any legal relation r(R),

for all pairs of tuples t1 and t2 in r such that t1[X] = t2[X],

there exist tuples t3 and t4 such that

t1[X] = t2[X] = t3[X] = t4[X]

t1[Y] = t3[Y]

t2[Y] = t4[Y]

t1[R-Y] = t4[R-Y]

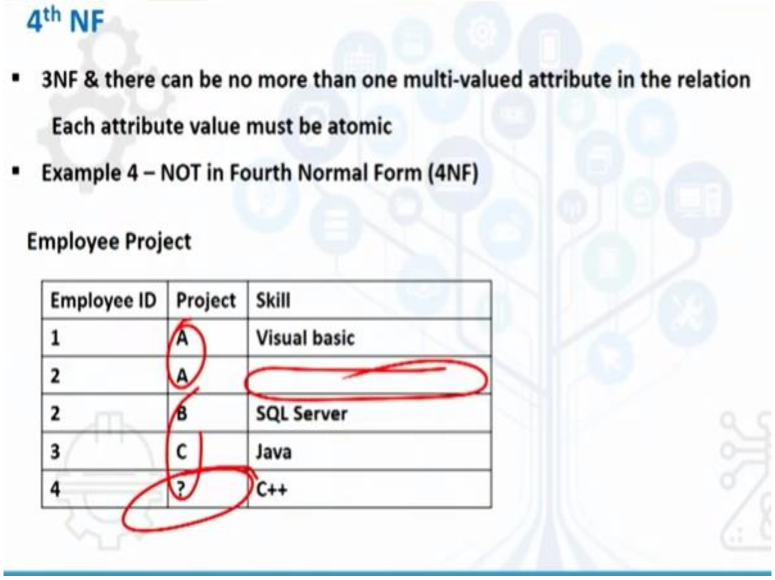
	X	Y	Z = R-Y-Z
t1:	1	a	f
t2:	1	b	g
t3:	1	a	g
t4:	1	b	f

So then there is in the fourth normal form the basic thing is it has to be in the third normal form and it has it can be 1 multivalued dependency when I say multivalued dependency for example when with definition X if X is represented something like this then it is called multivalued 1 value 2 value. So it has a multivalued function X multi determine Y is multi determined by or dependent on X okay. So if X if X or let us say if it is functionally Y is functionally dependent on X then X is multi determined by Y okay.

So determines Y sorry so when we are looking at this if you have any legal relationship all pair of tuples t1 and t2 if you have 2 tuples 2 rows which are actually having a particular primary key and attribute data. So if there are 2 tuples t1 and t2 in a particular relation the such that t1X is equal to t2X there exist tuples 3 and 4 such that okay if you look this particular thing I am trying to explain this t1X = t2X = t3X = t4X if you consider something like this because it is a multivalued function.

t1Y is equivalent to the t3Y and t2Y is equivalent to t4Y t1 of r – Y the super set is equivalent to t4 of r – Y which is again a superset okay. This is what is called as multivalued dependencies okay if you go back and try to understand how a this dependency is develop you will be able to understand this particular normal form in more detail.

(Refer Slide Time: 28:44)



4th NF

- 3NF & there can be no more than one multi-valued attribute in the relation
  Each attribute value must be atomic
- Example 4 – NOT in Fourth Normal Form (4NF)

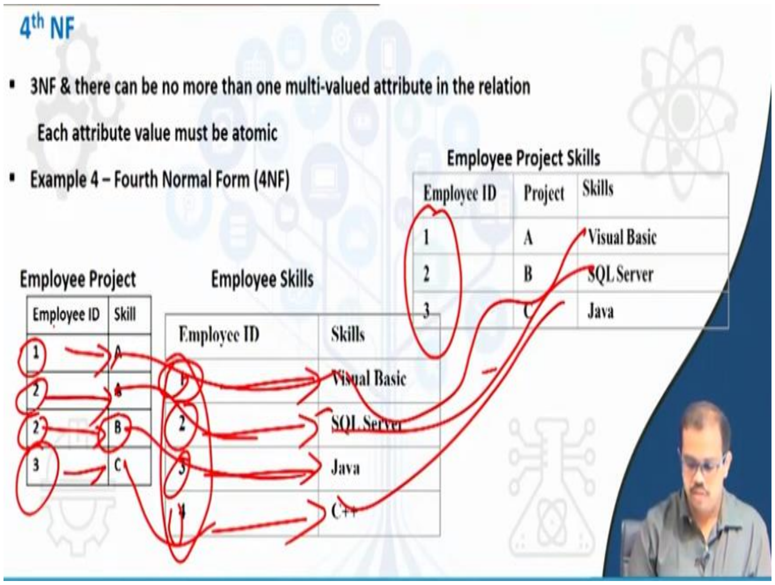
Employee Project

Employee ID	Project	Skill
1	A	Visual basic
2	A	
2	B	SQL Server
3	C	Java
4	?	C++

That is a fourth normal form for example if I have some database like this I have an employee ID which is 12234 then I have the project there are there is 2 different projects running okay. So please mind here what does fourth normal form says? Fourth normal form says the first thing it has to be in the third normal form okay first of all it is not in the third normal form very clearly it can be seen and no more than multivalued attribute in the relationship okay no more than 1 multivalued attribute.

But it also states each attribute value must be atomic when you see here attribute value is not atomic fine. So now the thing that we have to develop this into a fourth normal form how we will do this?

(Refer Slide Time: 29:37)



4th NF

- 3NF & there can be no more than one multi-valued attribute in the relation

  Each attribute value must be atomic
- Example 4 – Fourth Normal Form (4NF)

Employee Project Skills

Employee ID	Project	Skills
1	A	Visual Basic
2	B	SQL Server
3	C	Java

Employee Project

Employee ID	Skill
1	A
2	A
2	B
3	C

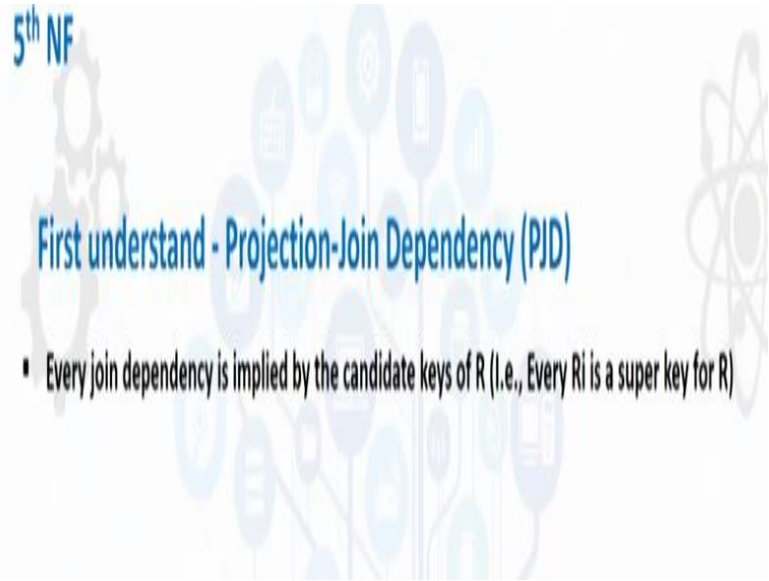
Employee Skills

Employee ID	Skills
1	Visual Basic
2	SQL Server
3	Java
4	C++

So first will develop the first subset of that particular table will develop another table which has only employee ID and the skill which means to say that employee ID number 1 as a skill A employee ID number 2 again has a skill A 2 also has skill B which means that is a multivalued function here okay and Y is functionally dependent on X then employee 3 is again has a skill C okay.

Now the same employee ID is a primary key now we are carrying forward the third normal form. So same employee ID which is the primary key here each of this 1, 2, 3, 4 refers to skills that is nothing but visual basic SQL server and this. So now it is indirectly relating to yes so now the thing as to be related to the project so use the employee ID here okay the same employee ID is now relating to the project okay. So this is how you drive the fourth normal form okay.

(Refer Slide Time: 30:58)



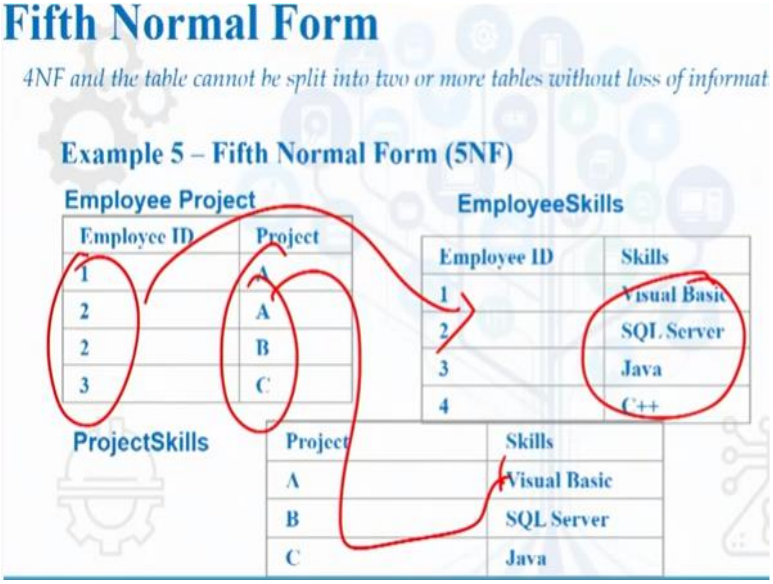
5th NF

First understand - Projection-Join Dependency (PJD)

- Every join dependency is implied by the candidate keys of R (I.e., Every Ri is a super key for R)

So then the fifth normal form is a bit complicated I would not really get into details of it but if someone is really interested please if you can understand what is projection joint dependency? So the concept of projection joint dependency is very important in terms to understand the fifth normal form so what is it state? Is every joint dependencies implied by the candidate keys of arc is every RI there are set of RI which ease a super for R so their exist always exist a super key for R that is what is the first thing of a fifth normal form.

(Refer Slide Time: 31:40)



Fifth Normal Form

4NF and the table cannot be split into two or more tables without loss of informat

Example 5 – Fifth Normal Form (5NF)

Employee Project

Employee ID	Project
1	A
2	A
2	B
3	C

EmployeeSkills

Employee ID	Skills
1	Visual Basic
2	SQL Server
3	Java
4	C++

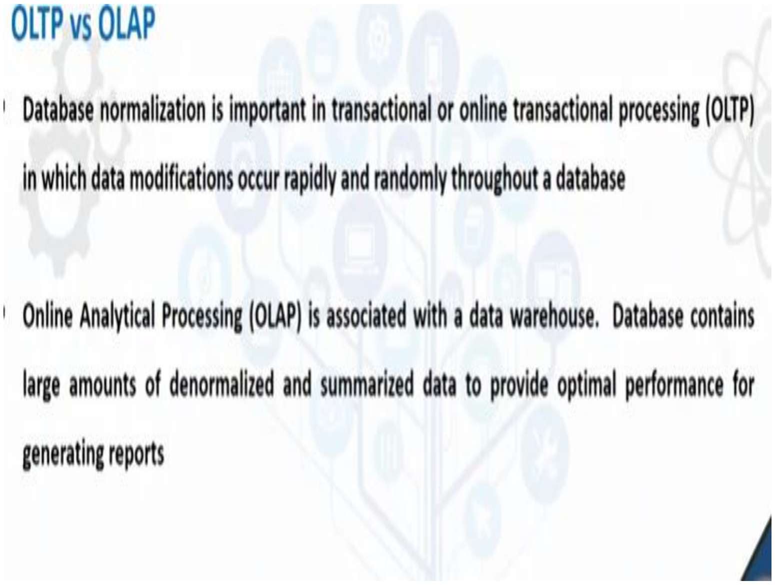
ProjectSkills

Project	Skills
A	Visual Basic
B	SQL Server
C	Java

When you are looking at it the this example is basically giving you fifth normal for only this is that there is a loss of information when you are looking at the fifth normal form. So this is an example of fifth normal form you have employee ID you have a project now the same project is giving you the skills now each of these and same employee is defining you the skills here so they are inter related that is how your fifth normal form is developed.

So if you want to do it please look at PJD so that would give you some information of or more information on how do develop a fifth normal form okay. So but I would suggest everyone learn that particular part because in case you are you want to write a exam in GIS fifth normal form can be extremely useful in terms of understanding okay.

(Refer Slide Time: 32:39)



OLTP vs OLAP

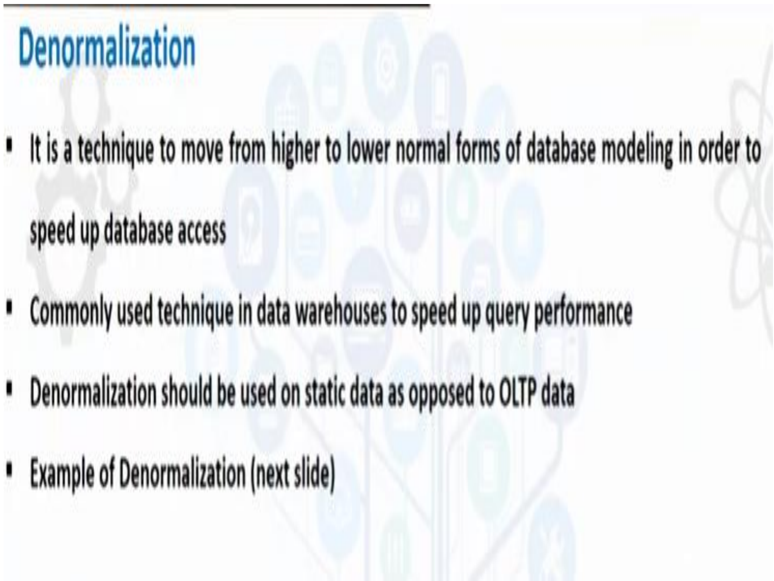
- Database normalization is important in transactional or online transactional processing (OLTP) in which data modifications occur rapidly and randomly throughout a database
- Online Analytical Processing (OLAP) is associated with a data warehouse. Database contains large amounts of denormalized and summarized data to provide optimal performance for generating reports

So yeah so this is again a example so I will not get into that then there is OLTP okay this is same example that I have shown here then the terms OLTP versus OLAP so when I say OLTP it is online transaction processing. So when you are trying to do a database normalization in whether we have to look at it whether it is transactional which is tactically transactional or online transactional processing so in which data when you say online transaction processing data modification happen very rapidly and randomly throughout the data base that is called online transaction processing okay.

So if you are looking at something where you are trying to analytically vary the entire database then it is called online analytical processing. Which means that data contains a large amount of be normalized and summarized data so you are analytically processing that particular data for optimal performance and putting it in form of a different normal forms in the database then it is called as online analytical processing okay or OLAP.

So we have OLTP and OLAP so when I say OLTP this transactional processing and when you saying it is when it is in the de-normalized form you are converting into certain ways of summarize data and trying to put it in normal if possible or not. So you are trying to put it in a online analytical processing that is a different between a OLTP and OLAP. I have just included this particular slide so that you understand the terms of OLTP and OLAP but it is not necessary that it has to be used in terms of processing okay.

(Refer Slide Time: 34:28)



Denormalization

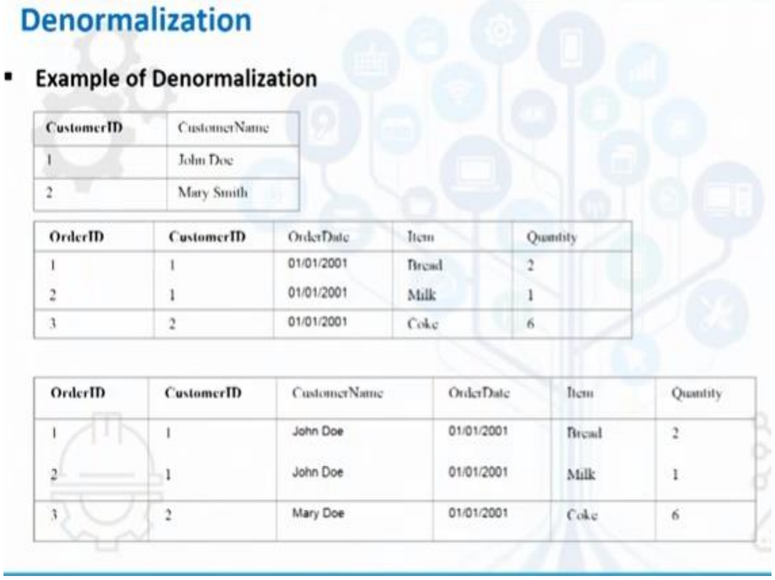
- It is a technique to move from higher to lower normal forms of database modeling in order to speed up database access
- Commonly used technique in data warehouses to speed up query performance
- Denormalization should be used on static data as opposed to OLTP data
- Example of Denormalization (next slide)

So the last thing that I would discuss here is the de-normalization when I say de-normalization now we have understood normalization which means that we have a database that is generated whatever the data that is generated then we have put it such as certain form so that it is accessible quickly understood and retrieve very fast. So now de-normalization means whatever you have done with the normalization you are converting back into the data that is what is de-normalization.

So it is to technique to move from higher to lower normal forms of a database modeling in order to speed up the database access many of the time the fourth normal form may take a bit of higher time why because if it is to it has lot of connection then reaching the nth connection may take a bit of time but not in today’s computational context okay. When you are looking at this commonly used technique in data warehouses to speed up query processing so you try to de-normalize and do the query processing.

De-normalization should be used mostly on static data as opposed to OLTP data so it is more on OLAP data itself okay. So or it is more on static data the other then OLTP and OLAP data it is more on static data.

(Refer Slide Time: 35:54)



Denormalization

- **Example of Denormalization**

CustomerID	CustomerName
1	John Doe
2	Mary Smith

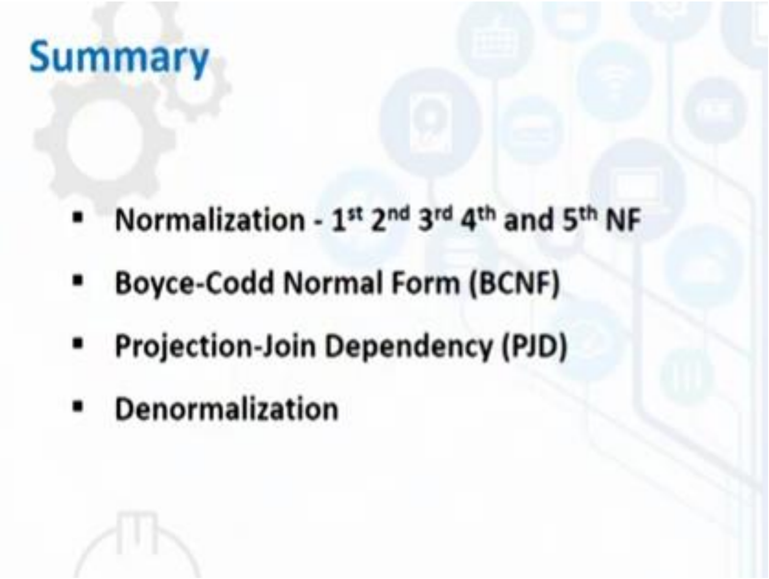
OrderID	CustomerID	OrderDate	Item	Quantity
1	1	01/01/2001	Bread	2
2	1	01/01/2001	Milk	1
3	2	01/01/2001	Coke	6

OrderID	CustomerID	CustomerName	OrderDate	Item	Quantity
1	1	John Doe	01/01/2001	Bread	2
2	1	John Doe	01/01/2001	Milk	1
3	2	Mary Doe	01/01/2001	Coke	6

So when we are looking at example of de-moralization this example can taken into context. Now if you have this is one example that i was showing in the first and second normal forms okay this is normalized data which means to say that see for example this is atomic okay this and also it is in the second normal form a basically dependent on your primary key. Now the same thing if it is put it back in the same order so it is atomic also you have processed from the second normal form to the first normal form which means you are de-normalizing by 1 extent so that is what called as a de-normalization.

You are getting back the entire database from second normalization or third similarly if you do it from fourth third to second then two steps. If you are doing from 4 to 1 then it is 3 starts the normalization so that is how de-normalization works okay.

(Refer Slide Time: 36:50)



Summary

- Normalization - 1st 2nd 3rd 4th and 5th NF
- Boyce-Codd Normal Form (BCNF)
- Projection-Join Dependency (PJD)
- Denormalization

So in summary we looked at normalization form the first, second, third, forth and I did not explain the fifth normal form but I just gave you a brief overview of what is a fifth normal form. Then we looked at Boyce- Codd normal form for the third normal form aspect please look at projection joint dependency I have not spoken here because it itself is a topic that can be easily dealt for particular entire week.

So please look at projection joint dependency which is also very which is extremely important if in case you are trying to understand the fifth normal form. Then we looked at de-normalization I have an example of how you de-normalize from second normal form to the second normal form okay. Any way you try to do it from any step to the any step so that is nothing but a de-normalization.

So with this I would end this week introduction to databases so in the next week we would look into more aspects including how do we query what kind of query we can do what kind of analysis we can do on the database what are different analysis that we can do and if you are looking at query languages what are different query languages and how they are useful so this is what I would speak in the next entire module of the next week so that if you have understood databases then querying is other part. So both of these can be extremely useful in terms of if you are trying to become an advanced GIS user so until then have a nice time thank you very much.