



IIT Madras

ONLINE DEGREE

Mathematics for Data Science 1.
Professor Madhavan Mukund.
Department of Computer Science
Mathematical Institute, Chennai.

Lecture-7A.

Relations: Examples.

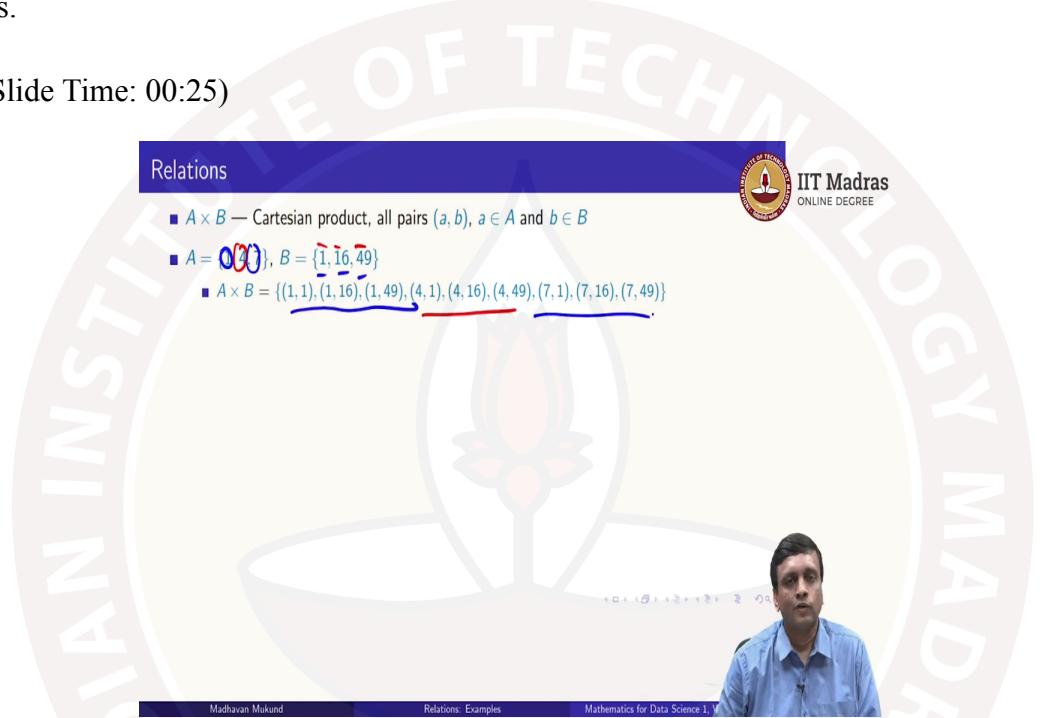
So, earlier we defined relations as subsets of elements of a Cartesian product which have special properties. So, let us take a look at relations again and understand why we are so interested in relations.

(Refer Slide Time: 00:25)

Relations

IIT Madras
ONLINE DEGREE

- $A \times B$ — Cartesian product, all pairs (a, b) , $a \in A$ and $b \in B$
- $A = \{1, 4, 7\}$, $B = \{1, 16, 49\}$
- $A \times B = \{(1, 1), (1, 16), (1, 49), (4, 1), (4, 16), (4, 49), (7, 1), (7, 16), (7, 49)\}$

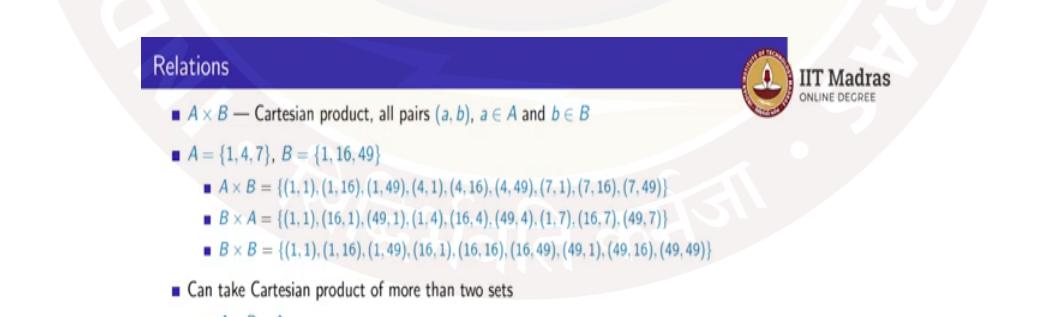


Madhavan Mukund Relations: Examples Mathematics for Data Science 1, V

Relations

IIT Madras
ONLINE DEGREE

- $A \times B$ — Cartesian product, all pairs (a, b) , $a \in A$ and $b \in B$
- $A = \{1, 4, 7\}$, $B = \{1, 16, 49\}$
- $A \times B = \{(1, 1), (1, 16), (1, 49), (4, 1), (4, 16), (4, 49), (7, 1), (7, 16), (7, 49)\}$
- $B \times A = \{(1, 1), (16, 1), (49, 1), (1, 4), (16, 4), (49, 4), (1, 7), (16, 7), (49, 7)\}$
- $B \times B = \{(1, 1), (1, 16), (1, 49), (16, 1), (16, 16), (16, 49), (49, 1), (49, 16), (49, 49)\}$
- Can take Cartesian product of more than two sets
 - $A \times B \times A = \{(1, 1, 1), (1, 1, 4), (1, 1, 7), (1, 16, 1), (1, 16, 7), \dots, (7, 49, 1), (7, 49, 16), (7, 49, 49)\}$
- A relation picks out certain tuples in the Cartesian product
 - $S \subseteq A \times B = \{(1, 1), (4, 16), (7, 49)\}$
 - $S = \{(a, b) \mid (a, b) \in A \times B, b = a^2\}$



Madhavan Mukund Relations: Examples Mathematics for Data Science 1, V

So, remember that a Cartesian product takes all pairs of elements from a collection of sets. In particular, if you say A cross B, you are taking 2 sets A and B, and you are taking every pair of elements of the form small a small b such that the first small a comes from capital A and small b comes from capital B. The order is important, the first element in the pair comes from the first set, the second comes from the second set. So concretely, let us look at these 2 sets.

So, suppose $A = \{1, 4, 7\}$, so it has 3 elements, and $B = \{1, 16, 49\}$. So, if you now look at $A \times B$, it looks at every pair. So, if you can take this one, and combine it with 1, 16 and 49 to get this. Then you can take this 4 and combine it again with 1, 16 and 49 to get these for 3 pairs, and finally you take 7 and then you combine it again with 1, 16 and 49 to get these pairs.

So, it is easy to see that if you have m elements in the first and n elements in the second, every one of those m elements is paired with every one of the n elements, so you get $m \times n$ pairs. Now, the first thing to remember is that the Cartesian product is ordered. So, there is a first and there is a second. So, if you reverse this and say $B \times A$, you do not get the same set of pairs, every pair is reverse. So, $(16,1)$ replaces $(1,16)$, $(49,1)$ replaces $(1,49)$. So, this is the first thing to remember about Cartesian products.

The other thing to remember is that there is no relation, there is no constraint on what you can take the Cartesian product of. You can very easily take the Cartesian product of a set with itself. So, the set to itself is not just pairs of identical elements, but also pairs of non identical elements. So, if you take $B \times B$, you get Of course, $(1,1)$, $(16,16)$, $(49,49)$. But you also get the dissimilar pairs like $(1,16)$, $(16,49)$, $(49,16)$, and so on.

So, this is an example with 2 sets, but there is nothing to restrict us to 2 sets. So, in general, a Cartesian product can take a large number of sets and gives us tuples. So, for instance, if we take 3 sets, we get these triples, each element has 3, each element in the Cartesian product has 3 elements in order.

So, here for instance, if I do $A \times B \times A$, I take every element in A, combine it with every element in B and then with A again. So, I have 1 from A, 1 from B and 1 from A. Then I have 1 from A, 1 from B and 4 from A, the second copy of A and the first copy of A are different.

So, I have $(1,1,1)$, $(1,1,4)$, $(1,1,7)$, then I move to the second element of B, I have $(1,16,1)$, $(1,16,7)$. Now, ultimately the Cartesian product is a set, so it does not matter in what order I write these triples. But to order to write them down systematically, it is convenient to write them down in this particular way, where we go through each set one by one, otherwise, we may miss out on something. So, the reason we need Cartesian products is because they are the building blocks of relations.

So finally, what we want is not all these pairs or triples, but some of them which are of interest to us. So, for example, from the first Cartesian product $A \times B$, we may be interested in the pairs where each element from A is paired with a corresponding position B. So, the first element in A is paired only with the first position in B, second with the second and so on. So, we might want to say that we want S, a set which is a subset of $A \times B$, which from those 9 different pairs picks out only 3 of them of interest, $(1,1)$, $(4,16)$ and $(7,49)$.

Now, if as in this case, there is some way of describing this, which is more abstract, you can also use a set comprehension. So, we can talk in terms of positions or observe that in this particular case, the second element is always a square of the first element. So, we could also write this as the set of pairs (a,b) , where (a,b) comes from $A \times B$, so we are generating every possible pair in the Cartesian product.

But then we are filtering, remember that we had these filters, so we are filtering it so that we only retain those pairs for which the second component B is the square of the first component. So, this is how relations are defined. They are typically defined as subsets of the Cartesian product. And we can either write out the subset explicitly or try to express it implicitly using the set comprehension notation.

(Refer Slide Time: 4:57)

Examples of relations



- Divisibility

- Pairs of natural numbers (d, n) such that $d \mid n$

Madhavan Mukund

Relations: Examples

Mathematics for Data Science 1, W



So, we saw some examples. So, let us look at these examples again more carefully, some examples from numbers. So, divisibility is an important relation when we are talking about natural numbers or integers. So, divisibility talks about pairs of natural numbers, such that the first one divides the second one. So, we want (d, n) such that d divides n , remember this notation, this perpendicular bar for numbers denotes, this is not the same as the one that we use in set comprehension.

So, here it is an operation, arithmetic operation which says d divides n , so if I divide n by d , there is no remainder, it is a 0, d perfectly divides n . So, this would have this divisibility relation would have pairs like $(7, 63)$ because $7 \times 9 = 63$, or $(17, 85)$, because $17 \times 5 = 85$, and so on. So, we have a large number of pairs of divisors and numbers which the divisors divide equally, evenly. So, this we can write in our set comprehension notation because this is an infinite set, so we have no other way of listing everything.

(Refer Slide Time: 06:09)

Examples of relations



■ Divisibility

- Pairs of natural numbers (d, n) such that $d|n$
- Pairs such as $(7, 63), (17, 85), (3, 9), \dots$
- $D = \{(d, n) \mid (d, n) \in \mathbb{N} \times \mathbb{N}, d|n\}$

Madhavan Mukund

Relations: Examples

Mathematics for Data Science 1, W



So, we take all pairs $\mathbb{N} \times \mathbb{N}$, (d, n) , such that $d | n$. So, this is our filter. So, we want to generate everything of this form, but filter out under the condition that d must be a divisor of n and keep all such pairs. And this we can call d , the divisibility relation.

(Refer Slide Time: 6:2 1)

Examples of relations

IIT Madras
ONLINE DEGREE

- Divisibility
 - Pairs of natural numbers (d, n) such that $d|n$
 - Pairs such as $(7, 63), (17, 85), (3, 9), \dots$
 - $D = \{(d, n) \mid (d, n) \in \mathbb{N} \times \mathbb{N}, d|n\}$
 - Can also extend to integer divisors
 - $E = \{(d, n) \mid (d, n) \in \mathbb{Z} \times \mathbb{N}, d|n\}$
 - Now $(-7, 63), (-17, 85), (-3, 9), \dots$ are also in E
- Prime powers
 - Pairs of natural numbers (p, n) such that p is prime and $n = p^m$ for some natural number m
 - Examples: $(3, 1), (5, 625), (7, 343), \dots$

$3^0 = 1 \quad n^0 = 1$

Madhavan Mukund Relations: Examples Mathematics for Data Science I, V

Now, this is the relation on pairs of natural numbers, so we only get positive divisors. If we extend it to integers, then we will get even negative divisors. We know that $(-7) \times (-9)$ is also 63, because the 2 negative signs will cancel out. So, if you extend the generating set from \mathbb{N} to \mathbb{Z} , from the natural numbers to the integers, then we get a larger set of divisor pairs. So, we get minus and plus elements for the same pairs that we had in the original relation.

Here is another example. Let us look at what we call prime powers. So, a prime power is something that is a prime multiplied by itself for a certain number of times. So, for instance, we can say that $5^5 = 3125$. So, $5^2 = 25$, 5^4 rather, $5^2, 5^3 = 125$, and $5^4 = 625$. So, 625 is a prime power, similarly $343 = 7^4$, so it is a prime power and so on. Why is $(3, 1)$ in this relation because anything to the power 0 is 1 by definition. So, $3^0 = 1$, in fact, anything to the power, so any number to the power 0 is 1. This is by definition. So, for every number comma 1 will be a prime power.

(Refer Slide Time: 7:41)

The slide is titled "Examples of relations". It contains two main sections:

- Divisibility**
 - Pairs of natural numbers (d, n) such that $d|n$
 - Pairs such as $(7, 63), (17, 85), (3, 9), \dots$
 - $D = \{(d, n) \mid (d, n) \in \mathbb{N} \times \mathbb{N}, d|n\}$
 - Can also extend to integer divisors
 - $E = \{(d, n) \mid (d, n) \in \mathbb{Z} \times \mathbb{N}, d|n\}$
 - Now $(-7, 63), (-17, 85), (-3, 9), \dots$ are also in E
- Prime powers**
 - Pairs of natural numbers (p, n) such that p is prime and $n = p^m$ for some natural number m
 - Examples: $(3, 1), (5, 625), (7, 343), \dots$
 - First define primes: $P = \{p \mid p \in \mathbb{N}, \text{factors}(p) = \{1, p\}, p \neq 1\}$
 - Prime powers: $PP = \{(p, n) \mid (p, n) \in P \times \mathbb{N}, n = p^m \text{ for some } m \in \mathbb{N}\}$

At the bottom right, there is a video player showing a person speaking. The video player has a progress bar and a timestamp of 7:41.

So, if you want to define prime powers, it is useful to first define primes. So, one way we can define primes is to say, give me a natural number, such as the factors of the natural number consists of exactly 2 elements, 1 and the number itself. And because in sets, we do not distinguish duplicates, in this definition, if I just say $\text{factors}(p) = \{1, p\}$, it includes a case where p is 1, because $\text{factors}(1) = \{1, 1\}$, which is just 1. But I do not want to count 1 as a prime number. So, we also specify that P is not 1. So, this is the set of primes.

And now, we can say the set of prime powers is the set of all pairs in $P \times \mathbb{N}$, where P is defined above, $P \times \mathbb{N}$, such that n is the power of p . So, $n = p^m$ for some m , which is a natural number, which could be 0. That is why we get $(3, 1)$. So, this is an example that we also talked about. It is saying that when you are writing the set comprehension, you can write these kinds of statements.

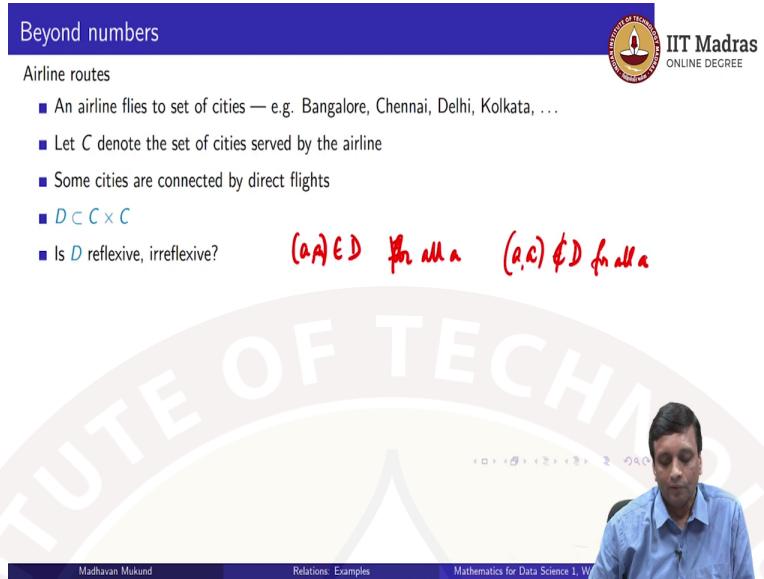
So, you do not have to be very precise about what you are writing mathematically in terms of notation, as long as the understanding is clear, there is no ambiguity about what you mean. So, you can write words like for some, you can also write it in a mathematical notation using symbols for there exists and for all and so on, but it is not necessary. As long as you are precise, you can use set comprehension notation in a flexible way.

(Refer Slide Time: 9:00)

Beyond numbers

Airline routes

- An airline flies to set of cities — e.g. Bangalore, Chennai, Delhi, Kolkata, ...
- Let C denote the set of cities served by the airline
- Some cities are connected by direct flights
- $D \subseteq C \times C$
- Is D reflexive, irreflexive?
 $(a,a) \in D \text{ for all } a \quad (a,a) \notin D \text{ for all } a$



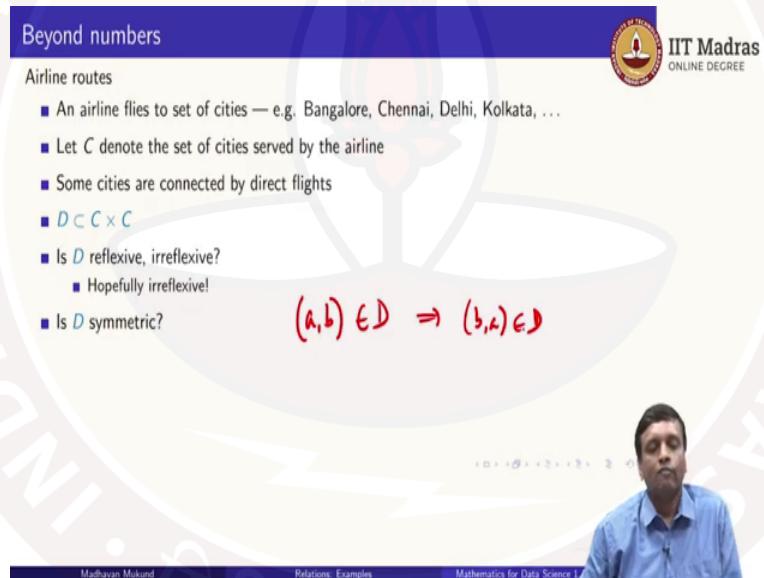
Madhavan Mukund Relations: Examples Mathematics for Data Science 1, W

IIT Madras
ONLINE DEGREE

Beyond numbers

Airline routes

- An airline flies to set of cities — e.g. Bangalore, Chennai, Delhi, Kolkata, ...
- Let C denote the set of cities served by the airline
- Some cities are connected by direct flights
- $D \subseteq C \times C$
- Is D reflexive, irreflexive?
 - Hopefully irreflexive!
- Is D symmetric?
 $(a,b) \in D \Rightarrow (b,a) \in D$



Madhavan Mukund Relations: Examples Mathematics for Data Science 1, W

IIT Madras
ONLINE DEGREE

Airline routes

- An airline flies to set of cities — e.g. Bangalore, Chennai, Delhi, Kolkata, ...
- Let C denote the set of cities served by the airline
- Some cities are connected by direct flights
- $D \subset C \times C$
- Is D reflexive, irreflexive?
 - Hopefully irreflexive!
- Is D symmetric?
 - If there is a direct flight from Bangalore to Delhi, is there always a direct flight back from Delhi to Bangalore
 - For bigger cities, yes
 - For smaller cities, may have a triangular route Chennai → Madurai → Salem → Chennai



Madhavan Mukund

Relations: Examples

Mathematics for Data Science 1, W

So, these are relations in a formal sense. But why are we so interested in relations especially in the context of computing and data. So, let us look at relations which go beyond numbers. So, here is an example. Supposing we are talking about an airline, which serves a set of cities and we are interested in the routes that this airline serves. So, let us C be the set of cities where the airline operates. So clearly, the airline operates between some pairs of cities, but not all of them.

So, some of these cities are connected by direct flights and for other situations, you have to take a hopping flight which goes from city A to city B and then from city B to C. So, let us look at that subset D of direct flights between cities in C . So, this is an example of a relation. Not every pair of cities is connected by a direct flight. So, if you take all possible pairs of cities, some of them are connected by direct flights, and some are not. So, this way, information about an airline's route is really a relation in the sense that we mean.

Now, we have defined certain properties of relations, we said that the relation is reflexive. Now, this is useful to ask this question because we are talking about a relation between a set and itself. So, we can ask whether every element in the set is related to itself or is not related to itself. So, reflexive means that always we have (a,a) in D , for all, for every a . And irreflexive means, exactly the opposite of this is never in D and for all A .

So, the question is, in terms of direct flights, is this going to be a reflexive relation and irreflexive relation or neither. Well, it is easy to see that this should not be reflexive. Because we do not expect an airline to actually operate a flight which takes off from an airport and then lands immediately in the airport. And in fact, we would precisely like it to be irreflexive, that is, this should never happen.

So, this should not be reflexive because we do not want every airport to serve itself and we want it to be irreflexive because we want no airport to serve itself. So, this is an example of an irreflexive relation. Now, is it a symmetric relation? So, symmetric relation says that whenever I have a pair of cities in the relation, then I will also have the reverse pair in the relation. So, if I can fly from one city to another directly, then I can also fly back.

So, concretely for instance, if I take any 2 cities and suppose there is a direct route from Bangalore to Delhi, then is there always a direct flight back from Delhi to Bangalore. Now, if you think about airlines, this is usually the case. But actually, if you look at domestic flights in particular, this is typically true only for the bigger cities, it will certainly be true for all the metro cities and the largest state capitals and so on. But if you look at smaller cities, this is not necessarily the case.

For instance, it is quite common for airlines to serve 3 cities in a triangular route. So, you might have a flight that takes you from Chennai to Madurai, but if you want to come back from Madurai to Chennai, you cannot fly back directly, but you may have to fly to Salem and then come. So, between these 3 cities you can get from one to another, either directly or indirectly depending on which direction you are going. So, this relation is going to be irreflexive but not necessarily symmetric, it depends on the context.

(Refer Slide Time: 12:29)

Tables as relations



- Flying distances between cities

Source	Destination	Distance (km)
Bangalore	Chennai	290
Chennai	Delhi	1752
Delhi	Bangalore	1735
Delhi	Chennai	1752
...

- Table is a relation: $\text{Dist} \subseteq C \times C \times \mathbb{N}$
- Some entries are useless: (Delhi, Delhi, 0)
- Restrict to cities served by direct flights
 $\text{Dist} = \{(a, b, d) \mid (a, b) \in D, d \text{ is distance from } a \text{ to } b\}$
- Distances are symmetric, even if D is not
- Save space by representing only one direction in the table



Madhavan Mukund

Relations: Examples

Mathematics for Data Science I, V

Now, one thing you can do is to extend this to a table. So, here is a useful table that we might want to keep, which might be used to derive other things such as how long it takes to fly or how expensive a ticket is like to be. So, here we are just recording a fact which is what is the flying distance between a pair of cities. So, this table says that if the source is Bangalore and the destination is Chennai, it is 290 kilometers, whereas if the source is Chennai and the destination is Delhi, it is 1752 kilometers.

So, for every direct flight which our airline operates, you can record this distance and put it in a table. So, what is important to recognize and this is why relations are so useful in computing and data is a table is just a relation. So, every column represents a potential set of values. Here, the first column represents a possible city, so it is taken from the set C , the second column is also taken from the set C , the third column is a natural number.

If you take pairs of cities which are the same, you could put 0, so it could be from Delhi to Delhi it is 0. So, in general, you have all possible pairs of cities and all possible numbers, but only some of them are interesting. Namely, when I have 2 cities which are actually connected by a flight and the distance the number is actually the real distance. So, it is a relation on $C \times C \times \mathbb{N}$.

As we said, some relations are useless so we would not record them even though we know them. We know that for every city, the flying distance from the city to itself is 0, so there is no reason

to record it in the table. The other thing is that unlike our direct flight's relation, this is actually a symmetric relation. So, first of all, we will only keep direct flights because we do not want indirect flights. But distances are definitely symmetric.

So, it doesn't really matter whether there is a direct flight from Chennai to Delhi and back or whether there is a direct flight from Chennai to Madurai and not back. It is enough to record the distance from Chennai to Delhi and Chennai to Madurai once each. I do not have to keep the distance from Delhi to Chennai separately as you can see above, in this example, Chennai to Delhi and Delhi to Chennai are both exactly the same distance 1752 because that is how distances work, distances are symmetric.

So, if we have symmetric entries, in a practical sense, when we represent a relation as a table, we can save on space by not recording the symmetric entries and making a note separately that this relation is symmetric. So, that is why it is important to know the property of the relation. It is not just an abstract question, is this reflexive, is this irreflexive, it is actually a practical consideration, a symmetric relation can be represented by only half the entries in the relation, the other half followed by symmetry.

(Refer Slide Time: 15:12)

Tables as relations ...



IIT Madras
ONLINE DEGREE

Roll no	Name	Date of birth
A71396	Abhay Shah	03-07-2001
B82976	Payal Ghosh	18-06-1999
F98989	Jeremy Pinto	22-02-2003
C93986	Payal Ghosh	14-05-2000
...

- Some columns are special — each student has a unique roll number
 - Such a column is called a **key**
 - Name is not a key, in general
- Given the roll number, can retrieve the data for a student
 - Function from Roll Numbers to (Name, Date of Birth)
 - (key,value) pairs



Madhavan Mukund Relations: Examples Mathematics for Data Science I, V

So, let us go further with this. So, another place where we often encounter tables are, for instance, when looking at data about people. Let us look at students. So, typically a college would record or a school would record information about students in this form. So, they would assign a roll number, then they would record maybe the name, the date of birth, and there would typically be other personal information like maybe their home address, phone number, and so on.

So here, what is important is that some columns are not natural in the sense. So, we know that everybody has a name and they are born on a particular date, but this roll number is actually assigned to them by the school or college. And this is something which is designed to be unique, so no 2 students get the same roll number. So, this kind of column is called a key. And this is because we want to identify, define each student directly and individually without getting confused about which student we are talking about.

And unfortunately, the other columns are not keys, 2 students could have the same name. And it is even possible for 2 students to have the same name and the same date of birth. So, we cannot rely on the fact that the other columns will uniquely distinguish. So now, if we have a unique roll number for every student, then each row is identified by the roll number. So, we can actually

think about the row as being something where if I give you the roll number, you can tell me which row it is and give me the other values in that thing.

So, this is more like a function. A function says given an input give me a unique output. So, given a roll number, tell me all the values associated with the roll number, the name, the date of birth, and so on. So, this kind of a stored table is also called sometimes a set of key value pairs, given the key there is a unique value. I can change the value for a given key by updating it. But if I add a new entry, I have to add a new key so there is no confusion.

(Refer Slide Time: 17:03)

The slide has a header 'Operations on relations' and the IIT Madras logo. It contains three tables:

- Table 1:** Roll No | Name | Date of birth
A71396 | Abhay Shah | 03-07-2001
B82976 | Payal Ghosh | 18-06-1999
F98989 | Jeremy Pinto | 22-02-2003
C93986 | Payal Ghosh | 14-05-2000
... | ... | ...
- Table 2:** Roll no | Subject | Grade
A71396 | English | B
B82976 | Mathematics | A
C93986 | Physics | B
B82976 | Chemistry | A
... | ... | ...
- Table 3 (Generated table):** Roll No | Name | Subject | Grade
A71396 | Abhay Shah | English | B
B82976 | Payal Ghosh | Mathematics | A
B82976 | Payal Ghosh | Chemistry | A
C93986 | Payal Ghosh | Physics | B
... | ... | ...

A note on the slide says: "■ Generate a table with roll numbers, names and grades". In Table 3, the second and third rows for Payal Ghosh are circled in red.

So, usually a school or college will maintain more than one table of this kind. For instance, there might be a separate table, where we maintain the marks of the student or the grades of a student in the courses that they do. And here for conciseness, we might keep only the roll numbers and the subject names and not the names of the students. So, for instance, in the second table, we have the roll number, subject and the grade. Here is a typical requirement when we have to generate a report card.

The grade card has, the grade table has the roll number and the subject and the grade but it does not tell us who the student is. And that is, for example, it may be difficult for an outsider who except for the student themselves to know whose roll number belongs to whom, because nobody

would recognize these strange character sequences. So, we want a table that looks like this which has the roll number and extra column with the name which is not there in the grade table which is taken from the first table and then we want the subject and the grade.

And here, we see why it is important to have keys because we have this name Payal Ghosh, which is ambiguous, there are 2 Payal Ghosh's. And in fact, they have 2 different entries in this table because they have 2 different roll numbers. So, the Payal Ghosh who got an A in mathematics is not the same as the Payal Ghosh who got a B in physics. So, this is an operation which combines these 2 tables. And remember that a table is a relation.

(Refer Slide Time: 18:23)

Operations on relations

Roll No	Name	Date of birth	Roll no	Subject	Grade)
A71396	Abhay Shah	03-07-2001	A71396	English	B
B82976	Payal Ghosh	18-06-1999	B82976	Mathematics	A
F98989	Jeremy Pinto	22-02-2003	C93986	Physics	B
C93986	Payal Ghosh	14-05-2000	B82976	Chemistry	A
...

■ Generate a table with roll numbers, names and grades
■ Join the relations on Roll No
■ $\{(r, n, s, g) | (n, d) \in \text{Students}, (r, s, g) \in \text{Grades}, r = r'\}$

Roll No	Name	Subject	Grade)
A71396	Abhay Shah	English	B
B82976	Payal Ghosh	Mathematics	A
B82976	Payal Ghosh	Chemistry	A
C93986	Payal Ghosh	Physics	B
...



IIT Madras
ONLINE DEGREE



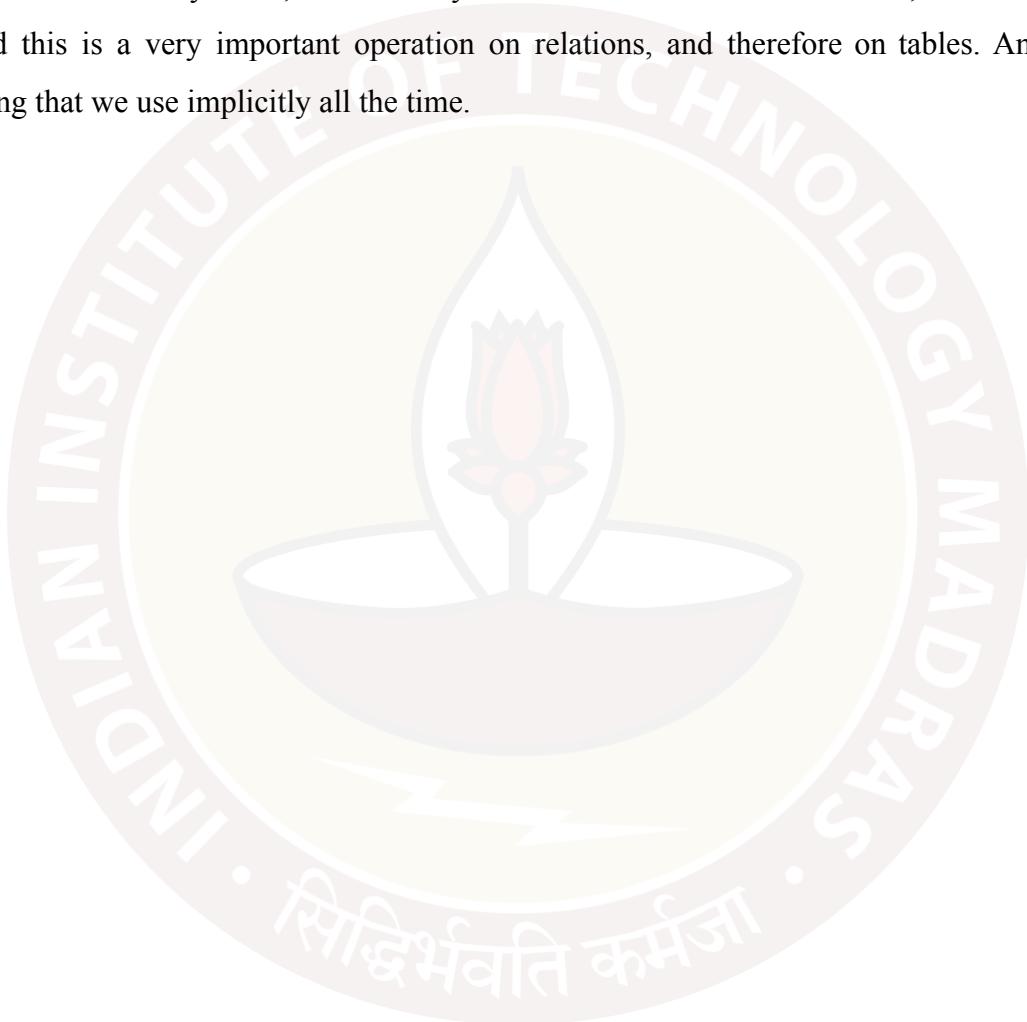
Madhavan Mukund Relations: Examples Mathematics for Data Science I, Week 1

So, this operation, which combines 2 tables is also an operation which combines 2 relations, and it is an important operation in computing and in data science called a Join. So formally, a Join takes tuples from 2 relations and combines them on common values. So here, for instance, you take any arbitrary roll number, name and date of birth from students, you take any arbitrary roll numbers subject and grade from grades, but you want that the roll number in the roll number of the 2 sides belongs the same.

So, the r comes from students and the r' comes from grades and you want $r = r'$. And if this is the case, then you put out a new tuple, which combines the n from the left hand side throws away the

date of birth, we are not interested in preserving the date of birth, keeps the n and keeps the subject and the grade s and g and of course keeps the roll number which is the same on both sides.

So, this will ensure that we do not get rows merged, where they correspond to 2 different students. So, the marks for Abhay, or the grade for Abhay will not be merged with the name and date of birth for Jeremy Pinto, because they have 2 different roll numbers. So, this is called the Join and this is a very important operation on relations, and therefore on tables. And this is something that we use implicitly all the time.



(Refer Slide Time: 19:39)



IIT Madras
ONLINE DEGREE

- A relation describes special tuples in a Cartesian product
- Data tables are essentially relations
- Combining information on tables can be described in terms of operations on relations

Madhavan Mukund

Relations: Examples

Mathematics for Data Science I, V



So, to summarize, a relation describes special tuples in a Cartesian product. And what is really important for us from a computing and data science point of view is that we work with tables all the time and tables are really relations. So, that is why relations play such a central role in many of the things that we are going to look at. So, it is important to get the terminology of relations right.

And when we combine information on tables, these are actually operations on relations such as the Join operation that we described, this is only one kind of Join we may have different types of operations, which we will see in other courses later on. But please, keep in mind that tables are relations. Thank you.