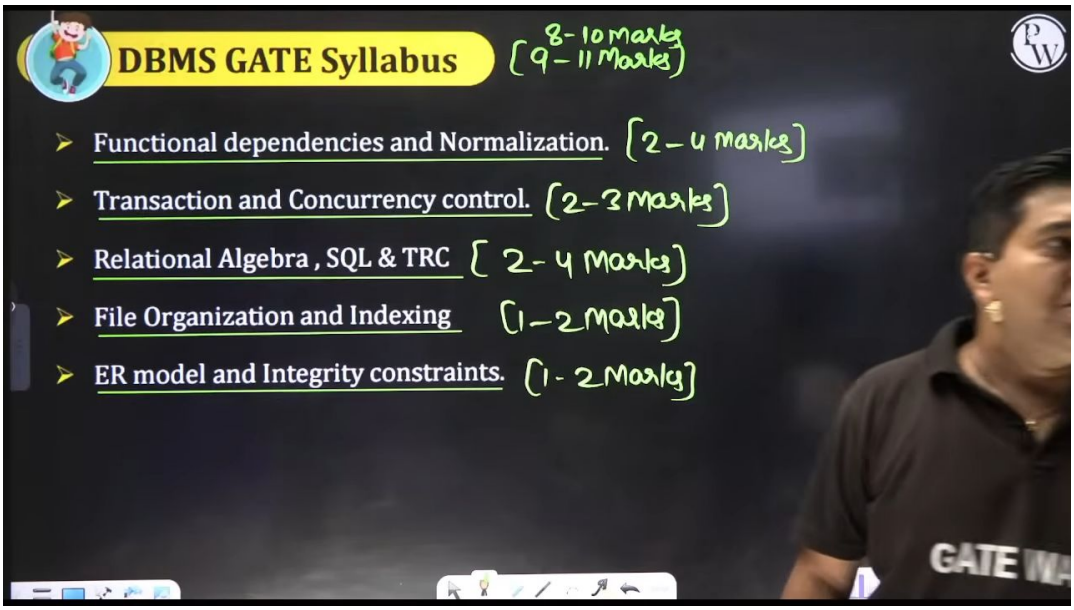
View, add and edit your notes in the app

DBMS 01 | Introduction Of RDBMS | CS & IT | GATE 2024 FastTrack Batch

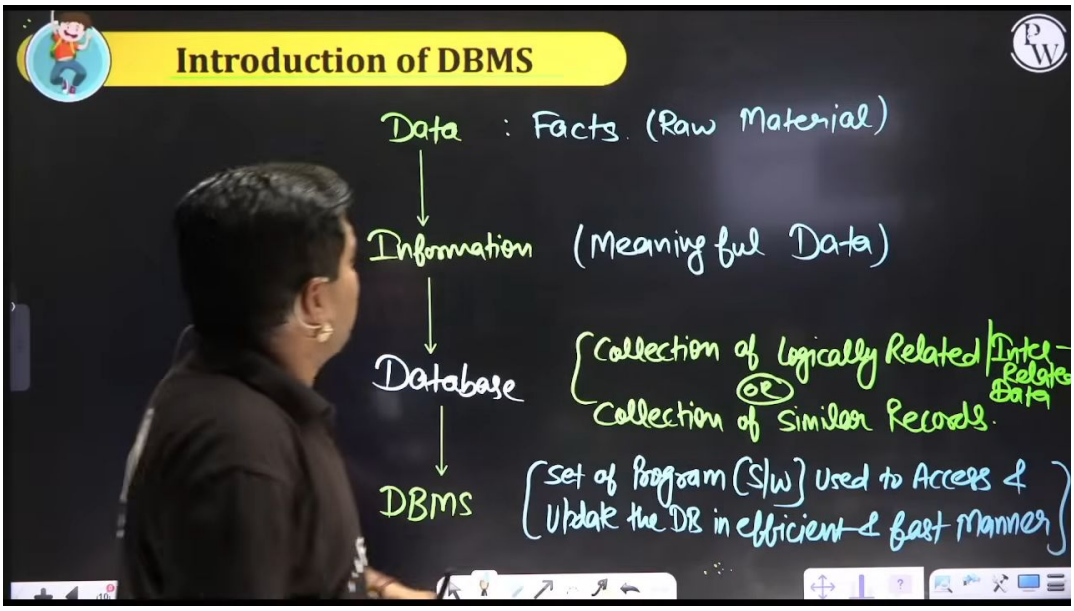
Generated on October 19, 2024

Summary

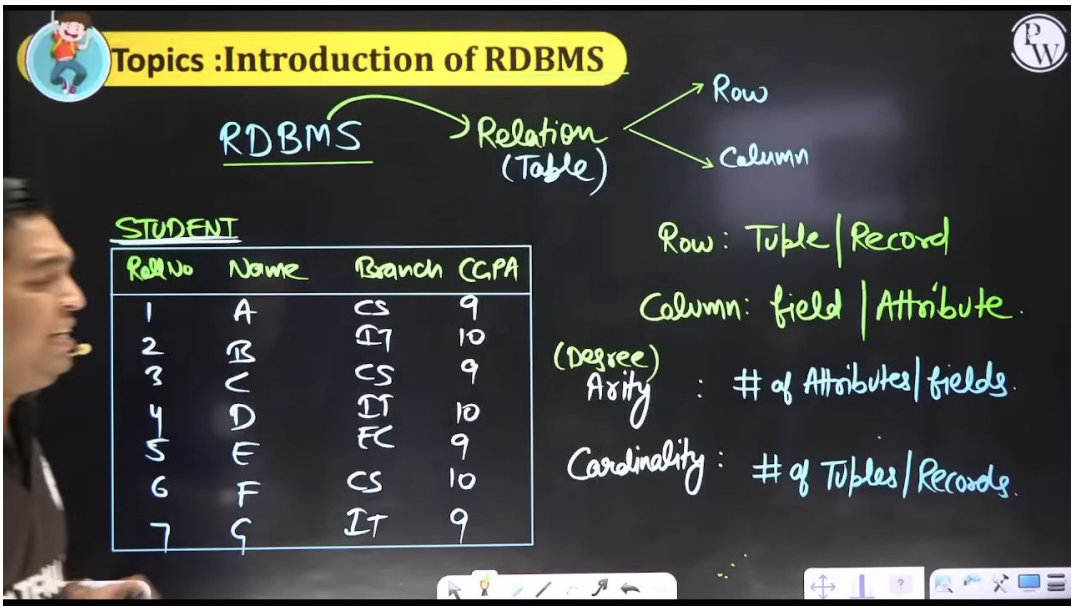
AI Notes	AI Slides	Text Notes	Screenshots
0	0	0	22



4:13

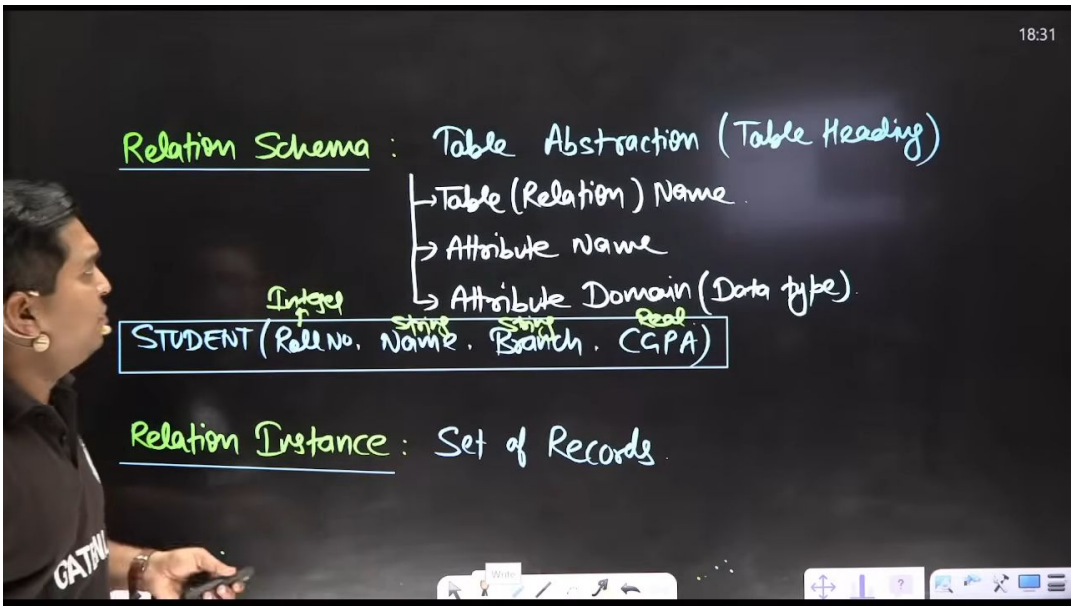


17:39

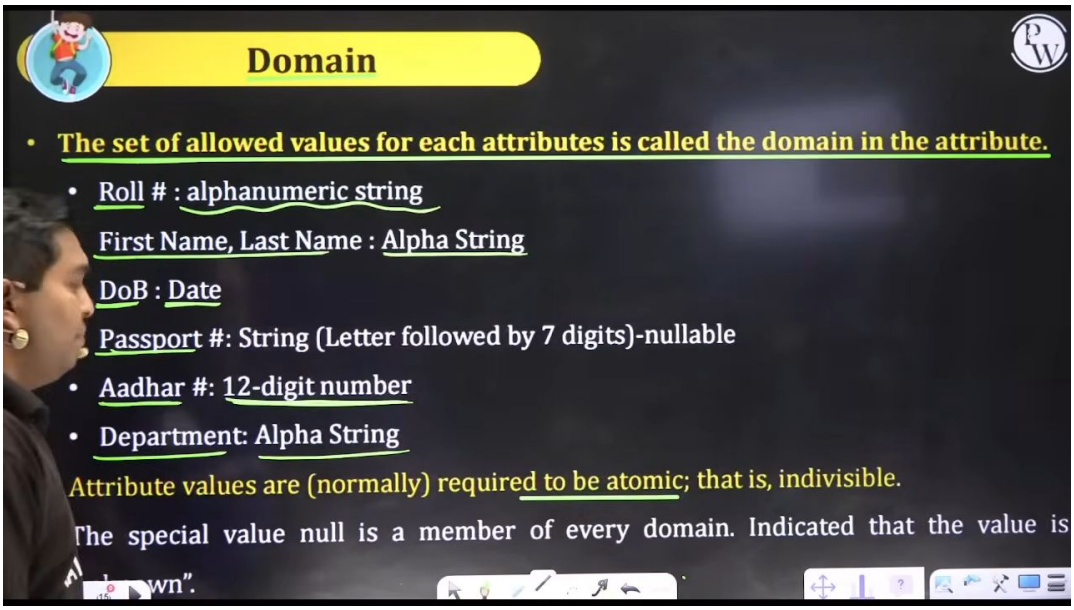


Roll No	Name	Branch	CGPA
1	A	CS	9
2	B	IT	10
3	C	CS	9
4	D	IT	10
5	E	EC	9
6	F	CS	10
7	G	IT	9

23:23


18:31
Relation Schema : Table Abstraction (Table Heading)
Table (Relation) Name
Attribute Name
Attribute Domain (Data type)
Integer
String
String
Real
STUDENT (Roll No. Name . Branch . CGPA)
Relation Instance : Set of Records

31:40


Domain
The set of allowed values for each attributes is called the domain in the attribute.
Roll # : alphanumeric string
First Name, Last Name : Alpha String
DoB : Date
Passport #: String (Letter followed by 7 digits)-nullable
Aadhar #: 12-digit number
Department: Alpha String
Attribute values are (normally) required to be atomic; that is, indivisible.
The special value null is a member of every domain. Indicated that the value is

35:04

Relation Schema and Instance

- $A_1, A_2, \ldots, An$ are attributes
- $R = (A_1, A_2, \ldots, A_n)$ is a relation schema

Example:

instructor = (ID, name, dept_name, salary)

- Formally, given sets $D_1, D_2, \ldots D_n$ a **relation r is a subset of**

  $D_1 \times D_2 \times \ldots \times D_n$

  Thus, a relation is a set of n-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
- The current values (**relation instance**) of a relation are specified by a table.
- An element t of r is a tuple, represented by a row in a table.

35:37

Relation Schema and Instance

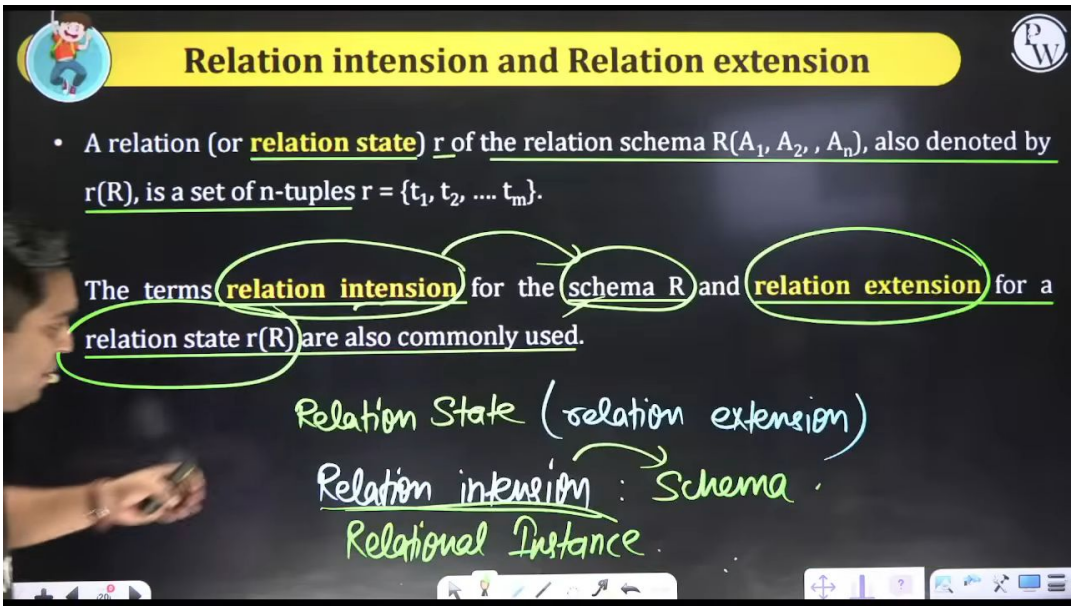
- **A relation schema R**, denoted by $R(A_1, A_2, , A_n)$, is made up of a relation name R, a list of attributes, $A_1, A_2, , A_n$ & Each attribute A_i domain D. D is called the domain of A_i and is denoted by $dom(A_i)$.
- The **degree (or arity)** of a relation is the number of attributes n of its relation schema.
- STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)
- A relation of degree seven, which stores information about university students, would contain seven attributes describing each student as follows:

  Using the data type of each attribute, the definition is sometimes written as:

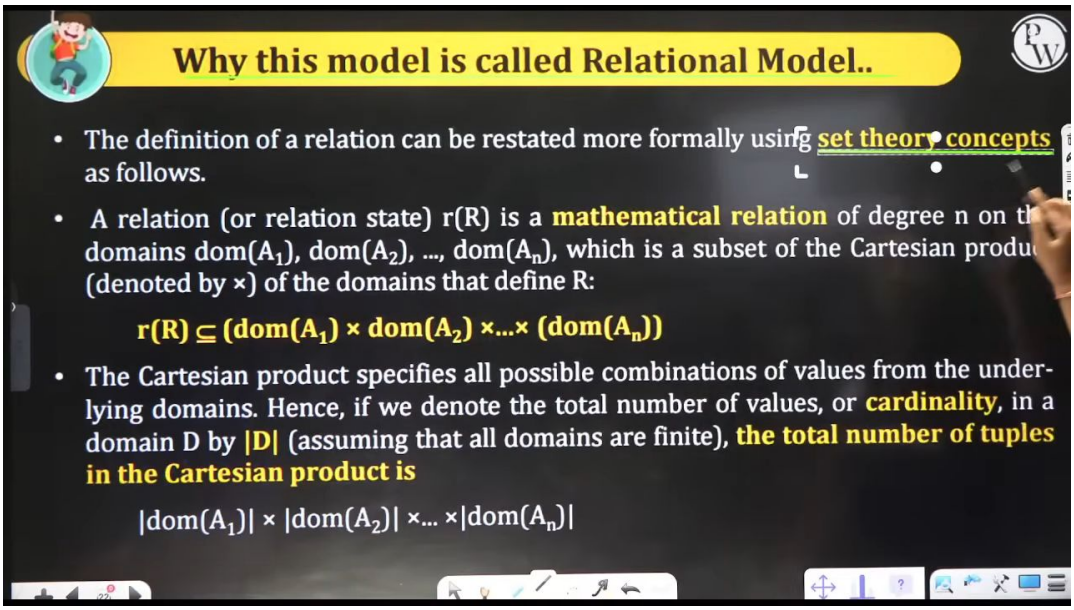
  Schema

  STUDENT(Name: string, Ssn: string, Home_phone: string, Address: string, Office_phone: string, Age: integer, Gpa: real)

43:29


Relation intension and Relation extension
A relation (or relation state) r of the relation schema R(A1, A2, , An), also denoted by r(R), is a set of n-tuples r = {t1, t2, ... tm}.
The terms relation intension for the schema R and relation extension for a relation state r(R) are also commonly used.
Relation State (relation extension)
Relation intension : Schema.
Relational Instance.

54:25


Why this model is called Relational Model..
The definition of a relation can be restated more formally using set theory concepts as follows.
A relation (or relation state) r(R) is a mathematical relation of degree n on th domains dom(A1), dom(A2), ..., dom(An), which is a subset of the Cartesian produ (denoted by ×) of the domains that define R:
r(R) ⊆ (dom(A1) × dom(A2) ×...× (dom(An))
The Cartesian product specifies all possible combinations of values from the under-lying domains. Hence, if we denote the total number of values, or cardinality, in a domain D by |D| (assuming that all domains are finite), the total number of tuples in the Cartesian product is
|dom(A1)| × |dom(A2)| ×... ×|dom(An)|

1:01:31

Why this model is called Relational Model..

- This product of cardinalities of all domains represents the total number of possible instances or tuples that can ever exist in any relation state r(R).
- Out Of all these possible combinations, a relation state at a given **time-the current relation state**-reflects only the valid tuples that represent a particular state of the real world.
- In general, as the state of the real world changes, so does the relation state, by being transformed into another relation state.

1:04:11

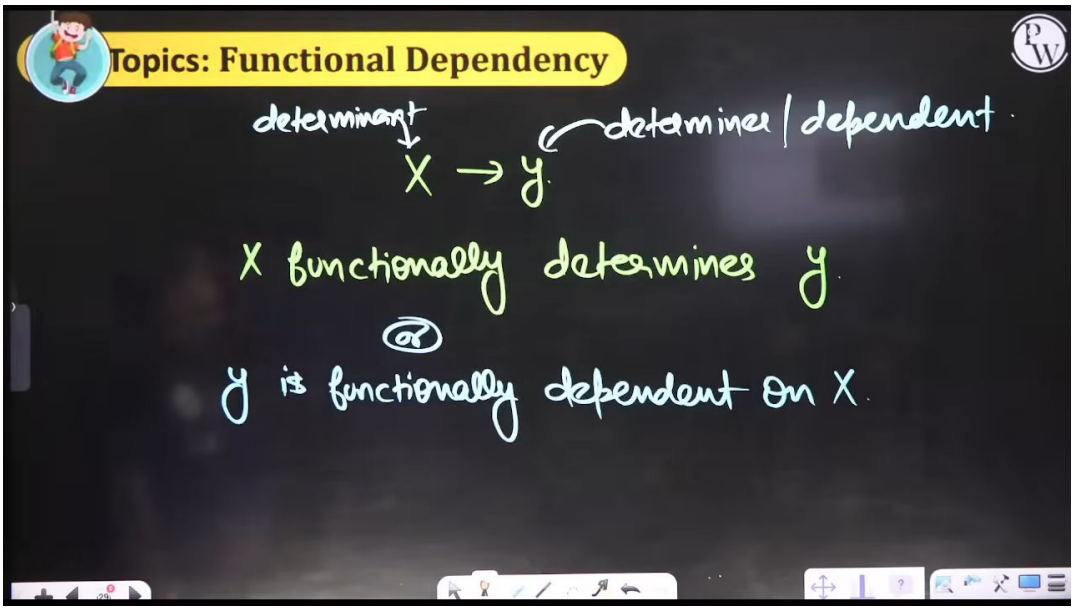
Topics: Functional Dependency $[X \rightarrow Y]$

Functional Dependency $[X \rightarrow Y]$

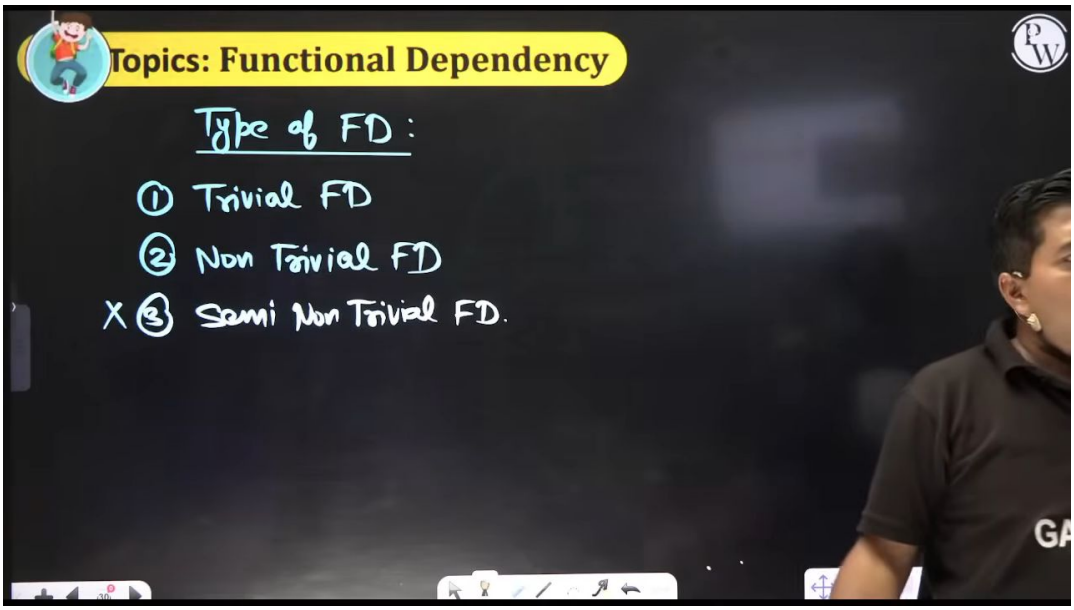
Let R be the Relational Schema, X & Y be the attribute Set of Relation R. & t_1 & t_2 Any Two Tuples such that $X \rightarrow Y$ iff.

$\boxed{\text{If } t_1.X = t_2.X \text{ then } t_1.Y = t_2.Y \text{ Must be Same.}}$

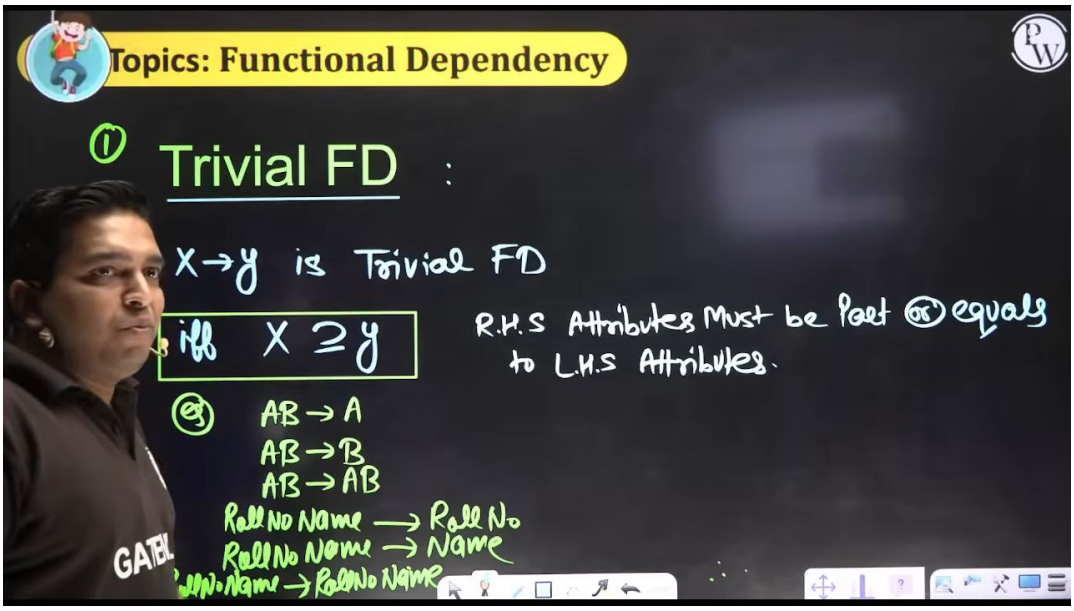
1:13:28


Topics: Functional Dependency
determinant
determined / dependent
X → y
X functionally determines y
or
y is functionally dependent on X

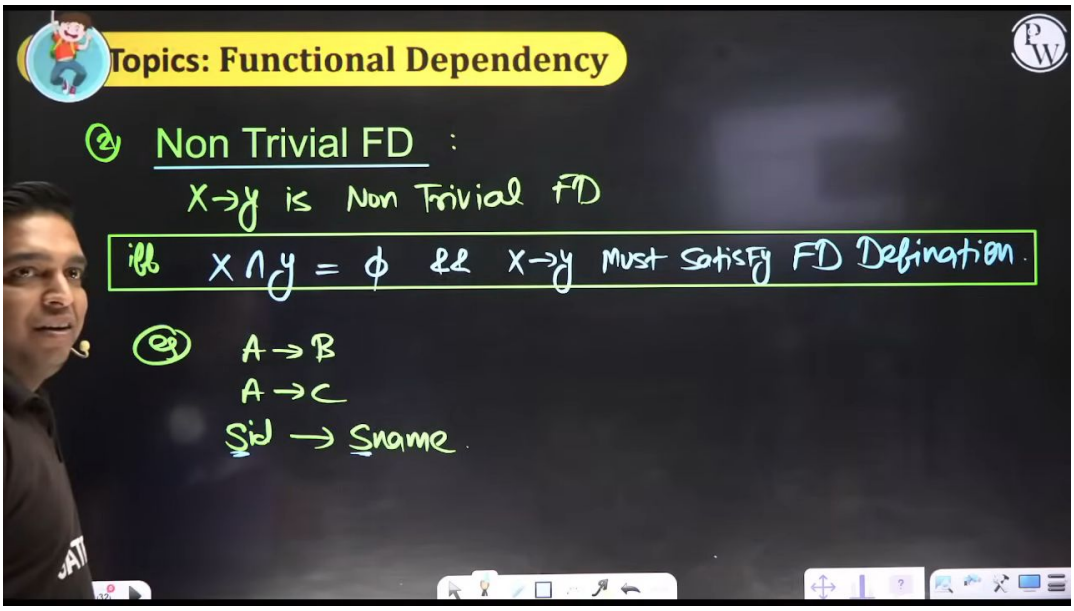
1:21:58


Topics: Functional Dependency
Type of FD :
① Trivial FD
② Non Trivial FD
X ③ Semi Non Trivial FD.

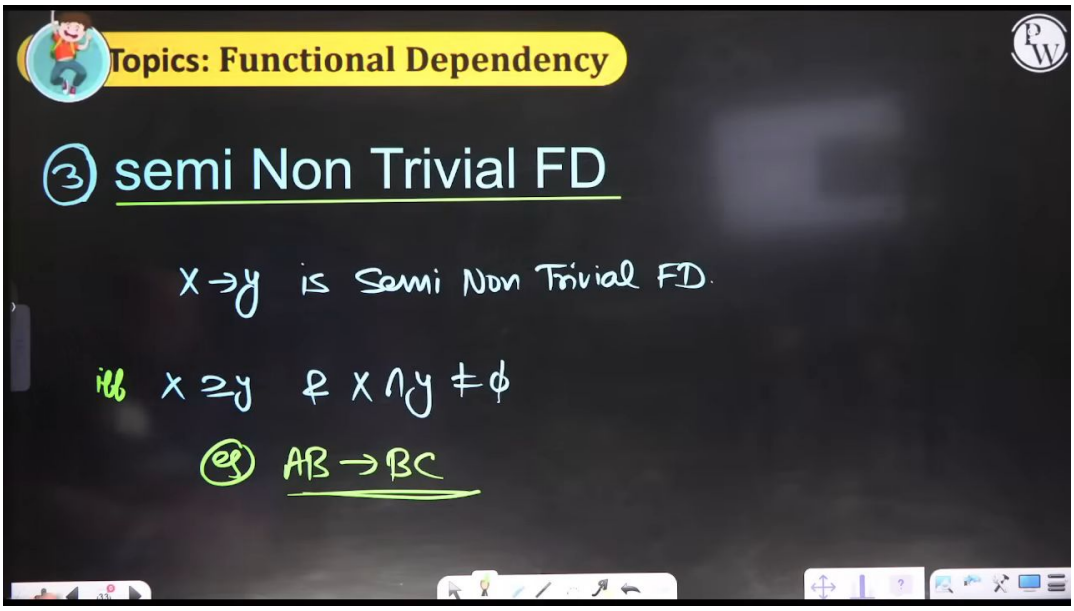
1:23:55


Topics: Functional Dependency
① Trivial FD :
X→y is Trivial FD
iff X ⊇ y
R.H.S Attributes Must be Part (or) equals to L.H.S Attributes.
eg AB → A
AB → B
AB → AB
Roll No Name → Roll No
Roll No Name → Name
Roll No Name → Roll No Name

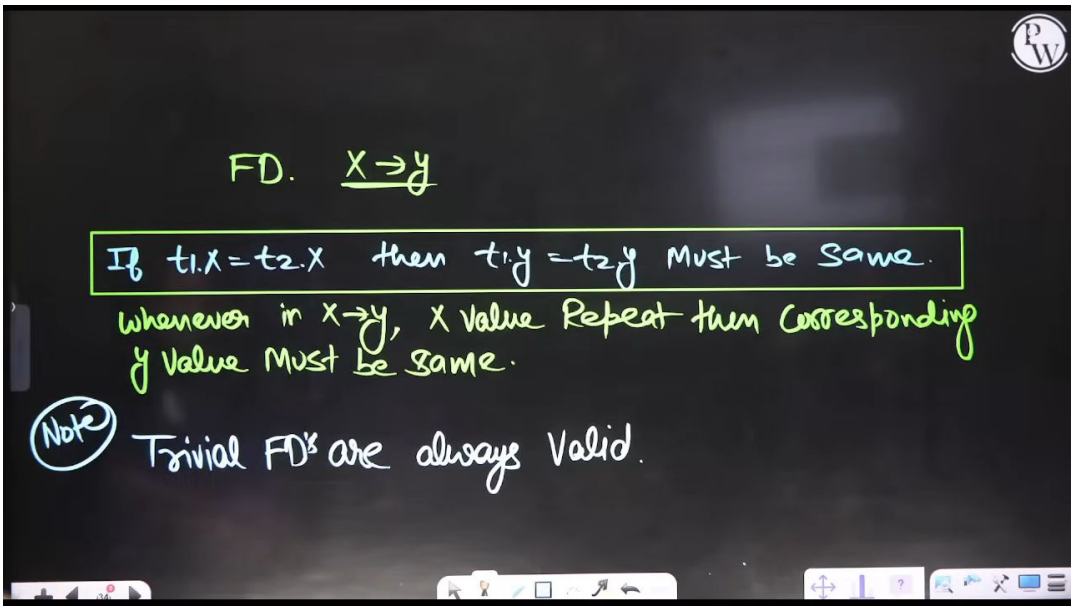
1:26:35


Topics: Functional Dependency
② Non Trivial FD :
X→y is Non Trivial FD
iff X ∩ y = ϕ && X→y Must Satisfy FD Defination.
eg A → B
A → C
Sid → Sname.

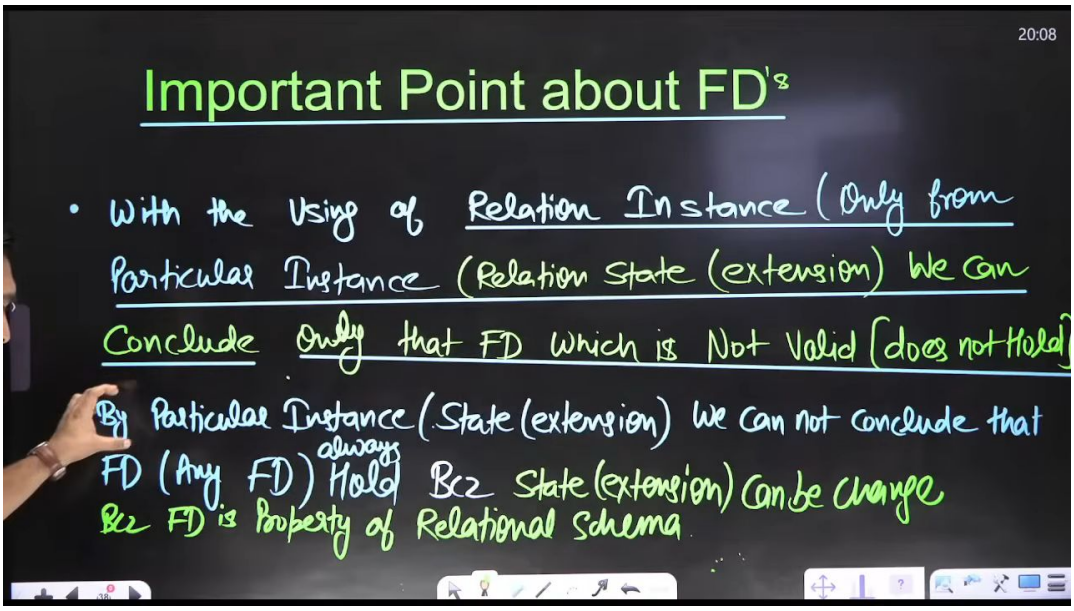
1:30:07


Topics: Functional Dependency
3 semi Non Trivial FD
X→Y is Semi Non Trivial FD.
iff X ⊇ Y & X ∩ Y ≠ ϕ
eg AB → BC

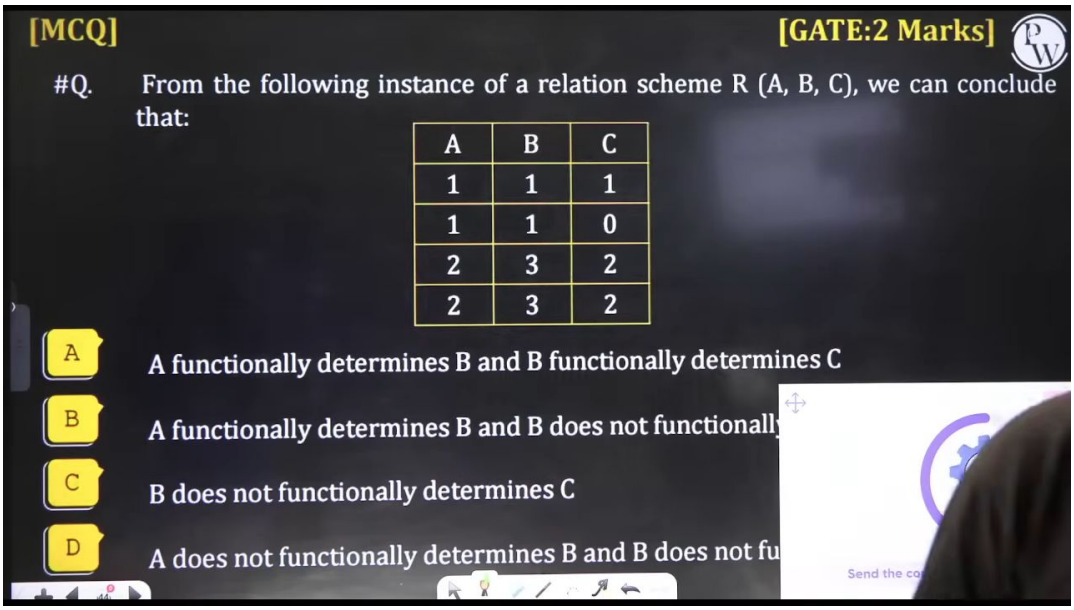
1:33:33


FD. X→Y
If t1.X = t2.X then t1.Y = t2.Y Must be Same.
whenever in X→Y, X Value Repeat then Corresponding Y Value Must be Same.
Note Trivial FD's are always Valid.

1:36:04


20:08
Important Point about FD's
• With the Using of Relation Instance (Only from Particular Instance (Relation State (extension) We can Conclude Only that FD which is Not Valid (does not Hold)
By Particular Instance (State (extension) We can not conclude that FD (Any FD) always Hold Bcz State (extension) Can be change
Bcz FD is Property of Relational Schema

2:08:25


[MCQ]
[GATE:2 Marks]
#Q. From the following instance of a relation scheme R (A, B, C), we can conclude that:
A B C
1 1 1
1 1 0
2 3 2
2 3 2
A A functionally determines B and B functionally determines C
B A functionally determines B and B does not functionally
C B does not functionally determines C
D A does not functionally determines B and B does not fu

2:12:05

[MCQ] [GATE:2 Marks]

#Q. Given the following relation instance.

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance?

A. $XY \rightarrow Z$ and $Z \rightarrow Y$

B. $YZ \rightarrow X$ and $Y \rightarrow Z$

C. $YZ \rightarrow X$ and $X \rightarrow Z$

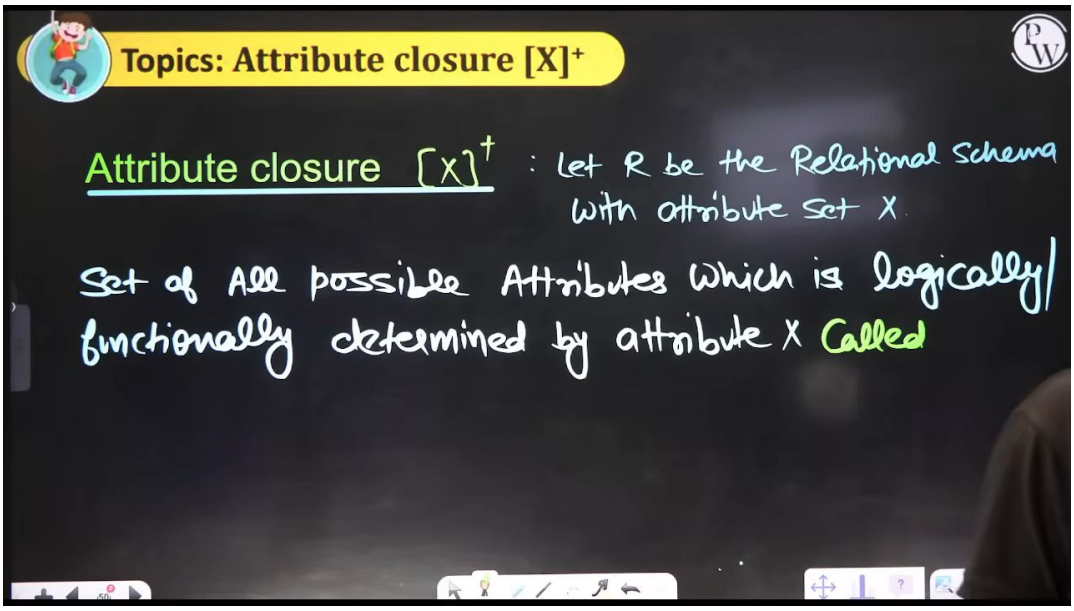
D. $XZ \rightarrow Y$ and $Y \rightarrow X$

2:16:12

Armstrong's Axioms/Inference Rules

- Axioms, or rules of inference, provide a simpler technique for reasoning about functional dependencies
- In the rules that follow, we use Greek letters ($\alpha, \beta, \gamma, \ldots$) for sets of attributes.
- We can use the following three rules to find logically implied functional dependencies.
- By applying these rules repeatedly, we can find all of F+, given F. This collection of rules called Armstrong's Axioms in honor of the person who first proposed it.
  - **Reflexivity Rule**: If α is a set of attributes and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ holds.
  - **Augmentation rule**: If $\alpha \rightarrow \beta$ holds and γ is a set of attributes, then $\gamma\alpha \rightarrow \gamma\beta$ holds.
  - **Transitivity Rule**: If $\alpha \rightarrow \beta$ holds and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$ holds.

2:20:37


Topics: Attribute closure [X]+
Attribute closure [X]+ : Let R be the Relational Schema with attribute set X.
Set of All possible Attributes which is logically/functionally determined by attribute X Called

2:28:21