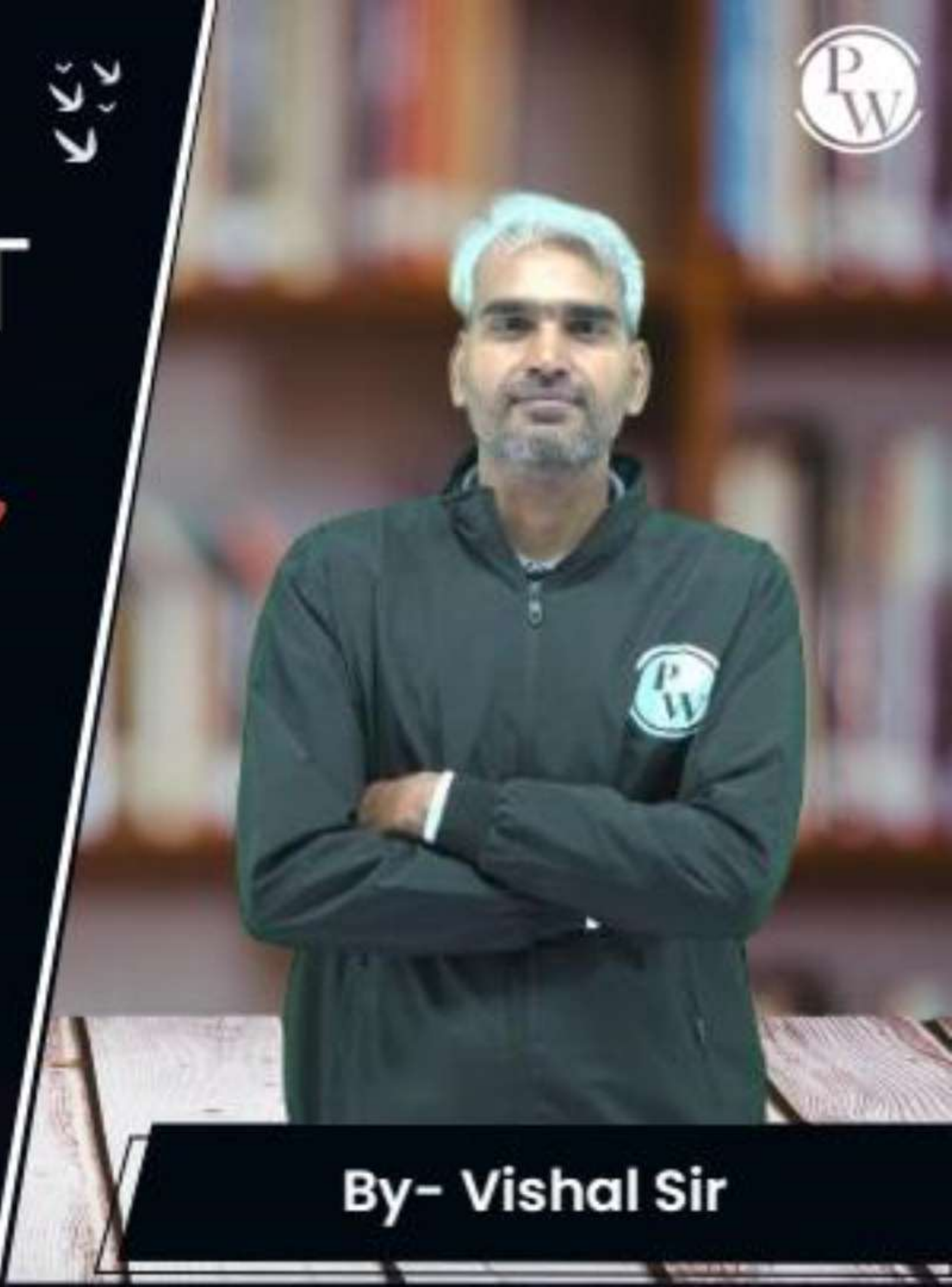
Computer Science & IT

Database Management System

Relational Model & Normal Forms

Lecture No. 04



By- Vishal Sir

Recap of Previous Lecture

Topic: Functional dependency

if $X \rightarrow Y$
then, if $t_1.X = t_2.X$
then $t_1.Y = t_2.Y$

Topic: Types of functional dependency

→ Trivial: if $X \supseteq Y$ then $X \rightarrow Y$ is trivial

→ Non-trivial: if $X \rightarrow Y$ exists,
and
$X \cap Y = \phi$, then non-trivial

→ Semi-non-trivial: if $X \rightarrow Y$ exist
and
$X \not\supseteq Y$ and $X \cap Y \neq \phi$, then semi

Slide

Topics to be Covered

- **Topic**: Properties of functional dependency
- **Topic**: Different types of keys in RDBMS
- **Topic**: Candidate key
- **Topic**: Super key

Slide

H.W.

#Q. Write all possible non-trivial FDs with respect to relational schema

R(A, B, C, D)

A → B	C → A	A → BC	C → AB	A → BCD
A → C	C → B	A → BD	C → AD	B → ACD
A → D	C → D	A → CD	C → BD	C → ABD
B → A	D → A	B → AC	D → AB	D → ABC
B → C	D → B	B → AD	D → AC	
B → D	D → C	B → CD	D → BC	

28 non-trivial dependencies with only one attribute in L.H.S.

AB → C	BD → A	AB → CD
AB → D	BD → C	AC → BD
AC → B	AD → B	BC → AD
AC → D	AD → C	BD → AC
BC → A	CD → A	AD → BC
BC → D	CD → B	CD → AB

18 FDs with two attributes in L.H.S.

ABC → D
ABD → C
BCD → A
ACD → B

4 FDs with 3 attributes in LHS.

Q: Let R(A,B,C,D) be a relation with 4 attributes, how many non-trivial FDs are possible with this 4 attributes

- Total no. of attributes = 4

Case ① when exactly one attribute in L.H.S. w.r.t. FD $X \to Y$

$X \longrightarrow Y$

$^4C_1$ and $(^3C_1$ or $^3C_2$ or $^3C_3)$

$4 * \{3+3+1\} = 4*7 = 28$

Case ② when exactly two attribute in L.H.S. w.r.t. FD $X \to Y$

$X \longrightarrow Y$

$^4C_2$ and $\{^2C_1$ or $^2C_2\}$

$6 * \{2+1\} = 6*3 = 18$

Case ③ when exactly three attribute in L.H.S. w.r.t. FD $X \to Y$

$X \longrightarrow Y$

$^4C_3$ and $\{^1C_1\} = 4*1 = 4$

Total No. of non-trivial FDs

$= 28+18+4$

$= 50$

Topic : Properties of Functional Dependencies

① Reflexivity :- If $X \supseteq Y$ then FD $X \rightarrow Y$ is called reflexive FD. and each reflexive FD always holds in the relation

② Augmentation :- Let $X \rightarrow Y$ exists in relation R, and Z is some set of attributes from relation R, then $XZ \rightarrow YZ$ will hold true in the relation.

③ Transitivity : Let $X \rightarrow Y$ and $Y \rightarrow Z$ exist in relation R. then $X \rightarrow Z$ will also hold true in the relation R.

This three Properties are called Armstrong's axioms

Topic : Properties of Functional Dependencies

(4) Decomposition (Splitting Rule) : If $X \rightarrow \underline{YZ}$ exist in relation R then $X \rightarrow Y$ and $X \rightarrow Z$ holds true in relation R.

Let $\underbrace{AB \rightarrow BC}$ exist in R
it is semi-non-trivial

$AB \rightarrow BC \Rightarrow$ Decomposition (Splitting) $\Rightarrow$ $AB \rightarrow B$ & $AB \rightarrow C$

trivial

non-trivial

We can always decompose semi-non-trivial FD into trivial & Non-trivial part

Note: If $\underline{XY} \rightarrow \underline{Z}$ exist in R.
then $X \rightarrow Z$ & $Y \rightarrow Z$ need not hold true

We are not allowed to split in LHS.

Topic : Properties of Functional Dependencies

(5) Union: If $X \rightarrow Y$ and $X \rightarrow Z$ exist in relation R.
then $X \rightarrow YZ$ will also hold true in relation R.

(6) Composition :- If $X \rightarrow Y$ and $P \rightarrow Q$ exist in relation R
then $XP \rightarrow YQ$ will also hold true in relation R

(7) Pseudo Transitivity :- if $X \rightarrow Y$ and $YW \rightarrow Z$ exist in relation R.
then $XW \rightarrow Z$

Topic : Different types of keys

- No two tuples of the relation can be completely same
{i.e., duplicate tuples are not allowed in a relation}

∴ Key is a very important Concept w.r.t. relation. Each tuple of the relation can be uniquely identified by the values of attributes of the key.

- There are various types of keys
  - ① Candidate key
  - ② Primary key
  - ③ Alternate key
  - ④ Super key
  - ⑤ Foreign key

Key:- The set of attributes that can uniquely identify each tuple of the relation is called a Key.

{**Note:** In general key need not be minimal}

Student

Sid	Sname	Branch
S_1	A	CS
S_2	A	IT
S_3	B	CS
S_4	C	EC
S_5	B	CS

If values of Sid are guaranteed to unique in the relation Student, then Sid will form one key of the relation

If it is guaranteed that values of attributes of a set will always be unique in the relation then that set of attributes will always form one of the key of that relation eg: Consider the following relation

Student(Sid, Sname, fee)

Student

Sid	Sname	fee
S_1	A	500
S_2	A	400
S_3	B	700
S_4	C	500
S_5	C	500
S_6	D	600

Let FDs that exist in the relation are

$$\boxed{\begin{aligned} Sid &\rightarrow Sname \\ Sid &\rightarrow fee \end{aligned}}$$

- Values of Sid will always be unique in the student table. ∴ Sid is a Key
- The values of (Sid, Sname) together will also be uniques in all the tuples.
  ∴ (Sid, Sname) is also a Key

eg: Consider the following relation

Enroll (Sid, Cid, I-id)

Instructor ID

Enroll

Sid	Cid	I-id
S_1	C_1	101
S_2	C_1	101
S_3	C_2	104
S_3	C_3	101
S_3	C_1	101

let following FD holds

$$\boxed{Cid \rightarrow I\text{-}id}$$

Together the values of (Sid, Cid) will always be unique in all the tuples.

∴ (Sid, Cid) is a Key.

2 mins Summary

Topic Properties of functional dependency

Topic Different types of keys in RDBMS

Topic Candidate key

Topic Super key

Slide

THANK - YOU

Slide