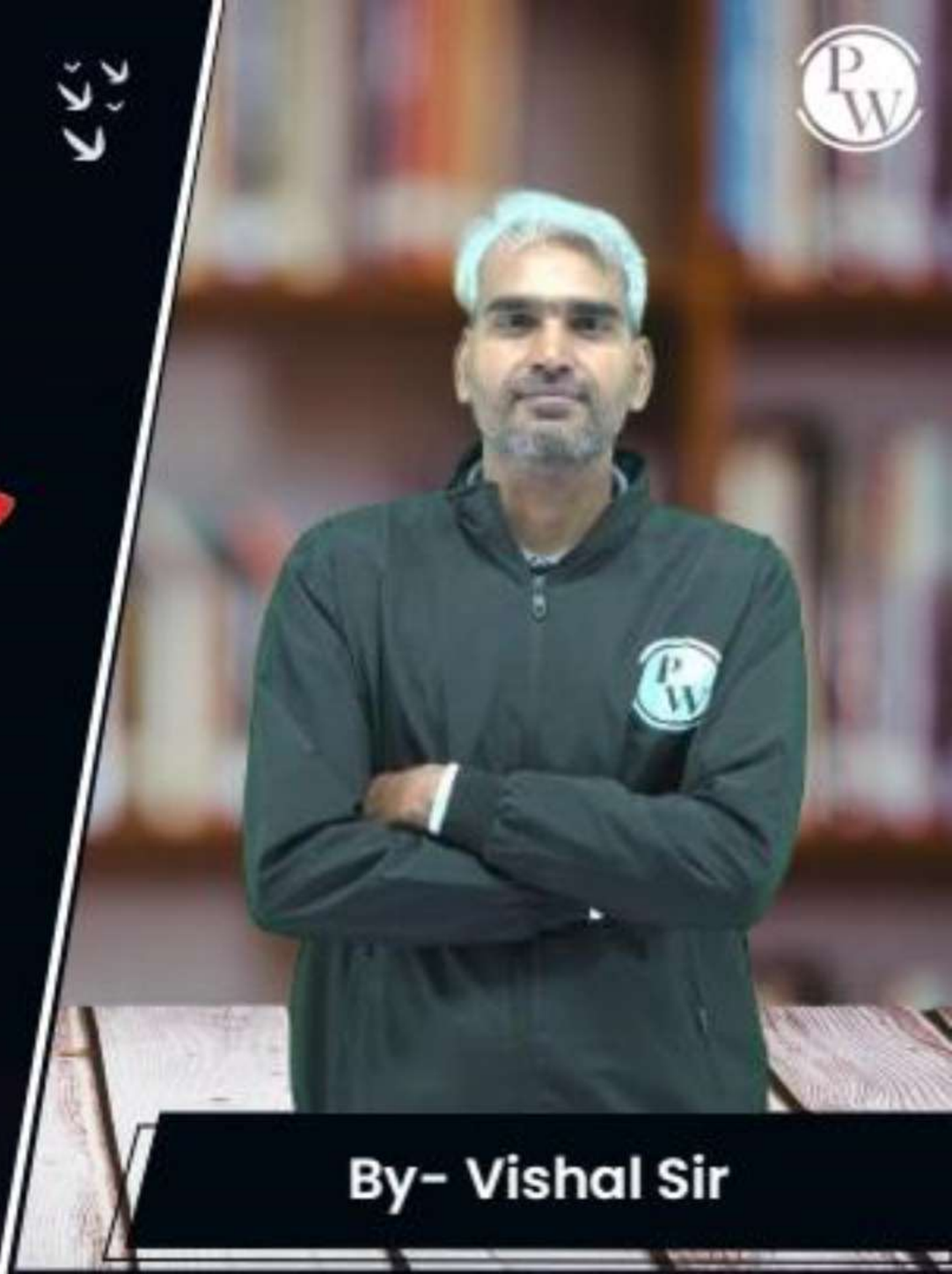
DS & AI

Database Management System

Super 1500+

Lecture No. 10



By- Vishal Sir

Recap of Previous Lecture

Topic File organization and Indexing

Slide

Topics to be Covered

- Topic: TRC { Tuple Relational Calculus }
- Topic: ER model and Integrity constraints

Slide

#Q.64 Consider the following relational schema

- actor (insta_id, name, language, age) Note: unique name of each actor.
- movie (movie_id, title, year, director_id) Note: title is unique for each movie.
- acts_in (insta_id, movie_id, character_name)
- director (director_id, name, language) Note: unique name of each director.

- Retrieve details of all movies that were released in 2010. The output schema should be the same as that of the movie table.

$$\{t \mid \underbrace{t \in movie \wedge t.year = 2010}_{\equiv\ movie(t)}\}$$

#Q.65 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve details of all actors that are not in their thirties(i.e., age<30 or age>39). The output schema should be the same as that of the actor table.

$\{t \mid t \in actor \wedge (t.age < 30 \vee t.age > 39)\}$

$t \in actor$ ≡ actor(t)

#Q.66 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve the names of all directors.

$\{ t.name \mid t \in director \}$

director(t)

t.name = t[name]

$t \in Relation \equiv Relation(t)$

$t.A_1 = t[A_1]$

any attribute

#Q.67 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve the names of all "Telugu" language directors.

$\{ t.name \mid t \in director \wedge t.language = 'Telugu' \}$

#Q.68 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve the name of each actor together with the titles of the movie he/she has performed in.

$$\{ t \mid \exists a \in actor \wedge \exists m \in movie \wedge \exists s \in acts\_in \wedge a.insta\_id = s.insta\_id \wedge m.movie\_id = s.movie\_id \wedge t.name = a.name \wedge t.title = m.title \}$$

#Q.69 Consider the following relational schema

- actor (insta_id, name, language, age) Note: unique name of each actor.
  movie (movie_id, title, year, director_id) Note: title is unique for each movie.
- acts_in (insta_id, movie_id, character_name)
  director (director_id, name, language) Note: unique name of each director.

Retrieve the names of all actors that have played the character of "Ravan".

- $\{ t.name \mid t \in actor \wedge \exists S \in acts\text{-}in (t.insta\text{-}id = S.insta\text{-}id \wedge S.character\_name = 'Ravan') \}$
- $\{ t \mid \exists a \in actor \wedge \exists S \in acts\text{-}in \wedge a.insta\text{-}id = S.insta\text{-}id \wedge S.character\_name = 'Ravan' \wedge t.name = a.name \}$

#Q.70 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve the names of all actors that have played the character of "Ravan", together with the year the corresponding movies were released.

$$\{ t \mid \exists a \in actor \wedge \exists m \in movie \wedge \exists s \in acts\_in \wedge a.insta\_id = s.insta\_id \wedge m.movie\_id = s.movie\_id \wedge s.character\_name = \text{'Ravan'} \wedge t.name = a.name \wedge t.year = m.year \}$$

#Q.71 Consider the following relational schema

- actor (insta_id, name, language, age) Note: unique name of each actor.
  movie (movie_id, title, year, director_id) Note: title is unique for each movie.
  acts_in (insta_id, movie_id, character_name)
  director (director_id, name, language) Note: unique name of each director.

  Retrieve all actors that acted in movie with title "Bahubali". The output schema should be the same as that of the actor table.

$$\{ t \mid t \in actor \wedge \exists m \in movie \wedge \exists s \in act\_in \wedge t.insta\_id = s.insta\_id \wedge s.movie\_id = m.movie\_id \wedge m.title = 'Bahubali' \}$$

#Q.72 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

✓ Find out the names of all actors that have performed in a movie directed by "Anurag Kashyap".

$\{t \mid \exists a \in actor \wedge \exists m \in movie \wedge \exists S \in acts\_in \wedge \exists D \in director \wedge a.insta\_id = S.insta\_id \wedge S.movie\_id = m.movie\_id \wedge m.director\_id = D.director\_id \wedge D.name = \text{'Anurag Kashyap'} \wedge t.name = a.name\}$

#Q.73 Consider the following relational schema

actor (insta_id, name, language, age) Note: unique name of each actor.

movie (movie_id, title, year, director_id) Note: title is unique for each movie.

acts_in (insta_id, movie_id, character_name)

director (director_id, name, language) Note: unique name of each director.

Retrieve the titles of all movies in which Amitabh and Jaya have co-acted.

$$\{t.title \mid t \in movie \wedge \exists a_1 \in actor \wedge \exists s_1 \in acts\_in \wedge a_1.insta\_id = s_1.insta\_id \wedge a_1.name = 'Amitabh' \wedge \exists a_2 \in actor \wedge \exists s_2 \in acts\_in \wedge a_2.insta\_id = s_2.insta\_id \wedge a_2.name = Jaya \wedge s_1.movie\_id = s_2.movie\_id \wedge t.movie\_id = s_1.movie\_id\}$$

Unsafe TRC query :-

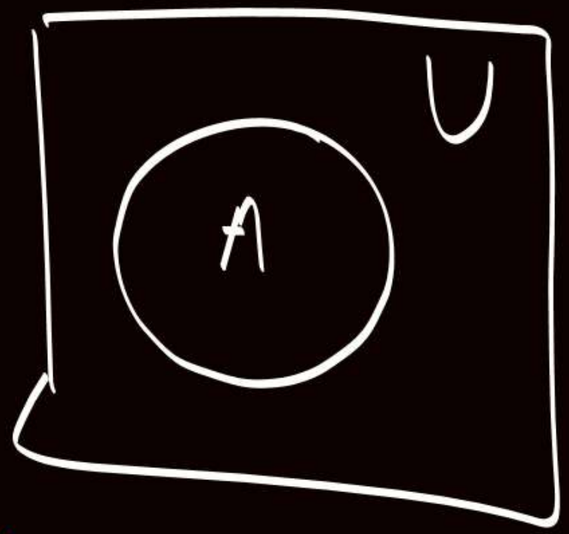
$\{ t \mid t \notin \text{Relation} \}$

$\Downarrow\Downarrow$

$\{ t \mid \sim (t \in \text{Relation}) \}$

Unsafe TRC query may produce infinite tuples in O/p if domain of attribute is infinite

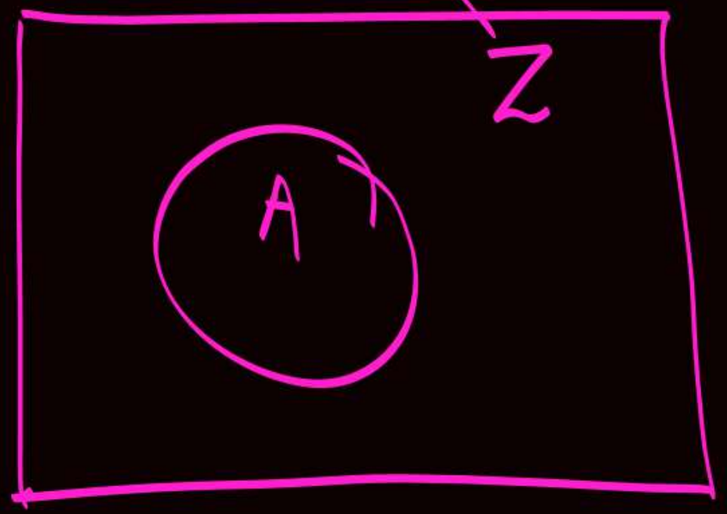
$U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$



$A = \{3, 5, 7, 9, 10\}$

$A^c = \{1, 2, 4, 6, 8\}$

Set of all integers



$A = \{3, 5, 7, 9, 10\}$

A^c = it will be an infinite set

* We can never write Relational algebra query that can produce the tuples which are not present in the given relation
  * i.e. we can never write an equivalent R.A query corresponding to an unsafe TRC query.
  * For Every Safe TRC query We can write an equivalent R.A query

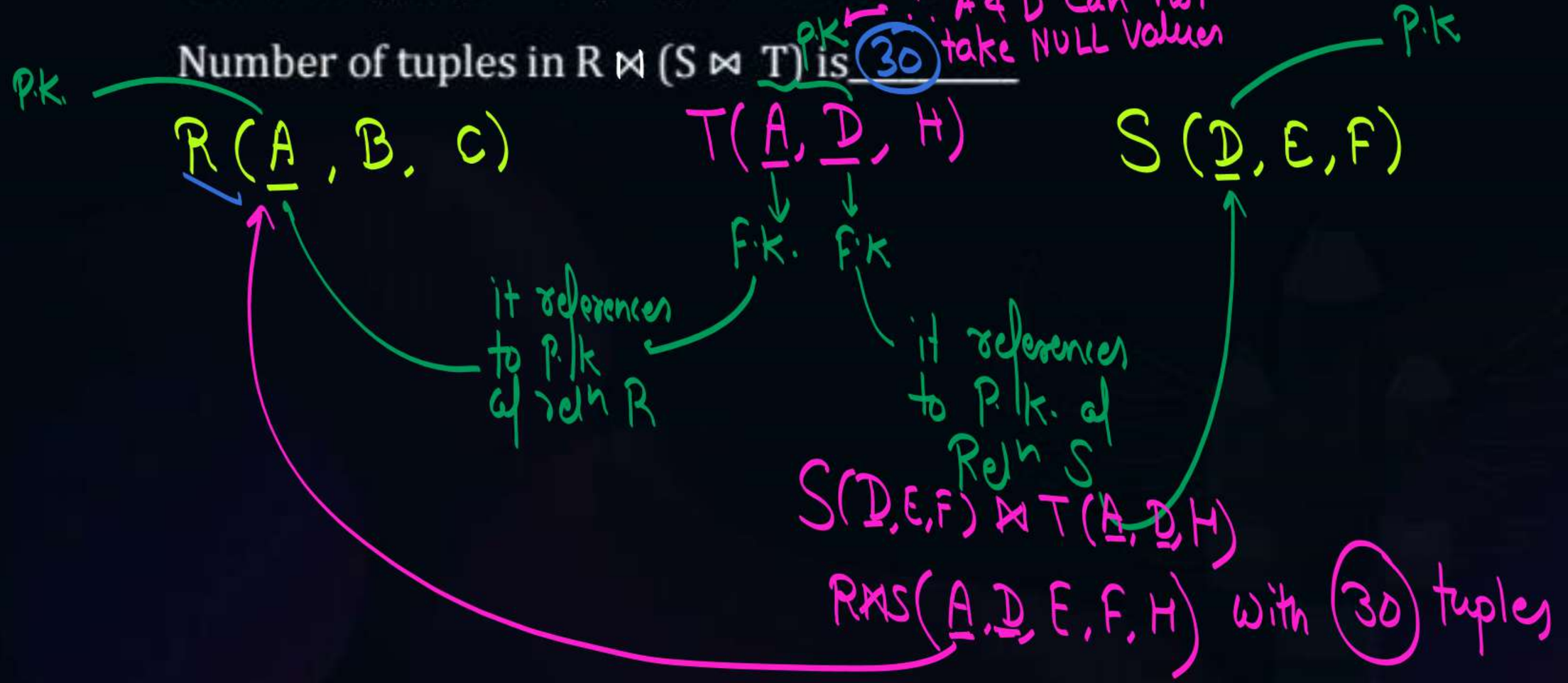
* Power of TRC query $\not\equiv$ Power of Relation algebra
  {safe + unsafe}

  Because of unsafe TRC

* Power of safe TRC query $\equiv$ Power of Relational Algebra

#Q.75 Let R(A,B,C) is a relation with 20 tuples, S(D,E,F) is a relation with 40 tuples and T(A,D,H) is a relation with 30 tuples, D in relation T is foreign key referencing to primary key of relation S.

Number of tuples in R ⋈ (S ⋈ T) is 30



$R \bowtie S$ => Then no. of tuples in $R \bowtie S$ will be same as no. of tuples in the relation with foreign key of i.e; same as no of tuples in S }

#Q.76 Which of the following is/are true for an ER model?

(a) Weak entity must have total participation in identifying relation.

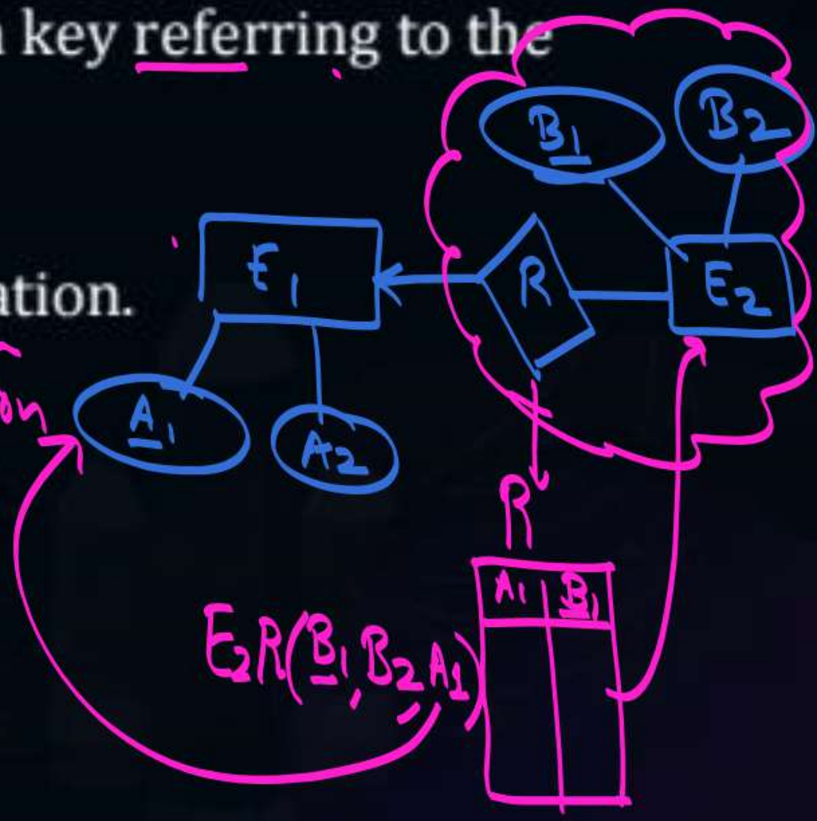
(b) Entity corresponding to 1 side will include foreign key referring to the primary key of many side entity.

(c) descriptive attributes are associated with entity.

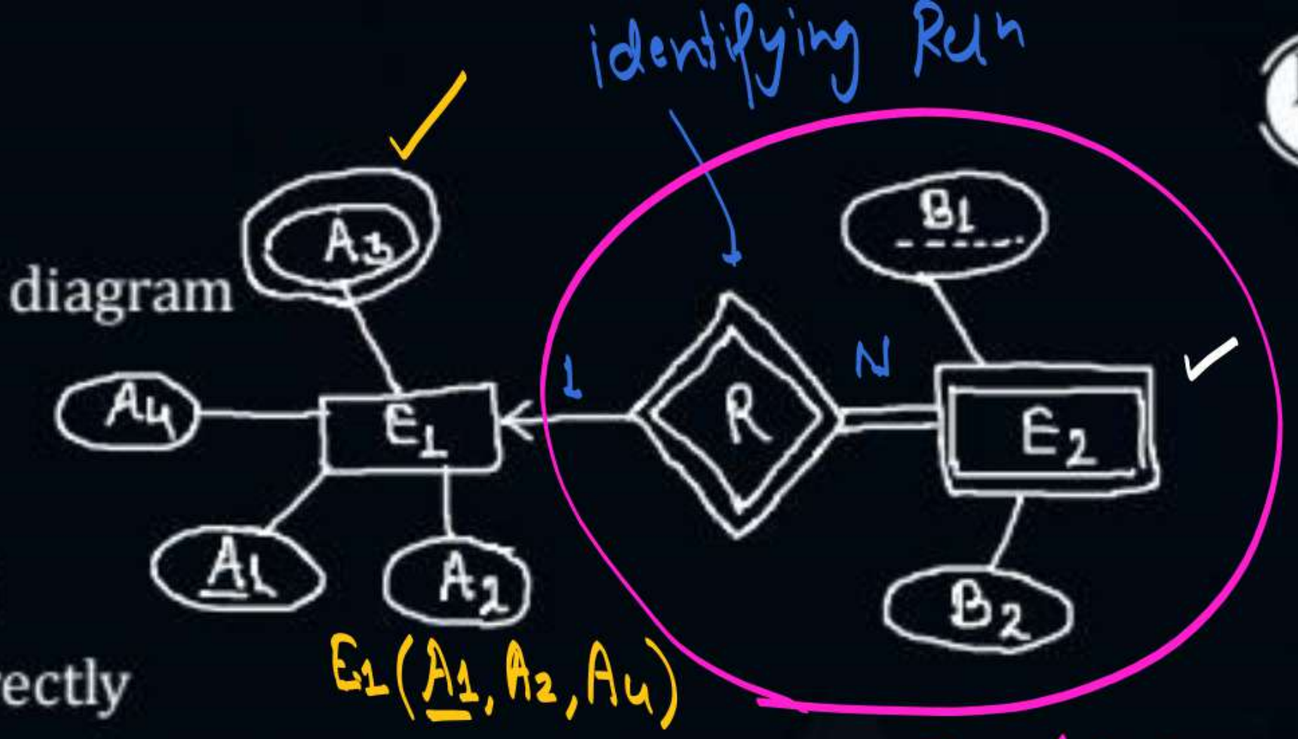
(d) minimum cardinality of '1' specifies total participation.

of '0' specifies partial participation

True Statement will be,
Entity Corresponding to many side will include foreign key referring to the primary key of '1' side entity



#Q.77 Consider the following ER diagram



$E_1(\underline{A_1}, A_2, A_4)$

$E_1'(\underline{A_1}, \underline{A_3})$

$E_2R(\underline{A_1}, \underline{B_1}, B_2)$

P.K.

- If given ER diagram is mapped to a relational model, then correctly depict the scenario the minimum number of tables required is

A 2

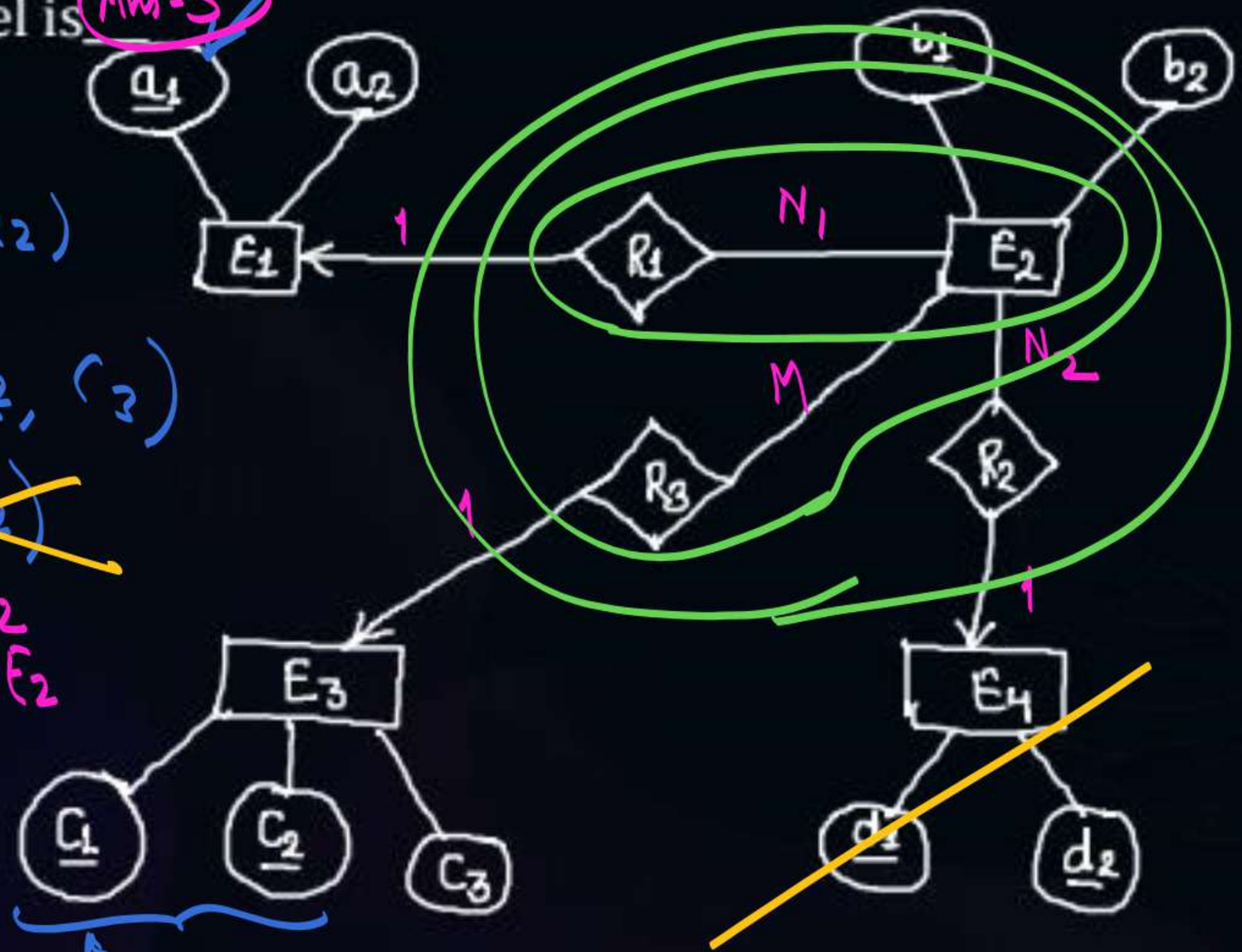
B 3

C 4

D 5

#Q.78 Minimum number of tables required to convert the ER diagram into relational model is

Ans=3



$E_1(\underline{a_1}, a_2)$

$E_3(\underline{c_1}, \underline{c_2}, c_3)$

~~$E_4(\underline{d_1}, \underline{d_2})$~~

Both d_1 & d_2 are part of E_2

$E_2(\underline{b_1}, b_2, a_1, d_1, d_2, c_1, c_2)$

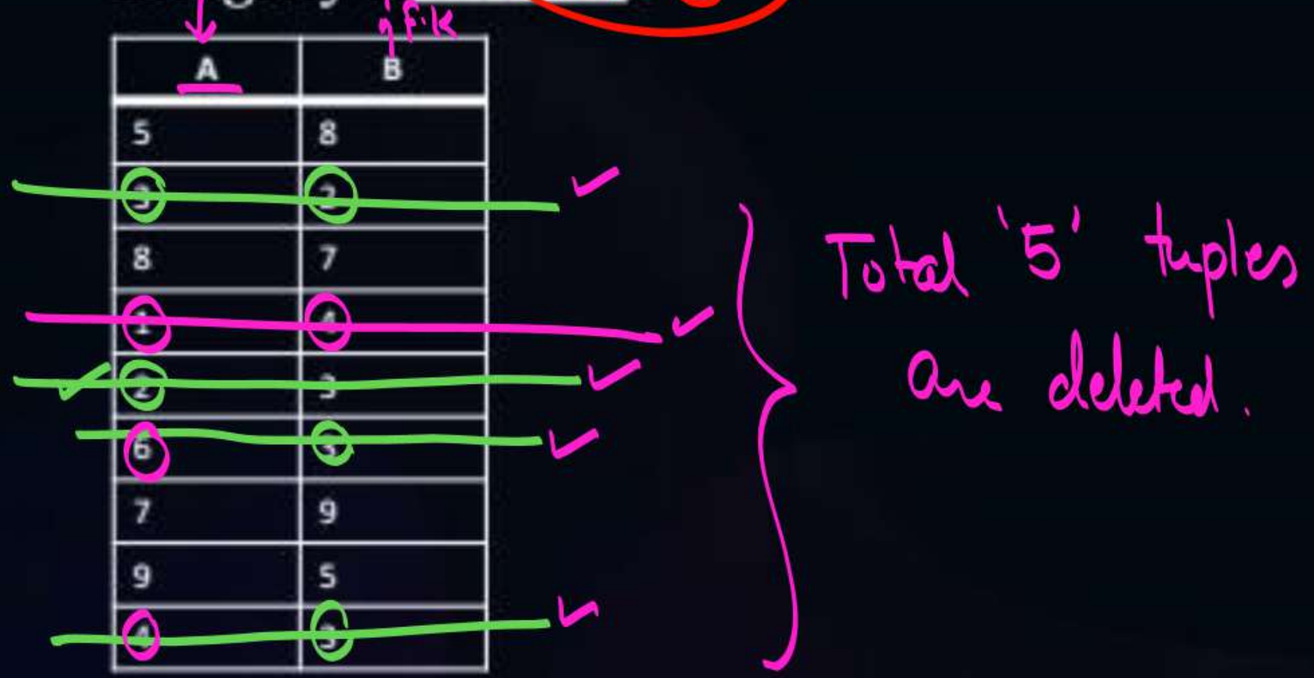
P.K.

F.K.

F.K

#Q.79 Consider a relational table R(A, B) as given below. A is the primary key of relation R and B is the foreign key referring to primary key A of relation R with on delete cascade. If we delete tuple (2, 3) from relation R, then total number of tuples (including (2, 3)) deleted from R to preserve referential integrity is _____

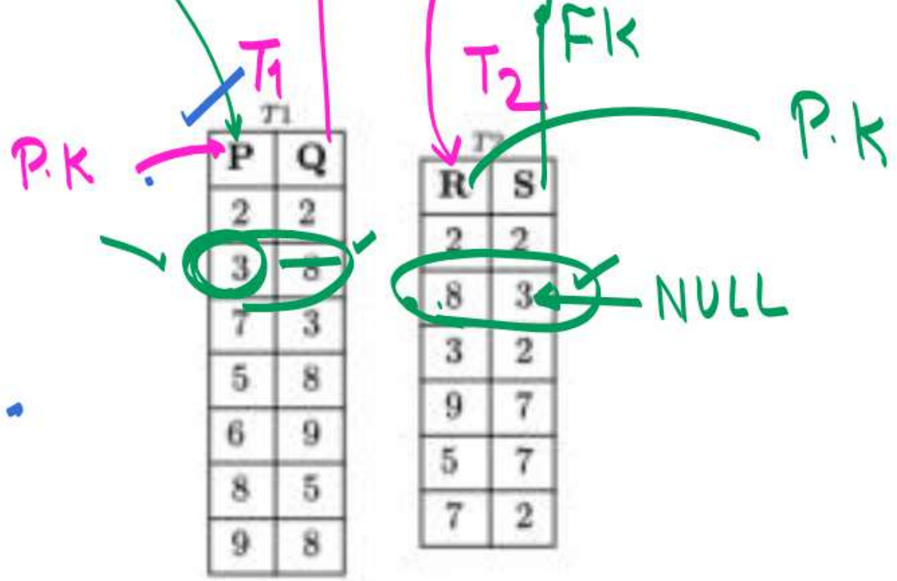
Ans = 5



#Q.80

GATE -PYQ

Consider the following tables $T1$ and $T2$.



T1

P	Q
2	2
3	8
7	3
5	8
6	9
8	5
9	8

T2

R	S
2	2
8	3
3	2
9	7
5	7
7	2

In table $T1$ **P** is the primary key and **Q** is the foreign key referencing **R** in table $T2$ with on-delete cascade and on-update cascade. In table $T2$, **R** is the primary key and **S** is the foreign key referencing **P** in table $T1$ with on-delete set NULL and on-update cascade. In order to delete record $(3, 8)$ from the table $T1$, the number of additional records that need to be deleted from table $T1$ is ____"0"____

Q. ✓ Movie (Id, CustomerRating)

✓ Genre (ID, Name)

✓ Movie-Genre (MovieID, GenreID)

```
Select *
From Movie, Genre, Movie-Genre
Where • Movie.CustomerRating > 3.4 AND
        Genre.Name = "Comedy" AND
        Movie_Genre.MovieID = Movie.ID AND
        Movie_Genre.GenreID = Genre.ID
```

(a) B^+ tree on all attributes

(b) Hash index on Genre.Name and B^+ tree index on remaining attributes

(c) Hash index on Movie.CustomerRating and B^+ tree on remaining

(d) Hash index on all the attributes.

2 mins Summary

Topic TRC

Topic ER model and Integrity constraints

Slide

THANK - YOU