# 1. ER-MODEL



## 1. INTRODUCTION TO DBMS

Data: Any fact that could be Recorded and stored (Text, Numbers, Image)

Database: collection of Related data.

⇒ The database that contains text and Numbers is called Traditional Database.

⇒ Realtime Databases: Supermarkets, Firms.

⇒ "Dataware house" contains large amount of Data, the data is going to be historical.

⇒ Now a days the Databases are computerised and there must be some software that Defines, Constructs, Manipulate the Databases. Now, the software that performs above operations on the Database is called "Database Management Systems".

⇒ $\boxed{\text{DB + DBMS = DATA BASE SYSTEMS}}$

## 2. MODELS IN DBMS

⇒ The various models that are used when designing the database is

1. High Level or Conceptual Models ⇒ NAIVE USERS ⇒ Diagrams ⇓ ER-model.
2. Representational / Implementation Model ⇒ used by programers. (Tables).
3. Logical Level / physical Data models ⇒ Structure, Datatype.

## 3. INTRODUCTION TO ER-MODEL

ER Model = Entity - Relationship Model. (Entity, Attributes, Relationships)

ENTITY: Any object in our Database.

ATTRIBUTES: The things that describe the Entities are called Attributes. (properties that are used to describe Entities better).

RELATIONSHIPS: Association among entities.

⇒ Entity type = Schema = Heading = intension = PERSON(Age, Name, Add).

⇒ Entity = (26, Raju, --) = Extension. (2)

⇒ In the ER- Model we use Entity types but not the Entities.

## 4. ATTRIBUTES

⇒ Attributes are useful inorder to describe the entities Better.

The Attributes are manily classified into

1> Simple Attributes (Vs) Composite Attributes.

2> Single valued (vs) Multi valued Attributes.

3> Stored (vs) Derived Attributes

4> Complex Attributes.

PERSON

Name = SurName, FirstName, Middle name, LastName (Composite Attribute)

Age = Single valued Attribute

PNO = Multi valued Attribute

DOB = Stored Attribute

Derives.

Age = Derived Attribute

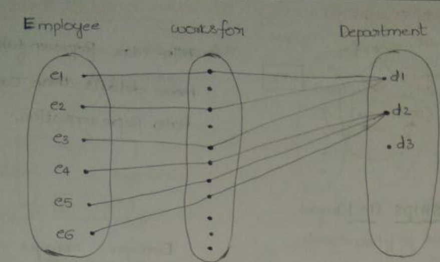
Address = Complex Attribute — Composite Attribute / Multi valued Attribute.

## 5. RELATIONSHIPS (1-M)

⇒ Relationship is nothing but Association among entities.

Requirement Analysis : 1. Every employee works for exactly a Dep and a Dep can have many Employees

2. New department need not have any Employee.




Degree : How many entities are participating in every Relationship.

deg = (2) {for above diagram} = Binary Relationship

Cardinality Ratio : what is the maximum no of Relationships that an entity can participate.

⇒ cardinality of employee = 1

⇒ cardinality of Department = Many = (M)/(N)

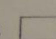
Participation or Existance : Min. no. of Relationships that an entity can participate

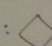
$\boxed{\text{participation = Min. cardinality}}$

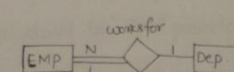
⇒ participation (employee) = 1

⇒ participation (Department) = 0. (Newly formed department cannot have employees).

cardinality Ratio
participation/Existance } Structural constraints.

Entities : 

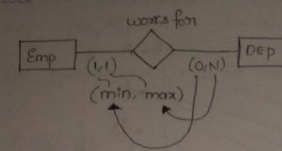
Relationships : 




↳ Total participation (All the entities of the set are participating in Relationship)

CARDINALITY RATIO REPRESENTATION.

MIN MAX REPRESENTATION

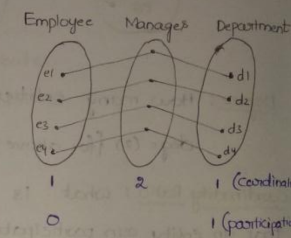


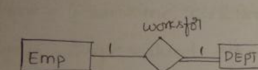

⇒ Min- Max Representation gives more details than Cardinality Ratio Representation.

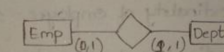
## 6. RELATIONSHIPS (1-1)

RA:

Every Department should have a manager and only one manager manages at 1 department and an employee can manage only one department. (Emp should work only in 1 department)

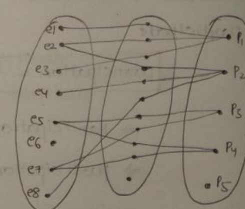



## 7. RELATIONSHIPS M-M EXAMPLES

RA: Every employee is supposed to work atleast on one project and he can work on many projects as well as every project is supposed to have many employees and is supposed to have atleast one employee.




P1: $e_1\ e_2\ e_3$

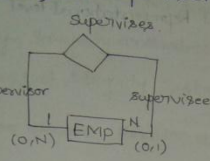
P2: $e_2\ e_4\ e_8$

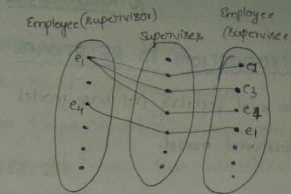
P3: $e_5\ e_7$

P4: $e_1\ e_5\ e_6$

## 8. RECURSIVE RELATIONSHIPS

RA: Every employee is supposed to have exactly one supervisor




Degree = 2

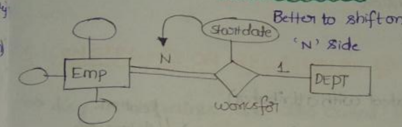
Cardinality of supervisor = N

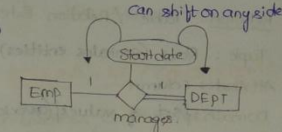
Supervisee = 1

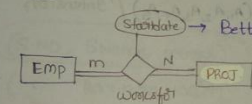
participation of supervisor = 0

of supervisee = 0

## 9. ATTRIBUTES TO RELATIONSHIPS




Startdate → Better not to shift on any side

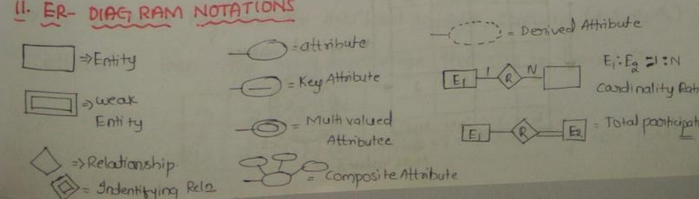
## 10. WEAK ENTITY

⇒ whenever any entity is not having any Key attribute then such an entity is called weak entity, so the weak entity should be associated to the strong entity with a Relationship called as "Identifying Relationship".

Identifying Relationship: ◇ Weak entity: ▭

⇒ V. Imp point : Always the participation of weak entity in an Identifying Relationship should be "Total participation".

## 11. ER- DIAGRAM NOTATIONS



# 2. RELATIONAL DB MODEL

## 1. INTRODUCTION TO RELATIONAL DATABASE

⇒ The most popular Database model used at Representational level is Relational model.

SQL , SEQUEL

ORACLE, IBm — RDBms

RDB

## 2. TERMINOLOGY OF RELATIONAL DATABASE

1. Relation : Table / Relation Extension.
2. Tuple : Row (contains entities)
3. Attribute : Column
4. Domain : {set of values} (associated with attributes)
5. Relational schema : Heading of the table = $R(A_1, A_2, A_3, A_4, A_5)$ / Relation Intension
6. Degree of a Relation : The no of Attributes

## 3. TUPLE, TUPLE VALUES AND NULL

⇒ whenever we store a Relation in a memory then it is stored in particular order.

⇒ In a Relation no two tuples can have the same values in all the attributes. (No duplicate values).

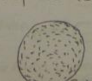
⇒ Some values of the attributes are not specified / present then we use "NULL" in that field.

⇒ For Example : Name Comprises of firstname, Middle Name, Lastname and now a person might not have middle name so Middle name = = NULL

| | | | |
|---|---|---|---|
| a | 1 | 2013 | 2 |
| | | | |
| | | | |
| a | 1 | 2013 | 2 |
| b | 2 | 2014 | NULL |

→ Duplicate values

NULL values.



Set

## 4. CONSTRAINTS ON RELATIONAL DB SCHEMA - DOMAIN CONSTRAINTS

1. Domain constraints ⇒ Entire schema should be Atomic.
2. Key constraints ⇒ No two Tuples should have same value
3. Entity- Integrity constraints ⇒ Entire touple should follow some constraints.
4. Referential Integrity constraints ⇒ Applied between two tables(Relations)

⇒ We can view a Relation as "Flat file structure:

| × S.NO | Name × | | |
|---|---|---|---|
| ✓ S.NO | FN | MN | LN |
| | | | |
| | | | |
| | | | |

Should be Atomic

Composite, Multivalued attributes are not allowed in Relations

## 5. CONSTRAINTS ON RELATIONAL DB SCHEMA - KEY CONSTRAINTS

⇒ Key-constraints are also called "Uniqueness Constraints."

(S.NO, SName, marks) ⇒ SUPER KEY

(1, Ravindra, 100)

(1, Ravindra, 100)

Student Marks

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| $t_1$ → | | | 100 |
| $t_2$ → | | | 100 |
| | | | |

$t_1 \neq t_2$

⇒ $\boxed{\text{SUPER KEY} \subseteq \text{Attributes}}$ for which no two touples have the same values in all the Attributes.

⇒ Any Minimal super key is a "Key". (SNO) $\boxed{\text{KEY} = \text{MINIMAL SUPERKEY}}$

⇒ Every Relation is going to have a super key by default and that super key is "set of all Attributes".

⇒ Any superset of a key is a superkey.

⇒ If A1 is a key, and the set/Relation/Table contains ($A_1 A_2 A_3 A_4$) attributes then the No. of super keys that can be formed is

| $A_1$ | $A_2$ | $A_3$ | $A_4$ |
|---|---|---|---|
| 1 | 2 | 2 | 2 |

= 1×2×2×2 = 8 super Keys are possible

⇒ If we are going to have two keys for a Relation then they are going to be candidate keys. (or) If we have two minimal super keys for a Relation then they are called "candidate keys." (8)

⇒ one of the keys of candidate keys is choosen and it is going to play some important role while we insert some numbers and that key is called "PRIMARY KEY"

$(\overset{\times}{A_1}\ A_2\ A_3\ A_4)$ - SK

$(\overset{\times}{A_2}\ A_3\ A_4)$ - SK

$(A_3\ A_4)$ - SK and Key

$(A_1\ A_2\ \overset{\times}{A_3}\ A_4)$

$(A_1\ A_2\ A_4)$ - SK and Key

$\therefore$ Keys = $\underbrace{(A_3 A_4\ ,\ A_1 A_2 A_4)}_{\text{Mimal superKeys}}$ ⇒ candidate Keys.

⇒ "primary key" doesnot allow "NULL" values.

## 6. ENTITY AND REFERENTIAL INTEGRITY CONSTRAINS

⇒ Entity Integrity says that no prime attribute should have null value

DNO = FOREIGN KEY

Employee

| ENO | ENA | DNO | PNO |
|---|---|---|---|
| 1 | a | 1 | 4 |
| | | | |

Department

| DNO | DNAME |
|---|---|
| 1 | |
| 2 | |
| 3 | |

"DNO" USUALLY WILL BE PRIMARY KEY IN DEPARTMENT RELATION.

| PNO | | |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |

⇒ Foreign Key can have "NULL" values unlike primary key.



## 7. ACTIONS UPON CONSTRAINT VIOLATIONS

⇒ The Actions that are performed on the database are

i) Insertion
ii) Deletion
iii) Update

} on performing these actions we should be sure that the constrains are not violated (Domain, Key, Entity, Referential constrains).

⇒ If the above actions violate the constrains the default action is to reject such actions which are resulting in violation.

⇒ while "deleting", the constraint that get violated is "Referential Integrity".

and the actions that must be taken are

1. Ignore it (Reject the action)
2. Cascade (Delete the tuple and also delete the tuples which are being referred by the above tuple & they should be deleted).
3. Set NULL OR some other value

## 8. COUNTING THE NO OF SK'S POSSIBLE - EXAMPLE - 1

⇒ Given a Relation $R(A_1\ A_2\ A_3 \cdots A_n)$

candidate key = $\{A_1\}$

Then the no. of possible super keys are

$$\begin{matrix} A_1 & \underbrace{A_2 \quad A_3 \quad A_4 \; \cdots \; A_n}_{(n-1)\text{ elements}} \\ \downarrow & \downarrow \quad \downarrow \quad \downarrow \qquad \downarrow \\ 1 & 2 \quad 2 \quad 2 \qquad 2 \end{matrix}$$

| A1 | A2 | |
|---|---|---|
| 1 | a | |
| 2 | a | |
| 3 | b | |
| 4 | b | |
| 5 | c | |

∴ Total no of keys = $1 \times 2 \times 2 \times 2 \times \cdots (n-1)$ times

$= 1 \times 2^{n-1}$

$$\boxed{\text{Total no. of SK's} = 2^{n-1}}$$

## 9. COUNTING THE NO·OF SK's POSSIBLE – EXAMPLE 2



Relation $R = (A_1\ A_2\ A_3 \cdots A_n)$

Candidate Key $= \{A_1 A_2\}$ candidate key is $A_1A_2$ not $A_1, A_2$

Single CK     Two CK's

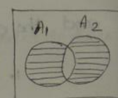
$$\underset{1}{A_1}\ \underset{1}{A_2}\ \underbrace{A_3\ A_4 \cdots A_{n-2}}_{(n-2)\text{ele}}$$

$2 \times 2 \times \cdots (n-2)$

$= 1 \times 1 \times 2 \times 2 \cdots (n-2)$ times

$$\boxed{SK = 2^{n-2} \text{ keys}}$$

Now, if candidate keys $= \{A_1, A_2\}$




The no·of super keys $= SK(A_1) + SK(A_2) - SK(A_1A_2)$

$= 2^{n-1} + 2^{n-1} - 2^{n-2}$

$$\boxed{SK = 2^n - 2^{n-2}}$$

## 10. COUNTING THE NO·OF SK'S POSSIBLE – EXAMPLE 3

Relation $R = (A_1\ A_2\ A_3 \cdots A_n)$

$CK = \{A_1, A_2A_3\}$

No·of SK's $= SK(A_1) + SK(A_2A_3) - SK(A_1A_2A_3)$

$$\boxed{SK = 2^{n-1} + 2^{n-2} - 2^{n-3}}$$

$CK = \{A_1A_2, A_3A_4\}$

No· of SK's $= SK(A_1A_2) + SK(A_3A_4) - SK(A_1A_2A_3A_4)$

$$\boxed{SK = 2^{n-2} + 2^{n-2} - 2^{n-4}}$$

$CK = \{A_1A_2, A_1A_3\} \Rightarrow$ No·of SK's $= SK(A_1A_2) + SK(A_1A_3) - SK(A_1A_2A_3)$

$$\boxed{SK's = 2^{n-2} + 2^{n-2} - 2^{n-3}}$$

11. COUNTI

Relation R=

CK

No·of SK

$\Rightarrow R = (A$

$CK = (A$

## 11. COUNTING THE No. OF SK's POSSIBLE - EXAMPLE - 4



Relation $R = (A_1\ A_2\ A_3 \dots A_n)$

$CK = (A_1, A_2, A_3)$

No. of SK's $= SK(A_1) + SK(A_2) + SK(A_3) - SK(A_1A_2) - SK(A_2A_3) - SK(A_1A_3) + SK(A_1A_2A_3)$

$$\boxed{SK = 2^{n-1} + 2^{n-1} + 2^{n-1} - 2^{n-2} - 2^{n-2} - 2^{n-2} + 2^{n-3}}$$

$\Rightarrow R = (A\ B\ C\ D)$

$CK = (A, BC)$

No. of SK's $= SK(A) + SK(BC) - SK(ABC)$

$= 2^3 + 2^2 - 2^1$

$$\boxed{SK = 8 + 4 - 2 = 10}$$

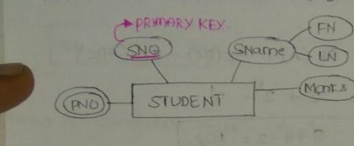
# 3. CONVERTION OF ER MODEL TO RELATIONAL MODEL



## 1. STEP 1

⇒ There are 7 steps to convert ERmodel (which is designed at conceptual level) to Relational model and RDms can be applied appropriately.

⇒ FOR EVERY ENTITY IN ER-MODEL WE HAVE TO COME UP WITH A RELATION IN RELATIONAL MODEL.




Now the Relation for this ER-diagram will be

Student | SNO | Marks | FN | LN |

⇒ Every simple attribute is represented in the table.

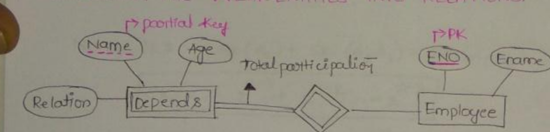
⇒ Composite attribute is further divided and atomic parts are represented in the table

⇒ Multivalued entities are not represented in the Relation.

⇒ Represent the primary key in the ER-model in the Relation also (underline)

## 2. STEP-2

⇒ CONVERTING THE WEAK ENTITIES INTO RELATIONS.




⇒ Create a table/Relation for the weak entity and add all the simple Attributes

⇒ Identify the partial key in the weak entity, and also identify the primary key in the owner entity (strong entity).

⇒ Now Add primary key of the owner entity (ENO) as the foriegn key of the weak entity, and make partial key of weak entity and the PK of strong entity as the PK of weak entity.

Depends Relation

| ENO | Name | Age | Relation |
|---|---|---|---|

↳ pk of weak entity.

Employee Relation.

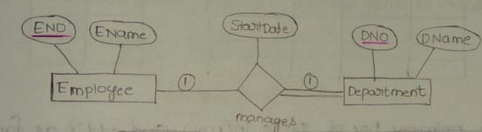
| ENO | EName |
|---|---|



⇒ The delete operation here is "CASCADE" Delete which means if you delete any row in the owner entity with some particular "ENO" then you should delete all the entries associated with that Eno in the weak entity also.

## 3. STEP-3

⇒ CONVERTION OF RELATIONSHIPS INTO RELATIONS (TABLES)




Employee

| ENO | EName |
|---|---|
| | |

ENO = Manager

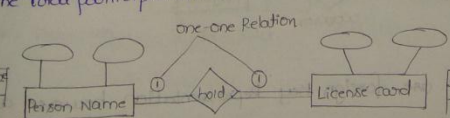
Department

| DNO | DName | ENO | SD |
|---|---|---|---|
| | | | |

⇒ Now Add the primary key of one side (one table) as the foriegn key of another table, generally add the pk of (non-total participation) side as the foriegn key of the total participation table.

⇒ If the Relationship itself contains attributes add the attributes on the total participation side.

| PNO | PName |
|---|---|
| | |




| LNO | L.Date |
|---|---|
| | |

⇒ when you have total participation on both the sides and one-one Relationship exists on both sides then In both the Relations the no. of tuples will be equal and In that case there is no need of adding the primary key of one side onto the other side we can just combine the two relations into a single Relation.

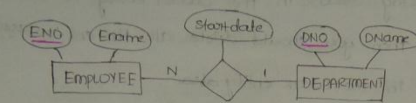
| PNO | PName | ENO | LDate |
|---|---|---|---|

and it also takes less space.

(14)

## 4. STEP-4

⇒ CONVERSION OF 1:N RELATIONSHIP INTO A RELATION.




Employee

| ENO | Ename | DNO | Start Date |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

Department

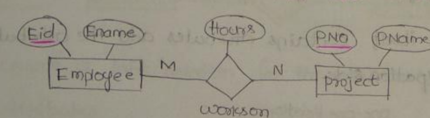
| DNO | DName |
|---|---|
| | |
| | |

⇒ Take the primary key of 1side in(1:N) and add it as foriegn key to the 'N' side

⇒ The Attributes on the Relationships are shifted to the 'N' side.

## 5. Step-5

⇒ CONVERSION OF MANY TO MAY RELATIONSHIP




⇒ we cannot use foreign key Representation because an employee can work for more than one project and a project can have more than one employee

⇒ Employee 1 works on 1, 2, 3

| Eid | En | PNO |
|---|---|---|
| 1 | | 1,2,3 |

| PNO | PNa |
|---|---|
| | |

✗

| Eid | En |
|---|---|
| | |

| PNO | PN | Eid |
|---|---|---|
| 1 | | 1,2 |

Project 1 has more than one employee ✗

⇒ The solution to ... attributes as ...

Eid+PNO= ...

⇒ Now the at... Newly formed ...

## 6. STEP-6

⇒ DEALING W...

⇒ For the mult...

⇒ Create a ... key of the ...

| PNO | SNO |
|---|---|
| P1 | 1 |
| P2 | 1 |

∴ PK = SNO ...

⇒ The solution to the above problem is create a new table having attributes as the primary keys of the participating entities.



| Eid | P.NO |
|---|---|
| | |

| Eid | E name |
|---|---|
| | |

| PNO | PName |
|---|---|
| | |

$\boxed{\text{Eid} + \text{PNO} = \text{PK of Newtable}}$

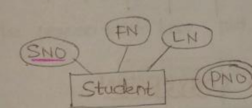
⇒ Now the attribute on the Relationship should be added to the Newly formed table.

| Eid | PNO | Hours |
|---|---|---|
| | | |

## 6. STEP-6

⇒ DEALING WITH MULTIVALUED ATTRIBUTES

⇒ For the multivalued attribute we are going to create a new table




⇒ Create a Relation having the multivalued attribute and primary key of the entity (SNO, PNO)

| SNO | PNO |
|---|---|
| | |
| | |

⇒ $\boxed{\text{SNO} + \text{PNO} = \text{PK for this table}}$

| PNO | SNO |
|---|---|
| P1 | 1 |
| P2 | 1 |

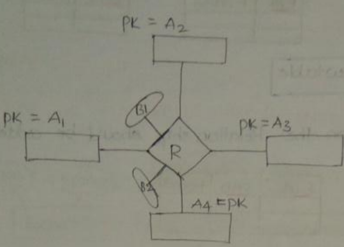
∴ PK = SNO + PNO

| SNO | FN | LN |
|---|---|---|
| 1 | a | b |
| | | |

∴ PK = SNO

## 7. STEP-7

⇒ DEALING WITH N-ARY RELATION SHIPS (MORE THAN 2 ENTITIES)



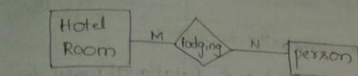

⇒ Now create a new table having the attributes as primary Keys of all the entities.

| A1 | A2 | A3 | A4 | B1 | B2 |
|---|---|---|---|---|---|
| | | | | | |

## 8. SUMMARY OF ER TO RDB CONVERSION

| ER-Model | Relational model |
|---|---|
| Entity type | " Entity " Relation |
| 1:1 and 1:N Relationship type | Foreign Key (or Relation) |
| M:N Relationship type | Relationship "relation" + 2 FK's |
| N-ary Relationship type | Relationship "relation" + n FK's |
| Simple Attribute | Attribute |
| Composite Attribute | set of simple component Attributes |
| Multivalued Attribute | Relation and Foreign Key. |
| value set | Domain |
| Key Attribute | primary Key. |

## 9. GATE IT 2005 QUESTION ON ER-DIAGRAMS




Lodging is many-many Relationship. Rent, payment to be made by person(s) occupying different hotel rooms should be added as an attribute to

a) Hotel ✓b) lodging c) person d) None.

Sol: Let us say Hotel Room contains HNO, HName, and if we add Rent

| HNO | HName | Rent |
|---|---|---|
| 1 | a | |
| 2 | b | |

→ Many people can Reside and we cannot put all the Rents in single tuple ⇒ option A ✗

⇒ Let us say the person table has pid, pname, Rent

| Pid | pname | Rent |
|---|---|---|
| 1 | a | |
| | | |

→ He may stay at many hotels and each hotel has its own Rent. so we cannot put all the Rents here, ⇒ option C ✗

⇒ In case of many-many Relationship we create a new table having the PK of participating entities.

| Pid | HNO | Rent |
|---|---|---|
| 1 | 1 | 10,000 |
| 2 | 2 | 5000 |

✓ ⇒ Table of the Relationship lodging.

∴ The Attribute Rent should be on Lodging.

## 10. GATE 12 QUESTION ON CONVERTING ER TO RDB

Given the basic ER and Relational models, which of the following is Incorrect

a) An attribute of an entity can have more than one value

b) An attribute of an entity can be composite.

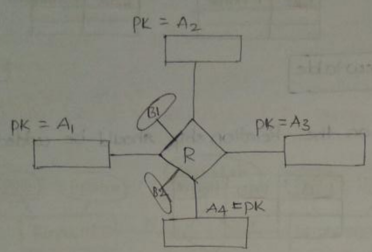
✓c) In a row of Relational table, an attribute can have more than one value

d) In a row of Relational table, an attribute can have exactly 1 value or NULL

## 7. STEP-7



⇒ DEALING WITH N-ARY RELATIONSHIPS (MORE THAN 4 ENTITIES)




⇒ Now create a new table having the attributes as primary keys of all the entities.

| A1 | A2 | A3 | A4 | B1 | B2 |
| --- | --- | --- | --- | --- | --- |
| | | | | | |

## 8. SUMMARY OF ER TO RDB CONVERSION

| ER-Model | Relational model |
| --- | --- |
| Entity type | "Entity" Relation |
| 1:1 and 1:N Relationship type | Foreign Key (or Relation) |
| M:N Relationship type | Relationship "relation" + 2 FK's |
| N-ary Relationship type | Relationship "relation" + n FK's |
| Simple Attribute | Attribute |
| Composite Attribute | Set of simple component Attributes |
| Multivalued Attribute | Relation and Foreign Key. |
| Value set | Domain |
| Key Attribute | primary Key. |

## 9. GATE IT

Hotel
Room

Lodging is n
by person(s)
attribute to
a) Hotel b)

Sol: Let us
Rent

⇒ Let us say

⇒ In case
the pk

∴ The f

## 10. GATE

Given the b
Incorrect
a) An attrib
b) An attrib
c) In a
d) In

16

ENTITIES)

nary Keys

on)

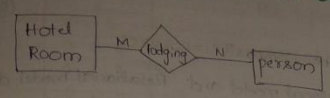
2FK's

n FK's

tributes



## 9. GATE IT 2005 QUESTION ON ER-DIAGRAMS




Lodging is many-many Relationship, Rent, payment to be made by person(s) occupying different hotel rooms should be added as an attribute to

a) Hotel ~~b)~~ lodging c) person d) None.

Sol: Let us say Hotel Room contains HNO, HName, and if we add Rent

| HNO | HName | Rent |
|---|---|---|
| 1 | a | |
| 2 | b | |

→ Many people can Reside and we cannot put all the Rents in single tuple ⇒ option A ✗

⇒ Let us say the person table has pid, pname, Rent

| Pid | Pname | Rent |
|---|---|---|
| 1 | a | |
| | | |

→ He may stay at many hotels and each hotel has its own Rent, so we cannot put all the Rents here, ⇒ option C ✗

⇒ In case of many-many Relationship we create a new table having the PK of participating entities.

| Pid | HNO | Rent |
|---|---|---|
| 1 | 1 | 10,000 |
| 2 | 2 | 5,000 |

✓ ⇒ Table of the Relationship Lodging.

∴ The Attribute Rent should be on Lodging.

## 10. GATE 12 QUESTION ON CONVERTING ER TO RDB

Given the basic ER and Relational models, which of the following is Incorrect

a) An attribute of an entity can have more than one value

b) An attribute of an entity can be composite.

~~c)~~ In a row of Relational table, an attribute can have more than one value

d) In a row of Relational table, an attribute can have exactly 1 value or NULL

a) option a talks about ER-model and multivalued Attributes in ER-model = CORRECT

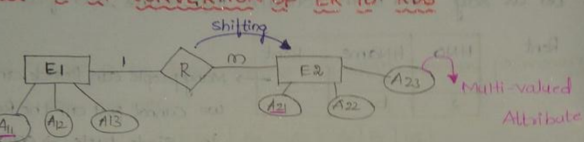
18

b) In ER-model Attributes can be composite = CORRECT

c) option 'c' talks about Relational model and Relational model doesn't allow multivalued Attributes and Composite Attributes = INCORRECT

d) option 'd' is correct since there should be exactly one value in each field and can have NULL values = CORRECT

## 11. GATE IT 04 CONVERTION OF ER TO RDB




Min no. of tables: ___

$\therefore$ $E_1(\underline{A_{11}}\ A_{12}\ A_{13})$

$E_2(\underline{A_{21}}\ R\ A_{22})$

$E_3(A_{23}, A_{11}, A_{21})$

3 Tables.

## 12. GATE 05 ON CASCADE DELETE IN CASE OF FOREIGN KEYS

The following table has two attributes A and C where A is the primary key and C is FK referencing with A on-delete cascade.

The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple (2,4) is deleted is:

a) (3,4) and (6,4) (b) (5,2)(7,2) (c)(5,2)(7,2)(9,5) d) (3,4)(4,3)(6,4)

| A | C |
|---|---|
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |
| 5 | 2 |
| 7 | 2 |
| 9 | 5 |
| 6 | 4 |

Sol

| A | C |
|---|---|
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |
| 5 | 2 |
| 7 | 2 |
| 9 | 5 |
| 6 | 4 |

14. GATE 08 QUES

$E_1, E_2$ - two entities

$R_1 R_2$ are two Rela

and $R_2$ is many t

own. what is the

Sol

PK = $A_1$ E

1

15. GATE 97 QU

Let R(a,b,c) and

that refers to t

'R' and 'S'.

a) Insert into R

which of the fo

a) None of a

b) All of a,b,c

c) Both a,b

d) Both b,c

Sol

| A | C |
|---|---|
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |
| 5 | 2 |
| 7 | 2 |
| 9 | 5 |
| 6 | 4 |

∴ The tuples that must be additionally deleted are (5,2) (7,2) (9,5) (9)

⇒ (2,4) is deleted ⇒ delete the rows containing two(2)

⇒ Now on deleting we are deleting the rows (5,2) (7,2) because they contain '2'

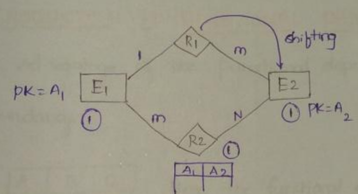
⇒ Now delete the rows containing (5)(7) = (9,5) deleted.

## 14. GATE 08 QUESTION ON CONVERTING ER TO RDB

$E_1 E_2$ - two entities

$R_1 R_2$ are two Relationships between $E_1$ and $E_2$. $R_1$ is one-to-many and $R_2$ is many to many. $R_1$ and $R_2$ doesnot have any attributes of their own. what is the min no. of tables required?

Sol




Min. no. of tables = 3

## 15. GATE 97 QUESTION ON REFERENTIAL INTEGRITY

Let R(a,b,c) and S(d,e,f) be two relations in which 'd' is the FK of 'S' that refers to the primary key of R. Consider the following four operations 'R' and 'S'.

a) Insert into R b) Insert into 'S' c) Delete from R d) Delete from 'S'.

Which of the following is true about the referential integrity constraint above?

a) None of a,b,c,d can cause its violation

b) All of a,b,c,d can cause violation

c) Both a,b can cause its violation

d) Both b,c can cause its violation.

801

R

| a | b | c |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |
| 6 | | |
| 10 | | |

S

| d | e | f |
|---|---|---|
| 1 | | |
| 1 | | |
| 2 | | |
| 2 | | |
| 3 | | |
| × 7 | | |
| 6 | | |

Violation (Inserting into 'S').

V.V. Imp

⇒ 'd' is depending on 'a' not 'a' is depending on 'b'.

⇒ Inserting into "S" and Deletion from 'R' are going to cause violations. (may cause. we are not sure about it).

20

# 4. NORMALISATION

## 1. INTRODUCTION TO NORMALISATION

⇒ If we hold the entire data in a single table it will take more space,

⇒ Less Redundancy

⇒ various Anamolies will occur
- Insert Anamolies
- Deletion Anamolies
- Update Anamolies.

⇒ splitting the tables into small tables such that our design will not contain all the above Anamolies and Redundancy is called "NORMALISATION".

⇒ In order to do Normalisation we use the concept of Functional Dependencies (FD's). and the concept of Candidate keys.

## 2. INTRODUCTION TO FUNCTIONAL DEPENDENCIES

⇒ The Advantage of the functional dependency is it Reduces "Redundancy."

| | A | B | C |
|---|---|---|---|
| | 1 | | |
| $t_1$ → | 2 | a | b |
| | 3 | | |
| $t_2$ → | 2 | a | b |

Here the functional dependency is $A \to BC$

⇒ for a value of 'A' you can get/derive the values of 'B' and 'C' uniquely

$A \to BC$

$2 \to ab$.

(or) $\boxed{\text{if } t_1(A) = t_2(A) \text{ then } t_1(BC) = t_2(BC)}$ ⇒ $\boxed{\text{if } (2=2) \text{ then } (ab)=(ab)}$ ✓

⇒ Initially we see that the table is in 1st Normal form.

⇒ Next 2nd NF

⇒ Next 3NF

⇒ Next BCNF

Now find $G \supseteq F$ then,

$F: \{A \to C, AC \to D, E \to AD, E \to H\}$ check whether these are derivable from 'F'.

$G$:

$A^+ = \{A, C, D\}$

$(AC)^+ = \{A, C, D\}$

$(E)^+ = \{E, A, H, C, D\}$

$\therefore$ all the functional dependencies in F are covered by G

$\therefore$ Both the FD's 'F' and 'G' are Equivalent.

## 29. EQUIVALENCE OF FD's EXAMPLE-1

$F: \{A \to B, B \to C, C \to D\}$

$G: \{A \to BC, C \to D\}$

check whether these two FD's are Equalent or not?

Sol

$F \supseteq G \Rightarrow$ Take Each FD of G and check whether it is derivable from 'F'.

$\Rightarrow A \to BC \Rightarrow$ Now Take $A^+$ from F and check BC is present in $A^+$

$A^+ = \{A, B, C, D\} \therefore A \to BC$ holds

$\Rightarrow C \to D \Rightarrow C^+ = \{D, C\} \Rightarrow$ holds $\therefore F \supseteq G$

$G \supseteq F \Rightarrow$ The FD's of 'F' are $A \to B, B \to C, C \to D$ check if they are covered by G or not.

in G: $A^+ = \{A, B, C\}$ $\{A \to B$ holds$\}$

$B^+ = \{B\}$ $\{B \to C$ is not covered by G$\}$

$\therefore G \not\supseteq F$

$\therefore$ Both the functional dependencies are not Equivalent.

## 30. EQUIVALENCE OF TWO FD'S EXAMPLE - 2

① $F: \{A\to B, AB\to C, D\to AC, D\to E\}$

$G: \{A\to BC, D\to AB\}$

② $F: \{A\to B, B\to C, C\to A\}$

$G: \{A\to BC, B\to A, C\to A\}$

Sol: $F \supseteq G \Rightarrow A^+ \text{ in } F = \{A,B,C\}$ ✓

$\Rightarrow D^+ \text{ in } F = \{D,E,A,C,B\}$ ✓ $\quad F \supseteq G$

$G \supseteq F \Rightarrow A^+ \text{ in } G = \{A,B,C\} \quad A\to B$ ✓

$\Rightarrow (AB)^+ = \{A,B,C\} \quad AB\to C$ ✓

$\Rightarrow D^+ = \{A,B,D,C\} \quad D\to AC$ ✓ $\quad \therefore G \not\supseteq F$

$\Rightarrow D^+ = \{A,B,C,D\} \quad D\to E$ ✗

$\therefore$ These two FD's are not Equivalent

②

$F \supseteq G \Rightarrow A^+ \text{ in } F = \{A,B,C\} \quad A\to BC$ ✓

$\Rightarrow B^+ \text{ in } F = \{B,C,A\} \quad B\to A$ holds ✓ $\quad \therefore \boxed{F \supseteq G}$

$\Rightarrow C^+ \text{ in } F = \{A,B,C\} \quad C\to A$ holds

$G \supseteq F \Rightarrow$ Now, $A^+ \text{ in } G = \{A,B,C\} \quad A\to B$ ✓

Now, $B^+ \text{ in } G = \{B,A,C\} \quad B\to C$ ✓ $\quad \therefore \boxed{G \supseteq F}$

Now, $C^+ \text{ in } G = \{C,A,B\} \quad C\to A$ holds ✓

$\therefore$ Bothe the functional dependencies are Equivalent.

## 31. MINIMAL COVER

If we have a set of functional Dependencies 'F' and if we could minimise it to other set of functional dependencies 'G' such that 'G' covers 'F' and 'F' covers 'G' and 'G' is minimal, then 'G' is called Minimal cover of 'F'.

## PROCEDURE TO FIND MINIMAL SET

1. Split the FD's such that RHS contain single attribute.

   Ex: $A \to BC$, $A \to B$ & $A \to C$

2. Find the Redundant FD's and delete them from the set.

   Ex: $\{A \to B, B \to C, A \to C\} \Rightarrow \{A \to B, B \to C\}$

3. Find the Redundant attributes on LHS and delete them.

   Ex: $AB \to C$, A - can be deleted if $B^+$ contains 'A' (or) $\Rightarrow B \to C$
   B - can be deleted if $A^+$ contains 'B' $\Rightarrow A \to C$

1) Minimize $\{A \to C, AC \to D, E \to AD, E \to H\}$

① $A \to C$, $AC \to D$, $E \to A$, $E \to D$, $E \to H$.

This production is useless (Remove/delete this production and try to find $E^+$ in the Remaining if it contains 'D' then "$E \to D$" is Redundant FD.

$E^+ = \{A, E, H, C, D\} \Rightarrow$ "$E \to D$" is Redundant.

② $A \to C$, $AC \to D$, $E \to AD$, $E \to H$

→ This will be Redundant if $C^+$ contains $A$}

$C^+ = \{C\}$ ∴

→ This will be Redundant if $A^+$ contains C then,

$A^+ = \{A, C\}$ $\Rightarrow AC \to D$ becomes $A \to D$

∴ $A \to C$, $A \to D$, $E \to A$, $E \to H$, $E \to D$

$\Rightarrow A \to CD$, $E \to AH$

## 32. MINIMAL COVER EXAMPLE - 1

Minimize $\{A \to B, C \to B, D \to ABC, AC \to D\}$

① $A \to B$ $C \to B$ $D \to A$ $\cancel{D \to B}$ $D \to C$ $AC \to D$.

Redundant $\Rightarrow D^+ = \{D, A, B\}$ $D \to B$ is derivable

Now, $(AC)^+ = \{ACB\}$ $AC \to D$ is not Redundant.

② $A \to B, C \to B, D \to A, D \to C, AC \to D$

Now, $AC \to D$ canbe deleted if $A^+$ contains C (or) $C^+$ contains A

Now, $A^+ = \{A, B\}$ $\therefore$ $AC \to D$ cannot be deleted.

$C^+ = \{C, B\}$

## 33. GATE - 2013 ON MINIMAL COVER

Is $\{AB \to C, D \to E, E \to C\}$ is the minimal cover of $\{AB \to C, D \to E, AB \to E, E \to C\}$

Sol

Step-1: $AB \to C$ $\quad D \to E$ $\quad \underbrace{AB \to E}$ $\quad E \to C$

$AB^+ = \{A, B, C\}$ $\therefore$ This cannot be deleted and must be in the minimal set

$\therefore$ $\{AB \to C, D \to E, E \to C\}$ is not the minimal cover, The minimal cover should be $\{D \to E, AB \to E, E \to C\}$

## 34. LOSSLESS DECOMPOSITION

⇒ Mainly Normalisation is about splitting the tables i.e decomposing the tables, so while decomposing we should see some properties. One such property is "Lossless decomposition."

⇒ when we decompose a Relation we should check that there is a common attribute in both of them, if not check what happens in below example.

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

A

BC

⇒

| A | BC |
|---|---|
| $a_1$ | $b_1c_1$ |
| $a_2$ | $b_2c_2$ |

cross product

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |
| $\boxed{a_2}$ | $\boxed{b_2}$ | $\boxed{c_2}$ |

→ New Tuple.

→ Spurious tuple

Now, if i choose a common Alphabet Randomly.

$$ABC \rightarrow AB,\ AC$$

| A | B |
|---|---|
| $a_1$ | $b_1$ |
| $a_2$ | $b_1$ |
| $a_1$ | $b_2$ |

| A | C |
|---|---|
| $a_1$ | $c_1$ |
| $a_2$ | $c_1$ |
| $a_1$ | $c_2$ |

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

Spurious Tuples.

New Tuples that are formed but they are not present in the given table.

**LOSSY DECOMPOSITION**

$$ABC \rightarrow AB,\ BC$$

| A | B |
|---|---|
| $a_1$ | $b_1$ |
| $a_2$ | $b_1$ |
| $a_1$ | $b_2$ |

| B | C |
|---|---|
| $b_1$ | $c_1$ |
| $b_2$ | $c_2$ |

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_2$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

**LOSLESS DECOMPOSITION**

⇒ The decomposition is loseless when the common attribute is key in one of the tables. (AB or BC)

∴ For Loseless decomposition.

$$R_T \rightarrow R_1,\ R_2$$

$(R_1 \cap R_2) \rightarrow R_1$ | $(R_1 \cap R_2) \rightarrow R_1 - R_2$

$(R_1 \cap R_2) \rightarrow R_2$ | $(R_1 \cap R_2) \rightarrow R_2 - R_1$

The common attribute should be a key in any one of the relation.

## 35. FD PRESERVING

$R(A_1 A_2 A_3 \dots A_n)$

FD — $F^+$ {set of all functional dependencies that are applicable on R}

R → R1, R2

$F_1 \subseteq F^+$ $\quad$ $F_2 \subseteq F^+$

Now, dependency preserving means $\boxed{(F_1 \cup F_2)^+ = F^+}$

⇒ The dependency preserving is not a mandatory thing

## 36. DECOMPOSITION - EXAMPLE 1

R(ABC) . FD: $\{A \to B, B \to C, C \to A\}$

$R_1(AB)$ $\quad$ $R_2(BC)$.

⇒ Now we need to check the lossless decomposition and dependency preservation.

⇒ $R_1(AB)$ $R_2(BC)$ ⇒ This decomposition is lossless because the common variable in $R_1 R_2 = B$ and 'B' is a key attribute in $R_2(BC)$ $\{\because B \to C\}$.

| Now, $R_1(AB)$ | $R_2(BC)$ |
|---|---|
| ✓ $A \to B$ | ✓ $B \to C$ |
| $B \to A$ ✓ | ✓ $C \to B$ } are the Non-trivial Dependencies. |
| Non-Trivial Dependencies | $B^+ = \{BCA\}$ $(B \to C)$✓ |
| | $C^+ = \{CAB\}$ $(C \to B)$✓ |

$F_1 \cup F_2 = \underbrace{A \to B \quad B \to A}_{①} \quad \underbrace{B \to C}_{②} \quad C \to B$

$= A \to B \quad B \to A \quad \underbrace{A \to C} \quad \underbrace{C \to B}_{\text{Redundant}}$

$= A \to B \quad B \to C \quad C \to A$

∴ The decomposition is lossless and Dependency preserving.

## 37. DECOMPOSITION EXAMPLE 2



R(ABCD)

F: $(A \to B, B \to C, C \to D, D \to A)$

D: (AB, BC, CD)

**Sol** Given D = (AB, BC, CD) (or) D = (AB BC CD)

$B^+ = \{B, C, D, A\}$

$C^+ = \{A, B, C, D\}$

LOSELESS DECOMPOSITION

(ABC) (CD) → $C^+ = \{A, B, C, D\}$

(ABCD) ∴ creately

$C^+ = \{ABCD\}$

(AB) (BCD) LOSELESS DECOMPOSITION

(ABCD)

$B^+ = \{ABCD\}$

Now comming to functional dependencies

| $R_1$(AB) | $R_2$(BC) | CD = $R_3$ |
|---|---|---|
| ✓ $A \to B$ | $B \to C$ ✓ | $C \to D$ ✓ → Given in Question |
| ✓ $B \to A$ | $C \to B$ ✓ | $D \to C$ |
| $B^+ = \{ABCD\}$ | $C^+ = \{ABCD\}$ | $D^+ = \{ABCD\}$ |

Now, $F_1 \cup F_2 \cup F_3 = A \to B, B \to C, C \to D, D \to A$ {indirectly covered}

$\boxed{F_1 \cup F_2 \cup F_3 = F}$ F.D. is preserved. (Both)

## 38. DECOMPOSITION EXAMPLE 3

R(ABCD)

F = $\{AB \to CD, D \to A\}$

D = {AD, BCD}

**Sol** The common attribute is 'D' now 'D' should be a key in any one of the Relations $R_1$(AD) $R_2$(BCD)

Now, $D^+ = \{D, A\}$ $D \to A$ ∴ 'D' is a key attribute in $R_1$.

∴ The decomposition is losless decomposition

Now,

| $R_1(AD)$ | $R_2(BCD)$ | |
|---|---|---|
| $A \to D$ × | $B^+ = (B)$ × | $(BC)^+ = (BC)$ × |
| $D \to A$ ✓ | $C^+ = (C)$ × | $(BD)^+ = (BDAC)$ ✓ |
| Now, $A^+ = \{A\}$ | $D^+ = (D, A)$ × | $(CD)^+ = (CDA)$ × |

∴ $\boxed{BD \to C}$

Now. $F_1 = D \to A$, $F_2 = BD \to C$ | $F_1 \cup F_2 = D \to A,\ BD \to C$ × ($(AB)^+ = (AB) \Rightarrow$ ∴ The FD $AB \to CD$ is not preserved)

∴ The Decomposition is losless and non-FD preserving.

## 39. DECOMPOSITION EXAMPLE 4

R (ABCDEG)

F: $\{AB \to C, AC \to B, AD \to E, B \to D, BC \to A, E \to G\}$

D: (ABC, ABDE, EG).

D: ABC ABDE EG

ABCDE EG

ABCDEG

Now $AB^+ = \{A, B, C, D, \}$ Now, $E^+ = \{E, G\}$

∴ $AB \to C$ is possible ⇒

The decomposition is loseless

Now

| $R_1(ABC)$ | $R_2(ABDE)$ | $R_3(EG)$ | |
|---|---|---|---|
| $A \to A$ × | $B \to D$. | $E \to G$ | $A^+ = A$ |
| $B \to D$ × (D not in table ABC) | $AB \to DE$ | | $B^+ = BD$ |
| $C \to C$ × | $AD \to E$ | | $C^+ = C$ |
| $(AB)^+ = ABCDEG \Rightarrow AB \to C$ | $(BE) \to D$ | | $(AB)^+ = ABCDEG$ |
| $BC \to A$ | $ABD \to E$ | | $(BC)^+ = BCDAEG$ |
| $AC \to B$ | $ABE \to D$ | | $(AC)^+ = ACBDEG$ |
| | ~~ABDE→D~~ | | |

∴ The

40. DECOMPO

R(ABCDE)

F: (A → BC, C

R₁: (ABCD) R₂

Now, Given

∴ ABCD

$A \to BCD$ ✓

$B \to B$ ×

$C \to D$ ✓

$D \to D$ ×

$(BC) \to D$.

Now $A^+ =$ AB

$B^+ =$ B

$C^+ =$ C

$D^+ =$ D,

$(BC)^+ =$

$(BD)^+ =$

$(CD)^+ =$

$(AB)^+$

$(AC)^+$

$(AD)^+$

(42)

∴ The decomposition is lossless and Dependency preserving.

## 40. DECOMPOSITION EXAMPLE 5

(43)

R(ABCDE)

F: (A → BC, C → DE, D → E)

$R_1$: (ABCD) $R_2$: (DE).

Now, Given (ABCD) (DE)

∴ ABCD

A → BCD ✓

B → B ✗

C → D ✓

D → D ✗

(BC) → D.

Now D⁺ = {E, D}

$R_2$ (DE)

D → E ✓

E → D ✗ $E^+ = \{E\}$

Now $A^+$ = ABCDE

$B^+$ = B

$C^+$ = CDE

$D^+$ = D, E

$(BC)^+$ = BCDE

$(BD)^+$ = BDE

$(CD)^+$ = CDE

∴ FD's = 
A → BCD
C → D
BC → D
D → E
} = $F_1 \cup F_2$ = F

$(AB)^+$, $(AC)^+$, $(AD)^+$ } I will not include becuase 'A' itself is going to derive every thing the obviously AB, AC, AD will derive all Attributes and they will become SK's.

∴ The decomposition is lossless and dependency preserving.

= (AB) ⇒ ∴ The AB → CD is not preserved)

= {E, G}

A
BD
C
= ABCDEG
= BCDAEG

# 41. DECOMPOSITION EXAMPLE -6



R(ABCDEG)

F :{ AB→C, AC→B, AD→E, B→D, BC→A, E→G }

D: (AB, BC, ABDE, EG)

D: (AB BC ABDE EG)

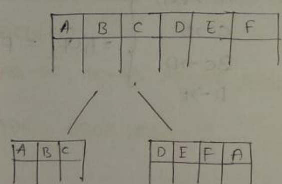
ABC

Now, $B^+ = \{B, D\}$ B is not the key of A,C

∴ The decomposition is lossy decomposition.

# 42. FIRST NORMAL FORM

⇒ The process of Removing the Redundancy is Normalisation

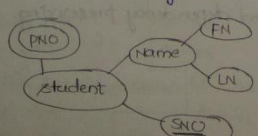



⇒ Our ultimate aim is to Reduce a table to BCNF, it is known that by the time you convert the table into BCNF there wont be any Redundancy (Redundancy = 0%)

⇒ 1NF → 2NF → 3NF → BCNF

⇒ 1NF says that the table has to be flat

⇒ By default Every Relational DataBase is in First Normal form




⇒

Student

| SNO | FN | LN |
|---|---|---|
| | | |

phone no.

| SNO | PNO |
|---|---|
| | |

⇒ No multivalued and composite values are allowed in 1NF



## SECOND NORMAL FORM INTRODUCTION

⇒ Let us assume we have a table having attributes ABC and the functional dependencies that are allowed are $AB \rightarrow C$, $B \rightarrow C$

⇒ Now, $\left.\begin{array}{l} AB \rightarrow C \\ B \rightarrow C \end{array}\right\}$ $\left.\begin{array}{l} A^+ = A \\ B^+ = BC \\ C^+ = C \end{array}\right\}$ with one attribute, key is not possible ∴ Try with 2 attributes.

$(AB)^+ = (ABC)$ ∴ (AB) has the capacity to become Candidate Key.

Now, $AB \rightarrow C$ → say that AB (Combination) can determine 'C'

$B \rightarrow C$ → Say that B (itself) can determine 'C' uniquely

| A | B | C |
|---|---|---|
| 1 | a | c1 |
| 2 | a | c1 |
| 1 | b | c2 |
| 2 | b | c2 |
| 1 | c | c3 |
| 2 | c | c3 |
| 3 | c | c3 |
| 4 | c | c3 |
| 5 | c | c3 |

⇒ The 2NF says that if your candidate key is containing more than one attribute then a part of the key should not determine anything else.

⇒ 2NF is based on full Functional Dependency

⇒ partial dependency is not allowed (A part of the key determines some thing).

↓

This table is Not in 2NF because there is a partial dependency ($B \rightarrow C$) in the candidate key $(\underbrace{AB}_{\text{partial key}} \rightarrow C)$ so we should eliminate partial dependency.

Find $B^+ = \{BC\}$

$A^+ = \{A\}$

ABC

AB | BC → $B^+ = BC$

$\boxed{B \rightarrow C}$ is the only dependency.

× $A^+ = A$

× $B^+ = BC$

× $AB = \{ABC\}$

Loseless decomposition.

Not FD preserving.

# 44. 2NF EXAMPLE 1



R(ABCD)

FD: {AB → C, B → D} what is the highest Normal Form satisfied?

⇒ By default Every Relation will be in 1NF

⇒ Now find all the candidate keys

$A^+ = A$

$B^+ = BD$ ∴ (AB) is the candidate key.

$AB^+ = ABCD$

Now the three attribute candidate keys are possible and they should not contain (AB) if they include then it will become superkey.

$(ACD)^+ = (ACD)$

$(BCD)^+ = (BCD)$

∴ (AB) → ABCD

B → D → partial Dependency

Now, find $A^+ = A$ $B^+ = BD$

R(ABCD)

R(ACB) R(BD)

Remaing / we inserted so that we should have some common part

R(ABCD)

R(ACB) R(BD)

| A | C | B |
|---|---|---|

| B | D |
|---|---|

AB → C B → D

FD preserving too.

Lossless decomposition.

45. 2NF E

R(ABCDE

AB → C

BD → EF

AD → GH

A → I

H → J

Now, find

of left ha

all the c

$(AB)^+ =$

R(ABCDEF

AB → C

| A | B |
|---|---|

$A^+ = (A, I$

| A | I |
|---|---|

A → I

46. 2NF E

R(ABCDE)

F: {A → B, B -

The par

Now,

## 45. 2NF EXAMPLE 2

R(ABCDEFGHIJ)

AB → C
BD → EF
AD → GH
A → I
H → J

Candidate Key = ABD.

(ABD) → (ABCDEFGHIJ)

$$\left.\begin{array}{l} BD \rightarrow EF \\ AD \rightarrow GH \\ A \rightarrow I \\ AB \rightarrow C \end{array}\right\} \text{partial Dependencies}$$

Now, find all the closures of all partial Dependencies (find closures of left hand side) and try to create a new table by taking away all the attributes which are determined by such dependencies

$(AB)^+ = (\underline{ABC})$

$R(AB\cancel{C}DEFGHIJ)$

AB→C

| A | B | C |
|---|---|---|
| | | |

$(BD)^+ = (\underline{BD}EF)$

$R(A\cancel{B}\cancel{C}\cancel{D}\cancel{E}\cancel{F}\ GHIJ)$

BD→EF

| B | D | E | F |
|---|---|---|---|
| | | | |

$(AD)^+ = (A, D, G, H, J)$

$R(\cancel{A}B\cancel{C}\cancel{D}\cancel{E}\cancel{F}\ \cancel{G}\cancel{H}\cancel{I}\cancel{J})$

| A | D | G | H | J |
|---|---|---|---|---|
| | | | | |

AD → GH
H → J

$A^+ = (A, I)$

| A | I |
|---|---|
| | |

A→I

ABD

| A | B | D |
|---|---|---|
| | | |

## 46. 2NF EXAMPLE 3

R(ABCDE)

Candidate Key = (AC)

F: {A→B, B→E, C→D}

The partial dependencies are A→B
C→D

Now, $A^+ = \{A, B, E\}$

$R(A\cancel{B}CD\cancel{E})$

A→B
B→E

| A | B | E |
|---|---|---|
| | | |

$C^+ = \{C, D\}$

$R(A\cancel{B}C\cancel{D}\cancel{E})$

C→D

| C | D |
|---|---|
| | |

AC

| A | C |
|---|---|
| | |

Now, if the candidate key for a table is a single attribute and that is the only candidate key then the Relation is in 2nd NF.



## 47. THIRD NORMAL FORM INTRODUCTION

3NF: No Transitive Dependencies

Transitive Dependency: Non prime attribute Transitively depending on the key.

R(ABC)

FD: $\{A \to B, B \to C\}$  CK = $\{A\}$ ⇒ No partial Dependencies.

Prime attribute: The attribute which is a part of some candidate key.

Non-prime attribute: The attribute which is not a part of some candidate the key.

$A \to B$  $B \to C$ ⇒ Transitive Dependency

Now, $B^+ = \{B, C\}$

R (A B $\cancel{C}$)

| A | B |
|---|---|
| | |

$A \to B$

| B | C |
|---|---|
| | |

$B \to C$

Lossless Decomposition

FD preserving

## 48. 3NF EXAMPLE 1

R(ABCDE)

FD = $\{AB \to C, B \to D, D \to E\}$

CK = (AB)

Now, partial dependencies are $B \to D$

$B^+ = \{B, D, E\}$  R(A B C $\cancel{D}$ $\cancel{E}$)

| B | D | E |
|---|---|---|
| | | |

$B \to D$

$D \to E$

| A | B | C |
|---|---|---|
| | | |

$AB \to C$

2NF

Here $D \to E$ is ... in 3NF.

$D \to E$ is ...

⇒ Now $D^+ = \{$...

R (B ...

| B | |
|---|---|

$B \to D$

## 49. 3NF EXAM...

R(ABC)

FD: $\{AB \to C, C \to$ ...

V.V.V.V. Imp.

partial depend...

Here $C \to A$ ...

∴ $\boxed{C \to A}$ is ...

Ex

R(...

CK...

A → ...

A → ...

A → ...

B → ...

Here $D \to E$ is the Transitive dependency. ∴ The table (BDE) is not in 3NF.

$D \to E$ is the Transitive Dependency present in table BDE

⇒ Now $D^+ = \{D, E\}$

R (B D $\cancel{E}$)

| B | D |
|---|---|
| | |

$B \to D$

| D | E |
|---|---|
| | |

$D \to E$

| A | B | C |
|---|---|---|
| | | |

$AB \to C$

3NF

## 49. 3NF EXAMPLE 2

R(ABC)

FD: $\{AB \to C, C \to A\}$

Now, Candidate Key = (B) = ($\acute{A}B\acute{C}$)

Now, $B^+ = \{B\}$

$(AB)^+ = \{ABC\}$ ✓

$(BC)^+ = \{BCA\}$ ✓

∴ Candidate Keys = (AB) (BC)

∴ All are prime attributes

v.v.v.v. Imp.

partial dependency = part of key → Non-prime attribute

⇒ prime attribute → Non-prime attribute

Here $C \to A$ is not partial dependency because $C \to$ prime attribute (A)

∴ $\boxed{C \to A}$ is not partial dependency.

Ex:

R(ABCDEF)

CK: ABC

$A \to D$ = partial dependency.

$\left.\begin{array}{l} A \to B \\ A \to C \\ B \to C \end{array}\right\}$ = Not partial dependencies

In the above table there are no partial dependencies ∴ The Relation is in 2NF

(50)

Transitive dependency

$$\boxed{\begin{array}{c} \text{Non prime attribute} \rightarrow \text{Nonprime attributes} \\ \Downarrow \\ \text{subset of Non-prime attributes} \rightarrow \text{Subset of Non prime attributes} \end{array}}$$

$$\left.\begin{array}{l} R(ABCDEF) \\ CK = ABC \end{array}\right\} \Rightarrow \left.\begin{array}{l} D \rightarrow E \\ E \rightarrow F \end{array}\right\} \text{Transitive Dependencies.}$$

$$\left.\begin{array}{l} B \rightarrow C \\ D \rightarrow C \end{array}\right\} \text{Not Transitive dependency}$$

∴ The above Relation doesnot contain Transitive dependency

⇒ ∴ The Relation is in 3NF

## 50. FORMAL DEFINITION OF 3NF

A Relational schema 'R' is in 3NF only if every Non-trivial FD $X \rightarrow Y$ either 1> X is a superkey
or
2> Y is a prime attribute

Which of the following is allowed in 3NF?

a) proper subset of CK → Non prime (partial dependency)

b) Non-prime → Non-prime (Transitive dependency)

c) proper subset of CK + Nonprime → Nonprime (Transitive dependency)

d) proper subset of CK → proper subset of other CK (Not a PD, TD) ✓

51. 3NF E

R(ABCDEF)

Now, candi

Now, (A

Now,

Now, $A^+$=

$A \rightarrow BC$

$C \rightarrow D$

⇓

Transitive Dependen

Now, C

52. BCNF

A Relational

functional d

of 'R'. i.e d

superkeys.

R(ABC) FD

⇒ Candida

## 51. 3NF EXAMPLE 3

$R(ABCDEF)$ $F: \{A \to FC, C \to D, B \to E\}$

Now, candidate Key = $(AB)^+ = (ABCDEF)$

Now, $(AB)^+ = (ABFCDE)$ ∴ $(AB)$ is the candidate key

Now, $\left.\begin{array}{l} A \to FC \\ B \to E \end{array}\right\}$ are the partial dependencies.

Now, $A^+ = \{A, F, C, D\}$ $B^+ = \{B, E\}$

∴

| A | F | C | D |
|---|---|---|---|
| | | | |

| B | E |
|---|---|
| | |

| A | B |
|---|---|
| | |

∴ The Relation is in 2NF

$A \to FC$

$C \to D$ ⇓ Transitive Dependency.

$B \to E$

Now, $C^+ = \{C, D\}$

| A | C | F |
|---|---|---|
| | | |

$A \to FC$

| C | D |
|---|---|
| | |

$C \to D$

| B | E |
|---|---|
| | |

$B \to E$

| A | B |
|---|---|
| | |

3NF

## 52. BCNF INTRODUCTION

A Relational schema 'R' is in BCNF if whenever a Non-trivial functional dependency $X \to A$ holds in R, then 'X' is a superkey of 'R'. i.e determinants of all functional dependencies should be superkeys.

$R(ABC)$ FD $\{A \to B, B \to C, C \to A\}$

| A | B |
|---|---|
| | |

| B | C |
|---|---|
| | |

| C | A |
|---|---|
| | |

⇒ candidate Keys = $\{A, B, C\}$

BCNF

## 53. BCNF Example 1



$R(ABC)$  $F: \{AB \to C, C \to B\}$

$\Rightarrow$ The candidate key = $(A)^+ = (ABC)$

$A^+ = \{A\}$

$\therefore (AB)^+ = \{ABC\}$  $\therefore$ (AB) (AC) are candidate keys $\Rightarrow$ A, B, C are prime attributes

$(AC)^+ = \{ACB\}$

$\therefore$ There are no partial dependencies in the table

$\therefore$ The table is in 3NF because the candidate key contains all the attributes (All the attributes in the Relation are prime attributes)

Now, Acc to the BCNF the LHS should be a superkey

$\Rightarrow (AB)^+ = \{ABC\}$

$C^+ = \{C, B\}$ Not superkey.

Now,

ABC → AC, CB

| A | C |
|---|---|

| C | B |
|---|---|

$C \to B$

BCNF

$\Rightarrow$ Lossless Decomposition

$\Rightarrow$ NOT FD preserving ($AB \to C$ is lost)

## 54. BCNF Example 2

R (ABCDEFGHIJ)

$F: (AB \to C, A \to DE, B \to F, F \to GH, D \to IJ)$

candidate key = $(AB)^+ = (ABCDEFGHIJ)$

Now, $(AB)^+ = (ABCFGHDIJE) \Rightarrow$ A, B are only the prime attributes

Now, the partial dependencies are $\left\{ \begin{array}{l} A \to DE \\ B \to F \end{array} \right\}$

$A^+ = (ADEIJ)$ $B^+ = (BFGH)$

Now the Relation R(ABCDEFGHIJ)

| $R_1$(ADEIJ) | $R_2$(BFGH) | $R_3$(ABC) ⇒ No PD's ⇒ 2NF |
|---|---|---|
| A→DE | B→F | AB→C |
| D→IJ | F→GH | ↓ |
| TD | TD | (ABC) |
| | | AB→C |

$D^+$=(DIJ)
$F^+$=(FGH)

| (ADE) | (DIJ) | (BF) | (FGH) | (ABC) in 3NF |
|---|---|---|---|---|
| A→DE | D→IJ | B→F | F→GH | AB→C |

⇒ To check BCNF definition every LHS of the dependency applicable on a table should be a SuperKey.

∴ The Relations that we get are

| $R_1$(ADE) | $R_2$(DIJ) | $R_3$(BF) | $R_4$(FGH) | $R_5$(ABC) |
|---|---|---|---|---|
| A→DE | D→IJ | B→F | F→GH | AB→C |
| $A^+$={ADE} | $D^+$={DIJ} | $B^+$={FB} | $F^+$={FGH} | $(AB)^+$={ABC} |
| A=SK | D=SK | B=SK | F=SK | AB=SK |

BCNF

## 55. BCNF Example 3

R(ABCDEFGHIJ) FD: { AB→C, B→D, D→EF, A→GH, H→IJ)

Candidate key = ( AB $)^+$ = (ABCDEFGHIJ)

Now, $(AB)^+$ = (ABCDEFGHIJ) (AB) is the candidate key

⇒ A, B are prime attributes. ⇒ $A^+$ = {AGHIJ}
$B^+$ = {BDEF}

⇒ Now B→D, A→GH is the partial dependency ⇒ $B^+$ = {B,D,E,F}
B→D is possible

Now, (ABCDEFGHIJ)

| $R_1$(BDEF) | $R_2$(AGHIJ) | $R_3$(ABC) | Both lossless and FD preserving |
|---|---|---|---|
| B→D | A→GH | AB→C | |
| D→EF | H→IJ TD. | | |

Transitive Dependency

Now,

R(ABCDEF GHIJ)

| AGHIJ | BDEF | ABC |
|---|---|---|
| A→GH | B→D | AB→C |
| H→IJ TD | D→EF TD | |
| | | AB→C |

Now,
$H^+$ {HIJ}

| (HIJ) | (AGHI) | DEF | BD |
|---|---|---|---|
| H→IJ | A→GH | D→EF | B→D |

∴ The tables are
1) HIJ
2) AGHI
3) DEF
4) BD
5) ABC
} are in BCNF

## 56. BCNF EXAMPLE 4

R(ABCD) FD: {AB→C, BC→D}

candidate Key: $(AB)^+ = (ABCD)$

$(AB)^+ = \{(ABCD)\}$ (AB) is the candidate Key

Now, There are no partial dependencies

Now, BC→D is the Transitive dependency, BC is not a superkey

ABCD

| BCD | ABC |
|---|---|
| BC→D | AB→C |

} 3NF and BCNF

## 57. BCNF EXAMPLE 5

R(ABCD) FD: {A→B, C→D} CK: {AC} A, C } are prime attributes

partial dependency: A→B, C→D } Now, $A^+ = \{A,B\}$

$C^+ = \{C,D\}$

ABCD

| AB | ACD |
|---|---|
| A→B | AC→D |
| 3NF | C→D |

CD | AC

⇒ If a table ... be in BC...

58. BCNF EXA...

R=(ABDLPT)

Candidate K...

Now, parti...

59. BCNF

R(ABCDE)

Now, To ch... dependencie...

A→BC
CD→E
B→D
E→A

⇒ If a table contains only 2 attributes then the Relation will definitely be in BCNF.



## 58. BCNF Example - 6

R:(ABDLPT)  FD{B→PT, T→L, A→D}

Candidate Key = $(AB)^+ = (ABDLPT) \Rightarrow (AB)^+ = (ABDPTL)$//

$\boxed{CK = AB}$

Now, partial Dependencies = B→PT, A→D } $B^+ = (B, P, T, L)$, $A^+ = (A, D)$

R(ABDLPT)

$R_1$(AD) — A→D — SK — BCNF

$R_2$(BPTL) — B→PT — SK; T→L — Not SK

TL — T→L — SK — BCNF

BPT — B→PT — SK — BCNF

$R_3$(AB) — BCNF

2NF

The tables are [AD, TL, BPT, AB]

## 59. BCNF Example 7

R(ABCDE)  FD{A→BC, CD→E, B→D, E→A}  Candidate Key = {A, E, CD, BC}

All the attributes are prime

⇒ 3NF

Now, To check whether the Relation is in BCNF check the functional dependencies and find whether the LHS is a super key or not

A→BC ⇒ $A^+ = (ABCDE)$ ⇒ A is SK

CD→E ⇒ $(CD)^+ = (CDEAB)$ ⇒ CD is SK

B→D ⇒ $(B)^+ = (B, D)$ ⇒ Not SK

E→A ⇒ $(E)^+ = (E, A, B, C, D)$ = E is SK

ABCDE

BD — B→D

ABCE — A→BC, E→A

ABCDE

BD — ABCE

$B \to D$ | $A \to BCE$, $E \to ABC$, $BC \to AE$ } BCNF

## 61. BCNF EXAMPLE 8 - PART 1 AND 62 - PART 2

R(ABCD)

FD{$AB \to CD$, $D \to A$}

⇒ candidate key= {(AB)} {(DB)} ∴ (AB) (DB) are candidate keys.

⇒ partial dependencies are absent. Therefore the relation is in 2NF

⇒ Transitive dependencies are not there and 'D' is not superkey but A is prime attr.

⇒ For BCNF the LHS should always be a super key, and here 'D' is not super key in $D \to A$

$D^+ = \{D, A\}$

Lossless Decomposition

ABCD → DA, BCD

DA: $\boxed{D \to A}$ ✓ BCNF (2 attributes)

BCD: $B^+ = (B)$, $C^+ = (C)$, $D^+ = (D, A)$

$(BC)^+ = (BC)$

$(BD)^+ = (BDAC) \Rightarrow BD \to C$ ✓

$(CD)^+ = (CDA)$

Now, find if $AB \to CD$ is preserved using $D \to A$ and $BD \to C$

⇒ $(AB)^+ = (AB)$ ∴ $AB \to CD$ is not preserved.

∴ Dependency preserving failed.

## 63. BCNF EXAMPLE 9

R(ABCD) $\{A \to B, B \to A, B \to C, C \to D\}$

The candidate key = $(\ \ )^+ = (ABCD)$

$\therefore$ Now, $A^+ = \{ABCD\}$ ✓
$B^+ = \{B, A, C, D\}$ ✓ } candidate keys

$C^+ = \{C, D\}$ $\quad (CD)^+ = (CD)$ ×

$D^+ = \{D\}$

Now, A, B are prime attributes ⇒ Now, partial dependencies are not presen (only single attribute CK/s).

Now for 3NF check the LHS, LHS must be a super key or the RHS should be a prime attribute in $C \to D$ ⇒ violating 3NF

Not SK $\quad$ Not prime attribute

Now, $C^+ = \{C, D\}$

R(ABCD)

$R_1(CD)$ $\quad$ $R_2(ABC)$

$\boxed{C \to D}$ $\quad$ SK $\underline{A} \to B$, SK $\underline{B} \to A$, SK $\underline{B} \to C$ } 3NF and BCNF

3NF and BCNF (2 attribute so)

lossless and FD preserving

## 64. BCNF - EXAMPLE 10 PART 1 AND 65. PART-2

R(ABCDE) FD: $\{AB \to C, C \to D, D \to E, E \to A\}$

candidate key = $(\ B\ )^+ = (ABCDE)$

$(B)^+ = \{B\}$ $\quad$ Now $(AB)^+ = \{A, B, C, D, E\}$
$(CB)^+ = \{B, C, D, E, A\}$
$(DB)^+ = \{B, D, E, A, C\}$
$(EB)^+ = \{B, E, A, C, D\}$

} All the attributes are prime attributes ⇒ 'R' is in 3NF

e keys.
n is in 2NF
uperkey but
A is prime attrs
y, and here

BD → C ✓

nd BD → C

For BCNF, every LHS of the functional Dependency should be a superkey

$AB \to C \Rightarrow AB(SK)$

$C \to D$
$D \to E$
$E \to A$ } C, D, E are not SK's

Now, $C^+ = (CDEA)$

ABCDE

(C DEA) | (BC)

Not SK — D → E, E → A; C → D

(BC) → 2-attributes, BCNF, $B \to C$

Now, $D^+ =$ DEA

(DEA) | (CD)

$D \to E$ | $C \to D$ (BCNF)

Not SK ← $E \to A$

Now, $E^+ = \{EA\}$

(EA) | (ED)

$E \to A$ BCNF | $D \to E$ BCNF

↓ (A BC) → Adding 'A' to 'BC'

$AB \to C$

$A^+ = (A)$

$B^+ = (B)$

But ($AB \to C$) is not preserved, so add 'A' to 'BC' and preserve the dependency.

| |
|---|
| THE 2ND NF AND 3NF ARE LOSELESS AND FD PRESERVING |
| BUT BCNF IS LOSLESS AND FD MAY OR MAY NOT BE PRESERVED. |

## 66. GATE QUESTION ON NORMALISATION 1

GATE-94

State True or False with Reason. There is always a decomposition into BCNF that is losless and Dependency preserving.

Ans: FALSE (Refer above Note) (we cannot guarentee dependency preserving)

GATE-98

which Normal form is considered adequate for Normal Relational Database design?

a) 2NF  b) 5NF  c) 4NF  ~~d)~~ 3NF (Actual ans is BCNF)

GATE-99

Let R= (ABCDEF) be a relation scheme with the following dependency $C \to F$, $E \to A$, $EC \to D$, $A \to B$ what is the key of R?

$R(ABCDEF) \Rightarrow (EC)^+ = (ABCDEF) \Rightarrow (EC)^+ = \{E, C, A, B, F, D\}$

$\therefore$ EC is the key.

GATE-01

consider the schema R(ABCD) and functional dependencies $A \to B$ and $C \to D$. Then the decomposition of 'R' into $R_1(AB)$ and $R_2(CD)$ is

a) FD preserving and lossless

b) lossless but FD preserving fails

~~c)~~ FD preserving but not lossless join

d) Not FD preserving and not lossless join

R(ABCD)

$R_1(AB)$  $R_2(CD)$ $\Rightarrow$ No common attribute

$\therefore$ lossy decomposition

| Now, $R_1(AB)$ | $R_2(CD)$ |
|---|---|
| $A^+ = (A, B)$ | $C^+ = (C, D)$. |
| $B^+ = (B)$ | $D^+ = (D)$ |
| $F_1 : (A \to B)$ | $F_2 : (C \to D)$ |

$\boxed{F_1 \cup F_2 = F}$ $\therefore$ Dependency preserving.

GATE 02

Relation R with an associated set of FD's (F) is decomposed into BCNF. The Redundancy (arising out of functional dependency) in the resulting set of Relations is

a) ✓ Zero   BCNF = (0% Redundancy)

b) More than Zero but less than of 3NF decomposition

c) proportional to the size of $F^+$

d) Indeterminate

## 67. GATE QUESTION ON NORMALISATION 2

GATE-05

which one of the following statements about Normal Forms is False?

(a) BCNF is stricter than 3NF (Left hand side should be a SK in BCNF) (TRUE)

(b) lossless, FD preserving into 3NF is always possible (TRUE)

(c) ✓ lossless, " " " BCNF " " " (WE CANNOT GUARENTEE)

(d) Any Relation with 2 attributes is in BCNF. (TRUE) LHS = Key (SK)
RHS =

GATE-IT-05

A table has fields $F_1\ F_2\ F_3\ F_4\ F_5$ with the following functional dependencies $F_1 \to F_3$, $F_2 \to F_4$, $F_1F_2 \to F_5$ what is the NF of the Relation?

a) ✓ 1NF   b) 2NF   c) 3NF   d) None

Now, candidate key: $(F_1F_2)^+ = (F_1\ F_2\ F_3\ F_4\ F_5)$

$(F_1F_2)^+ = (F_1F_2F_3F_4F_5)$ ∴ $F_1F_2$ is candidate key.

Now, ~~No~~ partial dependencies are present ∴ 2NF ✗ $\begin{pmatrix} F_1 \to F_3 \\ F_2 \to F_4 \end{pmatrix}$

Now, LHS should be super key for 3NF ⇒ $\left.\begin{matrix} F_1 \\ F_2 \\ F_1F_2 \end{matrix}\right\}$ are SK's = 3NF

GATE-12

which of th

a) Every r

b) A Relation

functiona

c) ✓ Every r

d) NO Relat

68. GATE

GATE-14

A prime a

a) In all

b) ✓ In son

c) In a

d) Only in

GATE-95

Consider

∴ Now,

GATE-97

R(a,b,c,d)

FD: {a→c

Now

GATE-12

Which of the following is True?

a) Every relation in 3NF is also in BCNF (Not always)

b) A Relation 'R' is in 3NF if every non-prime attribute of 'R' is fully functionally dependent on every Key of R → some key

c) ✓ Every relation in BCNF is in 3NF

d) NO Relation can be in both BCNF and 3NF (FALSE)

## 68. GATE QUESTION ON NORMALISATION 3

GATE-14

A prime attribute of a relation scheme R is an attribute that appears

a) In all candidate Keys of R

b) ✓ In some candidate Key of R

c) In a foreign Key of R

d) Only in the primary Key of R.

GATE-95

Consider $R(ABC)$ FD: $\{AB \to C, C \to A\}$ show that R is in 3NF but not BCNF

∴ Now, candidate Key = $(B)^+ = (AB\acute{C})$

$(AB)^+ = (ABC)$
$(CB)^+ = (ABC)$ } All are prime attributes ∴ the relation is in 3NF

∴ $AB \to C$ SK
$C \to A$ Not SK } Not in BCNF

GATE-97

$R(a,b,c,d)$, a,b,c,d contains atomic values (No composite and multivalue)

FD: $\{a \to c \quad b \to d\}$

Now, candidate Key = $(ab)^+ = (abcd)$

⇒ ab is the CK

⇒ $\left.\begin{matrix} a \to c \\ b \to d \end{matrix}\right\}$ are partial Dependency.

a) ✓ 1NF but not in 2NF

b) 2NF but not in 3NF

c) IN 3NF

d) None of the above.