INTRODUCTION

DATABASE :- Collection of interrelated data.

DBMS :- S/W used to manage the database and access the DB efficiently. (File system + Operating System)

efficiency decrease as size of DB increase

Limitations of File System:

(1)

(University) (Database)

F1
F2
F3
⋮
FN

(500GB)

?? Retrieve data from DB i.e students who scored more than 80%.

⇓

Alternatives

- Program
- Manually

• an appln programmer should have knowledge about

Physical details (low-level details)
- Fname
- Location of file
- Format
- Permissions

→

» Accessing data using physical level details too complex.

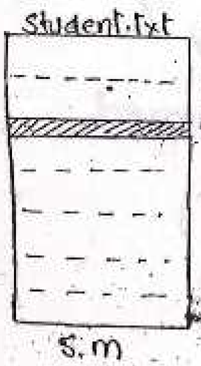
⇒ Database MS (DBMS) provide "Data Independency" i.e user can use the data without knowing any physical details.

· It is achieved by using querying language.

Select * from Student where marks > 80%

(2)

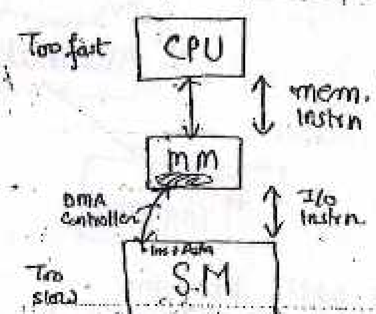
500 GB | Student.txt (10 GB) | (Disk) ?? (student whose roll no (21 to 25)) (Just a small access in 10 GB DB.)



whole 10 GB need to be t/f in mm in worst case .... too much overhead as compared to only 4 records required.

(1 Human Architecture)
Stored Programs

Ins + Data that is required to execute by CPU should be stored in High speed MM

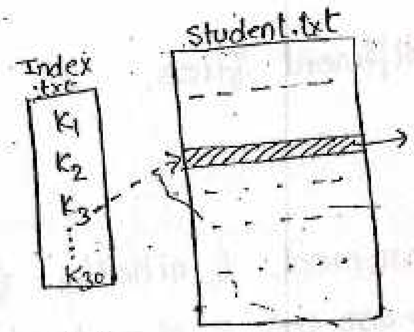


» If DB file is too huge, it take more IO cost to retrieve the data.

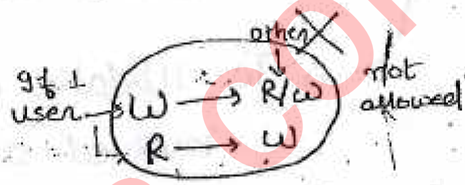
⇒ DBMS solution to it :- "Index to database"

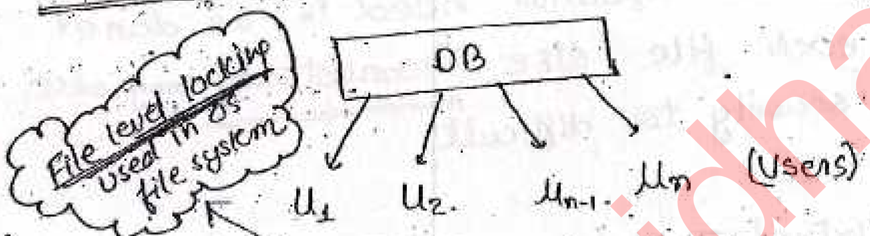
access a page in 1000 page book

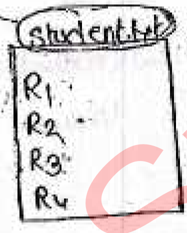
{ Eg 1000 pg book .... worst case 1000 pg turn.
has 30-pg-index file, then to retrieve only (30+1) = 31 access required.



③ Concurrency control :







no conflict even then
{ U_1: Update R_1
U_4: Update R_4 }

while R_1 accessed by U_1, it acquire lock on student.txt thereby preventing U_4 from accessing the file. Thus (Less concurrency level)

DBMS Solution

{ $U_1: R_1^{*L}$
$U_4: R_4^{*L}$ }

» Allow record level locking. i.e does not allow multiple users to update same record but provide access (read) to different records. (High Concurrency level)

④ Security :

student.txt

Sid	Sname	DOB	Marks	Ph no	Address

→ outsider (Sname, Sid)
→ Admin (Sid, Sname, address)
→ Faculty (Sid, Sname, marks)

File System Security

PASSWORD Security only

(multi-level security)

→ This type of security not there in OS. To forcefully provide that ... file. (REDUNDANCY)

» Problems if we maintain different files.

→ Redundancy

→ Too much space consumed (initially file=500GB) now it will be » 500 GB ∵ of redundancy.

→ To update any column, too much overhead involved as updation need to be done in each file else inconsistency may arise.

» Different levels of security too difficult.

DBMS Solution

(VIEW) (Virtual Table)

V(F1), V(F2), V(F3) } Views of stud/faculty, stud, admin, std. outsider

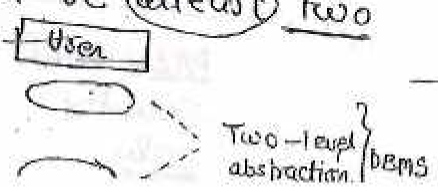
view (virtual table) of files

→ Only one physical table is there from which different virtual table (view) are created.

DATA INDEPENDENCY : [DATA MODEL] → given by CODD

Hiding the low level (physical) details from the external user.

Codd [Relational DBMS] : To provide data independency there should be atleast two levels of abstraction.



⇒ 3-levels of Abstraction : [DBMS Architecture]

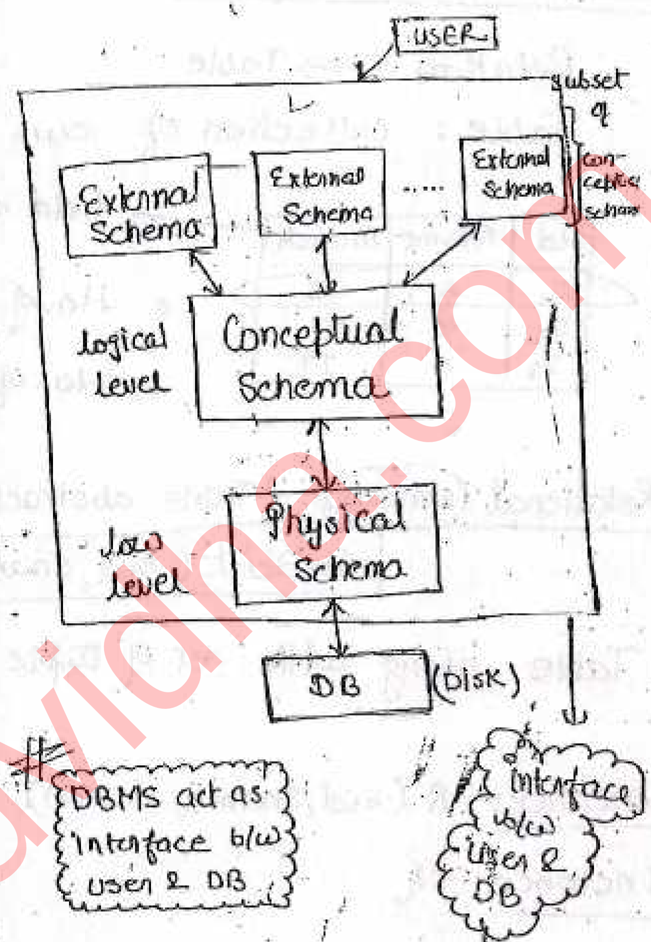
Physical Schema :-

- (metadata)
  - → Location
  - → Name
  - → Format
  - → [student] ⟷ std.fm

"define how data physically stored in database."

⇒ CREATE TABLE STUDENT
   (....
   );

⇒ If user wish to store a file of its own format, it can't be achieved because of the format not defined in physical schema.



Conceptual Schema :- Don't know how and where the data is stored, but knows

Hiding physical details.

"WHAT DATA IS STORED"

Know Student (sid, sname, marks, Ph no, add)
there is a student table with so & so fields.

External schema : it is the subset of conceptual schema.

RELATIONAL DBMS

Relation ⟶ Table

Table : collection of rows and column

Sid	Sname	Branch
S_1	A	CS
S_2	B	IT
S_3	A	IT

Branch → field or attribute

record or tuple

- No. of attributes : Arity : set of attributes R
- No. of tuples : Cardinality : set of tuples r

o» Relational schema : (Table abstraction)

$\boxed{\text{Student (sid, sname, Branch)}}$

Table along with set of tuple : Relational Instance

Schema : R (sid, sname, Branch)

Instance : r(

Codd rules for Relational DBMS :

① No two tuples of the relation should be same or

- There should be set of attributes in relation R such that all tuples of R can be differentiated.

Key :- minimum no. of attributes used to differentiate all the tuples of the relation (non null tuples) *

mini. no. of attributes used to differentiate ALL the NON NULL tuples of a relation.

Sid = Key

(Sid, sname) : not Key ∵ ~~it is~~ not minimum

→ PPno — Key — but — PPno — Null — A, Null — D

Enrolled (Sid, Courseid, fee)

S_1	C_1	...
S_1	C_2	..
S_2	C_2	..

(Sid, Cid) : Key

Employee (Eid, Ename, PPno, DOB, Fname)

Let Constraint : "No two employees same DOB & Fname"

Eid	Ename	PPno	DOB	Fname
			10/10	x
			10/10	y
			11/12	y

Keys : { Eid, PPno, {DOB, Fname}}

» Simple Key : (Keys with single attribute)

{Eid}, {PPno}

» Compound Key : (Keys with two or more attributes)

{DOB, Fname}

Prime attribute set : set of attributes that belong to any key
or (Key attribute set)

{Eid, PPno, DOB, Fname}

Non Prime / Non-Key Attributes : set of attributes not belonging to any key.

{Ename}

Candidate Keys : set of keys in a Relation = Key

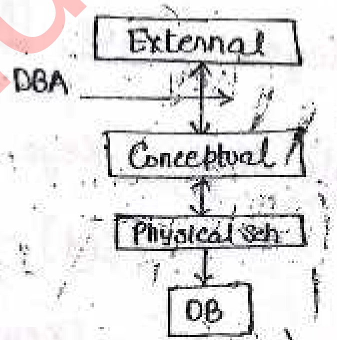
at least one CK should be non-null

[Eid, PPno, {DOB, Fname}] → mini no of attribute used to distinguish each table

» **Database Administrator:** person who grants security to every user.

1. Design the tables
2. Design key constraints
3. Monitors database to improve performance
4. Continuously take the backup of DB to avoid failures.
5. Can do indexing i.e choosing an attribute as index.

» DBA is a person existing b/w ~~physical~~ or operate b/w the conceptual and external schema. or external user.



PRIMARY Key :- one of the candidate key.
can't be null.

» Candidate keys : {Eid, PPno, (DOB, Fname)}
» Primary key (let) : {Eid} → at most only one primary key per table allowed.

ALTERNATIVE / SECONDARY KEYS : All candidate keys except primary keys.

→ If {Eid} = Primary Key

→ {PPno, (DOB, Fname)} = Alternative key

PRIMARY KEY	ALTERNATE CANDIDATE KEY
① At most one PK per table	① More than one AK possible
② Not null	② can be null
③ PRIMARY KEY	③ UNIQUE ← key word

NULL → Unknown (DOB)
→ Unexisted (PPno)
→ No two null values are equal

→ All NULL are non-zero

→ DBMS assigns random ASCII values to every NULL such that no two NULL's are equal.

CREATE TABLE Employee (Eid varchar(30) PRIMARY Key, — PK
Ename varchar (30) NOT NULL,
PPno Varchar (20) UNIQUE, — AK
DOB date, — data type
Fname varchar(30),
Unique (DOB, Fname)); — AK

» SUPER KEY :- set of attributes used to differentiate tuples of the relation.

- Eno
- Eno, Ename
- PPno
- PPno, Eno

} Super Key

- (Ename, Fname) : X not a super key

Candidate Key = minimal super key #

Q:- Let R be the Relational Schema with n attributes, $A_1, A_2, A_3, \ldots A_n$.

- How many super Keys are possible?
  - (i) with only candidate Key A_1
  - (ii) " " " Keys A_1, A_2
  - (iii) " " " Keys A_1A_2, A_3A_4
  - (iv) " " " " A_1A_2, A_2A_3
  - (v) " " " " A_1, A_2, A_3

(I) Super keys = ? =

Candidate key = A_1

n elements = no. of subsets
$= 2^n$

Any combination with A_1 is a super key

A_1
$A_1 A_2$
$A_1, A_2 A_3$
$A_1, A_2, A_3 \ldots$
⋮
$A_1, \underbrace{A_2, A_3, \ldots A_n}_{n-1}$

$\boxed{2^{n-1}}$ super keys

(ii)

A_1	A_2
$A_1 A_2$	$A_2 A_1$
$A_1 A_2 A_3$	$A_2 A_3$
	$A_2 A_3 A_1$
⋮	⋮
$A_1 A_2, A_3, \ldots A_n$	$A_2 A_3 A_1 A_4 \ldots A_n$
2^{n-1}	2^{n-1}

$= 2 \cdot 2^{n-1}$ —— BUT

there are some keys counted twice.

A_1	A_2
$A_1 A_2$	~~$A_2 A_1$~~
$A_1 A_2 A_3$	~~$A_2 A_1 A_3$~~
$A_1 A_2 A_3 A_4$	~~$A_2 A_1 A_3 A_4$~~
A	$A_2 A_3 A_4$
⋮	
$A_1 A_2 A_3 \ldots A_n$	~~$A_2 A_1 \ldots A_n$~~

All such keys are repeated.

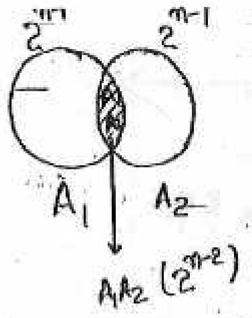
i.e any combination of $A_1 A_2$ is repetative.

$\underline{A_1 A_2}\ \underbrace{A_3 \ldots A_n}$
⇓
2^{n-2} —— Duplicated

Thus need to be subtracted.

$\boxed{2 \cdot 2^{n-1} - 2^{n-2}}$ #

$\Leftrightarrow$



$\Rightarrow \boxed{2^{n-1} + 2^{n-1} - 2^{n-2}}$

(iii) Cand key

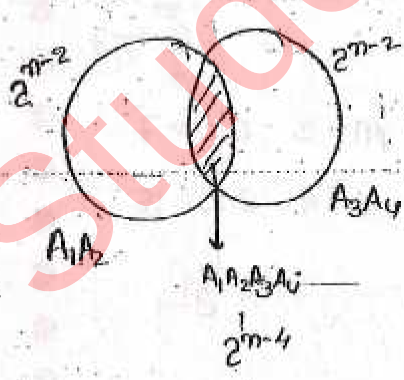
$\underline{A_1A_2}$	$\underline{A_3A_4}$
$A_1A_2\ A_3$	$A_3A_4A_1$
$A_1A_2\ A_4A_3$	$A_3A_4A_1A_2$
⋮	⋮
$A_1A_2\ A_3 \dots A_n$	$A_3A_4\ A_1 \dots A_n$
2^{n-2}	2^{n-2}

$A_1A_2A_3A_4\ \underbrace{A_5 \dots A_n}$ } Repeting keys

2^{n-4}

Thus total no. of keys

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

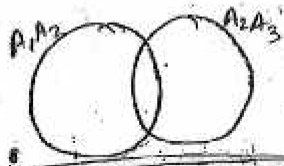


(iv) A_1A_2 A_2A_3

$A_1A_2A_3\ \underbrace{A_4 \dots A_n}_{n-3}$

$\Downarrow$

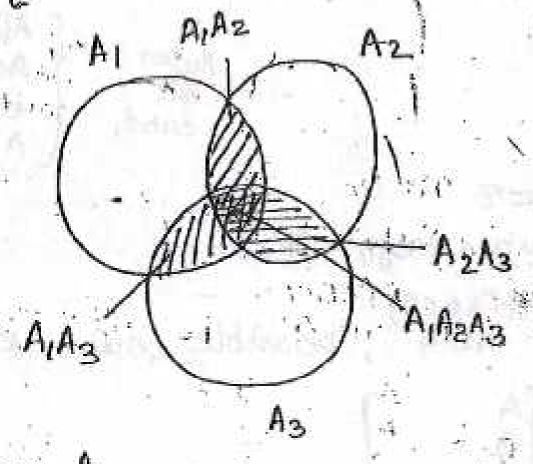
2^{n-3}



$$\boxed{2^{n-2} + 2^{n-2} - 2^{n-3}}$$

(v)

A_1	A_2	A_3
A_1A_2	A_1A_2	A_3A_1
$A_1A_2 \dots A_n$	$A_2A_1 \dots A_n$	$A_3A_1 - A_n$
2^{n-1}	2^{n-1}	2^{n-1}



$$= \underset{A_1}{2^{n-1}} + \underset{A_2}{2^{n-1}} + \underset{A_3}{2^{n-1}}$$

$$- [\underset{A_1A_2}{2^{n-2}} + \underset{A_2A_3}{2^{n-2}} + \underset{A_3A_1}{2^{n-2}}] + \underset{A_1A_2A_3}{2^{n-3}}$$

Q GATE R(ABCDE)

{A, BC} — candidate ke

Super key = ??

A	BC	$\underbrace{ABC}$ DE
AB	BCA	
	BCAD	
2^{5-1}	2^{5-2}	2^{5-3}

Common keys

$$= \boxed{2^{5-1} + 2^{5-2} - 2^{5-3}}$$

Q DRDO

R is the Relational schema with n attributes. How many super keys possible?

⇒ Sol^n No specification about candidate key

Let R(A,B,C) — Possible super keys

Cand + super { A, B, C }

Super not cand. { AB, AC, BC, ABC }

7 super keys

n=3

Super keys = 7

R(ABCD)

Let n=4, possible super keys =

C.K + S.K { A, B, C, D }

S.K but not C.K. { AB, AC, AD, BC, BD, CD, ABC, ABD, ACD, BCD, ABCD }

15 super keys

⇓

Possible only when each key is candidate key

$2^3-1=7$

Thus n=3 SK = 7

n=4 S.K = 15

2^4-1

n=n — $\boxed{2^n-1}$

PARENT RELATION

CHILD RELATION

Foreign Key :- avoid inconsistency due to insertion / deletion.

Student

Sid	Sname	login
S_1	A	@
S_2	A	@
S_3	B	@
S_4	C	@

[Sid = PRIMARY KEY]

{All student of the organization}

Enrolled

Sid	cid	fee
S_1	C_1	---
S_1	C_2	---
S_2	C_2	---
S_3	C1	---
S_8	C2	---

insertions: ok — S_3; X — S_8

Allowed

Restricted

$S_8 \notin R$

[(Sid, cid) : PRIMARY KEY]

{student who are enrolled for some course}

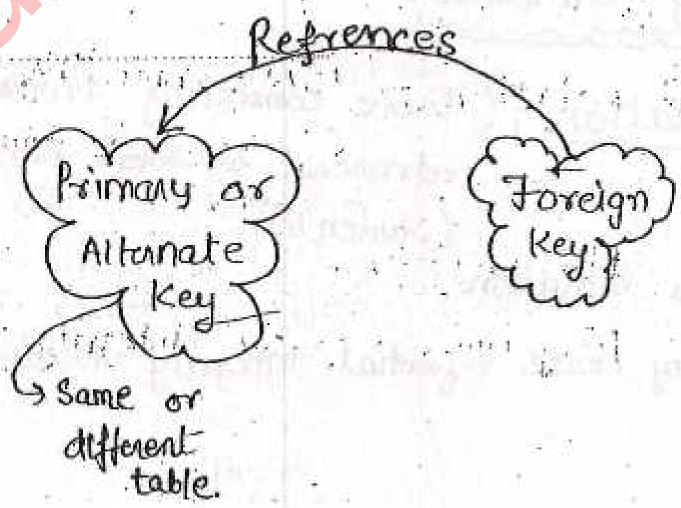
Deletion

If S_4 deleted,
no problem — allowed.

If S_1 is deleted
then enrolled table — Restricted
become inconsistent
Hence S_1 deletion
not allowed.

Sid of Enrolled table is the foreign key referencing to the primary key of student table.

» **Foreign Key** is a set of attributes, references to the primary key or alternative key of the same table or some other table.



26th Aug 2011

FOREIGN KEY

F.Key — Referencing relation

referenced relation — Student

Sid	Sname	login
S1	A	-
S2	A	-
S3	B	-
S4	C	-

referenced attribute

Sid = Primary key

Enrolled

Sid	cid	fee
S1	c1	-
S1	c2	-
S2	c1	-

(Sid, cid = PK)

Foreign key (sid) attribute are Refrencing Attribute

```
CREATE TABLE Enrolled
  Sid varchar (10),
  Cid varchar (10),
  fee integer (20),
  Primary key (sid, cid),
  FOREIGN KEY (Sid) Refrences student);
```

When Sid = PK of student

By default F.K refrences to Primary key

If Sid not primary (let login PK) then to refer to sid in student table use:

FOREIGN KEY (Sid) Refrences Student (Sid)

When sid not the PK of student

Sometimes Foreign key refrences to the Primary key of same table.

PK, FK

Eid	Ename	Sup_id
E1		E2
E2		Null
E3		E2
E4		~~E6~~ E4
E5		

⇒ Supervisor is one of the employee only.

Refrential Integrity Constraints :-

(a) Refrenced Relation : (Table consisting Primary key refrenced by some foreign key) (Student)

(i) Insertion : No violation

(ii) Deletion : May cause refrential integrity violation.

Constraints : (what to do in case of violation of Ref. intrity)

ⓐ ON DELETE NO ACTION (default) :

⇒ Deletion is prohibited if it cause Refrential Integrity violation.

⇒ FOREIGN KEY (sid) Refrences student (sid) ON DELETE NO ACTION;

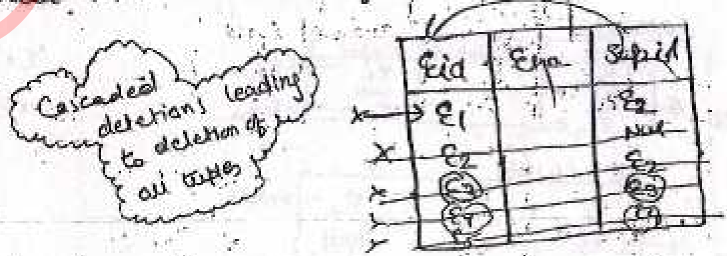
ⓑ ON DELETE CASCADE :

⇒ Deletion happen in both refrencing and refrenced relation if refrential integrity violation occur.

⇒ FOREIGN KEY (sid) Refrences Student (sid) ON DELETE CASCADE;

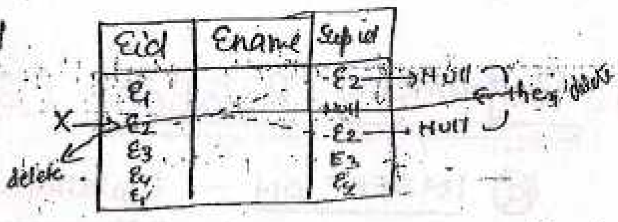
⇒ Whenever Primary Key deleted, corresponding tuple in refrencing relation also deleted.

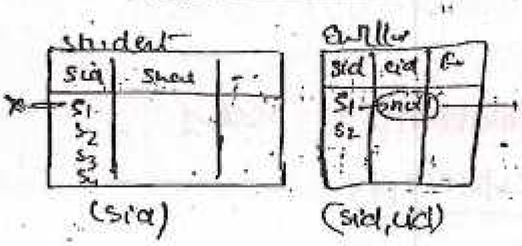
⇒ Not suggestable for one table PK & FK system :: deletion of one item may result in deletion of full table



ⓒ ON DELETE SET NULL :

If violation occur then foreign key field values tried to set NULL, if success then deletion allowed in the refrenced relation.





tries to set it to null in Enrolled table, but that not possible ∵ Sid [Enrolled] is a part of primary key. So deletion prohibited.

* If Foreign Key = NOT NULL attribute then –

$\boxed{\text{ON DELETE NO ACTION = ON DELETE SET NULL}}$

* FOREIGN KEY can be $\boxed{\text{NULL}}$, But if it is the primary key of refrencing relation then NULL not allowed.

ex

Eid	Ename	Sup
E1		Null
E2		Null
E3		Null
E4		E2

→ Foreign Key

(iii) UPDATION : (may cause integrity violation)

* ON UPDATE NO ACTION — Updation Restricted
* ON UPDATE CASCADE — Updation in both Relation
* ON UPDATE SET NULL — refrencing attribute tried to set NULL, if success, updation of Refrenced relation allowed.

cascade

Eid	Ename	Super
E1		E2 → E20
E2 ← E20		Null
E3		E2 → E20

SET NULL

E1		E2 → Null
E2 ← E20		null
E3		E2 → Null

(II) Refrencing Relation : (Enrolled)

(a) INSERTION — violation may occur, if violation occur then insertion restricted.

(b) DELETION — No violation

(c) Updation (Ref attribute) — may cause violation, if occur updation restricted.

Parent Relation modification can lead to modification in child relation, but vice versa not possible.

Q:

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

A = Primary key
C = Foreign key refrences A with ON DELETE CASCADE.
– What are othe tuples deleted to preserve ref. integrity constraint when [2,4] tuple deleted?

⇒ [5,2] [7,2] and [9,5] Ans

ENTITY – RELATIONSHIP DIAGRAM: {High Level Database Design}

Entity :- An object that can be differentiated from other objects.

Entity set :- Collection of similar entities ▭

Attribute ⬭

Key Attribute ⬭

Multivalued Attribute

⇒ one or more value for any attribute of a given record.

Sid	Sname	Cname	Pno
A1	x	c/c++	99/98

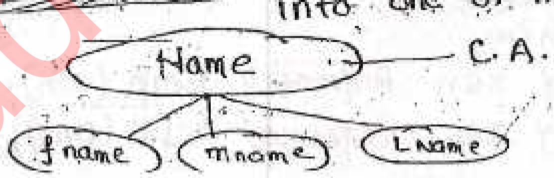
denoted by: ◎

Derived Attribute: value of the attribute that can be derived from other attributes. {Not stored in DB. Age calculated on the fly}

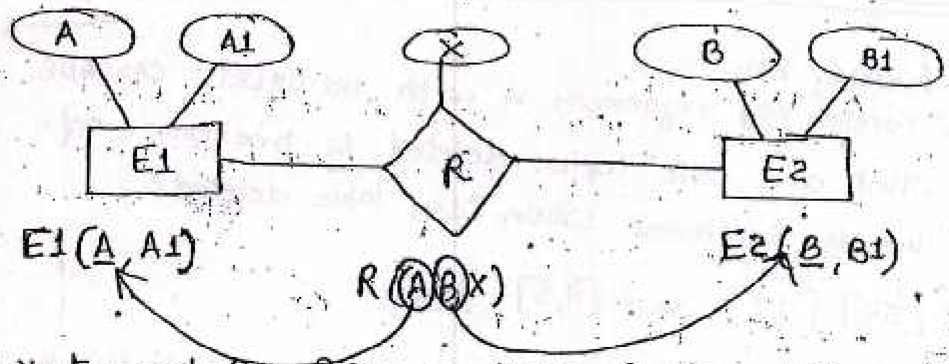
DOB	AGE

AGE = Current Date — DOB

Compound Attributes: Attribute that can be further subdivided into one or more other attributes.



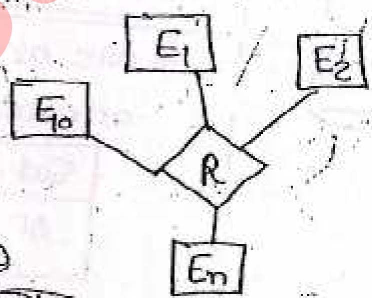
⇒ **Relationship Set** :- used to relate two or more entity sets.

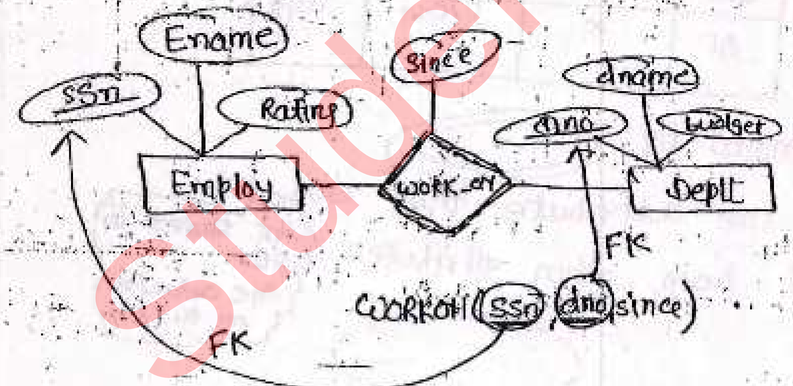


⇒ **Attribute set R**: Primary keys of the entities that are relating to the relationship set and descriptive attribute of R (x)

∘ The related attributes of E1 and E2 are kept in relation set R while independent attributes of E1 or E2 do not contribute to R.

N-ary Relationship Set :





WORK_ON(ssn, dno, since)

Schema for WORKSON

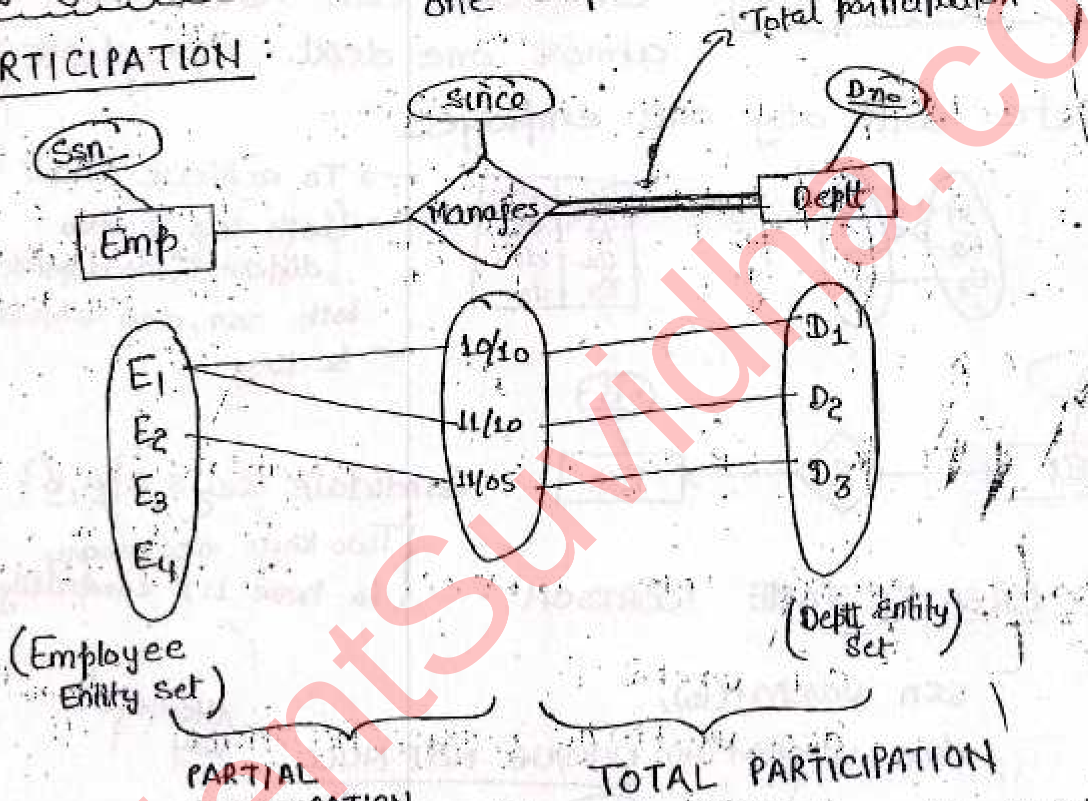
\# CREATE TABLE WORKSON (
Ssn varchar(20),
Dno varchar(20),
Since integer(10),
FOREIGN KEY SSN References Emp (ssn),
FOREIGN KEY Dno References Deptt (Dno)); ON DELETE CASCADE

⟹ In N-ary Relationship set, minimum n foreign keys are necessary corresponding to mini. one key per entity set.

RELATIONSHIP SET :

Integrity Constraints :- Employee can manage multiple deptt. Each deptt. managed by only one Emp.

(i) PARTIPATION :



PARTIAL PARTICIPATION — when some of the entities do not participate in relationship set.

TOTAL PARTICIPATION ↳ Every entity of entity set relating with the Relationship set.

Denoted by:- { ═══ }

(ii) CARDINALITY : How entity of an Entity Set is related with entities of other Entity Sets.

Eg in above diagram there exist one to many cardinality.

∵ an employee can manage more than one department.

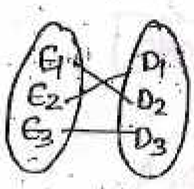
{In ER diagram
→ : one cardinality
—|— : many cardinality}

Types of Cardinality:-

(i) One to one (1:1)
(ii) One to Many (1:m)
(iii) many to one (m:1)
(iv) many to Many (m:m)

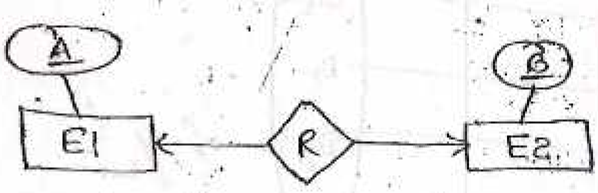
⇒ Primary key may dye based on cardinality, Foreign Key remain same.

One to One Cardinality :- One emp can relate with atmost one deptt. One deptt co-relate with only one employee.



Ssn	Dno
E1	d2
E2	d1
E3	d3

→ To achieve that (both ssn & dno different, no duplicate) both ssn, dno should be key.

⇒ 

Candidate Keys = {A, B}
{Two keys necessary to have 1:1 cardinality}

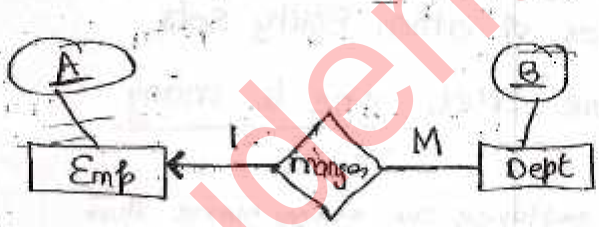
Schema: CREATE TABLE WORKSON
(
ssn varchar(10),
dno varchar(10) UNIQUE NOT NULL ← Alternate Key
PRIMARY KEY (ssn)
FOREIGN KEY ssn Refrences Employee
" Dno " Deptt);

One to Many cardinality :-

ssn	dno
E1	d1
E1	d2
E2	d3

can be duplicate — ssn
dno unique (P.K)

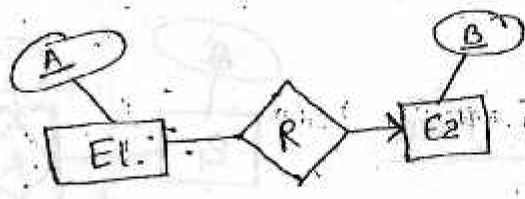
Candidate key
B



Schema :-
one to many side wali
PRIMARY KEY Dno

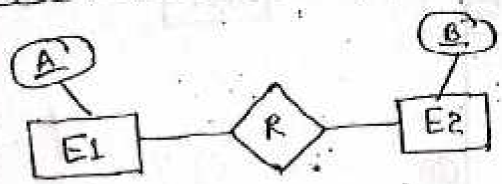
o)) Relation is opp as compared to ER-diagram. B/w emp i-e one Emp can associate with many deptt. But each deptt manged by

Many to One Cardinality:



Candidate key = A

Many to many cardinality (1:m) && (m:1)

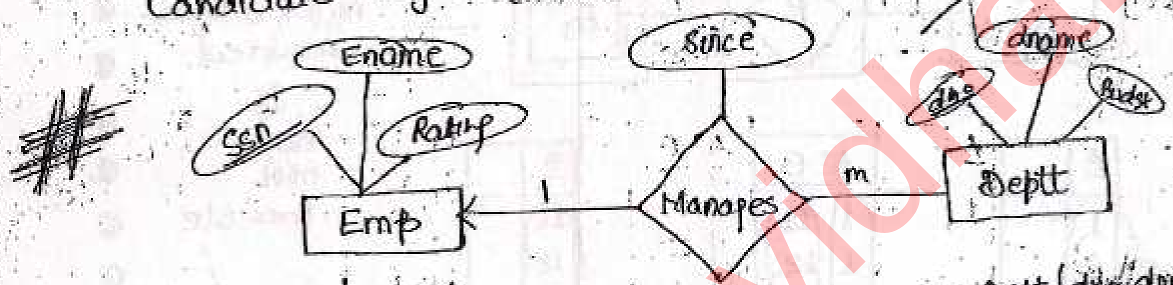


ssn	dno
E_1	d_1
E_1	d_2
E_3	d_2
E_2	d_3

Cardinality b/w E_1 & E_2 C=1 m & m
E_1 & E_2 m ≠ m opp.

Candidate key = (A B) – combined.

#



Emp (ssn, Ename, Rtg) Mang (ssn, dno, since) Dept (dno, dnm, budgt)

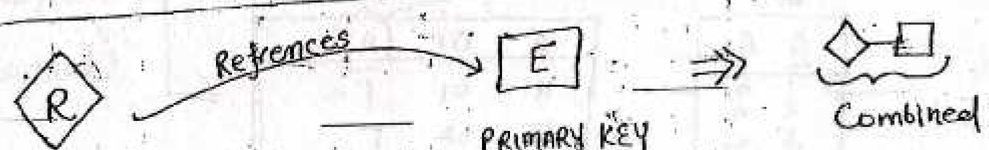
Primary key : dno
Foreign key : ssn
dno

PK
Mang. Dept (dno, dname, budgt, ssn, since)

ssn Foreign key Refrences Employee

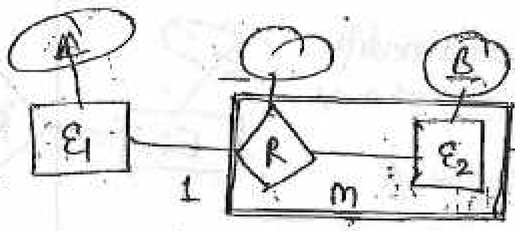
Minimizing Criteria

If FOREIGN KEY of Relationship Set R is also PRIMARY KEY of R and refrencing to the Primary key of Entity Set E, then R & E can be join into a single table.



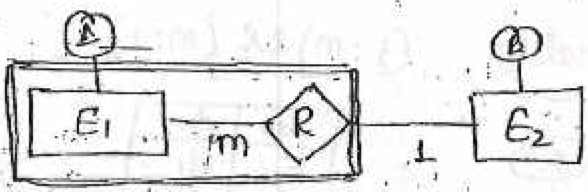
FOREIGN KEY (R) = PRIMARY KEY (R)   PRIMARY KEY [E]

⇒ Minimized Form



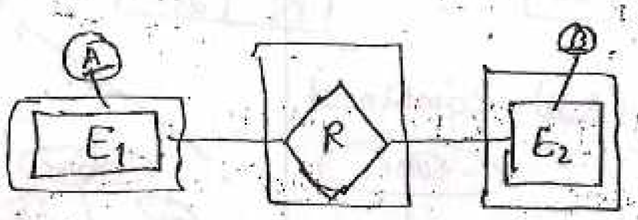
1:m — Combined into one table

Combined into one →



m:1

m:m



Cand. key AB

mini. 3 tables required.
⇓
no mini. possible

A
1
2
3

A	B
1	11
1	12
2	12

B
11
12
13

distinct

not distinct

not distinct

distinct

RE1 can't be combined

RE2 → can't be combined

⇒ Partial Participation in R

13 not in R hence partial Ppn

A	A1
1	a1
2	a1
3	a2

A	B
1	11
1	12

B	B1
11	b1
12	b2
13	b2

two tables merged into 1 (Partial Part)

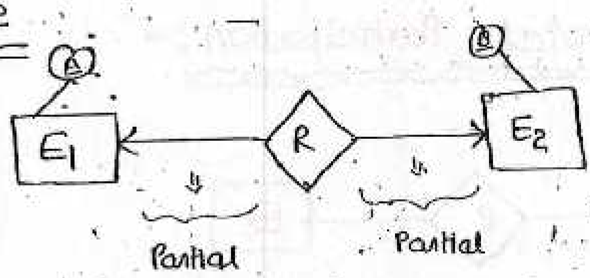
F.K.

A	A1
1	a1
2	a1
3	a2

B	B1	A
11	b1	1
12	b2	1
13	b2	NULL

One to one

(a)



mini. no. of tables required = ?

One to one cardinality.

Take some data satisfying all conditions

A	A_1
1	a_1
2	a_1
3	a_2

A	B
1	13
2	11

B	B_1
11	b_2
12	b_2
13	b_1

Partial

Partial

E1R

A	A_1	B
1	a_1	13
2	a_1	11
3	a_2	NULL

E1RE2

A	A_1	B	B_1
1	a_1	13	b_1
2	a_1	11	b_2
3	a_2	NULL	NULL
NULL	NULL	12	b_2

This table not possible. No PRIMARY KEY possible.

Primary Key = A

Alternate key = B.

↳ Minimum Two tables are possible in one to one cardinality with partial - partial participation.

If RE2 → $\boxed{R \Leftrightarrow E_2}$ ~~Total participation~~

AT LEAST ONE TOTAL PARTICIPATION

(b) At least one total Participation :-

A	A1
1	a_1
2	a_1
3	a_2
4	a_2

Parti

A	B
1	11
3	12
4	13

Total

B	B1
11	b_1
12	b_2
13	b_2

Partial participation side primary key is the final primary key.

E1RE2

A	A1	B	B1
1	a_1	11	b_1
2	a_1	null	null
3	a_2	12	b_2
4	a_2	13	b_2

» If atleast one total participation exists in 1:1 cardinality. Then mini. no. of table = 1.

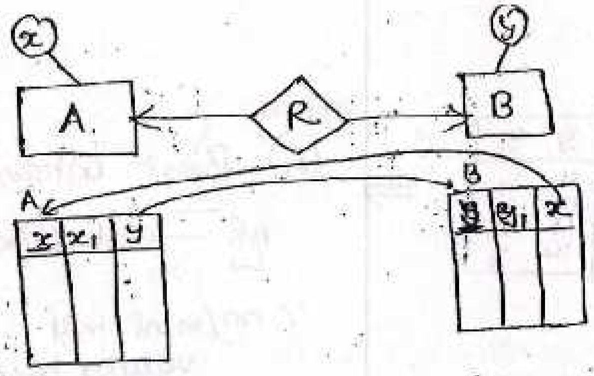
A: Primary key
B: Alternate key

Q: Relationship b/w two entity types A & B is 1:1 and the relationship is optional at the end A. Only 50% of B entities relates to A entity. Select the Best stmt for mapping.

(a) A & B should be kept separate with FK in the A relation
(b) " " " " " " both A&B "
(c) " " " " " " the B relation
(d) none of these.

Option Evaluation »

(b)



X

In 1:1 only one frgn key is sufficient.

→ Bad design (X)

(c)

optional (Partial)

x	x1
1	a1
2	a1
3	a2

x	y
1	13
2	11

y	y1
11	y1
12	y1
13	y2
14	y2

Let Partial

A

x	x1
1	a1
2	a1
3	a2

FK

y	y1	x
11	y1	2
12	y1	null
13	y2	1
14	y2	null

FK in B means: RB combined & referencing to some PK in A.

If AR → Partial then both (c) & (A) options are equivalent.

(A) Foreign key in A.

x	y1	y
1	a1	13
2	a1	11
3	a2	

y	y1
11	y1
12	y1
13	y2
14	y2

⇒ when total participation at AR. Then

A

x	x1
1	a1
2	a1
3	a2

R

x	y
1	13
2	11
3	12

B

y	y1
11	y1
12	y1
13	y2
14	y2

Total Partial

x	y_1	y
1	a_1	13
2	a_1	11
3	a_2	12

y	y_1
11	y_1
12	y_2
13	y_3
14	y_2

⇒ option

Ⓐ Best when

AR —— total partn

∵ no/mini. null values here as compared to option C.

⇒ Note

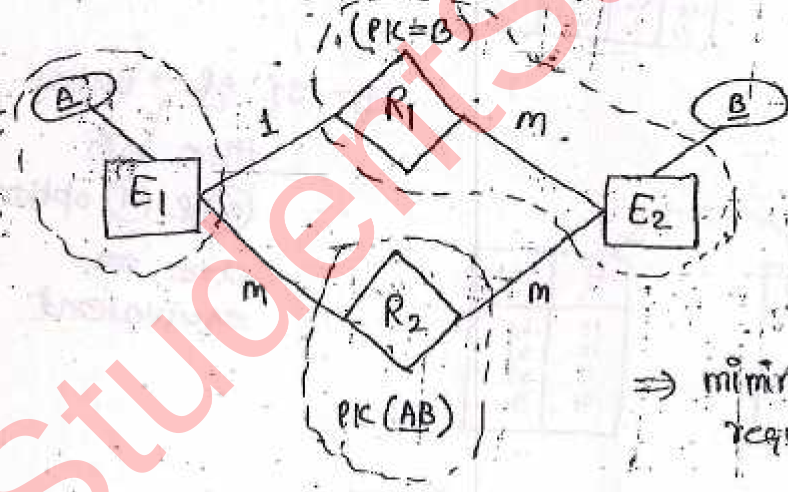
Q. E_1 & E_2 are entity sets and R1 and R2 are relationship set relating with E_1 and E_2 with cardinality 1:M and M:M.

How many minimum no. of tables required.

(a) 2 (b) 3 (c) 4 (d) 5

Sol^n :-

Let

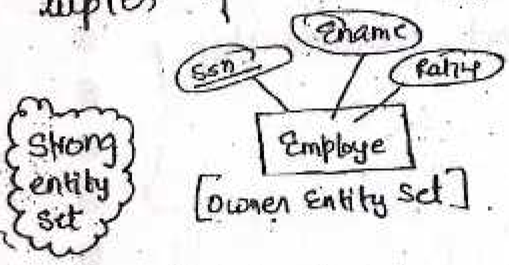


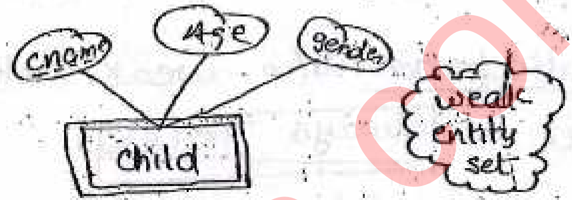
⇒ minimum 3 tables required

27th Aug 2011 :

WEAK Entity: Entity set with : no Primary key

Using one or more attributes of weak entity may not be possible to differentiate all the tuples of the entity set.



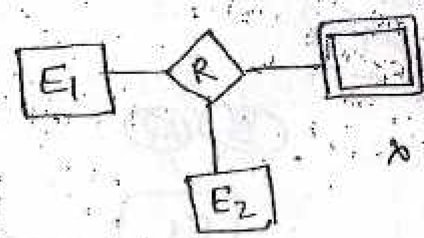


Cname	age	gender
A	10	M
A	10	M
B	8	F
B	8	F

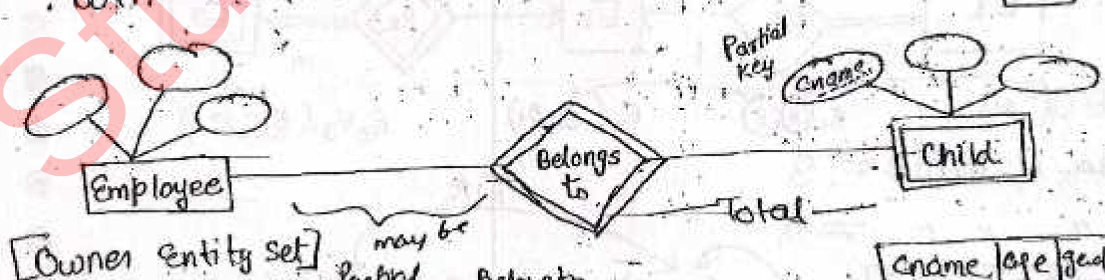
» For every weak entity set, there should be a unique Owner Entity set.

» A weak entity has only one owner entity set.

» Weak entity related to another weak entity set is not possible.



Both not possible



Ssn	
E_1	
E_2	
E_3	
E_4	

Belongs to

ssn	cname
E_1	A
E_2	A
E_3	B
E_4	B

cname	age	gndr
A	10	M
A	10	M
B	8	F
B	8	F

Total Participation

» For every weak entity there should be an identified owner entity.

» Relationship set not differentiating uniquely is called as weak Relationship set.

» Participation b/w weak entity set and relationship set is $\boxed{\text{always TOTAL}}$.

» Cardinality ⇒ 1:M

» Always weak entity set and weak Relationship set should be kept in single DB table.

Foreign key (not the primary key of relation)

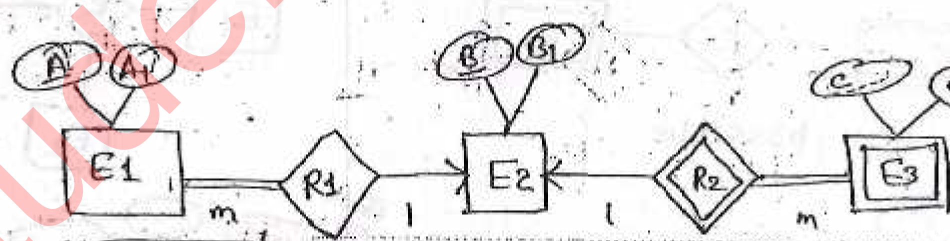
← reference employee(ssn)

Ssn	Cname	Age	Gender
E1	A	10	m
E2	A	11	m
E1	A	10	m
E2	A	10	m

✗ → (row 3) ; → ok (row 4)

Same weak entity set A, (o/m) should not be associated with same owner entity (E1) if E1 replaced by E2 then ok.

Q



E($\underline{A}$, A1) R1 (A)(B) E2($\underline{B}$, B1) R2E3 ($\underline{B}\,\underline{C}$ C1)

mini. no. of table = 3.

Maxi. " " " = 4.

= FK is the key of the relation

$\boxed{E_1R_1 (\underline{A}, A_1, B)}$ E2($\underline{B}$, B1) R2E3 ($\underline{BC}$ C1)

combined

FK

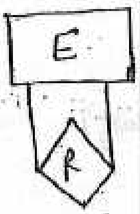
FK

Pure entity set = no foreign key

Foreign key = 2
mini. table = 3
max. table = 4

Self Refrential Relationship set:-

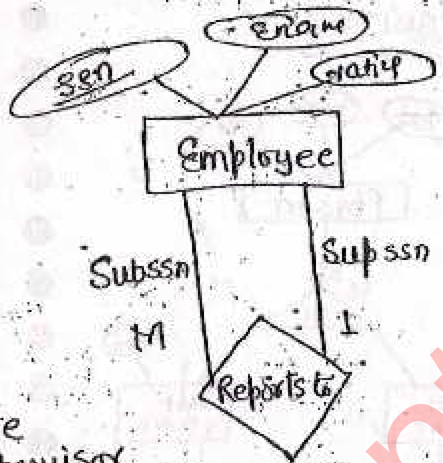
⇒ Relationship set relating to the same entity set.



⇒ Every supervisor and subordinate are employee.

⇒ Supervisor can supervise one or more subordinates

⇒ Subordinates reports to atmost one supervisor.



One supervisor can supervise many subordinates.

Employee(ssn, Ename, Rating)

Emp_reports(ssn PK, Ename, Rating, Supssn F.K.)

Reports to(Subssn, supssn)

⇒ Foreign key subssn refrences to Emp

⇒ " " supssn refrences to Emp

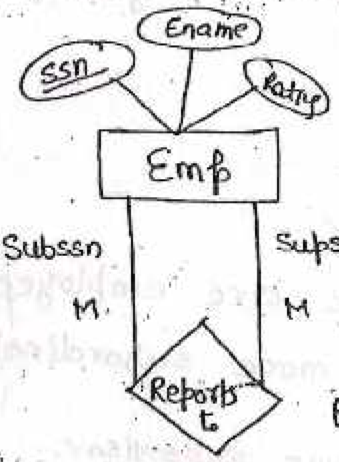
PRIMARY KEY : Subssn

⇒ Minimization Criteria :— Foreign key refrences to ssn which is the (primary key). So two tables can be combined.

Mini. 1 table required

RECURSIVE / Self Refrential Entity set

What ?? if Cardinality is m:m =

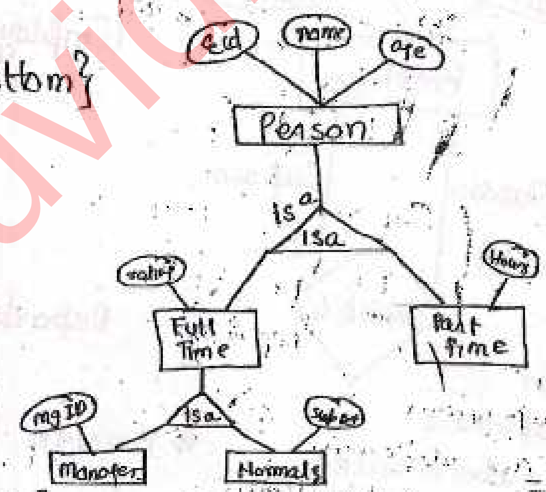


Emp(ssn, Ename, Rate)

Reports to (Subssn, supssn)

Here no FK is a complete P.K., Thus not possible to reduce. Thus mini. two table required.

~~Generalization~~ Generalization : {Top to Bottom}
Specialization



~~Specialization~~ Specialization : {Bottom to top}
Generalization