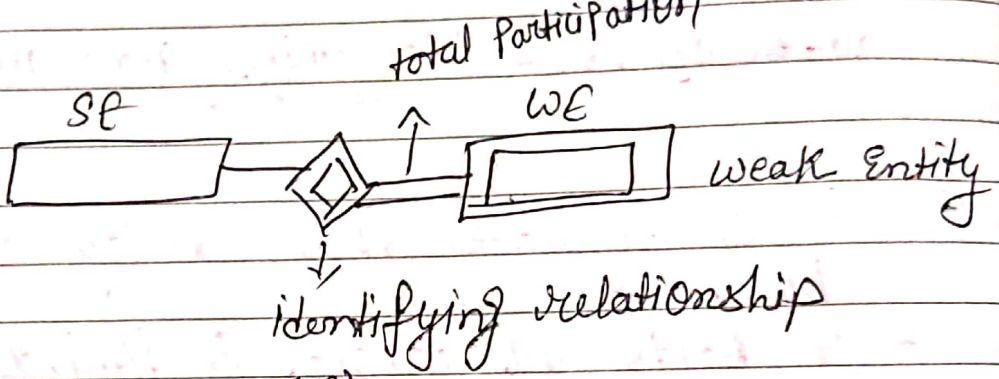
Page No.
Date

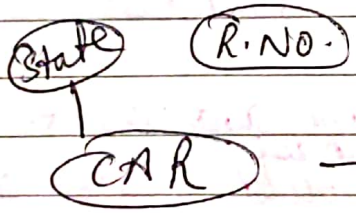
Lecture-6 24/8/2020

① Weak Entity & strong entity set

② How to represent in ER diagram



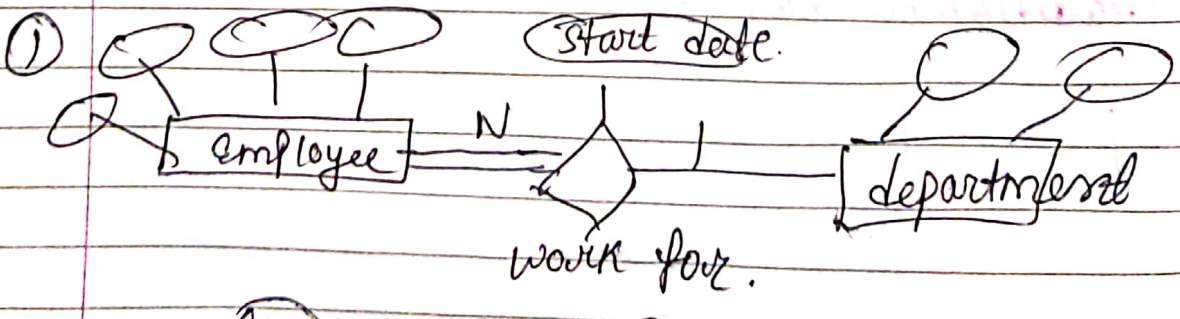
without (SE) it (WE) can't exist.

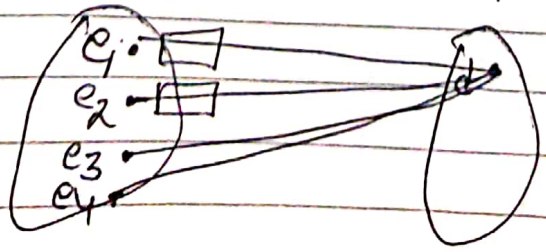


— Key attribute,

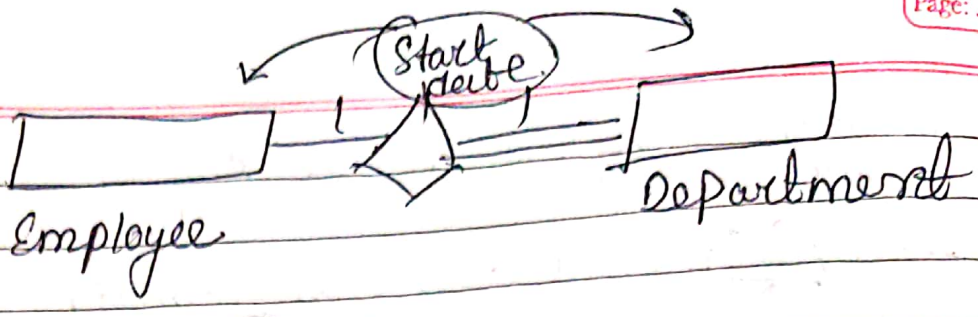
attributs to relationship

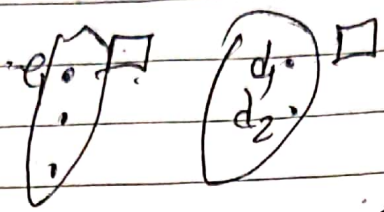
①





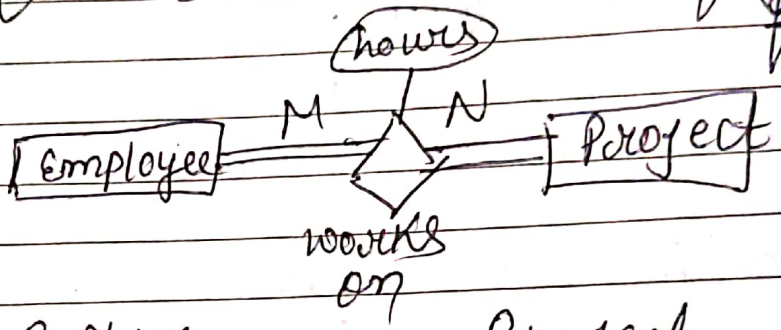
(2)

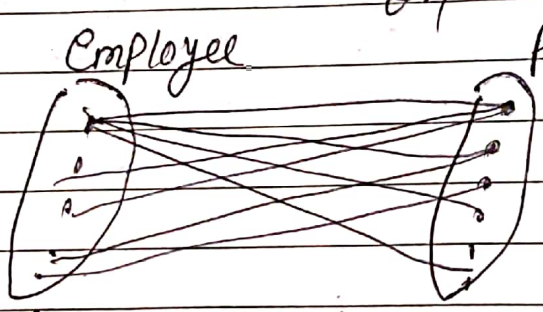




(shifting is not feasible).

(3)



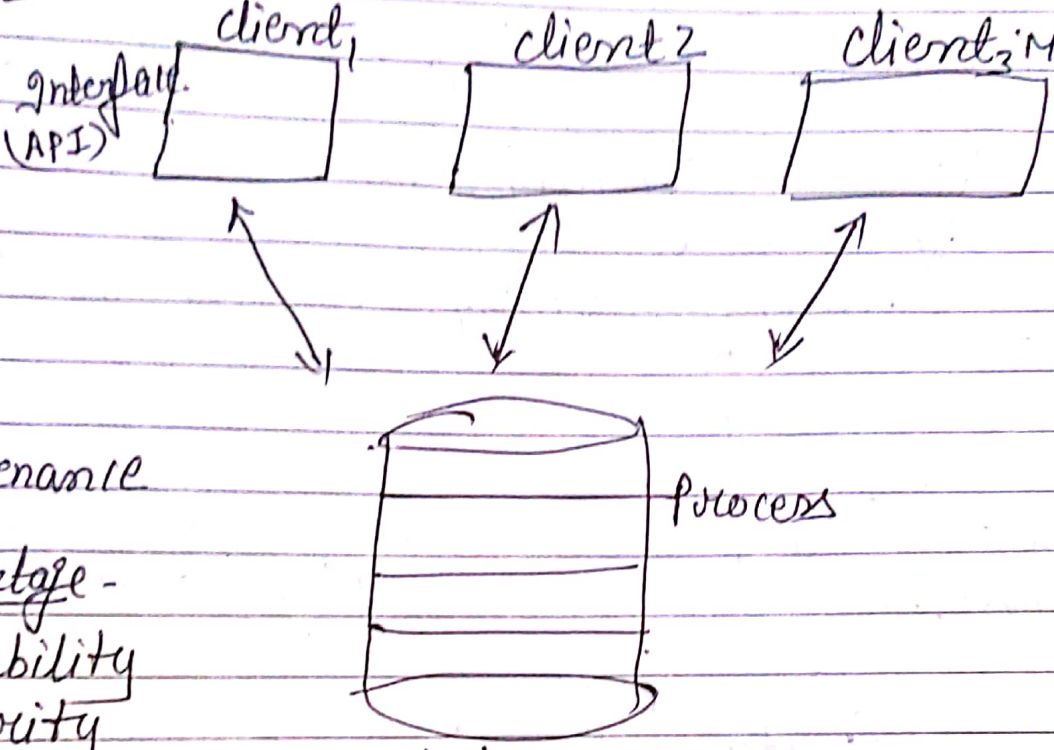


- for every ~~every~~ employee, for every project, how many hours did this employee works on that Project.
- It is not possible to put hours on employee or project).

2 tier architecture – client/server architecture

↓

2 layers.



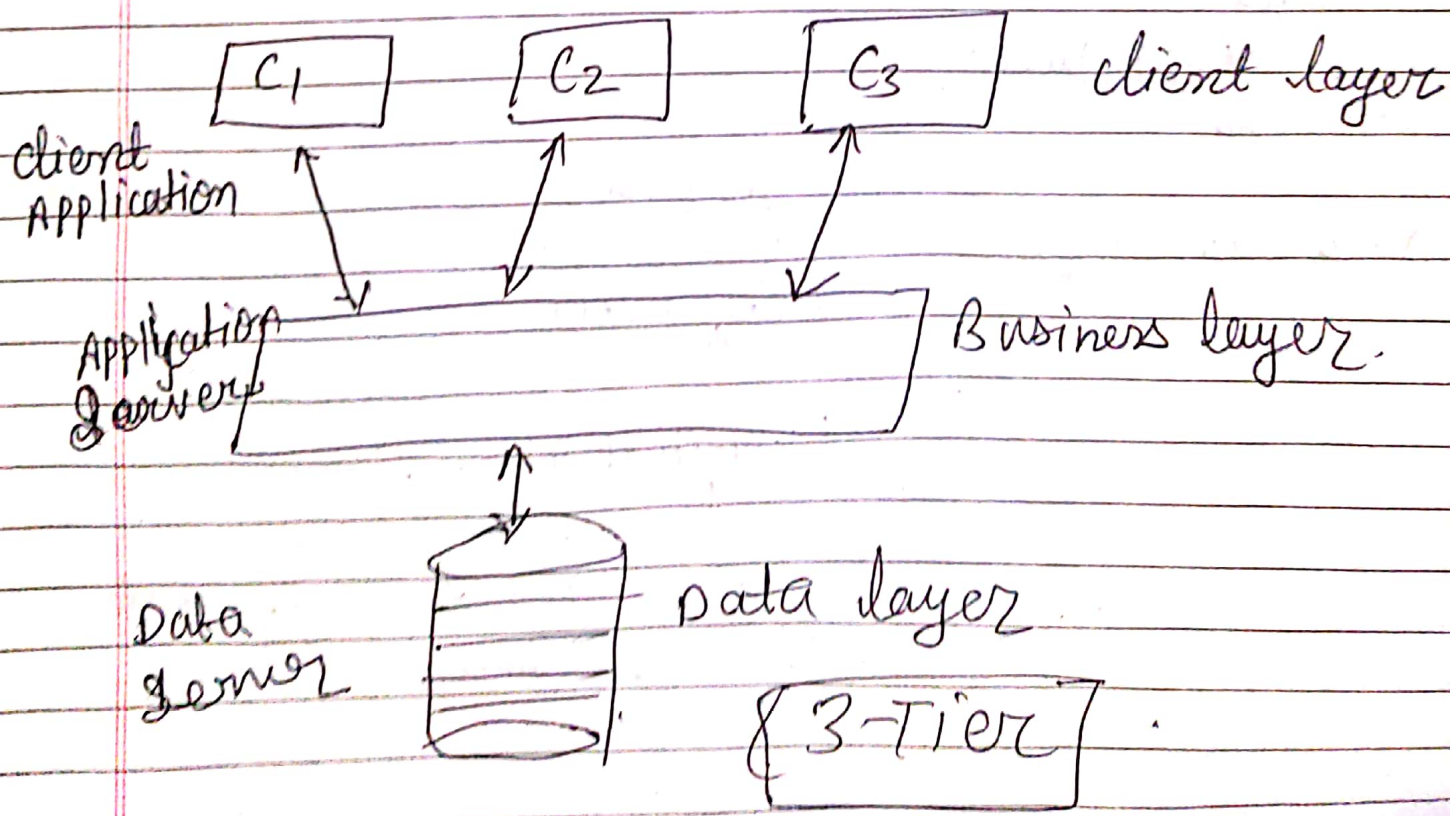
Advt –

① Maintenance

Disadvantage –

① Scalability

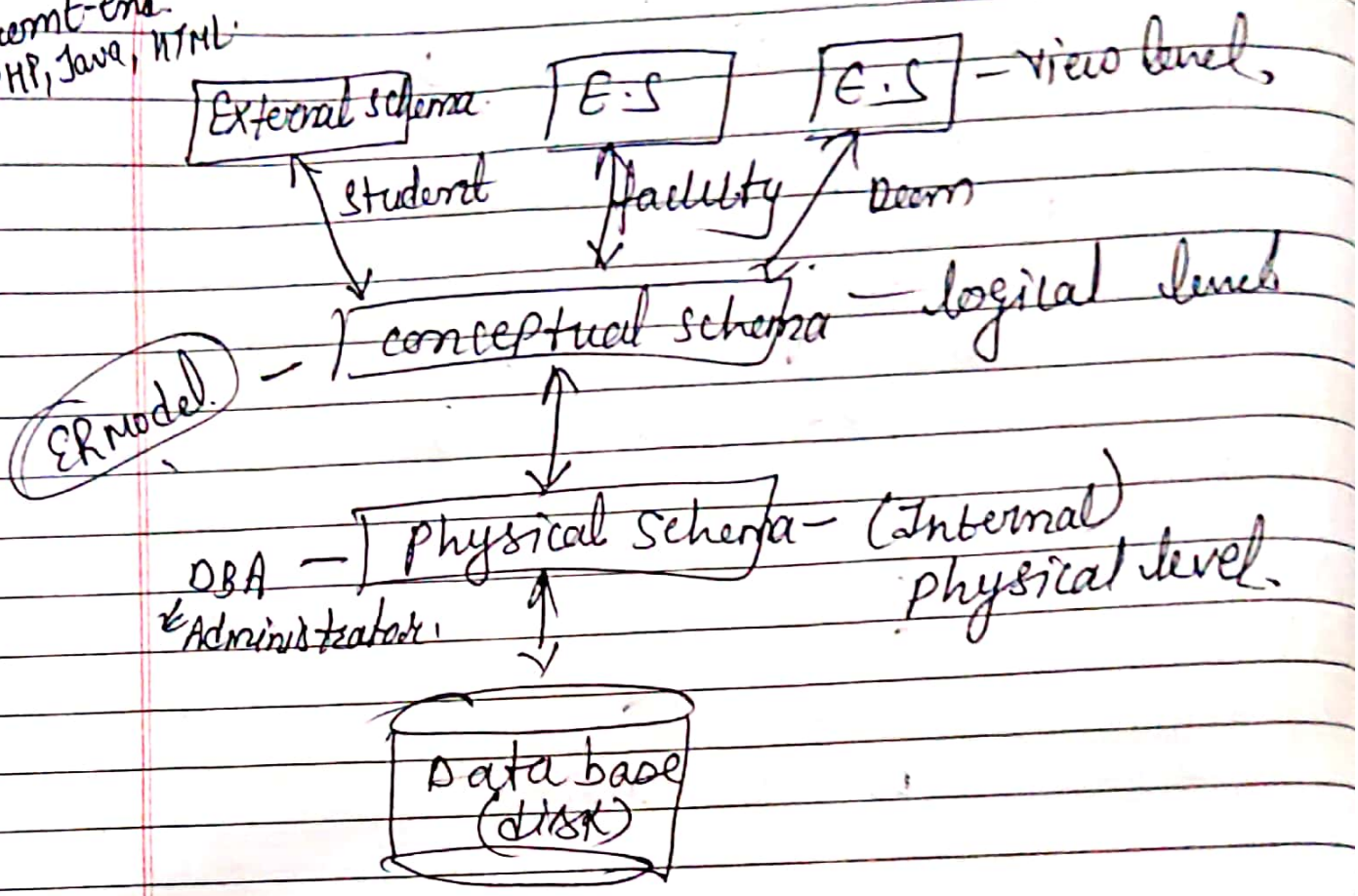
② Security



Data abstraction

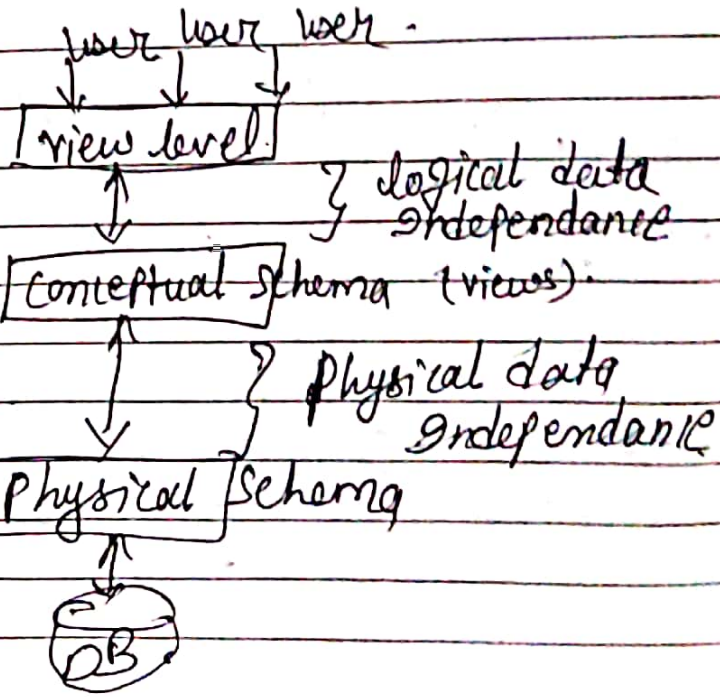
3-Tier Schema architecture:

front-end
PHP, Java, HTML



Data i
- Centralized.
- Distributed.

Data Independence.



Candidate Key & Primary Key

* Key. -
* use of Key -
* Candidate Key
* Primary Key
* Foreign Key.

Key : * are very important part of relational database Model.
* It is used for identifying unique rows from table.
* it also establishes relationship among tables.
* Keys in DBMS is an attribute or set of attributes which help you to identify a row (tuple) in a relation (table).

Eg -

A	B	C	D
1	a	X	
2	b	Y	
3	b	X	
4	C	Y	

Row/tuple/record →

[columns have name but rows didnot name]

keys: A → BC (Based on the value of A, we can find the row uniquely)
BC → A
C → AB X

① Super Key → Super Key is a set of attributes which may contain one or more attributes, using the value of which you can identify or select a row uniquely among the set of rows.

Key.

Candidate Key → attributes.

R C A B C D)

↙ Table

		Super Key	Cand Key
①	A → BCD	✓	✓
②	AB → CD	✓	X
③	ABC → D	✓	X
④	BD → AE	✓	✓
⑤	X (C) → AD	X	X

(B is missing)

↳ not able to find the Entire row.

* Both (left + write) = complete row
* from left side, we can find right hand side.

→ All the above Four Keys satisfy themselves to become a Super Key.

→ But we will try to find a Minimal Super Key (means can you do it in a Minimal no. of attributes).

→

ABC is a S.K / Key.
↳ is this a Candidate Key. [No].

→ Only Minimal Super Key becomes Candidate Key.

→ * if a proper subset of Super Key is also a Super Key, then you are not a candidate.

A / BD

Candidate Key → if a proper subset of S.K is not a ~~subset~~ superkey, then it is a candidate key.

E.g → ABC C AB [It is not minimal]
AB C A
A is a candidate key. (there is no subset)

BD — there exist no proper subset of this Super key,
(is B a S.K?) — is this a Super key ✓
(is D a S.K?) — is this minimal. ✓
— so it a candidate key.

Primary Key → There is no difference b/w candidate key and primary key.
→ we can have more than one CK in DB.
→ but we have only one primary key for a table, chosen by DBA.

K, SK, CK & PK.

K ⟷ SK → CK → PK.

Alternate Key → all candidate key's except PK are called Alternate keys.

R (A B C D)

① R(ABCD)

$A \to BC$

X Key / SK.

② R(ABCD)

$ABC \to D$ ✓

$AB \to CD$ ✓

$A \to BCD$ ✓

S.K is the basic Key.

$X \longleftrightarrow SK$.

③ R(ABCD)

$B \to ACD$

$ACD \to B$

effiient version of SK is CK.

B, ACD
AC ✗
AD ✗
CD ✗
A ✓
C ✗
D ✗.

④ R(ABCD)

$AB \to C$

$C \to BD$

$D \to A$

→ A PK is that candidate key which is selected by a DBA as a primary mean to identify Tuple.

→ P.K (Unique + Not Null)

Functional dependies:

fd: $X \to Y$

$f(x) \to y$

	X	Y
$t_1 \to$	a	1
$t_2 \to$	a	2
	c	3
	d	4

[If I tell you the value of X, then using that value you can search the value of Y)

$a \to 1$ ✓

$a \to 2$ ✗

$b \to 1$ ✓

(we can get different value of B, on same value of X)

fd: $\alpha \to \beta$

trivial.	non-trivial
$AB \to A$	$B \not\subseteq X$
if $B \subseteq X$	

$\alpha \rightarrow \beta$ [if we are getting 2 different values of β on same value of α, then this dependency is not valid.].

Eg →

(a) $A \rightarrow BC$
(b) $DE \rightarrow C$
X (c) $C \rightarrow DE$
(d) $BC \rightarrow A$

R

A	B	C	D	E
a	2	3	4	5
2	a	3	4	5
a	2	3	6	5
a	2	3	6	6

Eg →

$\alpha \rightarrow \beta$
$A \rightarrow B$
$A \rightarrow CD$
$A \rightarrow BCDE$

R

A	B	C	D	E
a	2	3	4	5
b	a	3	4	5
c	2	3	6	5
d	2	3	6	6

[if all the values of α are different, then dependency holds.].

Eg - $A \rightarrow C$
$BD \rightarrow C$
$ABDE \rightarrow C$

[if BC has same value, then dependency is valid]
→ in this case α becomes irrelevent.

Closure set / Attributes closure / Attributes set /

Closure of attribute set →

Attribute closure of an attribute set e_A 9 can be defined as a set of attributes which can be functionally ~~def~~ determined from it.

- Denoted by F^+

E.g - R(ABC)

$A \rightarrow B$

$B \rightarrow C$

$(A)^+ = A$
AB
ABC

(attributes which we can identify from A) (directly or indirectly)

R(ABCDEF)

$A \rightarrow B$

$C \rightarrow DE$

$AC \rightarrow F$

$D \rightarrow AF$

$E \rightarrow CF$

$(D)^+ \Rightarrow D$
$= ADF$
$= ABDF$

$(DE)^+ = DE$
$= AFDE$
$= AFCDE$
$= ABCDEF$

Date:
Page:

Lecture-10

Key →

① **Super Key** - A SuperKey is any combination of attributes within a relation that uniquely identify each record / tuple within that relation.

Example - STUDENT (RollNo., Name, Address) Relation

SuperKey
- RollNo.
- RollNo., Name.
- RollNo., Address
- Name, Address

* - More than one super key is possible in a relation.

② **Candidate Key** → A candidate key is unique & Not null that identifies uniquely identifies a record / tuple of a relation.

- Candidate Keys are Minimal Super Keys.

Example -

Super Key	Candidate Key
① RollNo.	① RollNo.
② RollNo., Name	
③ RollNo., Address	② Name, Address
④ Name, Address	

- Each candidate Key can work as ~~possible~~ Primary Key

③ Primary Key → A primary key is a set of one or more attributes of a relation that uniquely identify a record / tuple of a relation.

- Primary Key attribute has no duplicate & Null values.
- Only one candidate key can be a primary key.

Eg - Candidate Key
① Roll No.
② Name, Address

Primary Key
① Roll No.

- A Relation can have only one primary key.
- A primary key is a candidate key that is chosen by the database designer to identify the records uniquely.

④ Alternate Key →

- A candidate keys that are not selected to be the primary key are called as alternate keys.
- Example- Candidate Key
① Roll No. —— selected as primary key
② ~~Roll No.~~ Name, Address — Alternate key.

* A PK that is made up of more than one attribute is known as composite key.

Example: Work (EID, PID, Hours-worked)

EID	PID	Hours-worked
1	1	200
1	2	300

⑤ Composite Key -

When a single attribute can't be used for unique identification, then a combination of attributes are used as a key. Such a key is called a composite key.

⑥ Foreign Key -

- Primary key in one relation (table) can be put in another relation, where it will act as foreign key.
- Foreign key is used to establish relationship b/w the two relations (tables)
- Foreign key can have null values & duplicate values as well.
- Foreign key is the attribute of a relation that is primary key in another relation.

Self Referencing foreign key - (or Recursive foreign key)

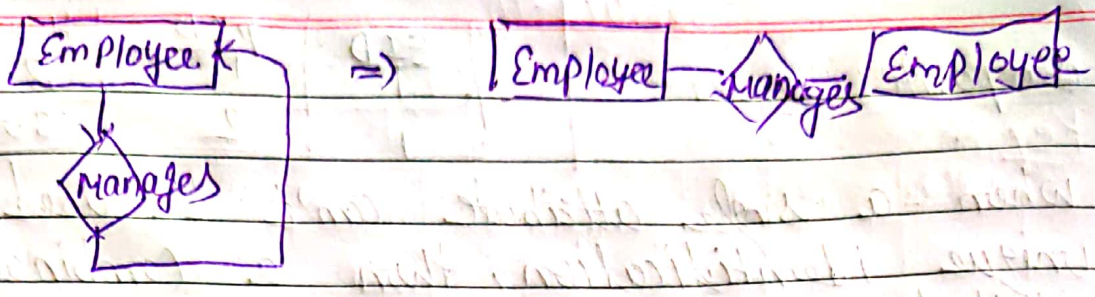
- A foreign key that refers to its own relation is called self-referencing foreign key.

Example: Employee RELATION

Primary key ↓ ↓ Foreign key

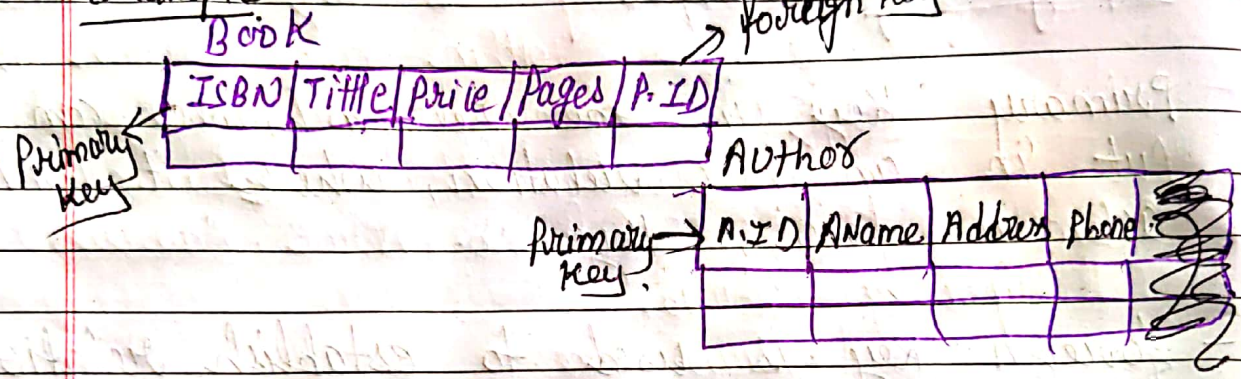
EID	ENAME	D.No.	salary	Manager
E01	RAM	D01	25,000	E10
E02	ALICE	D07	30,000	E07
E03	BOB	D09	25,000	E07
E04	NEHA	D10	50,000	NULL

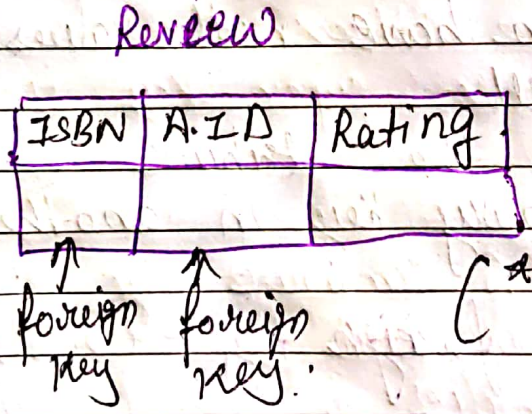
Fig - Employee relation with self-referencing foreign key



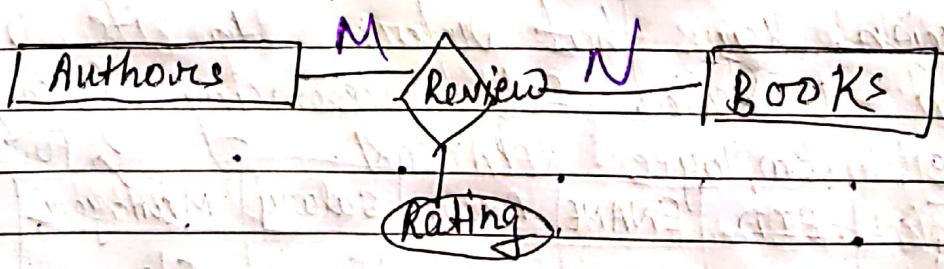
- One relation can have more than one foreign key.

Example -





(* more them one foreign key)



Difference between Primary Key & Foreign Key.

	Primary Key	Foreign Key
1.	A PK is a set of one or more attributes of a relation that uniquely identify a record / tuple of a relation.	1. Foreign Key is the attributes of a relation that is Primary Key in another relation.
2.	A PK is used to uniquely identify each tuple in a relation.	2. A FK is used to create a link between two relations.
3.	A PK value can never be Null.	3. FK can have NULL values.
4.	PK attribute has no duplicate value.	4. FK attribute can have duplicate values.
5.	There can be only one PK of a relation.	5. There can be more than one FK in a relation.

Example:

Book

ISBN	Title	Price	Pages	P.ID
978-1-004-111	Maths	250	100	P001
601-187-1802	C	250	150	P001
001-786-5503	C++	200	150	P002

PK

F.K

Publisher.

PID	P.Name	Address
P001	XYZ	G.N
P002	ABC	Delhi

P.K.