Ex 1 (1)

→ In mobile phones, diff. price at Flipkart, Amazon & Store so, not Integrity.

(2.) In university, a Student Reg no. to same at every office in the university, in library, in dean office. so, Integrity present here.

Ex 2

Student (Base Table or Referenced Table)

Roll No.	name	add.
1	A	Delhi
2	B	Mumbai
3	A	Chd.
4	C	Chd.

P.K. ✓ — Deletion

C. Id	C. name	Roll. No
C_1	DBMS	1
C_2	networks	2
C_3	Ckt	7

F.K.

Course (Referencing table).

(#) Referenced Table →

1.) Insert :→ No violation (We can easily add).

2.) Delete :→ May cause violation (bcz, if we delete a row & with same roll. no. a row is in referencing table, Then, it's not possible).

Sol^{n} :→ 1.) On delete Cascade
(Also delete from every table).

2.) On delete set Null
(Insert NULL on other tables).

~	~	Roll. No
		NULL

f.k. can, take reference from P.K. of same table, also.

But, if same col^n is also a primary key in that other table. Then, we can't use this sol^n. bcz,

primary key cannot be NULL.
<unique, not NULL>

3.) on delete No Action.
(in this sol^n, our row dont get deleted, first we have to delete from other tables, than we can delete in our table).

3.) updation: → may cause viola^n (bcz, if we make '20' of '2', then now how we can get Roll No '2' in referencing table).
Sol^n !: →
1.) On update Cascade
2.) On update Set NULL.
3.) On update No Action.

(#). Referencing table: →

1.) Insert: → May cause viola^n (bcz, it '1'le not base table, then how it in Referencing table)

2.) Delete: → No violation

3.) updation: → May cause violation. (bcz, we cant make '20' of 2 if '20' is not in base table).

Note: 1.) Same key can be foreign & primary key in a table.
2.) Many table can have a foreign key from a single table by take reference of its P.K.

(12) Q:-

Let $R_1(a, b, c)$ and $R_2(x, y, z)$ be 2 relations in which 'a' is foreign key in R_1 that refers to primary key of R_2. Consider, 4 options →

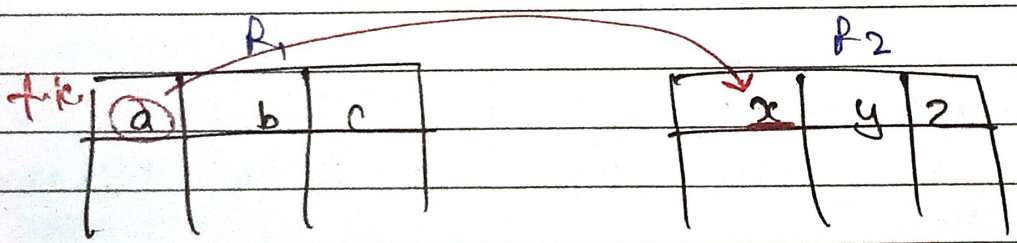
a) Insert into R_1 — ✗
b) Insert into R_2 — ✓
c) Delete from R_1 — ✓
d) Delete from R_2 — ✗

Which is correct regarding referential Integrity ?

1.) option a & b cause violation
2.) option b & c will cause violation
3.) option c & d — " —
4.) option d & a — " — → (Ans)

Let x be p.k. in R_2.

Sol^n :-



Referencing Table. Base Table (or) Referenced Table.

Note :- If f.k. is not there, then we can do anything (insert, delete, update) without any violation.

(13) SUPER KEY in DBMS :→

2). Super key is a combination of all possible attributes which can uniquely identify 2 tuples (row) in a table.

→ Candidate key is minimal.

Eg:-

Candidate key (C.K.) = Roll No. (also Super key)

Roll no.	name	age

- Roll no, name
- Roll no, age
- Roll no, name, age

all are Super key.

Candidate key (C.K.) के साथ कुछ भी add कर दो, तो super key हो जाएगी।

2). name, age × (not Super key).

→ Super set of any Candidate key is Super key.

Q:- R(A_1, A_2, ... A_n) then how many Super keys are possible.

If → A_1 is Candidate key.
→ A_1, A_2 are candidate keys.

Ans:- power Set :→ how many Subsets can be possible of given set.

Eg:- → A_1 A_2 A_3 (Either take or not take So, 2 possibility)

$2 \times 2 \times 2$ → 8

→ 2^3.

Now, Sol^n → R ($A_1, A_2, \ldots A_n$).

i) ⇒ (इसमें A_1 तो आएगा ही).

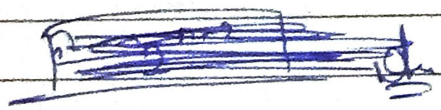
A_1
A_1, A_2
A_1, A_3
A_1, A_2, A_3, A_4 .

R ($A_1, A_2, A_3, \ldots A_n$)
↙ $n-1$
Compulsory.

$1 \times \underbrace{2 \times 2 \times \ldots 2}_{n-1}$.

$\therefore$ 2^{n-1} Ans.

ii) R ($A_1, A_2, A_3, \ldots A_n$)
$n-2$



→ A_1, A_2 both C.K.

when A_1 →	when A_2 →
A_1	A_2
A_1, A_2	A_2, A_1
A_1, A_2, A_3	A_2, A_1, A_3
2^{n-1}	2^{n-1}

But, some sets are common (bcz $A_1A_2 = A_2A_1$).

$\therefore$ Common are those who has both A_1, A_2.

$\therefore$ those are 2^{n-2}.

Now,

$2^{n-1} + 2^{n-1} - 2^{n-2}$ ✓

(or)

$2^{n-1} - 2^{n-2}$.

iii) $A_1 \cdot A_2$ combined is C.K.

then,

2^{n-2} ✓.

iv) $A_1 A_2$, $A_3 A_4$ are C.K.

2^{n-2} 2^{n-2}

Now,

$A_1 A_2 A_3 A_4$	$A_1 A_2 A_3 A_4$
$A_1 A_2 A_3 A_4 - -$	$A_3 A_4 A_1 A_2 - -$

To remove these common Elements.

$A_1 A_2 A_3 A_4 - - - - A_n$

2^{n-4}

$2^{n-2} + 2^{n-2} - 2^{n-4}$ ✓

$2^{n-1} - 2^{n-4}$.

06.

14. E-R Model → (Entity Relationship Model)

→ used for logical representation.

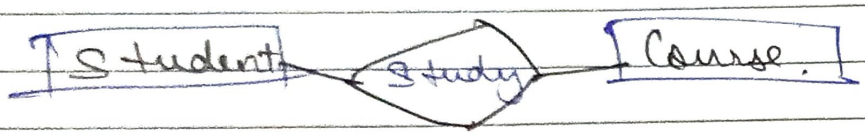
→ To see logical structure before implementation (Design).

→ It does the job of database design.

Entity - Any object which has physical existence is Entity.

Ex:- Student (Roll no, age, address)
(Entity) attributes

→ Relationship → Relationship b/w 2 or more Entities



Relationship b/w student & course is of Study.

⇒ Student (Roll no, age, address)
Entity type (Schema).

⇒ We implement these by using SQL.
(Structured Query lang.).

→ Entity.
→ Attributes - Characteristics of Entity.
↳ types.
→ Relationship - ↳ types → 1 to 1
1 to many
many to 1
many to many } 4 types

Entity → Student → represented by rectangle
Attribute →
Relationship →

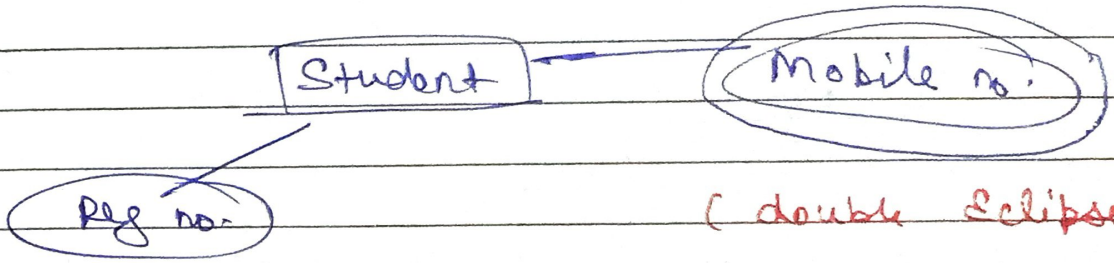
(15.) Types of Attributes in ER Model :-

[Student]

1.) Single vs Multivalued attributes :→

Single → Reg no.

Multivalued → Mobile no. or address. (May be more than 1 C.No. of a student)



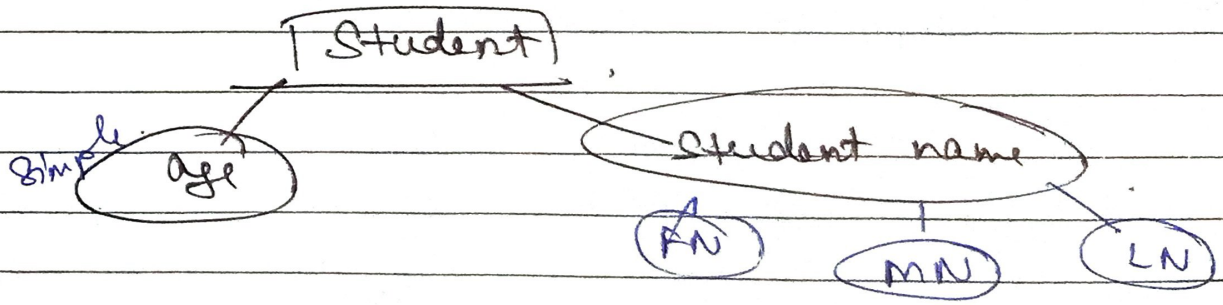
(double Eclipse).

2.) Simple vs Composite Attributes.

Simple - can't be further divided.

Composed - composed of more than 1 value.

Ex:- name (first name, middle name, last name)



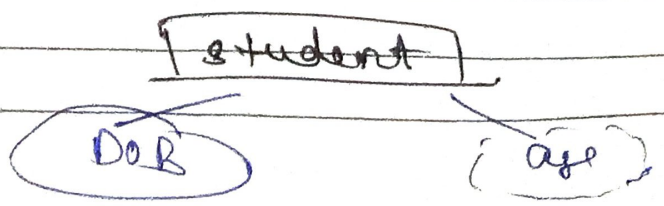
3.) Stored & Derived Attributes :→

Stored → These are stored & can't be derived.

Ex - DOB

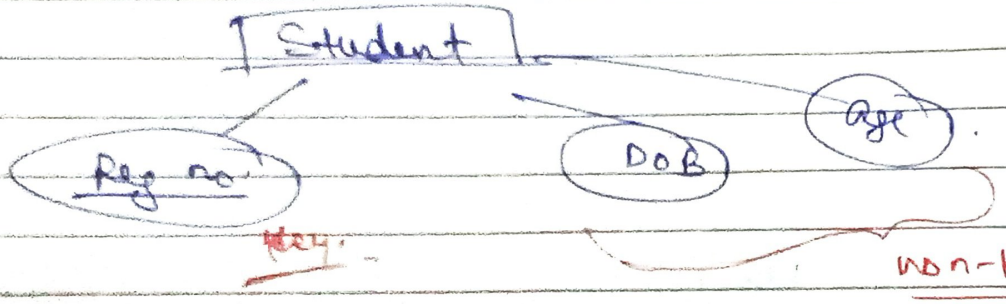
Derived → Ex - Age. (derived from D.O.B.)

dotted Eclipse.



4.) Key vs Non-key Attributes :- →
Key - used to uniquely identified.
Unique (No Repitition)

Ex:- Reg No is always unique for a student.
Represent with underline (___).

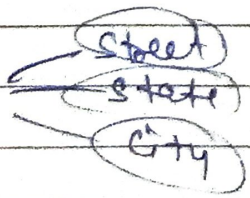


5.) Required vs optional Attributes :- →
Required → These are Mandatory (*) Name
optional → can also be leave. D.oB,
Add.,

6.) Complex Attribute :- →
(Composite + Multivalued)

Eg:- If a student have 2 Residential Add, & in each Add., he have 2 phone no.
Multivalued.

Add. is Composite

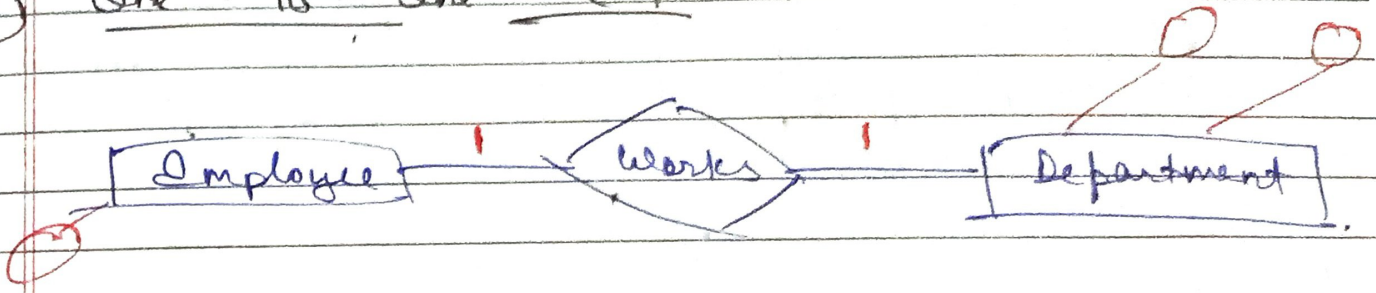


16. Degree of Relationship :- → (Cardinality).
→ how the Entities are connected with each other.

4 types! →

1) 1 - 1 — One to one
2) 1 - M
3) M - 1
4) M - M (M - N) — Many to many

(1) One - to - One (1 - 1) : →



Convert Entity into Table : →

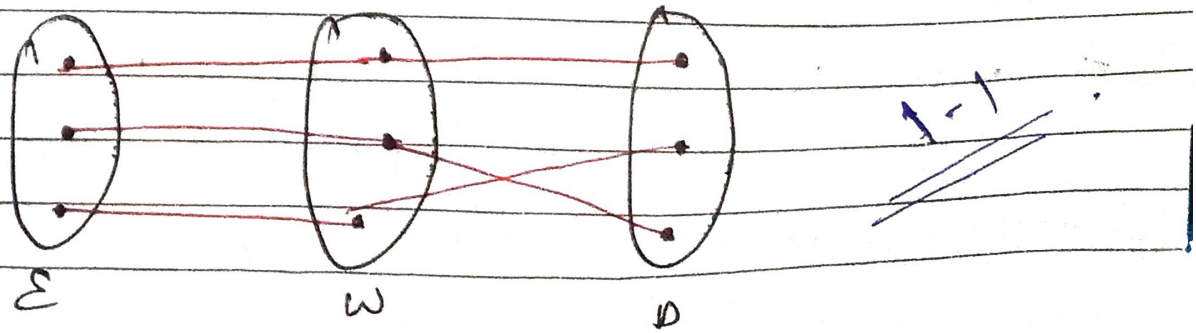
(Relo'ship का भी Table बनेगा, तो हमें Employee & Department का Rel^ बनाना है।)

Rel'ship Table! → Attributes? → 2 always (primary keys of both the table)

E.ID & D.ID

These, E.ID & D.Id works as a foreign key (F.K)

→ When we have to enter data in this Relo^n ship table, then we have to see Rel.ship (1-1, 1-M, ---).



→ P.K. = Either E_Id or D_Id.
(primary key)

E_Id	E name	age
E_1	A	20
E_2	B	25
E_3	C	28
E_4	A	24
E_5	B	25

f.k. f.k.

EID	D_ID
E_1	D_1
E_3	D_2
E_2	D_3

D_ID	Dname	Loc
D_1	IT	Banglore
D_2	Prod.	Delhi
D_3	HR	Delhi

Both

PK = E_ID

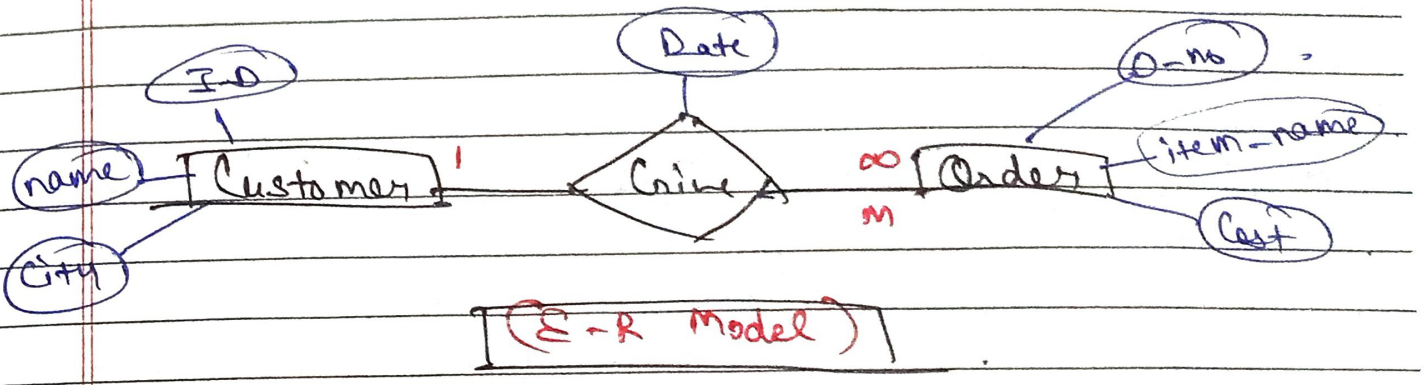
* Can we Merge?
Now, In Table 1 & Table 2, E_ID is the primary key.
Hence, we can merge Table 1 & 2.

E_ID	E name	age	D_ID
E_1	A	20	D_1
E_2	B	25	D_3
E_3	C	28	D_2
E_4	A	24	—
E_5	B	25	—

→ Now, we have 2 Table at final !!.
(Merge Table & Department Table)

Every Table must have its primary key.

(17) One to Many Rela^n ship :→ (1-M).

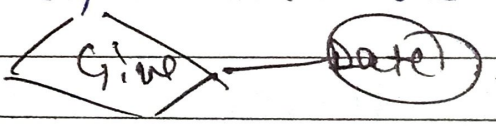


(E-R Model)

→ When we physically implement the E-R Model, then we need Relational Model & we use Tables in Rela'al Model.

2) Rela'ship may have its attribute.

(Ex-1)



& we call it Descriptive Attribute.

f.k. f.k.

Id	Name	City
C_1	A	Tal.
C_2	B	Delhi
C_3	C	Mumb
C_4	A	Mumb

I-D	O-no	Date
C_1	O_1	—
C_1	O_2	—
C_2	O_3	—
C_2	O_4	—

O-no	Item name	Cost
O_1	Pizza	100
O_2	Burger	200
O_3	pasta	300
O_4	Cold-Drink	400

Here O-no is always diff. & so unique.

P.K. = (O_no).

Note:- Always P.K. of the many side in (1-m) is also the p.k. of the relo"ship Table.

Can we Merge Tables? (many diff side merge).

Yes.

By Relo"ship & Order Table.

(bcz both have same P.K.).

=>

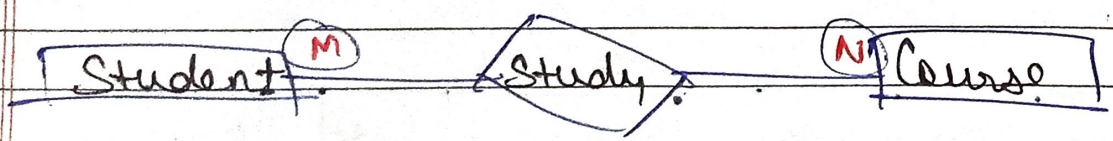
ID	O_no	item name	Cost	Date
~	~	~	~	~

Now, 2 Tables.

(M-1) is also same like this.

18. Many-to-Many Relo"ship :→

(M-N)



f.k. f.k.

Roll no	name	age
1	A	16
2	B	17
3	A	16
4	C	17
5	D	15

Base Table

Rollno	C-id
1	C1
2	C2
1	C2
2	C4
3	C3

Referencing Table

C-id	name	Credit
C1	Maths	4
C2	phy.	4
C3	Chem.	4
C4	Hindi	4

Base Table

Many - Many.

P.K. in Referencing Table (Relationship Table) :→

Roll. No repeats &
C-id also repeats.

So, Roll no. & C-id both make P.K. Combinely.

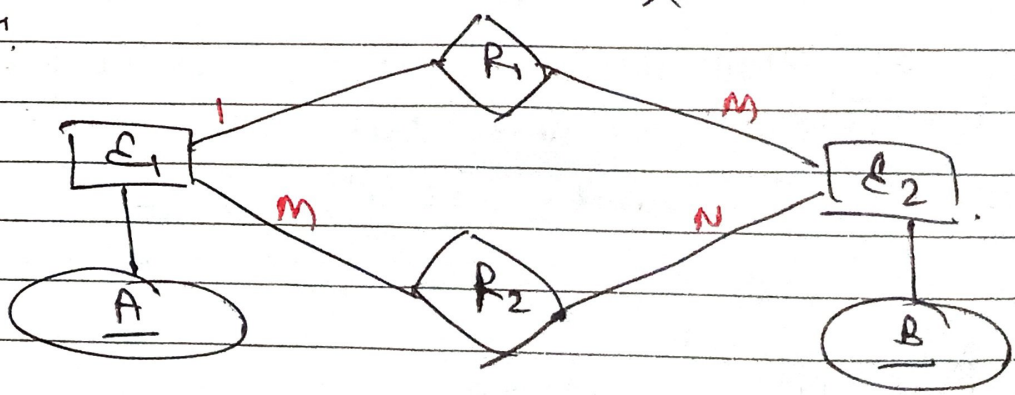
i.e, Composite key = Roll no. C-id

Can we Reduce Tables ?

→ No : bcz, p.k. is combined.

Note:1 p.k. in Relationship Table depends on Relationship (1-1, 1-M, ... M-M).

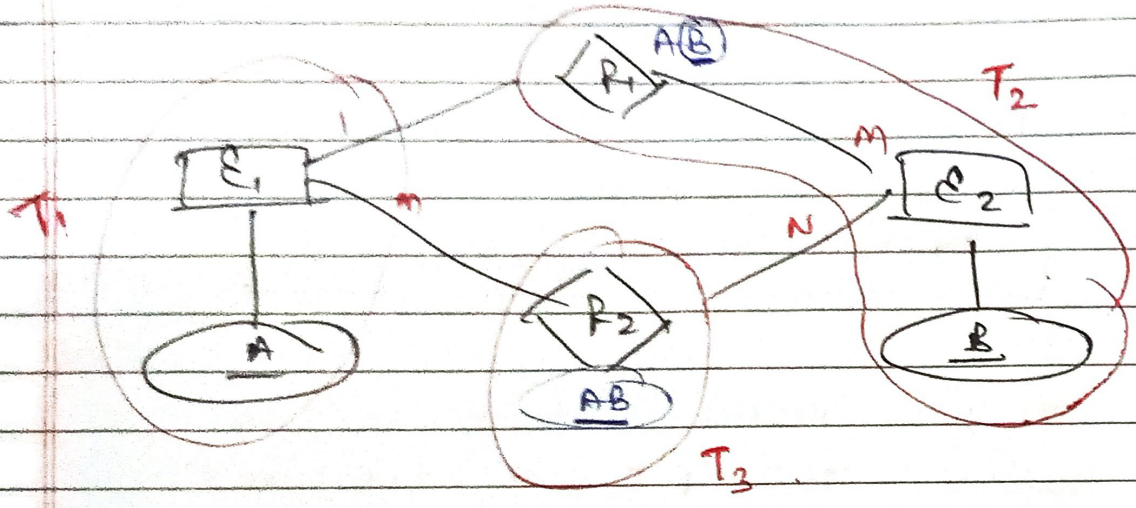
Q. Q:→



* What is the min^m no. of tables required to represent this E-R model into Relational Model ?

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

E_1
E_2
R_1
R_2
} 4 Tables ... But Minimum ?



Hence,

$T_1 = E_1$
$T_2 = R_1 E_2$
$T_3 = R_2$
} 3 Tables ✓

E_2 का सभी attributes $R_1 E_2$ के Combined Table में आ जाएगा। So Now, we also don't need Separate E_2 Table. $\boxed{\text{Min}^M = 3}$ ✓

---

(20) **Normalization** :→

It is a technique to remove or Reduce Redundancy (Duplicacy) from a Table.

→ There are 2 types of duplicacy in Database!→
1.) Row level
2.) Column "

(#) Row level!→

S-Id	S-name	Age
1	Ram	20
2	Varun	25
1	Ram	20

Same (Duplicate) =!

Row level.

→ We use the concept of primary key (P.K.)
We set a P.K. to any appropri. attribute.

Primary key (Unique + Not Null).

P.K. will take care of this duplicacy.

(#) Column-level!→

	Student		Course		Faculty		
P.K.	S-Id	S-name	C-Id	C-name	F-Id	F-name	Salary
	1	Ram	C_1	DBMS	f_1	John	30,000
	2	Ravi	C_2	Java	f_2	Bob	40,000
	3	Nitin	C_1	DBMS	f_1	John	30,000
	4	Anurag	C_1	DBMS	f_1	John	30,000
	5	Varun	C_{10}	MBBS			

⇒ 4 columns are same in many rows.

~~Anomaly~~

→ Insertion Anomaly
→ Deletion "
→ updation "

(→ Anomaly means problem, occurs on special occasion.)

Now,

S.Id	S.name	C.id	Cname	F.id	Fname	Salary
1	Ram	C_1	DBMS	f_1	John	~~30k~~ 40K
2	Rani	C_2	Java	f_2	Bob	40k
3	Nitin	C_1	DBMS	f_1	John	~~30k~~ 40K
4	Amritpal	C_1	DBMS	f_1	John	~~30k~~ 40K.
5	Varun	—	—	—	—	—
		C_{10}	MBBS			

1) Insertion Anomaly :→
→ We want to add data of a new st.
Let, Varun
Let
University starts a new Course,
C_{10} – MBBS
→ We can't insert this info. in table.
Even, we — " — the new faculty data bcz,
→ We only introduce the new Course C_{10}.
We don't talk about the student. and we don't have S. Id.
→ We also remain it (s.id) NULL - bcz it is a P.K.
→ So, we can't insert directly.

It is the our Insertion Anomaly.

2.) Deletion Anomaly :->

We have simple query = Remove the database of Roll. No. 1.

```
Delete from student
where S-id = 1.
```

:- It will delete the whole row.

We don't face any problm here.

Now,

We have to delete the data of Roll. No 2.

```
Delete from student
where S-id = 2
```

→ Row 2 is deleted fully from database.

Now,

Row 2 is blank there.

Now,

tell us who is teaching to Roll No. 2 & what was the course name of Roll. No 2.

likely be, It was only one student who was studying that particular course & that particular faculty is teaching that course.

⇒ We here, only delete the detail of student but, bcz of him, all the info. get deleted.

Course Info — lost
Faculty Info — lost.

i.e, xtra Info is removed here. & we can't recover it later.

3.) Updation Anomaly :→

Simple query → S-id-4 change name from Amrit to Amritpal.

```
Update student
Set Sname = 'Amritpal'
Where S-Id = 4
```
→ Code.

No problem here.

Now,
If we want to change the salary of faculty F1 from 30k to 40k.

Change Salary of F1 from 30k to 40k.

Now, how many times the F1 repeats in table, the same no. of times updation query runs & changes them all from 30k to 40k.

Note:- There is only 1 faculty F1, then his salary must also changes 1 times. But, due to the column level duplicacy, it runs no. of times. Hence, it takes more time.
It is updation anomaly.

* Now, Normalizn removes Redundancy.

How ?

→ a simple Rel^n may be, it we divide that table into multiple tables. Like,

P.K.	S-id	S-name

P.K.	C-id	C-name

P.K.	F-id	F-name	Salary.

This can be 1 of the Rel^n.

→ Now, we don't get any anomaly in insertion, deletion & upda^. There is no effect on others. Easy...

---

21. # First Normal Form :→ (1 NF)

Ef Codd → father of R.B.M.S

→ Table should not contain any multivalued attribute.

Student

Roll No.	Name	Course
1	Sai	C/C++
2	Anurag	Java
3	Onkar	C/DBMS

→ Not in 1st NF

Null means not available

Now, how to convert in 1st NF ?

1st way

Roll no.	Name	Course
1	Sai	C
1	Sai	C++
2	Anurag	Java
3	Onkar	C
3	Onkar	DBMS

primary key (P.K.) = Roll no. Course

(Combined, it is composite P.K.)

2nd Way

Roll no.	Name	Course 1	Course 2
1	Sai	C	C++
2	Anurag	Java	Null
3	Onkar	C	DBMS.

P.K. = Roll no.

3rd way

Roll no	Name
1	Sai
2	Anurag
3	Onkar

Base Table

P.K. = Roll no

F.K.

Roll no.	Course
1	C
1	C++
2	Java
3	C
3	DBMS

Referencing Table

P.K. = Roll no Course

F.K. = Roll no.

22. # Closure Method :→

helps
↳ To find all the Candidate keys in the Table.

Ex:- Candidate key (C.K.).

R(ABCD).

FD of { A→B , B→C , C→D } .

(Functional Dependency).

Meaning of closure is that what 'A' can determine.

Here, A is determing B (from FD ①).

closure Syn. ← $A^+ = B$

↳ $A^+ = BCDA$.

(A can determine itself also)

Ex:- Roll no. can determine itself.

A → B
B → C
Transitive .

Now, R(ABCD) has all 4 Attributes that are in $A^+ = BCDA$.

∴ A can determine all the attributes of Table. This is the prop. of the Candidate key.

Now, $B^+ = BCD$

Here, B not determine A.