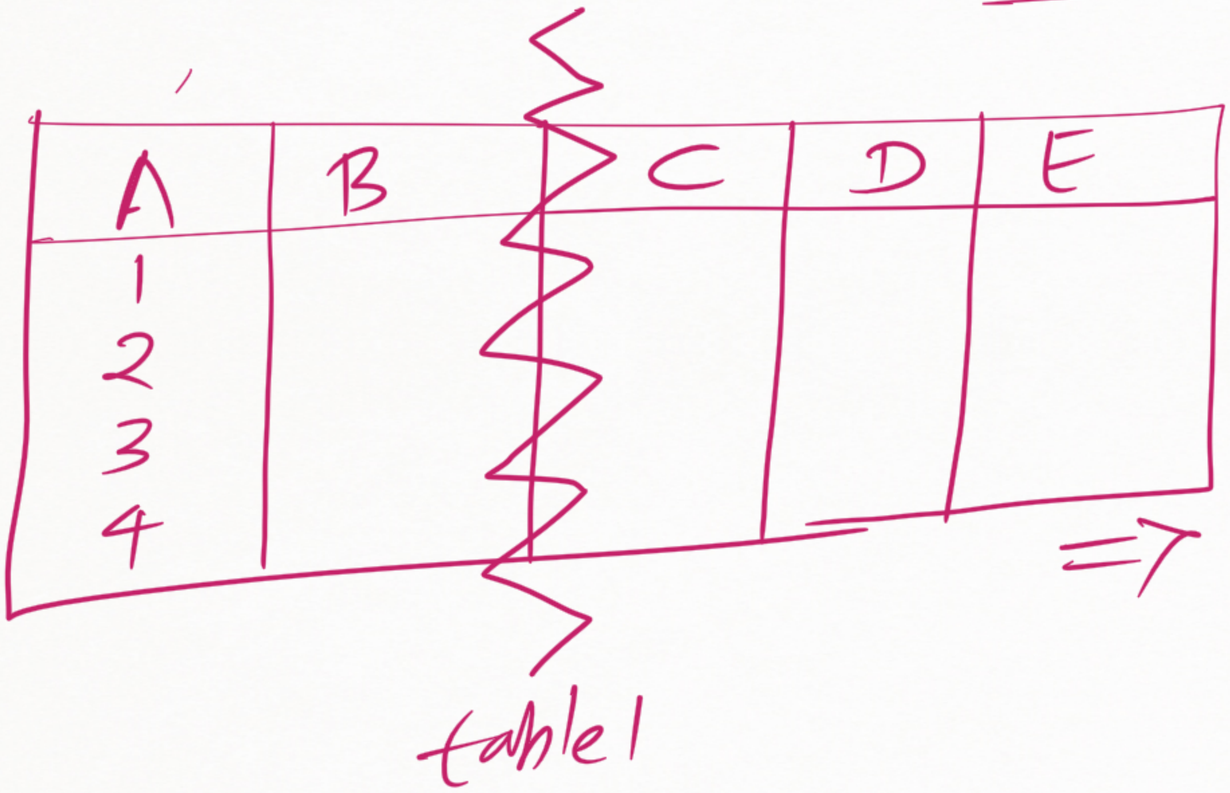
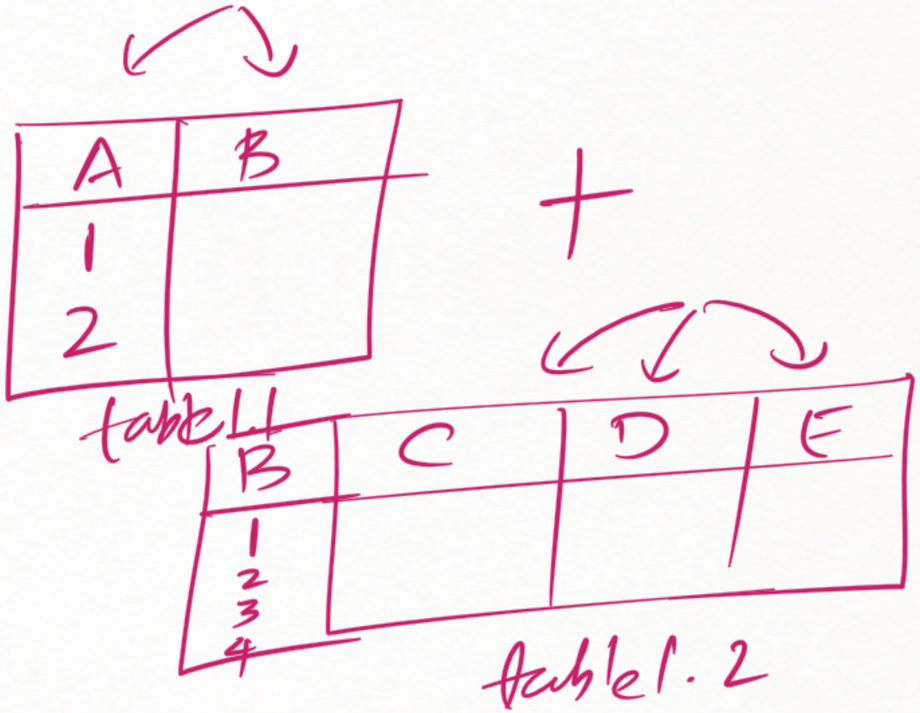
Normalization = Breaking down of table Along the colums by following some rules





Why we Need Normalization

Million

EID	E Name	Dep. ID	Dep. Name	Super ID
101	Syed	A	CS	105
102	Ankit	B	IT	110
103	Suvanna	A	CS	105
104	Mohit	—	IT	—

1000's

Millions of Records

Repeating

① Redundancy =) Repeated data

② Anomalies =) ① Insertion

② Deletion

③ Update

New Employee Joins

Training

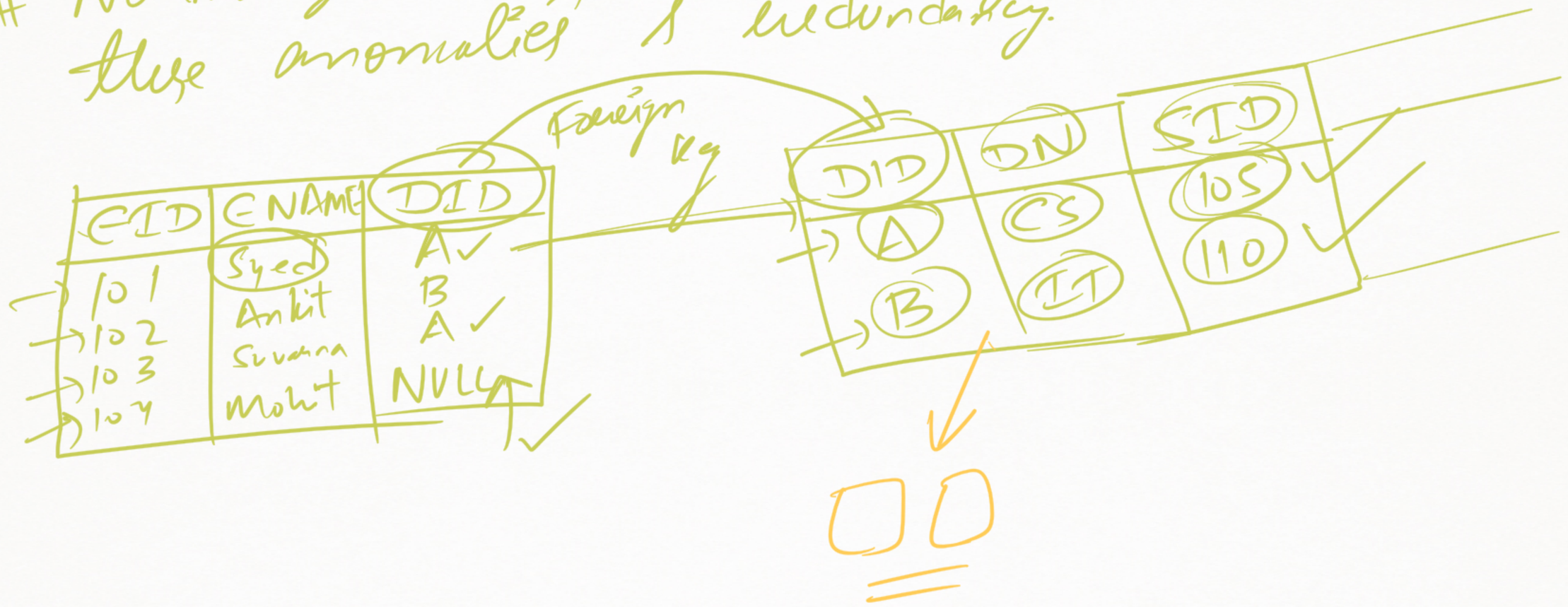
* All emp of DepA are leaving JOB.

=) Automatically DepA will also be deleted because DepA data was stored along with employees but not separately.

each & every one

IT → Information Technology

Normalization if done correctly can avoid these anomalies & redundancy.



Employee

emp_id	first_name	last_name	birth_date	sex	salary	super_id	branch_id
100	David	Wallace	1967-11-17	M	250,000	NULL	1
101	Jan	Levinson	1961-05-11	F	110,000	100	1
102	Michael	Scott	1964-03-15	M	75,000	100	2
103	Angela	Martin	1971-06-25	F	63,000	102	2
104	Kelly	Kapoor	1980-02-05	F	55,000	102	2
105	Stanley	Hudson	1958-02-19	M	69,000	102	2
106	Josh	Porter	1969-09-05	M	78,000	100	3
107	Andy	Bernard	1973-07-22	M	65,000	106	3
108	Jim	Halpert	1978-10-01	M	71,000	106	3

Branch

branch_id	branch_name	mgr_id	mgr_start_date
1	Corporate	100	2006-02-09
2	Scranton	102	1992-04-06
3	Stamford	106	1998-02-13

Client

client_id	client_name	branch_id
400	Dunmore Highschool	2
401	Lackawana Country	2
402	FedEx	3
403	John Daly Law, LLC	3
404	Scranton Whitepages	2
405	Times Newspaper	3
406	FedEx	2

Works_With

emp_id	client_id	total_sales
105	400	55,000
102	401	267,000
108	402	22,500
107	403	5,000
108	403	12,000
105	404	33,000
107	405	26,000
102	406	15,000
105	406	130,000

Branch Supplier

branch_id	supplier_name	supply_type
2	Hammer Mill	Paper
2	Uni-ball	Writing Utensils
3	Patriot Paper	Paper
2	J.T. Forms & Labels	Custom Forms
3	Uni-ball	Writing Utensils
3	Hammer Mill	Paper
3	Stamford Lables	Custom Forms

What if we create just single Table for this huge data?

What if we combine Employee along with the Branch

A LOT OF CHAOS

Normalization Forms
1NF
Reducing Redundancy
2NF
Reducing Redundancy
3NF
3.5NF
OR
BCNF
Boyce Codd Normal form
Normal Form
Betterment Criteria
1NF < 2NF < 3NF
< 3.5NF
Generally this is the ultimate Aim people look for.
Few
Last Normal Form

3.5NF = 3NF + something extra

3NF = 2NF + "

2NF = (1NF) + "

This method can reduce redundancy completely & falls in highest NF.

So many tables

But Query will increase so much

When we have to join so many tables for retrieved of Data.

How to Do Normalization

→ We have to follow some rules

→ To understand those rules, we will have to understand few concepts like Partial Dependency & Functional Dependency.

#Functional Dependency

A	B	C
1	a	b
2	a	c
3	d	e
4	f	g

A → BC

A determines BC

Given the value of A, you will be able to identify B & C

$x^2 = y + 2$

$y = x^2 - 2$

Formal Definition of FD

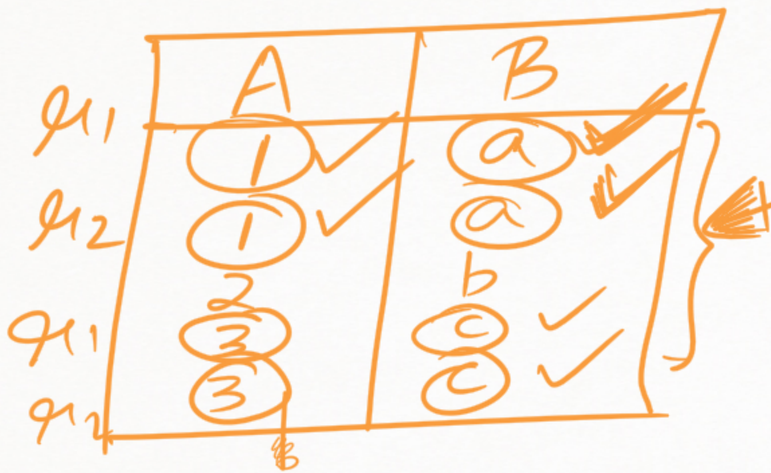
$X \rightarrow Y$

Collection of Attributes

Collection of Attributes

ex:- $AB \rightarrow CD$

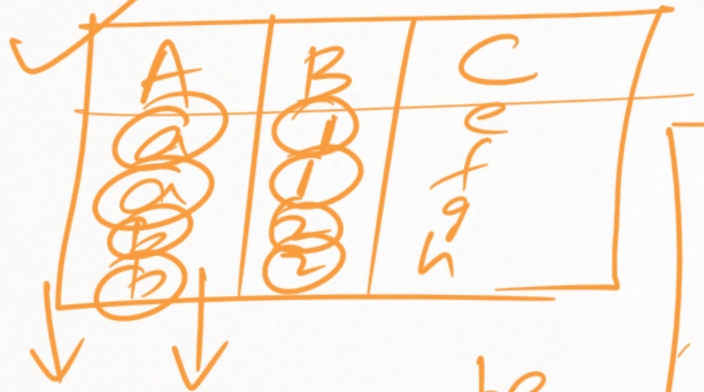
$A \rightarrow BC$



We don't have to worry

CASE	X	Y ✓
1	If t_1 & t_2 are same here ✓	They must be same here also
2	If t_1 & t_2 are diffent here	They may or not be diff here ✓

How to rule out the wrong FD



there can be so many rows, only few are shown to you !!

A → B

Q1. If it is a wrong FD

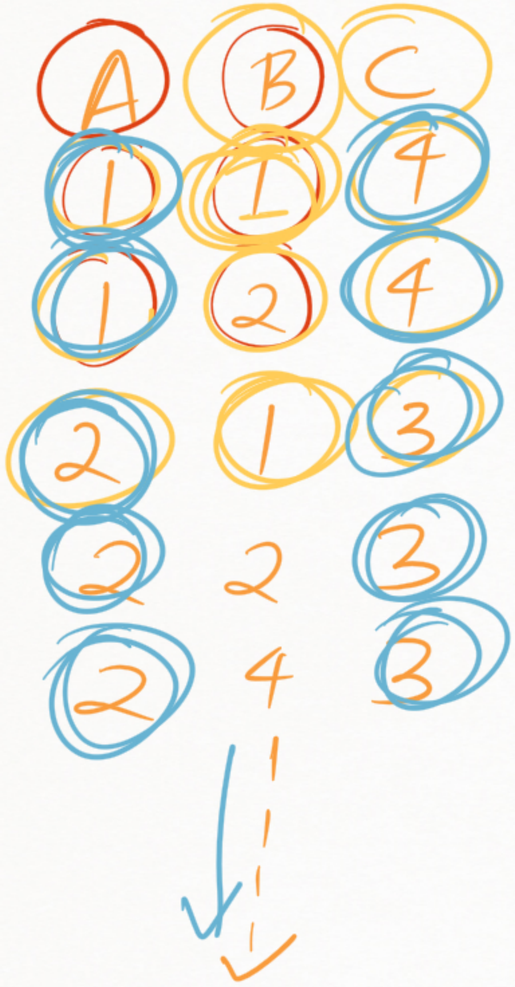
NO

Case 1 ⇒ whenever A repeats, B should also repeat.

A → C

Yes

whenever A repeats C does not.



FDS

1. $A \rightarrow B$
2. $B \rightarrow C$
3. $B \rightarrow A$
4. $C \rightarrow B$
5. $C \rightarrow A$ IDK / I don't know
6. $A \rightarrow C$ IDK

$\Rightarrow$ Which are wrong F.D's

1, 2, 3, 4

Closure Set of Attributes

A	B	C	D	E
✓	✓	✓	✓	✓

FD's

- $A \rightarrow B$
- $B \rightarrow D$
- $C \rightarrow DE$
- $CD \rightarrow AB$

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

Whenever something is here we can use that

(Closure set of A)

$A^+ = \{all\}$ ✓

$CB = \{C, B, all\}$ ✓

Closure of B

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

Closure of C

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

$D^+ = \{D\}$

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

$AC^+ = \{A, C, B, D, E\}$ all ✓

HW

AD^+

BD^+

$CD \rightarrow AB$

If we have complete CD only then

we can determine AB ✓

Candidate Key & Super Key

① Candidate Key => Given an attribute or a collection of attribute, we will be able to uniquely identify a row.

Note => If collection of Attributes, than those attributes have to be minimal.

=> In the news case

Possible to be minimised

CA = {all ✓}

C = {all ✓}

But it will violate the Note

Not a C_K

C_K

For Class II, what is Name Can you ans it without Doubt ?=) NO Mohan OR Rohan !!!

Ck

What is the Name for ID1 = 2 Can you ans it without any doubt ? =) YES Rohan ✓

$ID1^+$ = { Name, ID1, ID2, Class }

Can we Minimise? NO

ID2 Ck

* Primary Key => It is just one of the Candidate Key that we wish to choose.

=> There is a difference when we say Primary Key is a collection of Attributes/Column

VS

There are a ~~collection~~ wrong it does not happen of Primary keys

There is only 1 and 1 Primary Key

Super Key = Candidate Key + Not needed to uniquely identify a row.

Waste

example

ID1	ID2	Name	Class
1	A	M	1
2	B	R	2
3	C	M	1

CK

ID1

CK

ID2

ID1 + Name

ID1 + Last Name

Super Key

Sub	Teacher Name	Dept
A	Syed	IT
B	Ankit	CS
A	Mohit	CS
C	Syed	Maths

What are Ck's here?

What is the dep. of Syed?

Can you ans it without doubt? NO ✓

Any other Ck?

TeacherName + Dept ✓

Sub + Dept ✓

Ck could be a collection of attributes

Minimal

Sub + TeacherName	Dep
A Syed	IT
B Ankit	CS
A Mohit	CS
C Syed	Maths

Can we determine dep from Sub + Teachername without doubt?

Sub + TeacherName ✓

Ck