Database Systems
Normalization

WEEK 10 Lecture 1

Topics To Cover

- Normalization
- It’s Purpose (to remove redundancy)
- Anomalies related to Normalization
- Normal Forms

Normalization

The objective of normalization:

- **“to create relations where every dependency is on the key, the whole key, and nothing but the key”.**

Normalization

A technique of organizing data into multiple related tables, to minimize **DATA REDUNDANCY** and to make the design meaningful**.**

- Normalization works through a series of stages called **normal forms**:
- First normal form (1NF)
- Second normal form (2NF)
- Third normal form (3NF)
- BCNF
- Fourth normal form (4NF)
- Fifth normal form (5NF)

It’s Purpose

- The purpose of normalization is to remove redundancy (not completely but partially).
- Repetition of data increases the size of database.

STUDENTS TABLE

rollno	name	branch	hod	office_tel
1	Akon	CSE	Mr. X	53337
2	Bkon	CSE	Mr. X	53337
3	Ckon	CSE	Mr. X	53337
4	Dkon	CSE	Mr. X	53337

Anomalies Related to Data Redundancy

Following three problems arise due to redundancy in data:

- Insertion Anomaly
- Deletion Anomaly
- Updation Anomaly

Insertion Anomaly

- An **Insert Anomaly** occurs when certain attributes cannot be **inserted** into the database without the presence of other attributes.
- Or to insert redundant data for every new row is a **data insertion problem or anomaly.**

STUDENTS TABLE

rollno	name	branch	hod	office_tel
1	Akon	CSE	Mr. X	53337
2	Bkon	CSE	Mr. X	53337
3	Ckon	CSE	Mr. X	53337
4	Dkon	CSE	Mr. X	53337
5	Ekon	CSE	Mr. X	53337

Deletion Anomaly

- Loss of a related dataset when some other dataset is deleted.

STUDENTS TABLE

rollno	name	branch	hod	office_tel
1	Akon	CSE	Mr. X	53337

We have not stored the branch information anywhere else

STUDENTS TABLE

rollno	name	branch	hod	office_tel

Branch information deleted along with Student data.

Updation Anomaly

- An **update anomaly** is a data inconsistency that results from data redundancy and a partial **update**
- Need to update repeated information in each and every record.
- Data becomes inconsistent even if one record is missed.

STUDENTS TABLE

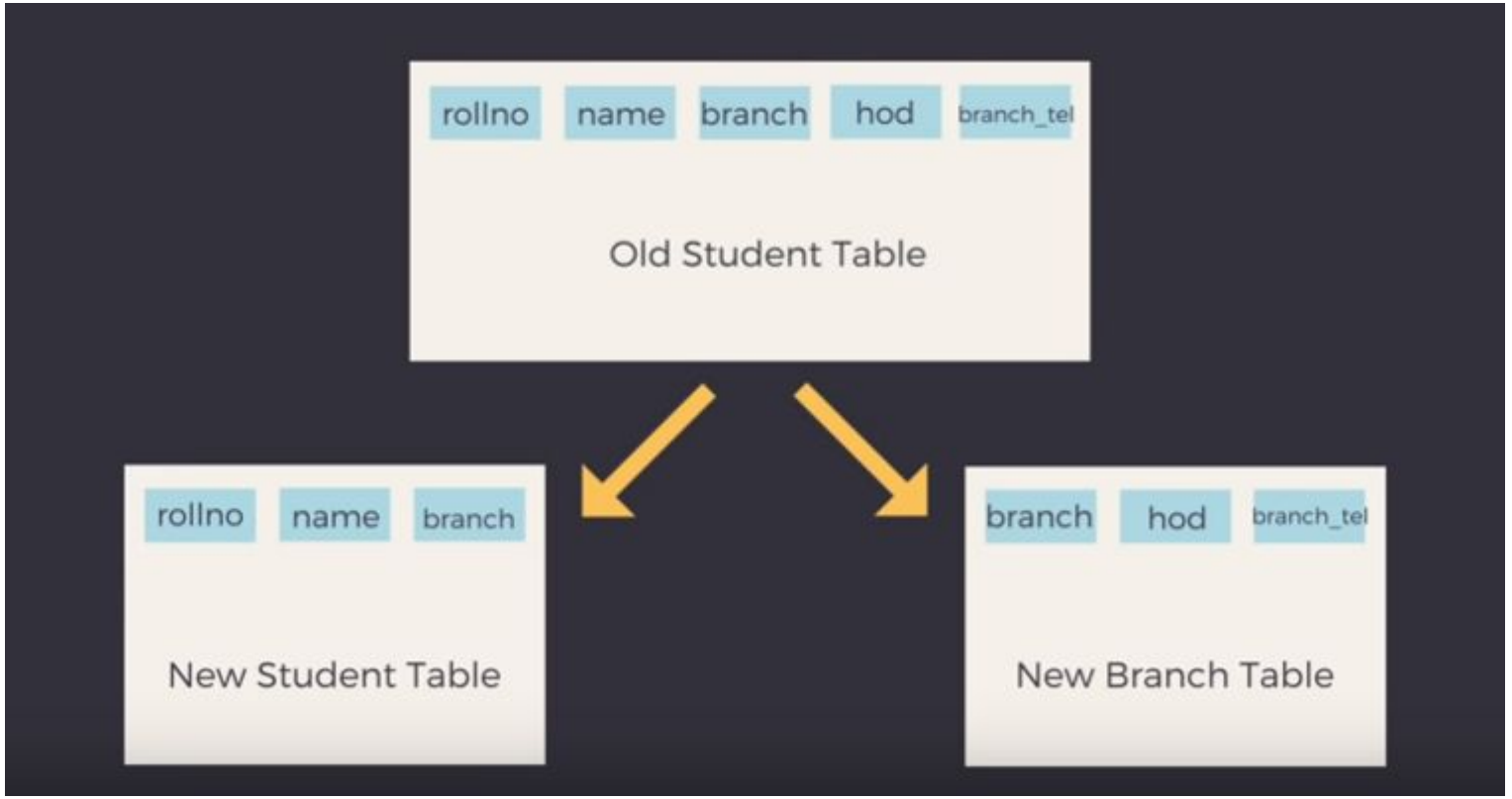
rollno	name	branch	hod	office_tel
1	Akon	CSE	Mr. X	53337
2	Bkon	CSE	Mr. X	53337
3	Ckon	CSE	Mr. X	53337
4	Dkon	CSE	Mr. X	53337

Mr. X leaves, and Mr. Y joins as the new HOD for CSE

STUDENTS TABLE

rollno	name	branch	hod	office_tel
1	Akon	CSE	~~Mr. X~~ Mr. Y	53337
2	Bkon	CSE	~~Mr. X~~ Mr. Y	53337
3	Ckon	CSE	Mr. X	53337
4	Dkon	CSE	~~Mr. X~~ Mr. Y	53337

Result after Normalization

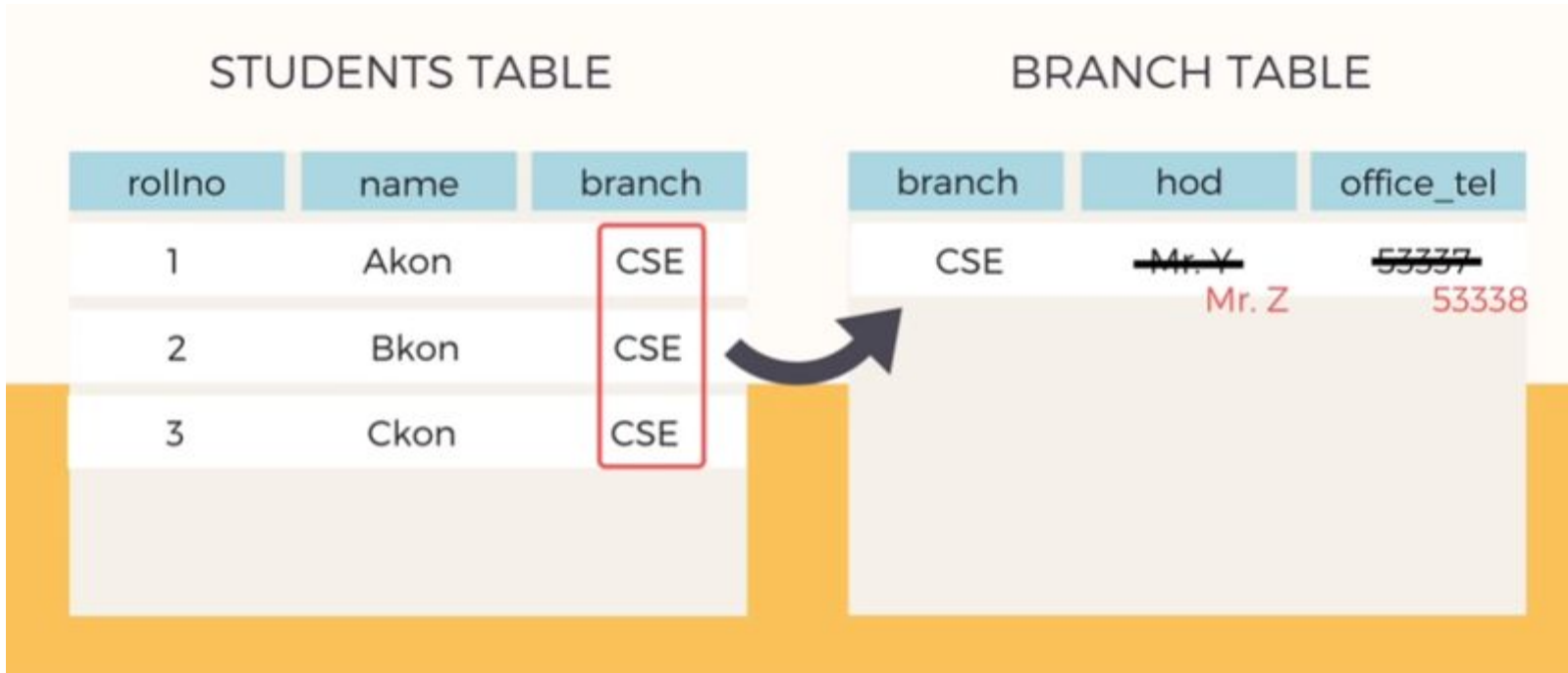


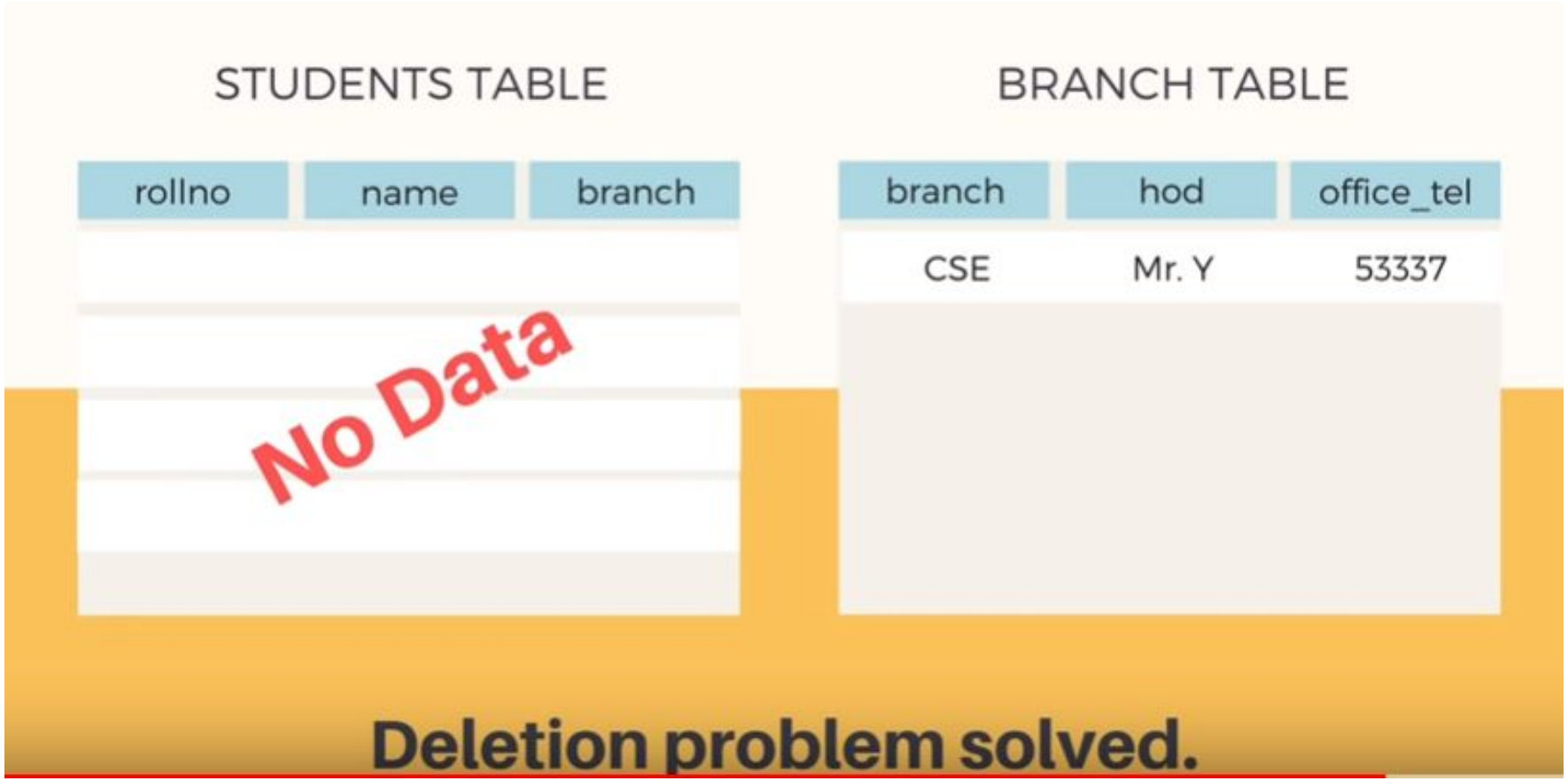
STUDENTS TABLE

rollno	name	branch
1	Akon	CSE
2	Bkon	CSE
3	Ckon	CSE

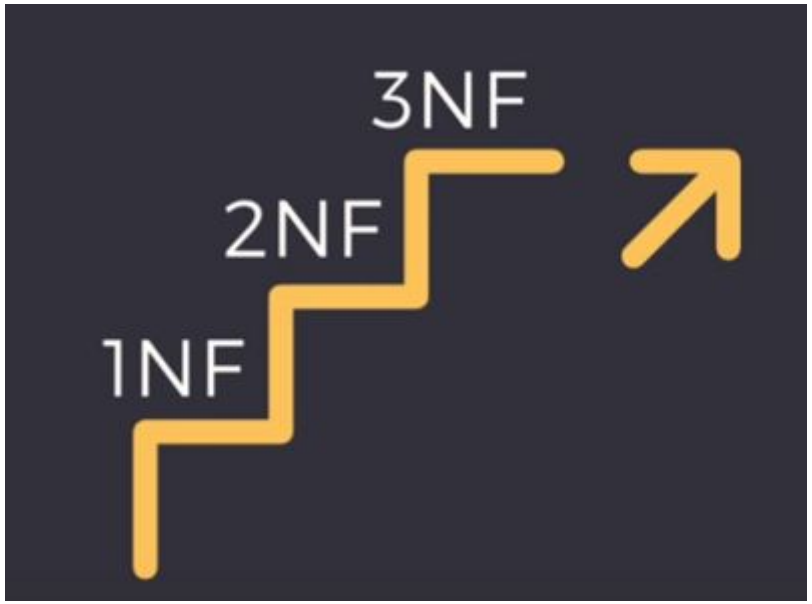
BRANCH TABLE

branch	hod	office_tel
CSE	Mr. Y	53337




STUDENTS TABLE
rollno
name
branch
No Data
BRANCH TABLE
branch
hod
office_tel
CSE
Mr. Y
53337
Deletion problem solved.

Normal Forms



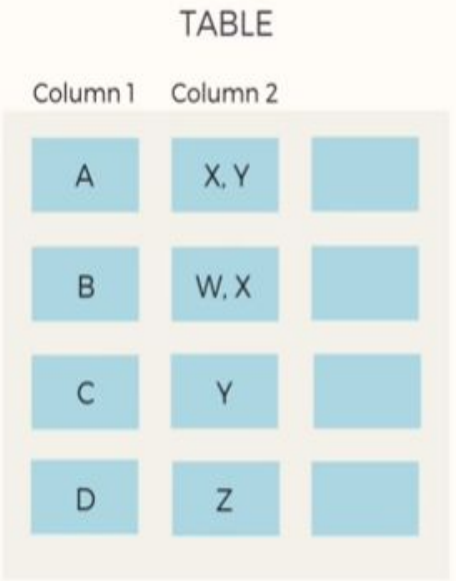
1st Normal Form

Rule 1: Single Valued Attributes

- Each column of your table should be single valued which means they should not contain multiple values. We will explain this with help of an example later, let's see the other rules for now.

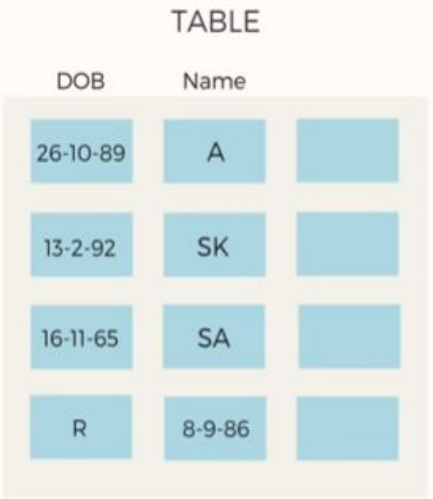
Rule 2: Attribute Domain should not change

- This is more of a "Common Sense" rule. In each column the values stored must be of the same kind or type.



RULE 1

- Each Column should contain atomic values.
- Entries like X, Y and W, X violate this rule.



RULE 2

- A Column should contain values that are of the same type.
- Do not inter-mix different types of values in any column.

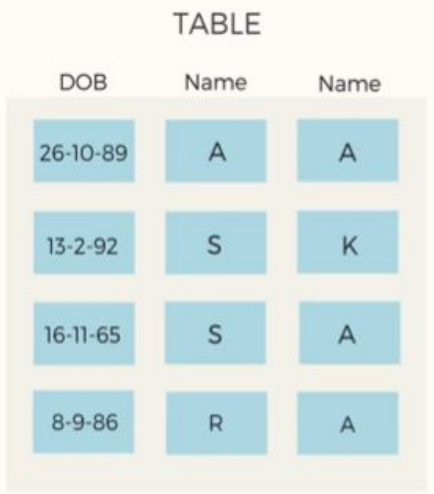
1st Normal Form

Rule 3: Unique name for Attributes/Columns

- This rule expects that each column in a table should have a unique name. This is to avoid confusion at the time of retrieving data or performing any other operation on the stored data.

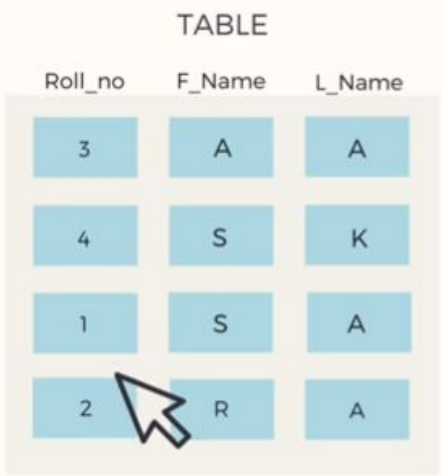
Rule 4: Order doesn't matters

- This rule says that the order in which you store the data in your table doesn't matter.



RULE 3

- Each column should have a unique name.
- Same names leads to confusion at the time of data retrieval



RULE 4

- Order in which data is saved doesn't matter.
- Using SQL query, you can easily fetch data in any order from a table.

STUDENTS TABLE

rollno	name	subject
101	Akon	OS, CN
103	Ckon	JAVA
102	Bkon	C, C++

STUDENTS TABLE

rollno	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	JAVA
102	Bkon	C
102	Bkon	C++

2nd Normal Form

For a table to be in the Second Normal Form, it must satisfy two conditions:

- The table should be in the First Normal Form.
- There should be no Partial Dependency.

2nd Normal Form

What is Dependency?

- Let's take an example of a **Student** table with columns student_id, name, reg_no(registration number), branch and address(student's home address).

- In this table, student_id is the primary key and will be unique for every row, hence we can use student_id to fetch any row of data from this table

2nd Normal Form

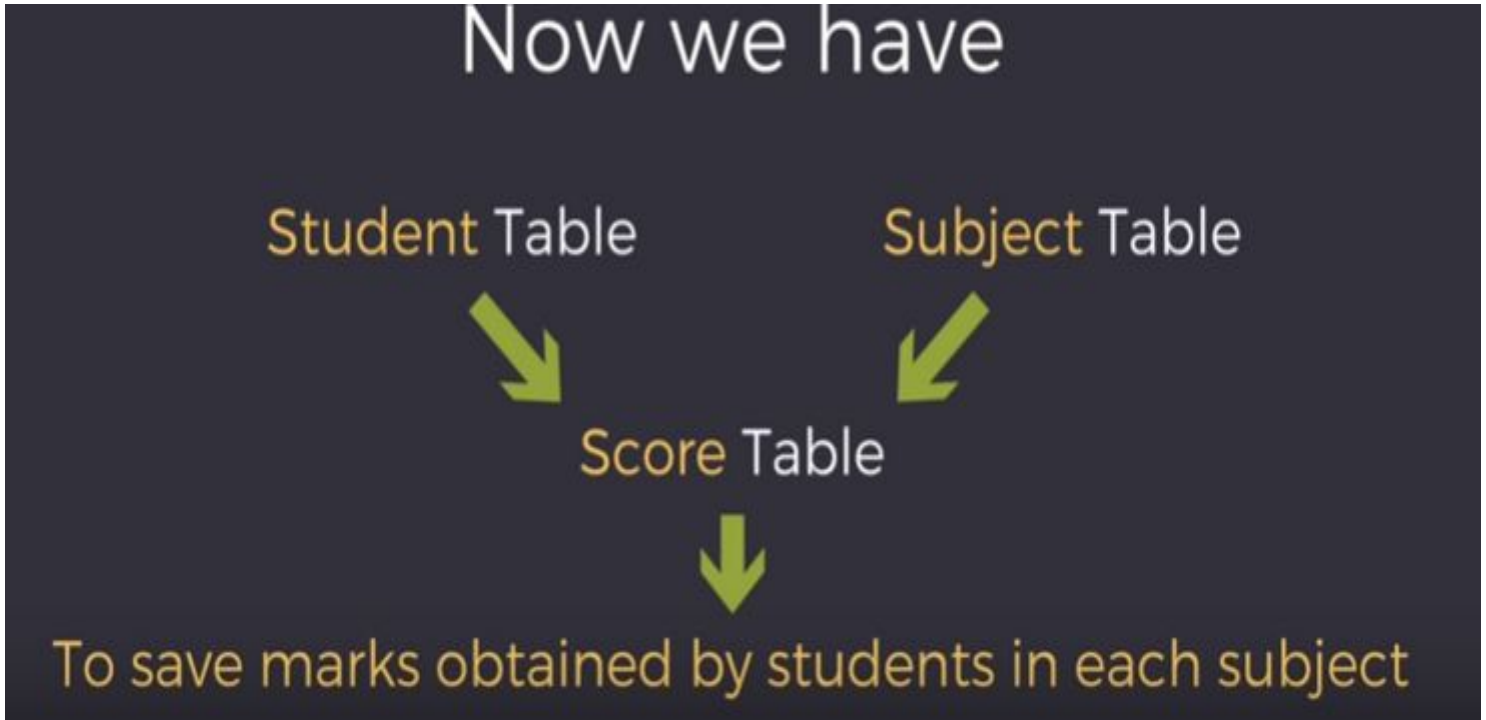
- Even for a case, where student names are same, if we know the student_id we can easily fetch the correct record.

student_id	name	reg_no	branch	address
10	Akon	07-WY	CSE	Kerala
11	Akon	08-WY	IT	Gujarat

- Hence we can say a **Primary Key** for a table is the column or a group of columns(composite key) which can uniquely identify each record in the table.
- his is **Dependency** and we also call it **Functional Dependency**.

2nd Normal Form

- So now let's extend our example to see if more than 1 column together can act as a primary key.
- Let's create another table for **Subject**, which will have subject_id and subject_name fields and subject_id will be the primary key.



2nd Normal Form

Primary Key should be score_id

But student_id + subject_id together makes a more meaningful primary key.

2nd Normal Form

2nd Normal Form

SCORE TABLE

score_id	student_id	subject_id	marks	teacher
1	1	1	82	Mr. J
2	1	2	77	Mr. C++
3	2	1	85	Mr. J
4	2	2	82	Mr. C++
5	2	4	95	Mr. P

2nd Normal Form

SCORE TABLE

score_id	student_id	subject_id	marks	teacher
1	10	1	82	Mr. J
2	10	2	77	Mr. C++
3	11	1	85	Mr. J
4	11	2	82	Mr. C++
5	11	4	95	Mr. P

teacher column only depends on subject and not on student.

This is Partial Dependency

SCORE TABLE

score_id	student_id	subject_id	marks	teacher
1	10	1	82	Mr. J
2	10	2	77	Mr. C++
3	11	1	85	Mr. J
4	11	2	82	Mr. C++
5	11	4	95	Mr. P

SUBJECT TABLE

subject_id	subject_name
1	Java
2	C++
3	C#
4	Php

Move teacher column to Subject Table

SUBJECT TABLE

subject_id	subject_name	teacher
1	Java	Mr. J
2	C++	Mr. C++
3	C#	Mr. C#
4	Php	Mr. P

3rd Normal Form

For a table to be in the third normal form,

- It should be in the Second Normal form.
- And it should not have Transitive Dependency.

3rd Normal Form

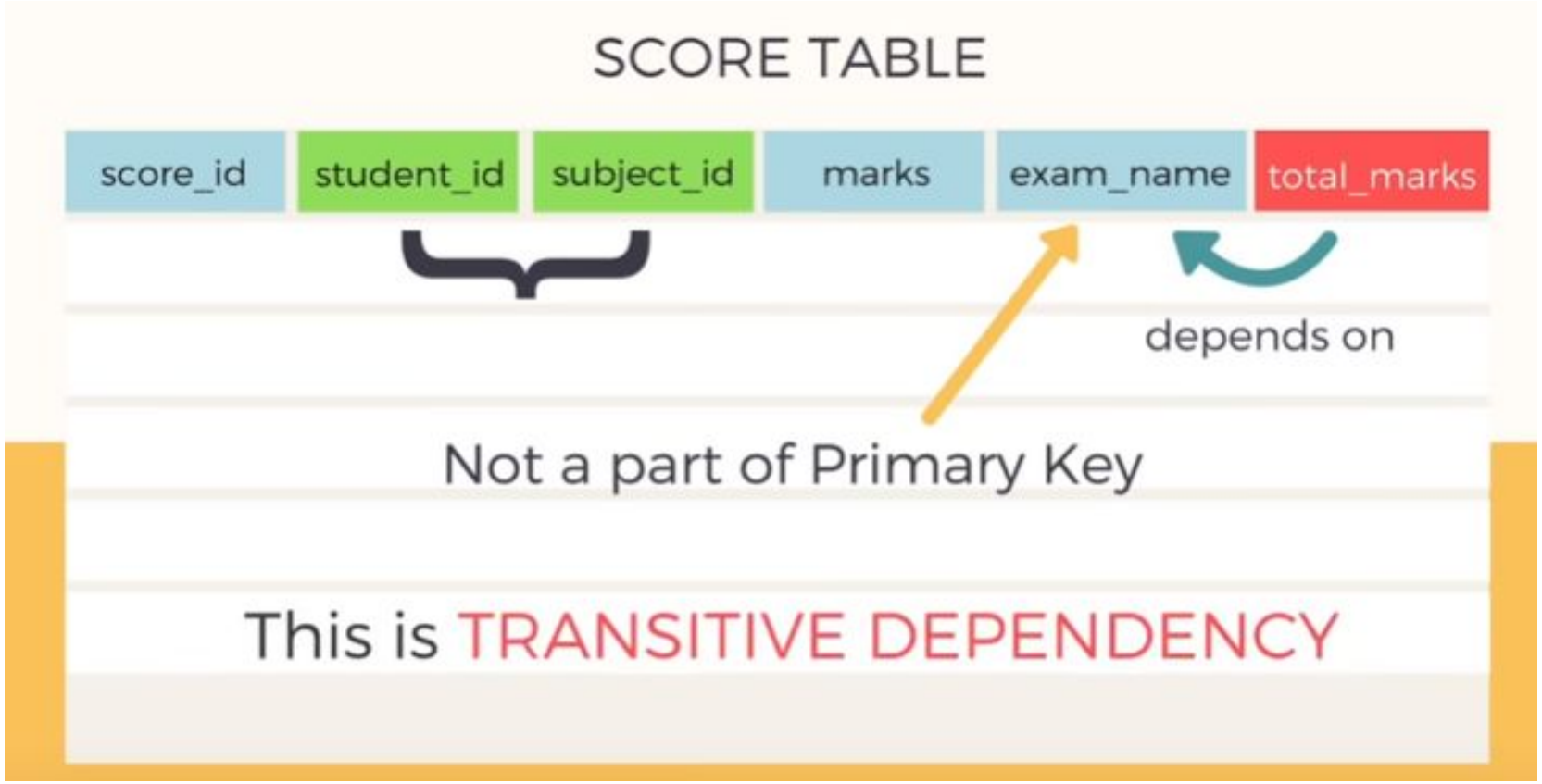
SCORE TABLE

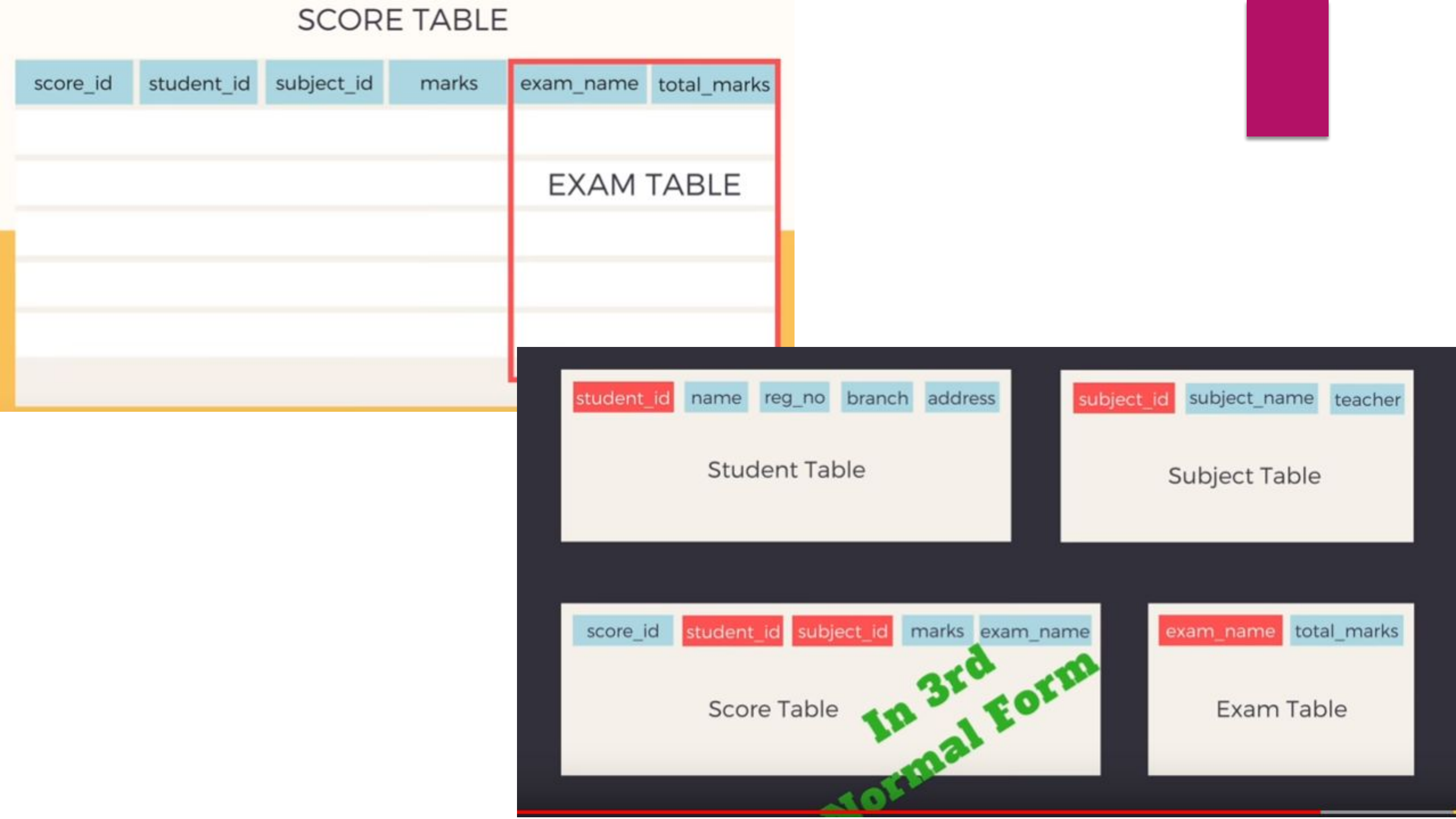
score_id	student_id	subject_id	marks	exam_name	total_marks

Transitive Dependency:

- Primary key for our Score table is a composite key, which means it's made up of two attributes or columns → **student_id + subject_id**.
- Our new column exam_name depends on both student and subject. For example, a mechanical engineering student will have Workshop exam but a computer science student won't. And for some subjects you have Practical exams and for some you don't. So we can say that exam_name is dependent on both student_id and subject_id.
- But, the column total_marks depends on exam_name as with exam type the total score changes. For example, practicals are of less marks while theory exams are of more marks.
- But, exam_name is just another column in the score table. It is not a primary key or even a part of the primary key.
- This is **Transitive Dependency**. When a non-prime attribute depends on other non-prime attributes.

3rd Normal Form




SCORE TABLE
score_id
student_id
subject_id
marks
exam_name
total_marks
EXAM TABLE
student_id
name
reg_no
branch
address
Student Table
subject_id
subject_name
teacher
Subject Table
score_id
student_id
subject_id
marks
exam_name
Score Table
In 3rd
Normal Form
exam_name
total_marks
Exam Table

EXAMPLE TO SOLVE

Supplier	Product	Cost	Markup	Price	Dept Code
21 – Very Veggie	4108 – tomatoes, plum	1.89	5%	1.99	PR
32 – Fab Fruits	4081 – bananas	0.20	75%	0.35	PR
32 – Fab Fruits	4027 – grapefruit	0.45	100%	0.90	PR
32 – Fab Fruits	4851 – celery	1.00	100%	2.00	PR
08 – Meats R Us	331100 – chicken wings	0.50	300%	1.50	BU
08 – Meats R Us	331105 – lean ground beef	0.60	400%	2.40	BU
08 – Meats R Us	332110 – boneless chicken breasts	2.50	100%	5.00	BU
10 – Jerry's Juice	411100 – orange juice	0.25	400%	1.00	FR
10 – Jerry's Juice	521101 – apple juice	0.25	400%	1.00	FR
45 – Icey Creams	866503 – vanilla ice cream	2.50	100%	5.00	FR
45 – Icey Creams	866504 – chocolate ice cream	2.50	100%	5.00	FR

NEXT LECTURE

- BCNF
- 4th Normal Form
- 5th Normal Form