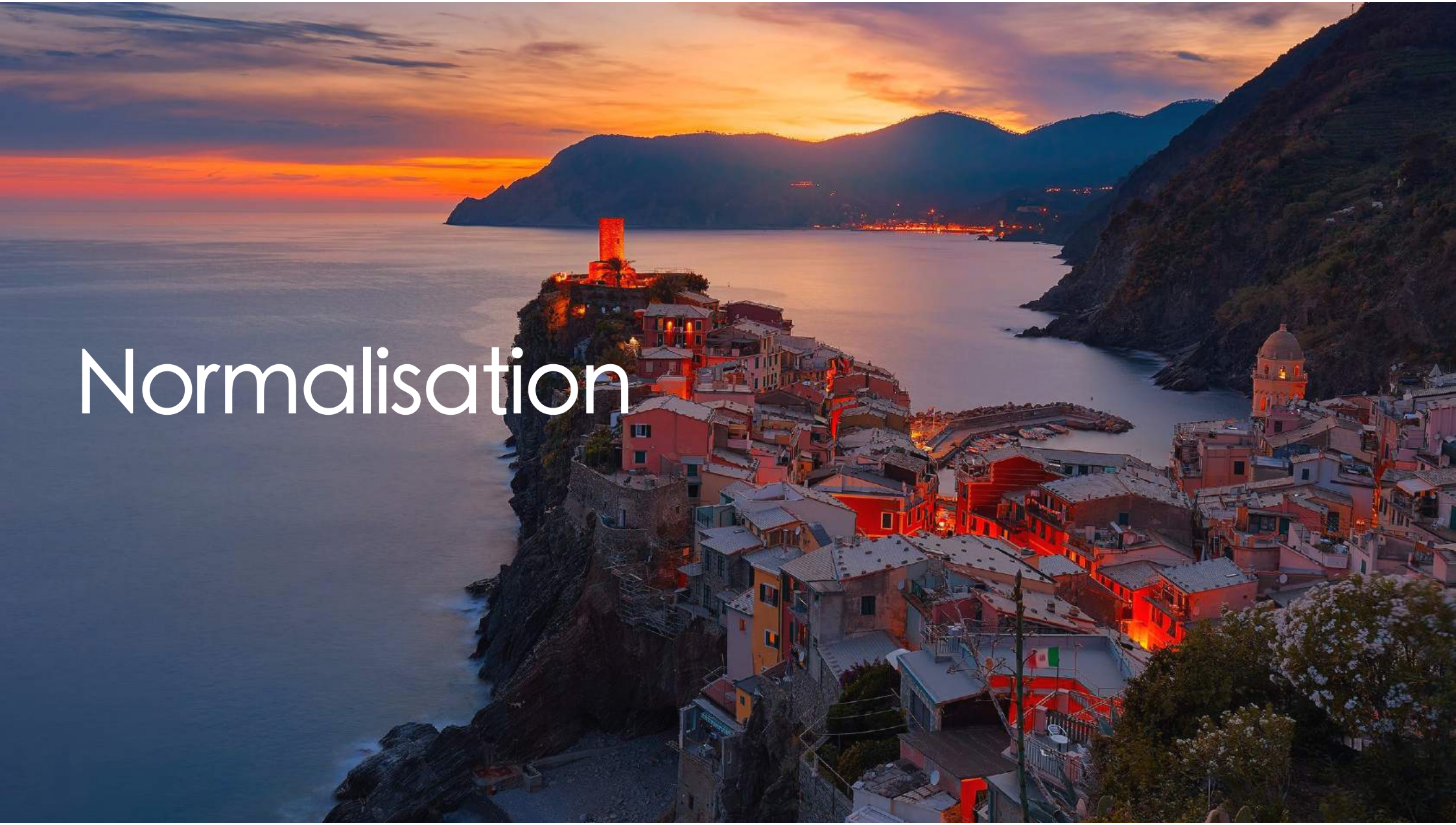
Normalisation



How to design the database?

There are two approaches

- E-R Modeling:
  - Identifying entity and relations
  - Conversion of ER diagram to tables
  - Top down approach
- Normalization:
  - Refinement of database
  - Design database by using tests or normal forms
  - Bottom up approach

Operations on Database

- Insertion
- Updation
- Deletion
- Retrieval

Normalization

- It is a process of decomposing a larger table into smaller tables so that it satisfy series of tests. If the database satisfies the test, then database is considered normalized according to that test or rule or degree, otherwise database is considered un-normalized.
- When a test fails, the relation violating that test must be decomposed into relations so that it individually meet the normalization tests.

Process of Normalization

		APPLY TEST		

Apple Test on a Table.

If the Table passes the test it is normalised. Then we do not need to decompose the table into smaller tables with partitioned attributes(columns).

If the table ideas not pass the test, it is unnormalised and we will keep dividing the table into smaller tables, and each smaller table will be tested until all the decomposed tables are in normalised form, I.e .passes the test.

Process of Normalization

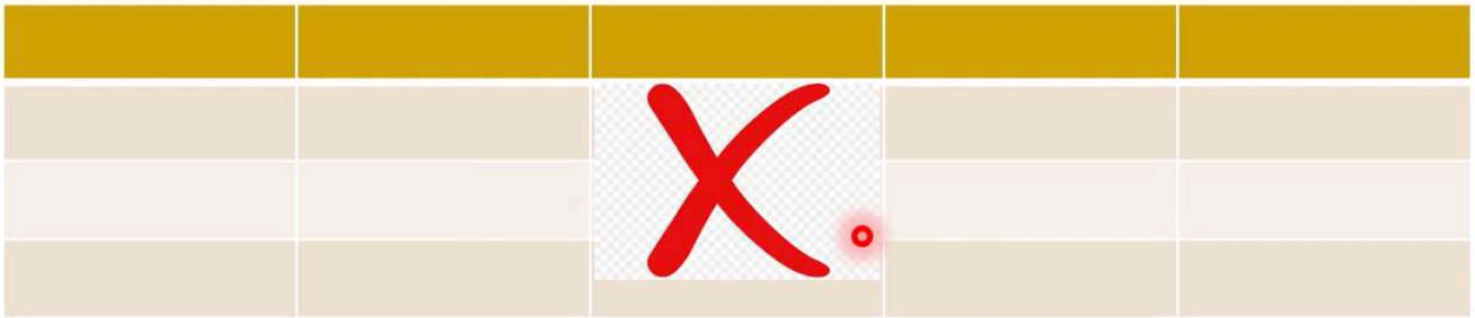
Process of Normalization

	Normalized			

Process of Normalization

	APPLY TEST			

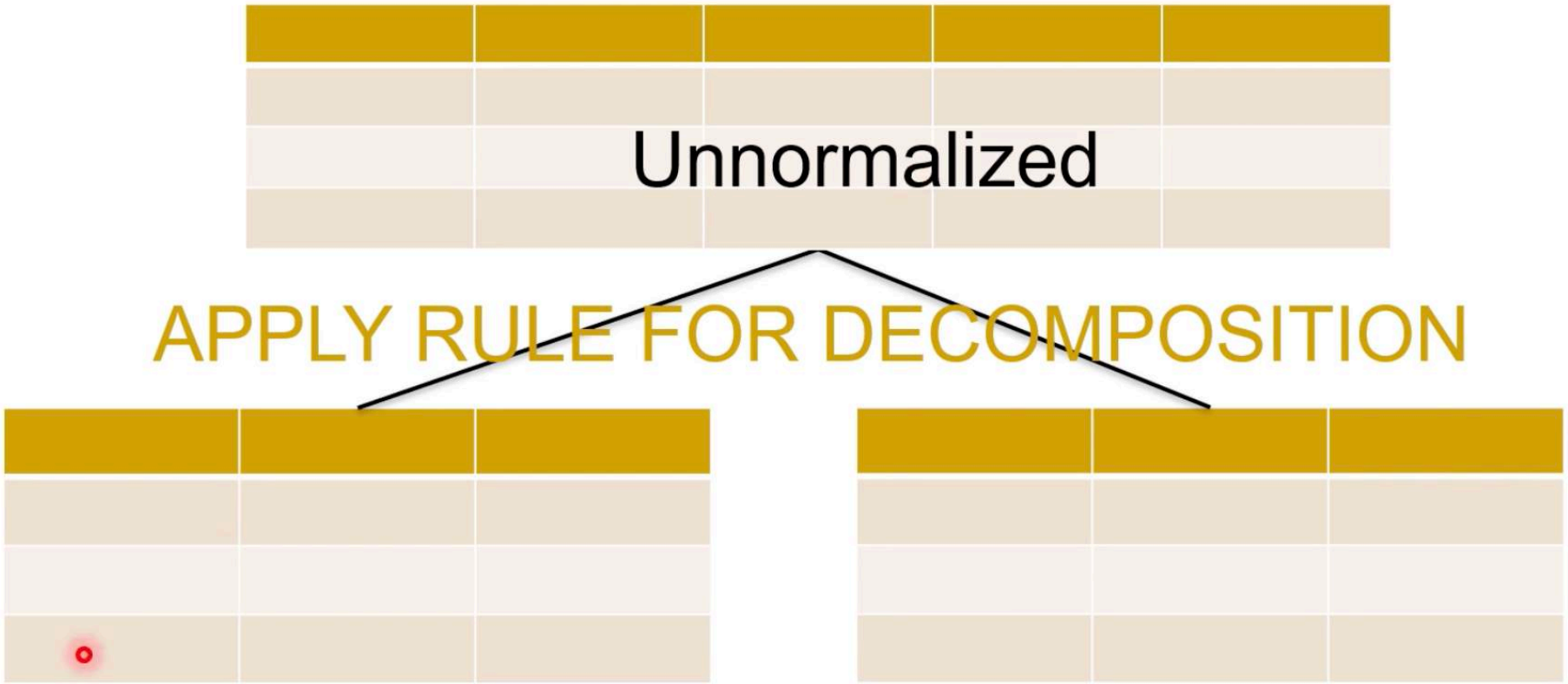
Process of Normalization



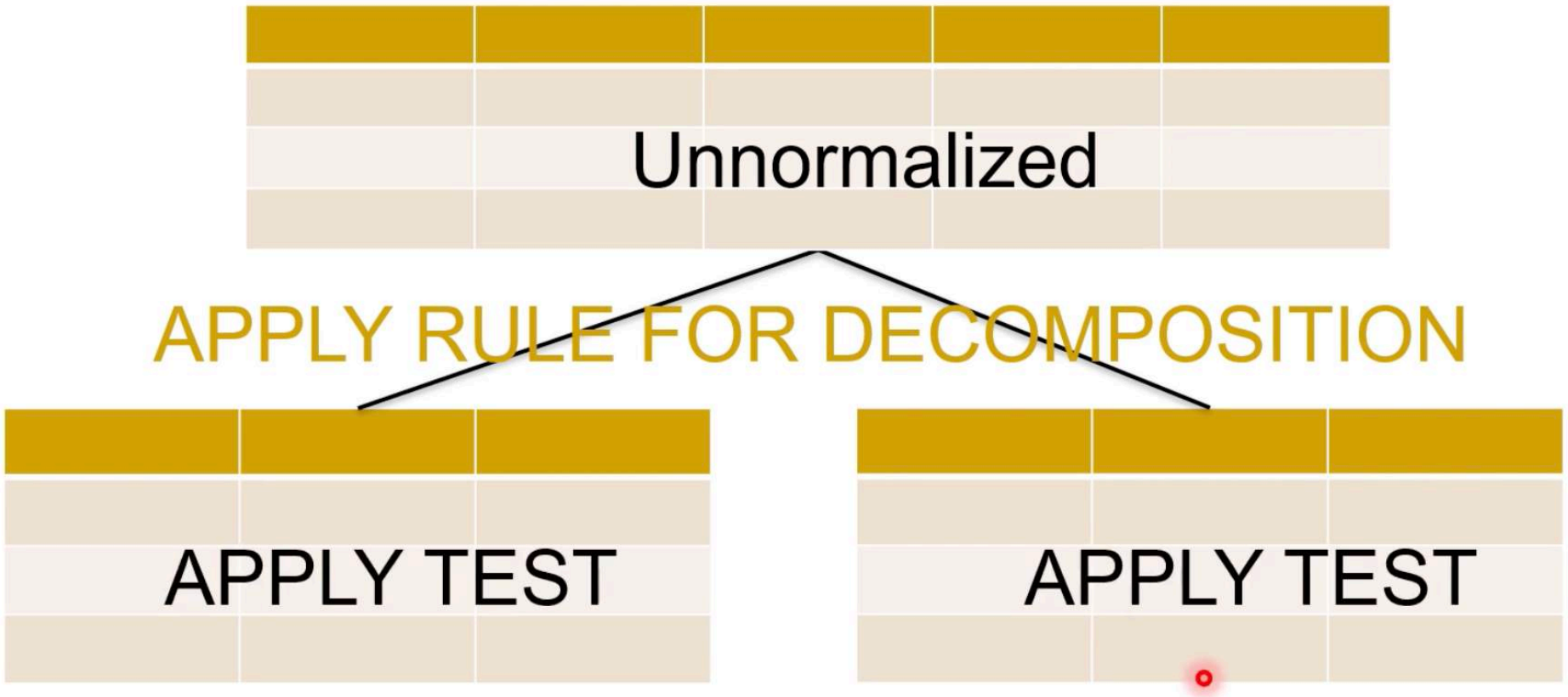
Process of Normalization

	Unnormalized			

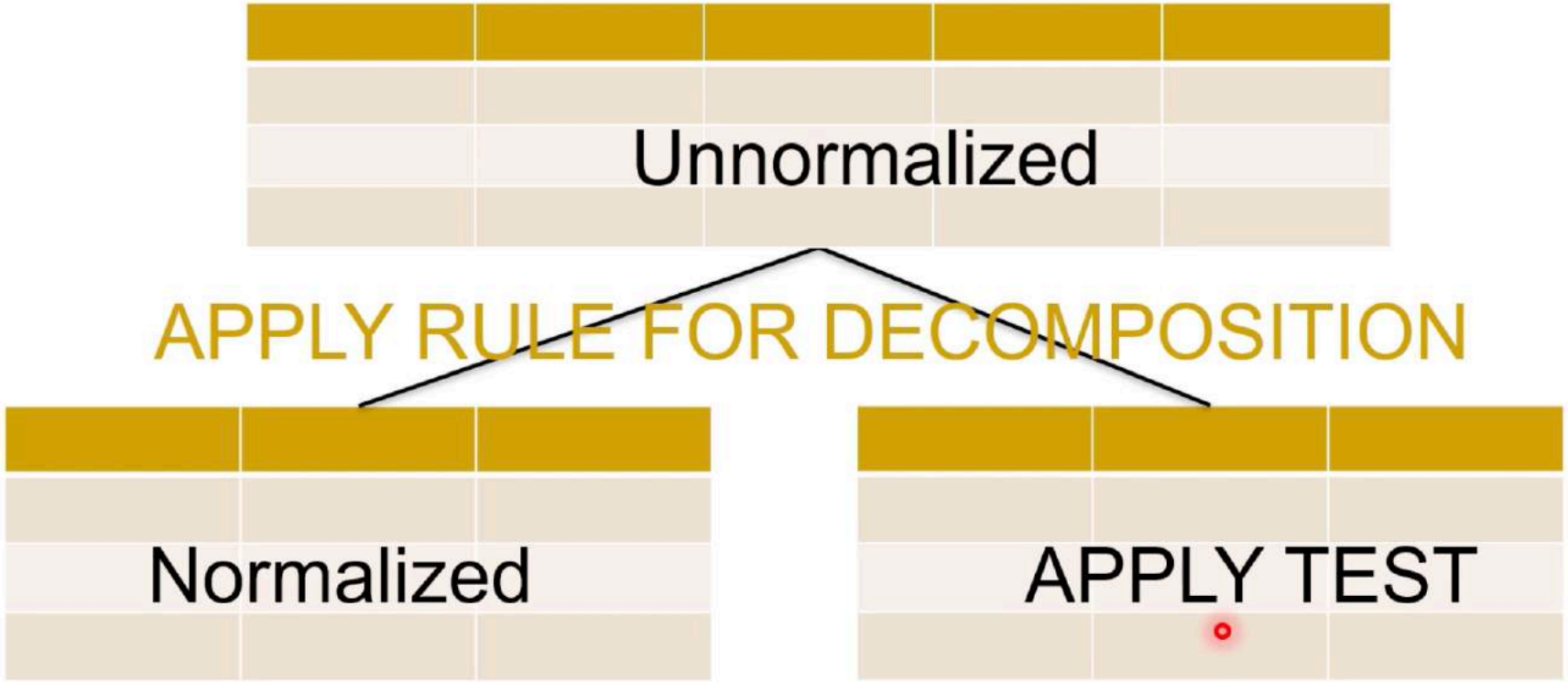
Process of Normalization



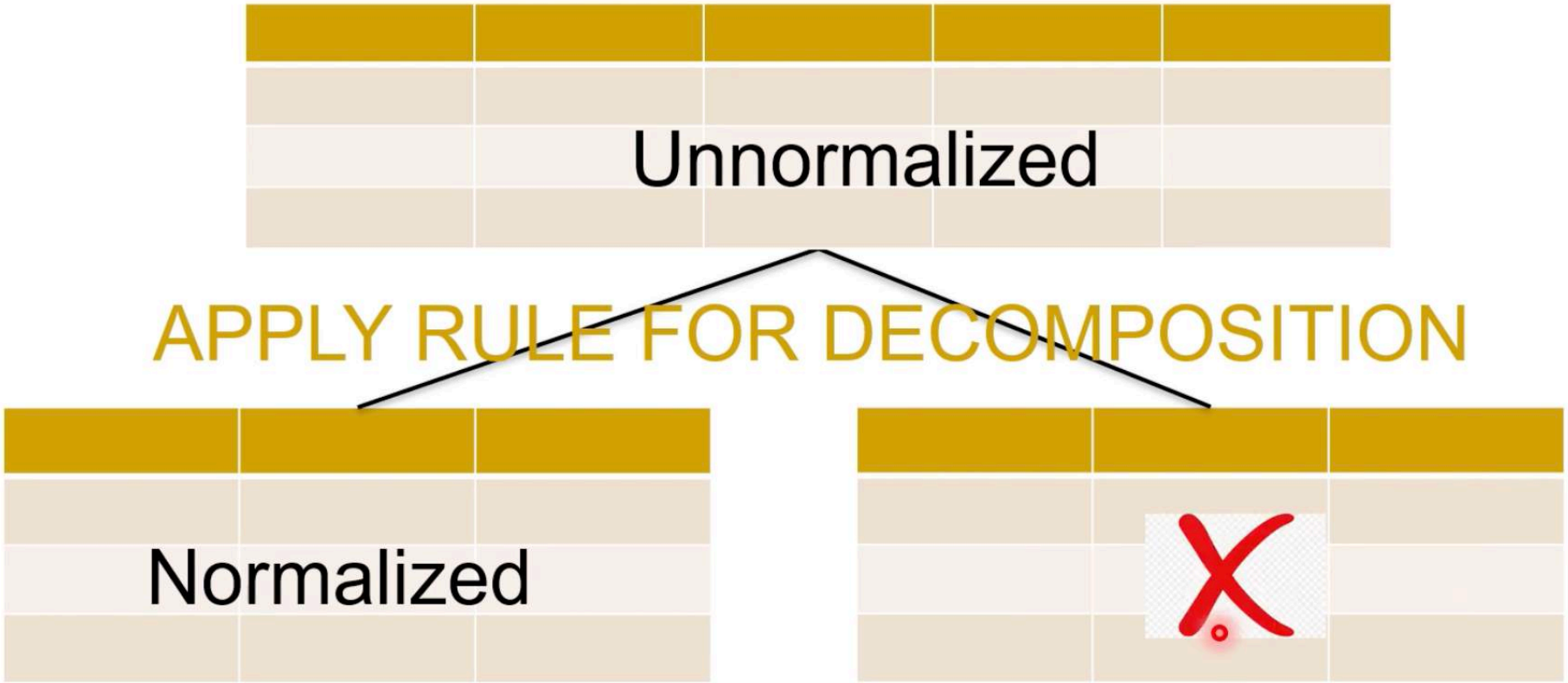
Process of Normalization



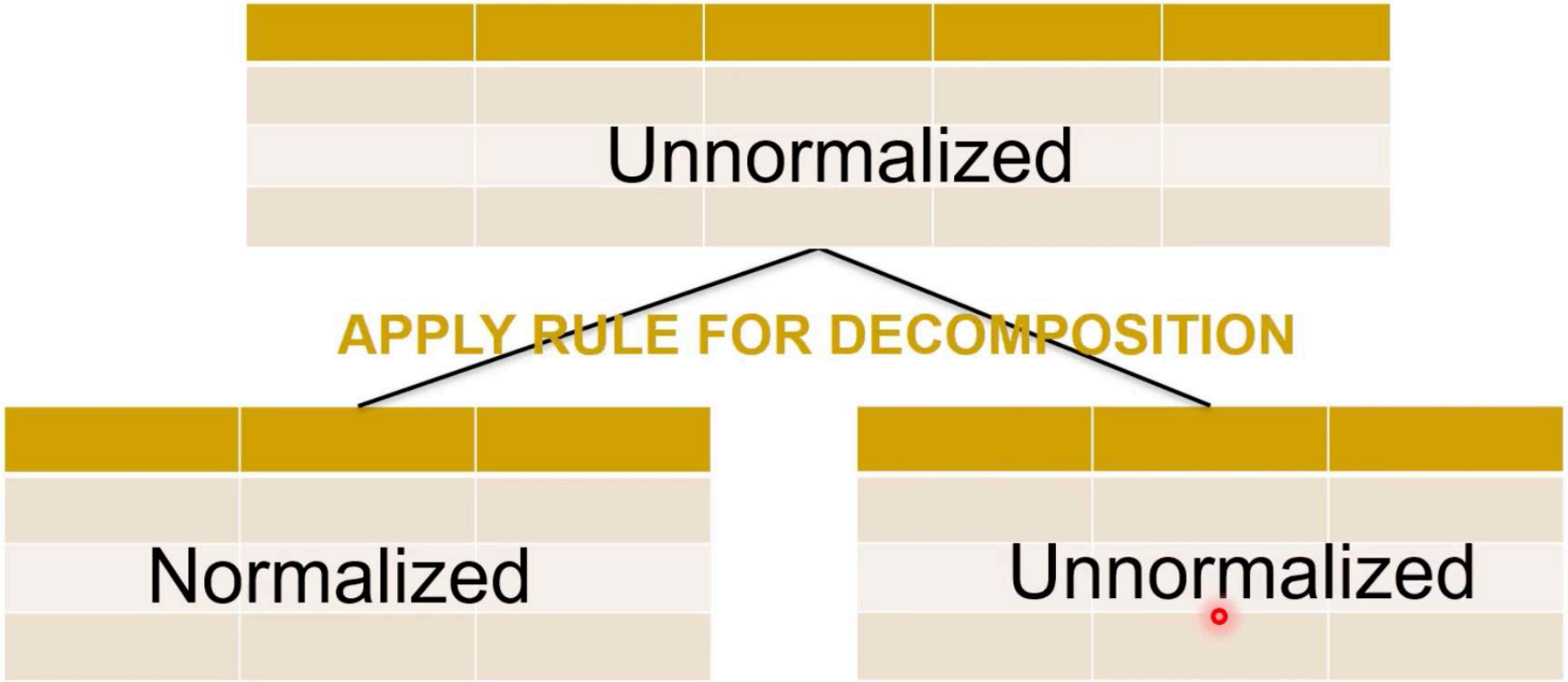
Process of Normalization



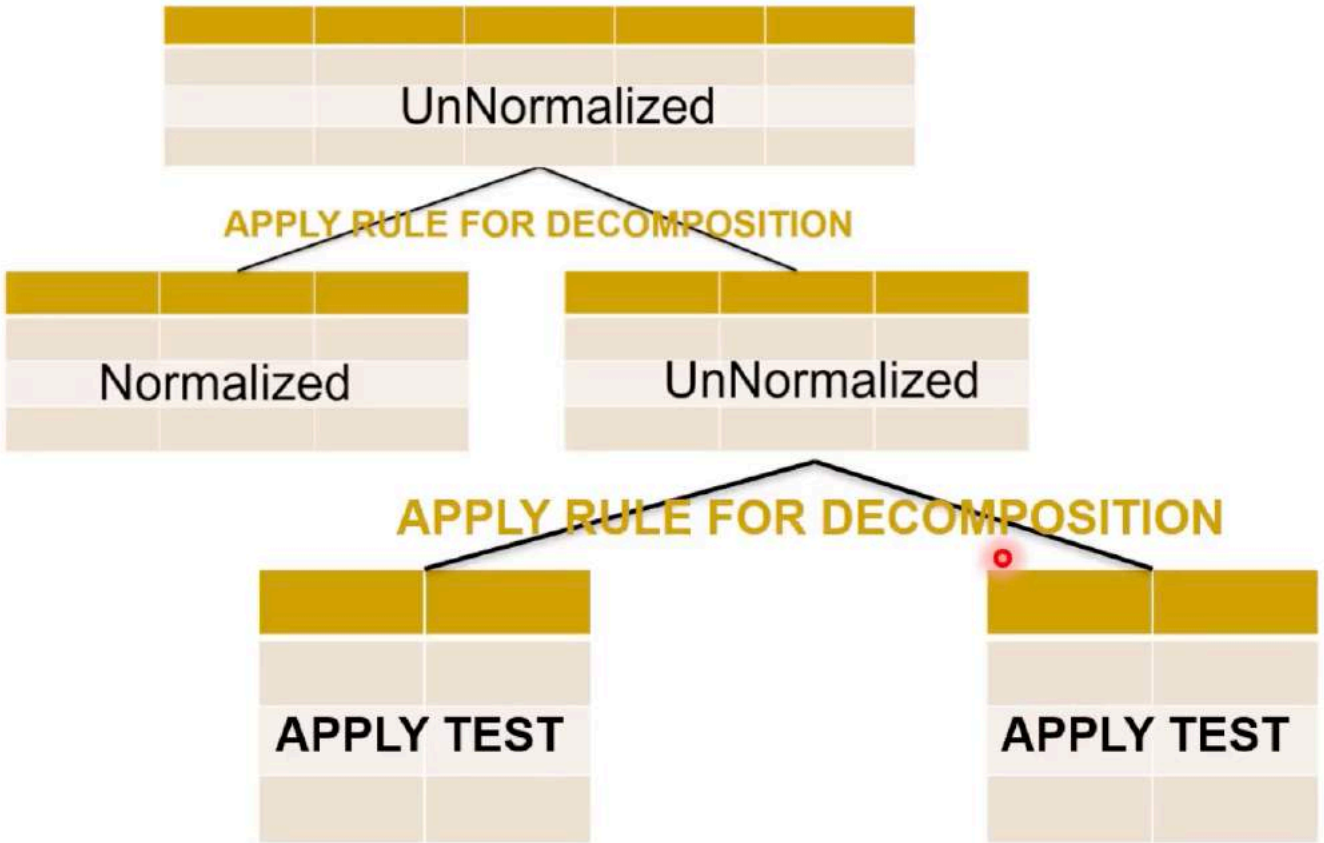
Process of Normalization



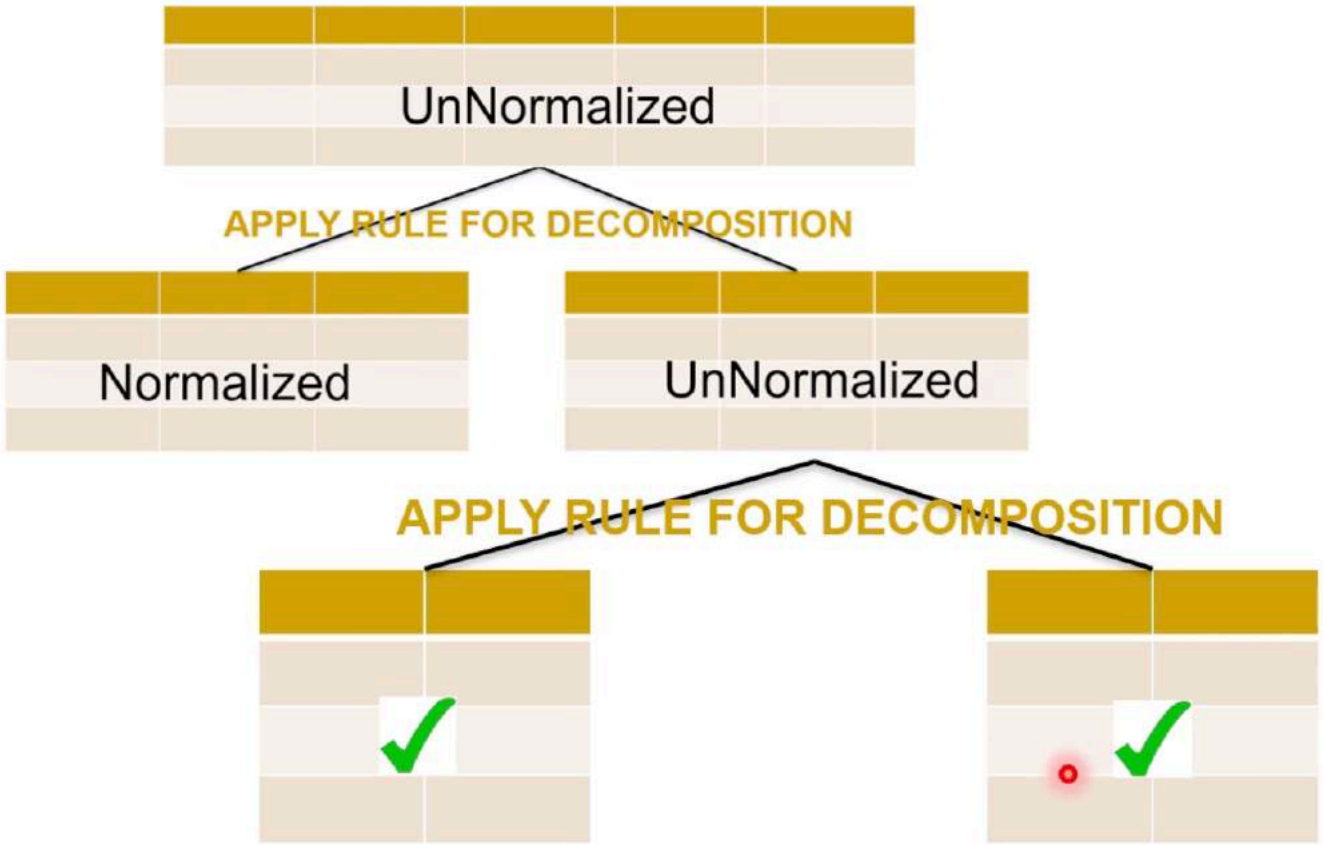
Process of Normalization



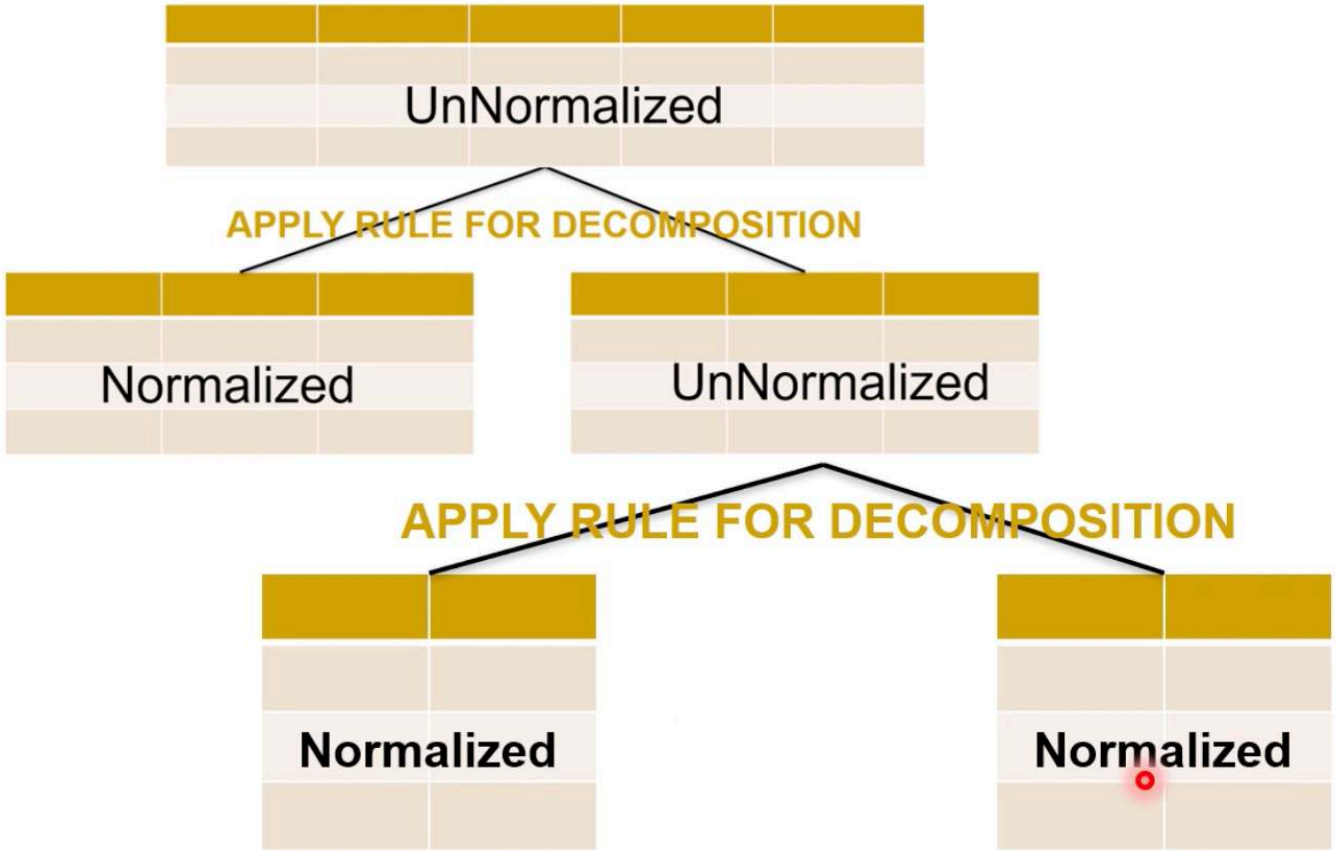
Process of Normalization



Process of Normalization



Process of Normalization



Objective

Objectives of Normalization

- To create a formal framework to verify that designed database is accurate and it ensures that we can perform insertion, update, delete and retrieve operation without any anomalies or issues.
- To have a formal framework normalization process involves series of test to verify that our designed database is accurate.
- Here, we analyze the relations based on their keys and different kind of dependencies among their attributes.

Objectives of Normalization

- To reduce the need for restructuring the relations as new data types are introduced.
- To obtain powerful relational retrieval algorithms.
- In simple words, it assure that our designed database is correct and we can start building our system over it.

Objectives of Normalization

- For this we carry out series of tests on individual relation schema so that the relational database can be normalized to some degree.
- When a test fails, the relation violating that test must be decomposed into relations that individually meet the normalization tests.

Normal Forms

First N/F	Second N/F	Third N/F	Fourth N/F	Fifth N/F
Eliminate Repeating Groups	Non-key attribute Fully Functional Dependence on PK	Eliminate Transitive Dependence on PK	Remove Multi Value Dependeency	Projection Join Dependency

Fully Functional Dependence

Functional Dependence (FD)

- In a relation R having two attributes X and Y and if for each value of X there is only one value of Y, then Y is called functionally dependent on X.
- This is represented as X→Y.
- Here, X is the determinant and Y is the determined or *X functionally determines Y.*
- In other words, *Y is functionally determined by* X.
- **For each value of the determinant there should be associated one and only one value of the determined.**

Normalization

- There are five series of test that we apply on the database, so there are five degree or rules of normalization which are known as Normal Forms.
- For example: First Normal Form, Second Normal Form and so on.

SIMPLIFIED

Functional Dependence (FD): Example

Rno	Name	Class
1	Raj	BE
2	Ramesh	BE
3	Suresh	MCA
4	Rahat	BE
5	Raj	BE
7	Rahat	MCA

Rno → Name ✓

Rno → Class ✓

Name ↛ Rno

Class ↛ Rno

Functional Dependence (FD): Example

A	B
1	1
2	4
3	9
4	16
2	4
7	9

3,7

$A \rightarrow B$ ✓ $B \not\rightarrow A$

For A-> B, A should be unique.

$B \not\rightarrow A$

For B->A, B should be unique.

Functional Dependence (FD): Example

A	B
1	1
2	4
3	9
4	16
3	10

unique

$A \not\rightarrow B$

$B \rightarrow A$

For B->A, B should be unique.

Functional Dependence (FD): Example

SNO	SNAME	CITY	STATUS
S1	RAJ	ASR	10
S2	RAM	ASR	10
S3	RAJ	DEL	20
S4	RISHAN	PTA	10

SNO → SNAME SNAME ↛ SNO
SNO → CITY CITY ↛ SNO
SNO → STATUS STATUS ↛ SNO

Rule: Every primary key of relation functionally determines all other non-key attributes of the relation.

Fully Functional Dependence (FFD)

Fully Functional Dependence(FFD) is defined as Attribute Y is FFD on attribute X, if it is FD on X and not FD on any proper subset of X.

$\underbrace{(X1, X2)}_{X} \rightarrow Y$

$X1 \not\rightarrow Y$

$X2 \not\rightarrow Y$

Fully Functional Dependence (FFD): Example

SNO	PNO	QTY
S1	P1	100
S1	P2	120
S2	P1	100
S2	P2	130

SNO, PNO → QTY

SNO ↛ QTY

PNO ↛ QTY

Fully Functional Dependence (FFD)

SNO	SNAME	CITY	STATUS
S1	RAJ	ASR	10
S2	RAM	ASR	10
S3	RAJ	DEL	20
S4	RISHAN	PTA	10

SNO → SNAME
SNO → CITY
SNO → STATUS

SNAME ↛ SNO
CITY ↛ SNO
STATUS ↛ SNO

Is SNO, SNAME → CITY is FFD?
SNO, SNAME → CITY
SNO → CITY
SNAME ↛ CITY
SNO, SNAME → CITY is not FFD.

Understanding DataSet

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every Course_Code there is one Course_Name
- For every Course_Code there is one Teacher_Name

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every Course_Code there are many students or RollNo
- For every Course_Code there are many values of Name

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every Course_Code there are many values of System_Used
- For every Course_Code there are many values of Hourly_Rate, Total_Hrs

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every RollNo there is only one value of Name
- For every RollNo there is only one value of System_Used

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every RollNo there is only one value of Hourly_Rate
- For every Coure_Code there are multiple values of Total_Hrs

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- For every Course_Code and RollNo combination there is only one value of Total_Hrs

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- **Non-Repeating Group**

Course_Code, Course_Name and Teacher_Name

- **Repeating Group**

RollNo, Name, System_Used, Hourly_Rate and Total_Hrs

First Normal Form

First Normal Form

Definition of First Normal Form

- A relation is said to be in First Normal Form (1NF) if and only if every entry of the relation (the intersection of a tuple and a column) has at most a single value.
- In other words "a relation is in First Normal Form if and only if all underlying domains contain atomic values or single value only."
- Here, Domain refers to all the valid values which a data element (column) may contain.

First Normal Form

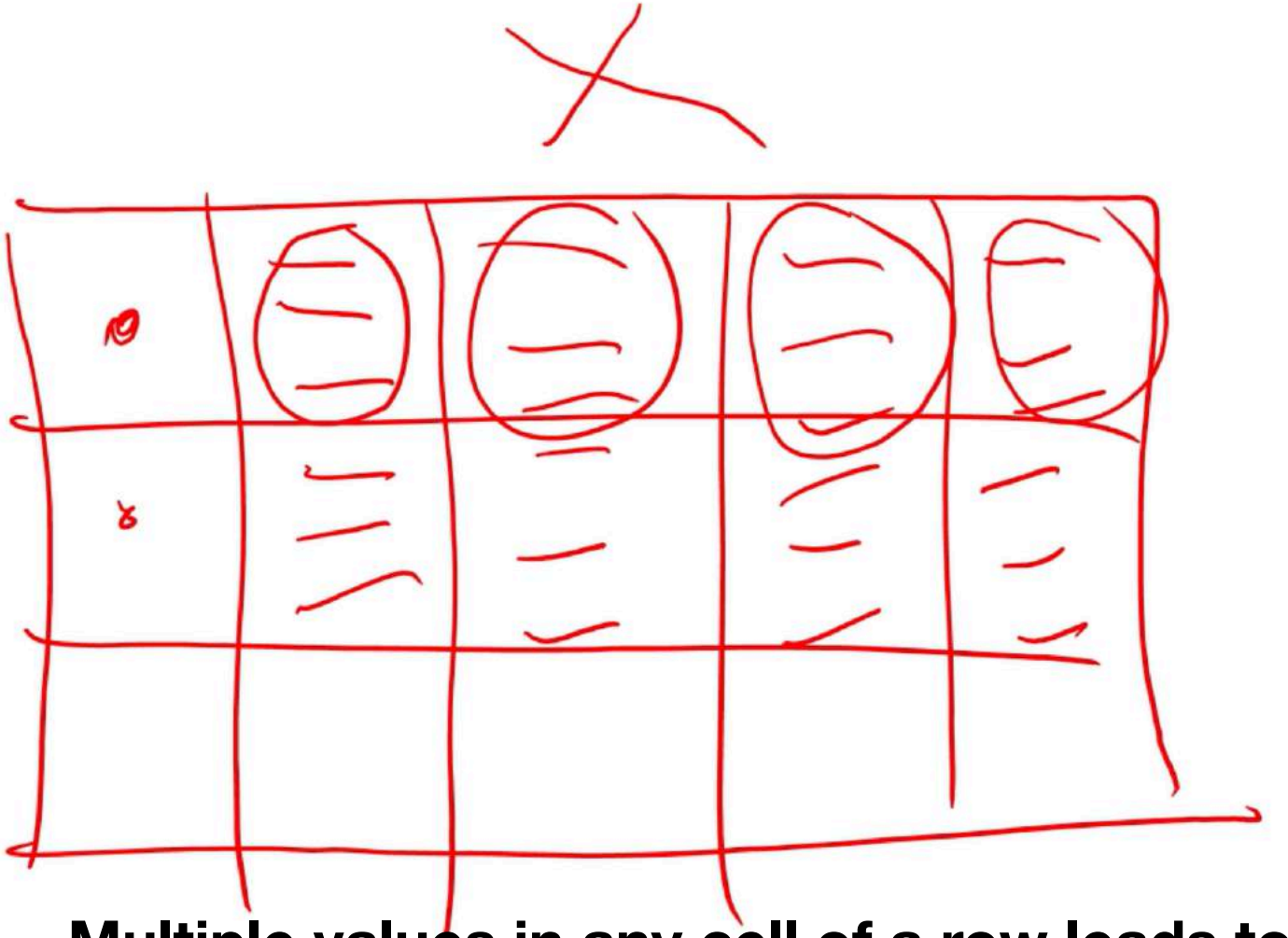
STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

RDBMS

	Null			Null

Every row should have almost 1 cell



Multiple values in any cell of a row leads to Unnormalised table.

First Normal Form

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-I	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

First Normal Form

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- A relation is said to be in First Normal Form (1NF) if and only if every entry of the relation (the intersection of a tuple and a column) has at most a single value. In other words "a relation is in First Normal Form if and only if all underlying domains contain atomic values or single value only."

First Approach: Flattening the table

- The first approach known as “flattening the table” removes repeating groups by filling in the “missing” entries of each “incomplete row” of the table with copies of their corresponding non-repeating attributes.

Understanding Dataset

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- **Non-Repeating Group**

Course_Code, Course_Name and Teacher_Name

- **Repeating Group**

RollNo, Name, System_Used, Hourly_Rate and Total_Hrs

First Approach: Flattening the table

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

First Approach: Flattening the table

This is normalised table,

Primary key is combination of Course Code and Roll no.

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Second Approach: Decomposition of the table

- The second approach for normalizing a table requires that the table be decomposed into two new tables that will replace the original table.
- However, before decomposing the original table it is necessary to identify an attribute or a set of its attributes that can be used as table identifiers.

Rule of decomposition

- One Table

Create a relation with Table identifier of the original table and all the non-repeating attributes.

- Second Table

The other table contains a copy of the table identifier and all the repeating attributes.

Unnormalised table does not have primary key. So, the so called primary key in unnormalised table is called Table identifier

Course Code is Table Identifier.

STUDENT (Unnormalized table)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
			101	A2	P-II	30	3
			102	A3	Celeron	10	6
			103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
			104	A5	P-III	35	3
			105	A6	P-II	30	1
			101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
			107	A8	P-IV	40	2
			108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Course Code is primary key

COURSE

Course_Code	Course_Name	Teacher_Name
C1	Visual Basic	ABC
C2	Oracle&Dev	DEF
C3	C++	KJP
C4	Java	Kumar

Course Code and Roll No.is primary key.
Course Code is foreign key.

COURSE_STUDENT

Course_ Code	Rollno	Name	System_Used	Hourly_Rate	Total_Hrs
C1	100	A1	P-I	20	7
C1	101	A2	P-II	30	3
C1	102	A3	Celeron	10	6
C1	103	A4	P-IV	40	1
C2	100	A1	P-I	20	7
C2	104	A5	P-III	35	3
C2	105	A6	P-II	30	1
C2	101	A2	P-II	30	2
C3	106	A7	P-IV	40	3
C3	107	A8	P-IV	40	2
C3	108	A9	P-I	20	1
C4	109	A10	Cyrix	20	2

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

PK is combination of Course_Code and RollNo.
It means Course_Code and RollNo cannot be null during insertion of a record.

Anomalies of First Normal Form: Insert

PK

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2
C5	SQL	PBH	NULL	NULL	NULL	NULL	NULL

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Insert Anomaly

Cannot insert the data of Course without Student.

In simple words, cannot insert the data of a course having no student.

Anomalies of First Normal Form: Insert

PK

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2
NULL	NULL	NULL	110	A11	P-I	20	2

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Insert Anomaly

Cannot insert the data of Student without Course.

In simple words, cannot insert the data of student who has not registered in any course.

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Update Anomaly

Change in Course information requires multiple changes.

For example, to change the name of Course_Code C1 it requires multiple update operations, which is equal to the number of students enrolled in that course.

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Update Anomaly

Change in Student information requires multiple changes.

For example, to change the name of RollNo 100 it requires multiple update operations, which is equal to the number of courses enrolled by that student.

Anomalies of First Normal Form: Delete

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
~~C1~~	~~Visual Basic~~	~~ABC~~	~~100~~	~~A1~~	~~P-I~~	~~20~~	~~7~~
~~C1~~	~~Visual Basic~~	~~ABC~~	~~101~~	~~A2~~	~~P-II~~	~~30~~	~~3~~
~~C1~~	~~Visual Basic~~	~~ABC~~	~~102~~	~~A3~~	~~Celeron~~	~~10~~	~~6~~
~~C1~~	~~Visual Basic~~	~~ABC~~	~~103~~	~~A4~~	~~P-IV~~	~~40~~	~~1~~
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Delete Anomaly

Deletion of course information causes deletion of student information too.

This problem is worst for a student who has enrolled in only one course and that course has been deleted.

Anomalies of First Normal Form: Delete

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
~~C4~~	~~Java~~	~~Kumar~~	~~109~~	~~A10~~	~~Cyrix~~	~~20~~	~~2~~

Anomalies of First Normal Form

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

- Delete Anomaly

Deletion of student information causes deletion of course information too.

This problem is worst for a course which has only one student enrolled in that course and that student information has been deleted.

Second Normal Form

Solution to Solve Anomalies of First Normal Form

To apply Second Form

Second Normal Form

- A relation R is in second normal form (2NF) if and only if it is in 1NF and every non-key attribute is fully functional dependent on the primary key.

Fully Functional Dependence (FFD)

Fully Functional Dependence(FFD) is defined as Attribute Y is FFD on attribute X, if it is FD on X and not FD on any proper subset of X.

$$\underbrace{(X1, X2)}_{X} \rightarrow Y$$

$$X1 \not\rightarrow Y$$

$$X2 \not\rightarrow Y$$

Fully Functional Dependence (FFD): Example

PK

SNO	PNO	QTY
S1	P1	100
S1	P2	120
S2	P1	100
S2	P2	130

SNO, PNO → QTY

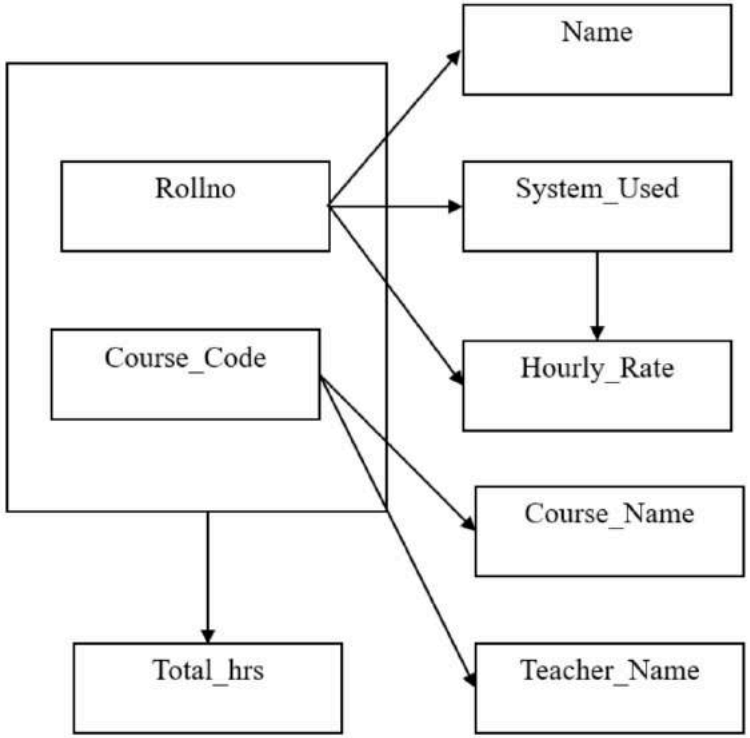
SNO ↛ QTY

PNO ↛ QTY

Process of Conversion to 2NF

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2



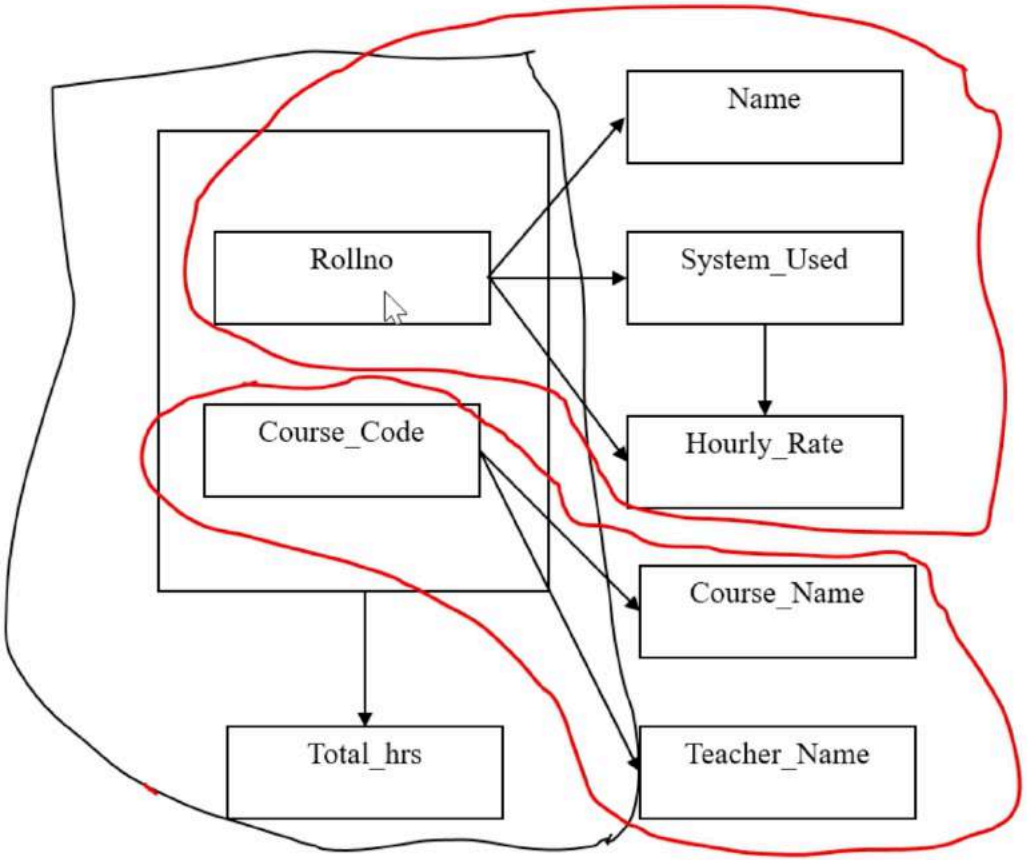
All the non key attributes should be functionally dependent on both the key attributes {Course Code, Roll No} and not on any of them when single.

However, we see that the Course name and teacher Name are dependent on Course Code and Name, System and Rate depends on Roll No,

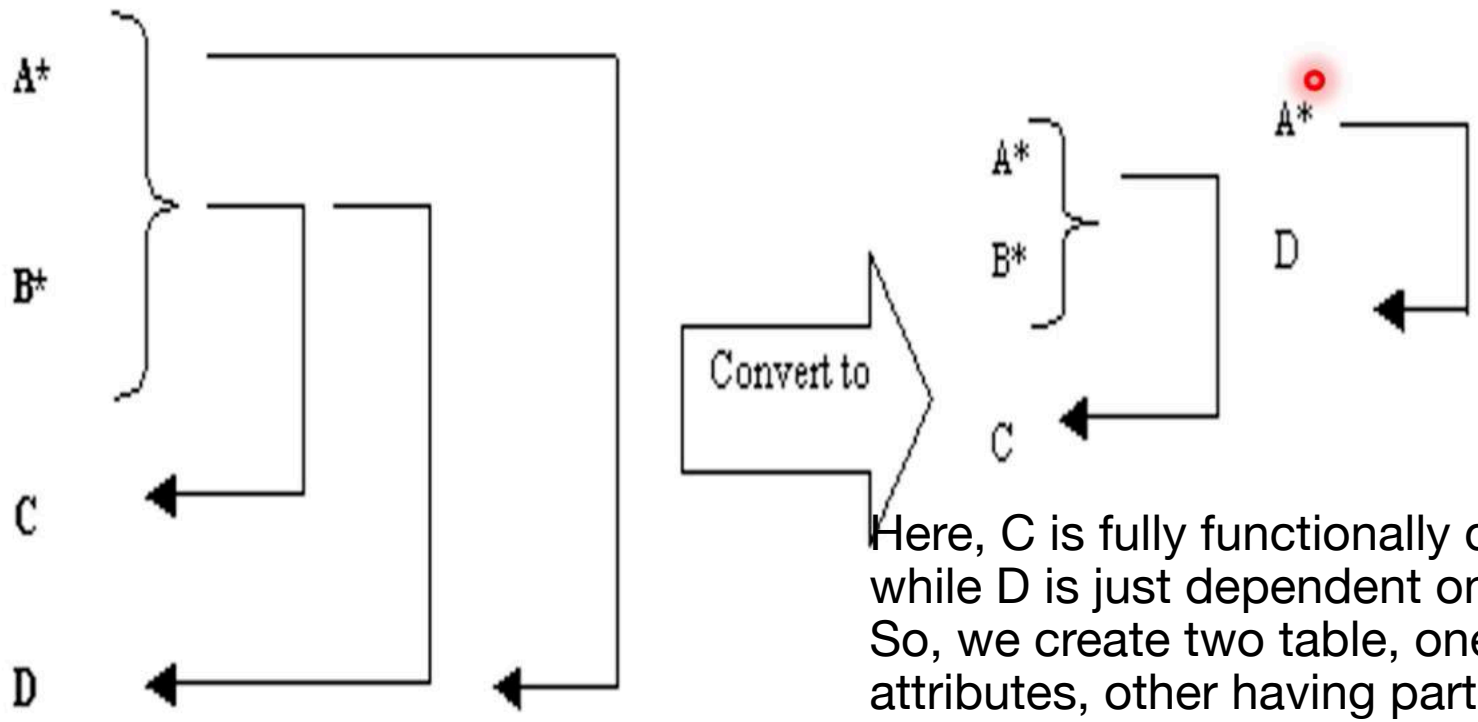
Only total hour depends on both of them together.

Hence, this table is not in second normal form.

Rule of Decomposition for 2NF



Rule of Decomposition for 2NF



Here, C is fully functionally dependent , while D is just dependent on A*.
So, we create two table, one having fed attributes, other having partially dependent attributes with the keys they are dependent on. (A* means PK)

Source Table

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Primary key

STUDENT_SYSTEM_CHARGE

RollNo	Name	System_Used	Hourly_ Rate
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
100	A1	P-I	20
104	A5	P-III	35
105	A6	P-II	30
101	A2	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40

Foreign key

Foreign key

HOURS_ASSIGNED

Course_Code	RollNo	Total__Hrs
C1	100	7
C1	101	3
C1	102	6
C1	103	1
C2	100	7
C2	104	3
C2	105	1
C2	101	2
C3	106	3
C3	107	2
C3	108	1
C4	109	2

Primary key

COURSE

Course_Code	Course_Name	Teacher_Name
C1	Visual Basic	ABC
C2	Oracle&Dev	DEF
C3	C++	KJP
C4	Java	Kumar

Primary key both Course Code and Roll No.

All the anomalies of 1 NF is removed.

We can insert data of student without course or course without student in respective table.
New courses and students can be linked in respective table and hours assigned table.

Any update on course or roll no has to be performed only once and not multiple times.

Any delete operation would not lead to loss of data. Delete of course with a student not registered in any other course won't lead to loss of data of that student. Moreover, delete operation will be performed in respective table and hours assigned table.

Anomalies in STUDENT_SYSTEM_CHARGE Table

STUDENT_SYSTEM_CHARGE			
Rollno	Name	System_Used	Hourly_Rate
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20

Anomalies in STUDENT_SYSTEM_CHARGE Table

STUDENT_SYSTEM_CHARGE			
Rollno	**Name**	**System_Used**	**Hourly_Rate**
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20
		Laptop	40

- Insert Anomaly
- Since Rollno is PK, so cannot insert the data of System_Used without Student
- In Simple words, cannot insert the Hourly_rate of Laptop as there is no student is using System Laptop.

Anomalies in STUDENT_SYSTEM_CHARGE Table

STUDENT_SYSTEM_CHARGE			
Rollno	**Name**	**System_Used**	**Hourly_Rate**
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20

- Update Anomaly
- For example, if we have to change the Hourly_Rate of System P-I from 20 to 30 we have to make multiple changes, which is equal to the number of students using P-I

Anomalies in STUDENT_SYSTEM_CHARGE Table

STUDENT_SYSTEM_CHARGE			
Rollno	**Name**	**System_Used**	**Hourly_Rate**
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20

- Delete Anomaly
- The deletion of Student data causes the deletion of System data too.
- This problem is worst for a system which is used by only one student.

Third Normal Form

Solution to Solve these anomalies

- To apply third normal form.

STUDENT_SYSTEM_CHARGE			
Rollno	**Name**	**System_Used**	**Hourly_Rate**
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20

Third Normal Form

A relation R is in Third Normal Form (3NF) if and only if the following conditions are satisfied simultaneously:

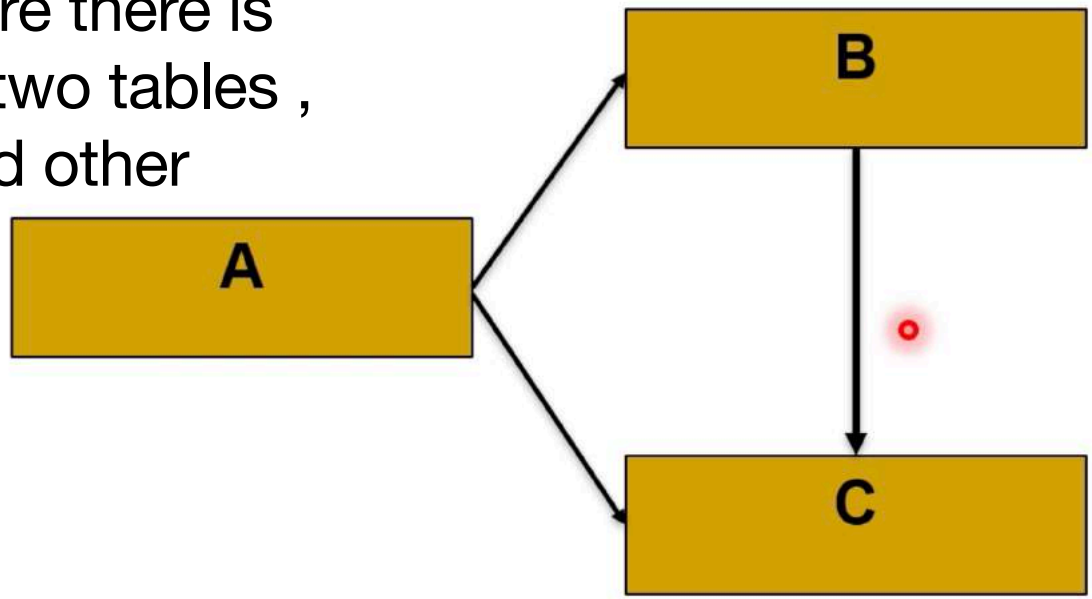
- R is already in 2NF
- No nonprime attribute is transitively dependent on the key.

Another way of expressing the conditions for Third Normal Form is as follows:

- R is already in 2NF
- No nonprime attribute functionally determines any other nonprime attribute.

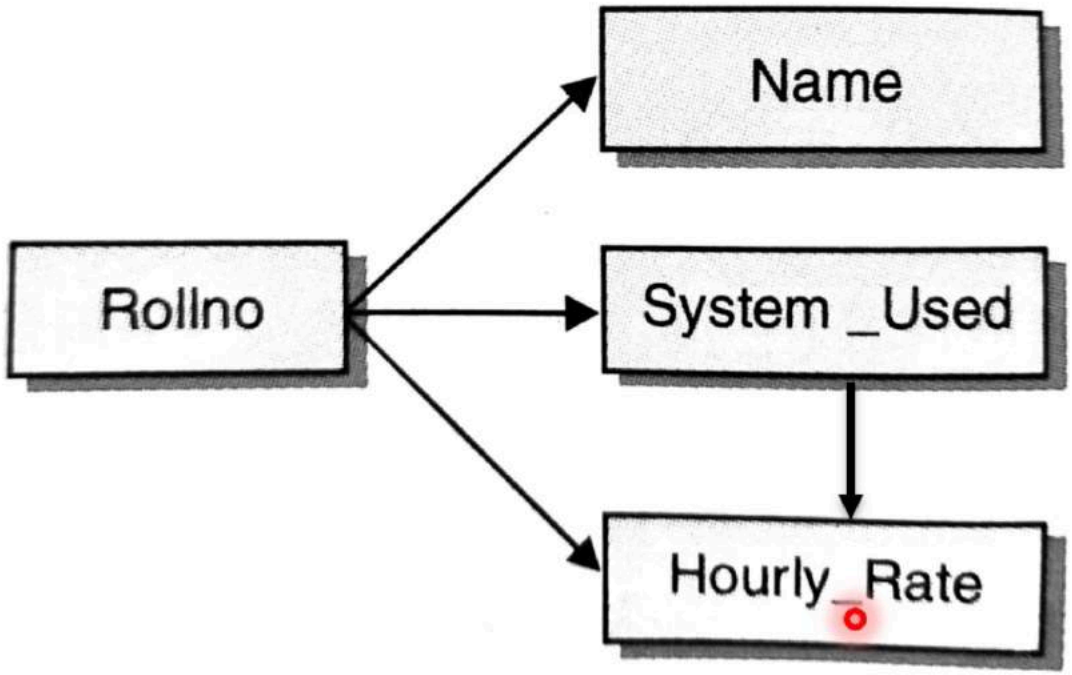
Concept of Transitive Dependence

In such a case, where there is an anomaly, create two tables , one being {A*,B} and other being {B*, C}
* - PK

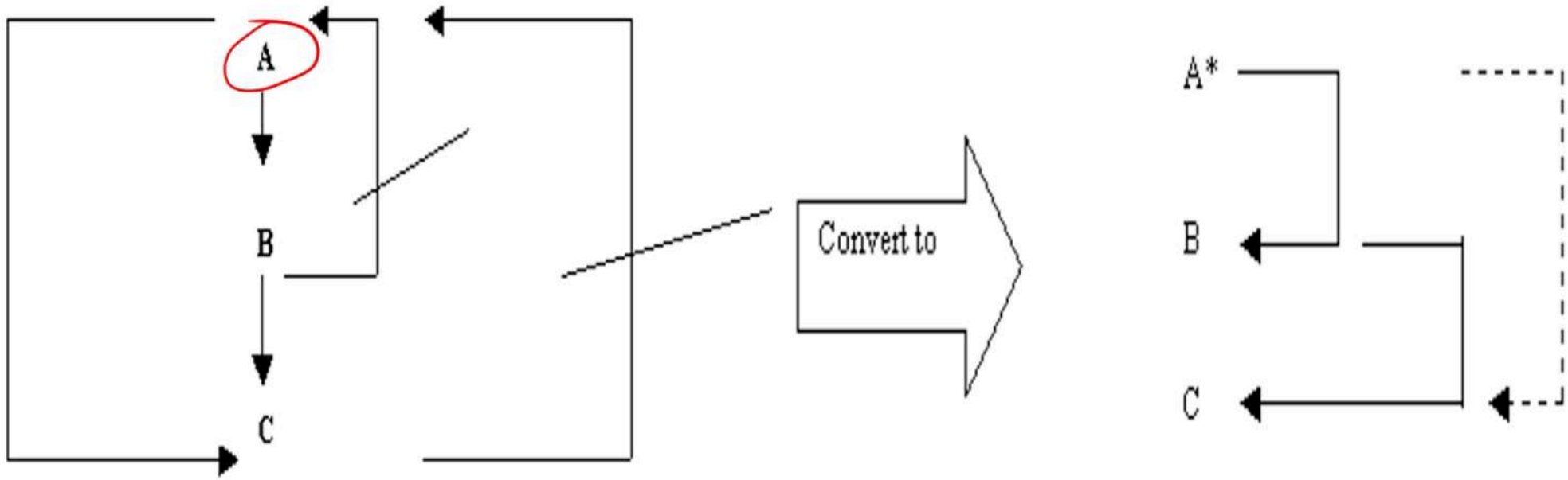


- C is dependent on PK A two ways: one directly and other indirectly through B.
- This indirect path of between A and C through B causes Transitive Dependence of C on A.

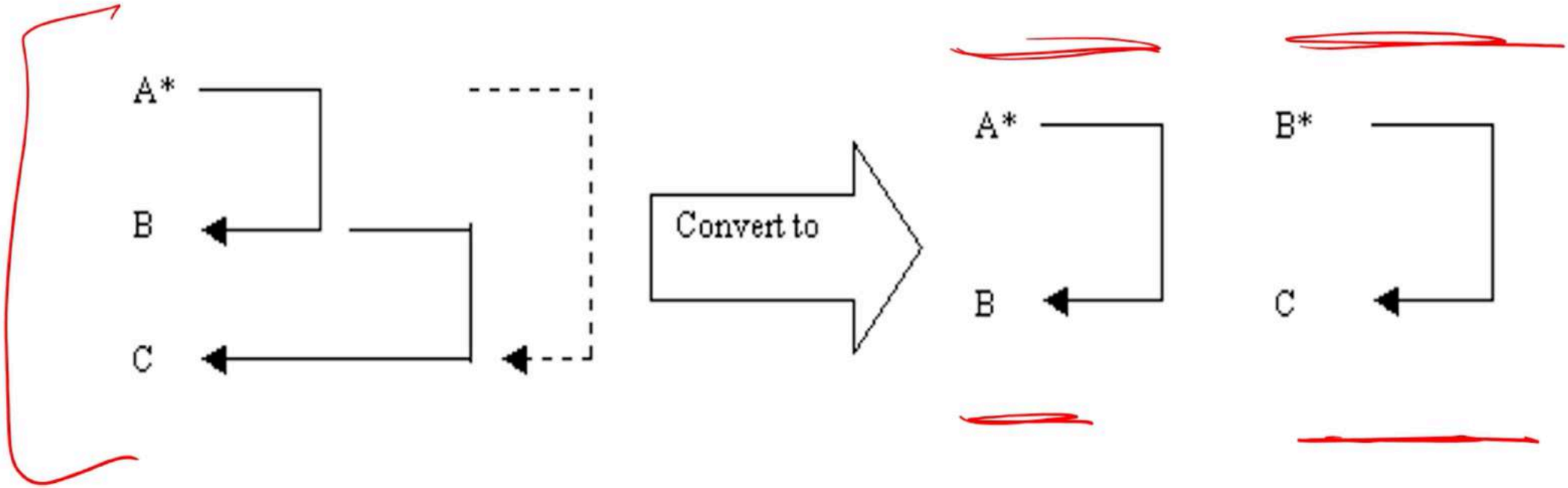
Concept of Transitive Dependence



Concept of Transitive Dependence



Role to Convert a Relation to Third Normal Form



STUDENT_SYSTEM_CHARGE			
Rollno	Name	System_Used	Hourly_Rate
100	A1	P-I	20
101	A2	P-II	30
102	A3	Celeron	10
103	A4	P-IV	40
104	A5	P-II	30
106	A7	P-IV	40
107	A8	P-IV	40
108	A9	P-I	20
109	A10	Cyrix	20

Primary key STUDENT_STSTEM

Rollno	Name	System_Used
100	A1	P-I
101	A2	P-II
102	A3	Celeron
103	A4	P-IV
100	A1	P-I
104	A5	P-III
105	A6	P-II
101	A2	P-II
106	A7	P-IV
107	A8	P-IV
108	A9	P-I
109	A10	Cyrix

Primary key CHARGES

System_Used	Hourly_ Rate
P-I	20
P-II	30
Celeron	10
P-IV	40
P-III	35
Cyrix	20

Foreign key

All the anomalies of 2 NF in Student Table is removed.

We can insert data of System used without a new Roll No.

Any update on System sued or Hourly rate does not need to be performed multiple times.

Any delete operation on Roll No, would not lead to loss of data of system used. Moreover, delete operation will be performed in respective table.

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

Primary key

COURSE

Course_Code	Course_Name	Teacher_Name
C1	Visual Basic	ABC
C2	Oracle&Dev	DEF
C3	C++	KJP
C4	Java	Kumar

Foreign key

Primary key

STUDENT_STSTEM

RollNo	Name	System_Used
100	A1	P-I
101	A2	P-II
102	A3	Celeron
103	A4	P-IV
100	A1	P-I
104	A5	P-III
105	A6	P-II
101	A2	P-II
106	A7	P-IV
107	A8	P-IV
108	A9	P-I
109	A10	Cyrix

Primary key

CHARGES

System_Used	Hourly_ Rate
P-I	20
P-II	30
Celeron	10
P-IV	40
P-III	35
Cyrix	20

Primary key

Foreign key

HOURS_ASSIGNED

Course_Code	RollNo	Total__Hrs
C1	100	7
C1	101	3
C1	102	6
C1	103	1
C2	100	7
C2	104	3
C2	105	1
C2	101	2
C3	106	3
C3	107	2
C3	108	1
C4	109	2

Foreign key

BCNF Normal Form

Manufacturer

Id_No	Name	Item_No	Quantity
M101	Electronics USA	H3772	1000
M101	Electronics USA	J08732	700
M101	Electronics USA	Y23490	200
M322	Electronics-R-Us	H3772	900

M.3

For example, consider a relation

SSP (Sno, Sname, Pno, Qty)

To normalise the table in 1 NF, each cell should have one element only. Hence this table is in 1 NF.

So, every row and column and each cell has one element.

For 2 NF,

1. **It should be 1NF**
2. **non key attributes should be fully functional dependent.**

Now, key attributes can be

1. **{Id No, Item No}**
   I. **In this particular case , we will have two non key attributes, Name and Quantity.**
   II. **Name is dependent on just Id No.**
   III. **Quantity is dependent on the combination PK {Id No, Item No}**
2. **{Name, Item No}**

   Since name is unique.
   I. **In this particular case , we will have two non key attributes, Id No and Quantity.**
   II. **Id No is dependent on just Name.**
   III. **Quantity is dependent on the combination PK {Id No, Item No}**

Now, we see that there can be two possibilities of Primary key of Key attributes.

Hence, we find that in {Id No, Item No} , Name is not a non key attribute but a overlapping candidate key.
And, in {Name, Item No} , id No is not a non key attribute but a overlapping candidate key.

Hence whatever be the key attributes, there is only one non key attribute which is Quantity and it is fully functional dependent.

hence, this particular table is in 2NF

Non Key attribute should be FFD on PK

2NF

over lapping of PK

Manufacturer

Id_No	Name	Item_No	Quantity
M101	Electronics USA	H3772	1000
M101	Electronics USA	J08732	700
M101	Electronics USA	Y23490	200
M322	Electronics-R-Us	H3772	900

1NF

Name → IdNo

ItemNo → Qty

FD2

IdNo → Name

ItemNo → Qty

FD1

For example, consider a relation

SSP (Sno, Sname, Pno, Qty)

Now, checking for 3NF,

1. **It should be 2NF**
2. **There should be no Transitional Dependence.**

For Transitional Dependence, there should be at least two non key attributes, but here we have only one non key attribute in form of Quantity, hence we find that it is in 3 NF

This table is a special case where 2NF and 3 NF fail because there can be Name as well as Id No which can be a part of Primary key.

So, we have to introduce BCNF.

BCNF is simply a stronger definition of 3NF. BCNF makes no explicit reference to first and second normal form as such, nor the concept of full and transitive dependence.

BCNF states that

A relation R is in Boyce/Codd N/F (BCNF) if and only if every determinant is a candidate key. Here determinant is a simple attribute or composite attribute on which some other attribute is fully functionally dependent.

For example Qty is FFD on (Sno, Pno)

(Sno, Pno)→ Qty, here

(Sno, Pno) is a composite determinant.

Sno → Sname

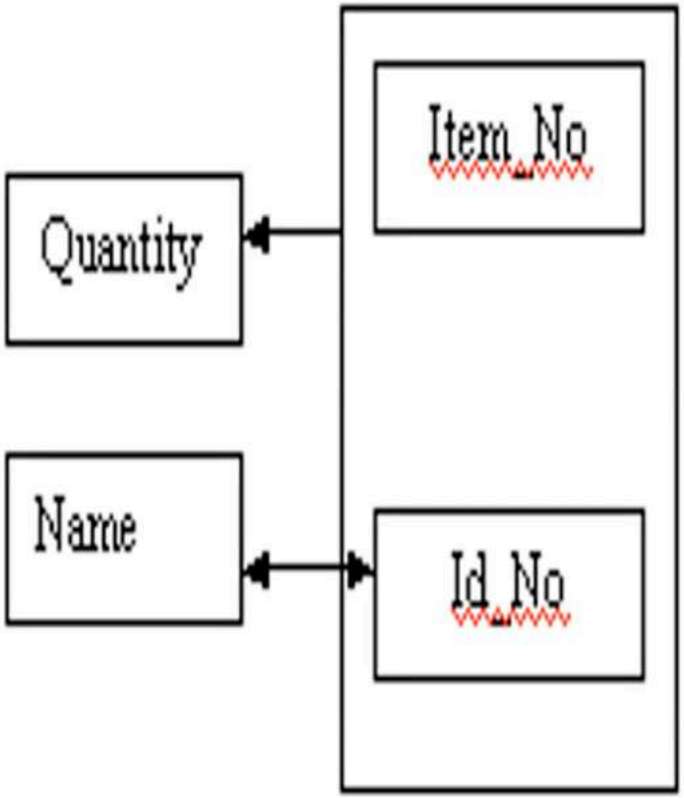
Here Sno is simple attribute determinat.

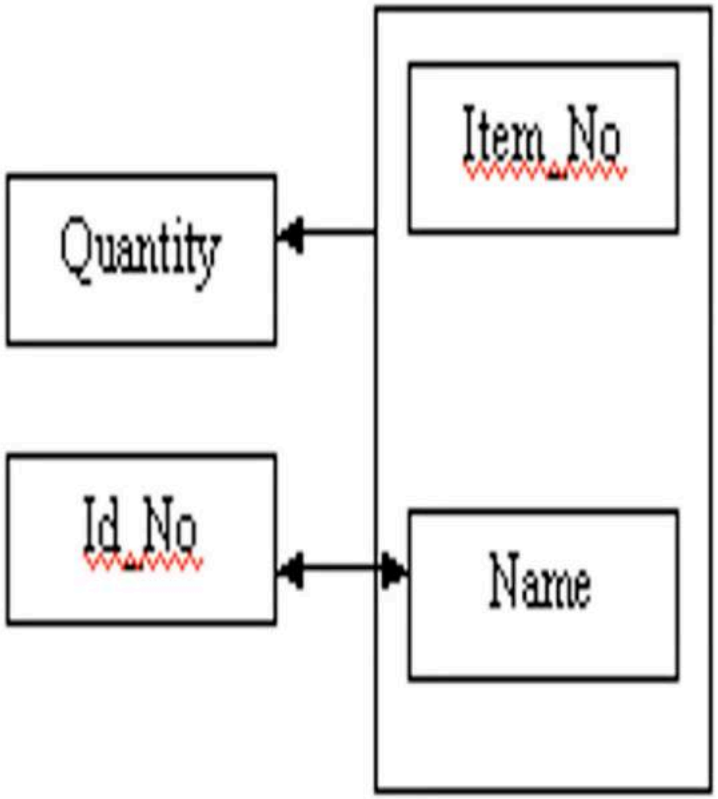
For BCNF, we find that every determinant should be candidate key .

A candidate key is unique and is irreducible(can’t be reduced to a subset which is unique)

X -> Y
For BCNF, X is determinant and hence X should be unique and irreducible.

Possible FD diagram of this case is:



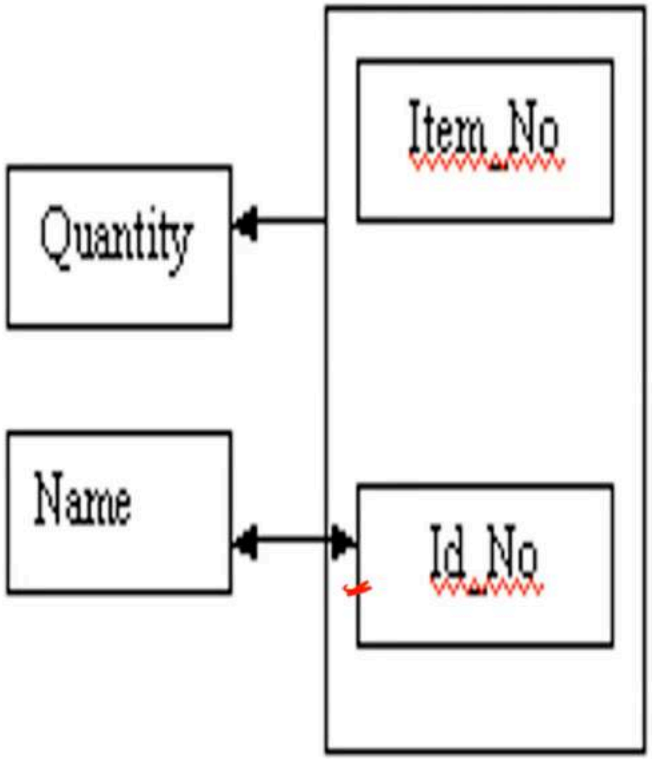


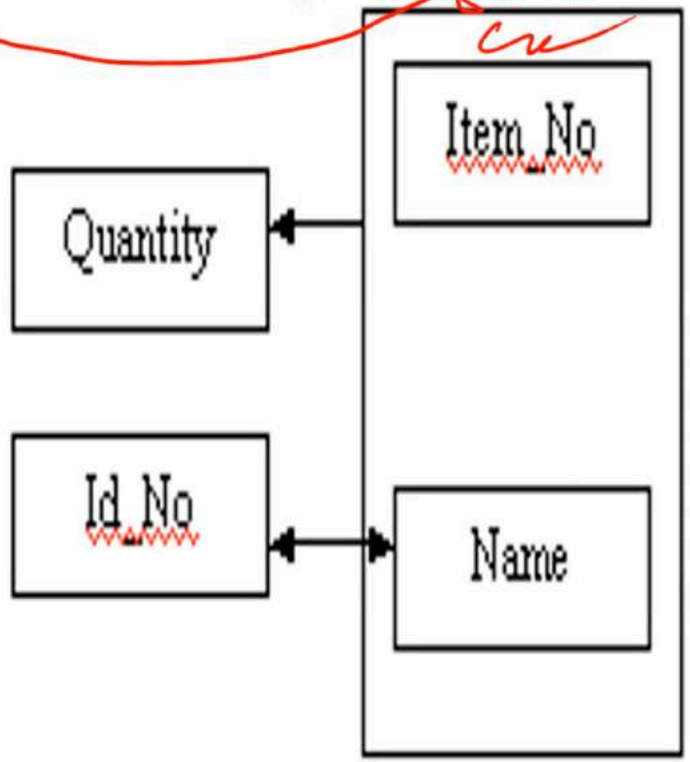
In this particular example, we have 4 determinants,

1. **{Id No, Item No}**
2. **{Id No, Name}**
3. **Id No.**
4. **Name**

Out of these only 1 and 2 are candidate key, the remaining 3 and 4 are not candidate key because they are not unique.

Possible FD diagram of this case is:





ItemNo IdNo

IdNO

Name

ItemNO NameOfthe

det

These

BCNF

So, We will create two tables, one contains these determinant which are not candidate key and other same as rest.

IdNo	Name

IdNo	ItemNo	Qty

BCNF

BCNF is better than 3 NF, because it works where 3 NF works and also where 3 NF fails.

2NF and 3 NF fails in case of candidate key.

STUDENT (Normalized Relation)

Course_ Code	Course_ Name	Teacher_ Name	RollNo	Name	System_ Used	Hourly_ Rate	Total_ Hrs.
C1	Visual Basic	ABC	100	A1	P-I	20	7
C1	Visual Basic	ABC	101	A2	P-II	30	3
C1	Visual Basic	ABC	102	A3	Celeron	10	6
C1	Visual Basic	ABC	103	A4	P-IV	40	1
C2	Oracle&Dev	DEF	100	A1	P-I	20	7
C2	Oracle&Dev	DEF	104	A5	P-III	35	3
C2	Oracle&Dev	DEF	105	A6	P-II	30	1
C2	Oracle&Dev	DEF	101	A2	P-II	30	2
C3	C++	KJP	106	A7	P-IV	40	3
C3	C++	KJP	107	A8	P-IV	40	2
C3	C++	KJP	108	A9	P-I	20	1
C4	Java	Kumar	109	A10	Cyrix	20	2

COURSE

Course_Code	Course_Name	Teacher_Name
C1	Visual Basic	ABC
C2	Oracle&Dev	DEF
C3	C++	KJP
C4	Java	Kumar

STUDENT_STSTEM

RollNo	Name	System_Used
100	A1	P-I
101	A2	P-II
102	A3	Celeron
103	A4	P-IV
100	A1	P-I
104	A5	P-III
105	A6	P-II
101	A2	P-II
106	A7	P-IV
107	A8	P-IV
108	A9	P-I
109	A10	Cyrix

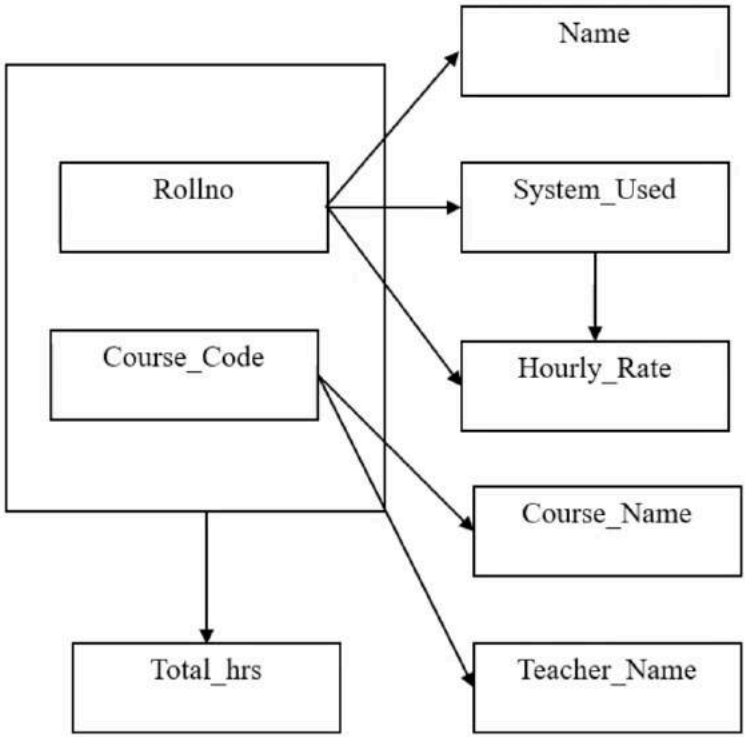
CHARGES

System_Used	Hourly_ Rate
P-I	20
P-II	30
Celeron	10
P-IV	40
P-III	35
Cyrix	20

HOURS_ASSIGNED

Course_Code	RollNo	Total__Hrs
C1	100	7
C1	101	3
C1	102	6
C1	103	1
C2	100	7
C2	104	3
C2	105	1
C2	101	2
C3	106	3
C3	107	2
C3	108	1
C4	109	2

FD Diagram



Determinants

RNo

C_Code

RNo_C_Code

System_Used

STUDENT_STSTEM

Rollno	Name	System_Used
100	A1	PentiumI
101	A2	PentiumII
102	A3	Celeron
103	A4	PentiumIV
104	A5	P-III
105	A6	P-II
106	A7	P-IV
107	A8	P-IV
108	A9	P-I
109	A10	Cyrix

CHARGES

System_Used	Hourly_ Rate
PentiumI	20
PentiumII	30
Celeron	10
PentiumIV	40
Pentium-III	35
Cyrix	20

COURSE

Course_Code	Course_Name	Teacher_Name
C1	Visual Basic	ABC
C2	Oracle&Dev	DEF
C3	C++	KJP
C4	Java	Kumar

HOURS_ASSIGNED

Course_Code	RollNo	Total__Hrs
C1	100	7
C1	101	3
C1	102	6
C1	103	1
C2	100	7
C2	104	3
C2	105	1
C2	101	2
C3	106	3
C3	107	2
C3	108	1
C4	109	2

Fourth Normal Form

Fourth Normal Form (4NF)

A relation R is in Fourth Normal Form (4NF) if and only if the following conditions are satisfied simultaneously:

- R is already in 3NF or BCNF.
- If it contains no multi-valued dependencies.

Multi-Valued Dependency (MVD)

- MVD can be defined informally as follows:
- It means that if in a relation R having A, B and C as attributes, B and C are multi-value facts about A, which is represented as A→→B and A→→C ,then multi value dependency exist only if B and C are independent of each other.
- A→→B **This means that for A, we have multiple values of B.**
- A→→C
- B and C should be independent to each other.

SIMPLIFIED APPROACH DBMS

Course_Code	Student_Name	Book_Name
DBMS	RAJ	Simplified Approach to DBMS
	RAM	Concepts of DBMS
		DBMS With Oracle
Machine Learning	RAM	Simplified Approach to ML
	SHAM	Concepts of ML
	RAJA	
Advanced DBMS	RAJ	Simplified Approach to DBMS
		Databases

Converting it into 1 NF

Course_Code	Student_Name	Book_Name
DBMS	RAJ	Simplified Approach to DBMS
DBMS	RAJ	Concepts of DBMS
DBMS	RAJ	DBMS With Oracle
DBMS	RAM	Simplified Approach to DBMS
DBMS	RAM	Concepts of DBMS
DBMS	RAM	DBMS With Oracle
Machine Learning	RAM	Simplified Approach to ML
Machine Learning	RAM	Concepts of ML
Machine Learning	SHAM	Simplified Approach to ML
Machine Learning	SHAM	Concepts of ML
Machine Learning	RAJA	Simplified Approach to ML
Machine Learning	RAJA	Concepts of ML
Advanced DBMS	RAJ	Simplified Approach to DBMS
Advanced DBMS	RAJ	Databases

SIMPLIFIED APPROACH TO DBMS

Course_Code	Student_Name	Book_Name
DBMS	RAJ	Simplified Approach to DBMS
DBMS	RAJ	Concepts of DBMS
DBMS	RAJ	DBMS With Oracle
DBMS	RAM	Simplified Approach to DBMS
DBMS	RAM	Concepts of DBMS
DBMS	RAM	DBMS With Oracle
Machine Learning	RAM	Simplified Approach to ML
Machine Learning	RAM	Concepts of ML
Machine Learning	SHAM	Simplified Approach to ML
Machine Learning	SHAM	Concepts of ML
Machine Learning	RAJA	Simplified Approach to ML
Machine Learning	RAJA	Concepts of ML
Advanced DBMS	RAJ	Simplified Approach to DBMS
Advanced DBMS	RAJ	Databases

- Table is 1NF.
- Primary Key is
- Combination of
  - **Course_Code**
  - **Student_Name**
  - **Book_Name**

Insert Anomaly

Course_Code	Student_Name	Book_Name
DBMS	RAJ	Simplified Approach to DBMS
DBMS	RAJ	Concepts of DBMS
DBMS	RAJ	DBMS With Oracle
DBMS	RAM	Simplified Approach to DBMS
DBMS	RAM	Concepts of DBMS
DBMS	RAM	DBMS With Oracle
Machine Learning	RAM	Simplified Approach to ML
Machine Learning	RAM	Concepts of ML
Machine Learning	SHAM	Simplified Approach to ML
Machine Learning	SHAM	Concepts of ML
Machine Learning	RAJA	Simplified Approach to ML
Machine Learning	RAJA	Concepts of ML
Advanced DBMS	RAJ	Simplified Approach to DBMS
Advanced DBMS	RAJ	Databases

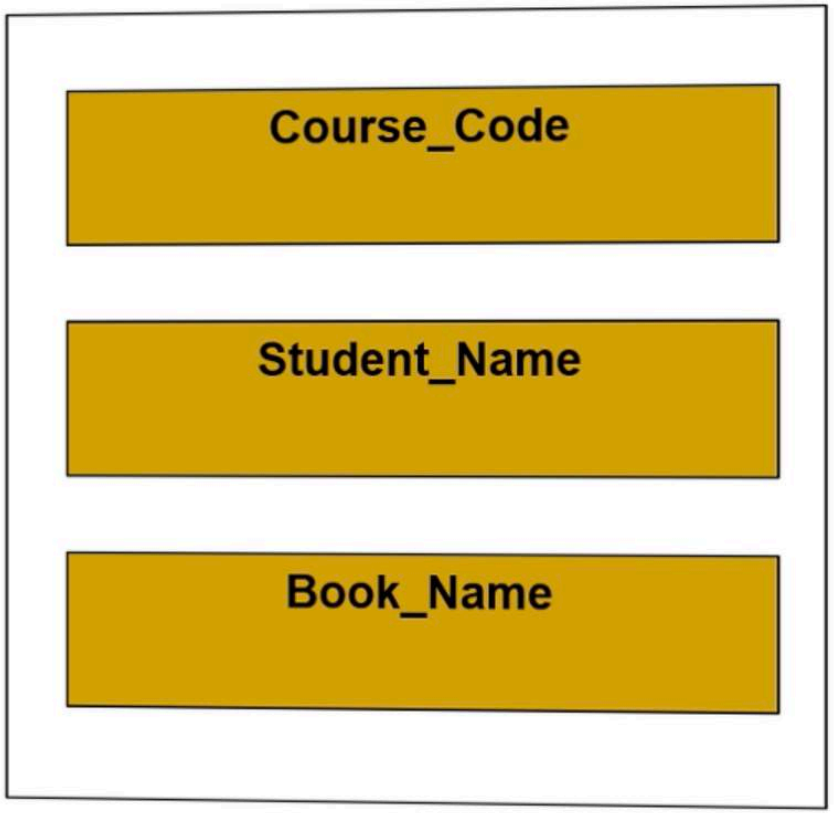
- Cannot insert the data of course and book without student
- Cannot insert the data of course and student without book
- In order to add a new book for a course, we have to make multiple entries which is equal to number of students taking that course.
- In order to add a new student for a course, we have to make multiple entries which is equal to number of books for that course.

Update Anomaly

Course_Code	Student_Name	Book_Name
DBMS	RAJ	Simplified Approach to DBMS
DBMS	RAJ	Concepts of DBMS
DBMS	RAJ	DBMS With Oracle
DBMS	RAM	Simplified Approach to DBMS
DBMS	RAM	Concepts of DBMS
DBMS	RAM	DBMS With Oracle
Machine Learning	RAM	Simplified Approach to ML
Machine Learning	RAM	Concepts of ML
Machine Learning	SHAM	Simplified Approach to ML
Machine Learning	SHAM	Concepts of ML
Machine Learning	RAJA	Simplified Approach to ML
Machine Learning	RAJA	Concepts of ML
Advanced DBMS	RAJ	Simplified Approach to DBMS
Advanced DBMS	RAJ	Databases

- Student name updation needs multiple changes and similarly book name updation needs multiple changes

Lets Normalize this database



To normalise the table in 1 NF, each cell should have one element only. Hence, we have made this table is in 1 NF.

For 2 NF,

1. **It should be 1NF**
2. **non key attributes should be fully functional dependent.**

No non key attributes, so it is in 2 NF

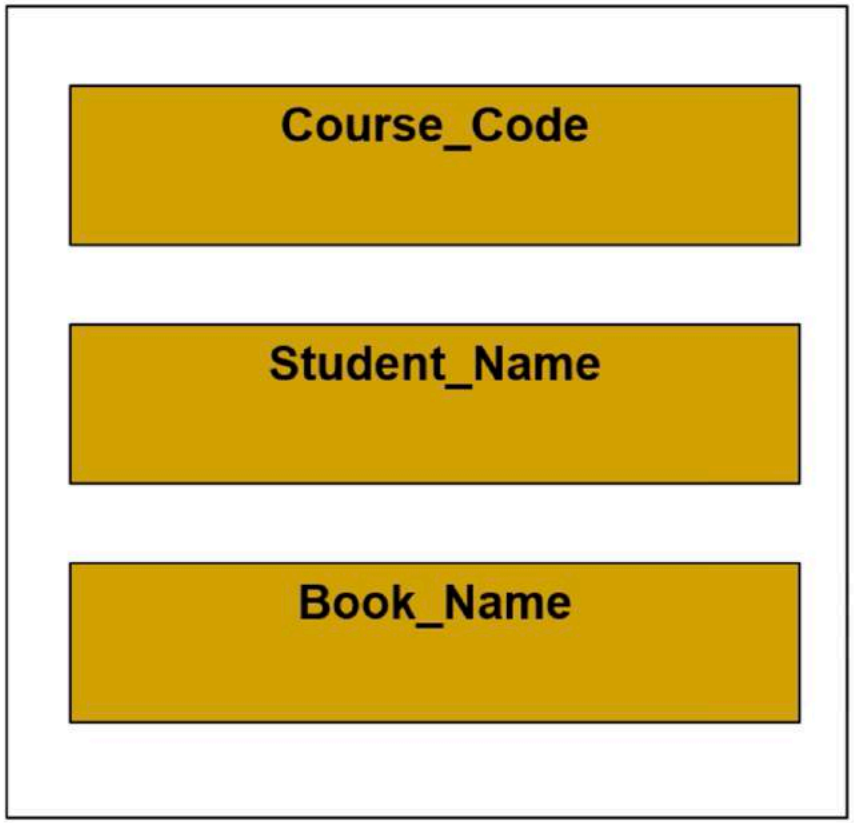
Now, checking for 3NF,

1. **It should be 2NF**
2. **There should be no Transitional Dependence.**

There is no functional dependence in this, so technically it is in 3 NF

For BCNF, we find that every determinant should be candidate key . There is no FFD relation.(No determinant). So, it is in BCNF

Lets Apply 4NF



- A→→B
- A→→C
- B and C should be independent to each other.
- Course_Code →→ Student_name
- Course_Code →→ Book_Name
- Student_name and Book_Name are independent to each other.

Rule to transform a relation into Fourth Normal Form

- A relation R having A,B, and C, as attributes can be non loss-decomposed into two projections R1(A,B) and R2(A,C) if and only if the MVD A→→B|C hold in R.

- Course_Code →→ Student_name|Book_Name

Decompose this to 4NF

- Course_Code →→ Student_name
- Course_Code →→ Book_Name
- To put it into 4NF, two separate tables are formed as shown below:
- COURSE_STUDENT (Course_Code,Student_name)
- COURSE_BOOK (Course_Code, Book_Name)

Final Tables in 4NF

Course_Code	Student_Name
DBMS	RAJ
DBMS	RAM
Machine Learning	RAM
Machine Learning	SHAM
Machine Learning	RAJA
Advanced DBMS	RAJ

Course_Code	Book_Name
DBMS	Simplified Approach to DBMS
DBMS	Concepts of DBMS
DBMS	DBMS With Oracle
Machine Learning	Simplified Approach to ML
Machine Learning	Concepts of ML
Advanced DBMS	Simplified Approach to DBMS
Advanced DBMS	Databases

End