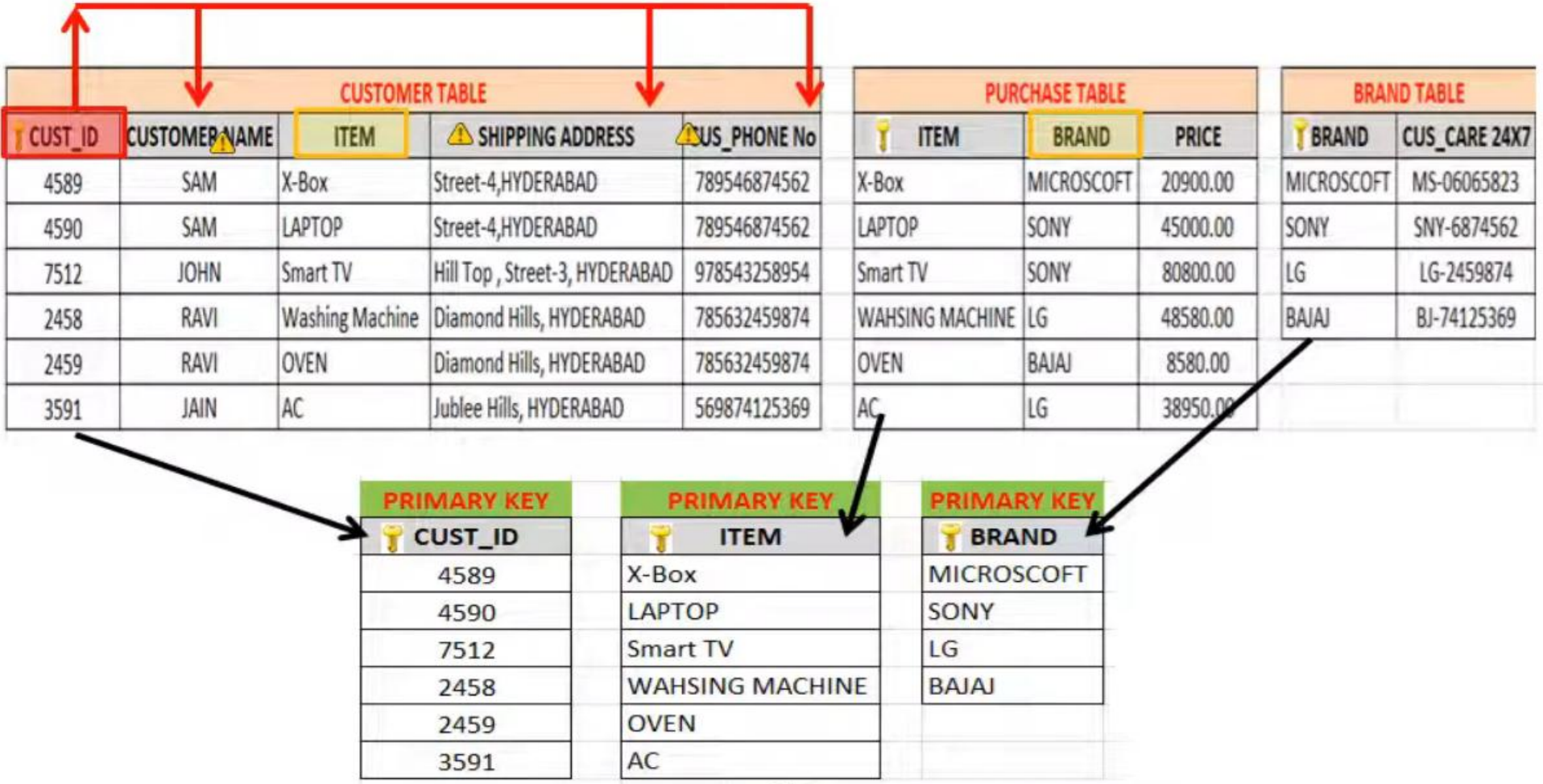
UNIT – III

Biswajit Senapati,
Faculty,
Department of CSE,
NIT AP,
Tadepalligudem, AP.

Functional Dependency

DEPENDENCIES IN DATABASE

{ Customer Name , Shipping Address , Cus_Phone No }
attributes are dependent on {Cus_ID} attribute



CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE

BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

PRIMARY KEY

CUST_ID
4589
4590
7512
2458
2459
3591

PRIMARY KEY

ITEM
X-Box
LAPTOP
Smart TV
WAHSING MACHINE
OVEN
AC

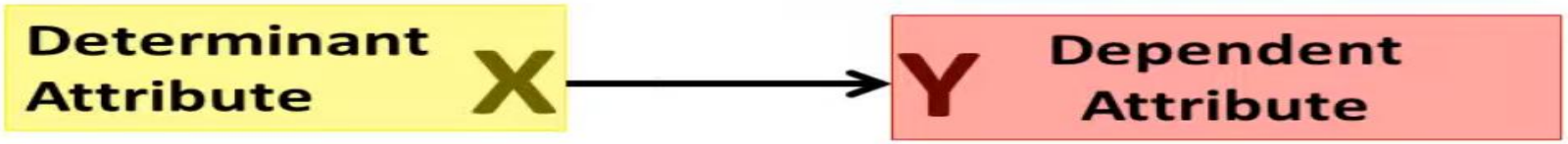
PRIMARY KEY

BRAND
MICROSCOFT
SONY
LG
BAJAJ

FUNCTIONAL DEPENDENCY IN RDBMS

- The phenomenon of determining an attribute of a table with another attribute of same table or relational table is called as Functional Dependency.
- Functional dependency helps to maintain the data integrity in the database.
- A functional dependency is denoted by an arrow →
- The functional dependency of X on Y is represented by

Lets consider X and Y be any two attributes of a table or relation

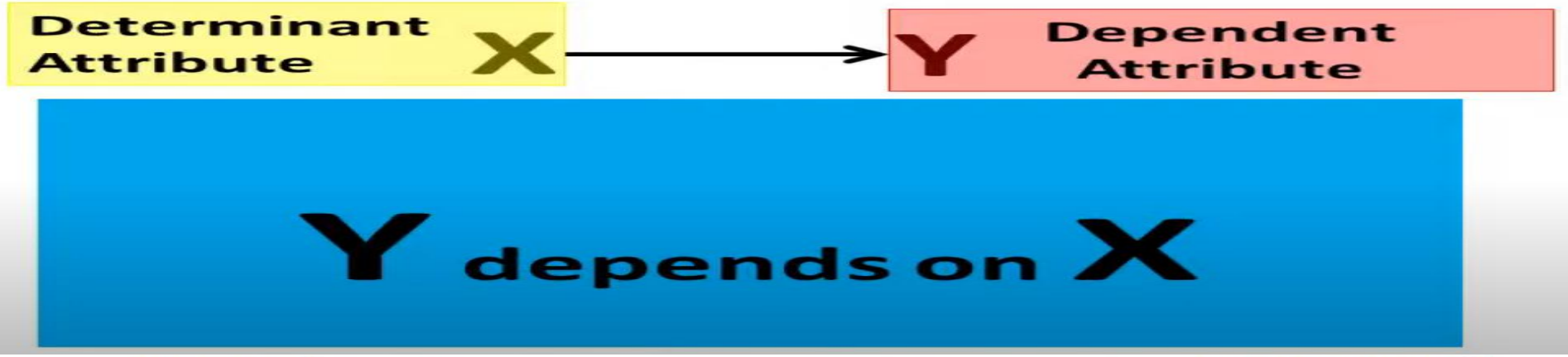


OR

X Determines Y

OR

Y Is Determined by X



HOW FUNCTIONAL DEPENDENCY IN DBMS WORKS ?

- Lets consider **X** and **Y** be any two attributes of a table or relation

X is EMP_ID and **Y** is EMP_NAME

$X \longrightarrow Y$
Y depends on X

CASE-1 $X \rightarrow Y$

EMP_ID	EMP_NAME
105	JOHN
105	JOHN

VALID CASE

CASE-2 $X \rightarrow Y$

EMP_ID	EMP_NAME
105	JOHN
121	SAM

VALID CASE

CASE-3 $X \rightarrow Y$

EMP_ID	EMP_NAME
105	JOHN
121	JOHN

VALID CASE

CASE-4 $X \rightarrow Y$

EMP_ID	EMP_NAME
105	JOHN
105	SAM

INVALID CASE

TYPES OF FUNCTIONAL DEPENDENCIES IN RDBMS

There are two types of functional dependencies in RDBMS, they are :

1. Trivial Functional Dependency

If $X \rightarrow Y$ holds, where Y is a subset of X

2. Non-Trivial Functional Dependency

If $X \rightarrow Y$ holds, where Y is a NOT subset of X

TRIVIAL FUNCTIONAL DEPENDENCIES IN RDBMS

- If a functional dependency (FD) $X \rightarrow Y$ holds, where Y is a subset of X, then it is called a trivial FD.
- Trivial functional dependencies always hold TRUE.
- In Trivial functional dependencies attributes always determines itself.
- If $X \rightarrow Y$ holds, then $X \cap Y \neq 0$

X	→	Y
{EMP_ID , E_Name}	→	{E_Name}

$X \cap Y =$ {E_Name}
Subset and NOT Null

NON-TRIVIAL FUNCTIONAL DEPENDENCIES IN RDBMS

- If an a functional dependency (FD) X → Y holds, where Y is not a subset of X, then it is called a non-trivial FD.
- If Non-Trivial functional dependencies hold then we need to find the validity of the dependency by assuming different cases.
- In Non-Trivial functional dependencies attributes always determines other attributes.
- If X → Y holds, then X n Y = ø

X	→	Y
{EMP_ID , E_Name}	→	{E_Sal}

X n Y = {ø}

Not a Subset and it is Null

PROPERTIES FUNCTIONAL DEPENDENCIES IN DBMS

1. REFLEXIVITY:

If Y is a subset of X then X→Y holds good

Example:

If X → Y holds, then X n Y ≠ 0

X	→	Y
{EMP_ID , E_Name}	→	{E_Name}

X n Y = {E_Name}
Subset and NOT Null

2. AUGMENTATION:

If X→ Y, then XZ → YZ holds good

Example:

If X → Y holds, then X n Y ≠ 0

XZ	→	YZ
{EMP_ID , E_Sal}	→	{E_Name , E_Sal}

X n Y = {E_Sal}

3. TRANSITIVE:

If X → Y and Y → Z, then X → Z holds good

Example:

If X → Y and Y → Z, then X → Z

X → Y
{EMP_ID} → {E_Name}

Y → Z
{E_Name} → {E_City}

X → Z
{EMP_ID} → {E_City}

4. UNION:

If X → Y and X → Z, then X → YZ holds good

Example:

If X → Y and X → Z, then X → YZ

X → Y
{EMP_ID} → {E_Name}

X → Z
{EMP_ID} → {E_City}

X → YZ
{EMP_ID} → {E_Name, E_City}

5. DECOMPOSITION:

If x → YZ , then X → Y and X → Z holds good

Example:

If x → YZ , then X → Y and X → Z

X → Y Z
{EMP_ID} → {E_Name , E_City}

X → Y
{EMP_ID} → {E_Name}

X → Z
{EMP_ID} → {E_City}

6. COMPOSITION:

If X → Y and Z → W , then XZ → YW is holds good

If X → Y and Z → W then XZ → YW

DEPENDENCY PRESERVING DECOMPOSITION IN DBMS

AIM: Dependencies of original table are preserved in decomposed tables

Table - T			
A	B	C	D

Functional Dependencies:

FD: $A \rightarrow B$, $B \rightarrow C$

Closure of FD

FD^+: $A \rightarrow C$

Table - T			
A	B	C	D

T-1	
A	B

FD 1

T-2	
C	D

FD 2

The **UNION** of FDs MUST be EQUAL to the original FD

$\{FD1\} \cup \{FD2\} = \{FD^+\}$

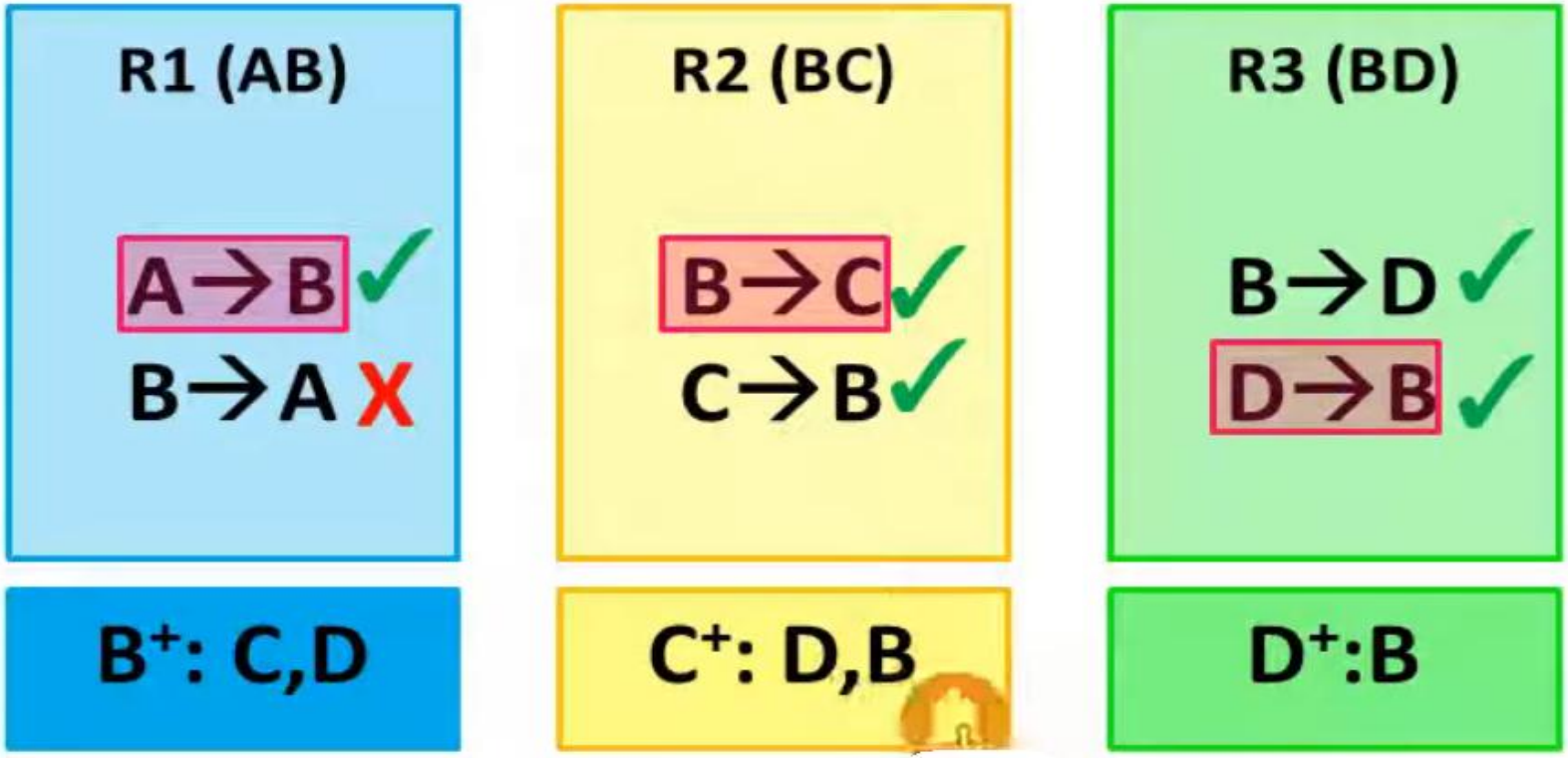
Example:

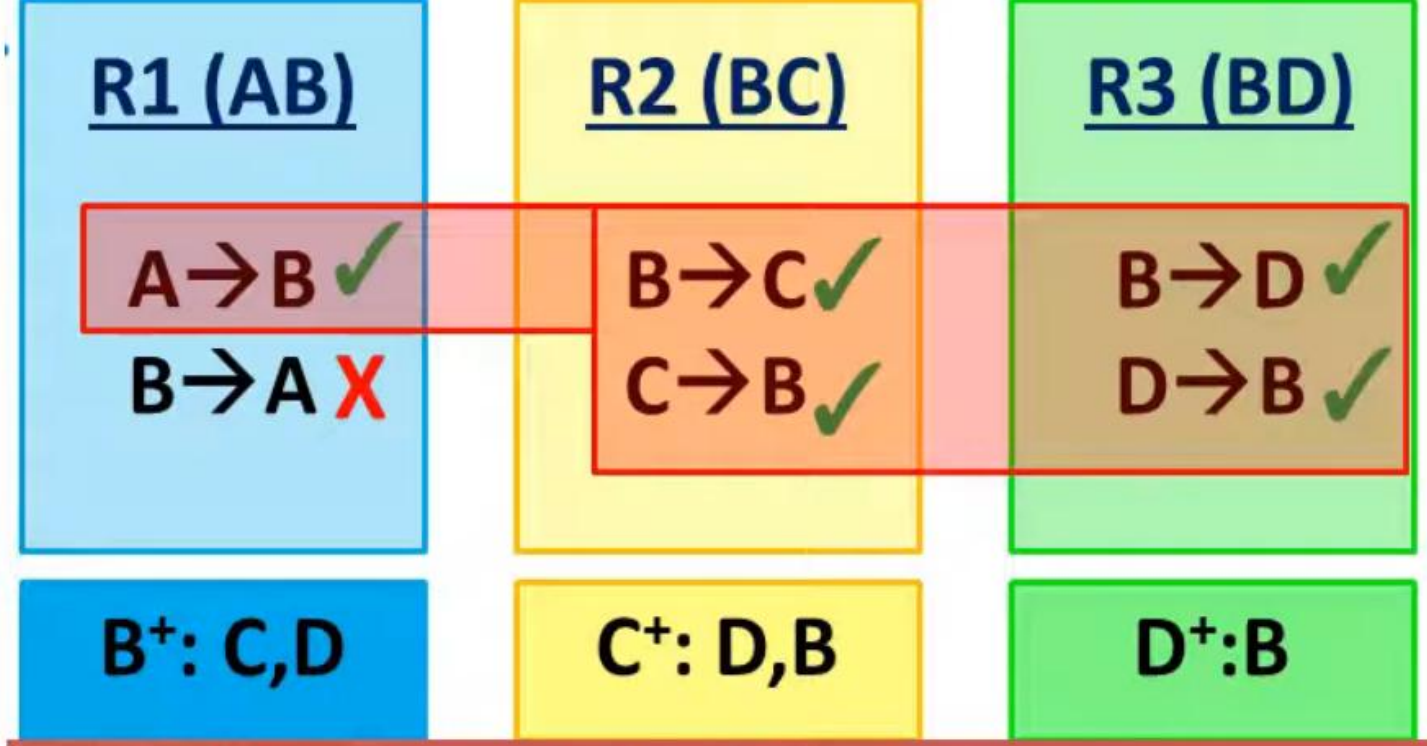
Dependency Preserving Decomposition

Let R(ABCD) with Function Dependencies

{A→B, B→C, C→D, D→B}

R is decomposed into R1(AB) , R2(BC) , R3(BD)





The **UNION** of FDs MUST be EQUAL to the original FD

{FD1} **U** {FD2} = {FD+}

{(A→B) U (B→C) U (C→B) U (B→D) U (D→B)}

{ (A→B) , (B→C), (C→D), (D→B) }

Hence

ALL the Dependencies of ORIGINAL TABLE are PRESERVED in DECOMPOSED TABLES

Why Normalization ?

NORMALIZATION:

- Normalization is a technique to REMOVE or REDUCE the REDUNDANCY (DUPLICATE DATA) from a table in database
- Duplication of Data is of two types

S.NO	SNAME	GRADE

ROW LEVEL

S.NO	SNAME	COURSE	FACULTY	FEE

COLUMN LEVEL

ROW LEVEL DUPLICATION OF DATA

- To Remove the Row level duplication of Data – PRIMARY KEY is used

PRIMARY KEY: Primary Key is a Key which uniquely define each row in a table (UNIQUE + NOT NULL)

S.NO	SNAME	GRADE
1001	RAVI	A
1005	RAJU	B+
1007	RAMESH	A
1008	SAM	C
1003	JAIN	A+
1005	RAJU	B+
1004	KUMAR	A+
1010	PREM	B+
1008	SAM	C

S.NO	SNAME	GRADE
1001	RAVI	A
1003	JAIN	A+
1004	KUMAR	A+
1005	RAJU	B+
1007	RAMESH	A
1008	SAM	C
1010	PREM	B+

COLUMN LEVEL DUPLICATION OF DATA

- Insertion Anomaly
- Deletion Anomaly
- Updation Anomaly

S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3000	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1003	RAJU	C1	C++	3000	F1	JOHN
1004	RAMESH	C1	C++	3000	F1	JOHN
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

INSERTION ANOMALY

Inserting a new data into table – Adding a New Course

S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3000	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1003	RAJU	C1	C++	3000	F1	JOHN
1004	RAMESH	C1	C++	3000	F1	JOHN
		C4	UNIX	5000		
.	.	.	.	.	.	.
Primary Key Should not be NULL				.	.	.
.	.	.	.	.	.	.

DELETION ANOMALY

Deleting an existing record from a table

```
Delete from Std_Table
Where S_ID = 1003
```

S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3000	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1003	RAJU	C1	C++	3000	F1	JOHN
1004	RAMESH	C1	C++	3000	F1	JOHN
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Deleting an existing record from a table

```
Delete from Std_Table
Where S_ID = 1003
```

S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3000	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1004	RAMESH	C1	C++	3000	F1	JOHN
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

The data related to Course & Faculty is also deleted

UPDATION ANOMALY

Updating an existing record in a table

```
Update Std_Table
Set C_FEE = 3500
Where C_ID = C1
```

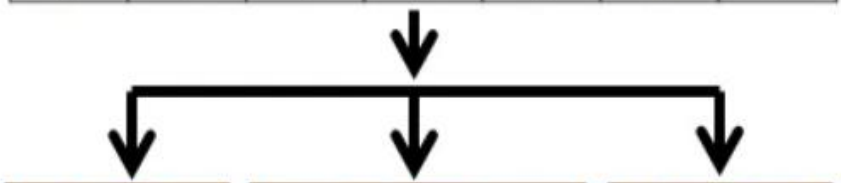
S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3500	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1003	RAJU	C1	C++	3500	F1	JOHN
1004	RAMESH	C1	C++	3500	F1	JOHN
.	.	.	.	.	.	.

In case of LARGE VOLUME of data
Updating data will take LONGER TIME,

NORMALIZATION

- Normalization is a process of **ORGANIZING** the data in database to **AVOID DATA REDUNDANCY**, **INSERTION** ANOMALY, **UPDATE** ANOMALY & **DELETION** ANOMALY.

S_ID	S_NAME	C_ID	COURSE	C_FEE	F_ID	F_NAME
1001	JAIN	C1	C++	3000	F1	JOHN
1002	KUMAR	C2	JAVA	4500	F2	BOB
1003	RAJU	C1	C++	3000	F1	JOHN
1004	RAMESH	C1	C++	3000	F1	JOHN
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.



Std_Table

S_ID	S_NAME
1001	JAIN
1002	KUMAR
1003	RAJU
1004	RAMESH
.	.

Course_Table

C_ID	COURSE	C_FEE
C1	C++	3000
C2	JAVA	4500
.	.	.
.	.	.
.	.	.

Faculty_Table

F_ID	F_NAME
F1	JOHN
F2	BOB
.	.
.	.
.	.

NORMALIZATION

- Here are the most commonly used Normal Forms:
  - First Normal Form(1NF)
  - Second Normal Form(2NF)
  - Third Normal Form(3NF)
  - Boyce & Codd Normal Form (BCNF)

First Normal Form(1NF)

RULES OF FIRST NORMAL FORM (1NF)

Rule-1: In First Normal Form (1NF), an attribute (column) of a table cannot hold multiple values.

Rule-2: It should hold only atomic values.

OR

Rule-1: Each cell to be single valued

Rule-2: Entries in a column are same type

Rule-3: Row uniquely identified – Add Unique ID or Add more columns to make unique

Rule-1: Each cell to be single valued

Rule-2: Entries in a column are same type

Rule-3: Row uniquely identified – Add Unique ID or Add more columns to make unique.

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	BRAND	CUS_PHONE No	PRICE
4589	SAM	X-Box , LAPTOP	Street-4,HYDERABAD	MICROSOFT, DELL	789546874562	20900 , 58000
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	SONY	978543258954	80800.00
2458	RAVI	Washing Machine , AC	Diamond Hills, HYDERABAD	LG , SAMSUNG	785632459874	35980 , 30000
3591	JAIN	Oven	Jublee Hills, HYDERABAD	BAJAJ	569874125369	15000.00

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	BRAND	CUS_PHONE No	PRICE
4589	SAM	X-Box	Street-4,HYDERABAD	MICROSOFT	789546874562	20900.00
4590	SAM	LAPTOP	Street-4,HYDERABAD	DELL	789546874562	58000.00
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	SONY	978543258954	80800.00
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	LG	785632459874	35980.00
2459	RAVI	AC	Diamond Hills, HYDERABAD	SAMSUNG	785632459874	30000.00
3591	JAIN	Oven	Jublee Hills, HYDERABAD	BAJAJ	569874125369	15000.00

PROBLEM IN FIRST NORMAL FORM(1NF)

FIRST NORMAL FORM(1NF)

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	BRAND	CUS_PHONE No	PRICE
4589	SAM	X-Box	Street-4,HYDERABAD	MICROSOFT	789546874562	20900.00
4590	SAM	LAPTOP	Street-4,HYDERABAD	DELL	789546874562	58000.00
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	SONY	978543258954	80800.00
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	LG	785632459874	35980.00
2459	RAVI	AC	Diamond Hills, HYDERABAD	SAMSUNG	785632459874	30000.00
3591	JAIN	Oven	Jublee Hills, HYDERABAD	BAJAJ	569874125369	15000.00

Second Normal Form(2NF)

RULES FOR SECOND NORMAL FORM(NF)

- **A table is said to be in 2NF if both the following conditions hold:**

Rule-1: **Table is in 1NF**

Rule-2: **No non-prime attribute is dependent on the proper subset of any candidate key of table.**

OR

Rule-1: **Table is in 1NF**

Rule-2: **All attributes (non-prime Column) dependent on key of table.**

Rule-1: Table is in 1NF

Rule-2: All attributes (non-prime Column) dependent on key of table.

CUST_ID	CUSTOMER NAME	CUS_PHONE No	ITEM	SHIPPING ADDRESS	BRAND	CUS_CARE 24X7	PRICE
4589	SAM	789546874562	X-Box	Street-4,HYDERABAD	MICROSCOFT	MS-06065823	20900.00
4589	SAM	789546874562	LAPTOP	Street-4,HYDERABAD	SONY	SNY-6874562	45000.00
7512	JOHN	978543258954	Smart TV	Hill Top , Street-3, HYDERABAD	SONY	SNY-6874562	80800.00
2458	RAVI	785632459874	WAHSING MACHINE	Diamond Hills, HYDERABAD	LG	LG-2459874	48580.00
2458	RAVI	785632459874	OVEN	Diamond Hills, HYDERABAD	BAJAJ	BJ-74125369	8580.00
3591	JAIN	569874125369	AC	Jublee Hills, HYDERABAD	LG	LG-2459874	38950.00

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	CUS_CARE 24X7	PRICE
X-Box	MICROSCOFT	MS-06065823	20900.00
LAPTOP	SONY	SNY-6874562	45000.00
Smart TV	SONY	SNY-6874562	80800.00
WAHSING MACHINE	LG	LG-2459874	48580.00
OVEN	BAJAJ	BJ-74125369	8580.00
AC	LG	LG-2459874	38950.00

Rule-1: Table is in 1NF

Rule-2: All attributes (non-prime Column) dependent on key of table.

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	CUS_CARE 24X7	PRICE
X-Box	MICROSCOFT	MS-06065823	20900.00
LAPTOP	SONY	SNY-6874562	45000.00
Smart TV	SONY	SNY-6874562	80800.00
WAHSING MACHINE	LG	LG-2459874	48580.00
OVEN	BAJAJ	BJ-74125369	8580.00
AC	LG	LG-2459874	38950.00

PRIMARY KEY	PRIMARY KEY
CUST_ID	ITEM
4589	X-Box
4590	LAPTOP
7512	Smart TV
2458	Washing Machine
2459	AC
3591	Oven

COMPOUND KEY

WHY THIRD NORMAL FORM(3NF) ?

SECOND NORMAL FORM(2NF)-TABLE

CUSTOMER TABLE				
CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE			
ITEM	BRAND	CUS_CARE 24X7	PRICE
X-Box	MICROSCOFT	MS-06065823	20900.00
LAPTOP	SONY	SNY-6874562	45000.00
Smart TV	SONY	SNY-6874562	80800.00
WAHSING MACHINE	LG	LG-2459874	48580.00
OVEN	BAJAJ	BJ-74125369	8580.00
AC	LG	LG-2459874	38950.00

Third Normal Form(3NF)

RULES FOR THIRD NORMAL FORM(3NF)

Rule-1: Table MUST be in 2NF

Rule-2: Transitive functional dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as non-prime attribute.

OR

Rule-1: Table MUST be in 2NF

Rule-2: All fields (Columns) can be determined ONLY by the key in the table and no other column

RULES FOR THIRD NORMAL FORM(3NF)

Rule-1: Table MUST be in 2NF

Rule-2: All fields (Columns) can be determined ONLY by the key in the table and no other column

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	CUS_CARE 24X7	PRICE
X-Box	MICROSCOFT	MS-06065823	20900.00
LAPTOP	SONY	SNY-6874562	45000.00
Smart TV	SONY	SNY-6874562	80800.00
WAHSING MACHINE	LG	LG-2459874	48580.00
OVEN	BAJAJ	BJ-74125369	8580.00
AC	LG	LG-2459874	38950.00

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE

BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

CUSTOMER TABLE				
CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE		
ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE	
BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

PRIMARY KEY
CUST_ID
4589
4590
7512
2458
2459

PRIMARY KEY
ITEM
X-Box
LAPTOP
Smart TV
WAHSING MACHINE
OVEN

PRIMARY KEY
BRAND
MICROSCOFT
SONY
LG
BAJAJ

WHY FOURTH NORMAL FORM(4NF) ?

CUSTOMER TABLE				
CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE		
ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE	
BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

THIRD NORMAL FORM (3NF)

PRIMARY KEY
CUST_ID
4589
4590
7512
2458
2459
3591

PRIMARY KEY
ITEM
X-Box
LAPTOP
Smart TV
WAHSING MACHINE
OVEN
AC

PRIMARY KEY
BRAND
MICROSCOFT
SONY
LG
BAJAJ

Fourth Normal Form(4NF)

RULES FOR FOURTH NORMAL FORM(4NF)

Rule-1: Table MUST be in 3NF

Rule-2: NO multi-valued dependencies

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	ITEM	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	X-Box	Street-4,HYDERABAD	789546874562
4590	SAM	LAPTOP	Street-4,HYDERABAD	789546874562
7512	JOHN	Smart TV	Hill Top , Street-3, HYDERABAD	978543258954
2458	RAVI	Washing Machine	Diamond Hills, HYDERABAD	785632459874
2459	RAVI	OVEN	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	AC	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE

BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

CUS_CART TABLE

CUST_ID	ITEM
4589	X-Box
4590	LAPTOP
7512	Smart TV
2458	Washing Machine
2459	OVEN
3591	AC

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	Street-4,HYDERABAD	789546874562
7512	JOHN	Hill Top , Street-3, HYDERABAD	978543258954
2459	RAVI	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	Jublee Hills, HYDERABAD	569874125369

PURCHASE TABLE

ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

BRAND TABLE

BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

RELATIONSHIP IN Fourth Normal Form(4NF)

CUSTOMER TABLE & CUS_CART TABLE

ONE Customer can purchase MULTIPLE items , so the relationship will to ONE-TO-MANY

CUS_CART TABLE & PURCHASE TABLE

ONE item can have MULTIPLE brands , so the relationship will to ONE-TO-MANY

CUS_CART TABLE & PURCHASE TABLE

MULTIPLE items can have ONE Cus_care_no., so the relationship will to MANY-TO-ONE

CUSTOMER TABLE

CUST_ID	CUSTOMER NAME	SHIPPING ADDRESS	CUS_PHONE No
4589	SAM	Street-4,HYDERABAD	789546874562
7512	JOHN	Hill Top , Street-3, HYDERABAD	978543258954
2459	RAVI	Diamond Hills, HYDERABAD	785632459874
3591	JAIN	Jublee Hills, HYDERABAD	569874125369

1 — N

CUS_CART TABLE

CUST_ID	ITEM
4589	X-Box
4590	LAPTOP
7512	Smart TV
2458	Washing Machine
2459	OVEN
3591	AC

1 — N

BRAND TABLE

BRAND	CUS_CARE 24X7
MICROSCOFT	MS-06065823
SONY	SNY-6874562
LG	LG-2459874
BAJAJ	BJ-74125369

1 — N

PURCHASE TABLE

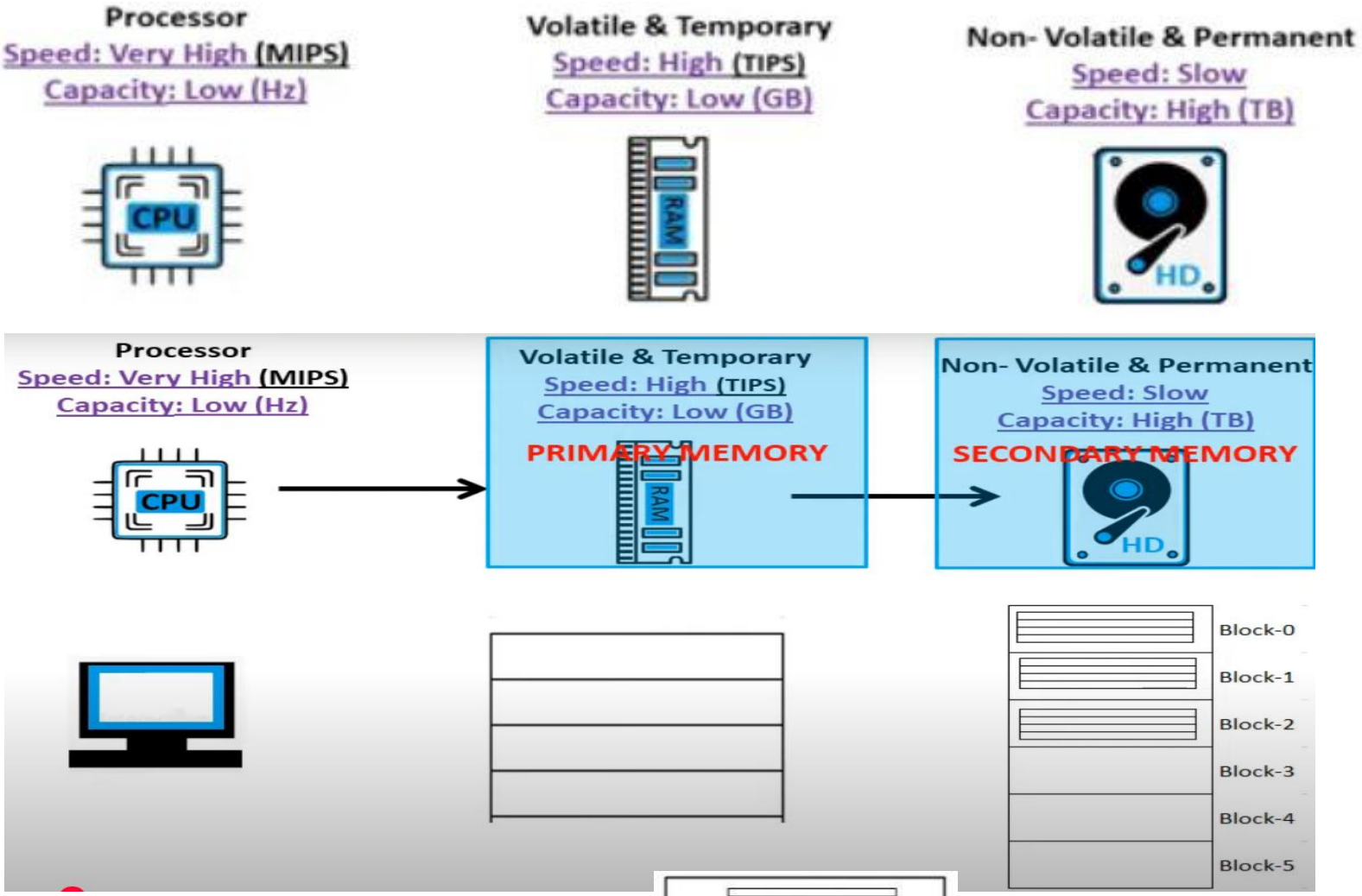
ITEM	BRAND	PRICE
X-Box	MICROSCOFT	20900.00
LAPTOP	SONY	45000.00
Smart TV	SONY	80800.00
WAHSING MACHINE	LG	48580.00
OVEN	BAJAJ	8580.00
AC	LG	38950.00

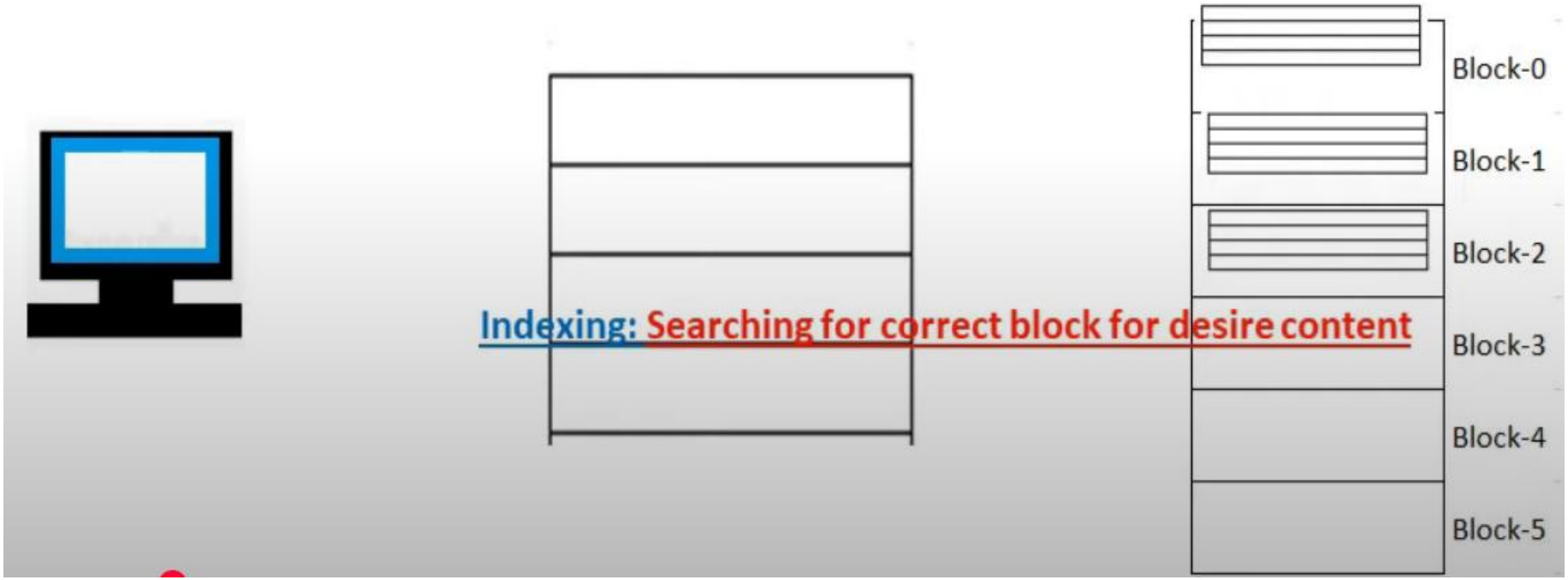
File organization and indexing (e.g., B and B+ trees)

FILE ORGANIZATION AND INDEXING

- A file organization is a way of arranging the records in a file when the file is stored on disk.
- Records should be organized on disk in a such a way to access them efficiently.
- The files and access method layer stores the records in a file, keeps track of pages allocated to each file, and available space within pages allocated to the file.

WHY INDEXING IS USED





INDEX in Books

S.NO.	TOPIC	PAGE NO.
1	Tables in Database	1
2	Keys in Database	6
3	Dependencies in Database	11
4	Normalization in Database	15
5	Indexing in Database	28
6	Data Dictionary in Database	35
7	E-R Model	42

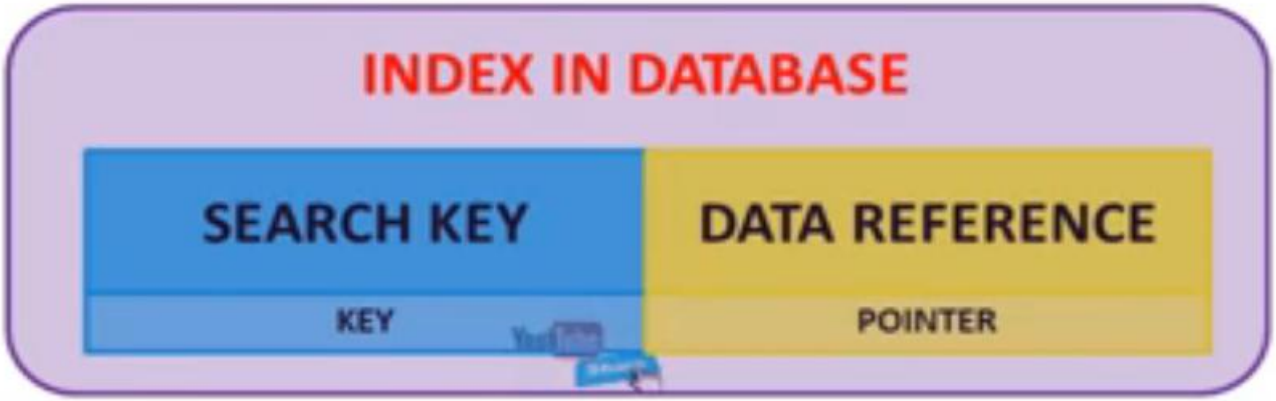
SEARCH KEY

DATA POINTER

INDEX IN DATABASE

- A database index is a DATA STRUCTURE that improves the SPEED OF DATA RETRIEVAL operations on a database table.
- It is used to LOCATE AND ACCESS the data in a database table QUICKLY, this process is called as INDEXING.
- Indexing is used to OPTIMIZE the performance of a database by MINIMIZING THE NUMBER OF DISK ACCESSES REQUIRED when a QUERY IS PROCESSED.

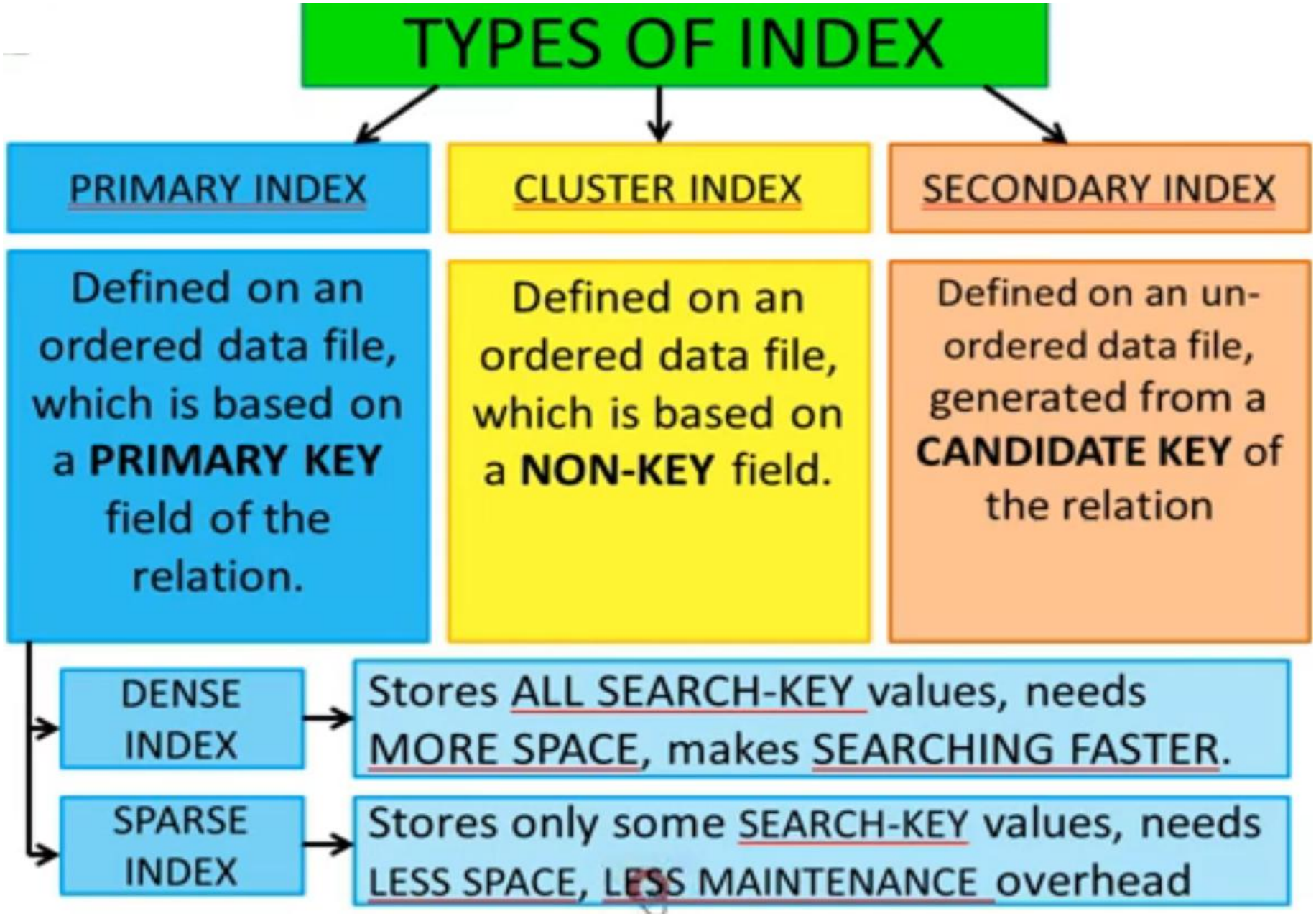
- Indexes are created using a FEW DATABASE COLUMNS.
- An INDEX is a small table having ONLY TWO COLUMNS.
- The FIRST COLUMN comprises a copy of the PRIMARY OR CANDIDATE KEY of a table. Its SECOND COLUMN contains a SET OF POINTERS for holding the ADDRESS of the DISK BLOCK where that SPECIFIC KEY value stored.



INDEXING IN DATABASE

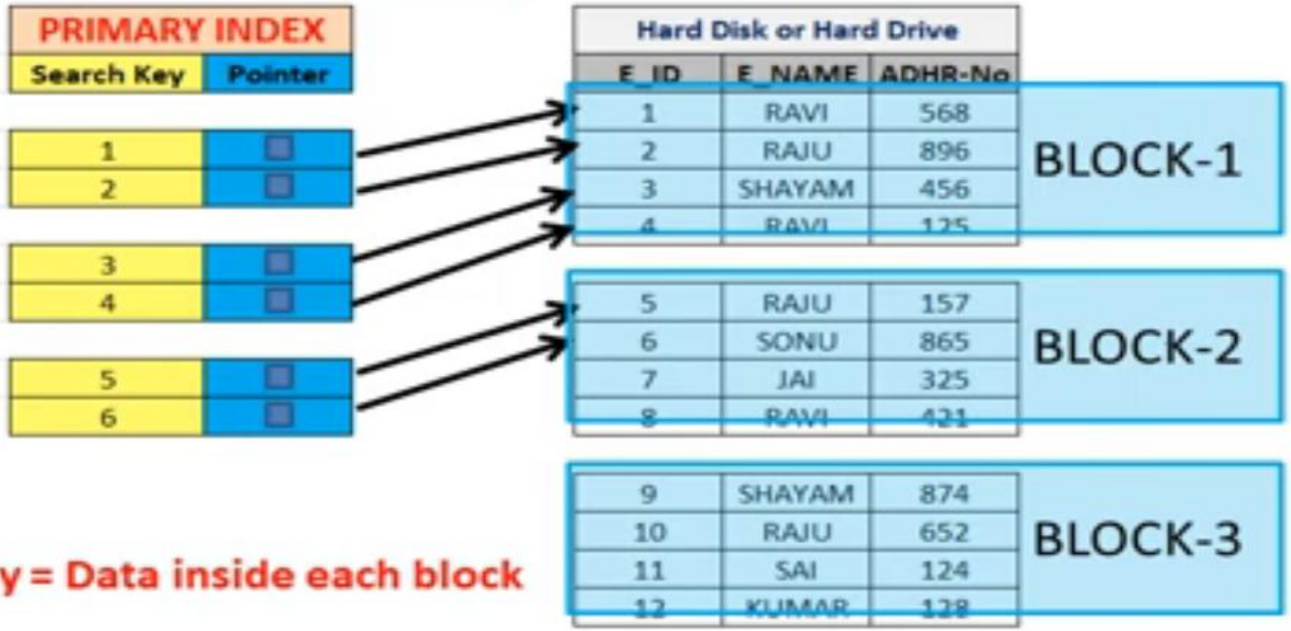
Indexing is defined based on indexing attributes, they are:

1. **ACCESS TYPES:** This refers to the type of access such as value based search, range access, etc.
2. **ACCESS TIME:** It refers to the time needed to find particular data element or set of elements.
3. **INSERTION TIME:** It refers to the time taken to find the appropriate space and insert a new data.
4. **DELETION TIME:** Time taken to find an item and delete it as well as update the index structure.
5. **SPACE OVERHEAD:** It refers to the additional space required by the index.


TYPES OF INDEX
PRIMARY INDEX
CLUSTER INDEX
SECONDARY INDEX
Defined on an ordered data file, which is based on a PRIMARY KEY field of the relation.
Defined on an ordered data file, which is based on a NON-KEY field.
Defined on an un-ordered data file, generated from a CANDIDATE KEY of the relation
DENSE INDEX
Stores ALL SEARCH-KEY values, needs MORE SPACE, makes SEARCHING FASTER.
SPARSE INDEX
Stores only some SEARCH-KEY values, needs LESS SPACE, LESS MAINTENANCE overhead

PRIMARY INDEX- DENSE INDEX

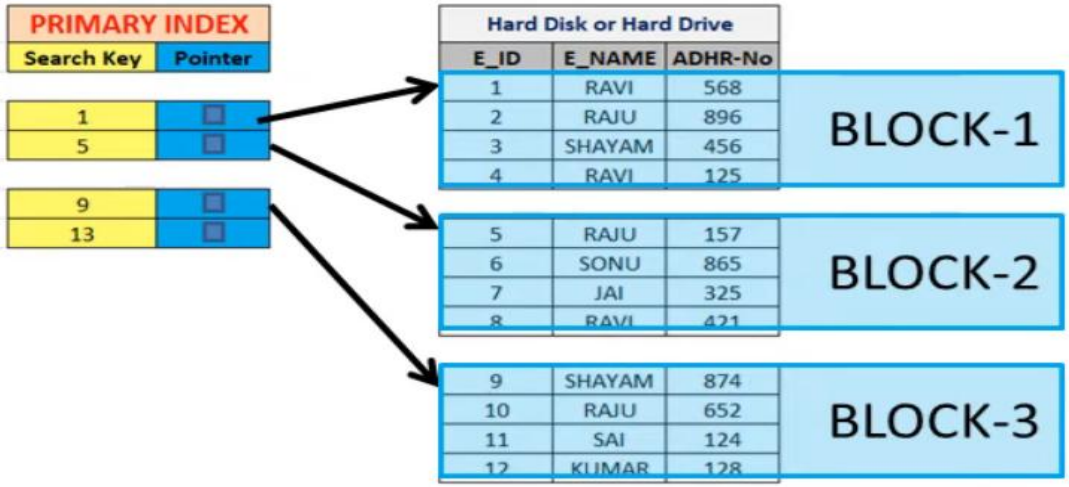
- Data is SORTED based on Primary Key.
- Dense Index Stores ALL SEARCH-KEY VALUES.
- Needs MORE SPACE and MAKES SEARCHING FASTER.
- Index records contain SEARCH KEY VALUE and a POINTER to the ACTUAL RECORD on the DISK.



Search Key = Data inside each block

PRIMARY INDEX- SPARSE INDEX

- Data is SORTED based on Primary Key
- Stores ONLY SOME SEARCH-KEY values.
- Needs LESS SPACE, LESS MAINTENANCE overhead.
- In Sparse indexing technique, a RANGE OF INDEX COLUMNS stores the same DATA BLOCK ADDRESS, and when data needs to be RETRIEVED, the block ADDRESS will be FETCHED.



INDEX – CLUSTERED INDEX

When to use Clustered Index ?

1. Ordered Data
2. Data already have Primary Index
3. Non-Key based search required.

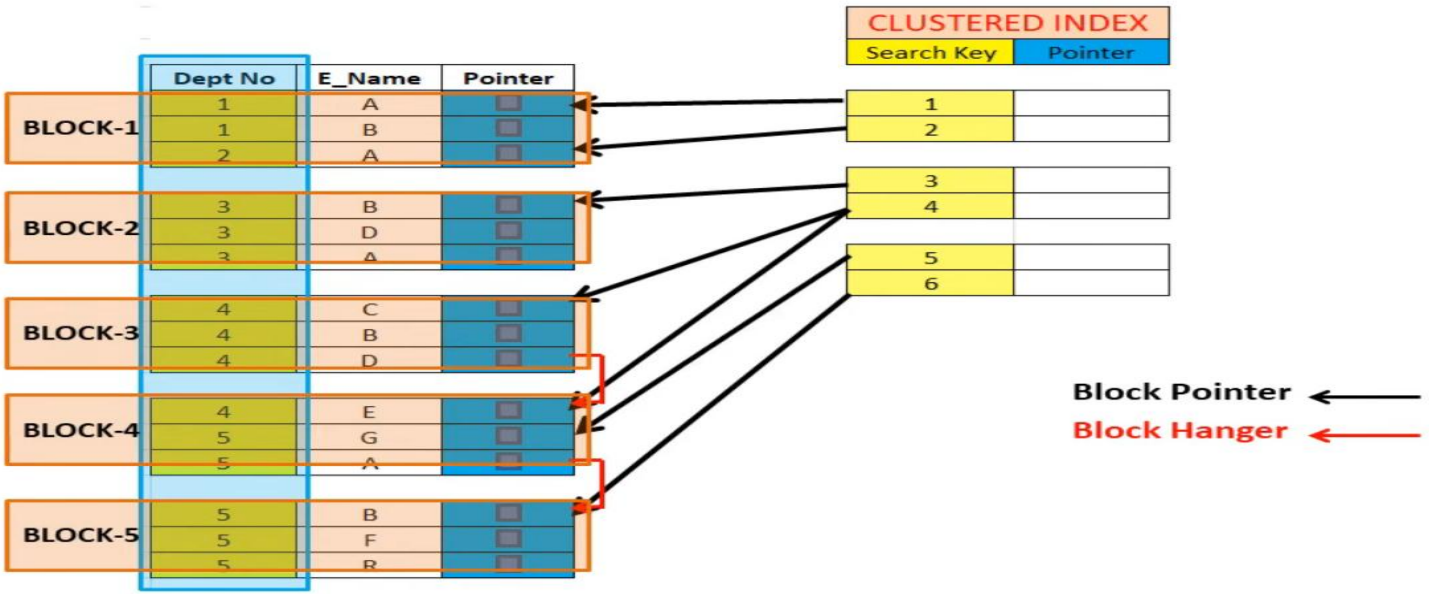
- Mostly clustered Index will have COMBINATION OF NON KEY ATTRIBUTE ALONG WITH CANDIDATE KEY of the tables.
- Clustered Index is a SPARSE INDEX TYPE, where only ONE SEARCH KEY VALUE per block.

Dept No	E_Name	Pointer
1	A	
1	B	
2	A	
3	B	
3	D	
3	A	
4	C	
4	B	
4	D	
4	E	
5	G	
5	A	
5	B	
5	F	
5	R	

CLUSTERED INDEX	
Search Key	Pointer
1	
2	
3	
4	
5	
6	

Duplicate Data

Dept No	E_Name	Pointer
1	A	
1	B	
2	A	
3	B	
3	D	
3	A	
4	C	
4	B	
4	D	
4	E	
5	G	
5	A	
5	B	
5	F	
5	R	



SECONDARY INDEX

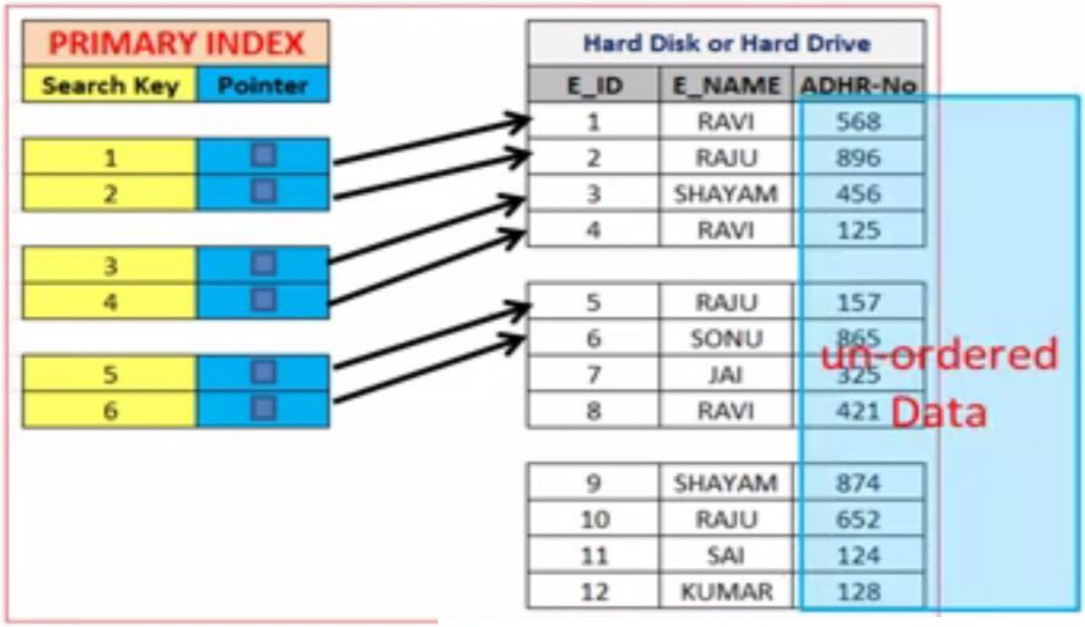
When to use Secondary Index ?

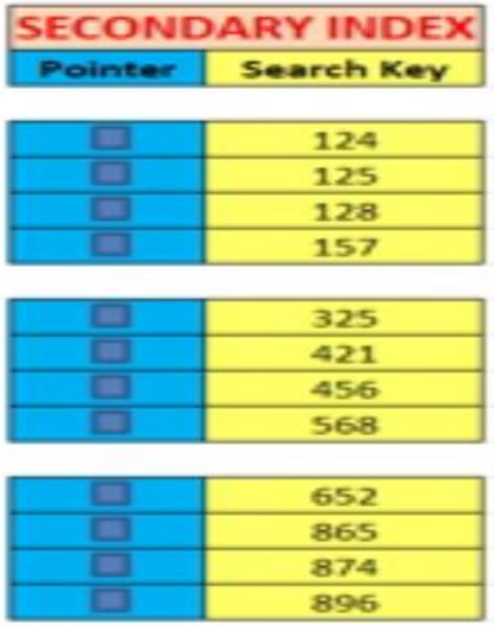
1. Unordered Data
2. Non-Key or Key based search required.
3. Data already have Primary Index

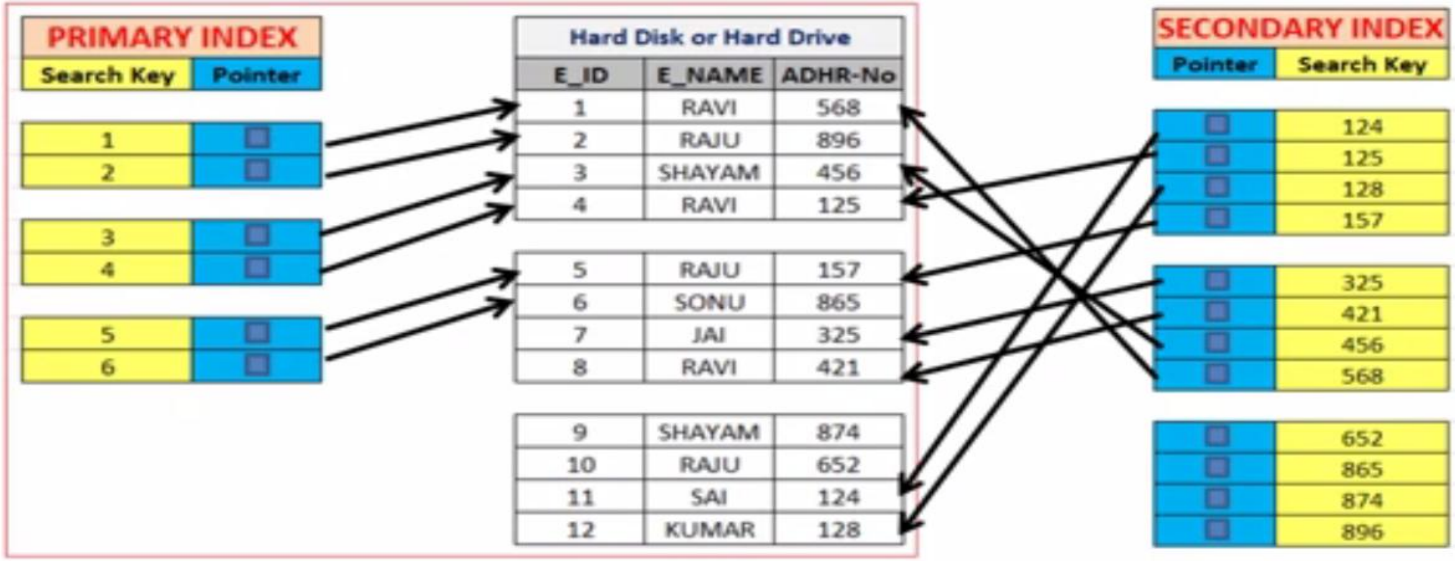
- Data searching requires MORE TIME as compared to the clustered index
- Secondary Index is of DENSE INDEX type, where ALL SEARCH KEY VALUE pointing to specific block.

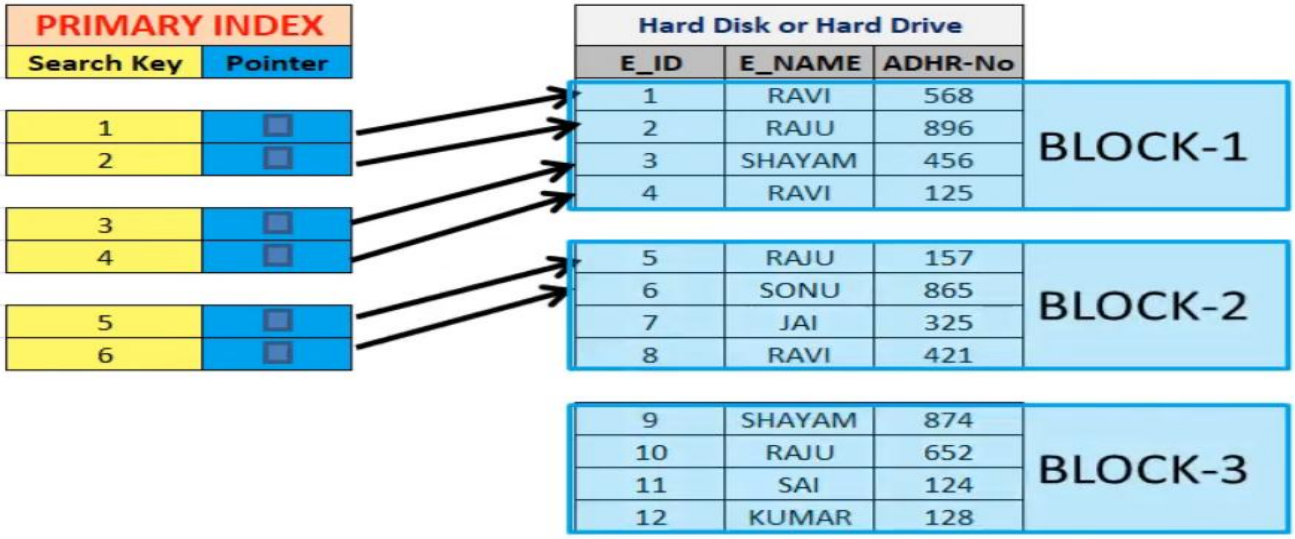
INDEX – SECONDARY INDEX

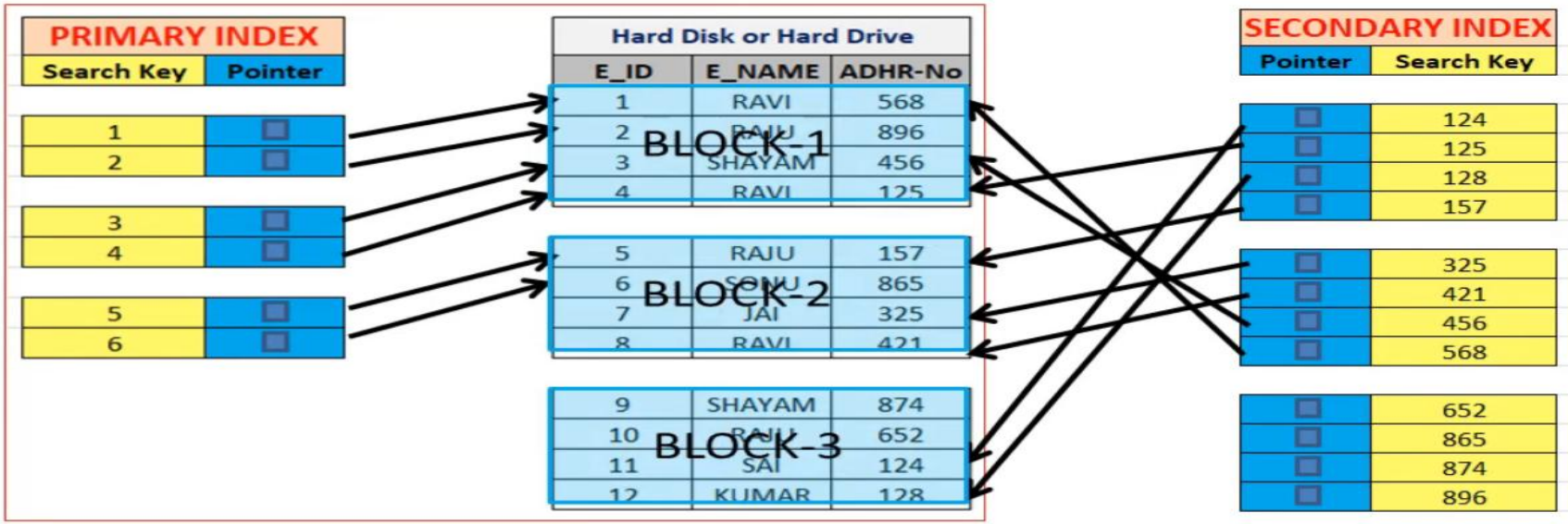
When you need to search data OTHER THAN KEY OF PRIMARY INDEX, such as Aadhar Card No, which is also a UNIQUE KEY IN DATABASE.





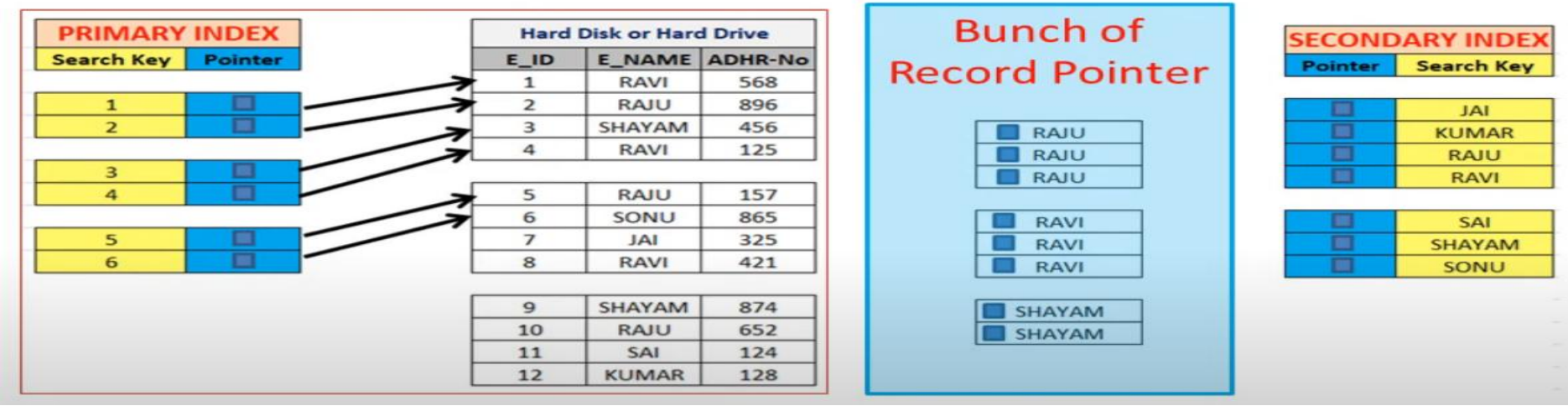


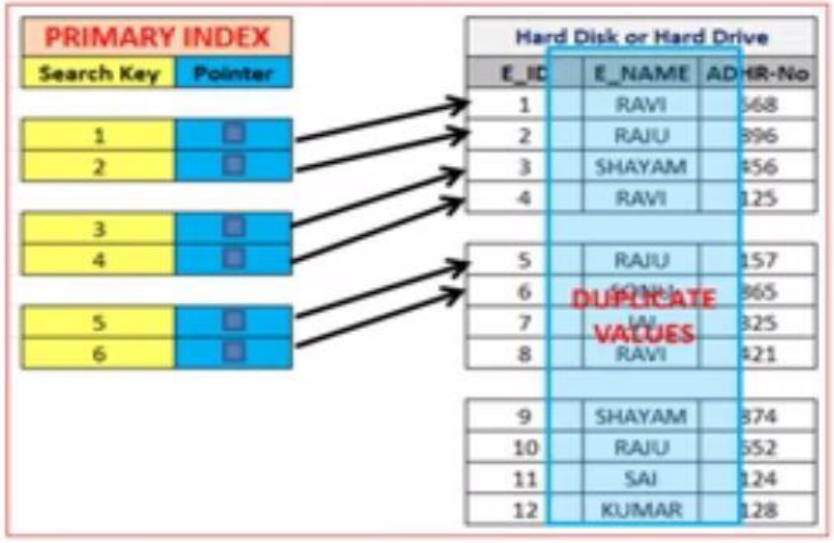

PRIMARY INDEX
Search Key
Pointer
1
2
3
4
5
6
Hard Disk or Hard Drive
E_ID
E_NAME
ADHR-No
1 RAVI 568
2 RAJU 896
3 SHAYAM 456
4 RAVI 125
BLOCK-1
5 RAJU 157
6 SONU 865
7 JAI 325
8 RAVI 421
BLOCK-2
9 SHAYAM 874
10 RAJU 652
11 SAI 124
12 KUMAR 128
BLOCK-3

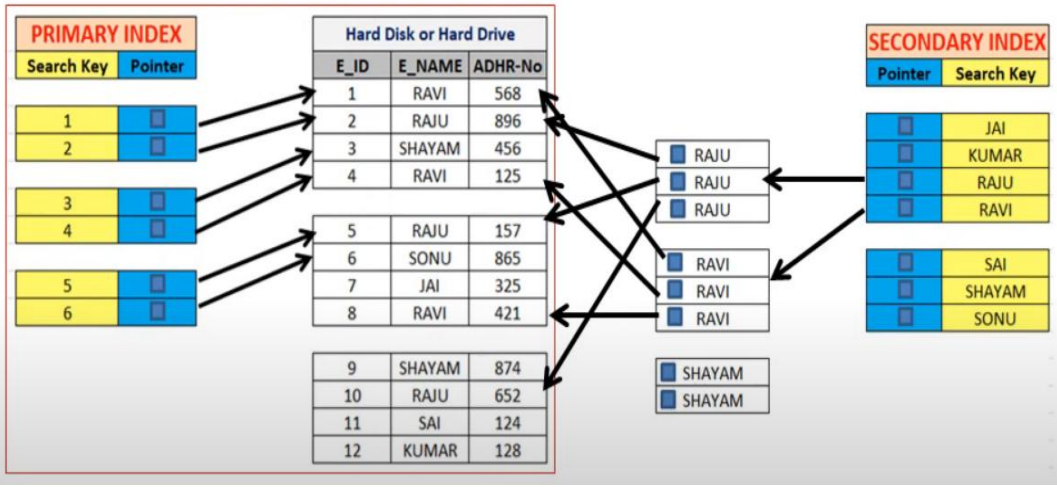

PRIMARY INDEX
Search Key
Pointer
1
2
3
4
5
6
Hard Disk or Hard Drive
E_ID
E_NAME
ADHR-No
1 RAVI 568
2 RAJU 896
3 SHAYAM 456
4 RAVI 125
BLOCK-1
5 RAJU 157
6 SONU 865
7 JAI 325
8 RAVI 421
BLOCK-2
9 SHAYAM 874
10 RAJU 652
11 SAI 124
12 KUMAR 128
BLOCK-3
SECONDARY INDEX
Pointer
Search Key
124
125
128
157
325
421
456
568
652
865
874
896

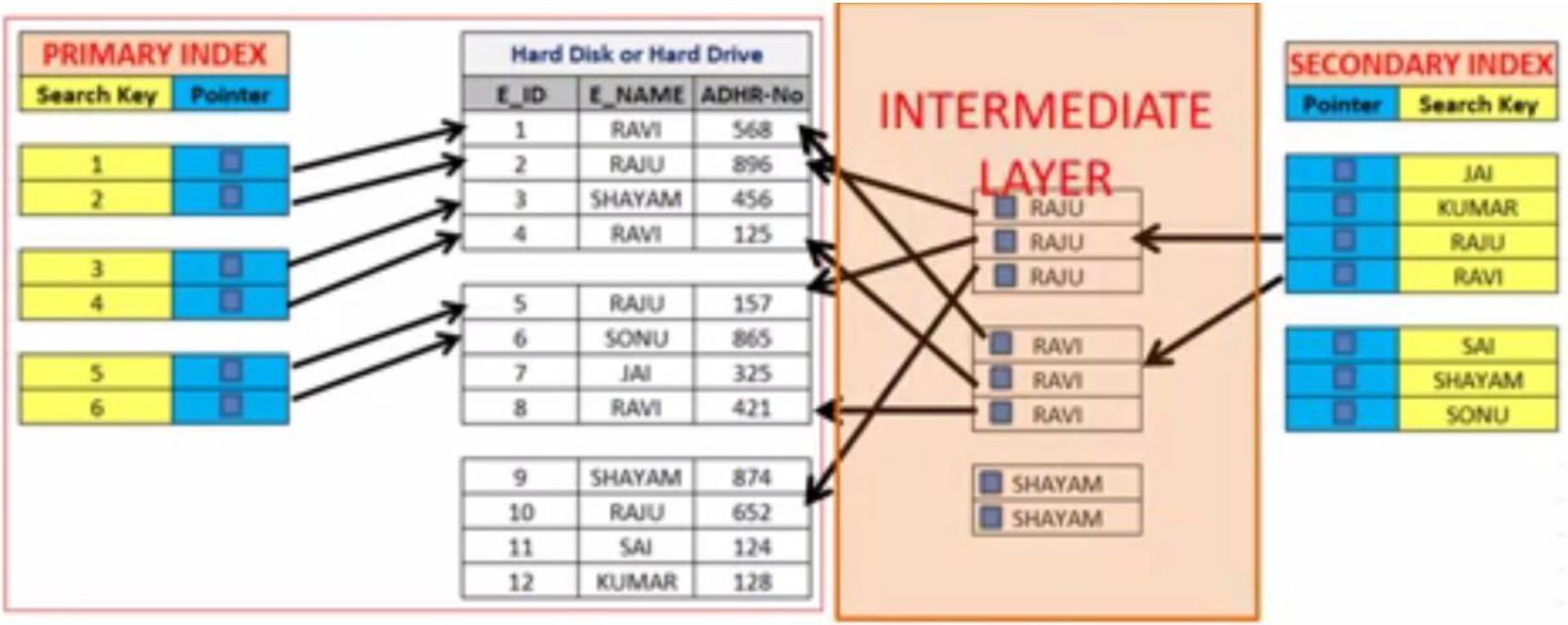
Index- Secondary Index

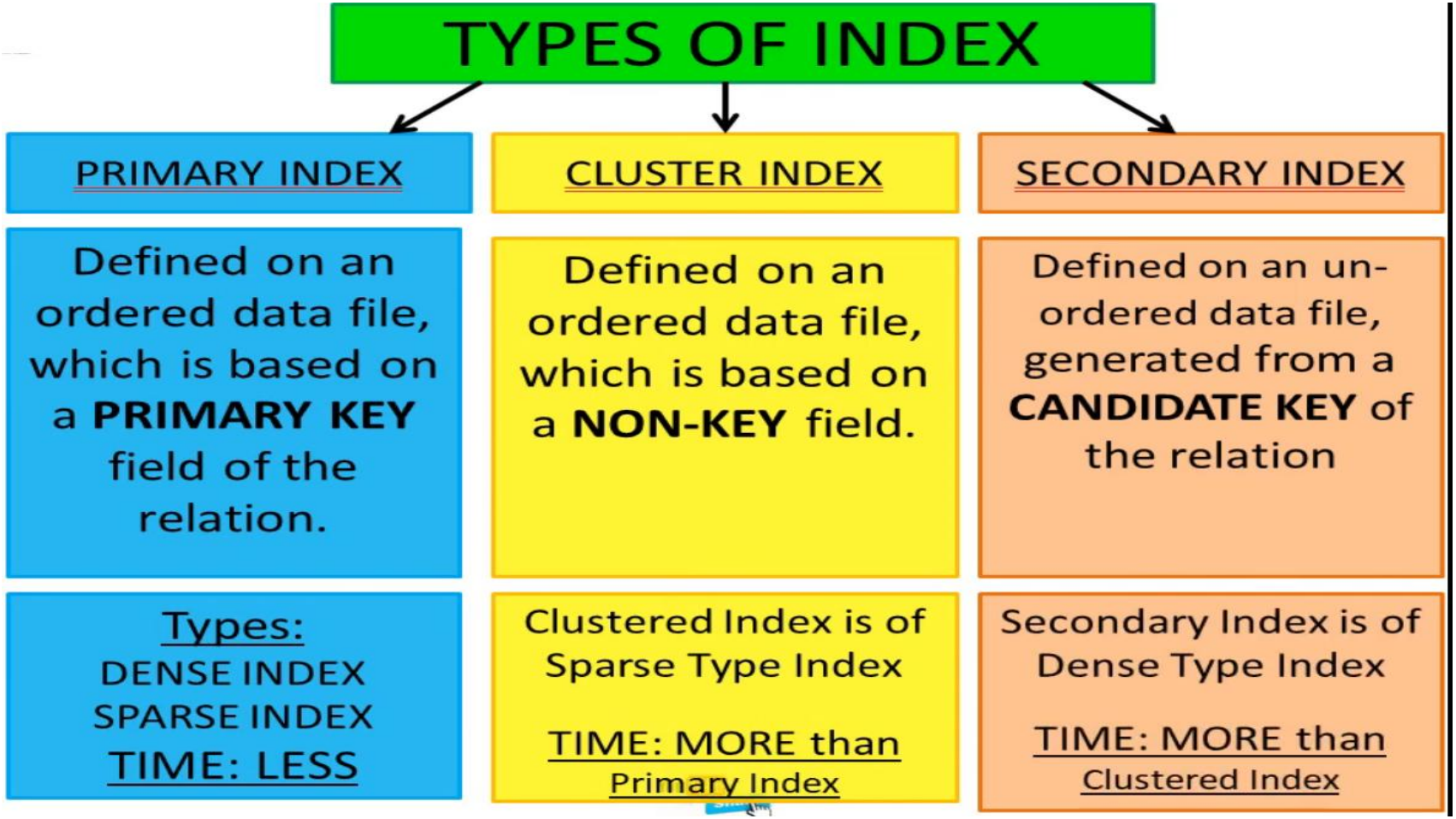
When you need to search data OTHER THAN KEY OF PRIMARY INDEX, such as E_Name, which is NON KEY and having duplicate values in table of a DATABASE.





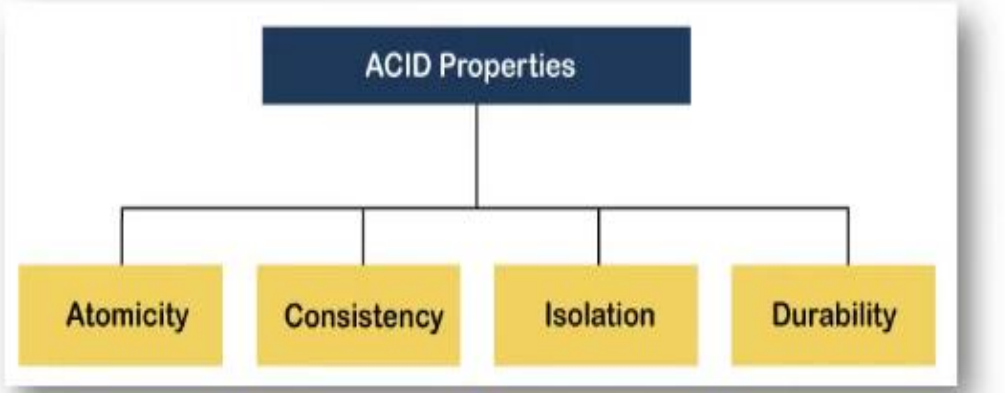



PRIMARY INDEX
Search Key
Pointer
1
2
3
4
5
6
Hard Disk or Hard Drive
E_ID
E_NAME
ADHR-No
1 RAVI 568
2 RAJU 896
3 SHAYAM 456
4 RAVI 125
5 RAJU 157
6 SONU 865
7 JAI 325
8 RAVI 421
9 SHAYAM 874
10 RAJU 652
11 SAI 124
12 KUMAR 128
INTERMEDIATE LAYER
RAJU
RAJU
RAJU
RAVI
RAVI
RAVI
SHAYAM
SHAYAM
SECONDARY INDEX
Pointer
Search Key
JAI
KUMAR
RAJU
RAVI
SAI
SHAYAM
SONU


TYPES OF INDEX
PRIMARY INDEX
CLUSTER INDEX
SECONDARY INDEX
Defined on an ordered data file, which is based on a PRIMARY KEY field of the relation.
Defined on an ordered data file, which is based on a NON-KEY field.
Defined on an un-ordered data file, generated from a CANDIDATE KEY of the relation
Types:
DENSE INDEX
SPARSE INDEX
TIME: LESS
Clustered Index is of Sparse Type Index
TIME: MORE than Primary Index
Secondary Index is of Dense Type Index
TIME: MORE than Clustered Index

About ACID Properties

- It is important to ensure that the database remains consistent before and after the transaction.
- To ensure the consistency of database, certain properties are followed by all the transactions occurring in the system.
- These properties are called as ACID Properties of a transaction.

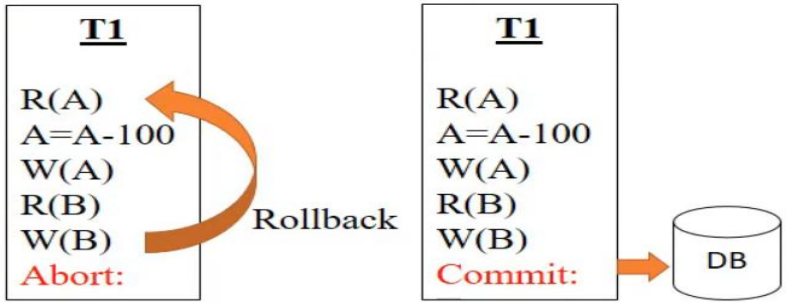


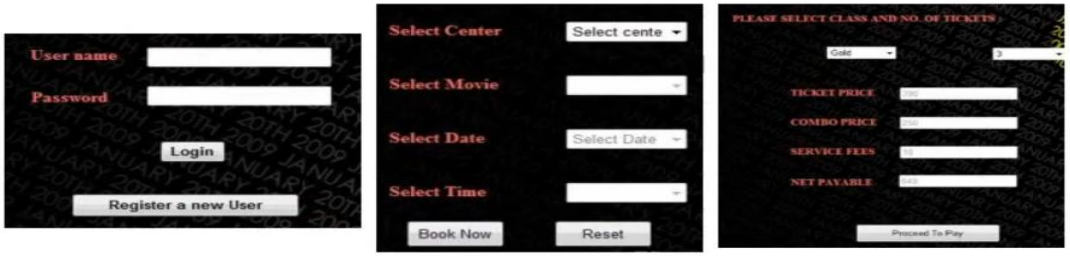
1. Atomicity

➢ **(ALL or NONE)**

- This property ensures that either the transaction occurs completely or it does not occur at all.
- In other words, it ensures that no transaction occurs partially.
- It is the responsibility of Transaction Control Manager to ensure atomicity of the transactions.
- The operations either get Abort or Commit.

Examples:





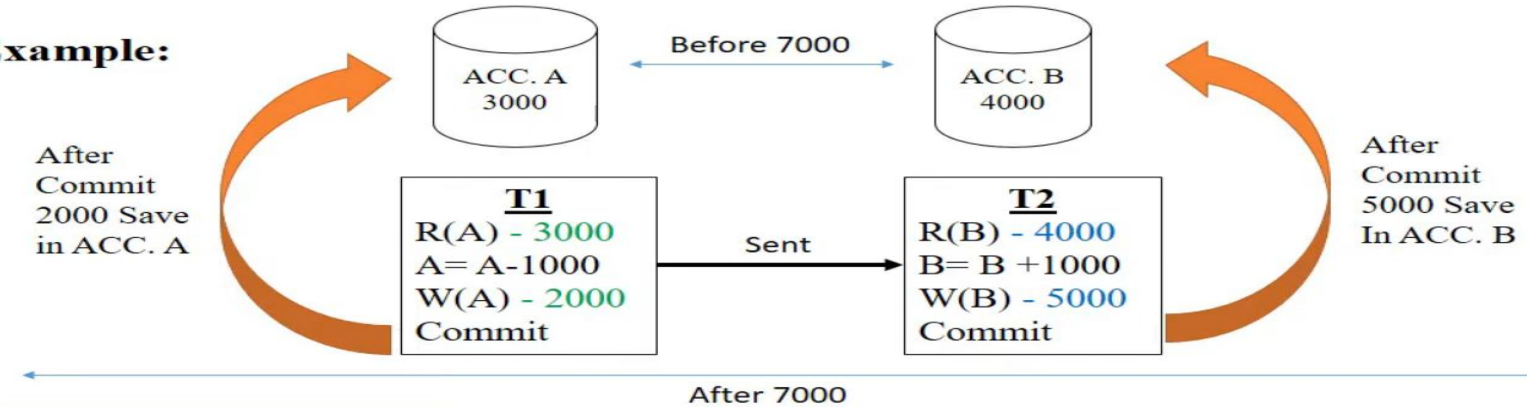
Movie Ticket Booking

2. Consistency

➢ **(Before & After the Transection, Sum same)**

- This property ensures that integrity constraints are maintained.
- It ensures that the database remains consistent before and after the transaction.
- It is the responsibility of DBMS application programmer to ensure consistency of the database.

Example:

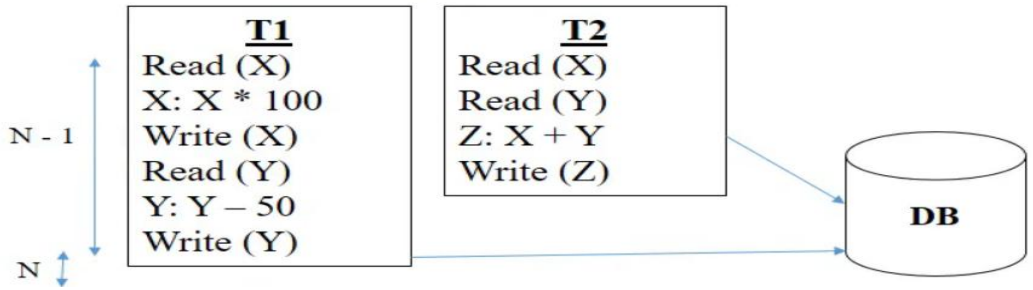


3. Isolation

➢ **(Multiple Transections Occur Independently)**

- The term 'isolation' means separation.
- It means if two operations are being performed on two different databases, they may not affect the value of one another.
- In the case of transactions, when two or more transactions occur simultaneously, the consistency should remain maintained.
- Any changes that occur in any particular transaction will not be seen by other transactions **until** the change is not committed in the memory.

Example:



4. Durability

➢ **(Successful Transection / Permanently Data Stored even if failure occures)**

- Durability ensures the permanency of something.
- This property ensures that all the changes made by a transaction after its successful execution are written successfully to the disk.
- It also ensures that these changes exist permanently and are never lost even if there occurs a failure of any kind.
- The COMMIT command must be used every time we make changes.
- It is the responsibility of Recovery Manager to ensure durability in the database.

About Concurrency Control

- Multiple users can access and use the same database at one time, which is known as the concurrent execution of the database.
- It ensures that Database transactions are performed concurrently and accurately.
- It confirms that produce correct results without violating data integrity of the respective Database.

Example:

✓Two people buy a movie ticket for the same movie and the same show time.

✓There is only one seat left, for the movie show in that particular theatre.

✓Concurrency Problem occur.

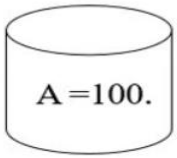
✓Without concurrency control it is not possible to buy ticket.

✓It provides a ticket to the buyer who has completed the transaction process first.

Concurrency Control Problems

1. **Lost Update Problem (W – W Problem):** It occur when multiple transactions select the same row and update the row based on the value selected.

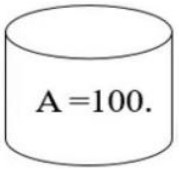
Example:



T1	T2
R(A)	
A = A-50 A=50	
	R(A)
	A=A+100 A=200
W(A)	
	W(A)

2. **Dirty Read Problem:** When one transaction updates an item of the database, and somehow the transaction fails, and before the data gets rollback, the updated database item is accessed by another transaction.

Example:

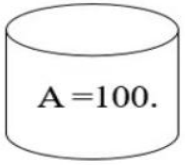


Rollback

T1	T2
R(A)	
A = A+50 A=150	
W(A)	
	R(A) A=150
Fail	

3. Unrepeatable Read Problem: Also known as Inconsistent Retrievals Problem that occurs when in a transaction, two different values are read for the same database item.

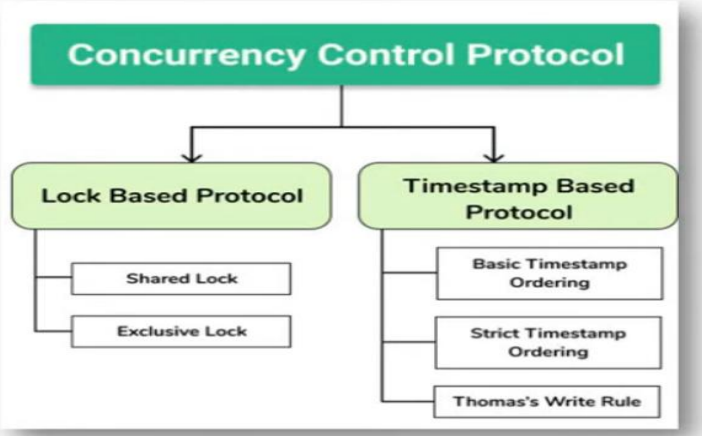
Example:



T1	T2
R(A) A = 100	
	R(A)
	A=A+100 A = 200
	W(A)
R(A) A = 200	

Concurrency Control

- Concurrency Control is the working concept that is required for controlling and managing the concurrent execution of database operations.
- It avoiding the inconsistencies in the database.
- The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions



Use of Concurrency Control Methods

1. To apply Isolation through mutual exclusion between conflicting transactions
2. To resolve read-write and write-write conflict issues
3. To preserve database consistency through constantly preserving execution obstructions
4. The system needs to control the interaction among the concurrent transactions.
5. Concurrency control helps to ensure serializability.
6. Maintain integrity of the database.

What is Non Serializable Schedule?

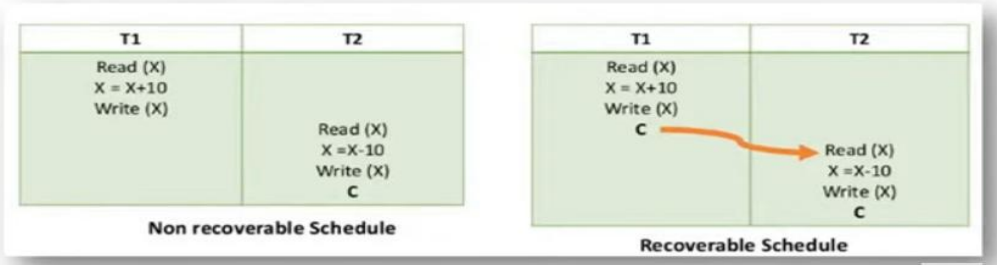
- A non-serial schedule which is not serializable is called as a non-serializable schedule.
- It has **two types-** Recoverable & Not Recoverable Schedule.

Characteristics:

1. May or may not be consistent
2. May or may not be recoverable

Hence,

No guarantee to work like serial schedule.



Non - Recoverable Schedule

- T1 perform operations.
- T2 performs a dirty read operation. (T1 -> T2)
- T2 commits before T1.
- T1 fails later and roll backs.
- The value that T2 read now stands to be incorrect.
- T2 can not recover since it has already committed.

DB
A = 100

Schedule S

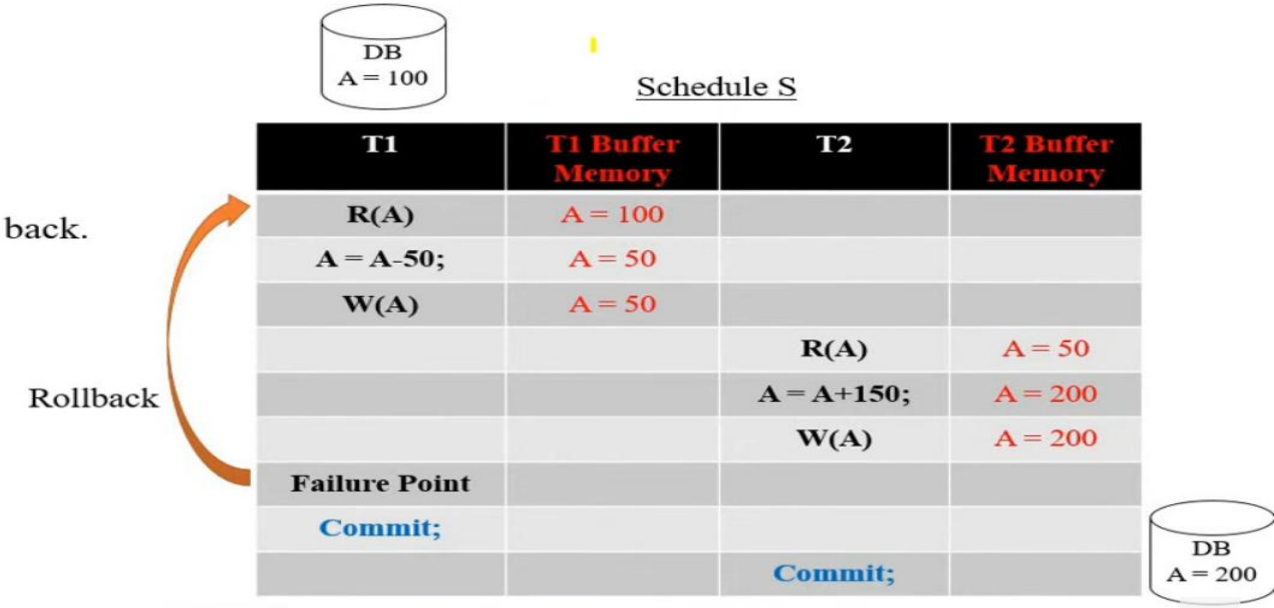
T1	T1 Buffer Memory	T2	T2 Buffer Memory
R(A)	A = 100		
A = A-50;	A = 50		
W(A)	A = 50		
		R(A)	A = 50
		A = A+150;	A = 200
		W(A)	A = 200
		Commit;	
Failure Point			
Commit;			

Rollback

DB
A = 200

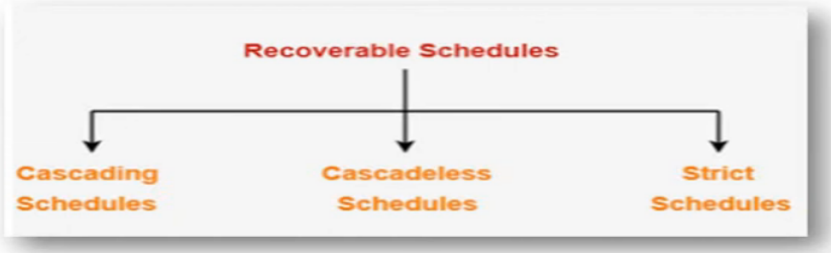
Recoverable Schedule

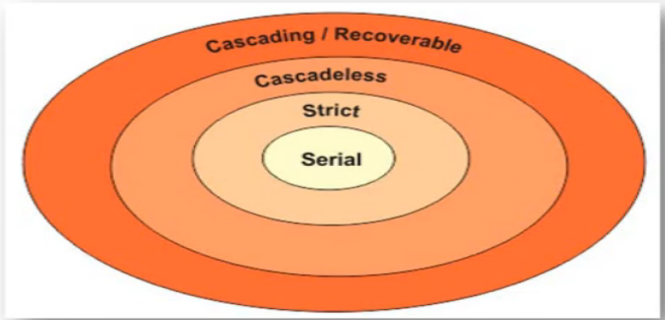
- T1 perform operations.
- T2 performs a dirty read operation.
- T1 failure occurred, get Rollback.
- T1 commits first.
- T2 is now allowed to commit.
- In case, T1 would have failed,
  T2 has a chance to recover by rolling back.



Recoverable Schedule

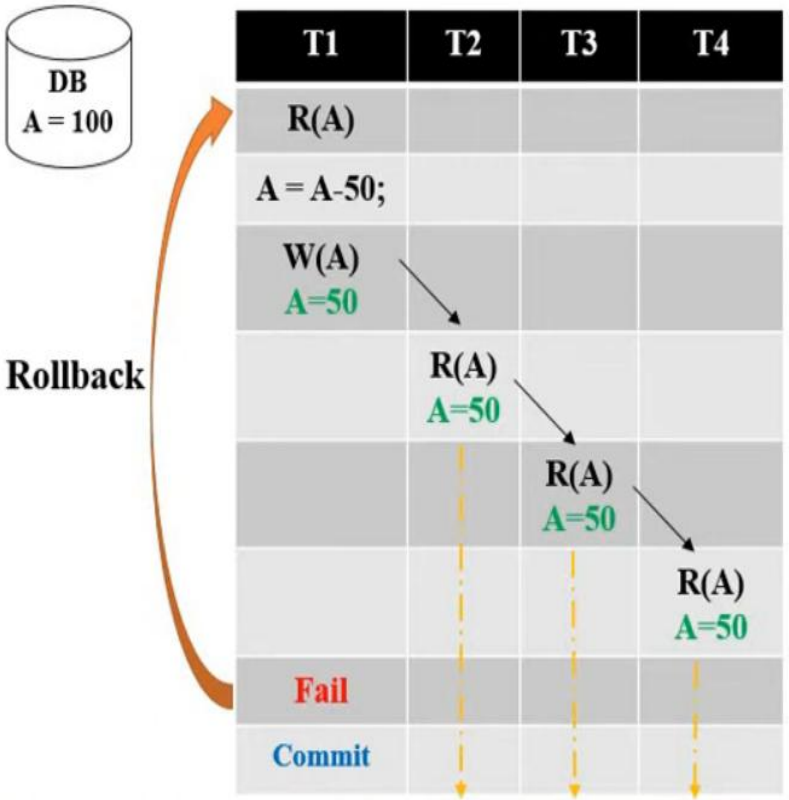
- Sometimes a transaction may not execute completely due to a software issue, system crash or hardware failure.
- In that case, the failed transaction has to be rollback.
- But some other dependent transaction may also have used value produced by the failed transaction.
- So we also have to rollback those transactions.





Cascading Schedule

- If in schedule, One transection fail then all the other dependent transection also rollback or fail / abort such a schedule called as Cascading Schedule or Cascading Rollback or Cascading Abort.
- It simply leads to the wastage of CPU time.



Here,

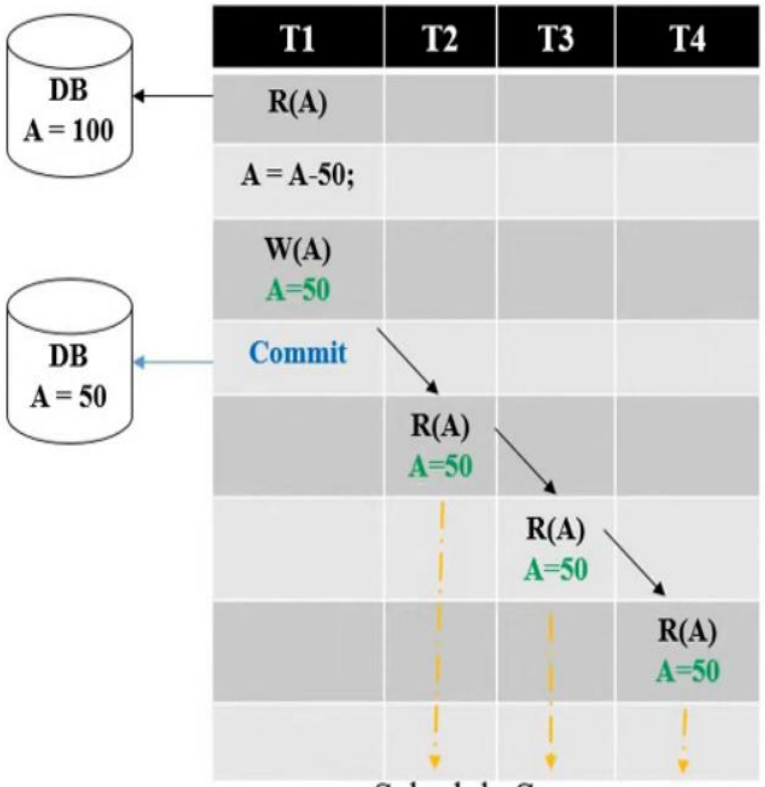
- Transaction T2 depends on transaction T1.
- Transaction T3 depends on transaction T2.
- Transaction T4 depends on transaction T3.

In this schedule,

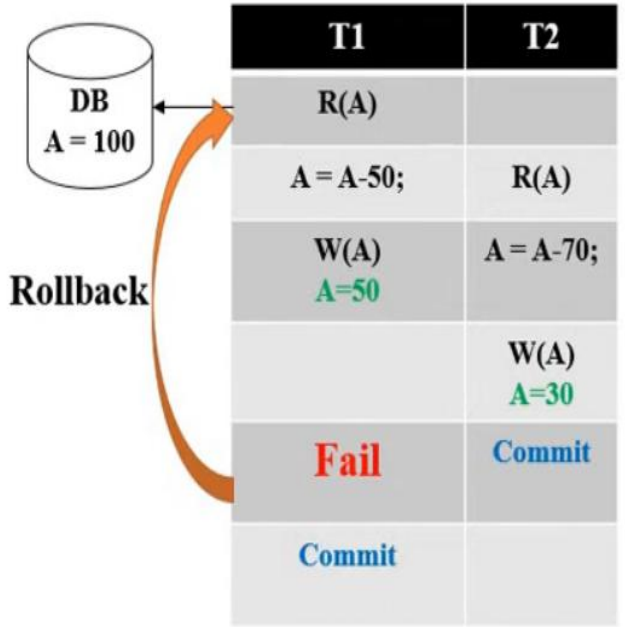
- ✓ The failure of transaction T1 causes the transaction T2 to rollback.
- ✓ The rollback of transaction T2 causes the transaction T3 to rollback.
- ✓ The rollback of transaction T3 causes the transaction T4 to rollback.

Cascadless Schedule

- If in schedule, **transaction is not allowed to read a data item** until the **last transaction that has written** it is committed or aborted, then such a schedule called as a **Cascadeless Schedule**.
- It avoids cascading roll back and thus saves CPU time.



Schedule S



Lost Updation Problem / W – W Problem

Strict Schedule

- Here, Transaction is neither allowed to read nor write a data item until the last transaction that has written it is committed or aborted.

Characteristics:

1. Strict schedule allows only committed read and write operations.
2. Clearly, strict schedule implements more restrictions than cascadeless schedule.

Example:

- Transection T1 perform the operations & commit it.
- Transaction T2 reads and writes the value of transaction T1 only after the transaction T1 commits.

T1	T2
R(A)	
W(A)	
Commit	
	W(A)
	R(A)
	Commit

Strict Schedule

Correlation between all schedules

T1	T2
R(A)	
W(A)	
Commit	
	W(A)
	R(A)
	Commit

Strict Schedule

T1	T2
R(A)	
W(A)	R(B)
Commit	
	W(A)
	R(A)
	Commit

Strict Schedule but
Not Serial Schedule

T1	T2
R(A)	
W(A)	
Commit	
	R(A)
	W(A)
	Commit

Strict , Serial Schedule but
Cascadless Schedule