

---

---

---

---

---



DBMS  
( Data Base Management System)

Data? → Bits | Bytes → in your memory

Image → { Collection of bytes }  
(Raw) → no meaning  
2 bits  
(0, 1)

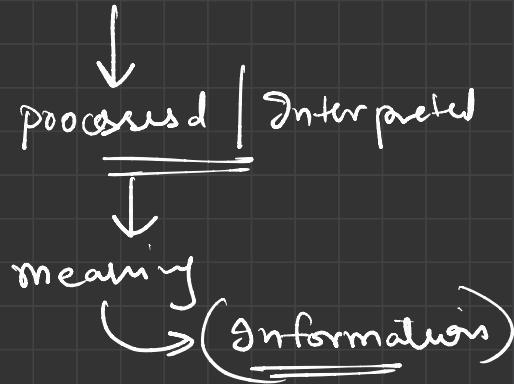
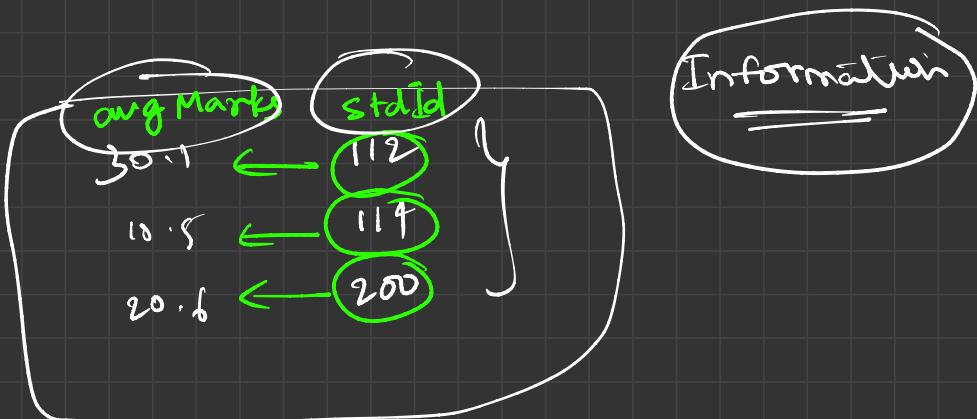
$$8 \text{ Bit} = 1 \underline{\text{Byte}}$$

$$2^8 = 256$$

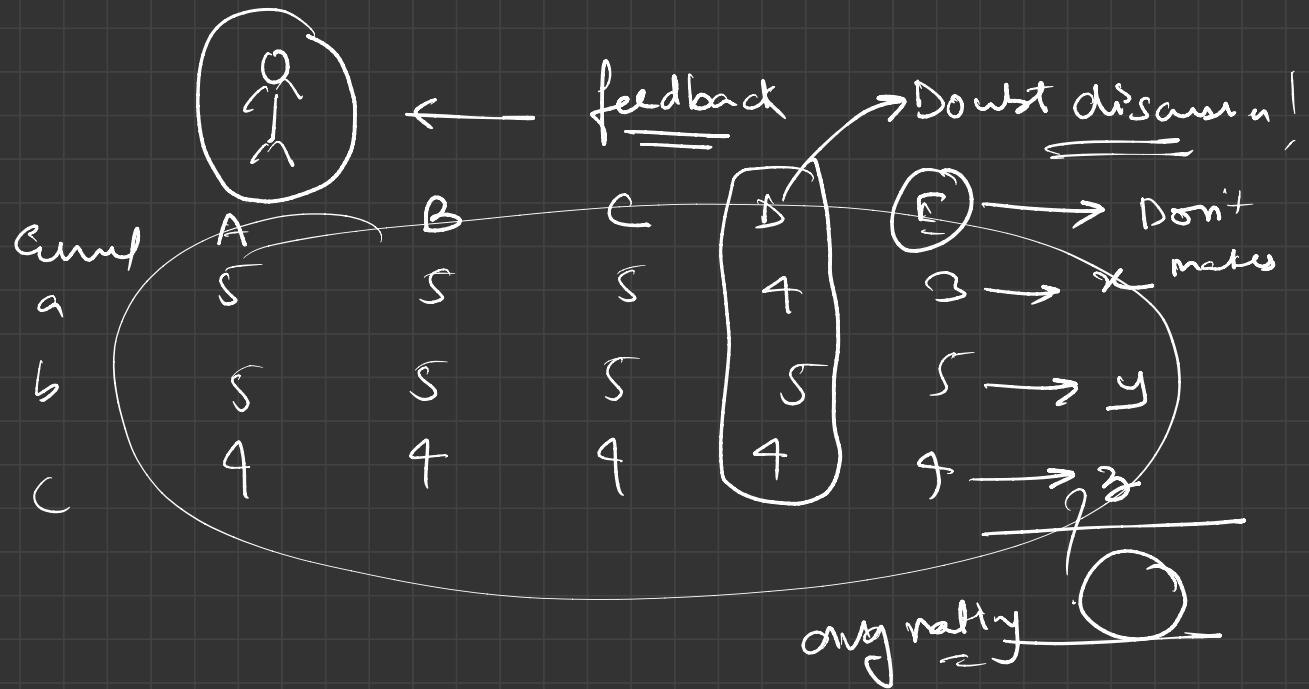
↓  
(RGB)  
+

(0 - 255)

Integers → Collection of Bytes

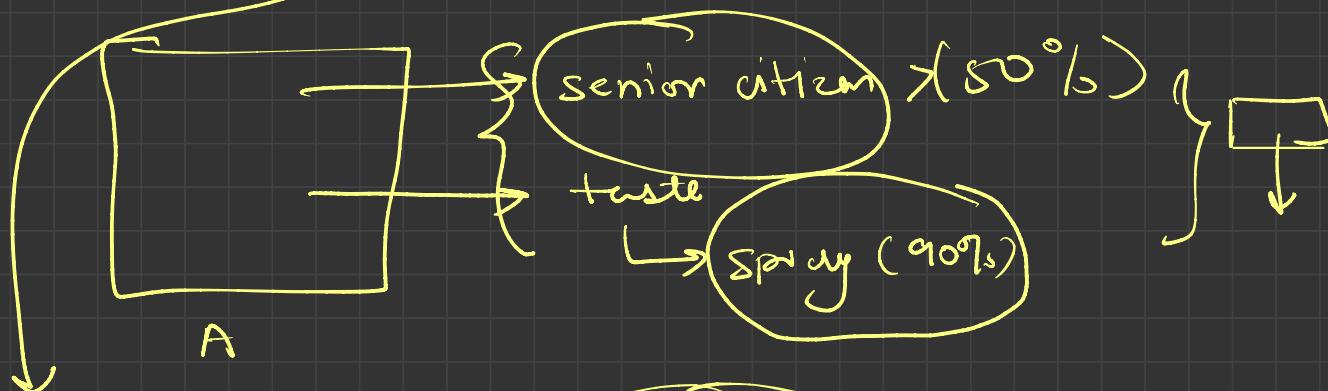


## Actions

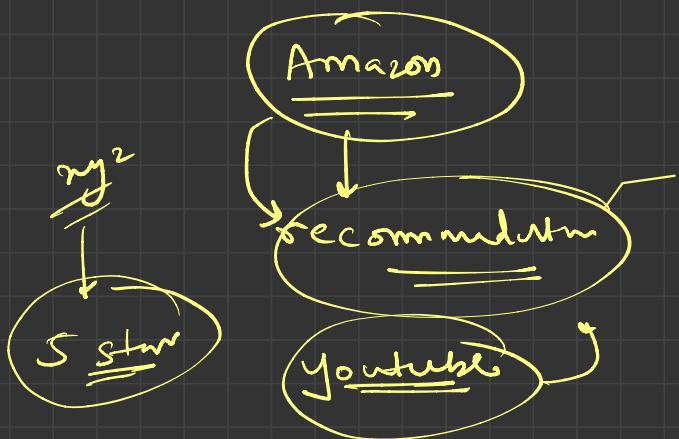


Info: is used to make decision.

"Data" in new oil"



{ Sex ratio  
Unmarried }



Data Base ( db )



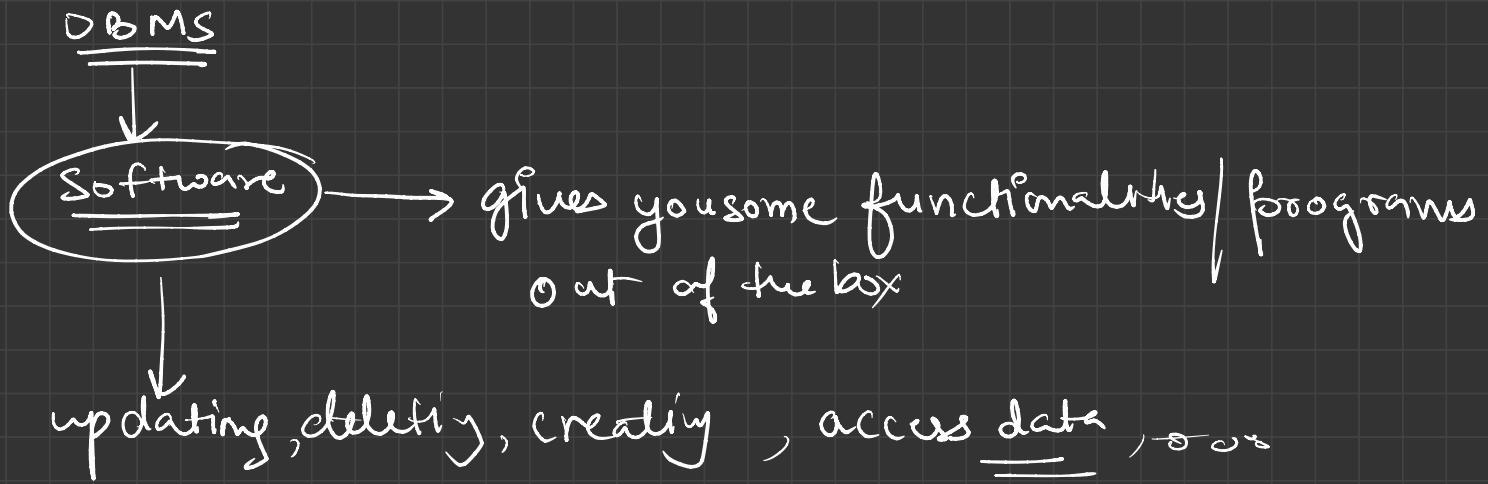
is a place where you store all the data.

memory in your computer  
in in in phone }

Memory

google drive → stores your things



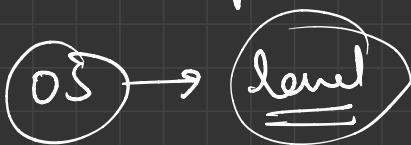
FS Module

{ → Oracle  
→ MySQL  
→ MS SQL Server  
→ IBM DB2  
→ PostgreSQL }

# File System Module



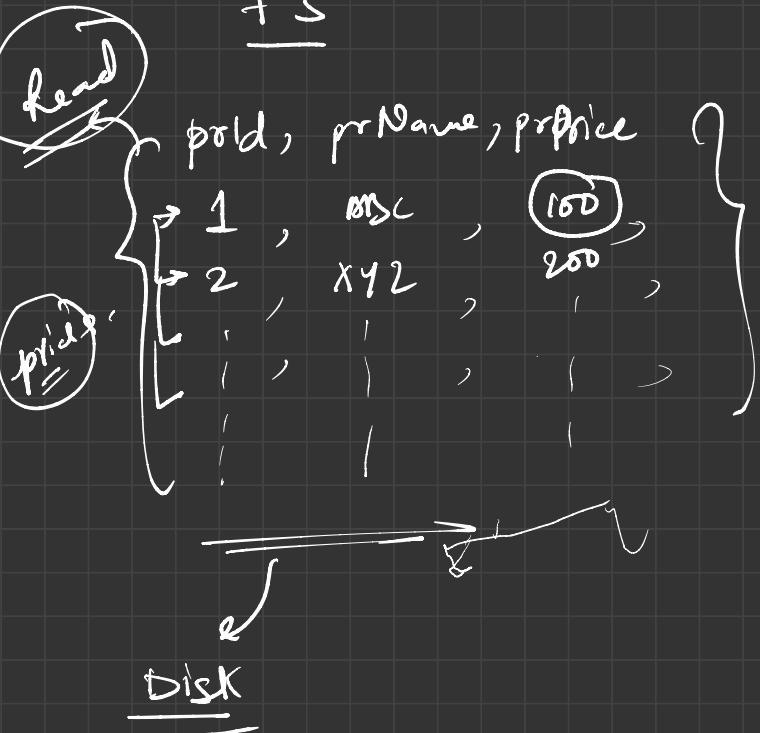
{ read , delete , create New files in your computer .



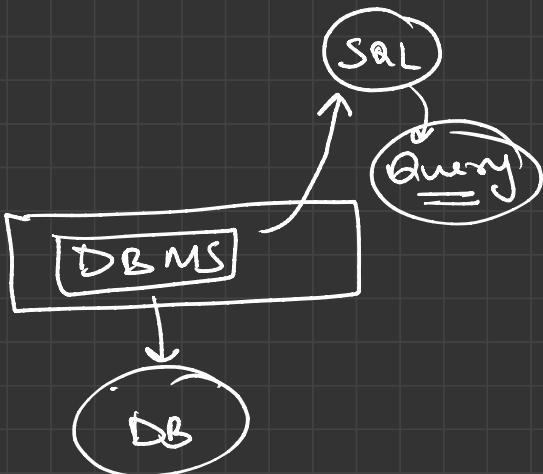
# ① Querying

point me all the products  $> \underline{\underline{100}}$

FS



DBMS



- ① No efforts in logic
- ② faster

② Redundancy

Cust Info

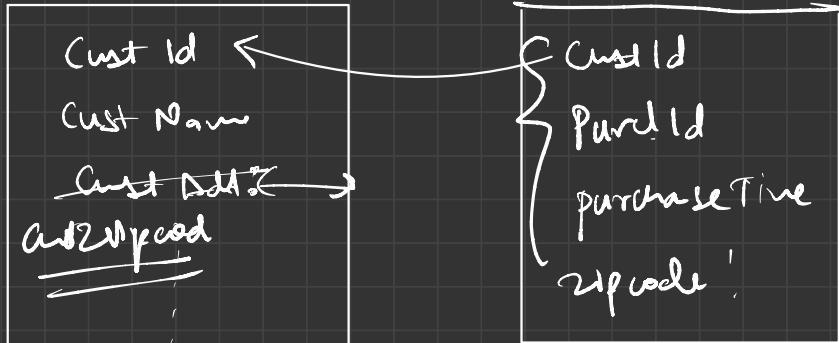
Purchase Info

repeatedly

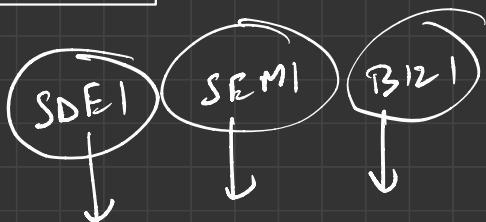
Story

Same

Info

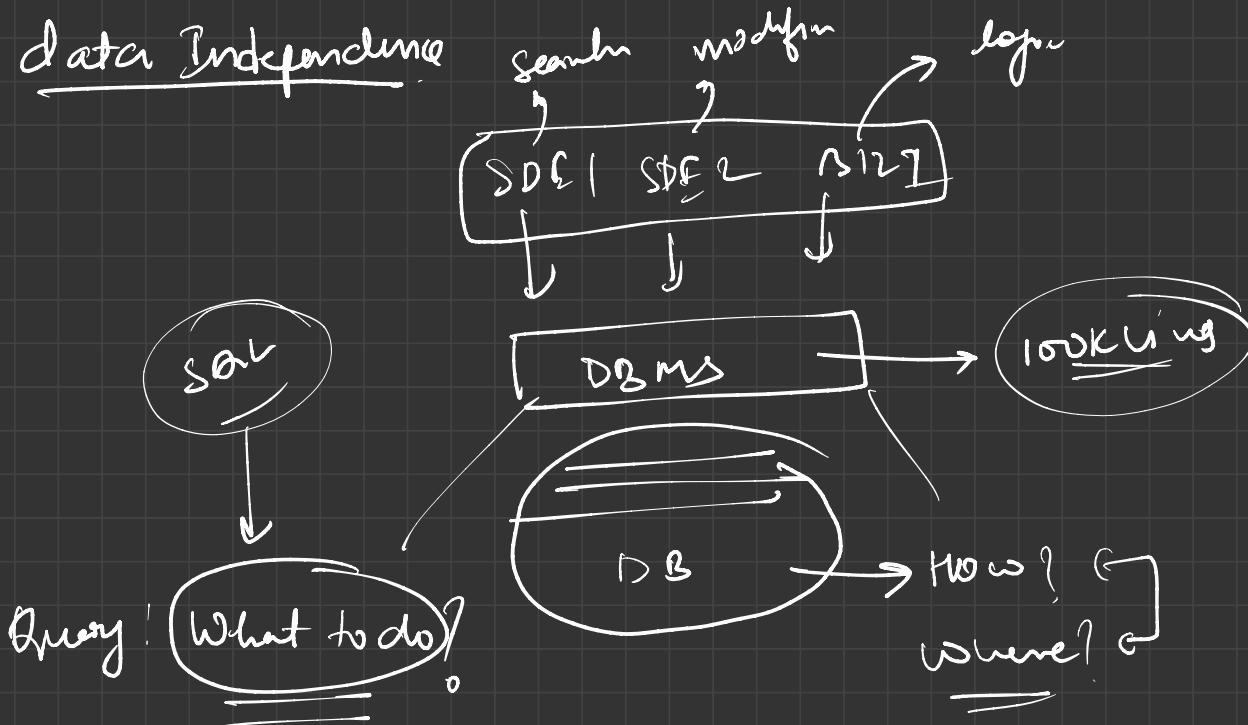


Database



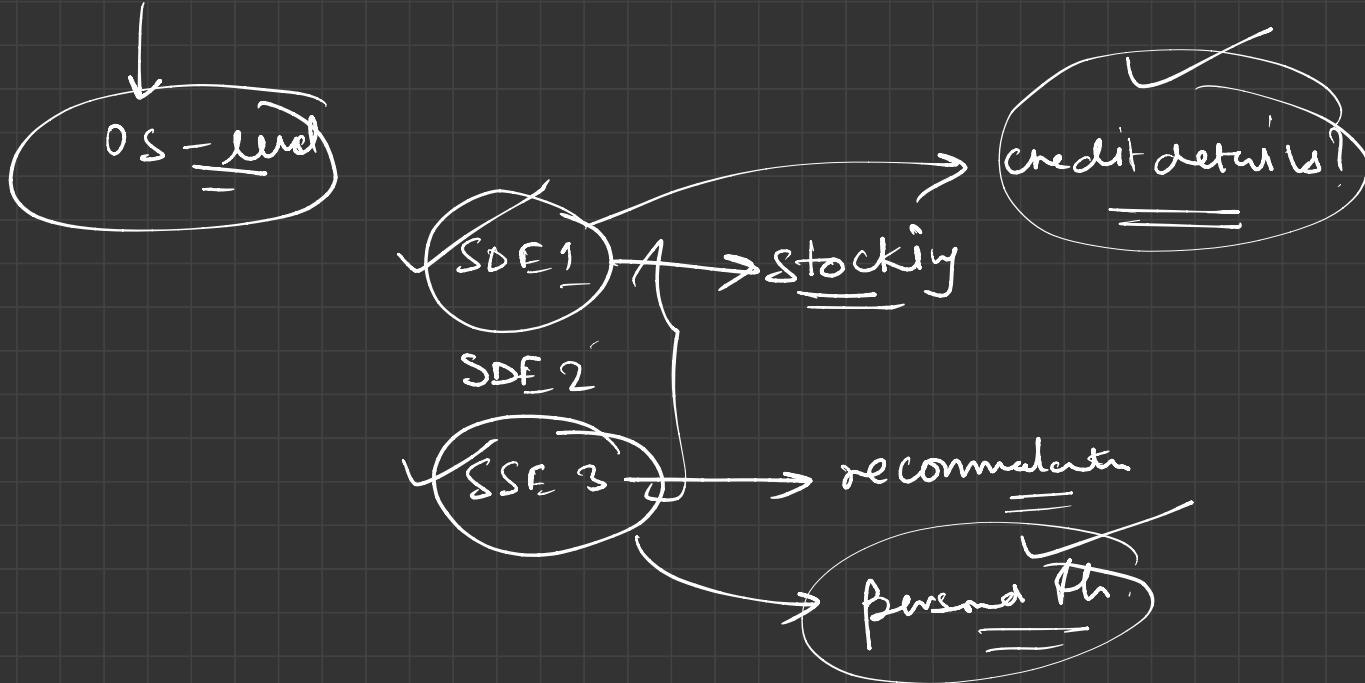
### ③ Consistency

### ④ Data Independence



Hiding low level details from the engineers using the DBMS

## ⑤ Security & Access Control



## Different types of DBMS :

- ① Relational DBMS
- ② Non - Relational DBMS
- ③ Graph DBMS

# Terminology

Relational Databases

Relation = Table

Table = Set of Rows & Columns

Table

tuple

Cust Table

columns / fields / attributes

Relation

DB designator

<u>Cust ID</u>	<u>Cust Name</u>	<u>Cust Add</u>	<u>Cust Zipcode</u>
1	abc	mmm	110084
2	def	bbb...	110007
NULL	mno	NULL	

records /  
{}  
Tuples  
row1  
row2

relation

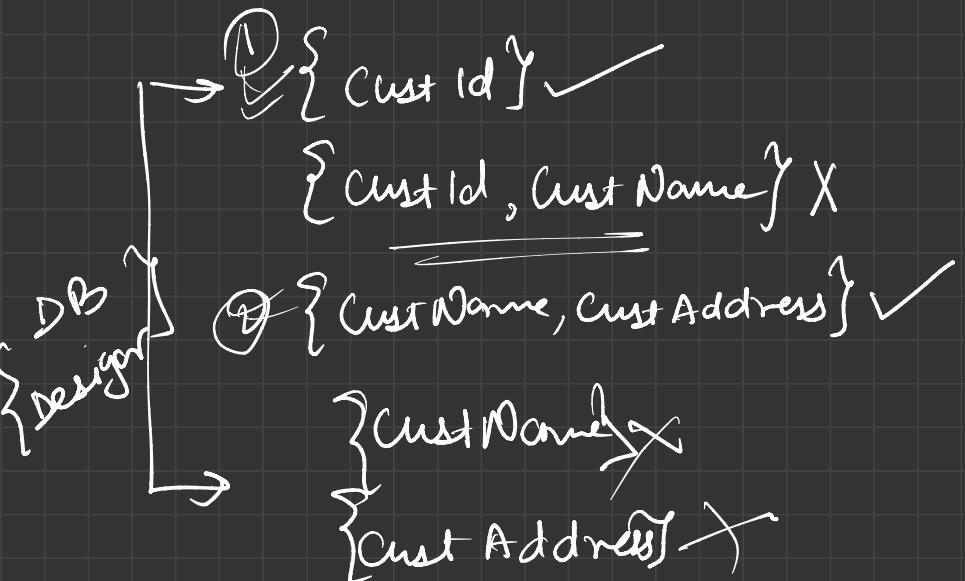
instance

set of tuples / rows

+ table structure!

1	2	3
4	5	6
7	8	9

Keys: minimum set of attributes | columns to uniquely  
identify a row | tuple.



Simple key: Keys with only one attribute | Column.

Cust Id

Compound key: Keys with multiple attributes | Column.

{Cust Name, Cust Address}

Candidate Key: Set of all unique keys

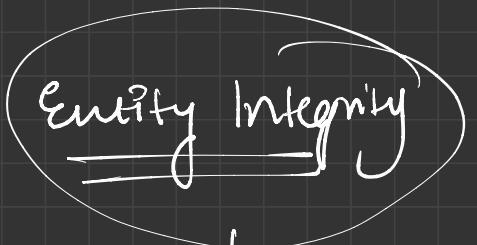
{ custId, { custName, custDob } }

---

Primary key: one of the Candidate key that a DB -designer chooses to maintain uniqueness



- NOT NULL for any row/tuple
- almost one primary key for a Relation
- Unique for each tuple / row



↓  
Rules predefined for a primary key

Alternative / Secondary keys :

Candidate keys that are  
not primary key.

DB design

Relational Schema : Table structure + Integrity Constraints }  
Table            Rules

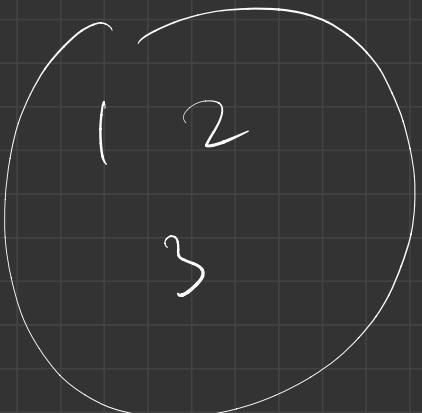
Super Key

Candidate Key  $\cup$  attribute

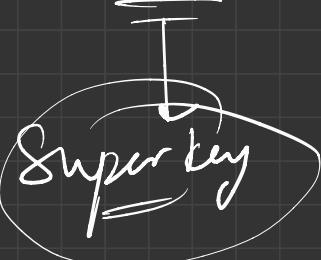
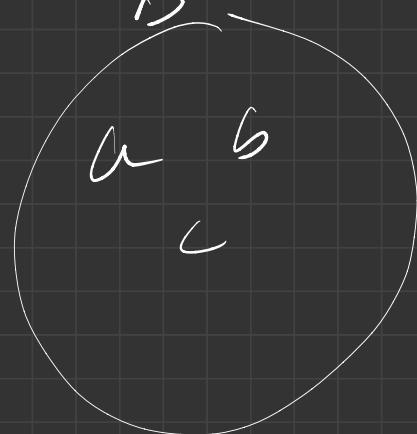
$\{ \text{CustId} \} \cup \{ \text{Zipcode} \}$

$= \{ \text{CustId, Zipcode} \}$

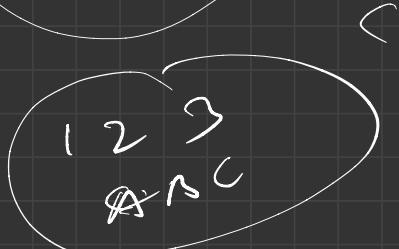
A



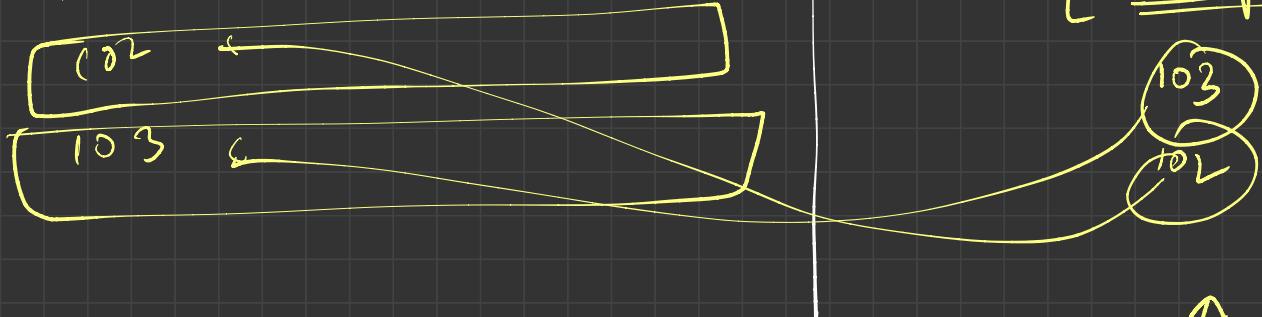
B



$$A \cup B =$$



# Foreign key

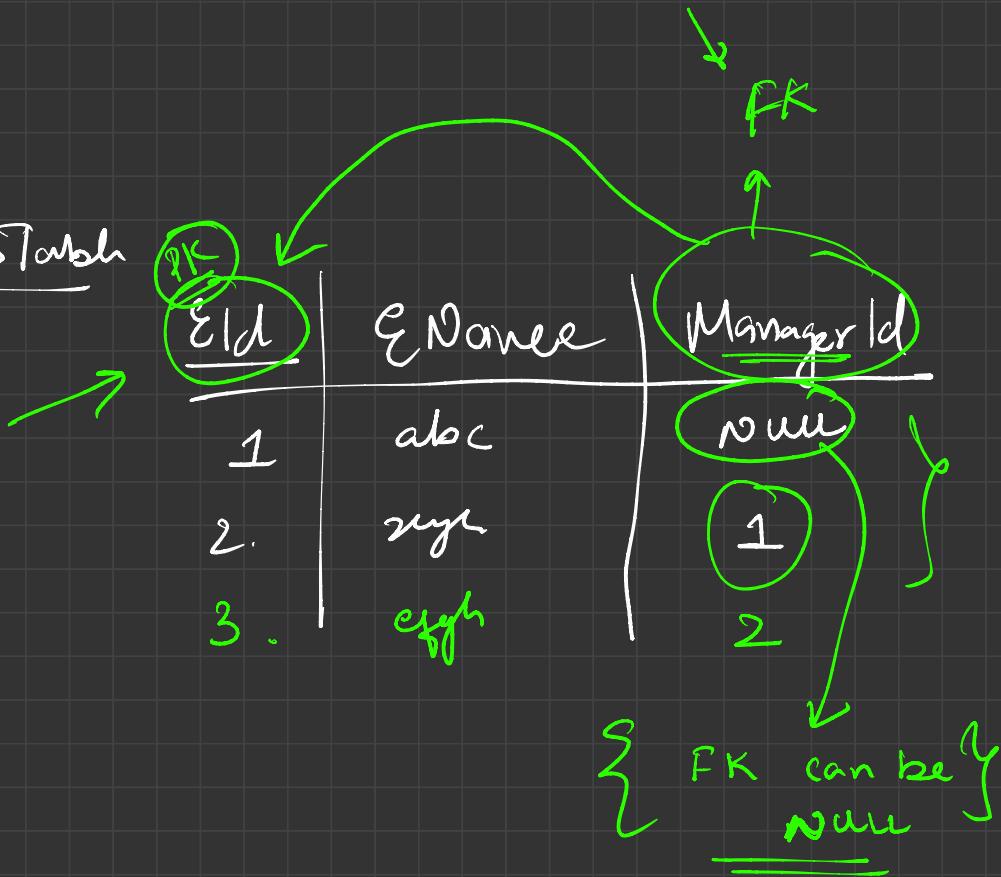


↑  
Integrity  
constraints

Self Referential FK

MicroSoft

Employee Task



{ FK can be  
null }

## Integrity Constraints

① Entity Integrity : Primary keys

- { ✓ every table must have only one PK
- ✓ unique
- ✓ NOT null

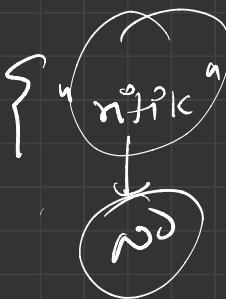
## ② Domain Integrity

CustId = { 1, \_\_\_\_\_ Hatch }

CustId



int



what are valid values for int

### ③ User-defined Integrity

Feedback Table

Id	(A)	(B)	(C)	(D)

$$\text{int} \leq 5 \\ \gamma = 0$$

## ④ Referential Integrity

foreign keys

Cust Table

<u>CustId</u>	CustName	CustAdd
1	abc	1232 -
2	xyz	1994 -
.	.	.
!	!	!

Purchase Table

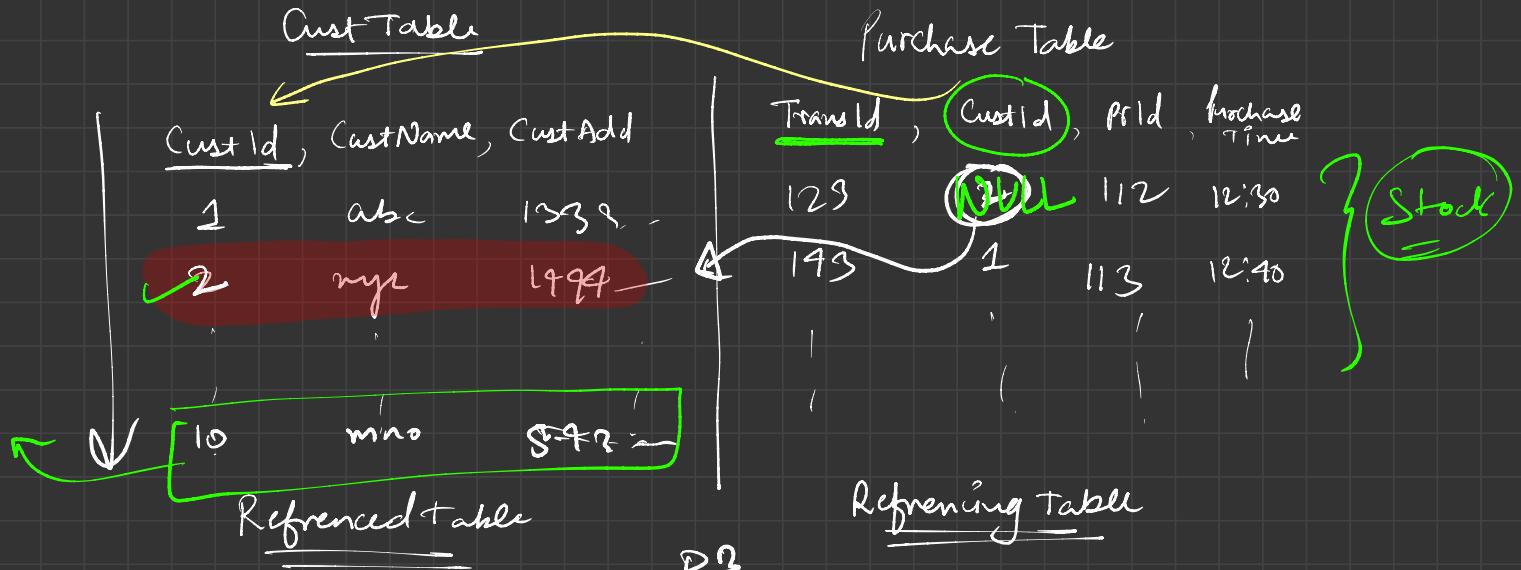
<u>TransId</u>	<u>CustId</u>	PrdId	PurchaseTime
123	2	112	12:30
193	1	113	12:40
1	1	1	1
1	1	1	1

Referenced Table

Referring Table

# Changes to Referenced Table

to move table



DB

① Insert New row →

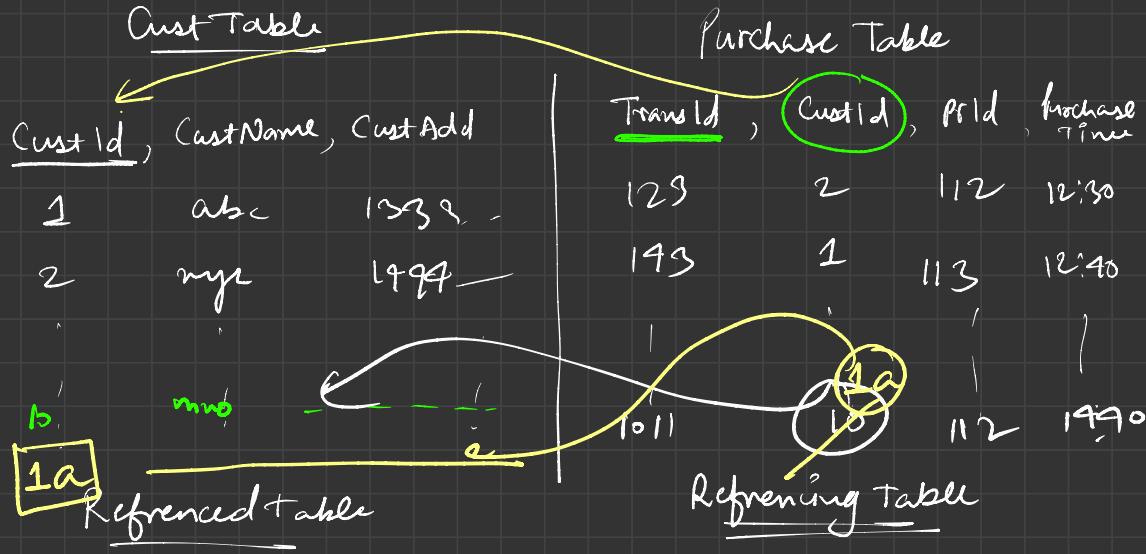
④ On delete - No action → violate

② Delete

⑤ On delete - Cascade → very dangerous

③ On delete → set NULL → safest

# Changes on Referencing Table



Insert : → check for any violation → error

Delete → do nothing

Edit → check for violations?

Examples

Tablet  
relation

<u>A1</u>	<u>A2</u>
2	2
3	4
9	3
2	2
3	5
6	4
10	9

ON - DELETE CASCADE

2, 4  $\leftarrow$  low / tuple

<u>A1</u>	<u>A2</u>
3	4
4	3
6	4

Multilevel  
Cascade

