

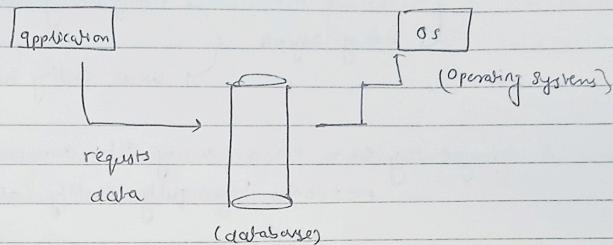
DELTAC
Date / /
Pg No.

~ Database Management Systems ~

+ Database - Collection of related data representing some aspect of the real world → this data is stored such that interrelated remains together

+ DBMS (Database management systems) - Software for storing and retrieving user's data while considering appropriate security measures

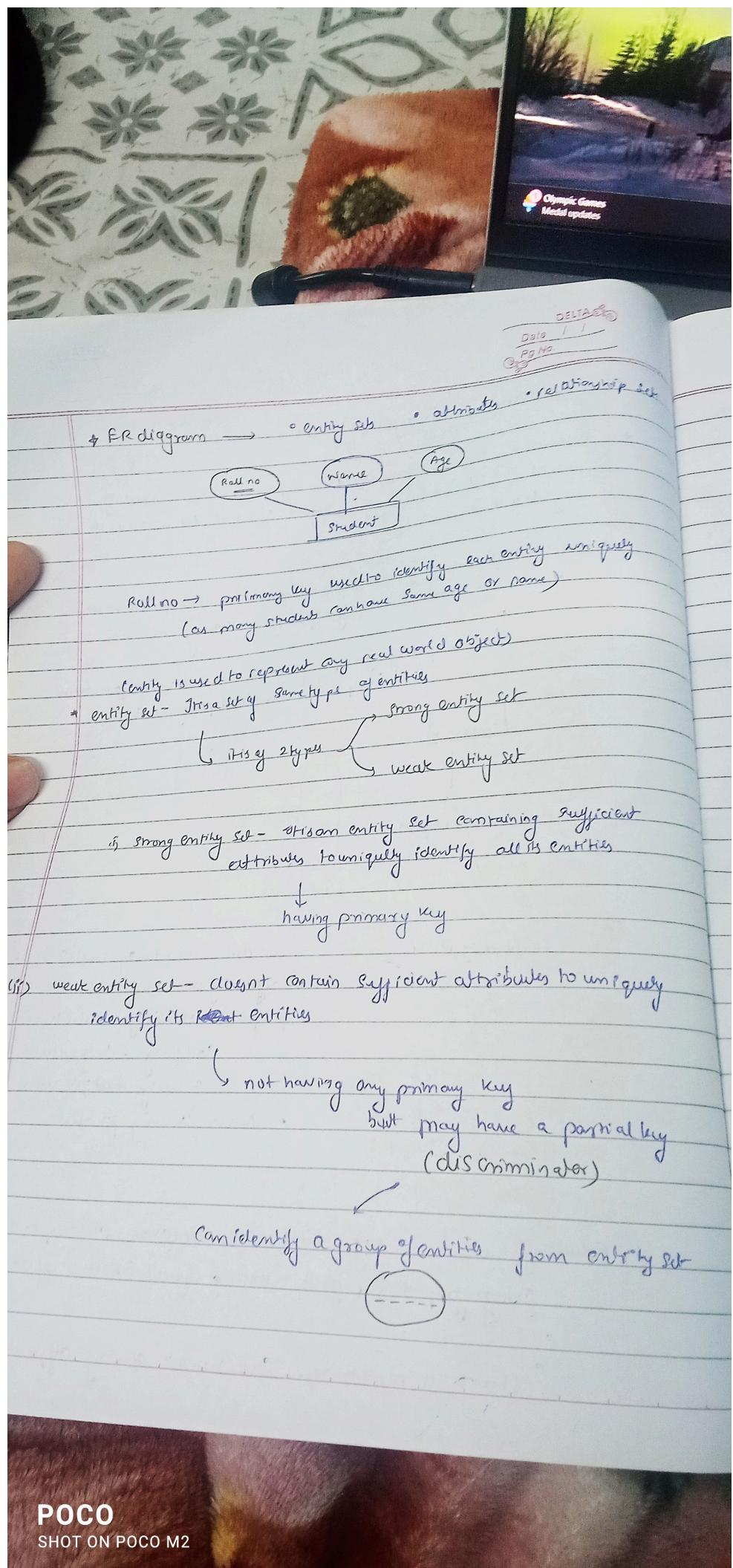
↳ managed by group of programs → manipulating databases
(helps users and third party users)

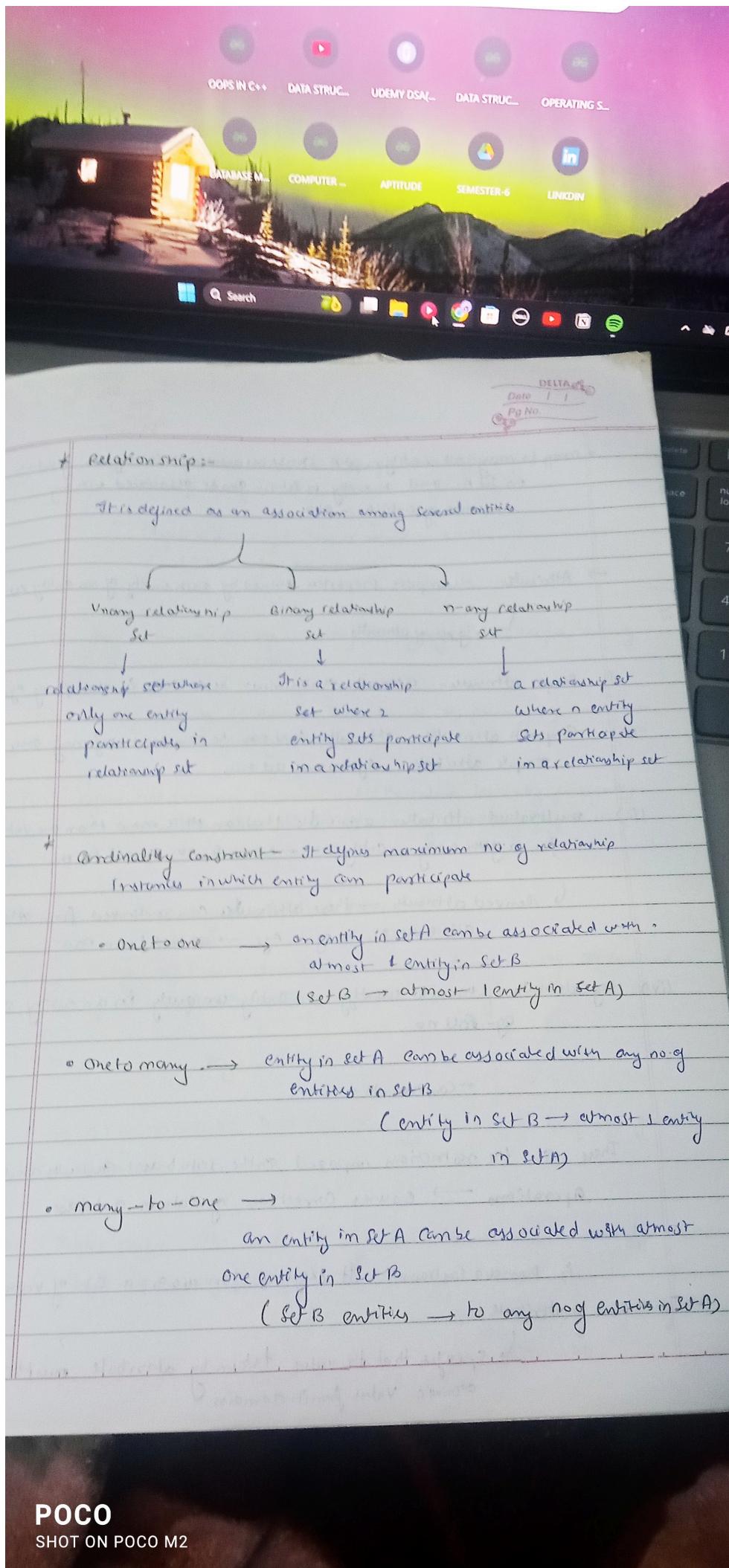


- features of DBMS →
- less data redundancy and inconsistency
 - difficulty in access data reduced
 - provides atomicity, data isolation, integrity, helps concurrent access by multiple users
 - Avoids security problems

+ ER diagram - (entity - relationship diagram), a conceptual model giving graphical representation of logical structure of database

↳ Shows all relationships and constraints among different components.





many to many → entity in set A can be associated with any no. of entities in set B and an entity in set B can be associated with any number of entities in set A

→ Attribute - descriptive properties owned by each entity of an entity set

↳ Types of attributes -

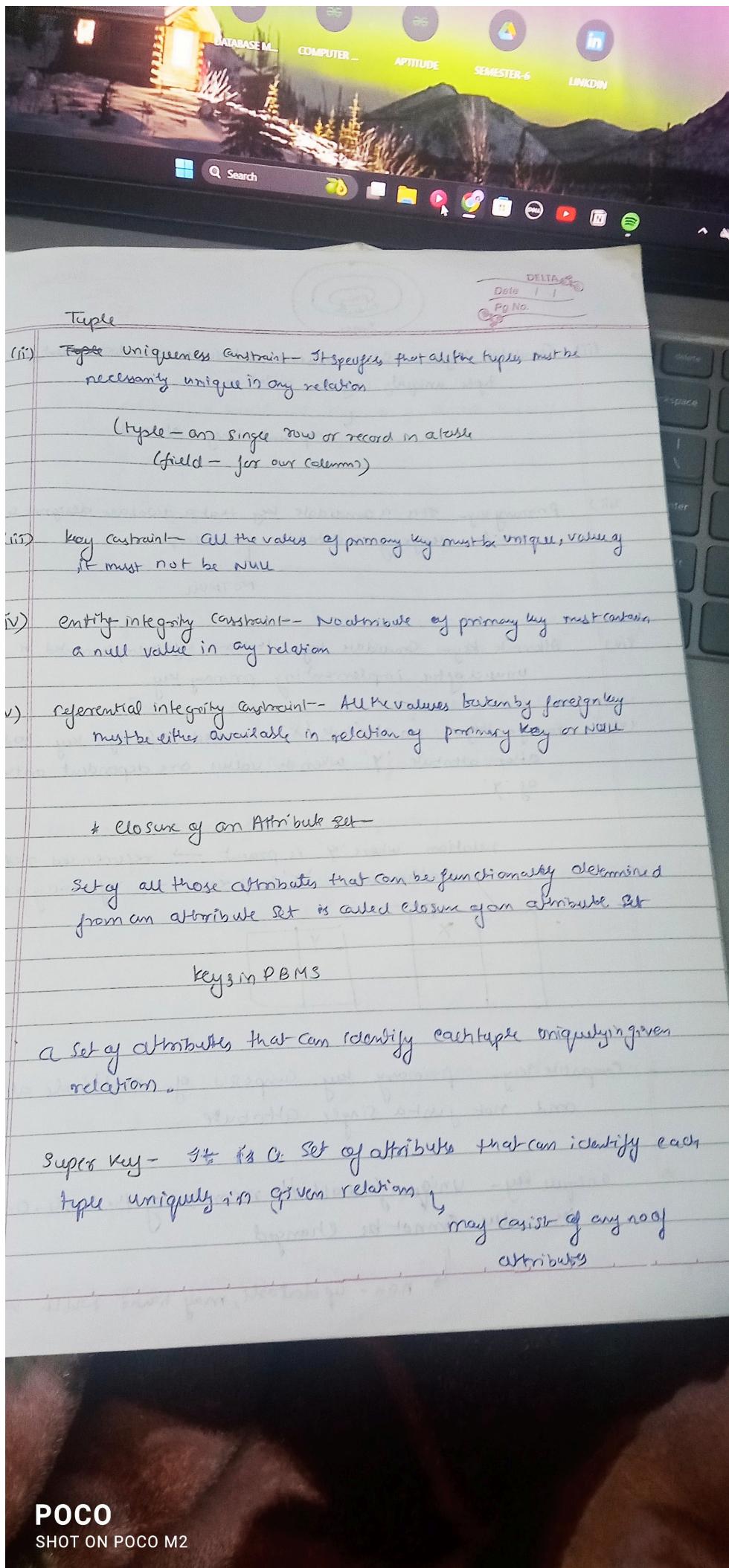
- (i) Simple attribute - attributes that cannot be divided further like age
- (ii) Composite attribute - attributes that can be composed of many other simple attributes e.g. name, address
- (iii) Multivalued attribute - attributes that can take more than one value for a given entity in entity set → mobile no, email id etc.
- ↳ derived attribute - these attributes can be derived from other attributes (e.g. Age → derived from DOB)
- ↳ Key attribute - used to identify an entity uniquely in an entity set
e.g. Roll no.

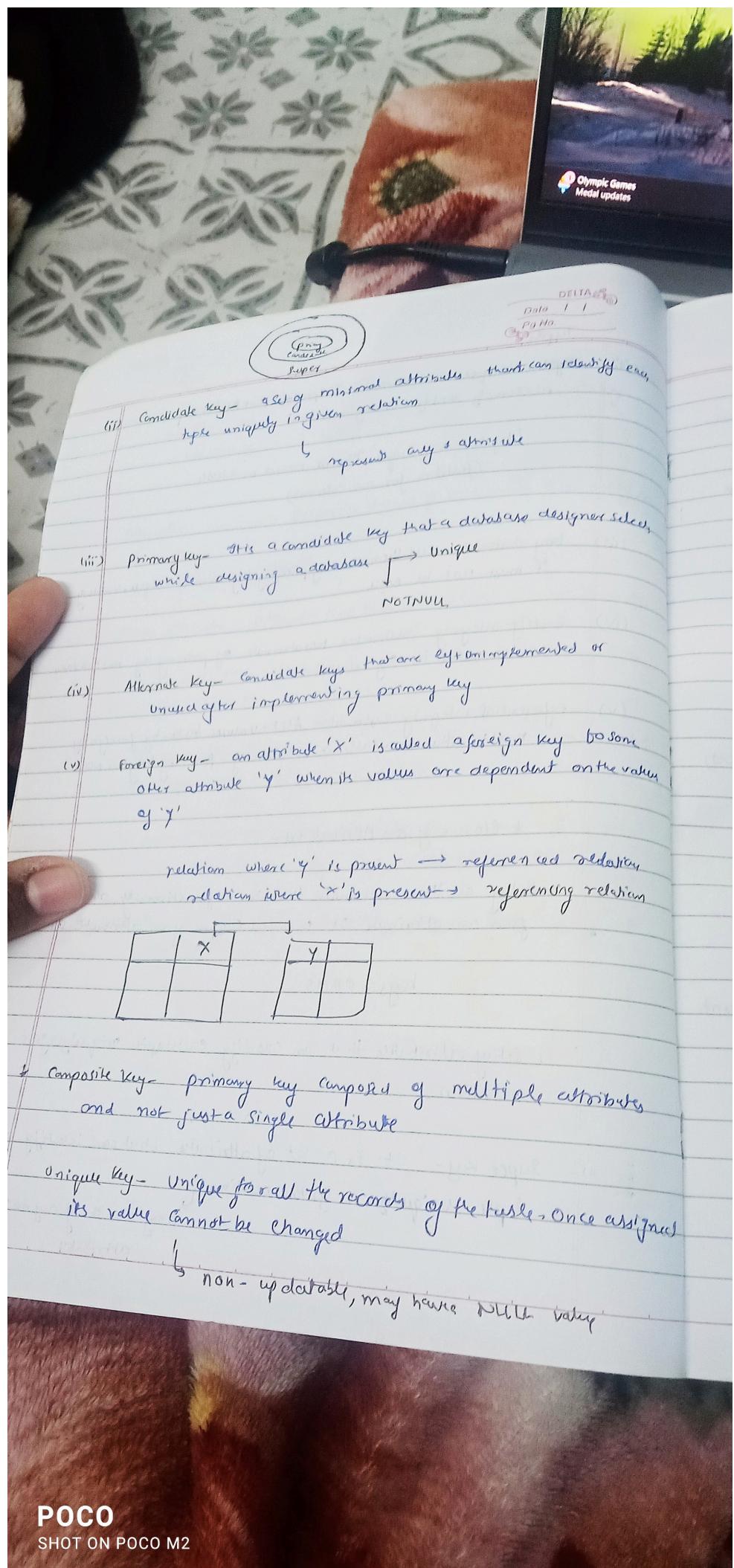
⇒ Constraints -

There are the restrictions imposed on the database contents and operations → ensures correctness of data in database

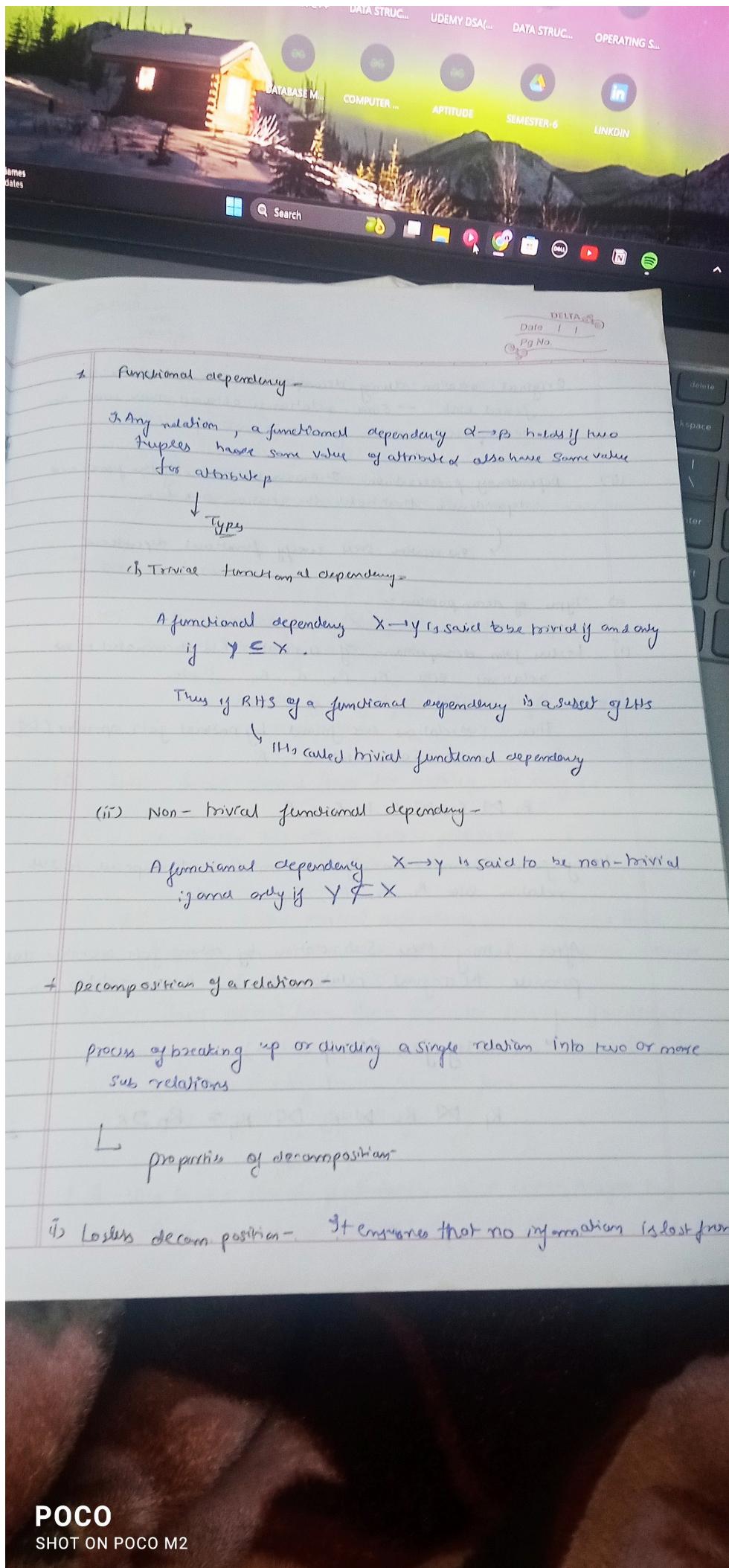
i) Domain constraint - It defines domain are a set of values for an attribute

↳ specifies that the value taken by attribute must be from its domain





POCO
SHOT ON POCO M2



DETAILED

Date 1/1

Pg No.

original relation during decomposition, when sub relations are joined back → some relation is obtained which was not decomposed.

(ii) dependency-preservation - It ensures that none of the functional dependencies that hold the relation are lost.
↳ sub relations still satisfy functional dependencies

⇒ Types of decomposition -

(i) Lossless join decomposition - If a relation R is decomposed to sub relations like $R_1, R_2, R_3, R_4, \dots$ etc.

These subrelations are joined by natural join operator (\bowtie)
leading to the same relation R

$$R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4 = R$$

(iii) Lossy join decomposition - If a relation R is decomposed to sub relations like $R_1, R_2, R_3, R_4, \dots$ etc.

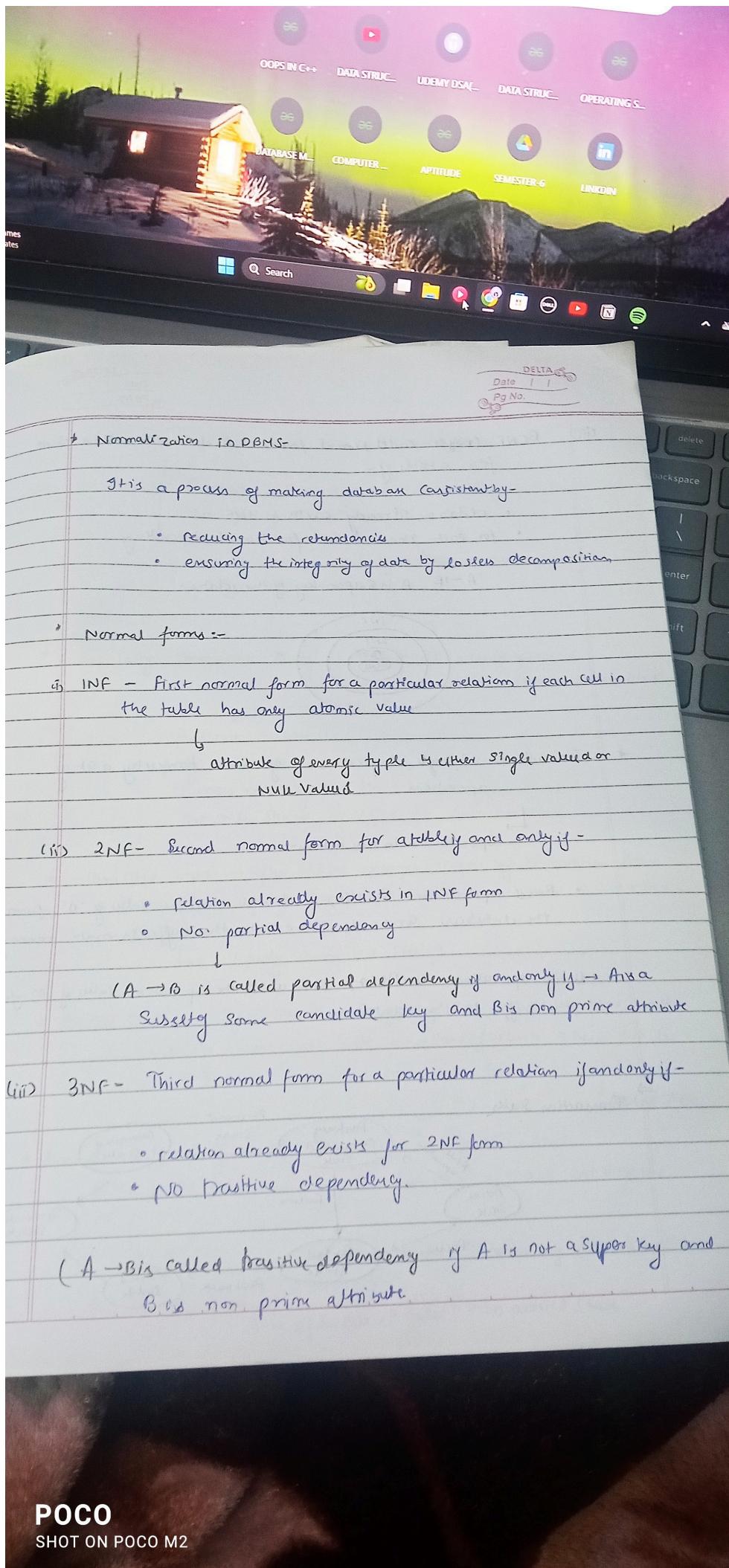
After joining these subrelations by natural join operation doesn't provide the original relation R but another relation R_n

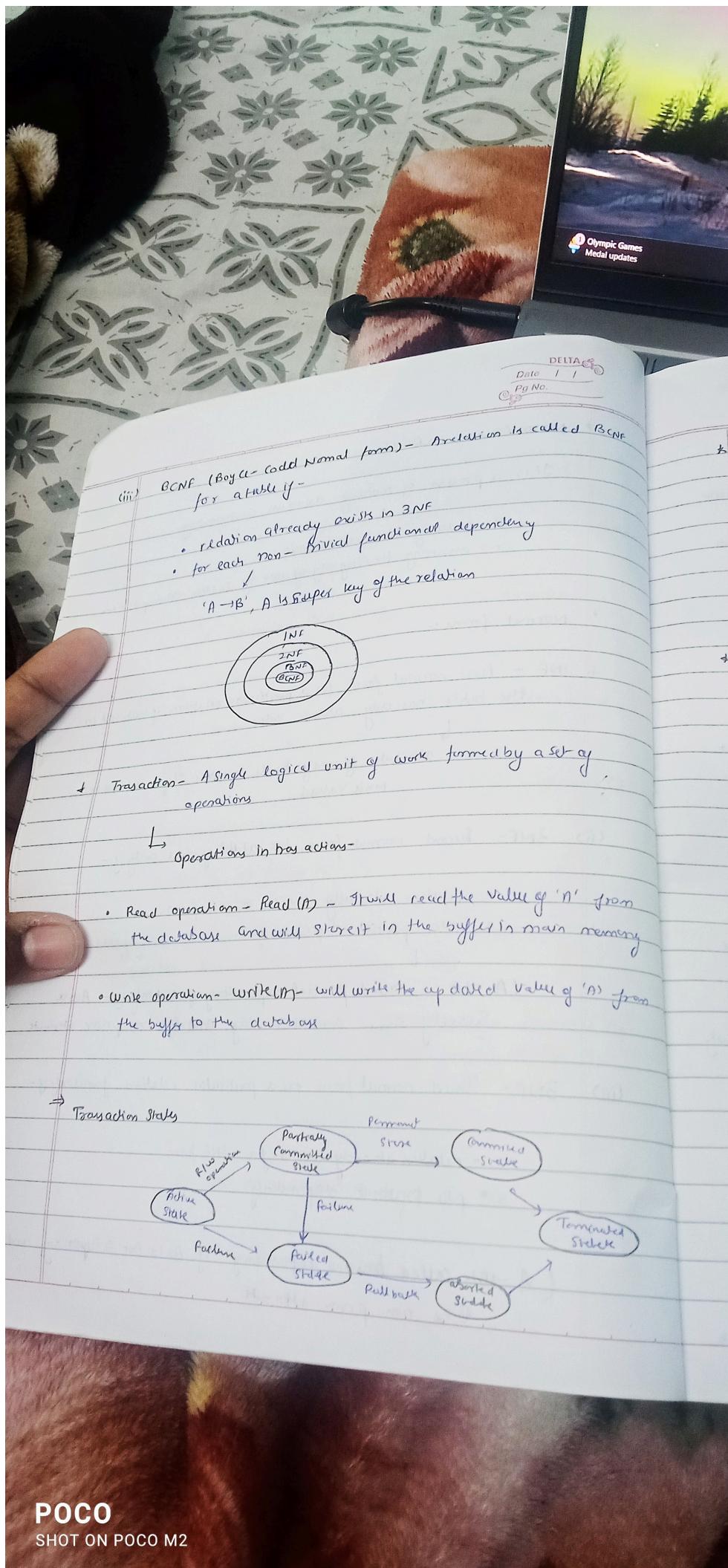
↳ (Lossy join decomposition)

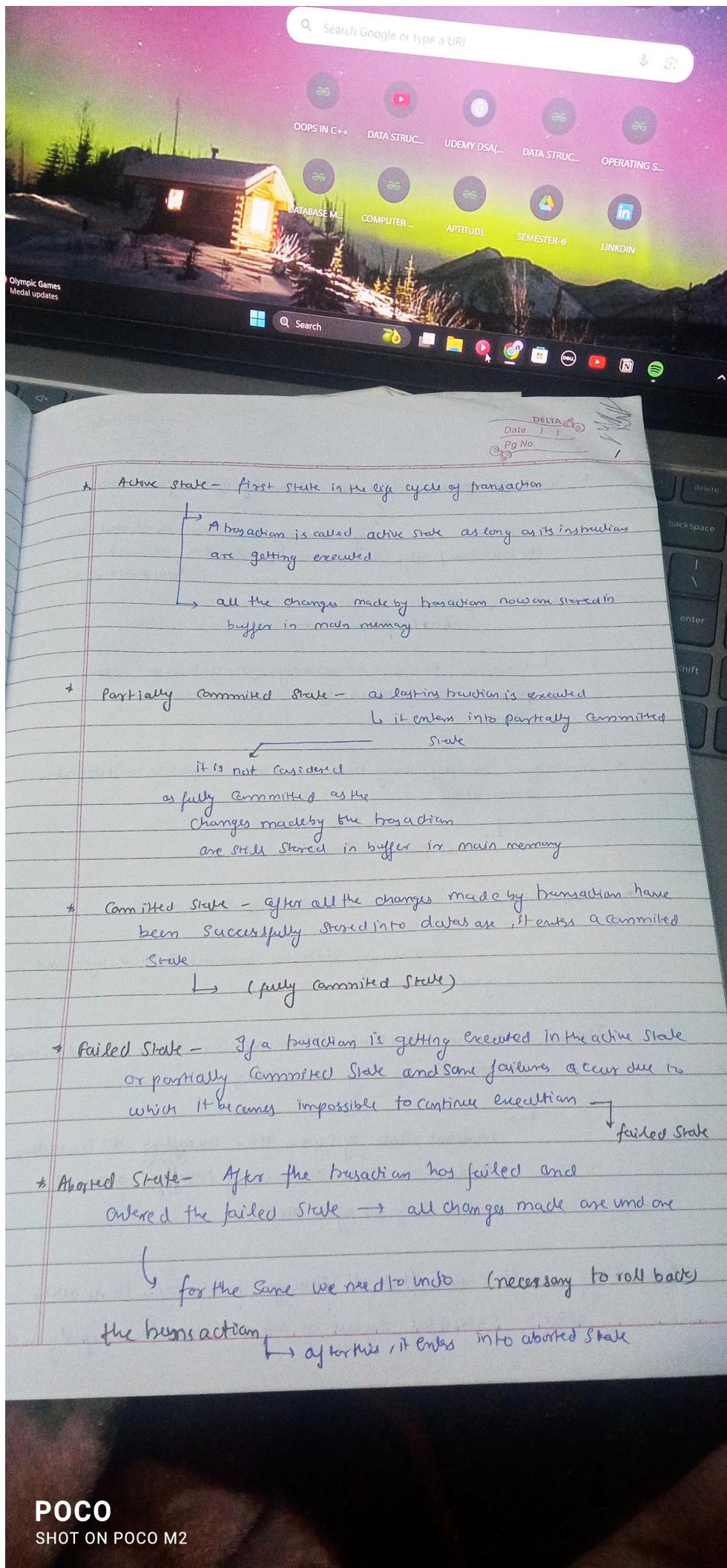
$$R_1 \bowtie R_2 \bowtie R_3 \bowtie R_4 = R_n \supset R$$

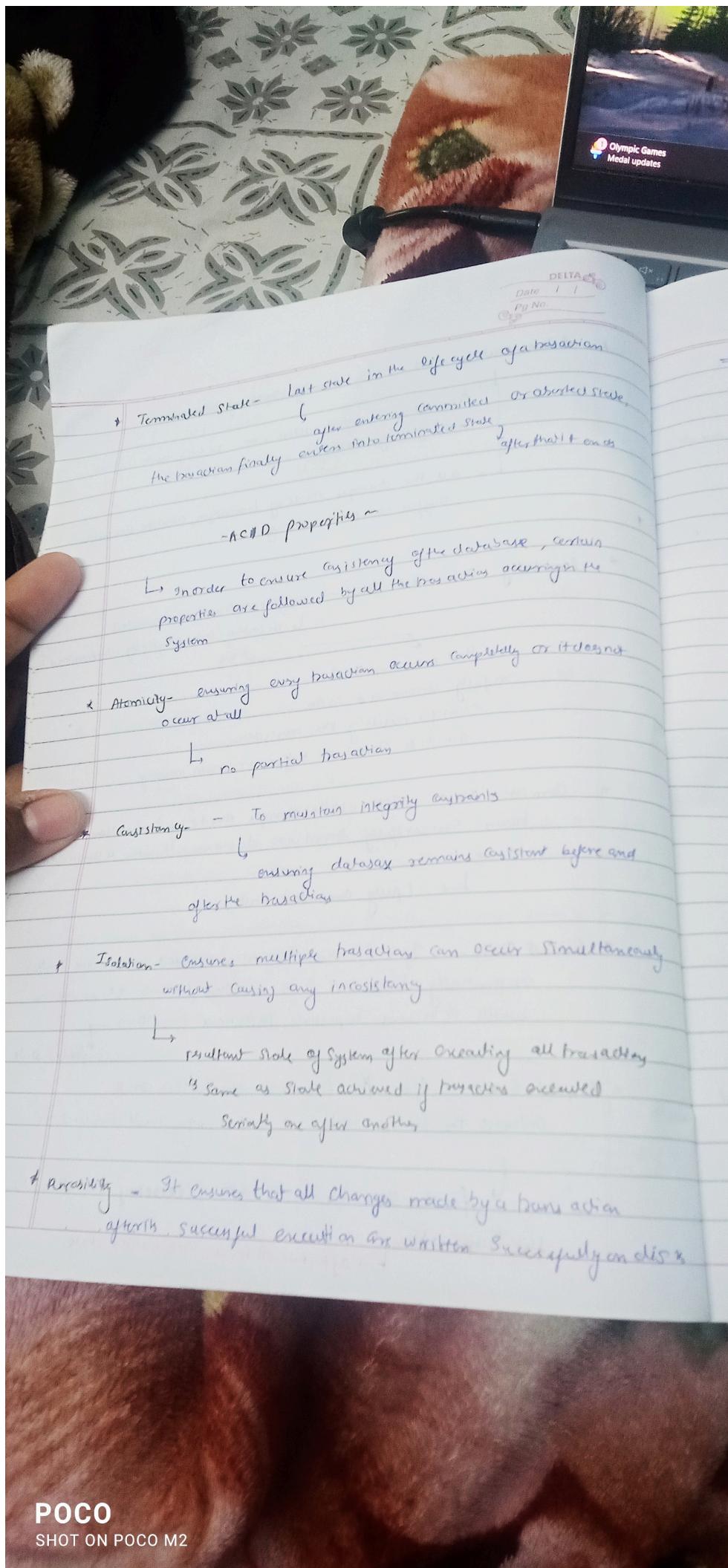
POCO

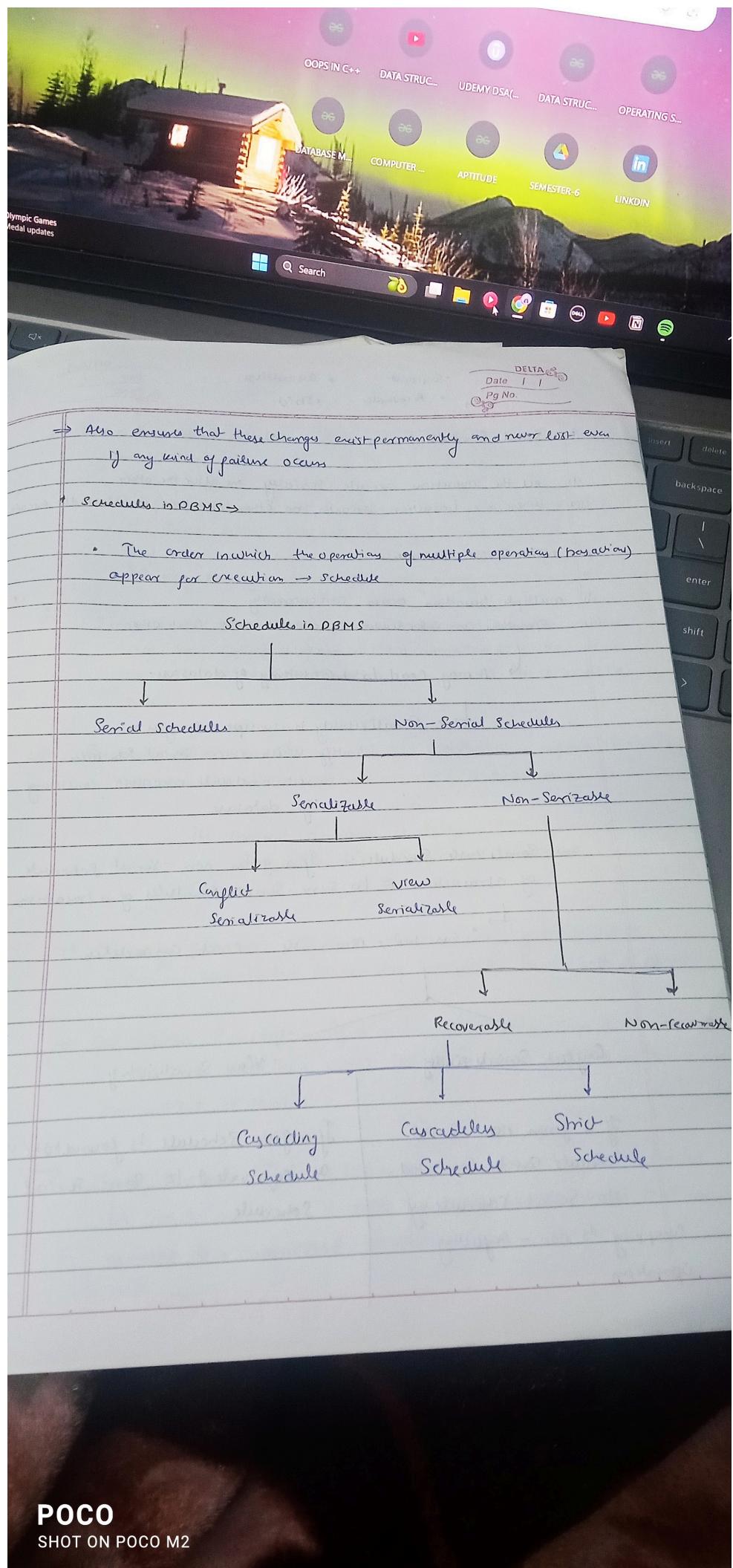
SHOT ON POCO M2

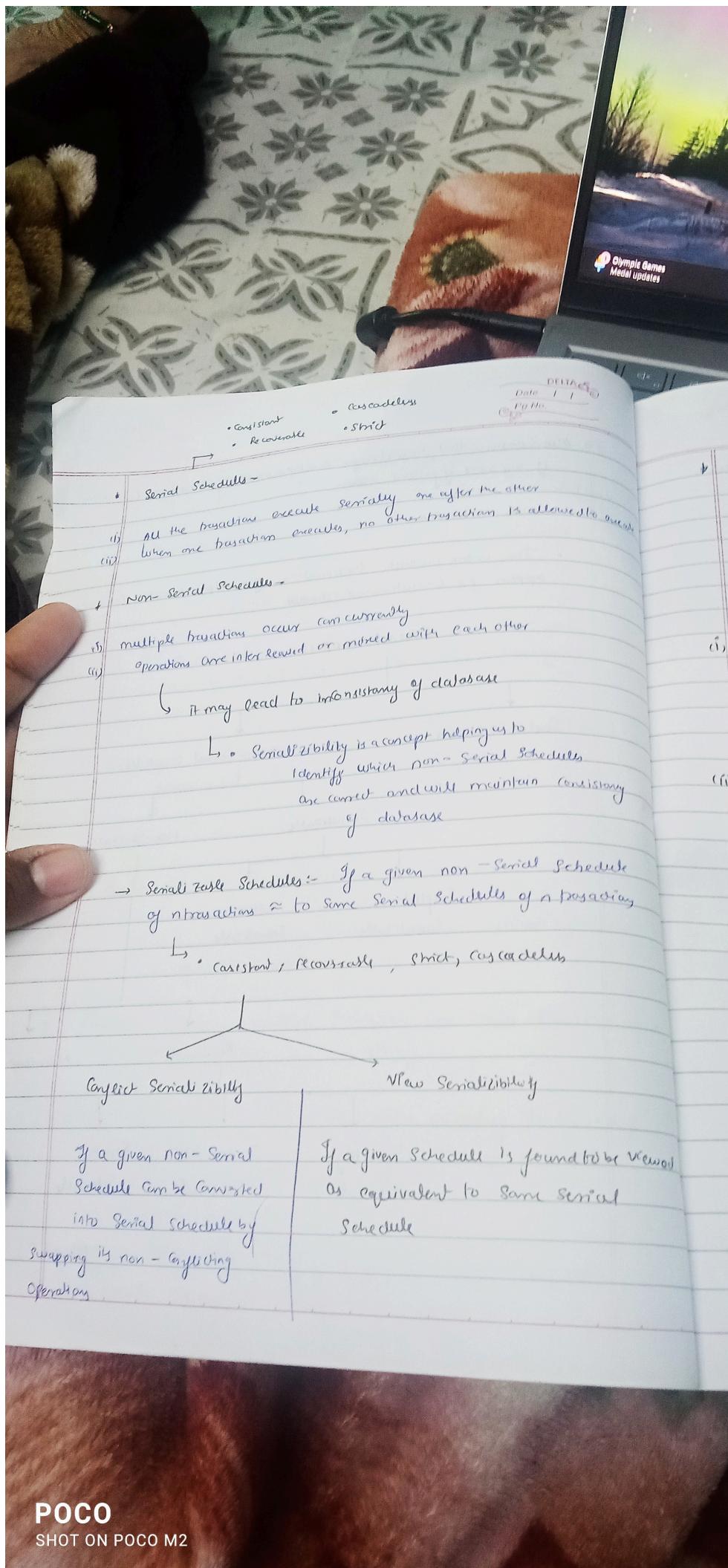


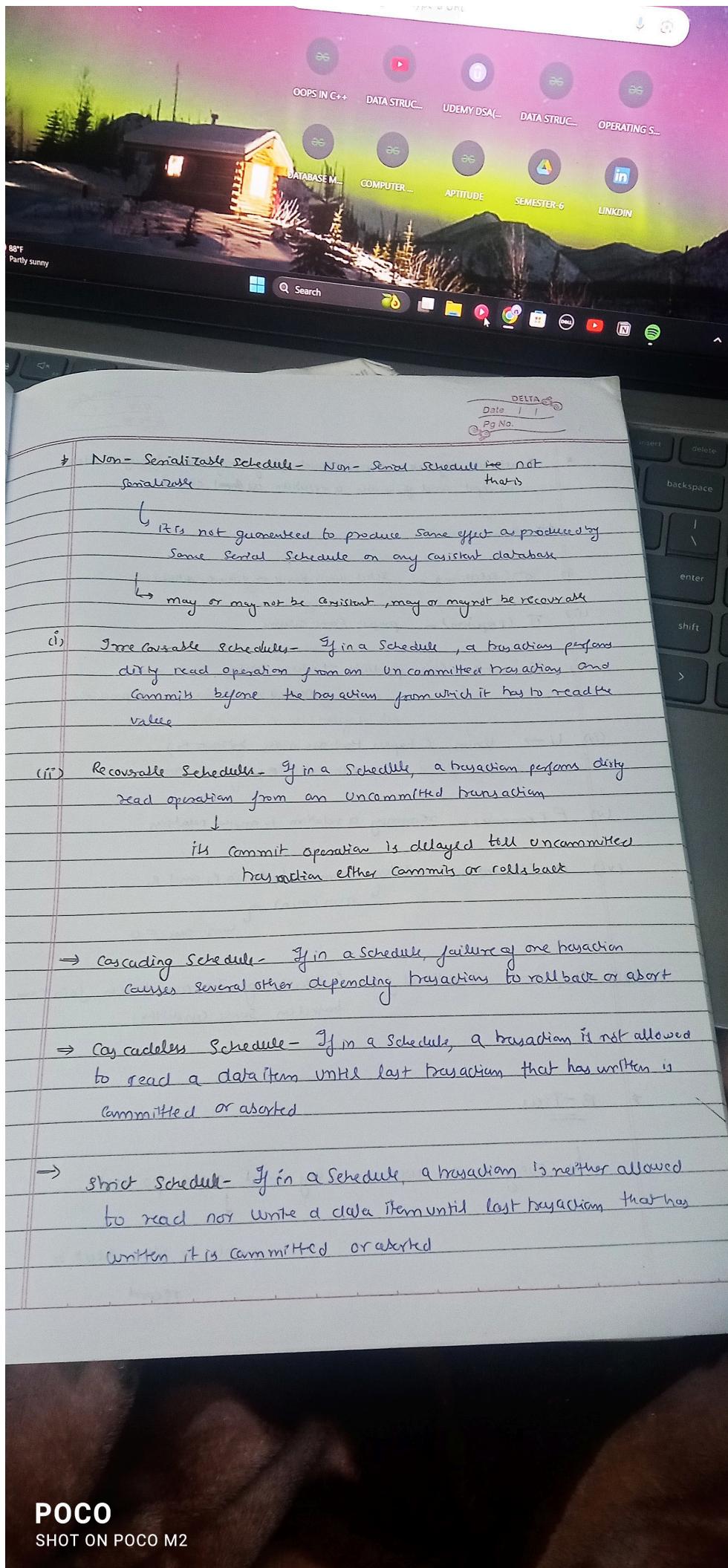












DELTA
Date 1/1
Pg No.
©

- * Relational algebra - A procedural query language taking a relation as input and generates a relation as output
- Some operators -
- (i) σ (selection) - select rows based on given condition
- (ii) π (projection) - project some columns
- (iii) $X \rightarrow$ cross products → returns (mn) rows where m and n → rows for R₁ and R₂
- (iv) U → Union (tuples that are in R₁ but not R₂)
worst case $R_1 \cup R_2 \leq M+N$
- (v) P (rename) - renaming a relation to another relation
- (vi) \cap (intersection) - selecting common tuples in R₁ and R₂
↳ min(m,n)
- (vii) \bowtie_c → Conditional join (Selection of 2 or more tables, join them based on same condition)

B-Trees

↳ every level we have a key and data pointer

↓
points to either block or
record

