✅ Summary of the Lecture Notes (DBMS Implementation – Introduction)

🎯 1. Purpose of the Course

This course provides insight into the internal workings of Database Management Systems (DBMS), focusing on:

- How databases are structured and implemented.
- Core system modules like storage, indexing, query processing, concurrency control, and recovery.

📚 2. Complementary Topics

Although the core focus is DBMS internals, the course connects with:

Topic	Description
Database Design	High-level schemas, ER diagrams, SQL data definitions.
Database Programming	Writing SQL queries and update commands.
DB Tuning	Understanding internal mechanisms to optimize performance.

📌 3. Why Study DBMS Internals?

Understanding DBMS internals is useful because:

- Databases are the backbone of modern information systems.
- Knowledge helps in designing robust applications.
- Enables fine-tuning DBMS configurations for better performance and reliability.

📖 4. Reference Books

Some recommended literature includes:

- *Database System Implementation* – Garcia-Molina, Ullman, Widom
- *Datenbanken: Implementierungstechniken* – Saake, Heuer, Sattler
- *Database Tuning* – Shasha & Bonnet

✅ 5. Codd's (Extended) Rules – What a DBMS Must Provide

Rule	Purpose
Integration	Uniform, non-redundant data structure
Operations	Store, search, modify data
Catalogue	Metadata accessible to users/programs
Views	Different views for different users
Integrity	Enforcing data correctness

Rule	Purpose
Security	Prevent unauthorized access
Transactions	Group operations into atomic units
Synchronization	Support concurrent transactions
Availability	Data recovery after failures

❌ 6. The Wrong Way to Implement a DBMS (Megatron 3000 Example)

A fictional system called *Megatron 3000* is used to illustrate bad practices.

Characteristics of Megatron 3000:

- Stores relations in plain text files.
- Uses no indexing, no buffer manager.
- Processes joins using nested loops over files.
- No concurrency control, no recovery, no security.
- No API or interaction with external applications.

Problems with this approach:

Issue	Explanation
Inefficient Storage	Text-based tuple storage causes rewrite costs on updates.
No Indexes	Full scan for every query → slow.
Expensive Joins	Brute-force nested loop joins.
No Buffering	Re-reads files every time → bad I/O efficiency.
No Transactions	Cannot guarantee atomicity or isolation.
No Recovery	Power failure → data may be lost.
No Security/API	File-based access only; unsafe for real-world use.

✅ 7. Key Takeaway

This first lecture emphasizes that:

- Building a reliable DBMS is not just about storing data.
- It requires careful handling of efficiency, concurrency, security, recovery, and abstraction.
- The rest of the course will cover **how to implement these core components correctly**.

---

two new exercise questions in the *same style and structure* as the one you provided for "Implementation of DBMS – Exercise Sheet 1".

✅ Exercise Sheet – New Version A

1) A random variable Y can take integer values from 10 to 500 uniformly.
What are the **minimum**, **maximum**, and **mean** values of Y?

---

2)
a) A USB drive can store **16 GB** of data. You have **250 GB** of data to back up.
How many USB drives are required at minimum?

b) A box holds **30 chocolates**. You have **365 chocolates** to pack.
What is the minimum number of boxes required?

c) A warehouse stores **stackable containers**, each of size:

- Length = 50 cm, Width = 50 cm, Height = 40 cm
  Each container holds **20 metal cylinders** with diameter 10 cm and height 20 cm.
  How many containers are needed to store **950 cylinders**?

---

3) You have an encyclopedia with **450 pages**. Information about a topic may appear on different pages.

a) If the topic appears on exactly **one random page**, how many pages do you need to inspect?
Calculate the **minimum, maximum, and expected (mean)** number of pages.

b) If the topic appears on **an unknown number of pages**, what are the **minimum, maximum, and expected** number of pages to inspect to find *all* information, assuming you start from page 1 and read sequentially?

c) Now assume the encyclopedia has an **index** on the last page listing page numbers for each topic.
The index says the topic is on pages **12, 199, and 412**.
How many pages must you inspect to get all the information?

---

4) Given the following relations:

R:

A	B
a	b
b	c
c	d

S:

B	C

b	x
c	y
d	z

Compute the following:

a) πA(R)
b) σB='c'(R)
c) πA(σB='b'(R))
d) R ⋈ S (natural join)

✅ Exercise Sheet – New Version B (More Challenging)

1) A random variable Z takes values from **0 to 999** uniformly.
Find the **mean**, **variance**, and **standard deviation** of Z.

---

2)
a) A hard disk has **8 TB** of capacity. A server generates **950 GB/day** of data.
After how many full days will the disk be completely full?

b) A drone can carry a maximum weight of **2.5 kg per flight**.
You must transport **73.2 kg** of medical supplies.
How many flights are required at minimum?

c) You are storing **spherical gas tanks** (radius = 1m) in a **rectangular room (Length=10m, Width=5m, Height=3m)**.
You must ensure tanks do not overlap and are fully contained inside the room.
What is the maximum number of tanks you can store?

---

3) You are searching for error logs in a **1000-page system report**.

a) You are told the error appears on exactly **one random page**. What are the **min, max, and expected pages to search**?

b) You know errors may appear on **multiple unknown pages**, and you must find all.
What is the **worst-case and expected number of pages** to search without an index?

c) If the table of contents lists exact error pages: **Page 241, 455, 700, 912**,
How many pages do you now have to inspect?

---

4) Given relations:

R:

A	B
1	x

2	y
3	x

S:

B	C
x	red
y	blue
z	green

Compute:

a) πB(R)
b) σA>1(R)
c) πA(σB='x'(R))
d) R × S
e) R ⋈ S on attribute B
