1. Explain DIKW Pyramid:

Ans: DIKW Model (Data–Information–Knowledge–Wisdom)

The DIKW Pyramid is a hierarchical model that explains how raw facts (data) are gradually transformed into useful insights and ultimately into wise decision-making. It has four stages:

1. Data (Base Layer)

Raw, unprocessed facts without context or meaning.

Can be numbers, text, images, or sounds.

On its own, data has no value unless processed.

Example: Temperature readings → 30°C, 32°C, 31°C, 29°C.

Healthcare Example: Patient's raw health readings like heart rate = 98, BP = 140/90.

2. Information

Data that is processed, organized, or structured.

Adds context so it becomes meaningful.

Answers questions like "who, what, when, where."

Example: Mapping temperature readings with days →

Monday: 30°C, Tuesday: 32°C, Wednesday: 31°C, Thursday: 29°C.

Healthcare Example: Creating a medical report showing when and how often BP/heart rate was recorded.

3. Knowledge

Results from analyzing and interpreting information.

Recognizes patterns, trends, or relationships.

Helps in predicting outcomes or making decisions.

Example: After analysis, we conclude temperatures are consistently high → indication of a heatwave.

Healthcare Example: Identifying that a patient's heart rate is consistently above normal, suggesting possible illness.

4. Wisdom (Top Layer)

The highest stage of the pyramid.

Applying knowledge + experience + judgment for decision-making.

Answers "why" and "what should be done."

Example: Based on knowledge of heatwave → decide to avoid outdoor work in the afternoon, drink more water, or schedule activities early.

Healthcare Example: Doctor decides that the patient must undergo further tests or treatment.

2. Explain Acid Properties:

1. Atomicity ("All or Nothing")

A transaction is one complete process.

It must finish fully or not happen at all.

If any step fails, the whole process is cancelled.

Example: You try to withdraw ₹5000 but ATM has only ₹3000 → transaction is cancelled, your balance remains the same.

2. Consistency

The database must always be in a valid and correct state.

After a transaction, rules of the database should not be broken.

Example: If you have ₹4000 in your account and try to withdraw ₹5000 → system stops it because balance cannot go negative.

3. Isolation

Even if many transactions run at the same time, they do not disturb each other.

Each one works as if it's running alone.

Example: You transfer money between two accounts while your friend is paying bills. Both happen together but don't affect each other's results.

4. Durability

Once a transaction is completed, the result is permanently saved.

Even if there is a crash or power cut, data will remain safe.

Example: You transfer ₹1000 online. Even if the server crashes after, the money still shows in the receiver's account.

A = All or nothing

C = Always valid

I = Independent

D = Data safe forever

3. What is Big data:

What is Big Data?

Big Data refers to extremely large and complex datasets that are difficult to store, process, and analyze using traditional database tools.

It is generated at high speed from various sources like social media, IoT devices, sensors, business transactions, and online platforms.

The goal of Big Data is to extract meaningful patterns, insights, and trends to support better decision-making.

Challenges in Handling Big Data

Handling Big Data is difficult because of its huge size, speed, and variety. The main challenges can be remembered like a library story:

1. Data Storage (Where to keep it?)

Big Data is in terabytes or petabytes, and traditional databases cannot store it easily.

Like trying to store mountains of books in a small library.

2. Data Processing (How to handle it?

Processing such a huge volume needs powerful systems and distributed computing.

Like reading all those books quickly and summarizing them.

3. Data Variety (Different formats)

Data comes in many forms → structured (tables), semi-structured (XML, JSON), and unstructured (images, videos, social media).

Like books written in different languages and formats.

4. Data Quality & Consistency

Big Data often has errors, duplicates, missing values.

Like books with missing pages or wrong printing.

5. Data Security & Privacy

Protecting sensitive data (health, finance, personal) is a big challenge.

Like locking the library so no one misuses the books.

6. Data Integration (Mixing from many places)

Big Data comes from different sources (social media, sensors, IoT, transactions).

It is hard to combine all into one system.

Easy Example: Like bringing books from many libraries and arranging them in one place.

7. Real-Time Processing (Fast results needed)

Some data must be processed instantly (fraud detection, stock market).

Delay can cause loss or wrong results.

Easy Example: Like trying to read a book while it is still being written.

8. Cost Management

Building infrastructure, servers, and storage for Big Data is very expensive.

Like building and maintaining a giant library costs a lot.

9. Analysis & Visualization

The biggest challenge is to turn raw data into useful insights and simple visuals.

Like creating a summary chart of thousands of books for easy understanding.

4. Properties of big data 5 Vs:

Big Data is defined by five important properties, called the 5 V’s:

1. Volume (Amount of Data)

Refers to the huge size of data produced every second.

Data is generated in terabytes, petabytes, or even zettabytes.

Traditional databases cannot handle such massive storage.

Example: Facebook generates more than 4 petabytes of data daily from posts, photos, and videos.

Side Point: Without large volume, data cannot be called “Big Data.”

2. Velocity (Speed of Data Generation)

Refers to the speed at which data is created, collected, and processed.

Many applications need real-time or near real-time analysis.

Delay in processing reduces the usefulness of data.

Example: Google processes 3.5 billion searches every day, and stock market systems update prices within seconds.

Side Point: Velocity is important in fields like fraud detection, traffic monitoring, and online payments.

3. Variety (Types of Data)

Refers to the different formats in which data exists.

Big Data is not only in tables but also in texts, images, videos, audio, logs, sensor data, etc.

Types of Data under Variety:

Structured Data – Organized data stored in rows and columns (databases, spreadsheets).

Example: Employee records with Name, ID, Salary.

Semi-Structured Data – Partially organized, not in fixed tables.

Example: XML, JSON files, log files.

Unstructured Data – No fixed structure, difficult to store in traditional databases.

Example: Emails, social media posts, videos, images.

Example: Amazon collects data in many forms –

Transaction history (structured)

Customer reviews (unstructured)

Clickstream data in JSON (semi-structured)

Side Point: Variety makes analysis difficult but powerful, because insights come from combining all types.

4. Value (Usefulness of Data)

Not all collected data is useful.

The real importance of Big Data is when it is analyzed to produce value for organizations.

Value means turning raw data into business insights, profits, or better decision-making.

Example: Netflix studies viewing patterns to:

Recommend personalized movies/shows.

Decide which new series to produce.

Side Point: Without value, Big Data is just wasteful storage.

5. Veracity (Accuracy of Data)

Refers to the quality, correctness, and trustworthiness of data.

Big Data may contain errors, duplicates, incomplete records, or fake information.

Low veracity leads to wrong insights and bad decisions.

Example: In healthcare, if patient records are incomplete or inaccurate, it can lead to wrong diagnosis or treatment.

Side Point: High veracity ensures reliable and safe use of Big Data.

5. CAP Theorem (Brewer’s Theorem)

The CAP theorem says that in a distributed system, you cannot achieve all three properties at the same time:

1. Consistency (C)

All nodes show the same updated data at the same time.

After an update, every user sees the latest data.

Example: If your bank balance is ₹500 and you spend ₹200, all servers must immediately show ₹300.

2. Availability (A)

The system must always respond to requests, even if some servers fail.

Every request gets a reply (may not always be the latest data).

Example: On Amazon, even if one server is down, the site still shows product details.

3. Partition Tolerance (P)

The system keeps working even if there is a network failure between servers.

Some servers can be disconnected, but the system still runs.

Example: In YouTube, if one server fails, another server shows previously stored video data.

Trade-off (Important Point):

A distributed system can only choose two out of three:

CA (Consistency + Availability): Works until a partition happens.

AP (Availability + Partition Tolerance): Always on, but may show outdated data.

CP (Consistency + Partition Tolerance): Keeps data correct, but may stop responding during partition.

6. BASE Model (Used in NoSQL Databases)

The BASE model is designed for distributed / big data systems, where strict ACID is not practical.

1. Basically Available

The system is always available to respond, even if some data is missing or outdated.

Example: In Amazon, even if a data center fails, you can still browse products and place orders.

2. Soft State

Data may be temporarily inconsistent across servers.

State keeps changing until updates are fully synchronized.

Example: If you update your profile picture, some friends may still see the old one for a while.

3. Eventually Consistent

The system guarantees that after some time, all servers will have the latest data.

Example: A WhatsApp message may take a few seconds to appear on all devices, but finally, everyone sees it.

Difference (ACID vs BASE)

ACID = Reliable, strict consistency (banks, finance).

BASE = Scalable, high availability, but accepts temporary inconsistency (social media, e-commerce).

7. Acid vs Base:

ACID	**BASE**
Strict rules	Relaxed rules
Data always correct immediately	Data correct after some time
May stop working to keep data safe	Always works, even if data not updated
All steps of a transaction must finish or none	Can save partly, then fix later
Best for banks, finance	Best for social media, e-commerce

8. Sql vs No Sql:

Feature	**SQL**	**NoSQL**
Data Model	Relational (tables, rows, columns)	Non-relational (document, key-value, graph)
Schema	Fixed, predefined	Flexible, schema-less
Scalability	Vertical (add power to one server)	Horizontal (add more servers)
Transactions	Follows **ACID** (strict)	Follows **BASE** (eventual consistency)
Query Language	Standard SQL	Custom (depends on DB)
Use Cases	Structured data, banking, finance	Unstructured data, social media, big data
Joins	Supports joins & relationships	Avoids joins (denormalized data)
Performance	Best for structured data	Best for large-scale, real-time data

9. Data Consistency:

Meaning: Data consistency means data should always be correct, reliable, and the same across systems. If one system updates data, others must also show the same update.

There are 8 types of consistency:

1. Structural Consistency

The structure of data (tables, fields, data types) should be the same everywhere.

Example: If Phone_Number is stored as text in one system, it should not be stored as number in another.

2. Value Consistency

The values stored must be accurate and follow rules.

Example: In sales, Total_Price = Quantity × Item_Price. If not, it breaks value consistency.

3. Temporal Consistency

Data must be correct with respect to time.

Example: Stock price shown at 1:00 PM in one system should not show 12:55 PM in another.

4. Cross-System Consistency

Data across different systems/databases should match.

Example: If inventory in System A says 10 items, System B should not show 7.

5. Logical Consistency

Data must follow rules and make sense.

Example: An employee's Hire_Date cannot be later than their End_Date.

6. Hierarchical Consistency

In parent–child (tree) data, relationships must be correct.

Example: If A is manager of B, B cannot be manager of A.

7. Referential Consistency

Links between tables (foreign keys) must be valid.

Example: If Student_ID = 105 is in Grades table, it must also exist in Students table.