1️⃣ FIRST: WHY Lambda Architecture exists (the STORY)

Imagine a system like:

- Uber rides
- Bank transactions
- University attendance
- IoT sensors
- Website clicks

Problem:

Data is coming **ALL THE TIME**.

You want:

- **Real-time answers** → "What's happening RIGHT NOW?"
- **Accurate answers** → "What is the CORRECT total after cleaning errors?"
- **Scalability** → data is huge
- **Fault tolerance** → machines can fail

But here's the catch:

❌ Real-time systems are **fast but sometimes wrong/incomplete**
❌ Batch systems are **accurate but slow**

👉 **Lambda Architecture exists to get BOTH speed + accuracy**

2️⃣ WHAT IS LAMBDA ARCHITECTURE (EXAM DEFINITION)

Lambda Architecture is a big data architecture that processes data using two parallel paths:

- a **batch layer** for accurate historical computation

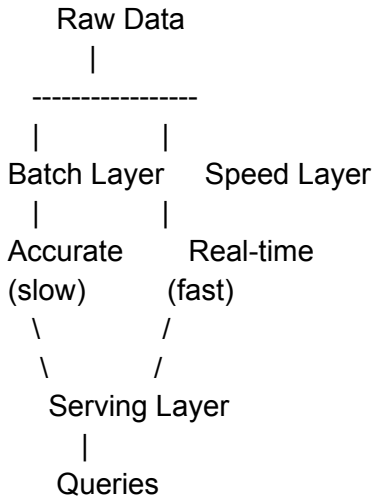
- a **speed layer** for real-time processing
  and merges results in a **serving layer**.

Keywords your teacher wants:

- batch processing
- real-time / stream processing
- fault tolerance
- scalability
- low latency + correctness

3️⃣ THE CORE IDEA (VERY IMPORTANT)

👉 **Same data goes to TWO paths at the same time**



You are NOT choosing batch OR streaming.
You are doing **BOTH in parallel**.

4️⃣ THE THREE LAYERS — IN DEPTH (THIS IS EXAM GOLD)

🟦 1. BATCH LAYER (the “truth keeper”)

What it does

- Stores **ALL historical data**
- Recomputes results **from scratch**
- Gives **accurate, correct results**

Characteristics

- Slow (minutes / hours)
- Very accurate
- Fault-tolerant (can recompute)
- Uses large-scale distributed processing

Technologies (Databricks side)

In exam, you write:

- **Databricks (Spark Batch)**
- Storage: **HDFS / Data Lake / S3 / DBFS**
- Processing: **Spark jobs on Databricks**

Example

“Every night, recompute total sales per product from ALL orders since day 1.”

👉 If there was bad data earlier, batch layer **fixes it**.

🟨 2. SPEED LAYER (the “fast but rough” layer)

What it does

- Processes **new incoming data immediately**
- Handles data **not yet processed by batch**
- Gives **low-latency results**

Characteristics

- Very fast (seconds)
- Might be incomplete
- Uses streaming

- Temporary results

Technologies (Databricks side)

In exam:

- **Databricks Structured Streaming**
- Input: **Kafka / Event Stream**
- Processing: Spark Streaming on Databricks

Example

“Show orders from the last 30 seconds.”

👉 This layer **fills the gap** while batch is running.

🟩 3. SERVING LAYER (the “answer machine”)

What it does

- Combines **batch results + speed results**
- Serves queries to users/apps
- Optimized for fast reads

Characteristics

- No heavy computation
- Read-optimized
- Always query-ready

Technologies

In exam:

- **HBase / Cassandra / MongoDB**
- Or Databricks Delta tables queried by BI tools

Example

Dashboard asks:

“Total sales till now”

Serving layer answers using:

batch_result + speed_result

5️⃣ WHERE DATABRICKS FITS (THIS IS CRITICAL)

Databricks is **NOT a single layer**.
It can be used in **MULTIPLE layers**.

Databricks roles:

Layer	Databricks Role
Batch Layer	Spark batch jobs
Speed Layer	Structured Streaming
Serving Layer	Delta Tables / SQL / BI

👉 This is WHY Databricks is perfect for Lambda Architecture.

6️⃣ EXAM-READY ARCHITECTURE #1 (CLASSIC LAMBDA)

Use this when you want SAFE MARKS

Scenario: E-commerce analytics

Flow:

1. Events (orders/clicks) generated
2. Events go to **Kafka**
3. Kafka sends data to:
    - Databricks Batch
    - Databricks Streaming

4. Batch computes historical aggregates
5. Streaming computes real-time aggregates
6. Results stored in **HBase**
7. Dashboard queries HBase

Write in exam:

- Batch: Databricks Spark batch
- Speed: Databricks Structured Streaming
- Serving: HBase

---

7 EXAM-READY ARCHITECTURE #2 (DELTA-BASED LAMBDA)

This is **very Databricks-specific** and looks advanced.

Flow:

1. Data ingested into **Delta Lake**
2. Batch jobs recompute tables nightly
3. Streaming jobs update same Delta tables
4. Delta handles versioning + ACID
5. BI tools query Delta tables

Why examiner likes this:

- Shows Databricks expertise
- Mentions Delta Lake
- Shows simplification of serving layer

---

8 EXAM-READY ARCHITECTURE #3 (IOT / SENSOR SYSTEM)

Scenario: Smart campus / sensors

Batch Layer

- Databricks batch jobs analyze historical sensor data

Speed Layer

- Streaming detects abnormal temperature / motion

Serving Layer

- Alerts + dashboards

Write example:
“Speed layer triggers alerts instantly, batch layer corrects long-term trends.”

9️⃣ EXAM-READY ARCHITECTURE #4 (BANKING / FRAUD)

Scenario: Fraud detection

- Speed layer flags suspicious transaction immediately
- Batch layer recalculates risk scores overnight
- Serving layer merges both

Key exam phrase:

“Speed layer provides immediate reaction, batch layer ensures correctness.”

🔟 HOW TO DRAW THIS IN THE EXAM (STEP-BY-STEP)

Always draw:

1. **Data source at top**
2. Split arrow into TWO paths
3. Label:
    - Batch Layer (Databricks Spark)
    - Speed Layer (Databricks Streaming)

4. Merge into Serving Layer
5. Arrow to Users / Dashboard

👉 Even a **simple clean diagram** gets marks.

1️⃣1️⃣ WHAT TO WRITE IN WORDS (20-MARK STRUCTURE)

Use this structure:

1. Definition of Lambda Architecture (2–3 lines)

2. Explain Batch Layer (role + tech)

3. Explain Speed Layer (role + tech)

4. Explain Serving Layer (role + tech)

5. Databricks integration

6. Real-world example

7. Advantages

8. Limitation (complexity)

1️⃣2️⃣ COMMON EXAM TRAPS (AVOID THESE)

❌ Saying Lambda = only streaming
❌ Forgetting serving layer
❌ Not mentioning Databricks role
❌ Not giving example
❌ No diagram

1️⃣3️⃣ ONE-PAGE MEMORY VERSION (LAST-MINUTE)

- Lambda = Batch + Speed + Serving
- Batch = slow + accurate + Databricks Spark
- Speed = fast + streaming + Databricks
- Serving = query layer
- Same data flows to BOTH
- Databricks can do BOTH batch + streaming