BUSINESS CONTEXT

You are working for a **digital learning platform**.

Data is coming from:

- User registrations
- Course enrollments
- User activity logs
- Payments system
- Metadata services

The data is **messy**, **inconsistent**, and **partially corrupt**.

Your job is to **clean, structure, analyze, and optimize**.

DATASET 1 — USER REGISTRATION (CORRUPTED SCHEMA)

```
raw_users = [
    ("U001","Amit","28","Hyderabad","['AI','ML','Cloud']"),
    ("U002","Neha","Thirty","Delhi","AI,Testing"),
    ("U003","Ravi",None,"Bangalore",["Data","Spark"]),
    ("U004","Pooja","29","Mumbai",None),
    ("U005","", "31","Chennai","['DevOps']")
]
```

Problems

- Age mixed types
- Skills mixed (string, array, different formats)
- Empty names
- Missing values

Exercises

1. Design an explicit schema using StructType
2. Normalize age into IntegerType
3. Normalize skills into ArrayType
4. Handle empty or missing names
5. Produce a clean `users_df`

DATASET 2 — COURSE CATALOG (NESTED STRUCT)

```
raw_courses = [
    ("C001","PySpark Mastery",("Data Engineering","Advanced"),"₹9999"),
    ("C002","AI for Testers",{"domain":"QA","level":"Beginner"},"8999"),
    ("C003","ML Foundations",("AI","Intermediate"),None),
    ("C004","Data Engineering Bootcamp","Data|Advanced","₹14999")
]
```

Problems

- Domain info as tuple, map, string
- Price as string with currency symbol
- Missing prices

Exercises

1. Create nested StructType for course metadata
2. Normalize domain and level
3. Convert price to IntegerType
4. Handle missing prices
5. Produce `courses_df`

DATASET 3 — USER COURSE ENROLLMENTS (JOIN + BROADCAST)

```
raw_enrollments = [
    ("U001","C001","2024-01-05"),
    ("U002","C002","05/01/2024"),
    ("U003","C001","2024/01/06"),
    ("U004","C003","invalid_date"),
    ("U001","C004","2024-01-10")
]
```

Exercises

1. Normalize enrollment dates
2. Identify invalid enrollments
3. Join with `users_df`
4. Join with `courses_df`
5. Decide which table should be broadcast
6. Prove your choice using `explain(True)`

DATASET 4 — USER ACTIVITY LOGS (ARRAY + MAP)

```
raw_activity = [
    ("U001","login,watch,logout","{'device':'mobile','ip':'1.1.1.1'}",120
    ("U002",["login","watch"],"device=laptop;ip=2.2.2.2",90),
    ("U003","login|logout",None,30),
    ("U004",None,"{'device':'tablet'}",60)
]
```

Problems

- Actions in multiple formats
- Metadata as JSON-like strings
- Missing actions

Exercises

1. Normalize actions into ArrayType

2. Normalize metadata into MapType
3. Handle missing actions safely
4. Explode actions and count frequency
5. Produce `activity_df`

DATASET 5 — PAYMENTS (WINDOW + AGGREGATES)

```
raw_payments = [
    ("U001","2024-01-05",9999),
    ("U001","2024-01-10",14999),
    ("U002","2024-01-06",8999),
    ("U003","2024-01-07",0),
    ("U004","2024-01-08",7999),
    ("U001","2024-01-15",1999)
]
```

Exercises

1. Convert dates properly
2. Compute total spend per user (GroupBy)
3. Compute running spend per user (Window)
4. Rank users by total spend
5. Compare GroupBy vs Window outputs

DATASET 6 — PARTITIONS & PERFORMANCE

Exercises

1. Check default partitions for all DataFrames
2. Repartition enrollments by `course_id`
3. Coalesce results before writing
4. Write outputs and inspect file counts
5. Explain why repartition caused shuffle

DATASET 7 — DAG & OPTIMIZATION

Exercises

1. For each major transformation, run `explain(True)`

2. Identify:

   - Shuffles
   - Sorts
   - Broadcast joins

3. Identify one **bad DAG**

4. Rewrite pipeline to improve it

5. Justify improvements using physical plan