REAL-TIME CASE STUDY

Learning Platform User Engagement & Performance Analytics

BUSINESS CONTEXT

You are working as a **Data Engineer / Analytics Engineer** for an **online learning platform** (similar to Coursera / Udemy / BotCampus).

The platform wants to **identify high-value learners**, **optimize course engagement**, and **improve certification conversions**.

Data is coming from **multiple independent systems**, with **inconsistent formats**.

Your task is to **clean, integrate, analyze, and optimize** this data using **PySpark**.

DATA SOURCES PROVIDED

You will receive **4 raw datasets**, each with its own problems.

DATASET 1 — USER MASTER (CORRUPTED SCHEMA)

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

Known Issues

- Age in mixed formats
- Skills as string / array / multiple delimiters
- Missing names
- Null values

DATASET 2 — COURSE CATALOG

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

Known Issues

- Price as string with currency
- Missing price

DATASET 3 — USER ENROLLMENTS

```
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"),
    ("U005","C002","2024-01-12")
]
```

Known Issues

- Multiple date formats
- Invalid dates
- Potential orphan records

DATASET 4 — USER ACTIVITY LOGS

```
raw_activity = [
    ("U001","login,watch,logout","{'device':'mobile'}",120),
    ("U002",["login","watch"],"device=laptop",90),
    ("U003","login|logout",None,30),
    ("U004",None,"{'device':'tablet'}",60),
    ("U005","login","{'device':'mobile'}",15)
]
```

Known Issues

- Actions in mixed formats
- Metadata in JSON-like strings
- Missing actions

BUSINESS QUESTIONS TO ANSWER

You must answer **all** of the following using PySpark.

PART A — DATA CLEANING & STRUCTURING

1. Design explicit schemas for all datasets
2. Normalize data types (age, price, dates)
3. Convert skills and actions into arrays
4. Handle missing and invalid records gracefully
5. Produce clean DataFrames:
    - `users_df`
    - `courses_df`
    - `enrollments_df`
    - `activity_df`

PART B — DATA INTEGRATION (JOINS)

6. Join users with enrollments
7. Join enrollments with courses
8. Decide which table(s) should be broadcast
9. Justify your decision using `explain(True)`
10. Eliminate orphan records

PART C — ANALYTICS & AGGREGATIONS

11. Total enrollments per course
12. Total revenue per course
13. Average engagement time per course
14. Total courses enrolled per user
15. Identify users with **zero activity**

PART D — WINDOW FUNCTIONS

16. Rank users by total time spent
17. Calculate running revenue per course by enrollment date
18. Identify top 2 users per course by engagement
19. Compare GroupBy vs Window results for at least one metric

---

PART E — UDF (ONLY IF REQUIRED)

20. Classify users into engagement levels:
    - High
    - Medium
    - Low

Rules:

- Use built-in functions where possible
- Use UDF only if unavoidable
- Explain why UDF was needed (or avoided)

PART F — SORTING & ORDERING

21. Sort courses by total revenue (descending)
22. Sort users by engagement within each city

23. Explain why sorting caused a shuffle

PART G — SET OPERATIONS

Create two DataFrames:

- Users who **enrolled**
- Users who **completed activity**

24. Find users who enrolled but never became active
25. Find users who are both enrolled and active
26. Explain why set operations are different from joins

PART H — DAG & PERFORMANCE ANALYSIS

27. For at least three operations, run `explain(True)`

28. Identify:

    - Shuffles
    - Broadcast joins
    - Sort operations

29. Suggest one performance improvement
