REAL-TIME CASE STUDY #2

Ride-Hailing Platform Trip & Driver Performance Analytics

BUSINESS CONTEXT

You are a **Data Engineer** working for a **ride-hailing platform** (similar to Uber / Ola / Rapido).

The business wants to:

- Identify **high-performing drivers**
- Understand **city-wise revenue**
- Detect **inactive or low-engagement drivers**
- Improve **trip completion and earnings**

Data is coming from **multiple operational systems** with **data quality issues**.

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

DATA SOURCES PROVIDED

You are given **4 raw datasets**.

DATASET 1 — DRIVER MASTER (CORRUPTED)

```
raw_drivers = [
    ("D001","Ramesh","35","Hyderabad","Car,Bike"),
    ("D002","Suresh","Forty","Bangalore","Auto"),
    ("D003","Anita",None,"Mumbai",["Car"]),
    ("D004","Kiran","29","Delhi","Car|Bike"),
    ("D005","", "42","Chennai",None)
]
```

Known Issues

- Age in mixed formats
- Vehicle types in string / array / multiple delimiters
- Missing names
- Null values

DATASET 2 — CITY MASTER (SMALL LOOKUP)

```
raw_cities = [
    ("Hyderabad","South"),
    ("Bangalore","South"),
    ("Mumbai","West"),
    ("Delhi","North"),
    ("Chennai","South")
]
```

Notes

- Small reference dataset
- Intended for broadcast join

DATASET 3 — TRIPS DATA

```
raw_trips = [
    ("T001","D001","Hyderabad","2024-01-05","Completed","450"),
    ("T002","D002","Bangalore","05/01/2024","Cancelled","0"),
    ("T003","D003","Mumbai","2024/01/06","Completed","620"),
    ("T004","D004","Delhi","invalid_date","Completed","540"),
    ("T005","D001","Hyderabad","2024-01-10","Completed","700"),
    ("T006","D005","Chennai","2024-01-12","Completed","350")
]
```

Known Issues

- Multiple date formats
- Invalid dates
- Fare as string
- Cancelled trips with zero revenue

DATASET 4 — DRIVER ACTIVITY LOGS

```
raw_activity = [
    ("D001","login,accept_trip,logout","{'device':'mobile'}",180),
    ("D002",["login","logout"],"device=laptop",60),
    ("D003","login|accept_trip",None,120),
    ("D004",None,"{'device':'tablet'}",90),
    ("D005","login","{'device':'mobile'}",30)
]
```

Known Issues

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

BUSINESS QUESTIONS TO ANSWER

You must answer **all sections**.

PART A — DATA CLEANING & STRUCTURING

1. Design **explicit schemas** for all datasets
2. Normalize:
    - Age
    - Fare
    - Dates
3. Convert vehicle types and actions into arrays
4. Handle missing and invalid records gracefully

5. Produce clean DataFrames:
    - `drivers_df`
    - `cities_df`
    - `trips_df`
    - `activity_df`

PART B — DATA INTEGRATION (JOINS)

6. Join trips with drivers
7. Join trips with cities
8. Decide which dataset should be **broadcast**
9. Prove your decision using `explain(True)`
10. Remove orphan trips (drivers not in master)

PART C — ANALYTICS & AGGREGATIONS

11. Total trips per city
12. Total revenue per city
13. Average fare per driver
14. Total completed trips per driver
15. Identify drivers with **no completed trips**

PART D — WINDOW FUNCTIONS

16. Rank drivers by total revenue (overall)
17. Rank drivers by revenue **within each city**
18. Calculate running revenue per city by date
19. Compare GroupBy vs Window for one metric

PART E — UDF (ONLY IF REQUIRED)

20. Classify drivers into performance levels:
    - High
    - Medium

- Low

Rules:

- Prefer built-in functions
- Use UDF only if unavoidable
- Justify your choice

PART F — SORTING & ORDERING

21. Sort cities by total revenue (descending)
22. Sort drivers by revenue within each city
23. Explain why sorting caused a shuffle

PART G — SET OPERATIONS

Create two DataFrames:

- Drivers who **completed trips**
- Drivers who **were active (login)**

24. Find drivers who logged in but never completed trips
25. Find drivers who completed trips and were active
26. Explain why set operations differ from joins

PART H — DAG & PERFORMANCE ANALYSIS

27. Run `explain(True)` for:
    - Join with city master
    - Window ranking
    - Sorting
28. Identify:
    - Shuffles
    - Broadcast joins
    - Sort stages
29. Suggest one performance improvement
