
🚗 Exercise — “Ride Share Analytics on Azure Databricks”

📂 Files to Upload to DBFS (`/FileStore/tables/`)

drivers.csv

```
driver_id,driver_name,city,experience_years,rating
1,Arun Kumar,Bangalore,4,4.6
2,Pooja Rao,Chennai,2,4.3
3,Imran Sheikh,Delhi,6,4.8
4,Nisha Patel,Hyderabad,1,4.0
5,Rajesh Naik,Mumbai,8,4.9
6,Sneha Menon,Bangalore,3,4.4
```

rides.csv

```
ride_id,driver_id,distance_km,price,ride_date
1001,1,10,250,2024-02-01
1002,2,7,180,2024-02-03
1003,3,15,500,2024-02-05
1004,4,5,120,2024-02-08
1005,1,12,300,2024-02-10
1006,5,20,800,2024-02-15
1007,6,9,230,2024-02-20
1008,7,11,270,2024-02-22
```

feedback.csv

```
ride_id,customer_feedback
1001,Excellent
1002,Good
1003,Excellent
1004,Average
1005,Good
1006,Excellent
1007,Average
1008,Good
```

🧩 Tasks for Participants

1. **Load all three files** into separate DataFrames.
2. **Inspect schemas** and verify correct data types.
3. Identify rides with **missing drivers** (notice one ride has no match).
4. Join rides ↔ drivers to include `city`, `experience_years`, `rating`.
5. Add a column `earnings_per_km = price / distance_km`.
6. Calculate **total revenue per city**.

7. Determine **average rating by city**.
8. Find the **top earning driver in each city** using a window function.
9. Identify **drivers with no rides** (Left Join).
10. Combine the **feedback** dataset to mark ride performance.
11. Count rides by feedback category (Excellent/Good/Average).
12. Using SQL view, find **cities with highest average ride price**.
13. Save final combined DataFrame as `rides_summary.csv` in DBFS.
14. Plot **total earnings per city** as a bar chart.

Learning Objectives

- Reading multiple CSV files from DBFS
- DataFrame joins and column transformations
- Window functions (`row_number` , `rank` , `dense_rank`)
- Aggregations and SQL queries
- Data export to DBFS and visualization
