PYSPARK DATA ANALYSIS – EXTENSIVE EXERCISES

Select • Filter • GroupBy • Aggregates • Window Functions

DATASET

Domain: Multi-Store Retail Sales Analytics

Business Context (Tell Students)

You work as a data engineer for a retail chain operating across multiple regions and stores.
Each record represents a **single sales transaction**.

Your task is to analyze sales data to derive **regional trends, product performance, and rankings**.

STEP 1 — CREATE THE DATASET

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("Retail Sales Analysis") \
    .getOrCreate()


sales_data = [
    ("T001","North","Delhi","Store-01","Laptop","2024-01-01",75000),
    ("T002","North","Delhi","Store-01","Mobile","2024-01-02",32000),
    ("T003","North","Chandigarh","Store-02","Tablet","2024-01-03",26000),
    ("T004","South","Bangalore","Store-03","Laptop","2024-01-01",78000),
    ("T005","South","Chennai","Store-04","Mobile","2024-01-02",30000),
    ("T006","South","Bangalore","Store-03","Tablet","2024-01-03",24000),
    ("T007","East","Kolkata","Store-05","Laptop","2024-01-01",72000),
    ("T008","East","Kolkata","Store-05","Mobile","2024-01-02",28000),
    ("T009","East","Patna","Store-06","Tablet","2024-01-03",23000),
    ("T010","West","Mumbai","Store-07","Laptop","2024-01-01",80000),
    ("T011","West","Mumbai","Store-07","Mobile","2024-01-02",35000),
    ("T012","West","Pune","Store-08","Tablet","2024-01-03",27000),
    ("T013","North","Delhi","Store-01","Laptop","2024-01-04",76000),
    ("T014","South","Chennai","Store-04","Laptop","2024-01-04",79000),
    ("T015","East","Patna","Store-06","Mobile","2024-01-04",29000),
    ("T016","West","Pune","Store-08","Laptop","2024-01-04",77000),
    ("T017","North","Chandigarh","Store-02","Mobile","2024-01-05",31000),
    ("T018","South","Bangalore","Store-03","Mobile","2024-01-05",34000),
    ("T019","East","Kolkata","Store-05","Tablet","2024-01-05",25000),
    ("T020","West","Mumbai","Store-07","Tablet","2024-01-05",29000),
    ("T021","North","Delhi","Store-01","Tablet","2024-01-06",28000),
    ("T022","South","Chennai","Store-04","Tablet","2024-01-06",26000),
    ("T023","East","Patna","Store-06","Laptop","2024-01-06",74000),
    ("T024","West","Pune","Store-08","Mobile","2024-01-06",33000)
]

columns = [
    "txn_id","region","city","store_id",
    "product","sale_date","amount"
]

df_sales = spark.createDataFrame(sales_data, columns)
df_sales.show(5)
df_sales.printSchema()
```

EXERCISE SET 1 — SELECT OPERATIONS

Objective

Practice column selection, renaming, and derived columns.

Exercises

1. Select only `txn_id`, `region`, `product`, and `amount`
2. Rename `amount` to `revenue`

3. Create a derived column `amount_in_thousands`
4. Select distinct combinations of `region` and `product`
5. Select all columns but exclude `store_id`
6. Create a new column `sale_year` extracted from `sale_date`
7. Reorder columns in a business-friendly format

EXERCISE SET 2 — FILTER OPERATIONS

Objective

Understand row-level filtering and predicate pushdown.

Exercises

1. Filter transactions where amount > 50000
2. Filter only `Laptop` sales
3. Filter sales from `North` and `South` regions
4. Filter sales between 25000 and 75000
5. Filter transactions from `Delhi` stores only
6. Apply multiple filters using both `filter` and `where`
7. Change the order of filters and compare `explain(True)`
8. Identify which filters Spark pushes down

EXERCISE SET 3 — GROUPBY & AGGREGATE FUNCTIONS

Objective

Perform summarization and understand shuffles.

Exercises

1. Total sales amount per region
2. Average sales amount per product
3. Maximum sale per city

4. Minimum sale per store
5. Count of transactions per region
6. Total revenue per store
7. Region-wise product sales count
8. Average transaction value per city
9. Identify regions with total sales above a threshold
10. Use `explain(True)` and identify shuffle stages

EXERCISE SET 4 — MULTI-DIMENSIONAL AGGREGATION

Objective

Work with multiple grouping keys.

Exercises

1. Region + Product wise total sales
2. City + Store wise average sales
3. Region + City wise transaction count
4. Product + Store wise max sale
5. Identify top-selling product per region using aggregation only

---

EXERCISE SET 5 — WINDOW FUNCTIONS (OVER)

Objective

Perform analytics without reducing rows.

Import (Students must use)

```
from pyspark.sql.window import Window
from pyspark.sql.functions import sum, rank, row_number, dense_rank
```

Exercises

1. Compute running total of sales per region ordered by date
2. Rank transactions by amount within each region
3. Assign row numbers per store ordered by sale amount
4. Use dense rank to rank products per region
5. Identify top 2 highest sales per region using window functions
6. Compare `rank` vs `dense_rank` output
7. Calculate cumulative sales per store
8. Identify first and last transaction per city using windows

EXERCISE SET 7 — DAG & PERFORMANCE OBSERVATION

Exercises

1. Run `explain(True)` for:
   - Simple select
   - Filter
   - GroupBy
   - Window function
2. Identify:
   - Shuffles
   - Exchanges
   - Sorts
3. Explain why window functions introduce sorting