

# DATASET: Retail Customer Dataset (30 Records)

```
data = [
    ("C001", "Arjun",      "Hyderabad", 25, 45000, "Electronics"),
    ("C002", "Meera",      "Chennai",    32, 52000, "Grocery"),
    ("C003", "Rajesh",     "Bangalore",  29, 61000, "Clothing"),
    ("C004", "Priya",      "Delhi",       22, 38000, "Grocery"),
    ("C005", "Sanjay",     "Mumbai",     35, 72000, "Electronics"),
    ("C006", "Kavya",      "Hyderabad",  28, 48000, "Grocery"),
    ("C007", "Imran",      "Delhi",       31, 53000, "Clothing"),
    ("C008", "Divya",      "Chennai",    27, 45000, "Electronics"),
    ("C009", "Anil",       "Bangalore",  40, 85000, "Furniture"),
    ("C010", "Ritu",       "Mumbai",     23, 39000, "Clothing"),
    ("C011", "Hari",       "Hyderabad",  33, 56000, "Grocery"),
    ("C012", "Sana",       "Delhi",       26, 47000, "Electronics"),
    ("C013", "Vikram",     "Chennai",    38, 91000, "Furniture"),
    ("C014", "Deepa",      "Mumbai",     30, 62000, "Clothing"),
    ("C015", "Asha",       "Bangalore",  24, 41000, "Grocery"),
    ("C016", "Kiran",      "Delhi",       29, 59000, "Furniture"),
    ("C017", "Farah",      "Hyderabad",  36, 70000, "Clothing"),
    ("C018", "Tarun",      "Chennai",    28, 53000, "Furniture"),
    ("C019", "Nisha",      "Mumbai",     21, 35000, "Grocery"),
    ("C020", "Yusuf",      "Bangalore",  34, 76000, "Electronics"),
    ("C021", "Pooja",      "Delhi",       27, 47000, "Clothing"),
    ("C022", "Zara",       "Hyderabad",  32, 58000, "Grocery"),
    ("C023", "Ajay",       "Chennai",    30, 51000, "Furniture"),
    ("C024", "Reema",      "Bangalore",  28, 49000, "Clothing"),
    ("C025", "Gautam",     "Mumbai",     39, 82000, "Furniture"),
    ("C026", "Swati",      "Delhi",       25, 46000, "Electronics"),
    ("C027", "Mahesh",     "Hyderabad",  41, 90000, "Furniture"),
    ("C028", "Anita",      "Chennai",    26, 44000, "Clothing"),
    ("C029", "Sameer",     "Bangalore",  33, 68000, "Electronics"),
    ("C030", "Leela",      "Delhi",       22, 36000, "Grocery")
]

columns = ["customer_id", "name", "city", "age", "annual_spend", "category"]
```

```
df = spark.createDataFrame(data, columns)
df.show()
```

---

# PYSPARK BASIC EXERCISES (MEDIUM LEVEL)

---

These exercises are strictly basic but with a larger dataset. Suitable for Lesson 1 and Lesson 2 practice.

---

## Exercise 1

Show the first 10 customers.

---

## Exercise 2

List all unique cities.

Hint:

```
df.select("city").distinct()
```

---

## Exercise 3

Display only customer\_id, name, and annual\_spend columns.

---

## Exercise 4

Filter all customers who spend more than 60000 annually.

---

## Exercise 5

Show all customers from Delhi who are younger than 30.

---

## Exercise 6

Create a new column named "spend\_lakh" = annual\_spend / 100000.

---

## Exercise 7

Create a new column "customer\_type"

Logic:

- spend > 70000 → Premium
- else → Standard

Use when() and otherwise().

---

## Exercise 8

Show customers whose name starts with the letter A.

---

## Exercise 9

Filter customers where category is either Clothing or Electronics.

---

## Exercise 10

Convert the city name to uppercase using the upper() function.

---

## Exercise 11

Remove the category column from the DataFrame.

---

## Exercise 12

Sort customers by age in descending order.

---

## Exercise 13

Find customers who are younger than the average age of the entire dataset.

Steps:

1. Calculate average age.

2. Filter df where age < avg\_age.

---

## Exercise 14

Display the top 5 highest spending customers.

---

## Exercise 15

Create a new DataFrame containing only customers from Mumbai.

---

# OPTIONAL BONUS EXERCISES

---

## Bonus Exercise 1

Extract the first letter of each name and store in a new column.

## Bonus Exercise 2

Mask the customer\_id.

Example: C001 → \*\*\*01

## Bonus Exercise 3

Create a new boolean column "is\_senior" where age > 35.

## Bonus Exercise 4

Count number of customers per city (uses groupBy).

---