- ▯ Aggregations
- ▯ GroupBy
- ▯ Joins
- ▯ Date fields
- ▯ Saving to files

▯ Dataset 1: `employee_data`

(Reuse from previous step — no changes needed)

▯ Dataset 2: `performance_data`

Prepare in notebook:

```
performance = [
    ("Ananya", 2023, 4.5),
    ("Rahul", 2023, 4.9),
    ("Priya", 2023, 4.3),
    ("Zoya", 2023, 3.8),
    ("Karan", 2023, 4.1),
    ("Naveen", 2023, 4.7),
    ("Fatima", 2023, 3.9)
]
columns_perf = ["Name", "Year", "Rating"]

df_perf = spark.createDataFrame(performance, columns_perf)
```

▯ PySpark Exercises – Set 2 (Advanced)

▯ GroupBy and Aggregations

1. Get the average salary by department.
2. Count of employees per department.
3. Maximum and minimum salary in Engineering.

▯ Join and Combine Data

4. Perform an inner join between `employee_data` and `performance_data` on `Name`.
5. Show each employee’s salary and performance rating.
6. Filter employees with rating > 4.5 and salary > 60000.

▯ Window & Rank (Bonus Challenge)

7. Rank employees by salary department-wise.
8. Calculate cumulative salary in each department.

▯ Date Operations

9. Add a new column `JoinDate` with random dates between 2020 and 2023.
10. Add column `YearsWithCompany` using `current_date()` and `datediff()`.

Writing to Files

11. Write the full employee DataFrame to CSV with headers.
12. Save the joined DataFrame to a Parquet file.
