
PySpark Task Set – Part 3

Domain: HR & Workforce Analytics Focus: DataFrame APIs, Joins, SQL, Date Logic, Aggregation, UDFs, Views

---

Dataset 1 – `employees.csv`

```
EmpID,Name,Department,JoinDate,Salary,ManagerID
1,Anita,HR,2021-05-01,55000,
2,Raj,Engineering,2020-03-15,80000,1
3,Simran,Engineering,2022-07-10,75000,1
4,Aamir,Marketing,2019-11-20,60000,1
5,Nisha,HR,2023-01-05,50000,1
```

Dataset 2 – `attendance.csv`

```
EmpID,Date,Status
1,2024-04-01,Present
1,2024-04-02,Present
2,2024-04-01,Absent
2,2024-04-02,Present
3,2024-04-01,Present
3,2024-04-02,Present
4,2024-04-01,Absent
4,2024-04-02,Absent
5,2024-04-01,Present
5,2024-04-02,Present
```

Dataset 3 – `bonuses.json`

```
[
  {"EmpID": 1, "Year": 2023, "Bonus": 5000},
  {"EmpID": 2, "Year": 2023, "Bonus": 7000},
  {"EmpID": 3, "Year": 2023, "Bonus": 6500},
  {"EmpID": 4, "Year": 2023, "Bonus": 6000},
  {"EmpID": 5, "Year": 2023, "Bonus": 4000}
]
```

TASKS

---

1. Ingestion & Exploration

- Read all 3 files (CSV + JSON) using PySpark.
- Show schemas and sample records.
- Count distinct departments.

---

2. DataFrame Operations

- Add a column `TenureYears` using `datediff()` and `round()`.
- Calculate `TotalCompensation = Salary + Bonus`.
- Filter employees with more than 2 years in the company.
- Show employees who report to a manager (`ManagerID is not null`).

□ 3. Aggregation

- Average salary per department.
- Number of employees under each manager.
- Count of absences per employee.

□ 4. Joins

- Join `employees` and `attendance` → Get attendance % (Present days / Total days).
- Join `employees` and `bonuses` → Show top 3 employees by TotalCompensation.
- Multi-level join: `employees` + `bonuses` + `attendance`.

□ 5. String & Date Functions

- Extract year and month from `JoinDate`.
- Mask employee names using regex.
- Use `substring()` to create `EmpCode` like "EMP001".

□ 6. Conditional & Null Handling

- Use `when/otherwise` to label performance:
    - “High” if Bonus > 6000
    - “Medium” if 4000–6000
    - “Low” otherwise
- Handle missing ManagerID using `fillna("No Manager")`.

□ 7. Spark SQL

- Create and use database `hr`.
- Save all DataFrames as tables: `employees`, `attendance`, `bonuses`.
- Write SQL queries:
    - Top paid employee in each department.
    - Attendance rate by department.
    - Employees joined after 2021 with salary > □70,000.

□ 8. Advanced (Optional)

- Use a UDF to classify department as "Tech" vs "Non-Tech".
- Create a view `emp_attendance_summary`.
- Save it as Parquet partitioned by `Department`.
