**Introduction to Task 10**

In this task, we are moving from Spark's **RDD-based programming** to a more **user-friendly and powerful tool called PySpark DataFrames**.

Unlike RDDs (which require more code and manual effort), **DataFrames** in PySpark behave like **tables** in a database or like **Excel spreadsheets** — they have **rows and columns**, and you can run SQL-like operations easily.

**Task 10 is divided into two parts:**

We will use **PySpark** (Spark with Python) to do two things:

### Part 1 (on a CSV file)

* Take a small employee CSV (table-like text file).
* Load it into a **Spark DataFrame** (a table in memory).
* Do simple analytics:
  1. Average salary per department.
  2. For each employee, show **% difference** from their department’s average salary (“Salary Increase”).
  3. Calculate **Years with Company** from join\_date.
  4. Label each employee’s **Salary Category** (Low / Medium / High) using simple thresholds.

### Part 2 (on a “database”)

* Treat **Hive** (built into our Cloudera VM) as the “database.”
* Load the same data into a Hive table.
* Read it into Spark as a DataFrame, then:
  1. Print basic details (schema, row count, simple stats).
  2. Check for **missing values** column-by-column.
  3. **Replace** missing values (e.g., with mean for numbers, most frequent for text).
  4. Save the **cleaned** result **back** to the database (Hive) as a new table.