TareaLayla / clase_02 / **practica_b_solucion.ipynb**

264 lines (264 loc) · 7.81 KB

In [0]:
```
from pyspark.sql import SparkSession

# Crear una sesión de Spark
spark = SparkSession.builder \
    .appName("Spark SQL Exercises") \
    .getOrCreate()
```

In [0]:
```
# Datos iniciales
data = [
    ("John Doe", "Engineering", 80000, "2023-01-15"),
    ("Jane Smith", "Marketing", 95000, "2022-11-23"),
    ("Alice Johnson", "Engineering", 85000, "2021-03-08"),
    ("Bob Brown", "Sales", 72000, "2024-02-25"),
    ("Charlie Davis", "Marketing", 90000, "2023-05-19")
]

# Esquema de columnas
columns = ["name", "department", "salary", "hire_date"]

# Crear el DataFrame
df = spark.createDataFrame(data, columns)

# Registrar el DataFrame como una tabla temporal llamada "employees"
df.createOrReplaceTempView("employees")
```

In [0]:
```
print("Ejercicio 1: Consultar todos los empleados")
spark.sql("SELECT * FROM employees").show()
```

```
Ejercicio 1: Consultar todos los empleados
+-------------+-----------+------+----------+
|         name| department|salary| hire_date|
+-------------+-----------+------+----------+
|     John Doe|Engineering| 80000|2023-01-15|
|   Jane Smith|  Marketing| 95000|2022-11-23|
|Alice Johnson|Engineering| 85000|2021-03-08|
|    Bob Brown|      Sales| 72000|2024-02-25|
|Charlie Davis|  Marketing| 90000|2023-05-19|
+-------------+-----------+------+----------+
```

In [0]:
```
print("Ejercicio 2: Empleados del departamento de Marketing")
spark.sql("SELECT * FROM employees WHERE department = 'Marketing'").sho
```

```
Ejercicio 2: Empleados del departamento de Marketing
+-------------+----------+------+----------+
|         name|department|salary| hire_date|
+-------------+----------+------+----------+
|   Jane Smith| Marketing| 95000|2022-11-23|
|Charlie Davis| Marketing| 90000|2023-05-19|
+-------------+----------+------+----------+
```

In [0]:
```
print("Ejercicio 3: Ordenar empleados por salario (descendente)")
spark.sql("SELECT * FROM employees ORDER BY salary DESC").show()
```

```
Ejercicio 3: Ordenar empleados por salario (descendente)
+-------------+-----------+------+----------+
|         name| department|salary| hire_date|
+-------------+-----------+------+----------+
|   Jane Smith|  Marketing| 95000|2022-11-23|
|Charlie Davis|  Marketing| 90000|2023-05-19|
|Alice Johnson|Engineering| 85000|2021-03-08|
|     John Doe|Engineering| 80000|2023-01-15|
|    Bob Brown|      Sales| 72000|2024-02-25|
+-------------+-----------+------+----------+
```

In [0]:

```
print("Ejercicio 4: Salario promedio por departamento")
spark.sql("""
    SELECT department, AVG(salary) AS avg_salary
    FROM employees
    GROUP BY department
""").show()
```

```
Ejercicio 4: Salario promedio por departamento
+-----------+----------+
| department|avg_salary|
+-----------+----------+
|Engineering|   82500.0|
|  Marketing|   92500.0|
|      Sales|   72000.0|
```



```
    SELECT *
    FROM employees
    WHERE hire_date > '2023-01-01'
""").show()
```

```
Ejercicio 5: Empleados contratados después del 1 de enero de 2023
+-------------+-----------+------+----------+
|         name| department|salary| hire_date|
+-------------+-----------+------+----------+
|     John Doe|Engineering| 80000|2023-01-15|
|    Bob Brown|      Sales| 72000|2024-02-25|
|Charlie Davis|  Marketing| 90000|2023-05-19|
+-------------+-----------+------+----------+
```

In [0]: