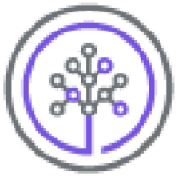
Hoja de trucos del Módulo 3: Apache Spark

Paquete/Método	Descripción	Ejemplo de código
appName()	Un nombre para tu trabajo que se mostrará en la interfaz web del clúster.	`from pyspark.sql import SparkSession` `spark = SparkSession.builder.appName("MyApp").getOrCreate()`
cache()	Una transformación de Apache Spark que se usa a menudo en un DataFrame, conjunto de datos o RDD cuando deseas realizar múltiples acciones. cache() almacena en caché el DataFrame, conjunto de datos o RDD especificado en la memoria de los trabajadores de tu clúster. Dado que cache() es una transformación, la operación de almacenamiento en caché se realiza solo cuando también se usa una acción de Spark (por ejemplo, count(), show(), take() o write()) en el mismo DataFrame, conjunto de datos o RDD en una sola acción.	`df = spark.read.csv("customer.csv")` `df.cache()`
count()	Devuelve el número de elementos con el valor especificado.	`count = df.count()` `print(count)`
createTempView()	Crea una vista temporal que se puede usar más tarde para consultar los datos. El único parámetro requerido es el nombre de la vista.	`df.createOrReplaceTempView("cust_tbl")`
filter()	Devuelve un iterador donde los elementos son filtrados a través de una función para probar si el elemento es aceptado o no.	`filtered_df = df.filter(df['age'] > 30)`
getOrCreate()	Obtiene o instancia un SparkContext y lo registra como un objeto singleton.	`spark = SparkSession.builder.getOrCreate()`
import	Se utiliza para hacer que el código de un módulo sea accesible en otro. Las importaciones en Python son cruciales para una estructura de código exitosa. Puedes reutilizar código y mantener tus proyectos manejables utilizando importaciones de manera efectiva, lo que puede aumentar tu productividad.	`from pyspark.sql import SparkSession`

len()	Devuelve el número de elementos en un objeto. Cuando el objeto es una cadena, la función len() devuelve el número de caracteres en la cadena.	`row_count = len(df.collect())` `print(row_count)`
map()	Devuelve un objeto de mapa (un iterador) de los resultados después de aplicar la función dada a cada elemento de un iterable dado (lista, tupla, etc.)	`rdd = df.rdd.map(lambda row: (row['name'],` `row['age']))`
pip	Para asegurar que las solicitudes funcionen, el programa pip busca el paquete en el Índice de Paquetes de Python (PyPI), resuelve cualquier dependencia e instala todo en tu entorno Python actual.	`pip list`
pip install	El comando pip install <package> busca la última versión del paquete e instala.	`pip install pyspark`
print()	Imprime el mensaje especificado en la pantalla u otro dispositivo de salida estándar. El mensaje puede ser una cadena u otro objeto; el objeto se convertirá en una cadena antes de ser escrito en la pantalla.	`print("Hello, PySpark!")`
printSchema()	Se utiliza para imprimir o mostrar el esquema del DataFrame o conjunto de datos en formato de árbol junto con el nombre de la columna y el tipo de dato. Si tienes un DataFrame o conjunto de datos con una estructura anidada, muestra el esquema en un formato de árbol anidado.	`df.printSchema()`
sc.parallelize()	Crea una colección paralelizada. Distribuye una colección local de Python para formar un RDD. Se recomienda usar range si la entrada representa un rango para un mejor rendimiento.	`rdd = sc.parallelize([1, 2, 3, 4, 5])`

select()	Se utiliza para seleccionar una o múltiples columnas, columnas anidadas, columna por índice, todas las columnas de la lista, mediante expresiones regulares de un DataFrame. select() es una función de transformación en Spark y devuelve un nuevo DataFrame con las columnas seleccionadas.	`selected_df = df.select('name', 'age')`
show()	La función show() de Spark DataFrame se utiliza para mostrar el contenido del DataFrame en un formato de tabla con filas y columnas. Por defecto, muestra solo veinte filas, y los valores de las columnas se truncarán a veinte caracteres.	`df.show()`
spark.read.json	Spark SQL puede inferir automáticamente el esquema de un conjunto de datos JSON y cargarlo como un DataFrame. La función read.json() carga datos de un directorio de archivos JSON donde cada línea de los archivos es un objeto JSON. Ten en cuenta que el archivo ofrecido como un archivo JSON no es un archivo JSON típico.	`json_df = spark.read.json("customer.json")`
spark.sql()	Para emitir cualquier consulta SQL, usa el método sql() en la instancia de SparkSession. Todas las consultas spark.sql ejecutadas de esta manera devuelven un DataFrame sobre el cual puedes realizar más operaciones de Spark si es necesario.	`result = spark.sql("SELECT name, age FROM cust_tbl WHERE age > 30")` `result.show()`

time()	Devuelve la hora actual en el número de segundos desde la Época Unix.	`from pyspark.sql.functions import current_timestamp`
		`current_time = df.select(current_timestamp().alias("current_time"))`
		`current_time.show()`



Skills Network