Master PySpark: From Zero to Big Data Hero!!

PySpark DataFrame Manipulation part 2: Adding, Renaming, and Dropping Columns

1. Adding New Columns with withColumn()

In PySpark, the withColumn() function is widely used to add new columns to a DataFrame. You can either assign a constant value using lit() or perform transformations using existing columns.

- Add a constant value column:

```
newdf = df.withColumn("NewColumn", lit(1))
```

- Add a column based on an expression:

```
newdf = df.withColumn("withinCountry", expr("Country == 'India'"))
```

This function allows adding multiple columns, including calculated ones:

- Example:
  - Assign a constant value with lit().
  - Perform calculations using existing columns like multiplying values.

2. Renaming Columns with withColumnRenamed()

PySpark provides the withColumnRenamed() method to rename columns. This is especially useful when you want to change the names for clarity or to follow naming conventions:

- Renaming a column:

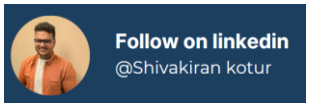
```
new_df = df.withColumnRenamed("oldColumnName", "newColumnName")
```

- Handling column names with special characters or spaces: If a column has special characters or spaces, you need to use backticks (`) to escape it:

```
newdf.select("`New Column Name`").show()
```

3. Dropping Columns with drop()

To remove unwanted columns, you can use the drop() method:



- Drop a single column:

```
df2 = df.drop("Country")
```

- Drop multiple columns:

```
df2 = df.drop("Country", "Region")
```

Dropping columns creates a new DataFrame, and the original DataFrame remains unchanged.

4. Immutability of DataFrames

In Spark, DataFrames are immutable by nature. This means that after creating a DataFrame, its contents cannot be changed. All transformations like adding, renaming, or dropping columns result in a new DataFrame, keeping the original one intact.

- For instance, dropping columns creates a new DataFrame without altering the original:

```
newdf = df.drop("ItemType", "SalesChannel")
```

This immutability ensures data consistency and supports Spark's parallel processing, as transformations do not affect the source data.

Key Points

- Use withColumn() for adding columns, with lit() for constant values and expressions for computed values.
- Use withColumnRenamed() to rename columns and backticks for special characters or spaces.
- Use drop() to remove one or more columns.
- DataFrames are immutable in Spark—transformations result in new DataFrames, leaving the original unchanged.