Read/Write to SQL Server using Databricks

(PySpark)

Requirements

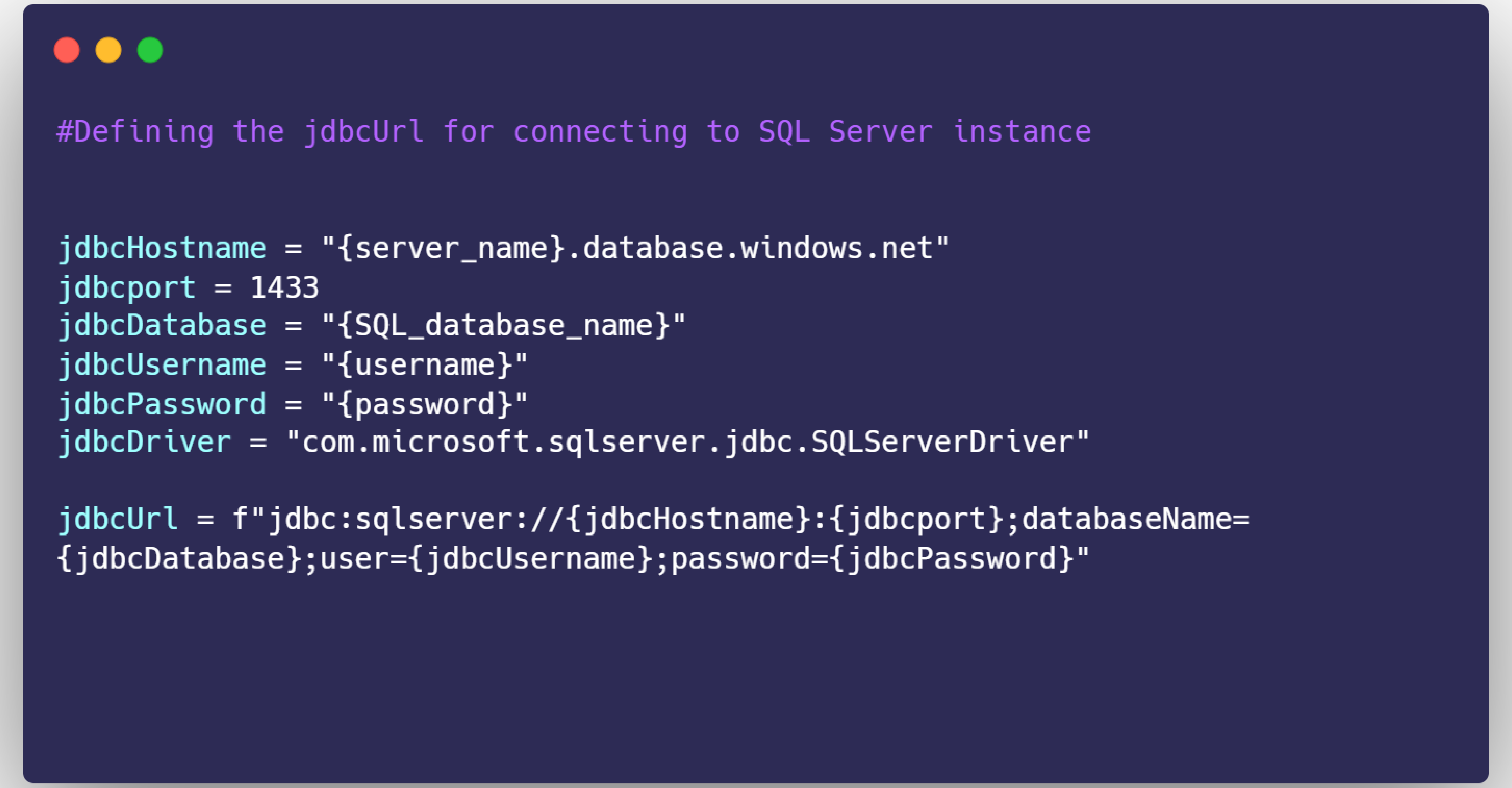
1 - Azure SQL database instance

2 - Azure Databricks Instance

SQL

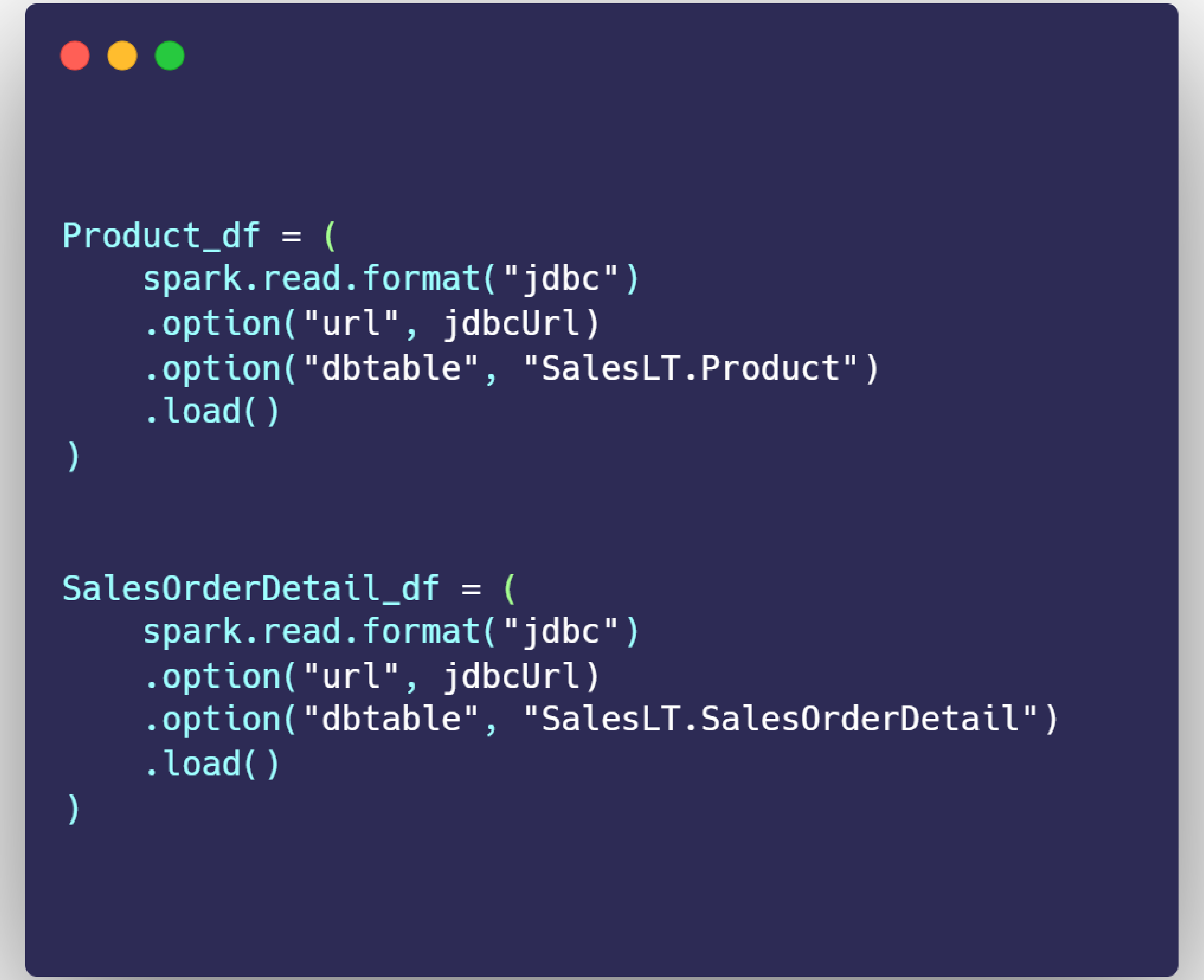
databricks

Defining the jdbcUrl for connecting from Databricks Notebook to SQL Server instance



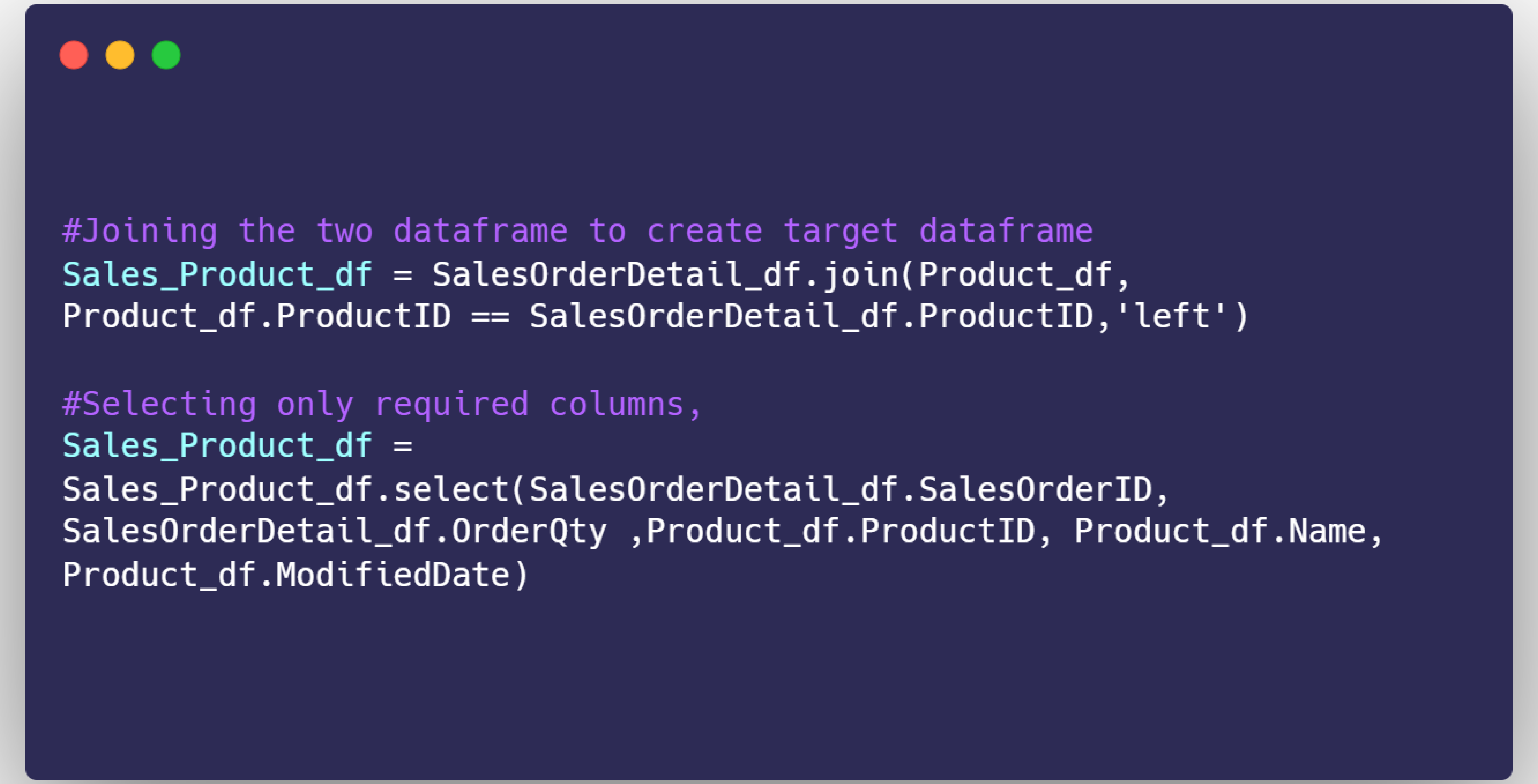
```
#Defining the jdbcUrl for connecting to SQL Server instance


jdbcHostname = "{server_name}.database.windows.net"
jdbcport = 1433
jdbcDatabase = "{SQL_database_name}"
jdbcUsername = "{username}"
jdbcPassword = "{password}"
jdbcDriver = "com.microsoft.sqlserver.jdbc.SQLServerDriver"

jdbcUrl = f"jdbc:sqlserver://{jdbcHostname}:{jdbcport};databaseName=
{jdbcDatabase};user={jdbcUsername};password={jdbcPassword}"
```

Reading the tables from SQL db and creating dataframe



```
Product_df = (
    spark.read.format("jdbc")
    .option("url", jdbcUrl)
    .option("dbtable", "SalesLT.Product")
    .load()
)


SalesOrderDetail_df = (
    spark.read.format("jdbc")
    .option("url", jdbcUrl)
    .option("dbtable", "SalesLT.SalesOrderDetail")
    .load()
)
```

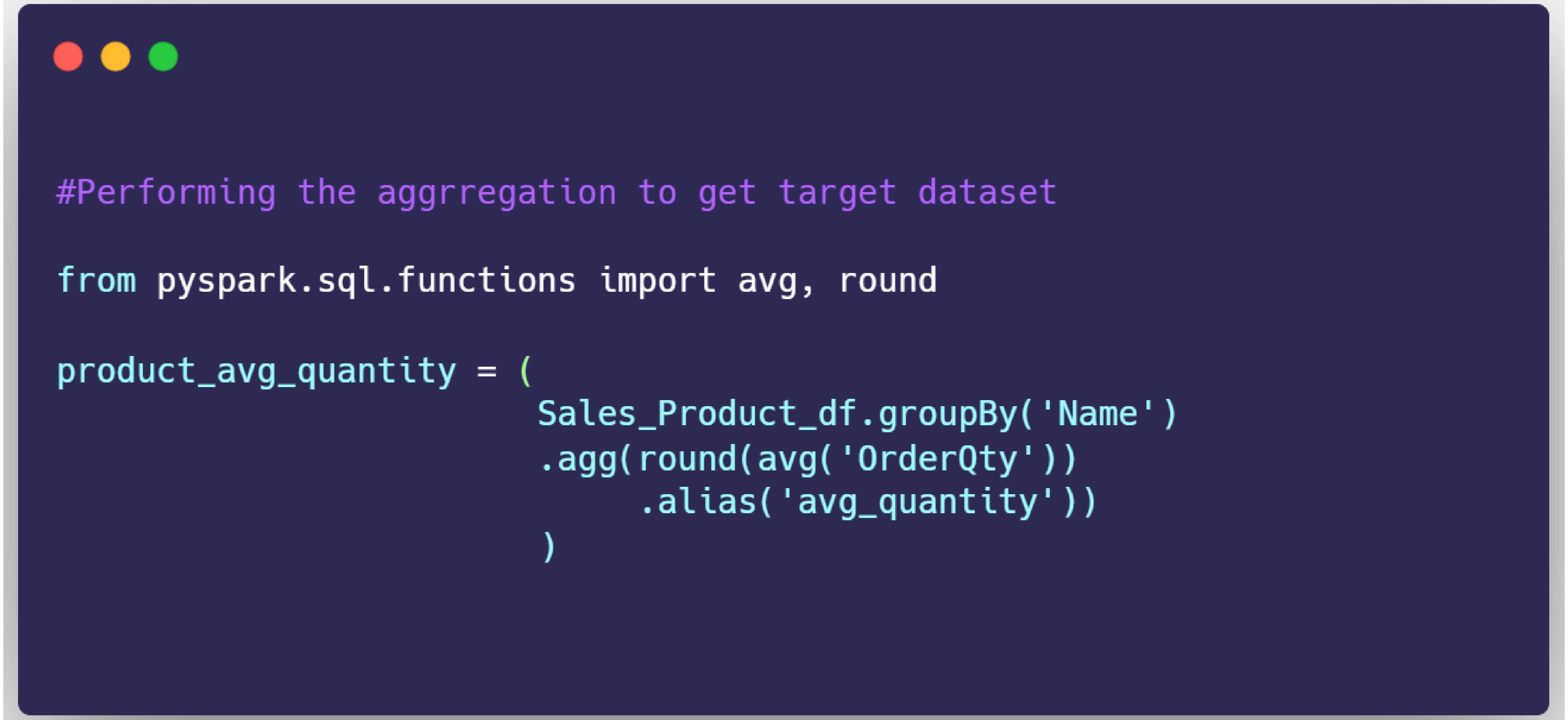
Joining the two dataframes to create the required dataframe



```
#Joining the two dataframe to create target dataframe
Sales_Product_df = SalesOrderDetail_df.join(Product_df,
Product_df.ProductID == SalesOrderDetail_df.ProductID,'left')

#Selecting only required columns,
Sales_Product_df =
Sales_Product_df.select(SalesOrderDetail_df.SalesOrderID,
SalesOrderDetail_df.OrderQty ,Product_df.ProductID, Product_df.Name,
Product_df.ModifiedDate)
```

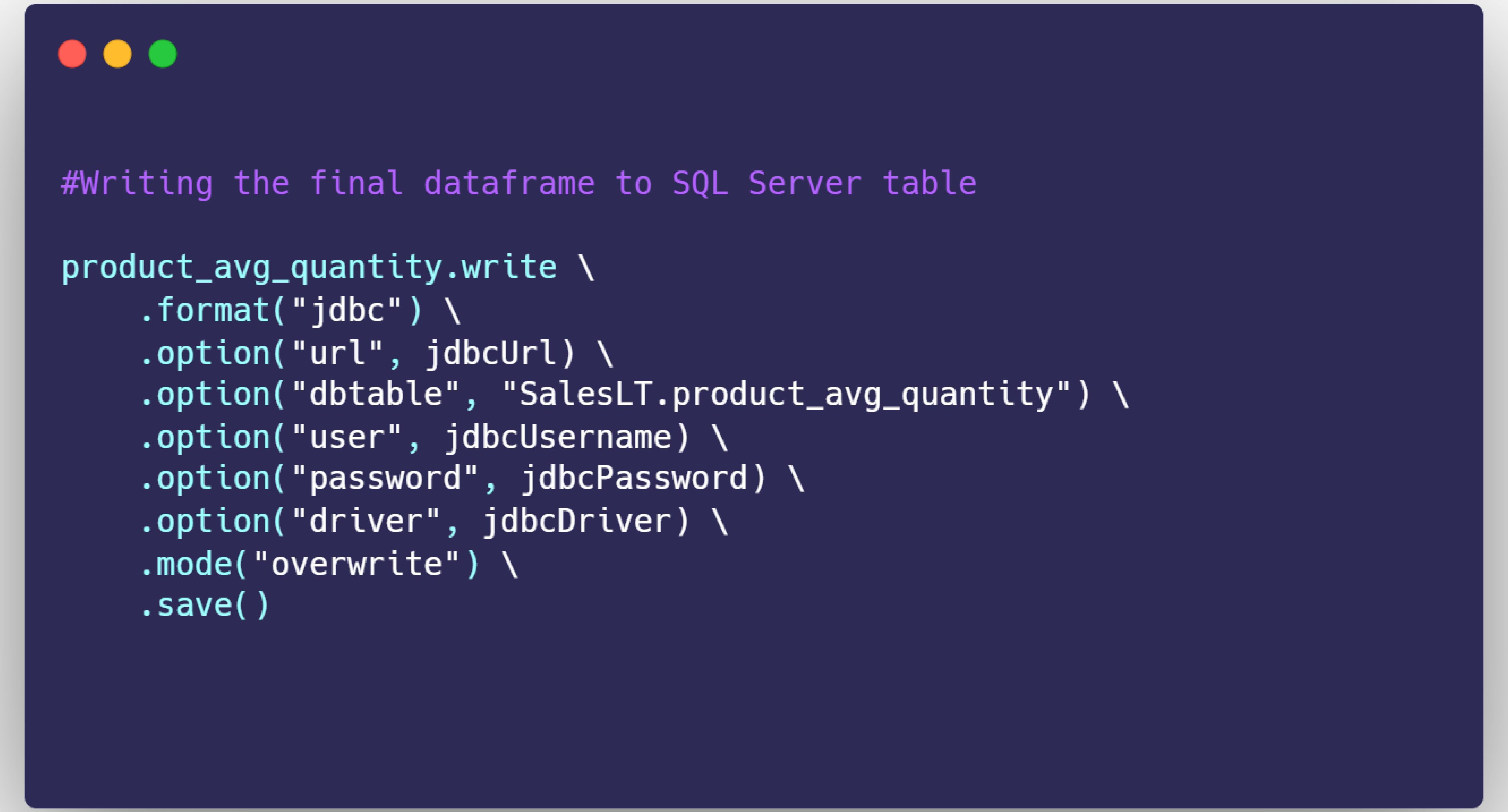
Performing the transformations on derived dataframe to get final dataset

Aggregating data based on product name and finding avg quantity ordered for each product

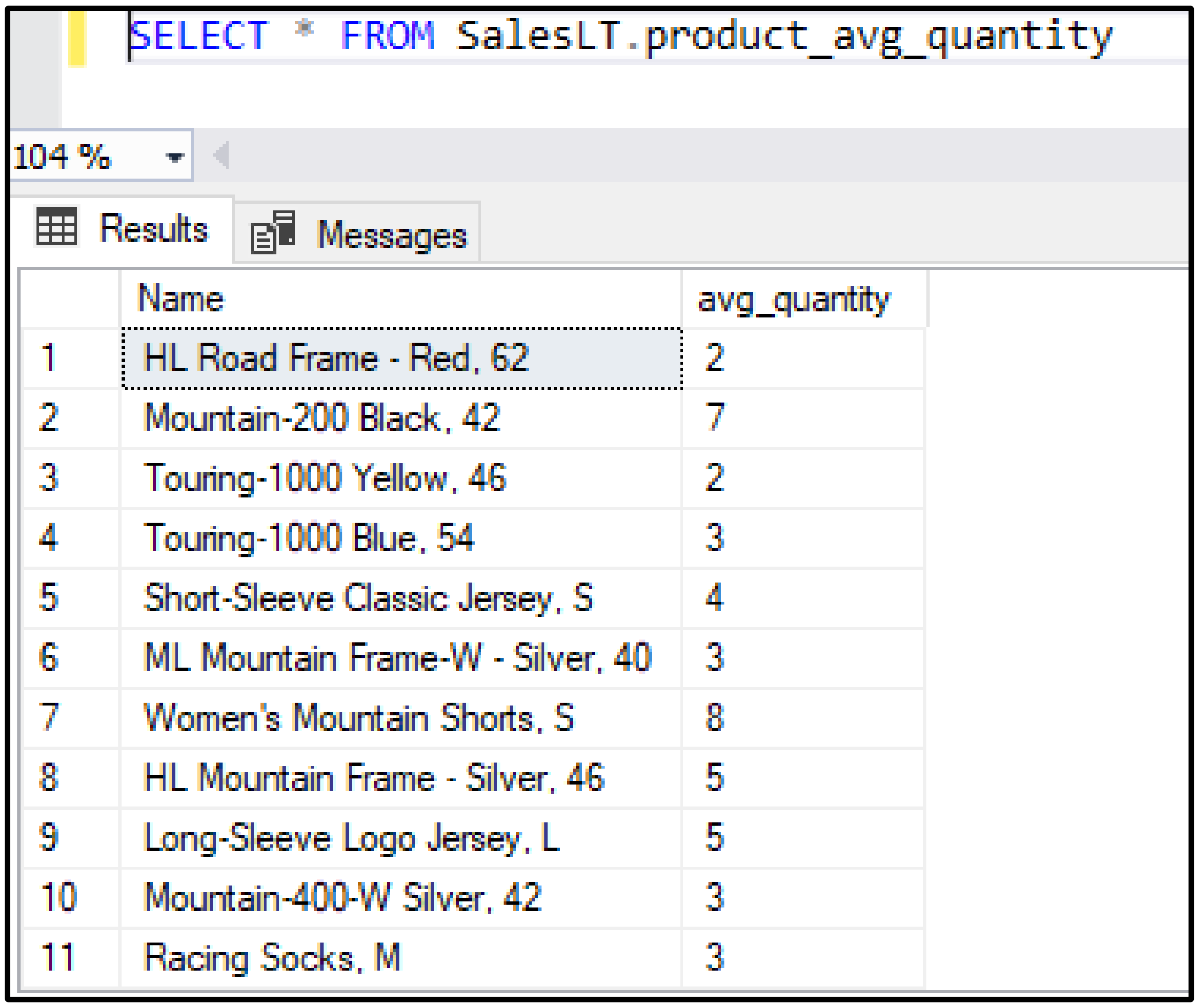


```
#Performing the aggrregation to get target dataset

from pyspark.sql.functions import avg, round

product_avg_quantity = (
                        Sales_Product_df.groupBy('Name')
                        .agg(round(avg('OrderQty'))
                             .alias('avg_quantity'))
                        )
```

Writing the dataframe to SQL Server table



```
#Writing the final dataframe to SQL Server table

product_avg_quantity.write \
    .format("jdbc") \
    .option("url", jdbcUrl) \
    .option("dbtable", "SalesLT.product_avg_quantity") \
    .option("user", jdbcUsername) \
    .option("password", jdbcPassword) \
    .option("driver", jdbcDriver) \
    .mode("overwrite") \
    .save()
```



```
SELECT * FROM SalesLT.product_avg_quantity
```

	Name	avg_quantity
1	HL Road Frame - Red, 62	2
2	Mountain-200 Black, 42	7
3	Touring-1000 Yellow, 46	2
4	Touring-1000 Blue, 54	3
5	Short-Sleeve Classic Jersey, S	4
6	ML Mountain Frame-W - Silver, 40	3
7	Women's Mountain Shorts, S	8
8	HL Mountain Frame - Silver, 46	5
9	Long-Sleeve Logo Jersey, L	5
10	Mountain-400-W Silver, 42	3
11	Racing Socks, M	3