## Retail Store:

```
from pyspark.sql import SparkSession

from pyspark.sql.functions import col

from pyspark.sql import functions as F


spark = SparkSession.builder.appName("RetailStore").getOrCreate()


retail_df = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/content/sample_data/retail_data.csv")
```

**# 1. Calculate the Total Revenue per Category**

```
total_revenue_per_category = retail_df.withColumn("total_revenue", col("price") * col("quantity")).groupBy("category").agg(F.sum("total_revenue").alias("total_revenue"))


total_revenue_per_category.show()
```

**# 2. Filter Transactions Where the Total Sales Amount is Greater Than $100**

```
high_transactions = retail_df.withColumn("total_sales", col("price") * col("quantity")).filter(col("total_sales") > 100)


high_transactions.show()
```

```
# 3. Find the Most Sold Product

most_sold_product = retail_df.groupBy("product_name").agg(F.sum("quantity").alias("total_quantity")).orderBy(col("total_quantity").desc()).limit(1)
most_sold_product.show()

# 4. Calculate the Average Price per Product Category

avg_price_category = retail_df.groupBy("category").agg(F.avg("price").alias("average_price"))
avg_price_category.show()

# 5. Find the Top 3 Highest Grossing Products

top_grossing_products = retail_df.withColumn("total_revenue", col("price") * col("quantity")).groupBy("product_name").agg(F.sum("total_revenue").alias("total_revenue")) \
.orderBy(col("total_revenue").desc()).limit(3)

top_grossing_products.show()

# 6. Calculate the Total Number of Items Sold per Day

items_sold_perDay = retail_df.groupBy("sales_date").agg(F.sum("quantity").alias("total_quantity"))
items_sold_perDay.show()
```

**# 7. Identify the Product with the Lowest Price in Each Category**

```
lowest_cost = retail_df.groupBy("category").agg(F.min("price").alias("price"))
lowest_cost.show()
```

**# 8. Calculate the Total Revenue for Each Product**

```
revenue_product = retail_df.withColumn("total_revenue", col("price") *
col("quantity")).groupBy("product_name").agg(F.sum("total_revenue").alias("t
otal_revenue"))
revenue_product.show()
```

**# 9. Find the Total Sales per Day for Each Category**

```
total_sales_per_category = retail_df.withColumn("total_sales", col("price") *
col("quantity")).groupBy("sales_date",
"category").agg(F.sum("total_sales").alias("total_sales"))

total_sales_per_category.show()
```

**# 10. Create a New Column for Discounted Price**

```
retail_df = retail_df.withColumn("discounted_price", col("price") * 0.9)

retail_df.show()
```