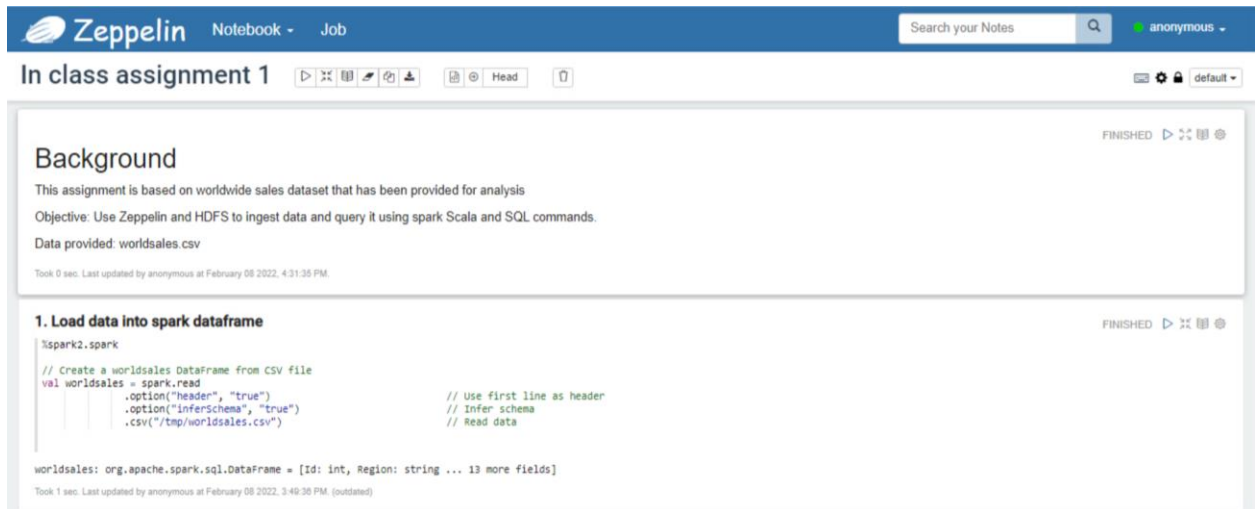


World Sales Analysis using Apache Spark and SQL API

Load data into a Spark dataframe



Zeppelin Notebook - Job

Search your Notes

anonymous

In class assignment 1

FINISHED

Background

This assignment is based on worldwide sales dataset that has been provided for analysis

Objective: Use Zeppelin and HDFS to ingest data and query it using spark Scala and SQL commands.

Data provided: worldsales.csv

Took 0 sec. Last updated by anonymous at February 08 2022, 4:31:35 PM.

1. Load data into spark dataframe

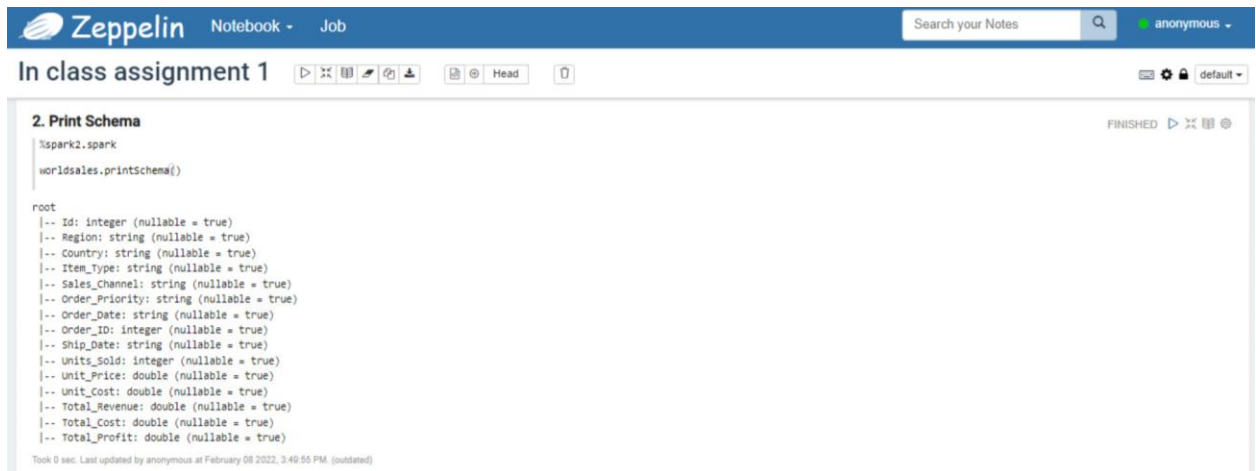
FINISHED

```
%spark2.spark
// Create a worldsales DataFrame from csv file
val worldsales = spark.read
  .option("header", "true")           // Use first line as header
  .option("inferSchema", "true")      // Infer schema
  .csv("/tmp/worldsales.csv")         // Read data

worldsales: org.apache.spark.sql.DataFrame = [Id: int, Region: string ... 13 more fields]
```

Took 1 sec. Last updated by anonymous at February 08 2022, 3:49:36 PM. (outdated)

Print the dataframe schema



Zeppelin Notebook - Job

Search your Notes

anonymous

In class assignment 1

FINISHED

2. Print Schema

```
%spark2.spark
worldsales.printSchema()

root
 |-- Id: integer (nullable = true)
 |-- Region: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- Item_Type: string (nullable = true)
 |-- Sales_Channel: string (nullable = true)
 |-- Order_Priority: string (nullable = true)
 |-- Order_Date: string (nullable = true)
 |-- Order_ID: integer (nullable = true)
 |-- Ship_Date: string (nullable = true)
 |-- Units_Sold: integer (nullable = true)
 |-- Unit_Price: double (nullable = true)
 |-- Unit_Cost: double (nullable = true)
 |-- Total_Revenue: double (nullable = true)
 |-- Total_Cost: double (nullable = true)
 |-- Total_Profit: double (nullable = true)
```

Took 0 sec. Last updated by anonymous at February 08 2022, 3:49:55 PM. (outdated)

Filter the dataframe to show units sold greater than 8000 and unit cost greater than 500 ("&&" operator can be used for multiple "AND" conditions)

Zepplin

Notebook · Job

Search your Notesanonymous

In class assignment 1

▶ 🔍 📄 🗑️ ⚙️

🔖 Head 🗑️

3. Units sold greater than 8000 and unit cost greater than 500

FINISHED ▶ 🔍 📄 🗑️ ⚙️

```
%spark2.spark  
  
val filteredWorldSales = worldSales  
.filter($"Units_Sold" > 8000 && $"Unit_Cost" > 500)  
  
filteredWorldSales.show()  
  
filteredWorldSales: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Id: int, Region: string ... 13 more fields]  
  
| Id |          Region | Country | Item_Type | Sales_Channel | Order_Priority | Order_Date | Order_ID | Ship_Date | Units_Sold | Unit_Price | Unit_Cost | Total_Revenue | Total_Cost | Total_Profit |  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 26 | Sub-Saharan Africa | Senegal | Household | Offline | L | 8/27/2012 | 247802054 | 9/8/2012 | 8989 | 668.27 | 502.54 | 6007879.03 | 4517332.06 | 1489746.97 |  
| 37 | Sub-Saharan Africa | Swaziland | Office Supplies | Offline | H | 10/3/2013 | 405785882 | 10/22/2013 | 9915 | 651.21 | 524.96 | 6456747.15 | 5204978.4 | 1251768.75 |
```

Took 0 sec. Last updated by anonymous at February 08 2022, 4:07:07 PM.

In class assignment 1

Create SQL view

```
%spark2
worldsales.createOrReplaceTempView("salesview");
```

Took 0 sec. Last updated by anonymous at February 08 2022, 4:00:07 PM. (outdated)

4. Aggregate by region and count

```
%spark2.sql
SELECT Region, count(*) as CountRegionwise FROM Salesview
GROUP BY Region
ORDER BY count(*) DESC
```

Region	CountRegionwise
Sub-Saharan Africa	15
Europe	12
Middle East and North Africa	6
Central America and the Caribbean	6
Asia	5
North America	3
Australia and Oceania	2

Zepplin

Notebook - Job

Search your Notes 🔍 anonymous ▾

In class assignment 1 ▶ ⌕ ↻ 📄 👤

🔖 Head 🗑️

5. Create a dataframe with the group by sales countFINISHED ▶ ⌕ ↻ 🔧

%spark2

```
val regionworldsales = worldsales.groupBy("Region")\n                                   .count()
```

regionworldsales: org.apache.spark.sql.DataFrame = [Region: string, count: bigint]
Took 0 sec. Last updated by anonymous at February 08 2022, 4:08:42 PM. (outdated)

%spark2

```
regionworldsales.show()
```

Region	count
Middle East and N...	6
Australia and Oce...	2
Europe	12
Sub-Saharan Africa	15
Central America a...	6
North America	3
Asia	5

Create two views using the “createOrReplaceTempView” command

View on “Salesview” from the first dataframe

View on “Regionview” from the second dataframe

Zeppelin Notebook Job Search your Notes anonymous

In class assignment 1

6. Save as csv FINISHED

```
%spark2
regionworldsales.coalesce(1).write.format("csv").option("header","true").save("/tmp/regionworldsales.csv")
```

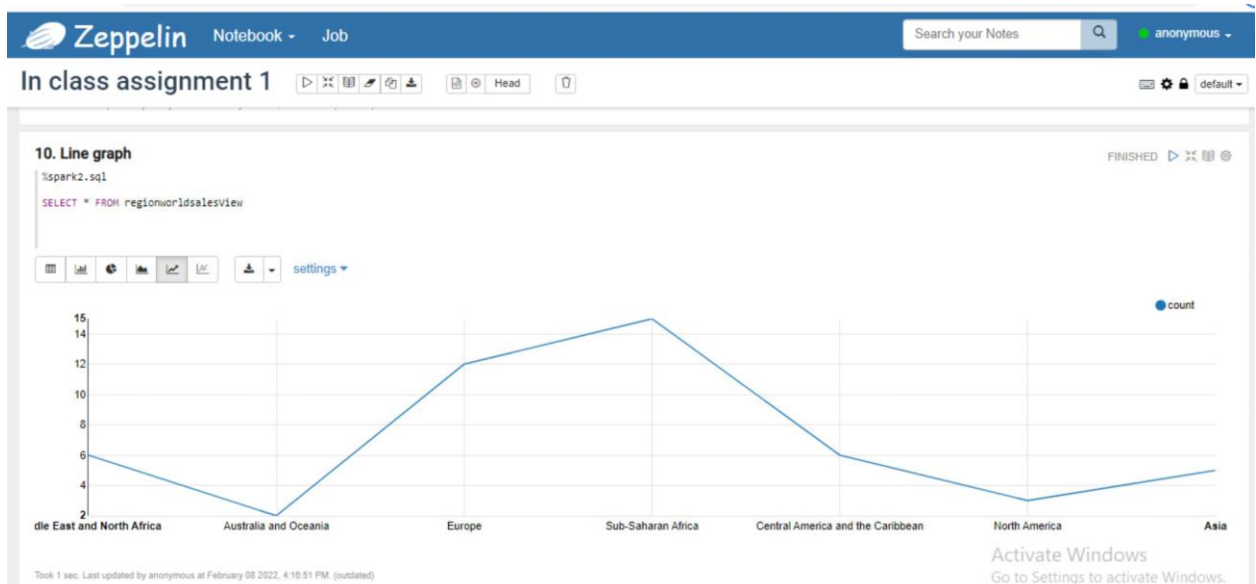
Took 1 sec. Last updated by anonymous at February 08 2022, 4:12:42 PM. (outdated)

7,8 & 9. Create views FINISHED

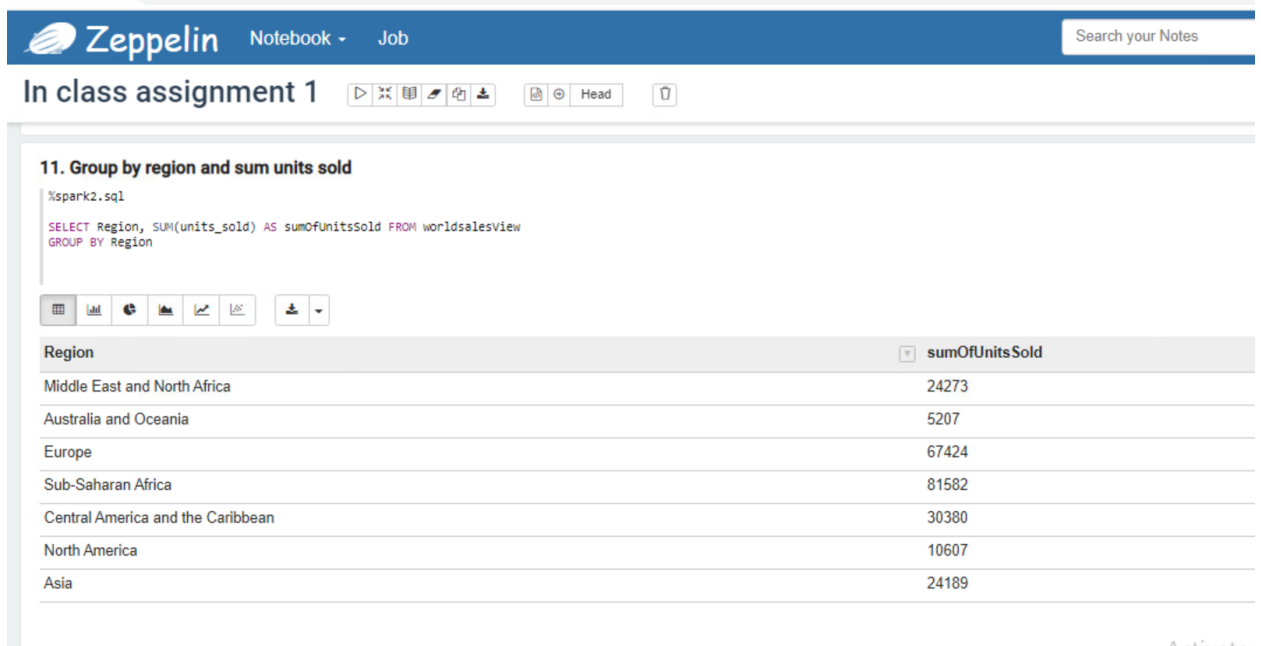
```
%spark2
worldsales.createOrReplaceTempView("worldsalesview");
regionworldsales.createOrReplaceTempView("regionworldsalesview");
```

Took 0 sec. Last updated by anonymous at February 08 2022, 4:15:37 PM. (outdated)

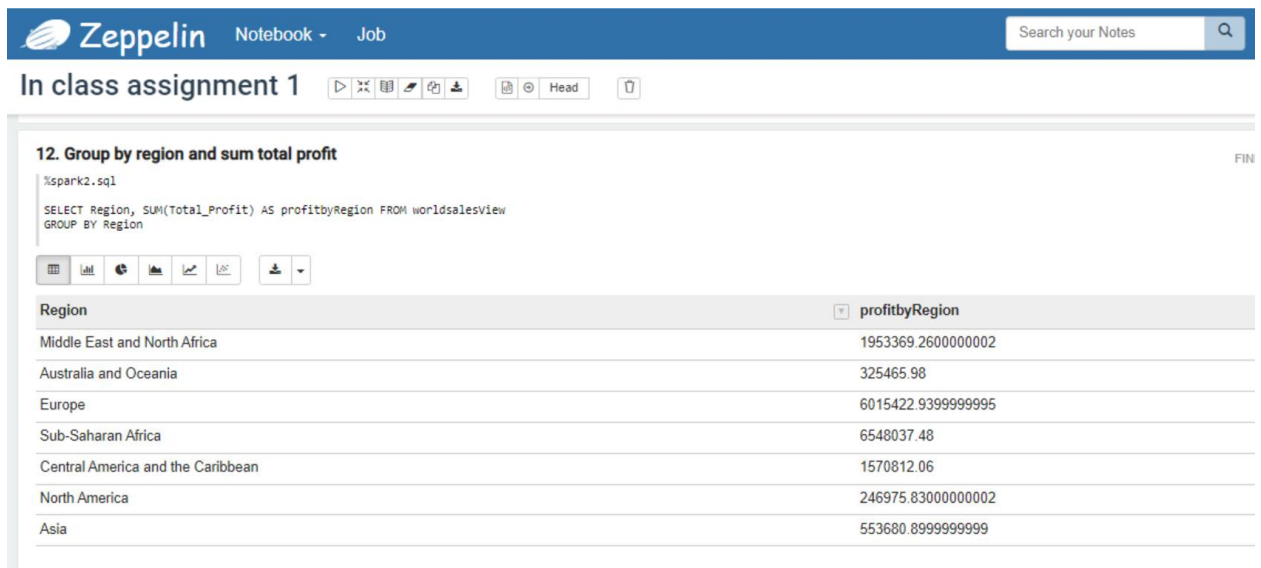
Using SQL select all from “Regionview” view and show in a line graph.



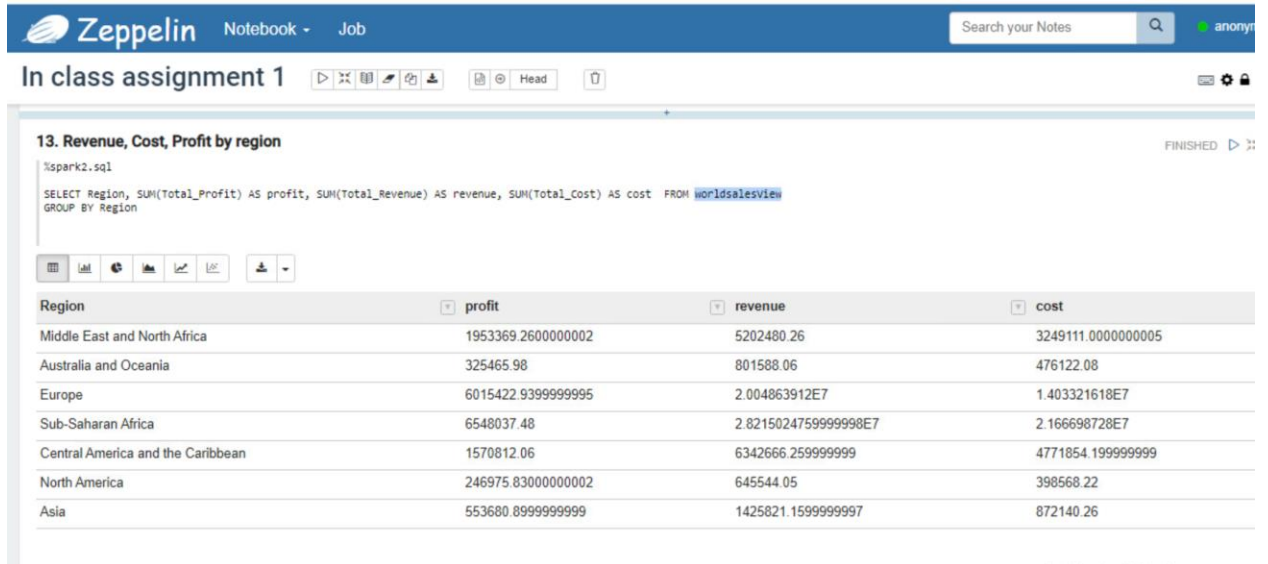
Using SQL, from the “Salesview” view, Select the region and sum of units sold, and group by region



Using SQL select from the “Salesview” view – the region and sum of total_profit and group by region and display in a Bar chart



Using SQL select from the “Salesview” view – show the total profit as profit, the total revenue as revenue and the total cost as cost from “Salesview”, group by region



The client is in the process of opening a new store and they are looking at the best location to do so -

They need to see the avg profit in each region as a percentage (pie chart) compared to other regions.

