databricks pyspark-project Business intelligence

(https://databricks.com)

```
spark
```

SparkSession - hive

SparkContext

Spark UI

Version
v3.3.2
Master
local[8]
AppName
Databricks Shell

```
from pyspark.sql.types import StringType, StructField, StructType, IntegerType, DateType
from pyspark.sql.functions import year, month, quarter
from pyspark.sql.functions import sum, col, count, countDistinct
```

```
sales_schema = StructType([
    StructField("product_id", IntegerType(), True),
    StructField("customer_id", StringType(), True),
    StructField("order_date", DateType(), True),
    StructField("location", StringType(), True),
    StructField("source_order", StringType(), True)
])
```

```
sales_df = spark.read.format("csv")\
    .option("header","false")\
    .option("inferSchema","false")\
    .schema(sales_schema)\
    .load("/FileStore/tables/sales_csv.txt")
```

```
sales_df.show(5)
```

```
+----------+-----------+----------+--------+------------+
|product_id|customer_id|order_date|location|source_order|
+----------+-----------+----------+--------+------------+
|         1|          A|2023-01-01|   India|      Swiggy|
|         2|          A|2022-01-01|   India|      Swiggy|
|         2|          A|2023-01-07|   India|      Swiggy|
|         3|          A|2023-01-10|   India|  Restaurant|
|         3|          A|2022-01-11|   India|      Swiggy|
+----------+-----------+----------+--------+------------+
only showing top 5 rows
```

```
sales_df = sales_df.withColumn("order_year", year(sales_df.order_date))
```

```
sales_df.show(5)
```

```
+----------+-----------+----------+--------+------------+----------+
|product_id|customer_id|order_date|location|source_order|order_year|
+----------+-----------+----------+--------+------------+----------+
|         1|          A|2023-01-01|   India|      Swiggy|      2023|
|         2|          A|2022-01-01|   India|      Swiggy|      2022|
|         2|          A|2023-01-07|   India|      Swiggy|      2023|
|         3|          A|2023-01-10|   India|  Restaurant|      2023|
|         3|          A|2022-01-11|   India|      Swiggy|      2022|
```

```
+----------+-----------+----------+--------+------------+----------+
only showing top 5 rows
```

```
  sales_df.printSchema()
```

```
root
 |-- product_id: integer (nullable = true)
 |-- customer_id: string (nullable = true)
 |-- order_date: date (nullable = true)
 |-- location: string (nullable = true)
 |-- source_order: string (nullable = true)
 |-- order_year: integer (nullable = true)
```

```
  sales_df = sales_df.withColumn("order_month", month(sales_df.order_date))
  sales_df = sales_df.withColumn("order_quater", quarter(sales_df.order_date))
```

```
  sales_df.show(5)
```

```
+----------+-----------+----------+--------+------------+----------+-----------+------------+
|product_id|customer_id|order_date|location|source_order|order_year|order_month|order_quater|
+----------+-----------+----------+--------+------------+----------+-----------+------------+
|         1|          A|2023-01-01|   India|      Swiggy|      2023|          1|           1|
|         2|          A|2022-01-01|   India|      Swiggy|      2022|          1|           1|
|         2|          A|2023-01-07|   India|      Swiggy|      2023|          1|           1|
|         3|          A|2023-01-10|   India|  Restaurant|      2023|          1|           1|
|         3|          A|2022-01-11|   India|      Swiggy|      2022|          1|           1|
+----------+-----------+----------+--------+------------+----------+-----------+------------+
only showing top 5 rows
```

```
  menu_schema = StructType([
      StructField("product_id", IntegerType(), True),
      StructField("product_name", StringType(), True),
      StructField("prize", StringType(), True),
  ])
```

```
  menu_df = spark.read.format("csv")\
      .option("header","false")\
      .option("inferSchema","true")\
      .schema(menu_schema)\
      .load("/FileStore/tables/menu_csv.txt")
```

```
  menu_df.show()
```

```
+----------+------------+-----+
|product_id|product_name|prize|
+----------+------------+-----+
|         1|       PIZZA|  100|
|         2|     Chowmin|  150|
|         3|    sandwich|  120|
|         4|        Dosa|  110|
|         5|     Biryani|   80|
|         6|       Pasta|  180|
+----------+------------+-----+
```

```
  sales_df.show(10)
```

```
+----------+-----------+----------+--------+------------+----------+-----------+------------+
|product_id|customer_id|order_date|location|source_order|order_year|order_month|order_quater|
+----------+-----------+----------+--------+------------+----------+-----------+------------+
|         1|          A|2023-01-01|   India|      Swiggy|      2023|          1|           1|
```

```
|         2|          A|2022-01-01|   India|      Swiggy|      2022|          1|           1|
|         2|          A|2023-01-07|   India|      Swiggy|      2023|          1|           1|
|         3|          A|2023-01-10|   India|  Restaurant|      2023|          1|           1|
|         3|          A|2022-01-11|   India|      Swiggy|      2022|          1|           1|
|         3|          A|2023-01-11|   India|  Restaurant|      2023|          1|           1|
|         2|          B|2022-02-01|   India|      Swiggy|      2022|          2|           1|
|         2|          B|2023-01-02|   India|      Swiggy|      2023|          1|           1|
|         1|          B|2023-01-04|   India|  Restaurant|      2023|          1|           1|
|         1|          B|2023-02-11|   India|      Swiggy|      2023|          2|           1|
+----------+-----------+----------+--------+------------+----------+-----------+------------+
only showing top 10 rows
```

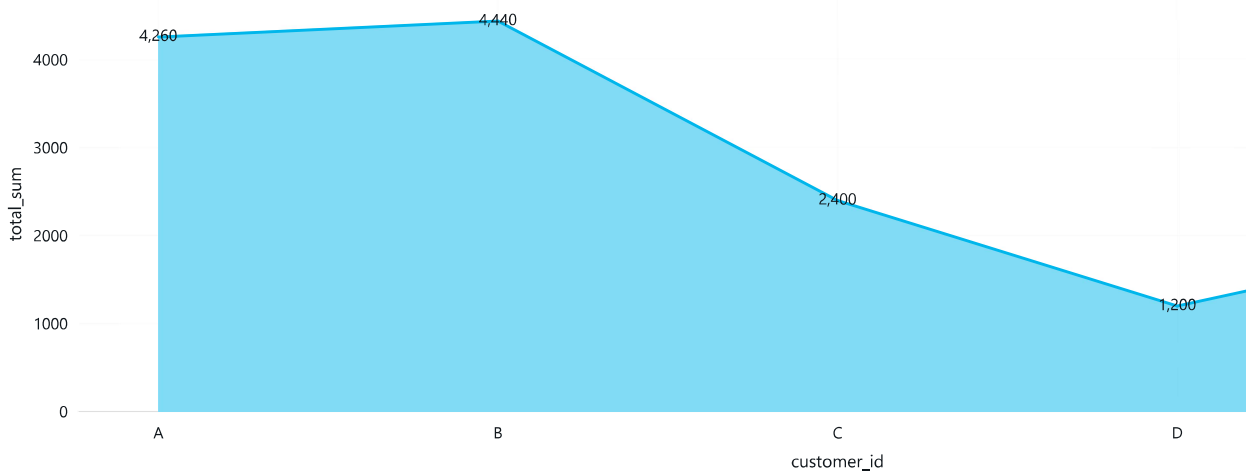
```
menu_df.show()
```

```
+----------+------------+-----+
|product_id|product_name|prize|
+----------+------------+-----+
|         1|       PIZZA|  100|
|         2|     Chowmin|  150|
|         3|    sandwich|  120|
|         4|        Dosa|  110|
|         5|     Biryani|   80|
|         6|       Pasta|  180|
+----------+------------+-----+
```

Total amount spend by each customer

```
total_amount_spend = (sales_df.join(menu_df, 'product_id'). groupBy('customer_id').agg(sum(col('prize').cast('float')).alias('total_s
display(total_amount_spend)
```

Table | Visualization 1



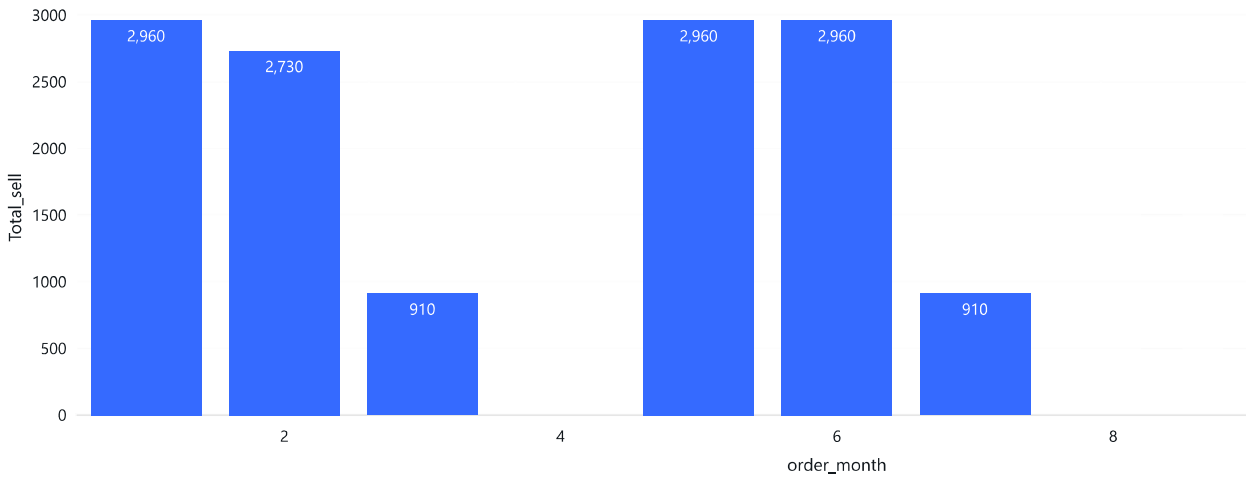
5 rows

Total amount sell in each month

```
each_month_sell = (sales_df.join(menu_df,('product_id')).groupBy('order_month').agg(sum(col('prize').cast('float')).alias('Total_sell
                  )

display(each_month_sell)
```

Table | Visualization 1



7 rows

```
each_month_sell.show()
```

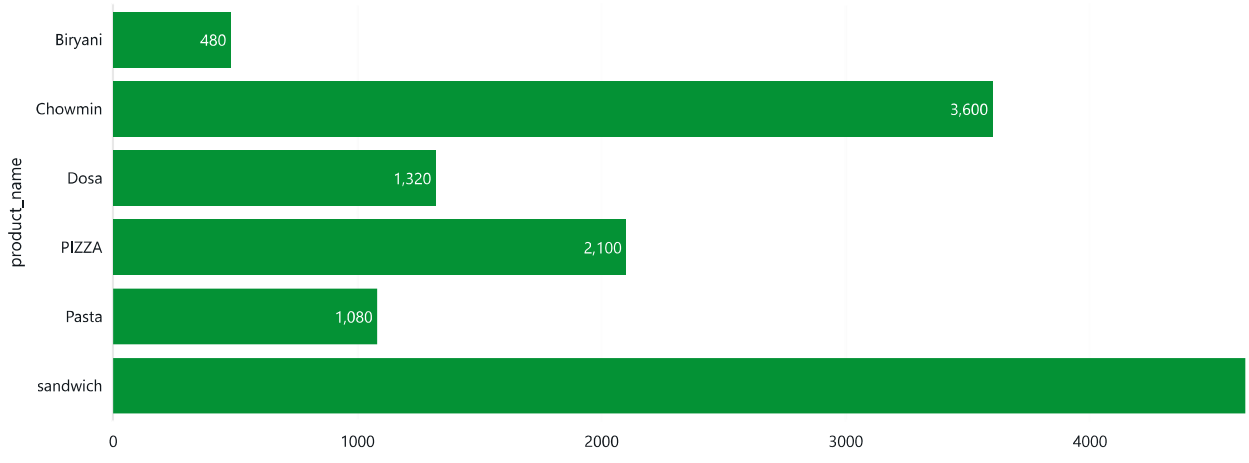
```
+-----------+----------+
|order_month|Total_sell|
+-----------+----------+
|          1|    2960.0|
|          2|    2730.0|
|          3|     910.0|
|          5|    2960.0|
|          6|    2960.0|
|          7|     910.0|
|         11|     910.0|
+-----------+----------+
```

Total amount spend by each food category

```
each_food_sell = sales_df.join(menu_df,'product_id').groupBy('product_id','product_name').agg(sum(col('prize').cast('float')).alias('

display(each_food_sell)
```

Table | Visualization 1



total_sell

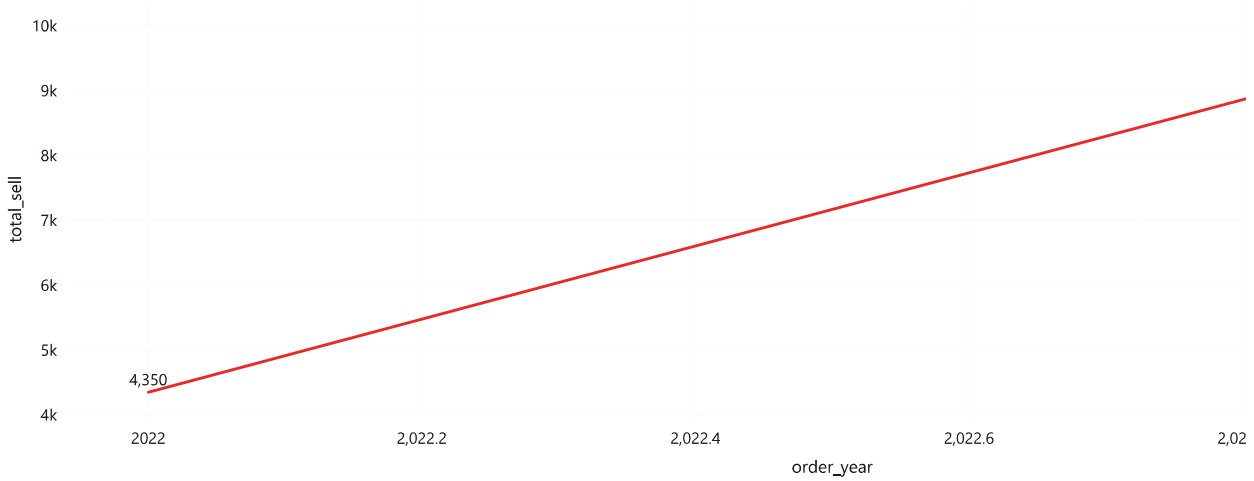
6 rows

Yearly Sale

```
yearly_sale = sales_df.join(menu_df,'product_id').groupBy('order_year').agg(sum(col('prize').cast('float')).alias('total_sell'))

display(yearly_sale)
```

Table | Visualization 1

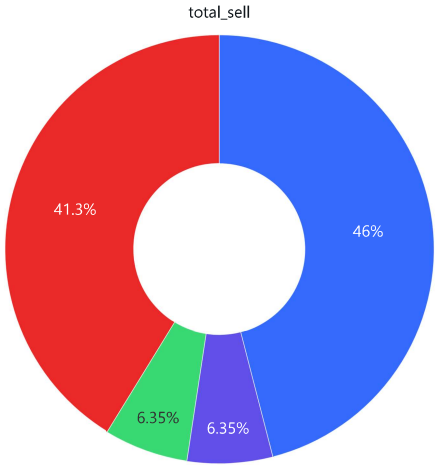


2 rows

```
quaterly_sale = sales_df.join(menu_df,'product_id').groupBy('order_quater').agg(sum(col('prize').cast('float')).alias('total_sell')).

display(quaterly_sale)
```

Table | Visualization 1



4 rows

Total number of order by each category

```
sell_by_each_product = sales_df.join(menu_df,'product_id').groupBy('product_id', 'product_name').agg(count('product_id').alias('produ
('product_count', ascending = 0).drop('product_id')

display(sell_by_each_product)
```

Table | Visualization 1

Steps	Value	% Max	% Previous
sandwich	48	100%	100%
Chowmin	24	50%	50%
PIZZA	21	43.75%	87.50%
Dosa	12	25%	57.14%
Pasta	6	12.50%	50%
Biryani	6	12.50%	100%

6 rows

Top ordered item

```
order_top_product = sales_df.join(menu_df,'product_id').groupBy('product_name').agg(count('product_id').alias('order_count')).orderBy
limit(1)
display(order_top_product)
```

Table | Visualization 1

48

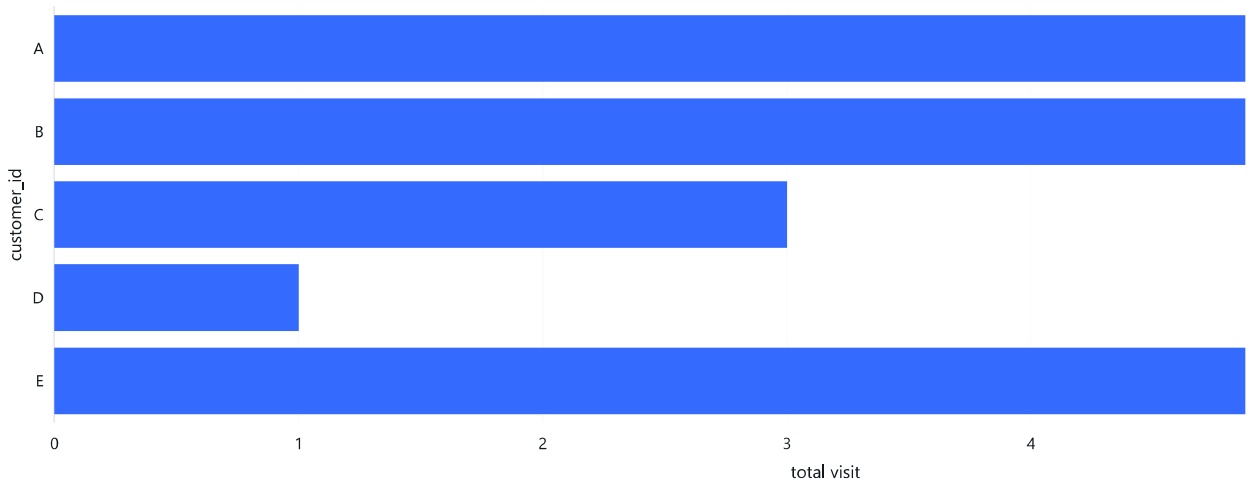
(sandwich)

1 row

Frequency of customer visited to Restourant

```
customer_visited_restourant = sales_df.filter(sales_df.source_order == 'Restaurant').groupBy('customer_id').agg(countDistinct('order_
display(customer_visited_restourant)
```

Table | Visualization 1 | Visualization 2



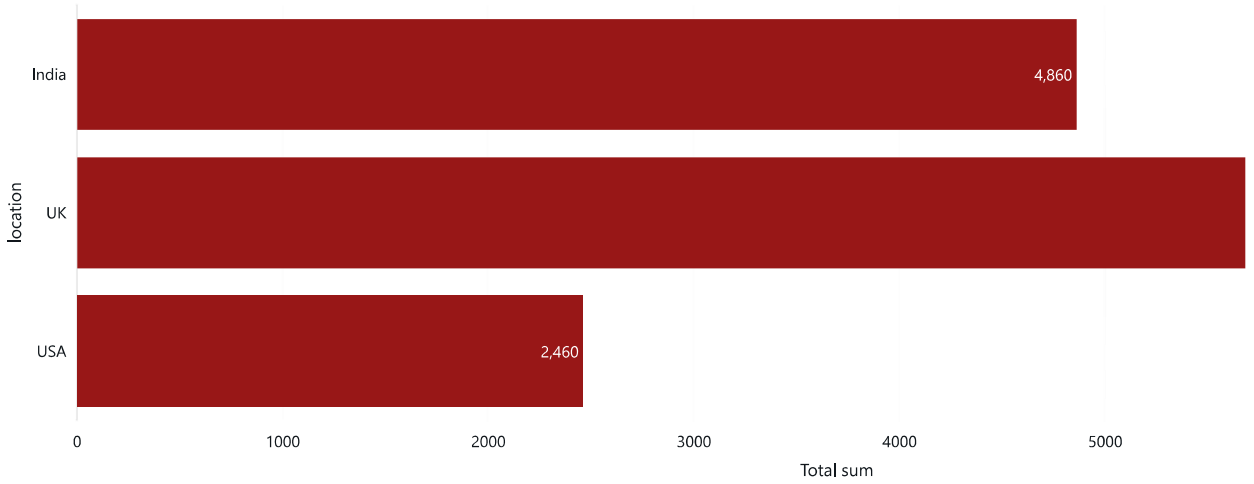
5 rows

total sales by each country

```
sales_each_country = sales_df.join(menu_df,'product_id').groupBy('location').agg(sum(col('prize').cast('float')).alias('Total sum'))

display(sales_each_country)
```

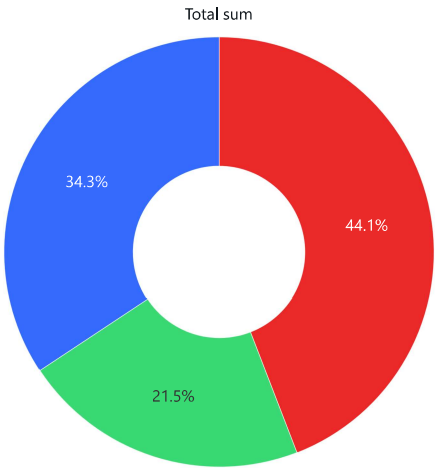
Table | Visualization 1



3 rows

total sales by each source

Table | Visualization 1



3 rows