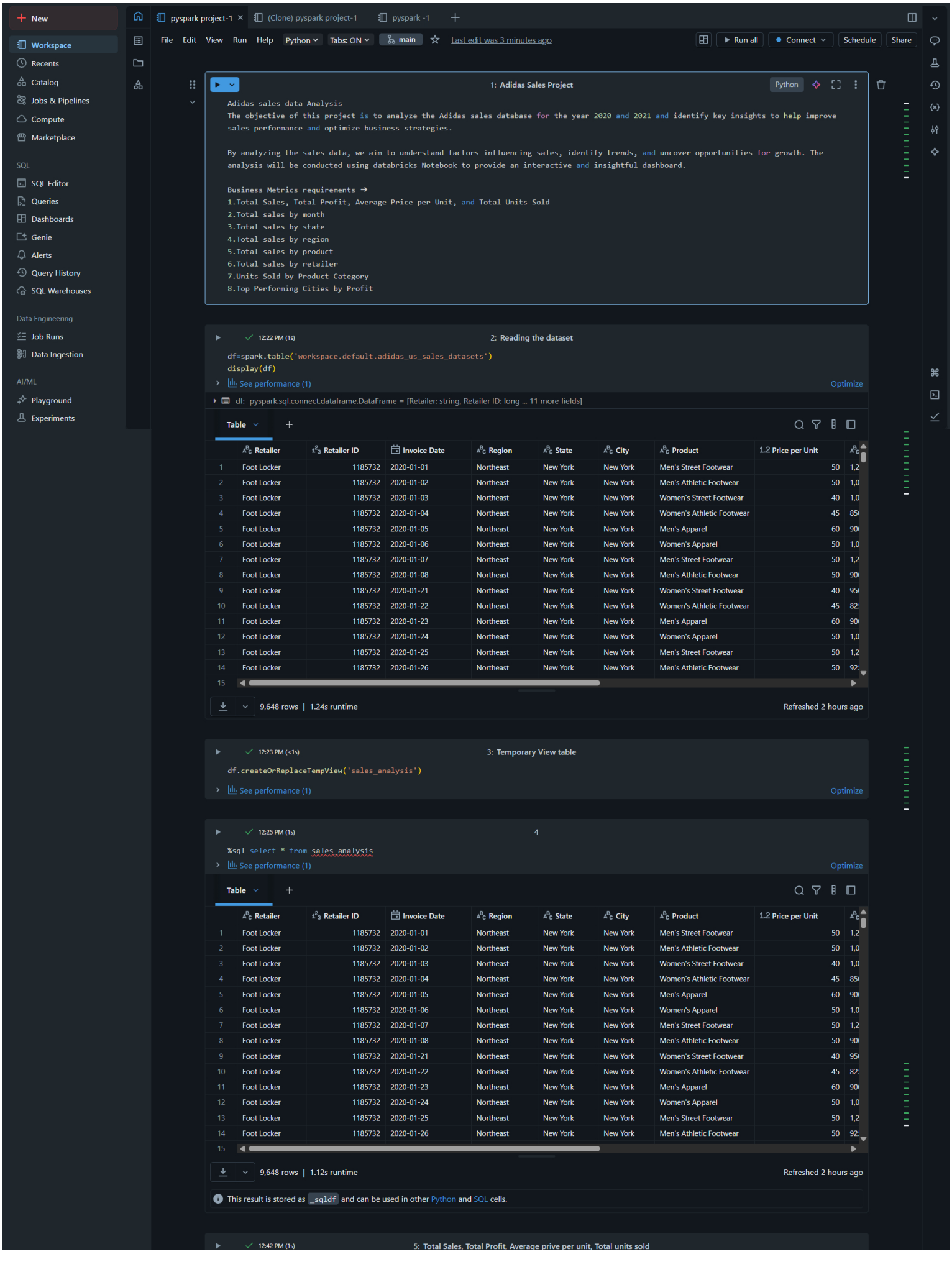

New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
AI/ML
Playground
Experiments
pyspark project-1
(Clone) pyspark project-1
pyspark -1
File Edit View Run Help Python Tabs: ON main Last edit was 3 minutes ago
Run all Connect Schedule Share
1: Adidas Sales Project
Python
Adidas sales data Analysis
The objective of this project is to analyze the Adidas sales database for the year 2020 and 2021 and identify key insights to help improve sales performance and optimize business strategies.
By analyzing the sales data, we aim to understand factors influencing sales, identify trends, and uncover opportunities for growth. The analysis will be conducted using databricks Notebook to provide an interactive and insightful dashboard.
Business Metrics requirements ➜
1.Total Sales, Total Profit, Average Price per Unit, and Total Units Sold
2.Total sales by month
3.Total sales by state
4.Total sales by region
5.Total sales by product
6.Total sales by retailer
7.Units Sold by Product Category
8.Top Performing Cities by Profit
12:22 PM (1s)
2: Reading the dataset
df=spark.table('workspace.default.adidas_us_sales_datasets')
display(df)
See performance (1)
Optimize
df: pyspark.sql.connect.dataframe.DataFrame = [Retailer: string, Retailer ID: long ... 11 more fields]
Table
Retailer | Retailer ID | Invoice Date | Region | State | City | Product | Price per Unit
1 Foot Locker 1185732 2020-01-01 Northeast New York New York Men's Street Footwear 50
2 Foot Locker 1185732 2020-01-02 Northeast New York New York Men's Athletic Footwear 50
3 Foot Locker 1185732 2020-01-03 Northeast New York New York Women's Street Footwear 40
4 Foot Locker 1185732 2020-01-04 Northeast New York New York Women's Athletic Footwear 45
5 Foot Locker 1185732 2020-01-05 Northeast New York New York Men's Apparel 60
6 Foot Locker 1185732 2020-01-06 Northeast New York New York Women's Apparel 50
7 Foot Locker 1185732 2020-01-07 Northeast New York New York Men's Street Footwear 50
8 Foot Locker 1185732 2020-01-08 Northeast New York New York Men's Athletic Footwear 50
9 Foot Locker 1185732 2020-01-21 Northeast New York New York Women's Street Footwear 40
10 Foot Locker 1185732 2020-01-22 Northeast New York New York Women's Athletic Footwear 45
11 Foot Locker 1185732 2020-01-23 Northeast New York New York Men's Apparel 60
12 Foot Locker 1185732 2020-01-24 Northeast New York New York Women's Apparel 50
13 Foot Locker 1185732 2020-01-25 Northeast New York New York Men's Street Footwear 50
14 Foot Locker 1185732 2020-01-26 Northeast New York New York Men's Athletic Footwear 50
15
9,648 rows | 1.24s runtime
Refreshed 2 hours ago
12:23 PM (<1s)
3: Temporary View table
df.createOrReplaceTempView('sales_analysis')
See performance (1)
Optimize
12:25 PM (1s)
4
%sql select * from sales_analysis
See performance (1)
Optimize
Table
Retailer | Retailer ID | Invoice Date | Region | State | City | Product | Price per Unit
1 Foot Locker 1185732 2020-01-01 Northeast New York New York Men's Street Footwear 50
2 Foot Locker 1185732 2020-01-02 Northeast New York New York Men's Athletic Footwear 50
3 Foot Locker 1185732 2020-01-03 Northeast New York New York Women's Street Footwear 40
4 Foot Locker 1185732 2020-01-04 Northeast New York New York Women's Athletic Footwear 45
5 Foot Locker 1185732 2020-01-05 Northeast New York New York Men's Apparel 60
6 Foot Locker 1185732 2020-01-06 Northeast New York New York Women's Apparel 50
7 Foot Locker 1185732 2020-01-07 Northeast New York New York Men's Street Footwear 50
8 Foot Locker 1185732 2020-01-08 Northeast New York New York Men's Athletic Footwear 50
9 Foot Locker 1185732 2020-01-21 Northeast New York New York Women's Street Footwear 40
10 Foot Locker 1185732 2020-01-22 Northeast New York New York Women's Athletic Footwear 45
11 Foot Locker 1185732 2020-01-23 Northeast New York New York Men's Apparel 60
12 Foot Locker 1185732 2020-01-24 Northeast New York New York Women's Apparel 50
13 Foot Locker 1185732 2020-01-25 Northeast New York New York Men's Street Footwear 50
14 Foot Locker 1185732 2020-01-26 Northeast New York New York Men's Athletic Footwear 50
15
9,648 rows | 1.12s runtime
Refreshed 2 hours ago
This result is stored as _sqldf and can be used in other Python and SQL cells.
12:42 PM (1s)
5: Total Sales, Total Profit, Average prive per unit, Total units sold

```sql
%sql
SELECT
  sum(cast(trim(replace(`Total Sales`, ',', '')) AS DOUBLE)) AS total_sales,
  sum(cast(trim(replace(`Operating Profit`, ',', '')) AS DOUBLE)) AS total_profit,
  avg(cast(trim(replace(`Price per Unit`, ',', '')) AS DOUBLE)) AS av_price_per_unit,
  sum(cast(trim(replace(`Units Sold`, ',', '')) AS DOUBLE)) AS total_unit_sold
FROM sales_analysis
```

See performance (1) Optimize

Table | Visualization 1 +

total_sales	total_profit	av_price_per_unit	total_unit_sold
899902125.00	332135122.00	45.22	2478861.00

Edit Visualization 1 row Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

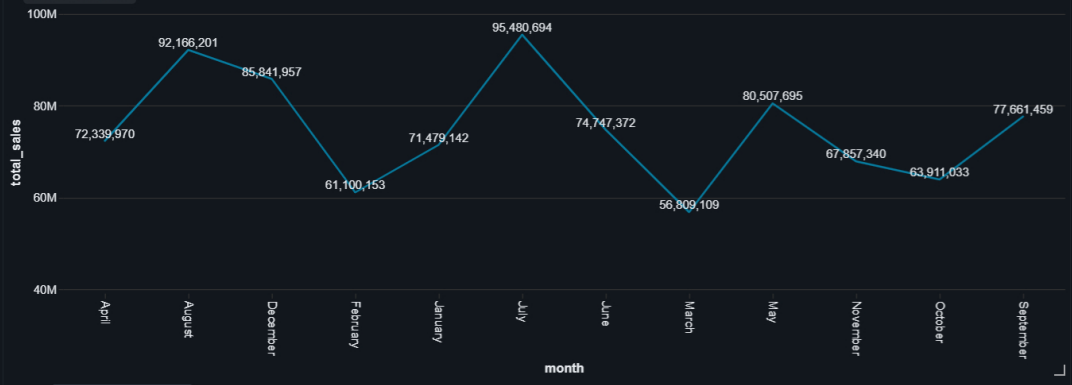
12:47 PM (1s) 6: total sales per month

```sql
%sql
SELECT
  sum(
    cast(
      trim(replace(`total sales`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  date_format(`invoice date`, 'MMMM') AS month
FROM sales_analysis
GROUP BY 2
```

See performance (1) Optimize

Table | Visualization 1 +

New charts: ON



Edit Visualization 12 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

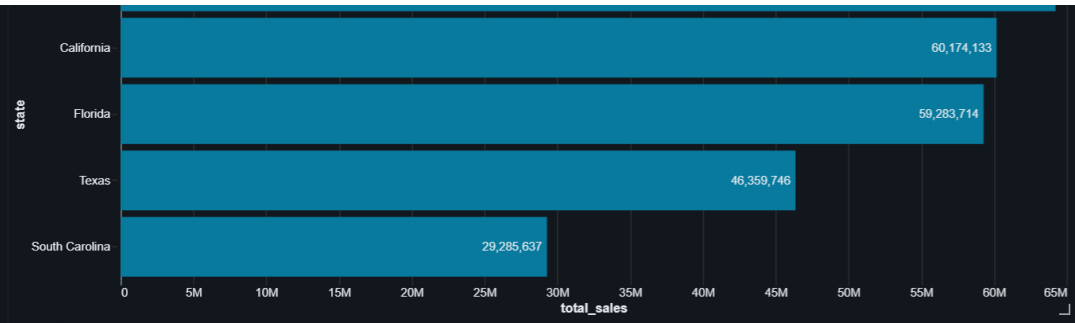
12:51 PM (1s) 7: total sales per state

```sql
%sql
SELECT
  sum(
    cast(
      trim(replace(`total sales`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  state
FROM sales_analysis
GROUP BY 2
order by 1 desc limit 5
```

See performance (1) Optimize

Table | Visualization 1 +

New charts: ON



Edit Visualization 5 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

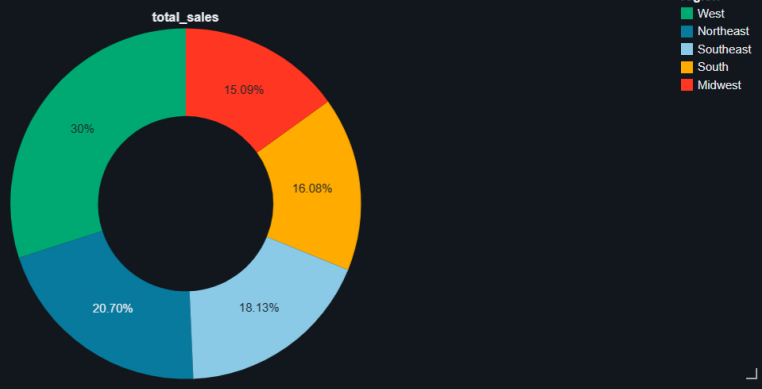
12:54 PM (1s) 8: total sales by region

```
%sql
SELECT
  sum(
    cast(
      trim(replace(`total_sales`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  region
FROM sales_analysis
GROUP BY 2
```

See performance (1) Optimize

Table Visualization 1 +

New charts: ON



Edit Visualization 5 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

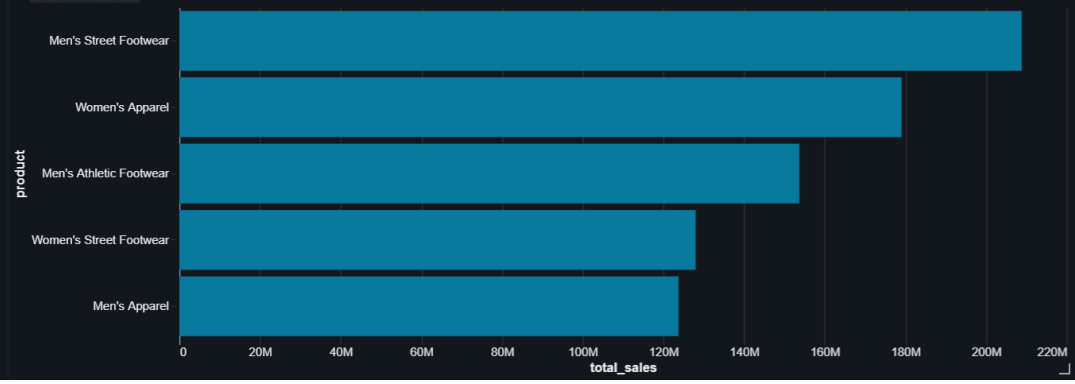
12:57 PM (1s) 9: total sales by product

```
%sql
SELECT
  sum(
    cast(
      trim(replace(`total_sales`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  product
FROM sales_analysis
GROUP BY 2
order by 1 desc limit 5
```

See performance (1) Optimize

Table Visualization 1 +

New charts: ON



Edit Visualization 5 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

12:59 PM (1s) 10: total sales by retailer

```
%sql
SELECT
  sum(
    cast(
      trim(replace(`total_sales`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  retailer
FROM sales_analysis
GROUP BY 2
order by 1 desc limit 5
```

See performance (1) Optimize

Table Visualization 1 +

Steps	Value	% Max	% Previous
West Gear	242,964,333	100%	100%
Foot Locker	220,094,720	90.59%	90.59%
Sports Direct	182,470,997	75.10%	82.91%
Kohl's	102,114,753	42.03%	55.96%
Amazon	77,698,912	31.98%	76.09%

Edit Visualization 5 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

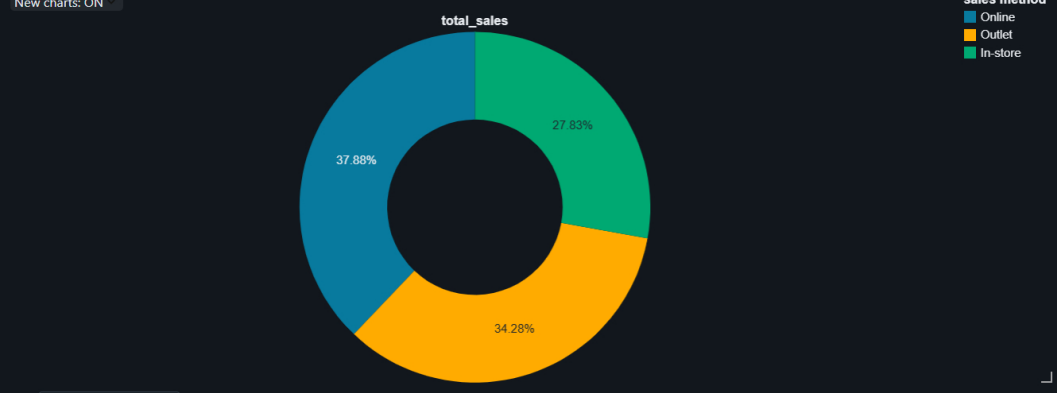
01:02 PM (1s) 11: units sold by category

```
%sql
SELECT
  sum(
    cast(
      trim(replace(`units sold`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
  `sales method`
FROM sales_analysis
GROUP BY 2
order by 1 desc limit 5
```

See performance (1) Optimize

Table Visualization 1 +

New charts: ON



Edit Visualization 3 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

01:04 PM (1s) 12: top performing cities by profit

```
%sql
SELECT
  sum(
    cast(
      trim(replace(`operating profit`, ',', '')) AS DOUBLE
    )
  ) AS total_sales,
```

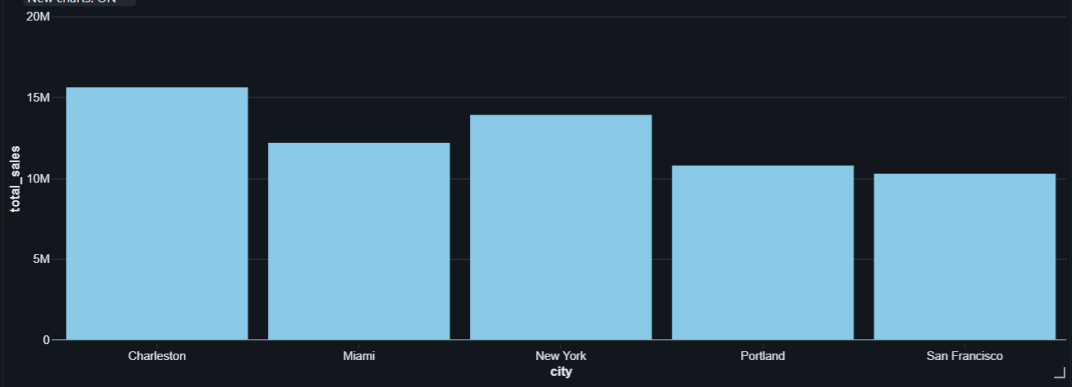
```
  city
FROM sales_analysis
GROUP BY 2
order by 1 desc limit 5
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [total_sales: double, city: string]

Table | Visualization 1 | +

New charts: ON



Edit Visualization 5 rows Refreshed 2 hours ago

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

13

Conclusion: The analysis of Adidas sales data for 2020 and 2021 provided valuable insights into sales performance across different time periods, regions, products, and customer segments. By evaluating metrics such as total sales, profit, average price per unit, and units sold, we identified the key factors driving revenue and profitability.

Monthly and regional trends revealed seasonal demand patterns and high-performing markets, while product and retailer analysis highlighted the most profitable categories and sales channels. Additionally, segmentation by product category and gender type helped uncover customer preferences.

Overall, the study uncovered opportunities for growth by identifying top-performing cities and profitable regions, which can be leveraged for strategic decision-making. The use of Databricks Notebook allowed for an interactive dashboard that can assist business leaders in monitoring sales performance and optimizing future strategies.

[Shift+Enter] to run and move to next cell
[Ctrl+Shift+P] to open the command palette
[Esc H] to see all keyboard shortcuts