

PROJECT 2

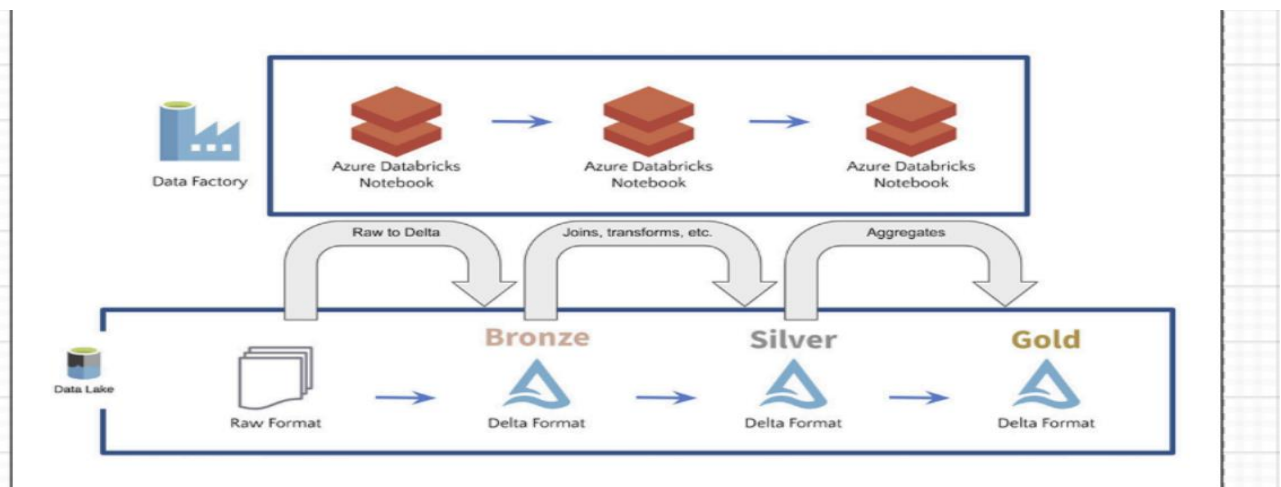
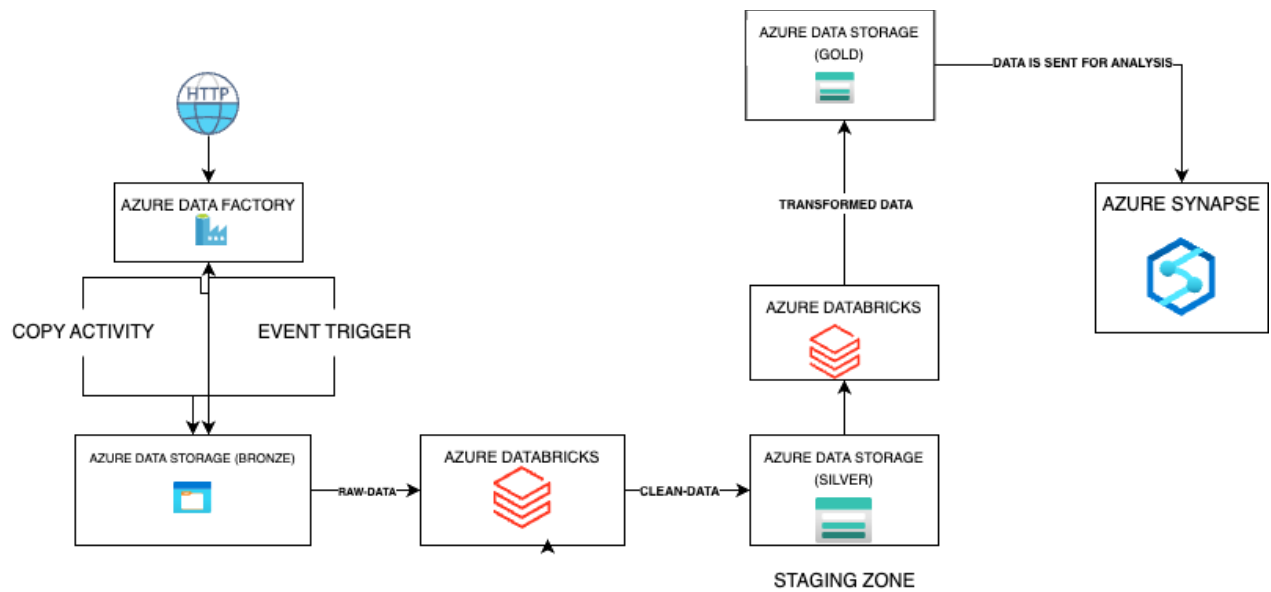
Customer Shopping Trends

The Customer Shopping Preferences Dataset offers valuable insights into consumer behavior and purchasing patterns. Understanding customer preferences and trends is critical for businesses to tailor their products, marketing strategies, and overall customer experience. This dataset captures a wide range of customer attributes including age, gender, purchase history, preferred payment methods, frequency of purchases, and more.

SCHEMA:

- **Customer ID** - Unique identifier for each customer
- **Age** - Age of the customer
- **Gender** - Gender of the customer (Male/Female)
- **Item Purchased** - The item purchased by the customer
- **Category** - Category of the item purchased
- **Purchase Amount (USD)** - The amount of the purchase in USD
- **Location** - Location where the purchase was made
- **Size** - Size of the purchased item
- **Color** - Color of the purchased item
- **Season** - Season during which the purchase was made
- **Review Rating** - Rating given by the customer for the purchased item
- **Subscription Status** - Indicates if the customer has a subscription (Yes/No)
- **Shipping Type** - Type of shipping chosen by the customer
- **Discount Applied** - Indicates if a discount was applied to the purchase (Yes/No)
- **Promo Code Used** - Indicates if a promo code was used for the purchase (Yes/No)
- **Previous Purchases** - The total count of transactions concluded by the customer at the store, excluding the ongoing transaction
- **Payment Method** - Customer's most preferred payment method
- **Frequency of Purchases** - Frequency at which the customer makes purchases (e.g., Weekly, Fortnightly, Monthly)

Architecture Diagram:



RAW-DATA

CREATED STORAGE CONTAINERS

USED COPY ACTIVITY WITH EVENT TRIGGER TO COPY DATA FROM HTTPS TO STORAGE ACCOUNT:

Microsoft Azure | Data Factory | datafacretail

Search factory and documentation

Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click [here](#) to get started with Fabric Data Factory!

» Data Factory | Validate all | Publish all

Preview experience: Off

Factory Resources

Filter resources by name

- Pipelines 1
 - pipeline
- Change Data Capture (preview) 0
- Datasets 2
 - Data flows 0
 - Power Query 0

Activities

Search activities

- Move and transform
- Synapse
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

Copy data

Copy data1

Parameters Variables Settings Output

Pipeline run ID: 18bc2cfb-0871-4fe8-bf8d-a2d1a203cfd Pipeline status: Succeeded View debug run consumption

All status

Showing 1 - 1 of 1 items

| Activity name | Activity status | Activity type | Run start | Duration | Integration runtime | User properties | Ac |
|---------------|-----------------|---------------|-----------------------|----------|------------------------|-----------------|----|
| Copy data1 | Succeeded | Copy data | 8/8/2024, 10:39:51 AM | 11s | AutoResolveIntegration | | ds |

Microsoft Azure | Data Factory | datafacretail

Search factory and documentation

Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click [here](#) to get started with Fabric Data Factory!

» Data Factory | Validate all | Publish all

Preview experience: Off

Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

+ New Refresh

Filter by name Annotations: Any

Showing 1 - 1 of 1 items

| Name | Type | Status | Related | Annotations |
|----------|----------------|---------|---------|-------------|
| trigger1 | Storage events | Started | 1 | |

The screenshot shows the Microsoft Azure portal interface. On the left, the 'Overview' tab for a container named 'bronze' is selected. The 'Authentication method' is set to 'Access key'. The 'Location' is 'bronze / ruthvikaa / Project-2 / main'. A search bar for blobs by prefix is visible. In the center, a file named 'shopping_trends_updated.csv' is listed. On the right, the 'Properties' tab for this file is open, showing details such as URL, last modified time (8/8/2024, 10:45:16 AM), creation time (8/8/2024, 10:45:16 AM), version ID, type (Block blob), size (406.85 KiB), access tier (Hot), and other metadata.

CREATED MICROSOFT ENTRAA ID :

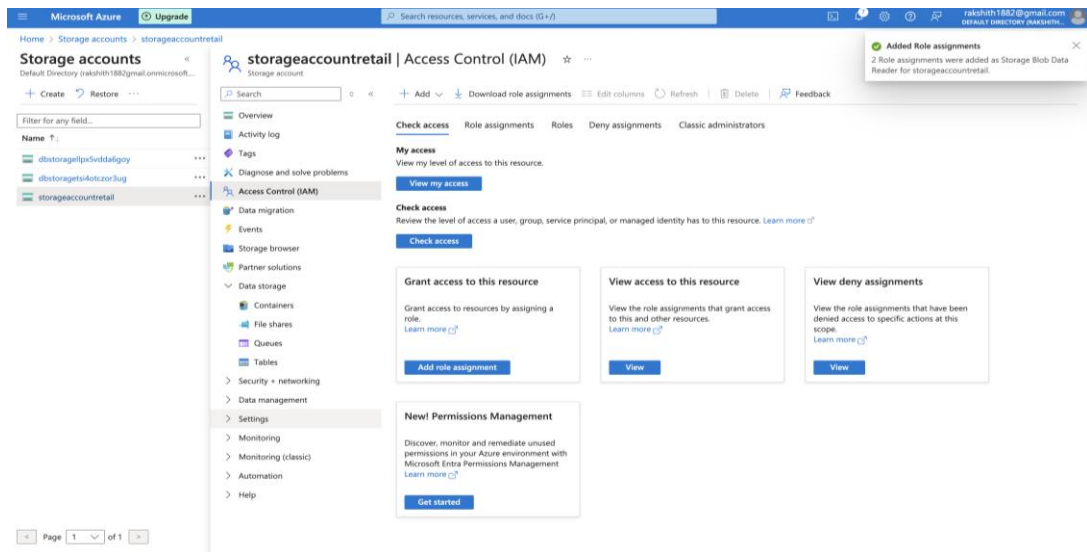
The screenshot displays the 'Overview' page for an application registration in the Microsoft Azure portal. The 'Essentials' section contains the following information:

- Display name: app
- Application (client) ID: 6aaa794d-6620-4b18-9d42-a1622baa9505
- Object ID: e9d78843-839a-4d88-bbb1-f85ccc
- Directory (tenant) ID: 76d20279-4371-4507-aab4-d514dc931793
- Supported account types: My organization only
- Client credentials: 0 certificate, 1 secret
- Redirect URIs: Add a Redirect URI
- Application ID URI: Add an Application ID URI
- Managed application in L: app

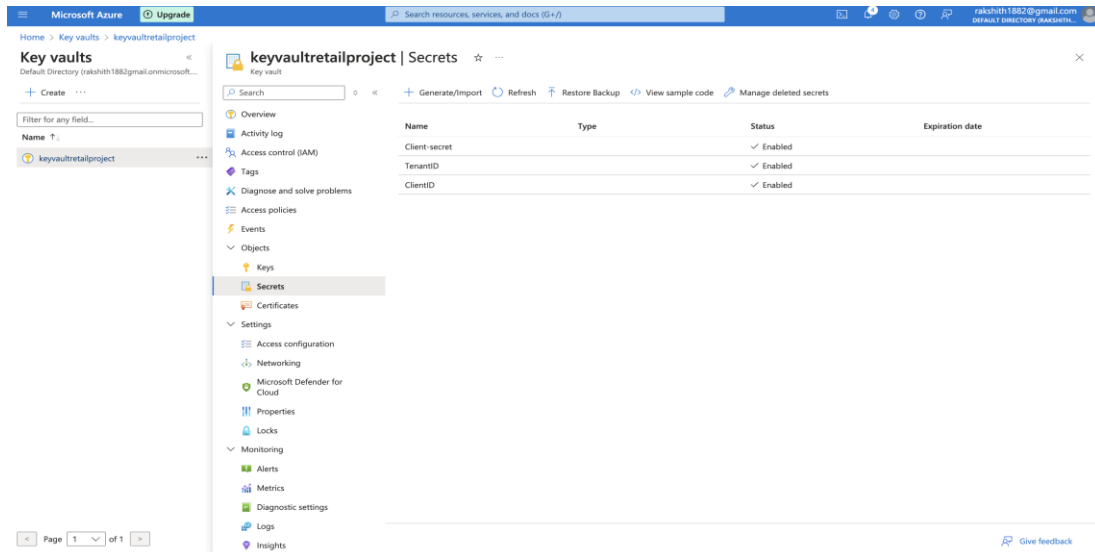
Below the essentials, there is a section titled 'Build your application with the Microsoft identity platform'. It includes a brief description of the platform and three main actions:

- Call APIs:** Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources. [View API permissions](#)
- Sign in users in 5 minutes:** Use our SDKs to sign in users and call APIs in a few steps. Use the quickstarts to start a web app, mobile app, SPA, or daemon app. [View all quickstart guides](#)
- Configure for your organization:** Assign users and groups, apply conditional access policies, configure single sign-on, and more in Enterprise applications. [Go to Enterprise applications](#)

GIVEN REQUIRED ACCESS PERMISSIONS:



CREATED KEYVAULT AND GENERATED SECRETS:



CONFIGURATION SETUP IN DATABRICKS:

The screenshot shows a Databricks notebook with five code blocks. Block 1 sets up secrets for client ID, tenant ID, and client secret. Block 2 defines OAuth2 configurations. Block 3 sets up Spark configurations for storage account authentication. Block 4 unmounts a storage account. Block 5 shows the unmount operation completed.

```
1
client_id = dbutils.secrets.get(scope="secretscope2",key="ClientID")
tenant_id = dbutils.secrets.get(scope="secretscope2",key="TenantID")
client_secret = dbutils.secrets.get(scope="secretscope2",key="Client-secret")

2
configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": client_id,
           "fs.azure.account.oauth2.client.secret": client_secret,
           "fs.azure.account.oauth2.client.endpoint": f"https://login.microsoftonline.com/{tenant_id}/oauth2/token"}

3
spark.conf.set("fs.azure.account.auth.type.storageaccountretail.dfs.core.windows.net", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type.storageaccountretail.dfs.core.windows.net", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.storageaccountretail.dfs.core.windows.net", client_id)
spark.conf.set("fs.azure.account.oauth2.client.secret.storageaccountretail.dfs.core.windows.net", client_secret)
spark.conf.set("fs.azure.account.oauth2.client.endpoint.storageaccountretail.dfs.core.windows.net", f"https://login.microsoftonline.com/{tenant_id}/oauth2/token")

4
dbutils.fs.unmount("/mnt/storageaccountretail/bronze")

/mnt/storageaccountretail/bronze has been unmounted.
True

5
```

DEFINED SCHEMA

The screenshot shows a Databricks notebook with three code blocks. Block 7 imports SparkSession and StructType. Block 8 defines a schema with 16 fields. Block 9 reads a CSV file into a DataFrame.

```
7
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DoubleType, FloatType, DateType

8
schema = StructType([
    StructField("Customer ID", IntegerType(), False),
    StructField("Age", IntegerType(), False),
    StructField("Gender", StringType(), False),
    StructField("Item Purchased", StringType(), False),
    StructField("Category", StringType(), False),
    StructField("Purchased Amount (USD)", FloatType(), False),
    StructField("Location", StringType(), False),
    StructField("Size", StringType(), False),
    StructField("Color", StringType(), False),
    StructField("Season", StringType(), False),
    StructField("Review Rating", FloatType(), False),
    StructField("Subscription Status", StringType(), False),
    StructField("Shipping type", StringType(), False),
    StructField("Discount Applied", StringType(), False),
    StructField("Promo Code Used", StringType(), False),
    StructField("Previous Purchases", IntegerType(), False),
    StructField("Payment Method", StringType(), False),
    StructField("Frequency of Purchases", StringType(), False),
])

9
df = spark.read.option("Header", True).schema(schema).csv("/mnt/storageaccountretail/bronze/ruthvikaa/Project-2/main/shopping_trends_updated.csv")

df: pyspark.sql.dataframe.DataFrame = [Customer ID: integer, Age: integer ... 16 more fields]
```

DATA AGGREGATIONS:

Microsoft Azure | databricks

raw-processed Python

File Edit View Run Help Last edit was 1 minute ago Provide feedback

Notebook detached cluster not in usable state

```
df_clean.write.mode("overwrite").parquet("/mnt/storageaccountretail/silver")
```

16

18 hours ago (8)

1 Spark Jobs

```
raw_folder_path = "/mnt/storageaccountretail/bronze"
processed_folder_path = "/mnt/storageaccountretail/silver"
presented_folder_path = "/mnt/storageaccountretail/gold"
```

17

18 hours ago (4)

```
df_clean = spark.read.parquet(f"{processed_folder_path}")

# Show the contents of the DataFrame
df_clean.show()
```

18

2 Spark Jobs

18 Bank Transfer|Fortnightly|Coat|Outerwear|72.0|Delaware|Yes|Winter|Express|

37 Verme|Fortnightly|Coat|Outerwear|51.0|New Hampshire|Yes|Spring|Express|

31 PayPal|Weekly|Coat|Outerwear|53.0|New York|Yes|Winter|Free Shipping|

34 Debit Card|Weekly|Skirt|Clothing|81.0|Rhode Island|Yes|Winter|Store Pickup|

44 Debit Card|Bi-Weekly|Sunglasses|Accessories|36.0|Alabama|Yes|Spring|Next Day Air|

36 Verme|Quarterly|Dress|Clothing|38.0|Mississippi|Yes|Winter|2-Day Shipping|

17 Verme|Sweater|Clothing|48.0|Montana|Yes|Summer|Free Shipping|

46 Debit Card|Bi-Weekly|Pants|Clothing|98.0|Rhode Island|Yes|Summer|Standard|

Microsoft Azure | databricks

raw-processed Python

File Edit View Run Help Last edit was 2 minutes ago Provide feedback

Notebook detached cluster not in usable state

```
Groupby_df=df_clean.groupby(["Payment_method","Category","Frequency_of_Purchases"]).count()
```

25

Groupby_df: pyspark.sql.dataframe.DataFrame

Payment_method: string

Category: string

Frequency_of_Purchases: string

count: long

```
display(Groupby_df)
```

26

2 Spark Jobs

| Payment_method | Category | Frequency_of_Purchases | count |
|----------------|-------------|------------------------|-------|
| Bank Transfer | Accessories | Fortnightly | 20 |
| Debit Card | Footwear | Every 3 Months | 18 |
| Debit Card | Outerwear | Quarterly | 5 |
| Cash | Clothing | Weekly | 46 |
| Cash | Accessories | Weekly | 32 |
| Debit Card | Accessories | Annually | 29 |
| Cash | Outerwear | Every 3 Months | 7 |
| Bank Transfer | Outerwear | Fortnightly | 11 |
| Bank Transfer | Outerwear | Bi-Weekly | 6 |
| Credit Card | Footwear | Monthly | 6 |
| Debit Card | Footwear | Monthly | 14 |
| Verme | Accessories | Every 3 Months | 29 |
| Verme | Footwear | Monthly | 18 |

DATA AGGREGATIONS:

Microsoft Azure databricks

raw-processed Python ☆ Last edit was 2 minutes ago Provide feedback

Notebook detached
cluster not in usable state

```

18 hours ago (c/c) 21
male_df_clean = df_clean_filter.filter("Gender= 'Male'")
male_df_clean: pyspark.sql.dataframe.DataFrame = [Customer_ID: integer, Age: integer ... 11 more fields]

18 hours ago (c/c) 22
from pyspark.sql.functions import count
male_df_clean.select(count("*")).show()
(2) Spark Jobs
[count(1)]
1
1893]

18 hours ago (c/c) 23
female_df_clean = df_clean_filter.filter("Gender= 'Female'")
female_df_clean: pyspark.sql.dataframe.DataFrame = [Customer_ID: integer, Age: integer ... 11 more fields]

18 hours ago (c/c) 24
female_df_clean.select(count("*")).show()
(2) Spark Jobs
[count(1)]
1
589]

```

Microsoft Azure databricks

raw-processed Python ☆ Last edit was 3 minutes ago Provide feedback

Notebook detached
cluster not in usable state

18 hours ago (c/c) 27

```

from pyspark.sql.functions import asc
Final_Groupby_df = df_clean.groupby("Age", "Payment_method", "Category", "Frequency_of_Purchases", "Gender") \
    .count() \
    .orderBy(asc("Age"))
Final_Groupby_df: pyspark.sql.dataframe.DataFrame
Age: integer
Payment_method: string
Category: string
Frequency_of_Purchases: string
Gender: string
count: long

```

18 hours ago (c/c) 28

```

display(Final_Groupby_df)
(2) Spark Jobs

```

| | Age | Payment_method | Category | Frequency_of_Purchases | Gender | count |
|---|-----|----------------|-------------|------------------------|--------|-------|
| 1 | 18 | Bank Transfer | Outerwear | Bi-Weekly | Male | 1 |
| 2 | 18 | Cash | Accessories | Quarterly | Female | 1 |
| 3 | 18 | Venmo | Clothing | Monthly | Male | 1 |
| 4 | 18 | PayPal | Clothing | Bi-Weekly | Male | 1 |
| 5 | 18 | Credit Card | Clothing | Every 3 Months | Female | 1 |
| 6 | 18 | Credit Card | Clothing | Annually | Male | 1 |
| 7 | 18 | Bank Transfer | Clothing | Every 3 Months | Male | 3 |
| 8 | 18 | Venmo | Clothing | Bi-Weekly | Male | 1 |

CREATED SYNAPSE SERVERLESS DB AND CREATED EXTERNAL TABLE:

Microsoft Azure

Synapse Analytics

synapseworkspace-retail

Synapse live

Validate all

Publish all

Search

Workspace

Linked

Filter resources by name

SQL database

serverlessql (SQL)

External tables

dbo.finaldata

Customer ID (nvarchar...

Age (int, null)

Gender (nvarchar(400...

Item Purchased (nvarchar...

Category (nvarchar(40...

Purchase Amount (US...

Location (nvarchar(40...

Subscription Status (n...

Season (nvarchar(4000...

Shipping Type (nvarchar...

Payment Method (nva...

Frequency of Purchase...

dbo.retaildata

External resources

Views

Schemas

Security

sqlpool0db (SQL)

SQL script 2

Run

Undo

Publish

Query plan

Connect to

Built-in

Use database

serverlessql

Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

```
7
8
9 IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name = 'silver_storageaccountretail_dfs_core_windows_net')
10 CREATE EXTERNAL DATA SOURCE [silver_storageaccountretail_dfs_core_windows_net]
11 WITH (
12     LOCATION = 'abfss://silver@storageaccountretail.dfs.core.windows.net'
13 )
14 GO
15
16 CREATE EXTERNAL TABLE dbo.finaldata (
17     [Customer ID] nvarchar(4000),
18     [Age] INT,
19     [Gender] nvarchar(4000),
20     [Item Purchased] nvarchar(4000),
21     [Category] nvarchar(4000),
22     [Purchase Amount (USD)] INT
23 )
```

Results

Messages

View

Table

Chart

Export results

Search

| Customer ID | Age | Gender | Item Purchased | Category | Purchase Amo... | Location | Size |
|-------------|-----|--------|----------------|----------|-----------------|---------------|------|
| Customer ID | Age | Gender | Item Purchased | Category | Purchase Amo... | Location | Size |
| 1 | 55 | Male | Blouse | Clothing | 53 | Kentucky | L |
| 2 | 19 | Male | Sweater | Clothing | 64 | Maine | L |
| 3 | 50 | Male | Jeans | Clothing | 73 | Massachusetts | S |
| 4 | 21 | Male | Sandals | Footwear | 90 | Rhode Island | M |
| 5 | 45 | Male | Blouse | Clothing | 49 | Oregon | M |
| 6 | 46 | Male | Sneakers | Footwear | 20 | Wyoming | M |
| 7 | 63 | Male | Shirt | Clothing | 85 | Montana | M |
| 8 | 27 | Male | Shorts | Clothing | 34 | Louisiana | L |

00:00:03 Query executed successfully.

Properties

General

Related (0)

Name

SQL script 2

Description

Type

.sql script

Size

1,242 bytes

Results settings per query

First 5000 rows (default)

All rows