```
# Import libraries
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import davies_bouldin_score
import matplotlib.pyplot as plt
import seaborn as sns

# Load datasets
customers = pd.read_csv("Data/Customers.csv")
transactions = pd.read_csv("Data/Transactions.csv")

# Merge datasets
customer_transactions = transactions.groupby('CustomerID').agg({
    'TotalValue': 'sum', 'Quantity': 'sum'
}).reset_index()
customer_profiles = customers.merge(customer_transactions, on='CustomerID',
how='left').fillna(0)

# Features for clustering
features = customer_profiles[['TotalValue', 'Quantity']]

# Perform clustering
inertia = []
db_scores = []
for k in range(2, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(features)
    inertia.append(kmeans.inertia_)
    db_scores.append(davies_bouldin_score(features, labels))

# Choose optimal number of clusters
optimal_k = db_scores.index(min(db_scores)) + 2
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
customer_profiles['Cluster'] = kmeans.fit_predict(features)

# Visualize clusters
sns.scatterplot(
    x=customer_profiles['TotalValue'],
    y=customer_profiles['Quantity'],
    hue=customer_profiles['Cluster'],
    palette='tab10'
)
plt.title('Customer Segmentation')
plt.show()

# Save clustering metrics
clustering_report = {
    "Optimal Clusters": optimal_k,
    "DB Index": min(db_scores),
}
print("Clustering Metrics:", clustering_report)
```