**Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown, and Pivot**  
import pandas as pd

import numpy as np

from datetime import datetime

class OLAPOperations:

def \_\_init\_\_(self):

self.sales\_data = self.generate\_sales\_data()

def generate\_sales\_data(self):

products = ['Laptop', 'Phone', 'Tablet', 'Monitor', 'Keyboard']

regions = ['North', 'South', 'East', 'West']

categories = ['Electronics', 'Accessories']

data = []

for i in range(1000):

record = {

'transaction\_id': i + 1,

'date': pd.to\_datetime('2024-01-01') + pd.Timedelta(days=np.random.randint(0, 365)),

'product': np.random.choice(products),

'category': np.random.choice(categories),

'region': np.random.choice(regions),

'quantity': np.random.randint(1, 10),

'unit\_price': np.random.uniform(50, 2000),

'salesperson': f'SP{np.random.randint(1, 11):02d}'

}

record['total\_sales'] = record['quantity'] \* record['unit\_price']

data.append(record)

return pd.DataFrame(data)

def slice\_operation(self):

print("=== SLICE OPERATION ===")

slice\_result = self.sales\_data[

(self.sales\_data['product'] == 'Laptop') &

(self.sales\_data['region'] == 'North')

]

print(f"Sliced data: Laptops in North region")

print(f"Records found: {len(slice\_result)}")

print(slice\_result[['transaction\_id', 'product', 'region', 'total\_sales']].head())

return slice\_result

def dice\_operation(self):

print("\n=== DICE OPERATION ===")

dice\_result = self.sales\_data[

((self.sales\_data['product'].isin(['Phone', 'Tablet'])) &

(self.sales\_data['region'].isin(['East', 'West'])) &

(self.sales\_data['total\_sales'] > 1000))

]

print("Diced data: Phones/Tablets in East/West with sales > $1000")

print(f"Records found: {len(dice\_result)}")

print(dice\_result[['transaction\_id', 'product', 'region', 'total\_sales']].head())

return dice\_result

def rollup\_operation(self):

print("\n=== ROLLUP OPERATION ===")

self.sales\_data['year'] = self.sales\_data['date'].dt.year

self.sales\_data['quarter'] = self.sales\_data['date'].dt.quarter

region\_rollup = self.sales\_data.groupby('region')['total\_sales'].agg(['sum', 'mean', 'count']).round(2)

print("Rollup: Sales by Region")

print(region\_rollup)

time\_rollup = self.sales\_data.groupby(['year', 'quarter'])['total\_sales'].sum().round(2)

print("\nRollup: Sales by Year-Quarter")

print(time\_rollup)

return region\_rollup, time\_rollup

def drilldown\_operation(self):

print("\n=== DRILLDOWN OPERATION ===")

yearly\_sales = self.sales\_data.groupby('year')['total\_sales'].sum().round(2)

print("Yearly Sales:")

print(yearly\_sales)

quarterly\_sales = self.sales\_data.groupby(['year', 'quarter'])['total\_sales'].sum().round(2)

print("\nQuarterly Sales (Drilldown from Year):")

print(quarterly\_sales)

monthly\_sales = self.sales\_data.groupby([

self.sales\_data['date'].dt.year,

self.sales\_data['date'].dt.month

])['total\_sales'].sum().round(2)

print("\nMonthly Sales (Further Drilldown):")

print(monthly\_sales.head(12))

return yearly\_sales, quarterly\_sales, monthly\_sales

def pivot\_operation(self):

print("\n=== PIVOT OPERATION ===")

pivot\_table = pd.pivot\_table(

self.sales\_data,

values='total\_sales',

index='product',

columns='region',

aggfunc='sum',

fill\_value=0

).round(2)

print("Pivot: Total Sales by Product and Region")

print(pivot\_table)

pivot\_table\_count = pd.pivot\_table(

self.sales\_data,

values='transaction\_id',

index='category',

columns='region',

aggfunc='count',

fill\_value=0

)

print("\nPivot: Transaction Count by Category and Region")

print(pivot\_table\_count)

return pivot\_table, pivot\_table\_count

def execute\_all\_operations(self):

print("EXECUTING ALL OLAP OPERATIONS")

print("=" \* 50)

slice\_result = self.slice\_operation()

dice\_result = self.dice\_operation()

rollup\_result = self.rollup\_operation()

drilldown\_result = self.drilldown\_operation()

pivot\_result = self.pivot\_operation()

return {

'slice': slice\_result,

'dice': dice\_result,

'rollup': rollup\_result,

'drilldown': drilldown\_result,

'pivot': pivot\_result

}

olap = OLAPOperations()

results = olap.execute\_all\_operations()

print("\n" + "=" \* 50)

print("SUMMARY STATISTICS")

print(f"Total transactions: {len(olap.sales\_data)}")

print(f"Total sales: ${olap.sales\_data['total\_sales'].sum():,.2f}")

print(f"Average transaction: ${olap.sales\_data['total\_sales'].mean():,.2f}")

print(f"Unique products: {olap.sales\_data['product'].nunique()}")

print(f"Date range: {olap.sales\_data['date'].min().date()} to {olap.sales\_data['date'].max().date()}")