

```
pip install cassandra-driver
```

```
Collecting cassandra-driver
  Downloading cassandra_driver-3.29.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.2 kB)
Collecting geomet<0.3,>=0.1 (from cassandra-driver)
  Downloading geomet-0.2.1.post1-py3-none-any.whl.metadata (1.0 kB)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from geomet<0.3,>=0.1->cassandra-driver) (8.1.7)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from geomet<0.3,>=0.1->cassandra-driver) (1.16.0)
Downloading cassandra_driver-3.29.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.9 MB)
  3.9/3.9 MB 34.3 MB/s eta 0:00:00
Downloading geomet-0.2.1.post1-py3-none-any.whl (18 kB)
Installing collected packages: geomet, cassandra-driver
Successfully installed cassandra-driver-3.29.2 geomet-0.2.1.post1
```

```
pip install --upgrade astropy
```

```
Collecting astropy
  Downloading astropy-1.5.2-py3-none-any.whl.metadata (19 kB)
Collecting deprecation<2.2.0,>=2.1.0 (from astropy)
  Downloading deprecation-2.1.0-py2.py3-none-any.whl.metadata (4.6 kB)
Requirement already satisfied: httpx<1,>=0.25.2 in /usr/local/lib/python3.10/dist-packages (from httpx[http2]<1,>=0.25.2->astropy) (0.27)
Collecting pymongo>=3 (from astropy)
  Downloading pymongo-4.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (22 kB)
Requirement already satisfied: toml<0.11.0,>=0.10.2 in /usr/local/lib/python3.10/dist-packages (from astropy) (0.10.2)
Collecting uuid6>=2024.1.12 (from astropy)
  Downloading uuid6-2024.7.10-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from deprecation<2.2.0,>=2.1.0->astropy) (24.2)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Collecting h2<5,>=3 (from httpx[http2]<1,>=0.25.2->astropy)
  Downloading h2-4.1.0-py3-none-any.whl.metadata (3.6 kB)
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo>=3->astropy)
  Downloading dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)
Collecting hyperframe<7,>=6.0 (from h2<5,>=3->httpx[http2]<1,>=0.25.2->astropy)
  Downloading hyperframe-6.0.1-py3-none-any.whl.metadata (2.7 kB)
Collecting hpack<5,>=4.0 (from h2<5,>=3->httpx[http2]<1,>=0.25.2->astropy)
  Downloading hpack-4.0.0-py3-none-any.whl.metadata (2.5 kB)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx<1,>=0.25.2->httpx[http2]<1,>=0.25.2->astropy)
Downloading astropy-1.5.2-py3-none-any.whl (177 kB)
  177.1/177.1 kB 5.5 MB/s eta 0:00:00
Downloading deprecation-2.1.0-py2.py3-none-any.whl (11 kB)
Downloading pymongo-4.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.4 MB)
  1.4/1.4 MB 31.8 MB/s eta 0:00:00
Downloading uuid6-2024.7.10-py3-none-any.whl (6.4 kB)
Downloading dnspython-2.7.0-py3-none-any.whl (313 kB)
  313.6/313.6 kB 27.3 MB/s eta 0:00:00
Downloading h2-4.1.0-py3-none-any.whl (57 kB)
  57.5/57.5 kB 5.4 MB/s eta 0:00:00
Downloading hpack-4.0.0-py3-none-any.whl (32 kB)
Downloading hyperframe-6.0.1-py3-none-any.whl (12 kB)
Installing collected packages: uuid6, hyperframe, hpack, dnspython, deprecation, pymongo, h2, astropy
Successfully installed astropy-1.5.2 deprecation-2.1.0 dnspython-2.7.0 h2-4.1.0 hpack-4.0.0 hyperframe-6.0.1 pymongo-4.10.1 uuid6-2024.7
```

```
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider
import json
```

```

cloud_config = {
    'secure_connect_bundle': 'secure-connect-bigdata.zip'
}
with open("bigdata-token.json") as f:
    secrets = json.load(f)
CLIENT_ID = secrets["clientId"]
CLIENT_SECRET = secrets["secret"]
auth_provider = PlainTextAuthProvider(CLIENT_ID, CLIENT_SECRET)
cluster = Cluster(cloud=cloud_config, auth_provider=auth_provider)
session = cluster.connect()
if session:
    print("Session is active")
else:
    print("Session is not active")

```

→ WARNING:cassandra.cluster:Downgrading core protocol version from 66 to 65 for bf6a839c-970f-468d-9f9d-c3023a782d48-eu-west-1.db.astra.dat
 WARNING:cassandra.cluster:Downgrading core protocol version from 65 to 5 for bf6a839c-970f-468d-9f9d-c3023a782d48-eu-west-1.db.astra.dat
 WARNING:cassandra.cluster:Downgrading core protocol version from 5 to 4 for bf6a839c-970f-468d-9f9d-c3023a782d48-eu-west-1.db.astra.dat
 Session is active

```

create_table_query = """
CREATE TABLE sales.sales_data (
    Region TEXT,
    Country TEXT,
    Item_Type TEXT,
    Sales_Channel TEXT,
    Order_Priority TEXT,
    Order_Date TEXT,
    Order_ID BIGINT PRIMARY KEY,
    Ship_Date TEXT,
    Units_Sold INT,
    Unit_Price FLOAT,
    Unit_Cost FLOAT,
    Total_Revenue FLOAT,
    Total_Cost FLOAT,
    Total_Profit FLOAT
);
"""

session.execute(create_table_query)
print("Table created successfully!")

```

→ Table created successfully!

```

import pandas as pd
from cassandra.cluster import Cluster

csv_file_path = "sales_100.csv"
df = pd.read_csv(csv_file_path)

for _, row in df.iterrows():
    query = """
    INSERT INTO sales.sales_data (Region, Country, Item_Type, Sales_Channel, Order_Priority, Order_Date, Order_ID, Ship_Date, Units_Sold, Unit_Price, Unit_Cost, Total_Revenue, Total_Cost, Total_Profit)
    VALUES (%s, %s, %s)
    """
    session.execute(query, tuple(row))

print("Data imported successfully!")

```

→ Data imported successfully!

```

query = "SELECT * FROM sales.sales_data ;"
rows = session.execute(query)

```

```

for row in rows:
    print(row)

```

→ Row(order_id=294530856, country='Italy', item_type='Cereal', order_date='11/15/2011', order_priority='M', region='Europe', sales_chan
 Row(order_id=274930989, country='Dominica', item_type='Household', order_date='11/19/2011', order_priority='C', region='Central Ameri
 Row(order_id=498071897, country='Taiwan', item_type='Cereal', order_date='4/11/2010', order_priority='H', region='Asia', sales_channe
 Row(order_id=940980136, country='New Zealand', item_type='Beverages', order_date='10/11/2012', order_priority='M', region='Australia
 Row(order_id=324669444, country='France', item_type='Cosmetics', order_date='12/7/2015', order_priority='M', region='Europe', sales_c
 Row(order_id=830410039, country='Switzerland', item_type='Office Supplies', order_date='7/29/2014', order_priority='C', region='Europ
 Row(order_id=953361213, country='Zimbabwe', item_type='Office Supplies', order_date='3/28/2011', order_priority='C', region='Sub-Saha
 Row(order_id=425418365, country='United Arab Emirates', item_type='Office Supplies', order_date='6/23/2012', order_priority='M', regi

```

Row(order_id=349235904, country='Mauritius ', item_type='Clothes', order_date='11/17/2012', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=246248090, country='The Bahamas', item_type='Personal Care', order_date='1/19/2011', order_priority='C', region='Central America')
Row(order_id=368977391, country='Canada', item_type='Cosmetics', order_date='5/9/2011', order_priority='H', region='North America', sales_channel='Retail')
Row(order_id=572335612, country='Vanuatu', item_type='Cereal', order_date='6/20/2014', order_priority='C', region='Australia and Oceania')
Row(order_id=443368995, country='South Africa', item_type='Fruits', order_date='7/27/2012', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=156295812, country='East Timor', item_type='Cereal', order_date='12/7/2015', order_priority='M', region='Australia and Oceania')
Row(order_id=807785928, country='Ethiopia', item_type='Cosmetics', order_date='7/7/2011', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=678896129, country='Indonesia', item_type='Personal Care', order_date='10/22/2012', order_priority='L', region='Asia', sales_channel='Retail')
Row(order_id=520480573, country='Indonesia', item_type='Household', order_date='9/28/2011', order_priority='C', region='Asia', sales_channel='Retail')
Row(order_id=925136649, country='Serbia', item_type='Clothes', order_date='7/6/2016', order_priority='L', region='Europe', sales_channel='Retail')
Row(order_id=222504317, country='Belgium', item_type='Personal Care', order_date='11/1/2011', order_priority='H', region='Europe', sales_channel='Retail')
Row(order_id=563681733, country='Togo', item_type='Cosmetics', order_date='9/8/2015', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=488121116, country='The Bahamas', item_type='Fruits', order_date='6/18/2010', order_priority='L', region='Central America')
Row(order_id=839094388, country='Tonga', item_type='Baby Food', order_date='5/11/2016', order_priority='L', region='Australia and Oceania')
Row(order_id=500371730, country='Pakistan', item_type='Meat', order_date='12/28/2013', order_priority='M', region='Middle East and North Africa')
Row(order_id=845056617, country='Netherlands', item_type='Fruits', order_date='11/19/2016', order_priority='L', region='Europe', sales_channel='Retail')
Row(order_id=434008300, country='Thailand', item_type='Fruits', order_date='9/30/2012', order_priority='C', region='Asia', sales_channel='Retail')
Row(order_id=862861335, country='The Gambia', item_type='Fruits', order_date='11/20/2011', order_priority='C', region='Sub-Saharan Africa')
Row(order_id=756839835, country='Romania', item_type='Beverages', order_date='2/19/2012', order_priority='M', region='Europe', sales_channel='Retail')
Row(order_id=479969346, country='Tunisia ', item_type='Cosmetics', order_date='7/1/2012', order_priority='H', region='Middle East and North Africa')
Row(order_id=116205585, country='Turkmenistan', item_type='Vegetables', order_date='4/13/2015', order_priority='M', region='Asia', sales_channel='Retail')
Row(order_id=496523940, country='Ghana', item_type='Fruits', order_date='5/29/2016', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=710296428, country='Kazakhstan', item_type='Snacks', order_date='9/8/2013', order_priority='L', region='Asia', sales_channel='Retail')
Row(order_id=642134416, country='Finland', item_type='Vegetables', order_date='10/9/2012', order_priority='C', region='Europe', sales_channel='Retail')
Row(order_id=573998582, country='Nicaragua', item_type='Household', order_date='10/28/2015', order_priority='C', region='Central America')
Row(order_id=584356629, country='Democratic Republic of the Congo', item_type='Cosmetics', order_date='10/7/2012', order_priority='M')
Row(order_id=704205024, country='Lebanon', item_type='Meat', order_date='3/8/2017', order_priority='L', region='Middle East and North Africa')
Row(order_id=830192887, country='Sri Lanka', item_type='Fruits', order_date='11/7/2011', order_priority='L', region='Asia', sales_channel='Retail')
Row(order_id=835696351, country='Estonia', item_type='Household', order_date='9/1/2011', order_priority='H', region='Europe', sales_channel='Retail')
Row(order_id=731120240, country='Saint Lucia', item_type='Fruits', order_date='12/12/2015', order_priority='L', region='Central America')
Row(order_id=371502530, country='Israel', item_type='Beverages', order_date='9/8/2013', order_priority='H', region='Middle East and North Africa')
Row(order_id=425793445, country='Seychelles ', item_type='Beverages', order_date='1/18/2013', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=937431466, country='Samoa ', item_type='Household', order_date='12/5/2016', order_priority='L', region='Australia and Oceania')
Row(order_id=485078693, country='Haiti', item_type='Office Supplies', order_date='12/31/2010', order_priority='C', region='Central America')
Row(order_id=757619178, country='Turkmenistan', item_type='Meat', order_date='4/30/2017', order_priority='M', region='Asia', sales_channel='Retail')
Row(order_id=882908809, country='Egypt', item_type='Clothes', order_date='8/9/2016', order_priority='C', region='Middle East and North Africa')
Row(order_id=146634709, country='Liberia', item_type='Baby Food', order_date='6/6/2015', order_priority='H', region='Sub-Saharan Africa')
Row(order_id=363086831, country='Mali', item_type='Household', order_date='8/19/2010', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=179137074, country='Nepal', item_type='Meat', order_date='2/21/2014', order_priority='H', region='Asia', sales_channel='Retail')
Row(order_id=529276502, country='Burundi', item_type='Beverages', order_date='3/9/2011', order_priority='H', region='Sub-Saharan Africa')
Row(order_id=297876536, country='South Korea', item_type='Meat', order_date='3/16/2016', order_priority='L', region='Asia', sales_channel='Retail')
Row(order_id=459386289, country='Kuwait', item_type='Household', order_date='6/13/2011', order_priority='C', region='Middle East and North Africa')
Row(order_id=887124383, country='Greece', item_type='Cereal', order_date='8/22/2015', order_priority='H', region='Europe', sales_channel='Retail')
Row(order_id=824714744, country='Dominican Republic', item_type='Baby Food', order_date='8/25/2011', order_priority='H', region='Central America')
Row(order_id=842238795, country='Uganda', item_type='Cosmetics', order_date='2/28/2015', order_priority='M', region='Sub-Saharan Africa')
Row(order_id=797385394, country='Italy', item_type='Office Supplies', order_date='3/29/2011', order_priority='H', region='Europe', sales_channel='Retail')
Row(order_id=153842341, country='Brunei', item_type='Cereal', order_date='5/28/2013', order_priority='L', region='Asia', sales_channel='Retail')
Row(order_id=659878194, country='Tanzania', item_type='Beverages', order_date='11/30/2016', order_priority='L', region='Sub-Saharan Africa')

```

```

create_silver_table_query = """
CREATE TABLE IF NOT EXISTS sales.silver_sales_data (
    order_id UUID PRIMARY KEY,
    region TEXT,
    country TEXT,
    item_type TEXT,
    sales_channel TEXT,
    order_priority TEXT,
    order_date DATE,
    ship_date DATE,
    units_sold INT,
    unit_price DECIMAL,
    unit_cost DECIMAL,
    total_revenue DECIMAL,
    total_cost DECIMAL,
    total_profit DECIMAL
);
"""

```

```
session.execute(create_silver_table_query)
```

```

import pandas as pd
from cassandra.cluster import Cluster

```

```

from datetime import datetime

csv_file_path = "sales_100.csv"
df = pd.read_csv(csv_file_path)

df.columns = df.columns.str.strip()

print("Column names in the DataFrame:", df.columns)

df['Order Date'] = pd.to_datetime(df['Order Date'], format='%m/%d/%Y')
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='%m/%d/%Y')

required_columns = ['Order ID', 'Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
                    'Order Date', 'Ship Date', 'UnitsSold', 'UnitPrice', 'UnitCost', 'TotalRevenue',
                    'TotalCost', 'TotalProfit']

missing_columns = [col for col in required_columns if col not in df.columns]

if missing_columns:
    print(f"Missing columns in the DataFrame: {missing_columns}")
else:

    for _, row in df.iterrows():

        query = """
        INSERT INTO sales.silver_sales_data (
            order_id, region, country, item_type, sales_channel, order_priority,
            order_date, ship_date, units_sold, unit_price, unit_cost, total_revenue,
            total_cost, total_profit
        )
        VALUES (%s, %s, %s)
        """

        session.execute(query, (
            row['Order ID'], # UUID type
            row['Region'], # TEXT type
            row['Country'], # TEXT type
            row['Item Type'], # TEXT type
            row['Sales Channel'], # TEXT type
            row['Order Priority'], # TEXT type
            row['Order Date'].date(), # Convert to DATE format
            row['Ship Date'].date(), # Convert to DATE format
            int(row['UnitsSold']), # INT type
            float(row['UnitPrice']), # DECIMAL type
            float(row['UnitCost']), # DECIMAL type
            float(row['TotalRevenue']), # DECIMAL type
            float(row['TotalCost']), # DECIMAL type
            float(row['TotalProfit']) # DECIMAL type
        ))

    print("Silver table populated successfully!")

```

→ Column names in the DataFrame: Index(['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority', 'Order Date', 'Order ID', 'Ship Date', 'UnitsSold', 'UnitPrice', 'UnitCost', 'TotalRevenue', 'TotalCost', 'TotalProfit'], dtype='object')
 Silver table populated successfully!

```

query = "SELECT * FROM sales.silver_sales_data ;"
rows = session.execute(query)

for row in rows:
    print(row)

```

→

```
order_id, order_qty, order_date 2015-01-10, order_priority 'L', profit_per_unit None, region 'Sub-Saharan Africa', revenue_per_unit None', order_age=None, order_date='2016-12-05', order_priority='L', profit_per_unit=None, region='Australia and Oceania', revenue_per_unit None', order_age=None, order_date='2010-12-31', order_priority='C', profit_per_unit=None, region='Central America and the Caribbean', at', order_age=None, order_date='2017-04-30', order_priority='M', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_ch order_age=None, order_date='2016-08-09', order_priority='C', profit_per_unit=None, region='Middle East and North Africa', revenue_per od', order_age=None, order_date='2015-06-06', order_priority='H', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit , order_age=None, order_date='2010-08-19', order_priority='M', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit Non der_age=None, order_date='2014-02-21', order_priority='H', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_channel=' es', order_age=None, order_date='2011-03-09', order_priority='H', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit=t', order_age=None, order_date='2016-03-16', order_priority='L', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_cha d', order_age=None, order_date='2011-06-13', order_priority='C', profit_per_unit=None, region='Middle East and North Africa', revenue_ order_age=None, order_date='2015-08-22', order_priority='H', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_cha pe='Baby Food', order_age=None, order_date='2011-08-25', order_priority='H', profit_per_unit=None, region='Central America and the Car s', order_age=None, order_date='2015-02-28', order_priority='M', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit=Non pplies', order_age=None, order_date='2011-03-29', order_priority='H', profit_per_unit=None, region='Europe', revenue_per_unit=None, sa order_age=None, order_date='2013-05-28', order_priority='L', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_cha ges', order_age=None, order_date='2016-11-30', order_priority='L', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit l Care', order_age=None, order_date='2010-04-04', order_priority='M', profit_per_unit=None, region='Asia', revenue_per_unit=None, sale order_age=None, order_date='2011-11-02', order_priority='M', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per unit=None ', order_age=None, order_date='2014-07-16', order_priority='M', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_cha ', order_age=None, order_date='2013-09-14', order_priority='M', profit_per_unit=None, region='Middle East and North Africa', revenue_p order_age=None, order_date='2015-11-30', order_priority='C', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_cha = 'Meat', order_age=None, order_date='2015-05-15', order_priority='M', profit_per_unit=None, region='Australia and Oceania', revenue_pe , order_age=None, order_date='2010-11-29', order_priority='H', profit_per_unit=None, region='Middle East and North Africa', revenue_pe od', order_age=None, order_date='2010-10-02', order_priority='M', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_ ', order_age=None, order_date='2013-01-27', order_priority='C', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_ch s', order_age=None, order_date='2012-09-13', order_priority='H', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_c Care', order_age=None, order_date='2015-06-14', order_priority='H', profit_per_unit=None, region='Central America and the Caribbean', ', order_age=None, order_date='2014-10-20', order_priority='H', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per_unit=No = 'Clothes', order_age=None, order_date='2011-06-20', order_priority='L', profit_per_unit=None, region='Australia and Oceania', revenue type='Office Supplies', order_age=None, order_date='2011-10-02', order_priority='C', profit_per_unit=None, region='Central America and , order_age=None, order_date='2015-04-16', order_priority='M', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_cha _type='Clothes', order_age=None, order_date='2015-11-11', order_priority='L', profit_per_unit=None, region='Sub-Saharan Africa', reven cs', order_age=None, order_date='2011-02-18', order_priority='M', profit_per_unit=None, region='Middle East and North Africa', revenue order_age=None, order_date='2012-09-05', order_priority='L', profit_per_unit=None, region='Sub-Saharan Africa', revenue_per_unit=Non ges', order_age=None, order_date='2016-10-26', order_priority='L', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales order_age=None, order_date='2012-10-10', order_priority='L', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_chann , order_age=None, order_date='2013-11-03', order_priority='C', profit_per_unit=None, region='Australia and Oceania', revenue_per unit= 'Household', order_age=None, order_date='2010-12-17', order_priority='M', profit_per_unit=None, region='Australia and Oceania', revenue ld', order_age=None, order_date='2016-01-15', order_priority='H', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_ ges', order_age=None, order_date='2012-06-12', order_priority='L', profit_per_unit=None, region='Central America and the Caribbean', r order_age=None, order_date='2014-02-09', order_priority='M', profit_per_unit=None, region='Asia', revenue_per_unit=None, sales_channe Cosmetics', order_age=None, order_date='2014-03-22', order_priority='H', profit_per_unit=None, region='Europe', revenue_per_unit=None, s', order_age=None, order_date='2016-05-29', order_priority='M', profit_per_unit=None, region='Europe', revenue_per_unit=None, sales_c
```

```
#Total_profit, Total_sales,Total_units_sold by country and Item_type
from cassandra.cluster import Cluster
from datetime import datetime
```

```
create_gold_table_query = """
CREATE TABLE IF NOT EXISTS sales.gold_table (
    country TEXT,
    item_type TEXT,
    total_sales TEXT,
    total_profit TEXT,
    average_profit_margin TEXT,
    total_units_sold TEXT,
    PRIMARY KEY (country, item_type)
);
"""

session.execute(create_gold_table_query)
```

```
select_query = """
SELECT country, item_type, total_revenue, total_profit, units_sold
FROM sales.sales_data;
"""

rows = session.execute(select_query)
```

```
insert_query = """
INSERT INTO sales.gold_table (country, item_type, total_sales, total_profit, average_profit_margin, total_units_sold)
VALUES (%s, %s, %s, %s, %s, %s)
"""

aggregated_data = {}
```

```
for row in rows:
    key = (row.country, row.item_type)
    if key not in aggregated_data:
        aggregated_data[key] = {
```

```
'total_revenue': 0,
'total_profit': 0,
'total_units_sold': 0
}

aggregated_data[key]['total_revenue'] += row.total_revenue
aggregated_data[key]['total_profit'] += row.total_profit
aggregated_data[key]['total_units_sold'] += row.units_sold

for key, values in aggregated_data.items():
    country, item_type = key
    total_revenue = str(values['total_revenue'])
    total_profit = str(values['total_profit'])
    total_units_sold = str(values['total_units_sold'])
    average_profit_margin = str(values['total_profit'] / values['total_revenue']) if values['total_revenue'] != 0 else '0'

    session.execute(insert_query, (
        str(country), str(item_type), total_revenue, total_profit, average_profit_margin, total_units_sold
    ))
}

print("Gold table populated successfully!")
```

→ Gold table populated successfully!

```
query = "SELECT * FROM sales.gold_table ;"  
rows = session.execute(query)
```

```
for row in rows:  
    print(row)  
  
Row(country='Democratic Republic of the Congo', item_type='Cosmetics', average_profit_margin='0.3976898453838874', total_profit='5158 ▲  
Row(country='Switzerland', item_type='Office Supplies', average_profit_margin='0.19386986983797674', total_profit='711923.75', total_  
Row(country='South Africa', item_type='Fruits', average_profit_margin='0.2583065226968615', total_profit='3839.1298828125', total_sal  
Row(country='Kazakhstan', item_type='Snacks', average_profit_margin='0.3613842045282229', total_profit='74549.28125', total_sales='20  
Row(country='Papua New Guinea', item_type='Clothes', average_profit_margin='0.6720351659854138', total_profit='667716.5', total_sales  
Row(country='Papua New Guinea', item_type='Meat', average_profit_margin='0.13558035897076093', total_profit='20592.0', total_sales='1  
Row(country='India', item_type='Snacks', average_profit_margin='0.36138420252945097', total_profit='294943.875', total_sales='816150.  
Row(country='Thailand', item_type='Fruits', average_profit_margin='0.25830653446008867', total_profit='3747.550048828125', total_sale  
Row(country='Thailand', item_type='Meat', average_profit_margin='0.13558036732837153', total_profit='251737.203125', total_sales='185
```

```
Row(country='Ethiopia', item_type='Cosmetics', average_profit_margin=0.39768985029494475, total_profit=115101.557, total_sales=115101.557)
Row(country='Malawi', item_type='Meat', average_profit_margin=0.13558036468855506, total_profit=291376.8125, total_sales=2149107)
Row(country='Italy', item_type='Cereal', average_profit_margin=0.4306757397673301, total_profit=904149.53125, total_sales=209937)
Row(country='East Timor', item_type='Office Supplies', average_profit_margin=0.19386987096164565, total_profit=367766.25, total_sales=367766.25)
Row(country='Singapore', item_type='Cereal', average_profit_margin=0.4306757453203314, total_profit=22944.810546875, total_sales=22944.810546875)
Row(country='Oman', item_type='Snacks', average_profit_margin=0.36138416520689765, total_profit=423254.625, total_sales=117)
Row(country='Oman', item_type='Cosmetics', average_profit_margin=0.39768985029494475, total_profit=1380006.25, total_sales=34700)
```

```
# Total_Sales by Item_type
create_gold_table_query_2 = """
CREATE TABLE IF NOT EXISTS sales.gold_sales_by_item_type (
    item_type TEXT PRIMARY KEY,
    total_revenue DECIMAL,
    total_profit DECIMAL,
    average_profit_margin DECIMAL,
    total_units_sold INT
)
"""

session.execute(create_gold_table_query_2)

select_query = """
SELECT item_type, total_revenue, total_profit, units_sold FROM sales.silver_sales_data
"""
rows = session.execute(select_query)

aggregated_data = {}

for row in rows:
    item_type = row.item_type
    total_revenue = row.total_revenue
    total_profit = row.total_profit
    units_sold = row.units_sold

    if item_type not in aggregated_data:
        aggregated_data[item_type] = {
            'total_revenue': 0,
            'total_profit': 0,
            'total_units_sold': 0
        }

    aggregated_data[item_type]['total_revenue'] += total_revenue
    aggregated_data[item_type]['total_profit'] += total_profit
    aggregated_data[item_type]['total_units_sold'] += units_sold

for item_type, data in aggregated_data.items():
    total_revenue = data['total_revenue']
    total_profit = data['total_profit']
    total_units_sold = data['total_units_sold']

    average_profit_margin = total_profit / total_revenue if total_revenue != 0 else 0

    insert_query = """
INSERT INTO sales.gold_sales_by_item_type (
    item_type, total_revenue, total_profit, average_profit_margin, total_units_sold
)
VALUES (%s, %s, %s, %s, %s)
"""
    session.execute(insert_query, (
        item_type,
        float(total_revenue),
        float(total_profit),
        float(average_profit_margin),
        int(total_units_sold)
    ))
```

```
print("Gold table 2 (sales by item type) populated successfully!")
```

→ Gold table 2 (sales by item type) populated successfully!

```
query = "SELECT * FROM sales.gold_sales_by_item_type ;"
rows = session.execute(query)

for row in rows:
    print(row)

Row(item_type='Household', average_profit_margin=Decimal('0.24799856264263057'), total_profit=Decimal('9552677.1640625'), total_revenue=
Row(item_type='Office Supplies', average_profit_margin=Decimal('0.19386987389220692'), total_profit=Decimal('5405267.5'), total_revenue=
Row(item_type='Vegetables', average_profit_margin=Decimal('0.4097754086540353'), total_profit=Decimal('465141.83203125'), total_revenue=
Row(item_type='Snacks', average_profit_margin=Decimal('0.3613841827905915'), total_profit=Decimal('792747.78125'), total_revenue=Decimal
Row(item_type='Personal Care', average_profit_margin=Decimal('0.30661935895585124'), total_profit=Decimal('978467.7001953125'), total_re
Row(item_type='Meat', average_profit_margin=Decimal('0.13558036346429897'), total_profit=Decimal('2884996.390625'), total_revenue=Decim
Row(item_type='Fruits', average_profit_margin=Decimal('0.2583065353157176'), total_profit=Decimal('158867.19897460938'), total_revenue=
Row(item_type='Beverages', average_profit_margin=Decimal('0.33003161366413364'), total_profit=Decimal('707925.95703125'), total_revenue=
Row(item_type='Cereal', average_profit_margin=Decimal('0.4306757455281105'), total_profit=Decimal('4055295.841796875'), total_revenue=De
Row(item_type='Cosmetics', average_profit_margin=Decimal('0.3976898414804783'), total_profit=Decimal('11424476.09375'), total_revenue=De
Row(item_type='Baby Food', average_profit_margin=Decimal('0.37550924012126713'), total_profit=Decimal('1952859.90625'), total_revenue=De
Row(item_type='Clothes', average_profit_margin=Decimal('0.6720351423094536'), total_profit=Decimal('2948469.1796875'), total_revenue=De
```

#Sales by region and country

```
create_gold_table_query_3 = """
CREATE TABLE IF NOT EXISTS sales.gold_sales_by_region_or_country (
    region TEXT,
    country TEXT,
    total_revenue DECIMAL,
    total_profit DECIMAL,
    total_units_sold INT,
    PRIMARY KEY (region, country)
)
"""

session.execute(create_gold_table_query_3)
```

```
select_query = """
SELECT region, country, total_revenue, total_profit, units_sold FROM sales.silver_sales_data
"""

rows = session.execute(select_query)
```

aggregated_data_by_region_country = {}

```
for row in rows:
    region = row.region
    country = row.country
    total_revenue = row.total_revenue
    total_profit = row.total_profit
    units_sold = row.units_sold

    if (region, country) not in aggregated_data_by_region_country:
        aggregated_data_by_region_country[(region, country)] = {
            'total_revenue': 0,
            'total_profit': 0,
            'total_units_sold': 0
        }
```

```
aggregated_data_by_region_country[(region, country)]['total_revenue'] += total_revenue
aggregated_data_by_region_country[(region, country)]['total_profit'] += total_profit
aggregated_data_by_region_country[(region, country)]['total_units_sold'] += units_sold
```

```
sorted_data = sorted(
    aggregated_data_by_region_country.items(),
    key=lambda x: x[1]['total_profit'],
    reverse=True
)
```

```
for (region, country), data in sorted_data:
    total_revenue = data['total_revenue']
    total_profit = data['total_profit']
    total_units_sold = data['total_units_sold']
```

```
insert_query = """
INSERT INTO sales.gold_sales_by_region_or_country (
    region, country, total_revenue, total_profit, total_units_sold
)
VALUES (%s, %s, %s, %s, %s)
"""

Row(item_type='Household', average_profit_margin=Decimal('0.24799856264263057'), total_profit=Decimal('9552677.1640625'), total_revenue=
Row(item_type='Office Supplies', average_profit_margin=Decimal('0.19386987389220692'), total_profit=Decimal('5405267.5'), total_revenue=
Row(item_type='Vegetables', average_profit_margin=Decimal('0.4097754086540353'), total_profit=Decimal('465141.83203125'), total_revenue=
Row(item_type='Snacks', average_profit_margin=Decimal('0.3613841827905915'), total_profit=Decimal('792747.78125'), total_revenue=Decimal
Row(item_type='Personal Care', average_profit_margin=Decimal('0.30661935895585124'), total_profit=Decimal('978467.7001953125'), total_re
Row(item_type='Meat', average_profit_margin=Decimal('0.13558036346429897'), total_profit=Decimal('2884996.390625'), total_revenue=Decim
Row(item_type='Fruits', average_profit_margin=Decimal('0.2583065353157176'), total_profit=Decimal('158867.19897460938'), total_revenue=
Row(item_type='Beverages', average_profit_margin=Decimal('0.33003161366413364'), total_profit=Decimal('707925.95703125'), total_revenue=
Row(item_type='Cereal', average_profit_margin=Decimal('0.4306757455281105'), total_profit=Decimal('4055295.841796875'), total_revenue=De
Row(item_type='Cosmetics', average_profit_margin=Decimal('0.3976898414804783'), total_profit=Decimal('11424476.09375'), total_revenue=De
Row(item_type='Baby Food', average_profit_margin=Decimal('0.37550924012126713'), total_profit=Decimal('1952859.90625'), total_revenue=De
Row(item_type='Clothes', average_profit_margin=Decimal('0.6720351423094536'), total_profit=Decimal('2948469.1796875'), total_revenue=De
```

```

session.execute(insert_query, (
    region,
    country,
    float(total_revenue),
    float(total_profit),
    int(total_units_sold)
))

print("Gold table 3 (sales by region/country) populated successfully!")

```

Gold table 3 (sales by region/country) populated successfully!

```

query = "SELECT * FROM sales.gold_sales_by_region_or_country ;"
rows = session.execute(query)

```

```

for row in rows:
    print(row)

```

```

→ Row(region='Australia and Oceania', country='East Timor', total_profit=Decimal('22944.810546875'), total_revenue=Decimal('53276.30078')
Row(region='Australia and Oceania', country='New Zealand', total_profit=Decimal('90640.078125'), total_revenue=Decimal('274640.59375')
Row(region='Australia and Oceania', country='Papua New Guinea', total_profit=Decimal('688308.5'), total_revenue=Decimal('1145454.1562
Row(region='Australia and Oceania', country='Samoa ', total_profit=Decimal('937534.625'), total_revenue=Decimal('3780403.5'), total_u
Row(region='Australia and Oceania', country='Solomon Islands', total_profit=Decimal('700209.25'), total_revenue=Decimal('2823440.75')
Row(region='Australia and Oceania', country='Tonga', total_profit=Decimal('530201.6875'), total_revenue=Decimal('1411953.625'), total
Row(region='Australia and Oceania', country='Vanuatu', total_profit=Decimal('517101.130859375'), total_revenue=Decimal('1222089.30078
Row(region='Europe', country='Albania', total_profit=Decimal('795002.953125'), total_revenue=Decimal('2140268.640625'), total_units_s
Row(region='Europe', country='Belgium', total_profit=Decimal('70844.6171875'), total_revenue=Decimal('231050.703125'), total_units_so
Row(region='Europe', country='Czech Republic', total_profit=Decimal('1592127.625'), total_revenue=Decimal('4003440.5'), total_units_s
Row(region='Europe', country='Estonia', total_profit=Decimal('1653322.5'), total_revenue=Decimal('6666661.5'), total_units_sold=9976)
Row(region='Europe', country='Finland', total_profit=Decimal('1394432.73828125'), total_revenue=Decimal('5552597.8828125'), total_uni
Row(region='Europe', country='France', total_profit=Decimal('1001143.4375'), total_revenue=Decimal('2517397.5'), total_units_sold=575
Row(region='Europe', country='Greece', total_profit=Decimal('768429.6875'), total_revenue=Decimal('1784241.75'), total_units_sold=867
Row(region='Europe', country='Iceland', total_profit=Decimal('236007.3125'), total_revenue=Decimal('628499.375'), total_units_sold=24
Row(region='Europe', country='Italy', total_profit=Decimal('1271915.78125'), total_revenue=Decimal('3996348.9375'), total_units_sold=
Row(region='Europe', country='Moldova ', total_profit=Decimal('9003.759765625'), total_revenue=Decimal('34856.87890625'), total_units
Row(region='Europe', country='Monaco', total_profit=Decimal('40856.94140625'), total_revenue=Decimal('123797.046875'), total_units_so
Row(region='Europe', country='Montenegro', total_profit=Decimal('155472.484375'), total_revenue=Decimal('231345.765625'), total_units
Row(region='Europe', country='Netherlands', total_profit=Decimal('23827.669921875'), total_revenue=Decimal('92245.7109375'), total_u
Row(region='Europe', country='Romania', total_profit=Decimal('721723.2734375'), total_revenue=Decimal('1726589.328125'), total_units_
Row(region='Europe', country='Serbia', total_profit=Decimal('539637.125'), total_revenue=Decimal('802989.4375'), total_units_sold=734
Row(region='Europe', country='Slovakia', total_profit=Decimal('62217.1796875'), total_revenue=Decimal('188518.84375'), total_units_so
Row(region='Europe', country='Sweden', total_profit=Decimal('38915.1015625'), total_revenue=Decimal('117913.25'), total_units_sold=24
Row(region='Europe', country='Switzerland', total_profit=Decimal('711923.75'), total_revenue=Decimal('3672173.25'), total_units_sold=
Row(region='Europe', country='United Kingdom', total_profit=Decimal('180477.0625'), total_revenue=Decimal('453813.59375'), total_unit
Row(region='Middle East and North Africa', country='Algeria', total_profit=Decimal('1681149.0'), total_revenue=Decimal('4227287.0'),
Row(region='Middle East and North Africa', country='Egypt', total_profit=Decimal('87540.4765625'), total_revenue=Decimal('130261.7578
Row(region='Middle East and North Africa', country='Israel', total_profit=Decimal('73742.9375'), total_revenue=Decimal('223442.046875
Row(region='Middle East and North Africa', country='Kuwait', total_profit=Decimal('242960.1875'), total_revenue=Decimal('979683.8125
Row(region='Middle East and North Africa', country='Lebanon', total_profit=Decimal('501644.0'), total_revenue=Decimal('3699975.25'),
Row(region='Middle East and North Africa', country='Morocco', total_profit=Decimal('338631.84375'), total_revenue=Decimal('503890.093
Row(region='Middle East and North Africa', country='Oman', total_profit=Decimal('1380006.25'), total_revenue=Decimal('3470056.5'), to
Row(region='Middle East and North Africa', country='Pakistan', total_profit=Decimal('570226.8125'), total_revenue=Decimal('4205821.5'
Row(region='Middle East and North Africa', country='Tunisia ', total_profit=Decimal('425981.5'), total_revenue=Decimal('1071140.0'),
Row(region='Middle East and North Africa', country='United Arab Emirates', total_profit=Decimal('1212378.75'), total_revenue=Decimal(
Row(region='Central America and the Caribbean', country='Antigua and Barbuda ', total_profit=Decimal('794996.25'), total_revenue=Deci
Row(region='Central America and the Caribbean', country='Belize', total_profit=Decimal('88536.9765625'), total_revenue=Decimal('28875
Row(region='Central America and the Caribbean', country='Dominica', total_profit=Decimal('1266075.78125'), total_revenue=Decimal('500
Row(region='Central America and the Caribbean', country='Dominican Republic', total_profit=Decimal('26265.640625'), total_revenue=Dec
Row(region='Central America and the Caribbean', country='Haiti', total_profit=Decimal('259065.0'), total_revenue=Decimal('1336282.875
Row(region='Central America and the Caribbean', country='Nicaragua', total_profit=Decimal('1568333.625'), total_revenue=Decimal('5944
Row(region='Central America and the Caribbean', country='Saint Lucia', total_profit=Decimal('8630.2099609375'), total_revenue=Decimal
Row(region='Central America and the Caribbean', country='The Bahamas', total_profit=Decimal('240396.619140625'), total_revenue=Decima
Row(region='Asia', country='Brunei', total_profit=Decimal('374026.96875'), total_revenue=Decimal('868465.375'), total_units_sold=4222
Row(region='Asia', country='China', total_profit=Decimal('731113.75'), total_revenue=Decimal('3771157.0'), total_units_sold=5791)
Row(region='Asia', country='India', total_profit=Decimal('294943.875'), total_revenue=Decimal('816150.4375'), total_units_sold=5349)
Row(region='Asia', country='Indonesia', total_profit=Decimal('336458.1484375'), total_revenue=Decimal('1176911.59375'), total_units_s
Row(region='Asia', country='Japan', total_profit=Decimal('1390971.875'), total_revenue=Decimal('5608790.0'), total_units_sold=8393)
Row(region='Asia', country='Kazakhstan', total_profit=Decimal('74549.28125'), total_revenue=Decimal('206288.15625'), total_units_sold
Row(region='Asia', country='Malaysia', total_profit=Decimal('143351.640625'), total_revenue=Decimal('434357.3125'), total_units_sold=
Row(region='Asia', country='Nepal', total_profit=Decimal('543171.1875'), total_revenue=Decimal('4006267.5'), total_units_sold=9496)
Row(region='Asia', country='Singapore', total_profit=Decimal('423254.625'), total_revenue=Decimal('1171204.125'), total_units_sold=76
Row(region='Asia', country='South Korea', total_profit=Decimal('408465.1875'), total_revenue=Decimal('3012716.5'), total_units_sold=7
Row(region='Asia', country='Sri Lanka', total_profit=Decimal('3323.389892578125'), total_revenue=Decimal('12866.0703125'), total_unit
Row(region='Asia', country='Taiwan', total_profit=Decimal('851842.189453125'), total_revenue=Decimal('2007920.09375'), total_units_so
Row(region='Asia', country='Thailand', total_profit=Decimal('255484.75317382812'), total_revenue=Decimal('1871246.025390625'), total_ ▾

```

```

# Average shipping time by region
create_gold_table_query_4 = """
CREATE TABLE IF NOT EXISTS sales.gold_average_shipping_time_by_region (

```

```

region TEXT PRIMARY KEY,
average_shipping_time DECIMAL
)
"""

session.execute(create_gold_table_query_4)

select_query = """
SELECT region, order_date, ship_date FROM sales.silver_sales_data
"""
rows = session.execute(select_query)

shipping_time_data = {}
for row in rows:
    region = row.region
    order_date = datetime.strptime(row.order_date, "%Y-%m-%d")
    ship_date = datetime.strptime(row.ship_date, "%Y-%m-%d")
    shipping_time = (ship_date - order_date).days

    if region not in shipping_time_data:
        shipping_time_data[region] = {
            'total_shipping_time': 0,
            'order_count': 0
        }

    shipping_time_data[region]['total_shipping_time'] += shipping_time
    shipping_time_data[region]['order_count'] += 1

average_shipping_time_by_region = {
    region: data['total_shipping_time'] / data['order_count']
    for region, data in shipping_time_data.items()
}
for region, avg_shipping_time in average_shipping_time_by_region.items():
    insert_query = """
    INSERT INTO sales.gold_average_shipping_time_by_region (region, average_shipping_time)
    VALUES (%s, %s)
    """
    session.execute(insert_query, (region, float(avg_shipping_time)))

print("Gold table 4 (average shipping time by region) populated successfully!")

```

→ Gold table 4 (average shipping time by region) populated successfully!

```

query = "SELECT * FROM sales.gold_average_shipping_time_by_region ;"
rows = session.execute(query)

for row in rows:
    print(row)

→ Row(region='Australia and Oceania', average_shipping_time=Decimal('19.88888888888889'))
Row(region='Europe', average_shipping_time=Decimal('24.625'))
Row(region='Middle East and North Africa', average_shipping_time=Decimal('28.0'))
Row(region='Central America and the Caribbean', average_shipping_time=Decimal('26.636363636363637'))
Row(region='Asia', average_shipping_time=Decimal('29.94736842105263'))
Row(region='Sub-Saharan Africa', average_shipping_time=Decimal('19.45833333333332'))
Row(region='North America', average_shipping_time=Decimal('26.5'))

```