```python
import sqlite3
import pandas as pd
import logging
from ingestion_db import ingest_db

logging.basicConfig(
    filename="logs/get_vendor_summary.log",
    level=logging.DEBUG,
    format="%(asctime)s - %(levelname)s - %(message)s",
    filemode="a"
)

def create_vendor_summary(conn):
    """Merge different tables to get vendor summary and add new columns"""
    vendor_sales_summary = pd.read_sql_query("""
        WITH FreightSummary AS (
            SELECT
                VendorNumber,
                SUM(Freight) AS FreightCost
            FROM vendor_invoice
            GROUP BY VendorNumber
        ),

        PurchaseSummary AS (
            SELECT
                p.VendorNumber,
                p.VendorName,
                p.Brand,
                p.Description,
                p.PurchasePrice,
                pp.Price AS ActualPrice,
                pp.Volume,
                SUM(p.Quantity) AS TotalPurchaseQuantity,
                SUM(p.Dollars) AS TotalPurchaseDollars
            FROM purchases p
            JOIN purchase_prices pp
                ON p.Brand = pp.Brand
            WHERE p.PurchasePrice > 0
            GROUP BY p.VendorNumber, p.VendorName, p.Brand, p.Description, p.PurchasePrice, pp.Price, pp.Volume
        ),

        SalesSummary AS (
            SELECT
                VendorNo,
                Brand,
                SUM(SalesQuantity) AS TotalSalesQuantity,
                SUM(SalesDollars) AS TotalSalesDollars,
                SUM(SalesPrice) AS TotalSalesPrice,
                SUM(ExciseTax) AS TotalExciseTax
            FROM sales
            GROUP BY VendorNo, Brand


        SELECT
            ps.VendorNumber,
            ps.VendorName,
            ps.Brand,
            ps.Description,
            ps.PurchasePrice,
            ps.ActualPrice,
            ps.Volume,
            ps.TotalPurchaseQuantity,
            ps.TotalPurchaseDollars,
            ss.TotalSalesQuantity,
            ss.TotalSalesDollars,
            ss.TotalSalesPrice,
            ss.TotalExciseTax,
            fs.FreightCost
        FROM PurchaseSummary ps
        LEFT JOIN SalesSummary ss
            ON ps.VendorNumber = ss.VendorNo
            AND ps.Brand = ss.Brand
        LEFT JOIN FreightSummary fs
            ON ps.VendorNumber = fs.VendorNumber
        ORDER BY ps.TotalPurchaseDollars DESC""", conn)

    return vendor_sales_summary


def clean_data(df):
    """Clean the data"""
    # changing datatype
    df['Volume'] = df['Volume'].astype(float)

    # filling missing value with 0
    df.fillna(0, inplace=True)

    # stripping spaces
    df['VendorName'] = df['VendorName'].str.strip()
    df['Description'] = df['Description'].str.strip()

    # creating new columns
    df['GrossProfit'] = df['TotalSalesDollars'] - df['TotalPurchaseDollars']
    df['ProfitMargin'] = (df['GrossProfit'] / df['TotalSalesDollars']) * 100
    df['StockTurnover'] = df['TotalSalesQuantity'] / df['TotalPurchaseQuantity']
    df['SalestoPurchaseRatio'] = df['TotalSalesDollars'] / df['TotalPurchaseDollars']

    return df


if __name__ == '__main__':
    conn = sqlite3.connect('inventory.db')

    logging.info('Creating Vendor Summary Table ....')
    summary_df = create_vendor_summary(conn)
    logging.info(summary_df.head())

    logging.info('Cleaning Data.....')
    clean_df = clean_data(summary_df)
    logging.info(clean_df.head())

    logging.info('Ingesting data.....')
    ingest_db(clean_df, 'vendor_sales_summary', conn)
    logging.info('Complete')
```