Exploratory Data Analysis

Understanding the dataset to explore how the data is present in the database and if there is a need of creating some aggregated tables that can help with:

Vendor selection for profitability

Product Pricing Optimization

In [1]:

```
import pandas as pd
import sqlite3
```

In [2]:

```
# creating database connection
conn = sqlite3.connect('inventory.db')
```

In [3]:

```
# creating tables present in the database
tables = pd.read_sql_query("SELECT name FROM sqlite_master WHERE type ='table'
tables
```

Out[3]:

	name
0	begin_inventory
1	end_inventory
2	purchases
3	purchase_prices
4	sales
5	vendor_invoice

In [4]:

```
for table in tables ['name']:
    print('-'*50, f'{table}','-'*50)
    print("Count of records:", pd.read_sql(f"select count(*) as count from {ta
    display(pd.read_sql(f"select * from {table} limit 5", conn))
```

```
-------------------------------------------------- begin_inventory
--------------------------------------------------
Count of records: 619587
```

	InventoryId	Store	City	Brand	Description	Size	onHand
0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	8
1	1_HARDERSFIELD_60	1	HARDERSFIELD	60	Canadian Club 1858 VAP	750mL	7
2	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	6
3	1_HARDERSFIELD_63	1	HARDERSFIELD	63	Herradura Reposado Tequila	750mL	3
4	1_HARDERSFIELD_72	1	HARDERSFIELD	72	No. 3 London Dry Gin	750mL	6

```
-------------------------------------------------- end_inventory
--------------------------------------------------
Count of records: 673467
```

	InventoryId	Store	City	Brand	Description	Size	onHand
0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	11
1	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	7
2	1_HARDERSFIELD_63	1	HARDERSFIELD	63	Herradura Reposado Tequila	750mL	7
3	1_HARDERSFIELD_72	1	HARDERSFIELD	72	No. 3 London Dry Gin	750mL	4
4	1_HARDERSFIELD_75	1	HARDERSFIELD	75	Three Olives Tomato Vodka	750mL	7

```
-------------------------------------------------- purchases
--------------------------------------------------
Count of records: 7117422
```

	InventoryId	Store	Brand	Description	Size	VendorNumber	Vendo
0	69_MOUNTMEND_8412	69	8412	Tequila Ocho Plata Fresno	750mL	105	A BRAN
1	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AM V BEY
2	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	AM V BEY
3	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AM V BEY
4	76_DONCASTER_2034	76	2034	Glendalough Double Barrel	750mL	388	AT IMP CC

```
-------------------------------------------------- purchase_prices
--------------------------------------------------
Count of records: 36783
```

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	Vendo
0	58	Gekkeikan Black & Gold Sake	12.99	750mL	750	1	9.28	
1	62	Herradura Silver Tequila	36.99	750mL	750	1	28.67	
2	63	Herradura Reposado Tequila	38.99	750mL	750	1	30.46	
3	72	No. 3 London Dry Gin	34.99	750mL	750	1	26.11	
4	75	Three Olives Tomato Vodka	14.99	750mL	750	1	10.94	

```
-------------------------------------------------- sales
--------------------------------------------------
Count of records: 25650726
```

	InventoryId	Store	Brand	Description	Size	SalesQuantity	SalesD
0	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	
1	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	2	
2	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	
3	1_HARDERSFIELD_1004	1	1004	Jim Beam w/2 Rocks Glasses	750mL	1	
4	1_HARDERSFIELD_1005	1	1005	Maker's Mark Combo Pack	375mL 2 Pk	2	

```
-------------------------------------------------- vendor_invoice
--------------------------------------------------
Count of records: 11086
```

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate
0	105	ALTAMAR BRANDS LLC	2024-01-04	8124	2023-12-21	2024-02-16
1	4466	AMERICAN VINTAGE BEVERAGE	2024-01-07	8137	2023-12-22	2024-02-21
2	388	ATLANTIC IMPORTING COMPANY	2024-01-09	8169	2023-12-24	2024-02-16
3	480	BACARDI USA INC	2024-01-12	8106	2023-12-20	2024-02-05
4	516	BANFI PRODUCTS CORP	2024-01-07	8170	2023-12-24	2024-02-12

```
In [5]: purchases = pd.read_sql_query("Select * from purchases where VendorNumber = 44
        purchases
```

Out[5]:

	InventoryId	Store	Brand	Description	Size	VendorNumber	V
0	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
1	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	
2	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
3	38_GOULCREST_5215	38	5215	TGI Fridays Long Island Iced	1.75L	4466	
4	59_CLAETHORPES_5215	59	5215	TGI Fridays Long Island Iced	1.75L	4466	
...	...	...	...	...	...	...	
6571	81_PEMBROKE_5215	81	5215	TGI Fridays Long Island Iced	1.75L	4466	
6572	62_KILMARNOCK_5255	62	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
6573	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	
6574	6_GOULCREST_5215	6	5215	TGI Fridays Long Island Iced	1.75L	4466	
6575	35_HALIVAARA_5255	35	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	

6576 rows × 16 columns

In [6]:

```
purchase_prices = pd.read_sql_query("Select * from purchase_prices where Vendo
purchase_prices
```

Out[6]:

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	Ve
0	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
1	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
2	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	
3	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
4	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
5	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	
6	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
7	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
8	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	

In [7]:

```
sales = pd.read_sql_query("Select * from sales where VendorNo = 4466",conn)
sales
```

Out[7]:

	InventoryId	Store	Brand	Description	Size	SalesQuantity	S
0	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	1	
1	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	1	
2	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	1	
3	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	1	
4	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	1	
...	...	...	...	...	...	...	
18901	9_BLACKPOOL_5215	9	5215	TGI Fridays Long Island Iced	1.75L	1	
18902	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
18903	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
18904	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
18905	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	

18906 rows × 14 columns

```
In [8]: purchases.groupby(['Brand','PurchasePrice'])[['Quantity','Dollars']].sum()
```

Out[8]:

		Quantity	Dollars
Brand	**PurchasePrice**		
3140	**11.19**	13920	155764.80
5215	**9.41**	14769	138976.29
5255	**9.35**	18645	174330.75

```
In [9]: purchase_prices
```

Out[9]:

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	Ve
0	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
1	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
2	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	
3	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
4	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
5	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	
6	5215	TGI Fridays Long Island Iced	12.99	1750mL	1750	1	9.41	
7	5255	TGI Fridays Ultimte Mudslide	12.99	1750mL	1750	1	9.35	
8	3140	TGI Fridays Orange Dream	14.99	1750mL	1750	1	11.19	

```
In [10]: conn = sqlite3.connect("inventory.db")
         vendor_invoice = pd.read_sql("SELECT * FROM vendor_invoice", conn)
         print(vendor_invoice.head())
```

```
   VendorNumber                   VendorName InvoiceDate  PONumber  \
0           105  ALTAMAR BRANDS LLC           2024-01-04      8124   
1          4466  AMERICAN VINTAGE BEVERAGE    2024-01-07      8137   
2           388  ATLANTIC IMPORTING COMPANY   2024-01-09      8169   
3           480  BACARDI USA INC              2024-01-12      8106   
4           516  BANFI PRODUCTS CORP          2024-01-07      8170   

       PODate     PayDate  Quantity    Dollars  Freight Approval  
0  2023-12-21  2024-02-16         6     214.26     3.47     None  
1  2023-12-22  2024-02-21        15     140.55     8.57     None  
2  2023-12-24  2024-02-16         5     106.60     4.61     None  
3  2023-12-20  2024-02-05     10100  137483.78  2935.20     None  
4  2023-12-24  2024-02-12      1935   15527.25   429.20     None  
```

In [11]:
```
vendor_invoice['PONumber'].nunique()
```

Out[11]:
```
5543
```

In [12]:
```
vendor_invoice.columns
```

Out[12]:
```
Index(['VendorNumber', 'VendorName', 'InvoiceDate', 'PONumber', 'PODate',
       'PayDate', 'Quantity', 'Dollars', 'Freight', 'Approval'],
      dtype='object')
```

In [13]:
```
purchases
```

Out[13]:

	InventoryId	Store	Brand	Description	Size	VendorNumber	V
0	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
1	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	
2	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
3	38_GOULCREST_5215	38	5215	TGI Fridays Long Island Iced	1.75L	4466	
4	59_CLAETHORPES_5215	59	5215	TGI Fridays Long Island Iced	1.75L	4466	
...	...	...	...	...	...	...	
6571	81_PEMBROKE_5215	81	5215	TGI Fridays Long Island Iced	1.75L	4466	
6572	62_KILMARNOCK_5255	62	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	
6573	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	
6574	6_GOULCREST_5215	6	5215	TGI Fridays Long Island Iced	1.75L	4466	
6575	35_HALIVAARA_5255	35	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	

6576 rows × 16 columns

In [14]:
```
sales.groupby(['Brand'])[['SalesDollars', 'SalesPrice', 'SalesQuantity']].sum(
```

Out[14]:

Brand	SalesDollars	SalesPrice	SalesQuantity
3140	101062.20	60143.70	7780
5215	120832.98	83084.04	9302
5255	158374.08	102361.20	12192

The purchases table contains actual purchases data, including the data of purchases, products (brands) purchased by vendors, the amount paid (in dollars), and the quantity purchased.

The purchases price column is driven from the purchases_prices table, which provides product wise actual and purchases pricrs. The combination of vendor and brand is unique in this table.

The vendor_invoice table aggragates data from the purchases table, summarizing quantity and dollar amounts, along with an additional column for freight. This table maintains uniqueness based on venfor and PO number.

The sales table captures actual sales transations, detailing the brands purchased by vendors, the quantity sold, the selling price, and the revenue earned.

As the data that we need for analysis distributed tables, we need to create a summary table containing.

Purchases transactions made by vendors.

Sales transactions data.

Freight costs for each vendors.

Actual products prices from vendors.

In [25]:
```
vendor_invoice.columns
```

Out[25]:
```
Index(['VendorNumber', 'VendorName', 'InvoiceDate', 'PONumber', 'PODate',
       'PayDate', 'Quantity', 'Dollars', 'Freight', 'Approval'],
      dtype='object')
```

In [26]:
```
freight_summary = pd.read_sql_query(""" select VendorNumber, sum(freight) as F
from vendor_invoice
group by VendorNumber""", conn)
```

In [27]:

```
freight_summary
```

Out[27]:

	VendorNumber	FreightCost
0	2	54.16
1	54	0.96
2	60	735.04
3	105	124.78
4	200	12.38
...	...	...
121	98450	1712.04
122	99166	260.18
123	172662	356.68
124	173357	405.00
125	201359	0.18

126 rows × 2 columns

In [18]:

```
pd.read_sql_query("""SELECT
                    p.VendorNumber,
                    p.VendorName,
                    p.Brand,
                    p.PurchasePrice,
                    pp.Volume,
                    pp.Price as ActualPrice,
                    SUM(p.Quantity) as TotalPurchaseQuantity,
                    SUM(p.Dollars) as TotalPriceDollars
                    FROM purchases p
                    JOIN purchase_prices pp
                    ON p.Brand = pp.Brand
                    where p.PurchasePrice > 0
                    GROUP BY p.VendorNumber, p.VendorName, p.Brand
                    ORDER BY TotalPriceDollars""", conn)
```

Out[18]:

	VendorNumber	VendorName	Brand	PurchasePrice	Volume	ActualPric
0	7245	PROXIMO SPIRITS INC.	3065	0.71	50	0.9
1	3960	DIAGEO NORTH AMERICA INC	6127	1.47	200	1.9
2	3924	HEAVEN HILL DISTILLERIES	9123	0.74	50	0.9
3	8004	SAZERAC CO INC	5683	0.39	50	0.4
4	9815	WINE GROUP INC	8527	1.32	750	4.9
...	...	...	...	...	...	..
10687	3960	DIAGEO NORTH AMERICA INC	3545	21.89	1750	29.9
10688	3960	DIAGEO NORTH AMERICA INC	4261	16.17	1750	22.9
10689	17035	PERNOD RICARD USA	8068	18.24	1750	24.9
10690	4425	MARTIGNETTI COMPANIES	3405	23.19	1750	28.9
10691	1128	BROWN-FORMAN CORP	1233	26.27	1750	36.9

10692 rows × 8 columns

In [19]:

```
sales.columns
```

Out[19]:

```
Index(['InventoryId', 'Store', 'Brand', 'Description', 'Size', 'SalesQuantit
y',
       'SalesDollars', 'SalesPrice', 'SalesDate', 'Volume', 'Classification',
       'ExciseTax', 'VendorNo', 'VendorName'],
      dtype='object')
```

In [20]:

```
pd.read_sql_query(""" select
                    VendorNo,
                    Brand,
                    sum(SalesDollars) as TotalSalesDollars,
                    sum(SalesPrice) as TotalSalesPrice,
                    sum(SalesQuantity) as TotalSalesQuantity,
                    sum(ExciseTax) as TotalExciseTax
                    from sales
                    group by VendorNo, Brand
```

```
                order by TotalSalesDollars""", conn)
```

Out[20]:

	VendorNo	Brand	TotalSalesDollars	TotalSalesPrice	TotalSalesQuantity
0	8004	5287	1.96	1.96	4
1	9206	2773	1.98	1.98	2
2	3252	3933	3.96	1.98	4
3	3924	9123	3.96	1.98	4
4	10050	3623	3.96	3.96	4
...	...	...	...	...	...
11267	3960	3545	8446215.24	1091556.56	271676
11268	3960	4261	8951945.76	840100.02	400824
11269	17035	8068	9076241.20	922280.30	374280
11270	4425	3405	9638146.98	1123024.74	320494
11271	1128	1233	10203839.02	1345638.62	284098

11272 rows × 6 columns

In [30]:

```
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.PurchaseP
),

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)
```

In [31]:

```
vendor_sales_summary
```

Out[31]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	Actua
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	
...	...	...	...	...	...	
10687	9815	WINE GROUP INC	8527	Concannon Glen Ellen Wh Zin	1.32	
10688	8004	SAZERAC CO INC	5683	Dr McGillicuddy's Apple Pie	0.39	
10689	3924	HEAVEN HILL DISTILLERIES	9123	Deep Eddy Vodka	0.74	
10690	3960	DIAGEO NORTH AMERICA INC	6127	The Club Strawbry Margarita	1.47	
10691	7245	PROXIMO SPIRITS INC.	3065	Three Olives Grape Vodka	0.71	

10692 rows × 14 columns

The query generates a vendors-wise sales and purchases summary, which is valuable for:

Performace Optimization

- the query involves heavy joins and aggregation on large datasets like sales and purchases.
- Storing the pre-aggregated results avoids repeated expensive computations
- Helps in analyzing sales, purchases, and pricing for different vendors

and brands.

- Future benefits of storing this data for faster dashboarding & reporting.
- Instead of running expensive queries each time, dashboards can fetch data quickely from vendors_sales_summary

In [42]:
```
vendor_sales_summary.dtypes
```

Out[42]:
```
VendorNumber                int64
VendorName                 object
Brand                       int64
Description                object
PurchasePrice             float64
ActualPrice               float64
Volume                    float64
TotalPurchaseQuantity       int64
TotalPurchaseDollars      float64
TotalSalesQuantity        float64
TotalSalesDollars         float64
TotalSalesPrice           float64
TotalExciseTax            float64
FreightCost               float64
dtype: object
```

In [41]:
```
vendor_sales_summary.isnull().sum()
```

Out[41]:
```
VendorNumber             0
VendorName               0
Brand                    0
Description              0
PurchasePrice            0
ActualPrice              0
Volume                   0
TotalPurchaseQuantity    0
TotalPurchaseDollars     0
TotalSalesQuantity       0
TotalSalesDollars        0
TotalSalesPrice          0
TotalExciseTax           0
FreightCost              0
dtype: int64
```

In [40]:
```
vendor_sales_summary['VendorName'].unique()
```

```
Out[40]: array(['BROWN-FORMAN CORP', 'MARTIGNETTI COMPANIES', 'PERNOD RICARD USA',
               'DIAGEO NORTH AMERICA INC', 'BACARDI USA INC',
               'JIM BEAM BRANDS COMPANY', 'MAJESTIC FINE WINES',
               'ULTRA BEVERAGE COMPANY LLP', 'STOLI GROUP,(USA) LLC',
               'PROXIMO SPIRITS INC.', 'MOET HENNESSY USA INC', 'CAMPARI AMERICA',
               'SAZERAC CO INC', 'CONSTELLATION BRANDS INC', 'M S WALKER INC',
               'SAZERAC NORTH AMERICA INC.', 'PALM BAY INTERNATIONAL INC',
               'REMY COINTREAU USA INC', 'SIDNEY FRANK IMPORTING CO',
               'E & J GALLO WINERY', 'WILLIAM GRANT & SONS INC',
               'HEAVEN HILL DISTILLERIES', 'DISARONNO INTERNATIONAL LLC',
               'EDRINGTON AMERICAS', 'CASTLE BRANDS CORP.',
               'SOUTHERN WINE & SPIRITS NE', 'STE MICHELLE WINE ESTATES',
               'TRINCHERO FAMILY ESTATES', 'MHW LTD', 'WINE GROUP INC',
               'PERFECTA WINES', 'LUXCO INC', 'TREASURY WINE ESTATES',
               'DIAGEO CHATEAU ESTATE WINES', 'SHAW ROSS INT L IMP LTD',
               'PINE STATE TRADING CO', 'PHILLIPS PRODUCTS CO.',
               'CALEDONIA SPIRITS INC', 'STATE WINE & SPIRITS',
               'KOBRAND CORPORATION', 'BANFI PRODUCTS CORP',
               'VINEYARD BRANDS INC', 'DELICATO VINEYARDS INC',
               'FABRIZIA SPIRITS LLC', 'DUGGANS DISTILLED PRODUCTS',
               'Serralles Usa LLC', 'SEA HAGG DISTILLERY LLC',
               'OLE SMOKY DISTILLERY LLC', 'VRANKEN AMERICA', 'KLIN SPIRITS LLC',
               'LAIRD & CO', 'ADAMBA IMPORTS INTL INC',
               'LATITUDE BEVERAGE COMPANY', 'FREDERICK WILDMAN & SONS',
               'MCCORMICK DISTILLING CO', 'CHARLES JACQUIN ET CIE INC',
               'WESTERN SPIRITS BEVERAGE CO', 'MARSALLE COMPANY',
               'AMERICAN VINTAGE BEVERAGE', 'MANGO BOTTLING INC',
               'SWEET BABY VINEYARD', 'NICHE W & S', 'LABELLE VYDS AND WINERY',
               'FLAG HILL WINERY & VINEYARD', 'SMOKY QUARTZ DISTILLERY LLC',
               'PREMIUM PORT WINES, INC.', 'Russian Standard Vodka',
               'Dunn Wine Brokers', 'WEIN BAUER INC', 'BULLY BOY DISTILLERS',
               'ATLANTIC IMPORTING COMPANY', 'PREMIER DISTRIBUTORS',
               'VINILANDIA USA', 'PARK STREET IMPORTS LLC', 'TAKARA SAKE USA INC',
               'SEA BREEZE CELLARS LLC', 'STARK BREWING COMPANY', 'TY KU LLC',
               'PSP WINES', 'TAMWORTH DISTILLING', 'ZORVINO VINEYARDS',
               'SOUTHERN GLAZERS W&S OF NE', 'HOOD RIVER DISTILLERS, Inc.',
               'CRUSH WINES', 'POVERTY LANE ORCHARDS', 'DJINN SPIRITS LLC',
               'MOONLIGHT MEADERY', 'TALL SHIP DISTILLERY LLC',
               'FORTUNE WINE BROKERS LLC', 'BLACK COVE BEVERAGES', 'VINEXTRA INC',
               'SURVILLE ENTERPRISES CORP', 'JEWELL TOWNE VINEYARDS',
               'SWEETWATER FARM', 'ALTAMAR BRANDS LLC', 'CANDIA VINEYARDS',
               'INCREDIBREW INC', 'ALISA CARR BEVERAGES',
               'STELLAR IMPORTING CO LLC', 'FULCHINO VINEYARD INC',
               'IRA GOLDMAN AND WILLIAMS, LLP', 'Circa Wines',
               'VINEDREA WINES LLC', 'BLACK PRINCE DISTILLERY INC',
               'VINEYARD BRANDS LLC', 'THE IMPORTED GRAPE LLC',
               'WALPOLE MTN VIEW WINERY', 'GILMANTON WINERY & VINEYARD',
               'HAUNTING WHISPER VYDS', 'STAR INDUSTRIES INC.',
               'LOYAL DOG WINERY', 'R.P.IMPORTS INC', 'THE PIERPONT GROUP LLC',
               'APPOLO VINEYARDS LLC', 'BLACK ROCK SPIRITS LLC',
               'CENTEUR IMPORTS LLC', 'HIGHLAND WINE MERCHANTS LLC',
               'AMERICAN SPIRITS EXCHANGE', 'UNCORKED', 'BRONCO WINE COMPANY',
               'MILTONS DISTRIBUTING CO', 'TRUETT HURST', 'LAUREATE IMPORTS CO',
               'FANTASY FINE WINES CORP', 'AAPER ALCOHOL & CHEMICAL CO',
               'SILVER MOUNTAIN CIDERS', 'CAPSTONE INTERNATIONAL',
               'FLAVOR ESSENCE INC'], dtype=object)
```

In [35]:
```
vendor_sales_summary['Description'].unique()
```

Out[35]:
```
array(['Jack Daniels No 7 Black', "Tito's Handmade Vodka",
       'Absolut 80 Proof', ..., 'Crown Royal Apple',
       'Concannon Glen Ellen Wh Zin', 'The Club Strawbry Margarita'],
      shape=(9651,), dtype=object)
```

In [36]:
```
vendor_sales_summary['Volume'] = vendor_sales_summary['Volume'].astype('float6
```

In [37]:
```
vendor_sales_summary.fillna(0, inplace = True)
```

In [39]:
```
vendor_sales_summary['VendorName'] = vendor_sales_summary['VendorName'].str.st
```

In [43]:
```
vendor_sales_summary['GrossProfit'] = vendor_sales_summary['TotalSalesDollars'
```

In [44]:
```
vendor_sales_summary
```

Out[44]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	Actua
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	
...	...	...	...	...	...	
10687	9815	WINE GROUP INC	8527	Concannon Glen Ellen Wh Zin	1.32	
10688	8004	SAZERAC CO INC	5683	Dr McGillicuddy's Apple Pie	0.39	
10689	3924	HEAVEN HILL DISTILLERIES	9123	Deep Eddy Vodka	0.74	
10690	3960	DIAGEO NORTH AMERICA INC	6127	The Club Strawbry Margarita	1.47	
10691	7245	PROXIMO SPIRITS INC.	3065	Three Olives Grape Vodka	0.71	

10692 rows × 15 columns

In [45]:
```
vendor_sales_summary['GrossProfit'].min()
```

Out[45]:
```
np.float64(-24598223.999999996)
```

In [46]:
```
vendor_sales_summary['ProfitMargin'] = (vendor_sales_summary['GrossProfit'] /
```

In [47]:
```
vendor_sales_summary['StockTurnover'] = vendor_sales_summary['TotalSalesQuanti
```

In [48]:
```
vendor_sales_summary['SalestoPurchaseRatio'] = vendor_sales_summary['TotalSale
```

In [49]:
```
vendor_sales_summary.columns
```

```
Out[49]: Index(['VendorNumber', 'VendorName', 'Brand', 'Description', 'PurchasePrice',
                'ActualPrice', 'Volume', 'TotalPurchaseQuantity',
                'TotalPurchaseDollars', 'TotalSalesQuantity', 'TotalSalesDollars',
                'TotalSalesPrice', 'TotalExciseTax', 'FreightCost', 'GrossProfit',
                'ProfitMargin', 'StockTurnover', 'SalestoPurchaseRatio'],
               dtype='object')
```

```
In [50]: cursor = conn.cursor()
```

```
In [51]: cursor.execute(""" create table vendor_sales_summary (
             VendorNumber int,
             VendorName varchar(100),
             Brand int
             Description varchar(100),
             PurchasePrice decimal(10, 2),
             ActualPrice decimal(10,2),
             Volume,
             TotalPurchaseQuantity int,
             TotalPurchaseDollars decimal(15,2),
             TotalSalesQuantity int,
             TotalSalesDollars decimal(15,2),
             TotalSalesPrice decimal(15,2),
             TotalExciseTax decimal(15,2),
             FreightCost decimal(15,2),
             GrossProfit decimal(15,2),
             ProfitMargin decimal(15,2),
             StockTurnover decimal(15,2),
             SalesToPurchaseRatio decimal(15,2),
             Primary key (VendorNumber, Brand)
         );
         """)
```

```
Out[51]: <sqlite3.Cursor at 0x22712caa1c0>
```

```
In [54]: pd.read_sql_query("select * from vendor_sales_summary", conn)
```

Out[54]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	Actua
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black	26.27	
1	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka	23.19	
2	17035	PERNOD RICARD USA	8068	Absolut 80 Proof	18.24	
3	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum	16.17	
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka	21.89	
...	...	...	...	...	...	
10687	9815	WINE GROUP INC	8527	Concannon Glen Ellen Wh Zin	1.32	
10688	8004	SAZERAC CO INC	5683	Dr McGillicuddy's Apple Pie	0.39	
10689	3924	HEAVEN HILL DISTILLERIES	9123	Deep Eddy Vodka	0.74	
10690	3960	DIAGEO NORTH AMERICA INC	6127	The Club Strawbry Margarita	1.47	
10691	7245	PROXIMO SPIRITS INC.	3065	Three Olives Grape Vodka	0.71	

10692 rows × 18 columns

In [65]:

```
vendor_sales_summary.to_sql('vendor_sales_summary', conn, if_exists = 'replace
```

Out[65]:

```
10692
```

In [78]:

```
import os
print(os.getcwd())  # Notebook ka current working directory
print(os.listdir("logs"))  # logs folder ke andar kya files hain
```

```
D:\Data Analysis\Project Data Vendor Performance P
['.ipynb_checkpoints', '.log', 'ingestion_db.log']
```

In [81]:

```
log_path = r"D:/Data Analysis/Project Data Vendor Performance P/logs\get_vendo
os.makedirs(os.path.dirname(log_path), exist_ok=True)

logging.basicConfig(
    filename=log_path,
    level=logging.DEBUG,
    format="%(asctime)s - %(levelname)s - %(message)s",
    filemode="a"
)
```

In [76]:

```
```

In [ ]:

```
```

In [66]:

```
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.P
        ),

        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['TotalPurchaseDo
    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')
```

In [ ]: