

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]: ▶ # checking tables present in the database  
tables = pd.read_sql_query('SELECT name FROM sqlite_master WHERE type=  
tables
```

Out[3]:

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

```
In [4]: ▶ for table in tables['name']:  
        print('-'*50,f'{table}','-'*50)  
        print('Count of records:',pd.read_sql(f"SELECT count(*) AS count F  
        display(pd.read_sql(f'SELECT * FROM {table} LIMIT 5',conn))
```

----- begin_inventor
y -----
Count of records: 206529

	InventoryId	Store	City	Brand	Description	Size	onH
0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	
1	1_HARDERSFIELD_60	1	HARDERSFIELD	60	Canadian Club 1858 VAP	750mL	
2	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	
3	1_HARDERSFIELD_63	1	HARDERSFIELD	63	Herradura Reposado Tequila	750mL	

In [5]: `purchases = pd.read_sql_query('SELECT * FROM purchases WHERE VendorNum = purchases')`

Out[5]:

	InventoryId	Store	Brand	Description	Size	VendorNumber	Vendor
0	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AME VI BEV
1	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
2	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AME VI BEV
3	38_GOULCREST_5215	38	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
4	59_CLAETHORPES_5215	59	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
...
2187	81_PEMBROKE_5215	81	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
2188	62_KILMARNOCK_5255	62	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AME VI BEV
2189	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
2190	6_GOULCREST_5215	6	5215	TGI Fridays Long Island Iced	1.75L	4466	AME VI BEV
2191	35_HALIVAARA_5255	35	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AME VI BEV

2192 rows × 16 columns



In [6]: `purchase_prices = pd.read_sql_query('SELECT * FROM purchase_prices WHERE VendorNum = purchase_prices')`

Out[6]:

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	VendorNum
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	



```
In [7]: ► vendor_invoice = pd.read_sql_query('SELECT * FROM vendor_invoice WHERE  
vendor_invoice
```

Out[7]:

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
0	4466	AMERICAN VINTAGE BEVERAGE	2024-01-07	8137	2023- 12-22	2024- 02-21	15
1	4466	AMERICAN VINTAGE BEVERAGE	2024-01-19	8207	2023- 12-27	2024- 02-26	335
2	4466	AMERICAN VINTAGE BEVERAGE	2024-01-18	8307	2024- 01-03	2024- 02-18	41
3	4466	AMERICAN VINTAGE BEVERAGE	2024-01-27	8469	2024- 01-14	2024- 03-11	72
4	4466	AMERICAN VINTAGE BEVERAGE	2024-02-04	8532	2024- 01-19	2024- 03-15	79
5	4466	AMERICAN VINTAGE BEVERAGE	2024-02-09	8604	2024- 01-24	2024- 03-15	347
6	4466	AMERICAN VINTAGE BEVERAGE	2024-02-17	8793	2024- 02-05	2024- 04-02	72
7	4466	AMERICAN VINTAGE BEVERAGE	2024-03-01	8892	2024- 02-12	2024- 03-28	117
8	4466	AMERICAN VINTAGE BEVERAGE	2024-03-07	8995	2024- 02-19	2024- 04-02	129
9	4466	AMERICAN VINTAGE BEVERAGE	2024-03-12	9033	2024- 02-22	2024- 04-16	147
10	4466	AMERICAN VINTAGE BEVERAGE	2024-03-16	9180	2024- 03-03	2024- 04-19	211
11	4466	AMERICAN VINTAGE BEVERAGE	2024-03-23	9244	2024- 03-08	2024- 04-21	161
12	4466	AMERICAN VINTAGE BEVERAGE	2024-03-31	9371	2024- 03-17	2024- 05-13	176
13	4466	AMERICAN VINTAGE BEVERAGE	2024-04-09	9491	2024- 03-24	2024- 05-08	215
14	4466	AMERICAN VINTAGE BEVERAGE	2024-04-17	9583	2024- 03-31	2024- 05-12	110
15	4466	AMERICAN VINTAGE BEVERAGE	2024-04-20	9639	2024- 04-04	2024- 06-04	515
16	4466	AMERICAN VINTAGE BEVERAGE	2024-04-29	9800	2024- 04-15	2024- 06-07	275
17	4466	AMERICAN VINTAGE BEVERAGE	2024-05-09	9886	2024- 04-21	2024- 06-12	312

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
18	4466	AMERICAN VINTAGE BEVERAGE	2024-05-14	9999	2024- 04-29	2024- 06-26	310
19	4466	AMERICAN VINTAGE BEVERAGE	2024-05-16	10095	2024- 05-06	2024- 06-27	215
20	4466	AMERICAN VINTAGE BEVERAGE	2024-05-28	10169	2024- 05-11	2024- 07-04	327
21	4466	AMERICAN VINTAGE BEVERAGE	2024-06-04	10257	2024- 05-17	2024- 07-08	376
22	4466	AMERICAN VINTAGE BEVERAGE	2024-06-12	10346	2024- 05-23	2024- 07-20	640
23	4466	AMERICAN VINTAGE BEVERAGE	2024-06-17	10445	2024- 05-30	2024- 07-19	288
24	4466	AMERICAN VINTAGE BEVERAGE	2024-06-22	10600	2024- 06-09	2024- 08-01	308
25	4466	AMERICAN VINTAGE BEVERAGE	2024-06-29	10695	2024- 06-16	2024- 08-12	143
26	4466	AMERICAN VINTAGE BEVERAGE	2024-07-09	10777	2024- 06-22	2024- 08-15	18
27	4466	AMERICAN VINTAGE BEVERAGE	2024-07-08	10836	2024- 06-25	2024- 08-14	8
28	4466	AMERICAN VINTAGE BEVERAGE	2024-07-11	10969	2024- 06-29	2024- 08-18	94
29	4466	AMERICAN VINTAGE BEVERAGE	2024-07-19	11085	2024- 07-06	2024- 09-04	601
30	4466	AMERICAN VINTAGE BEVERAGE	2024-07-26	11187	2024- 07-14	2024- 08-26	1535
31	4466	AMERICAN VINTAGE BEVERAGE	2024-08-03	11244	2024- 07-18	2024- 09-04	266
32	4466	AMERICAN VINTAGE BEVERAGE	2024-08-11	11362	2024- 07-26	2024- 09-24	206
33	4466	AMERICAN VINTAGE BEVERAGE	2024-08-18	11489	2024- 08-04	2024- 09-16	768
34	4466	AMERICAN VINTAGE BEVERAGE	2024-08-24	11540	2024- 08-08	2024- 10-02	1207
35	4466	AMERICAN VINTAGE BEVERAGE	2024-09-02	11716	2024- 08-19	2024- 09-29	433

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
36	4466	AMERICAN VINTAGE BEVERAGE	2024-09-12	11771	2024-08-23	2024-10-11	370
37	4466	AMERICAN VINTAGE BEVERAGE	2024-09-20	11901	2024-09-01	2024-10-30	358
38	4466	AMERICAN VINTAGE BEVERAGE	2024-09-25	11993	2024-09-07	2024-10-23	233
39	4466	AMERICAN VINTAGE BEVERAGE	2024-10-01	12125	2024-09-16	2024-11-07	284
40	4466	AMERICAN VINTAGE BEVERAGE	2024-10-08	12235	2024-09-23	2024-11-20	258
41	4466	AMERICAN VINTAGE BEVERAGE	2024-10-09	12253	2024-09-23	2024-11-14	1
42	4466	AMERICAN VINTAGE BEVERAGE	2024-10-12	12321	2024-09-26	2024-11-19	172
43	4466	AMERICAN VINTAGE BEVERAGE	2024-10-20	12466	2024-10-05	2024-11-26	280
44	4466	AMERICAN VINTAGE BEVERAGE	2024-10-27	12515	2024-10-09	2024-11-30	178
45	4466	AMERICAN VINTAGE BEVERAGE	2024-11-07	12702	2024-10-21	2024-12-11	183
46	4466	AMERICAN VINTAGE BEVERAGE	2024-11-12	12752	2024-10-25	2024-12-11	216
47	4466	AMERICAN VINTAGE BEVERAGE	2024-11-20	12828	2024-10-30	2024-12-18	262
48	4466	AMERICAN VINTAGE BEVERAGE	2024-11-27	12929	2024-11-06	2025-01-04	270
49	4466	AMERICAN VINTAGE BEVERAGE	2024-11-28	13092	2024-11-16	2024-12-30	209
50	4466	AMERICAN VINTAGE BEVERAGE	2024-12-06	13134	2024-11-20	2025-01-18	305
51	4466	AMERICAN VINTAGE BEVERAGE	2024-12-16	13254	2024-11-28	2025-01-13	262
52	4466	AMERICAN VINTAGE BEVERAGE	2024-12-26	13432	2024-12-09	2025-01-27	231
53	4466	AMERICAN VINTAGE BEVERAGE	2024-12-30	13483	2024-12-13	2025-02-11	221

	VendorNumber	VendorName	InvoiceDate	PONumber	PODate	PayDate	Quantity
54	4466	AMERICAN VINTAGE BEVERAGE	2025-01-09	13627	2024- 12-22	2025- 02-05	413

```
In [8]: sales = pd.read_sql_query('SELECT * FROM sales WHERE VendorNo = 4466')
sales
```

Out[8]:

	InventoryId	Store	Brand	Description	Size	SalesQuantity	SalesD
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	
...
9448	9_BLACKPOOL_5215	9	5215	TGI Fridays Long Island Iced	1.75L	1	
9449	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
9450	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
9451	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	
9452	9_BLACKPOOL_5255	9	5255	TGI Fridays Ultimte Mudslide	1.75L	1	

9453 rows × 14 columns




```
In [9]: purchases.groupby(['Brand', 'PurchasePrice'])[['Quantity', 'Dollars']].s
```

```
Out[9]:
```

		Quantity	Dollars
Brand	PurchasePrice		
3140	11.19	4640	51921.60
5215	9.41	4923	46325.43
5255	9.35	6215	58110.25

```
In [10]: purchase_prices
```

```
Out[10]:
```

	Brand	Description	Price	Size	Volume	Classification	PurchasePrice	VendorN
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	

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

```
Out[11]: 55
```

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

```
Out[12]: (55, 10)
```

```
In [13]: sales.groupby('Brand')[['SalesDollars', 'SalesPrice', 'SalesQuantity']].
```

```
Out[13]:
```

	SalesDollars	SalesPrice	SalesQuantity
Brand			
3140	50531.10	30071.85	3890
5215	60416.49	41542.02	4651
5255	79187.04	51180.60	6096

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

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

The vendor_invoice table aggregates data from the purchases table, summarizing quantity and dollar amounts, along with an additional column for freight.

The purchase_price column is derived from the purchases_prices table, which provides product-wise actual and purchase prices. The combination of vendor and brand is unique in this table.

As the data that we need for analysis is distributed in different tables, we need to create a summary table containing:

-> purchase transactions made by vendors -> sales transaction data -> freight costs for each vendor -> actual product prices from vendors

```
In [14]: ▶ vendor_invoice.columns
```

```
Out[14]: Index(['VendorNumber', 'VendorName', 'InvoiceDate', 'PONumber', 'PODate',  
              'PayDate', 'Quantity', 'Dollars', 'Freight', 'Approval'],  
              dtype='object')
```

```
In [15]: ▶ sales.columns
```

```
Out[15]: Index(['InventoryId', 'Store', 'Brand', 'Description', 'Size', 'SalesQuantity',  
              'SalesDollars', 'SalesPrice', 'SalesDate', 'Volume', 'Classification',  
              'ExciseTax', 'VendorNo', 'VendorName'],  
              dtype='object')
```

This query generates a vendor-wise sales and purchase summary, which is valuable for:

Performance Optimization -> The query involves heavy joins and aggregations on large datasets like sales and purchases. -> Helps in analyzing sales, purchases and pricing for different vendors and brands. -> Storing the pre-aggregated results avoids repeated expensive computations. -> Future benefits of storing this data for faster dashboarding and reporting. -> Instead of running expensive queries each time, dashboards can fetch data quickly from vendor_sales_summary

```
In [16]: ▶ PurchaseSummary = pd.read_sql_query('''SELECT  
        p.VendorNumber,  
        p.VendorName,  
        p.Brand,  
        p.Description,  
        p.PurchasePrice,  
        pp.Price AS ActualPrice,  
        pp.Volume,  
        SUM(p.Quantity) AS TotalQuantity,  
        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''')
```

```
In [17]: SalesSummary = 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''',conn)
```

```
In [18]: FreightSummary = pd.read_sql_query('''SELECT VendorNumber, SUM(Freight
FROM vendor_invoice
GROUP BY VendorNumber''',conn)
```

```
In [19]: df1 = pd.merge(PurchaseSummary,SalesSummary, how = 'left',left_on = ['
◀────────────────────────────────────────────────────────────────────────────────▶▶
```

```
In [20]: vendor_sales_summary = pd.merge(df1,FreightSummary,how = 'left',left_o
```

```
In [21]: vendor_sales_summary.dtypes
```

```
Out[21]: VendorNumber      int64
VendorName      object
Brand           int64
Description     object
PurchasePrice   float64
ActualPrice     float64
Volume         object
TotalQuantity   int64
TotalPurchaseDollars float64
VendorNo        float64
TotalSalesDollars float64
TotalSalesPrice float64
TotalSalesQuantity float64
TotalExciseTax   float64
FreightCost     float64
dtype: object
```

```
In [22]: ▶ vendor_sales_summary.isnull().sum()
```

```
Out[22]: VendorNumber      0
VendorName      0
Brand          0
Description     0
PurchasePrice   0
ActualPrice     0
Volume         0
TotalQuantity   0
TotalPurchaseDollars  0
VendorNo       178
TotalSalesDollars 178
TotalSalesPrice 178
TotalSalesQuantity 178
TotalExciseTax   178
FreightCost      0
dtype: int64
```

```
In [23]: ▶ vendor_sales_summary['Volume'] = vendor_sales_summary['Volume'].astype
```

```
In [24]: ▶ vendor_sales_summary.fillna(0,inplace = True)
```

```
In [25]: ▶ vendor_sales_summary['VendorName'] = vendor_sales_summary['VendorName']
```

```
In [26]: ▶ # Checking DataTypes
vendor_sales_summary.dtypes
```

```
Out[26]: VendorNumber      int64
VendorName      object
Brand          int64
Description     object
PurchasePrice   float64
ActualPrice     float64
Volume         float64
TotalQuantity   int64
TotalPurchaseDollars  float64
VendorNo       float64
TotalSalesDollars  float64
TotalSalesPrice  float64
TotalSalesQuantity  float64
TotalExciseTax   float64
FreightCost     float64
dtype: object
```

```
In [27]: ▶ vendor_sales_summary['GrossProfit'] = vendor_sales_summary['TotalSales
```



```
In [28]: ▶ vendor_sales_summary['ProfitMargin'] = vendor_sales_summary['GrossProf
```

In [29]: ▶ `vendor_sales_summary['StockTurnover'] = vendor_sales_summary['TotalSal`

In [30]: ▶ `vendor_sales_summary['SalestoPurchaseRatio'] = vendor_sales_summary['T`



In [31]: ▶ `cursor = conn.cursor()`

```
In [32]: ▶ 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 DECIMAL(15,2),
TotalPurchaseQuantity INT,
TotalPurchaseDollars DECIMAL(15,2),
TotalSalesQuantity INT,
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)
);
""")
```


OperationalError Traceback (most recent call last)

Cell In[32], line 1

```
----> 1 cursor.execute("""CREATE TABLE vendor_sales_summary (
      2 VendorNumber INT,
      3 VendorName VARCHAR(100),
      4 Brand INT,
      5 Description VARCHAR(100),
      6 PurchasePrice DECIMAL(10,2),
      7 ActualPrice DECIMAL(10,2),
      8 Volume DECIMAL(15,2),
      9 TotalPurchaseQuantity INT,
     10 TotalPurchaseDollars DECIMAL(15,2),
     11 TotalSalesQuantity INT,
     12 TotalExciseTax DECIMAL(15,2),
     13 FreightCost DECIMAL(15,2),
     14 GrossProfit DECIMAL(15,2),
     15 ProfitMargin DECIMAL(15,2),
     16 StockTurnover DECIMAL(15,2),
     17 SalestoPurchaseRatio DECIMAL(15,2),
     18 PRIMARY KEY (VendorNumber,Brand)
     19 );
     20 """)
```

OperationalError: table vendor_sales_summary already exists

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

In [33]: `pd.read_sql_query('SELECT * FROM vendor_sales_summary', conn)`

Out[33]:

	VendorNumber	VendorName	Brand	Description	PurchasePrice	ActualPrice	V
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	Ch Lilian 09 Ladouys St Este	23.86	36.99	
1	2	IRA GOLDMAN AND WILLIAMS, LLP	90609	Flavor Essence Variety 5 Pak	17.00	24.99	
2	54	AAPER ALCOHOL & CHEMICAL CO	990	Ethyl Alcohol 200 Proof	105.07	134.49	
3	60	ADAMBA IMPORTS INTL INC	771	Bak's Krupnik Honey Liqueur	11.44	14.99	
4	60	ADAMBA IMPORTS INTL INC	3401	Vesica Vodka	11.10	14.99	
...	
10687	173357	TAMWORTH DISTILLING	2804	Camp Robber Whiskey	32.14	44.99	
10688	173357	TAMWORTH DISTILLING	3666	Art in the Age Chicory Root	18.79	24.99	
10689	173357	TAMWORTH DISTILLING	3848	Chicory Root Vodka	23.30	30.99	
10690	173357	TAMWORTH DISTILLING	3909	White Mountain Vodka	19.37	24.99	
10691	201359	FLAVOR ESSENCE INC	90609	Flavor Essence Variety 5 Pak	17.00	24.99	

10692 rows × 19 columns



In []: `vendor_sales_summary.to_sql('vendor_sales_summary', conn, if_exists = 'r'`

In []: