

Exploratory Data Analysis

June 6, 2025

1 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

```
[3]: import pandas as pd
import sqlite3
```

```
[4]: #creating database connection
conn = sqlite3.connect('inventory.db')
```

```
[5]: #checking tables present in database
tables = pd.read_sql_query("select name from sqlite_master where_
↪type='table'",conn)
tables
```

```
[5]:
```

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

```
[6]: for table in tables['name']:
print('-'*50, f'{table}', '-'*50)
print('Count of records:', pd.read_sql(f"select count (*) as count from_
↪{table}",conn)['count'].values[0])
display(pd.read_sql(f"select * from {table} limit 5",conn))
```

```
----- begin_inventory
-----
Count of records: 206529
```

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

	Size	onHand	Price	startDate
0	750mL	8	12.99	2024-01-01
1	750mL	7	10.99	2024-01-01
2	750mL	6	36.99	2024-01-01
3	750mL	3	38.99	2024-01-01
4	750mL	6	34.99	2024-01-01

----- end_inventory

Count of records: 224489

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

	Size	onHand	Price	endDate
0	750mL	11	12.99	2024-12-31
1	750mL	7	36.99	2024-12-31
2	750mL	7	38.99	2024-12-31
3	750mL	4	34.99	2024-12-31
4	750mL	7	14.99	2024-12-31

----- purchases

Count of records: 1048575

	InventoryId	Store	Brand	Description	Size \
0	69_MOUNTMEND_8412	69	8412	Tequila Ocho Plata Fresno	750mL
1	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L
2	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L
3	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L
4	76_DONCASTER_2034	76	2034	Glendalough Double Barrel	750mL

	VendorNumber	VendorName	PONumber	PODate \
0	105	ALTAMAR BRANDS LLC	8124	12/21/2023
1	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023
2	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023
3	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023
4	388	ATLANTIC IMPORTING COMPANY	8169	12/24/2023

	ReceivingDate	InvoiceDate	PayDate	PurchasePrice	Quantity	Dollars	\
0	1/2/2024	1/4/2024	2/16/2024	35.71	6	214.26	
1	1/1/2024	1/7/2024	2/21/2024	9.35	4	37.40	
2	1/2/2024	1/7/2024	2/21/2024	9.41	5	47.05	
3	1/1/2024	1/7/2024	2/21/2024	9.35	6	56.10	
4	1/2/2024	1/9/2024	2/16/2024	21.32	5	106.60	

	Classification
0	1
1	1
2	1
3	1
4	1

----- purchase_prices

Count of records: 12261

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

	PurchasePrice	VendorNumber	VendorName
0	9.28	8320	SHAW ROSS INT L IMP LTD
1	28.67	1128	BROWN-FORMAN CORP
2	30.46	1128	BROWN-FORMAN CORP
3	26.11	9165	ULTRA BEVERAGE COMPANY LLP
4	10.94	7245	PROXIMO SPIRITS INC.

----- sales

Count of records: 1048575

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

	SalesQuantity	SalesDollars	SalesPrice	SalesDate	Volume	Classification	\
0	1	16.49	16.49	1/1/2024	750	1	
1	2	32.98	16.49	1/2/2024	750	1	
2	1	16.49	16.49	1/3/2024	750	1	
3	1	14.49	14.49	1/8/2024	750	1	
4	2	69.98	34.99	1/9/2024	375	1	

	ExciseTax	VendorNo	VendorName
0	0.79	12546	JIM BEAM BRANDS COMPANY
1	1.57	12546	JIM BEAM BRANDS COMPANY
2	0.79	12546	JIM BEAM BRANDS COMPANY
3	0.79	12546	JIM BEAM BRANDS COMPANY
4	0.79	12546	JIM BEAM BRANDS COMPANY

----- vendor_invoice

Count of records: 5543

	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

----- vendor_sales_summary

Count of records: 8512

	VendorNumber	VendorName	Brand	Description \
0	1128	BROWN-FORMAN CORP	1233	Jack Daniels No 7 Black
1	3960	DIAGEO NORTH AMERICA INC	4261	Capt Morgan Spiced Rum
2	4425	MARTIGNETTI COMPANIES	3405	Tito's Handmade Vodka
3	17035	PERNOD RICARD USA	8068	Absolut 80 Proof
4	3960	DIAGEO NORTH AMERICA INC	3545	Ketel One Vodka

	PurchasePrice	ActualPrice	Volume	TotalPurchaseQuantity \
0	26.27	36.99	1750.0	60320
1	16.17	22.99	1750.0	96073
2	23.19	28.99	1750.0	62385
3	18.24	24.99	1750.0	75385
4	21.89	29.99	1750.0	58783

	TotalPurchaseDollars	TotalSalesQuantity	TotalSalesDollars \
0	1584606.40	9578.0	344712.22
1	1553500.41	20226.0	444810.74
2	1446708.15	9203.0	275162.97
3	1375022.40	11189.0	288135.11
4	1286759.87	11883.0	357759.17

	TotalSalesPrice	TotalExciseTax	FreightCost	GrossProfit	ProfitMargin	\
0	64889.97	17598.14	68601.68	-1239894.18	-359.689651	
1	43304.31	37163.76	257032.07	-1108689.67	-249.249753	
2	52289.50	16909.12	144929.24	-1171545.18	-425.764114	
3	48202.30	20557.97	123780.22	-1086887.29	-377.214457	
4	52774.51	21833.58	257032.07	-929000.70	-259.672086	

	StockTurnover	SalesToPurchaseRatio
0	0.158786	0.217538
1	0.210527	0.286328
2	0.147519	0.190199
3	0.148425	0.209549
4	0.202150	0.278031

```
[7]: purchases = pd.read_sql_query("select * from purchases where VendorNumber = 4466", conn)
purchases
```

```
[7]:
```

	InventoryId	Store	Brand	Description	Size	\
0	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	
1	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	
2	1_HARDERSFIELD_5255	1	5255	TGI Fridays Ultimte Mudslide	1.75L	
3	38_GOULCREST_5215	38	5215	TGI Fridays Long Island Iced	1.75L	
4	59_CLAETHORPES_5215	59	5215	TGI Fridays Long Island Iced	1.75L	
..	
860	34_PITMERDEN_5215	34	5215	TGI Fridays Long Island Iced	1.75L	
861	60_IRRAGIN_5215	60	5215	TGI Fridays Long Island Iced	1.75L	
862	60_IRRAGIN_5255	60	5255	TGI Fridays Ultimte Mudslide	1.75L	
863	60_IRRAGIN_3140	60	3140	TGI Fridays Orange Dream	1.75L	
864	34_PITMERDEN_5255	34	5255	TGI Fridays Ultimte Mudslide	1.75L	

	VendorNumber	VendorName	PONumber	PODate	\
0	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023	
1	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023	
2	4466	AMERICAN VINTAGE BEVERAGE	8137	12/22/2023	
3	4466	AMERICAN VINTAGE BEVERAGE	8207	12/27/2023	
4	4466	AMERICAN VINTAGE BEVERAGE	8207	12/27/2023	
..	
860	4466	AMERICAN VINTAGE BEVERAGE	10777	6/22/2024	
861	4466	AMERICAN VINTAGE BEVERAGE	10777	6/22/2024	
862	4466	AMERICAN VINTAGE BEVERAGE	10777	6/22/2024	
863	4466	AMERICAN VINTAGE BEVERAGE	10777	6/22/2024	
864	4466	AMERICAN VINTAGE BEVERAGE	10777	6/22/2024	

	ReceivingDate	InvoiceDate	PayDate	PurchasePrice	Quantity	Dollars	\
0	1/1/2024	1/7/2024	2/21/2024	9.35	4	37.40	
1	1/2/2024	1/7/2024	2/21/2024	9.41	5	47.05	

2	1/1/2024	1/7/2024	2/21/2024	9.35	6	56.10
3	1/7/2024	1/19/2024	2/26/2024	9.41	6	56.46
4	1/5/2024	1/19/2024	2/26/2024	9.41	6	56.46
..
860	6/28/2024	7/9/2024	8/15/2024	9.41	4	37.64
861	6/30/2024	7/9/2024	8/15/2024	9.41	5	47.05
862	6/30/2024	7/9/2024	8/15/2024	9.35	2	18.70
863	6/30/2024	7/9/2024	8/15/2024	11.19	6	67.14
864	6/28/2024	7/9/2024	8/15/2024	9.35	1	9.35

Classification	
0	1
1	1
2	1
3	1
4	1
..	...
860	1
861	1
862	1
863	1
864	1

[865 rows x 16 columns]

```
[8]: purchase_prices = pd.read_sql_query("select * from purchase_prices where
↳VendorNumber = 4466",conn)
purchase_prices
```

```
[8]:      Brand      Description  Price  Size Volume  Classification \
0   5215  TGI Fridays Long Island Iced  12.99  1750mL  1750          1
1   5255  TGI Fridays Ultimte Mudslide  12.99  1750mL  1750          1
2   3140    TGI Fridays Orange Dream  14.99  1750mL  1750          1
```

	PurchasePrice	VendorNumber	VendorName
0	9.41	4466	AMERICAN VINTAGE BEVERAGE
1	9.35	4466	AMERICAN VINTAGE BEVERAGE
2	11.19	4466	AMERICAN VINTAGE BEVERAGE

```
[9]: vendor_invoice = pd.read_sql_query("select * from vendor_invoice where
↳VendorNumber = 4466",conn)
vendor_invoice
```

```
[9]:      VendorNumber      VendorName  InvoiceDate  PONumber \
0          4466  AMERICAN VINTAGE BEVERAGE  2024-01-07    8137
1          4466  AMERICAN VINTAGE BEVERAGE  2024-01-19    8207
2          4466  AMERICAN VINTAGE BEVERAGE  2024-01-18    8307
```

3	4466	AMERICAN VINTAGE BEVERAGE	2024-01-27	8469
4	4466	AMERICAN VINTAGE BEVERAGE	2024-02-04	8532
5	4466	AMERICAN VINTAGE BEVERAGE	2024-02-09	8604
6	4466	AMERICAN VINTAGE BEVERAGE	2024-02-17	8793
7	4466	AMERICAN VINTAGE BEVERAGE	2024-03-01	8892
8	4466	AMERICAN VINTAGE BEVERAGE	2024-03-07	8995
9	4466	AMERICAN VINTAGE BEVERAGE	2024-03-12	9033
10	4466	AMERICAN VINTAGE BEVERAGE	2024-03-16	9180
11	4466	AMERICAN VINTAGE BEVERAGE	2024-03-23	9244
12	4466	AMERICAN VINTAGE BEVERAGE	2024-03-31	9371
13	4466	AMERICAN VINTAGE BEVERAGE	2024-04-09	9491
14	4466	AMERICAN VINTAGE BEVERAGE	2024-04-17	9583
15	4466	AMERICAN VINTAGE BEVERAGE	2024-04-20	9639
16	4466	AMERICAN VINTAGE BEVERAGE	2024-04-29	9800
17	4466	AMERICAN VINTAGE BEVERAGE	2024-05-09	9886
18	4466	AMERICAN VINTAGE BEVERAGE	2024-05-14	9999
19	4466	AMERICAN VINTAGE BEVERAGE	2024-05-16	10095
20	4466	AMERICAN VINTAGE BEVERAGE	2024-05-28	10169
21	4466	AMERICAN VINTAGE BEVERAGE	2024-06-04	10257
22	4466	AMERICAN VINTAGE BEVERAGE	2024-06-12	10346
23	4466	AMERICAN VINTAGE BEVERAGE	2024-06-17	10445
24	4466	AMERICAN VINTAGE BEVERAGE	2024-06-22	10600
25	4466	AMERICAN VINTAGE BEVERAGE	2024-06-29	10695
26	4466	AMERICAN VINTAGE BEVERAGE	2024-07-09	10777
27	4466	AMERICAN VINTAGE BEVERAGE	2024-07-08	10836
28	4466	AMERICAN VINTAGE BEVERAGE	2024-07-11	10969
29	4466	AMERICAN VINTAGE BEVERAGE	2024-07-19	11085
30	4466	AMERICAN VINTAGE BEVERAGE	2024-07-26	11187
31	4466	AMERICAN VINTAGE BEVERAGE	2024-08-03	11244
32	4466	AMERICAN VINTAGE BEVERAGE	2024-08-11	11362
33	4466	AMERICAN VINTAGE BEVERAGE	2024-08-18	11489
34	4466	AMERICAN VINTAGE BEVERAGE	2024-08-24	11540
35	4466	AMERICAN VINTAGE BEVERAGE	2024-09-02	11716
36	4466	AMERICAN VINTAGE BEVERAGE	2024-09-12	11771
37	4466	AMERICAN VINTAGE BEVERAGE	2024-09-20	11901
38	4466	AMERICAN VINTAGE BEVERAGE	2024-09-25	11993
39	4466	AMERICAN VINTAGE BEVERAGE	2024-10-01	12125
40	4466	AMERICAN VINTAGE BEVERAGE	2024-10-08	12235
41	4466	AMERICAN VINTAGE BEVERAGE	2024-10-09	12253
42	4466	AMERICAN VINTAGE BEVERAGE	2024-10-12	12321
43	4466	AMERICAN VINTAGE BEVERAGE	2024-10-20	12466
44	4466	AMERICAN VINTAGE BEVERAGE	2024-10-27	12515
45	4466	AMERICAN VINTAGE BEVERAGE	2024-11-07	12702
46	4466	AMERICAN VINTAGE BEVERAGE	2024-11-12	12752
47	4466	AMERICAN VINTAGE BEVERAGE	2024-11-20	12828
48	4466	AMERICAN VINTAGE BEVERAGE	2024-11-27	12929
49	4466	AMERICAN VINTAGE BEVERAGE	2024-11-28	13092

50	4466	AMERICAN VINTAGE BEVERAGE	2024-12-06	13134
51	4466	AMERICAN VINTAGE BEVERAGE	2024-12-16	13254
52	4466	AMERICAN VINTAGE BEVERAGE	2024-12-26	13432
53	4466	AMERICAN VINTAGE BEVERAGE	2024-12-30	13483
54	4466	AMERICAN VINTAGE BEVERAGE	2025-01-09	13627

	PODate	PayDate	Quantity	Dollars	Freight	Approval
0	2023-12-22	2024-02-21	15	140.55	8.57	None
1	2023-12-27	2024-02-26	335	3142.33	16.97	None
2	2024-01-03	2024-02-18	41	383.35	1.99	None
3	2024-01-14	2024-03-11	72	673.20	3.30	None
4	2024-01-19	2024-03-15	79	740.21	3.48	None
5	2024-01-24	2024-03-15	347	3261.37	17.61	None
6	2024-02-05	2024-04-02	72	675.36	3.17	None
7	2024-02-12	2024-03-28	117	1096.05	5.15	None
8	2024-02-19	2024-04-02	129	1209.27	5.44	None
9	2024-02-22	2024-04-16	147	1377.87	6.61	None
10	2024-03-03	2024-04-19	211	1979.33	9.50	None
11	2024-03-08	2024-04-21	161	1510.69	6.95	None
12	2024-03-17	2024-05-13	176	1649.20	8.91	None
13	2024-03-24	2024-05-08	215	2016.43	10.08	None
14	2024-03-31	2024-05-12	110	1035.10	5.69	None
15	2024-04-04	2024-06-04	515	5323.01	25.02	None
16	2024-04-15	2024-06-07	275	2775.01	13.60	None
17	2024-04-21	2024-06-12	312	3050.86	13.73	None
18	2024-04-29	2024-06-26	310	3117.82	14.97	None
19	2024-05-06	2024-06-27	215	2123.01	11.46	None
20	2024-05-11	2024-07-04	327	3298.55	15.83	None
21	2024-05-17	2024-07-08	376	3708.18	20.39	None
22	2024-05-23	2024-07-20	640	6458.38	29.71	None
23	2024-05-30	2024-07-19	288	2937.56	15.86	None
24	2024-06-09	2024-08-01	308	3213.28	17.03	None
25	2024-06-16	2024-08-12	143	1585.75	8.09	None
26	2024-06-22	2024-08-15	18	179.88	0.99	None
27	2024-06-25	2024-08-14	8	85.96	0.43	None
28	2024-06-29	2024-08-18	94	993.12	5.06	None
29	2024-07-06	2024-09-04	601	5883.09	31.18	None
30	2024-07-14	2024-08-26	1535	15406.69	77.03	None
31	2024-07-18	2024-09-04	266	2888.22	15.31	None
32	2024-07-26	2024-09-24	206	2171.64	11.08	None
33	2024-08-04	2024-09-16	768	7292.98	37.92	None
34	2024-08-08	2024-10-02	1207	12242.43	64.88	None
35	2024-08-19	2024-09-29	433	4334.43	20.37	None
36	2024-08-23	2024-10-11	370	3640.18	18.20	None
37	2024-09-01	2024-10-30	358	3626.74	16.32	None
38	2024-09-07	2024-10-23	233	2264.85	12.23	None
39	2024-09-16	2024-11-07	284	2813.46	15.47	None

40	2024-09-23	2024-11-20	258	2592.08	11.92	None
41	2024-09-23	2024-11-14	1	11.19	0.05	None
42	2024-09-26	2024-11-19	172	1685.98	7.59	None
43	2024-10-05	2024-11-26	280	2798.60	14.55	None
44	2024-10-09	2024-11-30	178	1763.28	8.64	None
45	2024-10-21	2024-12-11	183	1812.77	8.70	None
46	2024-10-25	2024-12-11	216	2156.16	10.57	None
47	2024-10-30	2024-12-18	262	2597.36	12.99	None
48	2024-11-06	2025-01-04	270	2622.32	12.85	None
49	2024-11-16	2024-12-30	209	2036.17	11.20	None
50	2024-11-20	2025-01-18	305	3018.89	15.70	None
51	2024-11-28	2025-01-13	262	2530.36	11.39	None
52	2024-12-09	2025-01-27	231	2263.43	11.32	None
53	2024-12-13	2025-02-11	221	2178.27	11.33	None
54	2024-12-22	2025-02-05	413	3985.03	19.53	None

```
[10]: sales = pd.read_sql_query("select * from sales where VendorNo = 4466",conn)
sales
```

```
[10]:
```

	InventoryId	Store	Brand	Description	Size	\
0	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	
1	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	
2	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	
3	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	
4	1_HARDERSFIELD_5215	1	5215	TGI Fridays Long Island Iced	1.75L	
..	
426	15_WANBORNE_5215	15	5215	TGI Fridays Long Island Iced	1.75L	
427	15_WANBORNE_5255	15	5255	TGI Fridays Ultimte Mudslide	1.75L	
428	15_WANBORNE_5255	15	5255	TGI Fridays Ultimte Mudslide	1.75L	
429	15_WANBORNE_5255	15	5255	TGI Fridays Ultimte Mudslide	1.75L	
430	15_WANBORNE_5255	15	5255	TGI Fridays Ultimte Mudslide	1.75L	

	SalesQuantity	SalesDollars	SalesPrice	SalesDate	Volume	\
0	1	12.99	12.99	1/9/2024	1750	
1	1	12.99	12.99	1/12/2024	1750	
2	1	12.99	12.99	1/15/2024	1750	
3	1	12.99	12.99	1/21/2024	1750	
4	1	12.99	12.99	1/23/2024	1750	
..	
426	1	12.99	12.99	2/23/2024	1750	
427	1	12.99	12.99	2/3/2024	1750	
428	1	12.99	12.99	2/23/2024	1750	
429	1	12.99	12.99	2/24/2024	1750	
430	2	25.98	12.99	2/27/2024	1750	

	Classification	ExciseTax	VendorNo	VendorName
0	1	1.84	4466	AMERICAN VINTAGE BEVERAGE

1	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
2	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
3	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
4	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
..
426	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
427	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
428	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
429	1	1.84	4466	AMERICAN VINTAGE BEVERAGE
430	1	3.67	4466	AMERICAN VINTAGE BEVERAGE

[431 rows x 14 columns]

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

```
[11]:
```

		Quantity	Dollars
Brand	PurchasePrice		
3140	11.19	1552	17366.88
5215	9.41	2267	21332.47
5255	9.35	2135	19962.25

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

```
[12]:
```

	SalesDollars	SalesPrice	SalesQuantity
Brand			
5215	3897.00	2416.14	300
5255	4273.71	3182.55	329

- 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 purchase price column is derived from the purchase_prices table, which provides product-wise actual and purchase prices. The combination of vendor and brand is unique in this table.
- The vendor_invoice table aggregates data from the purchases table, summarizing quantity and dollar amounts, along With an additional column for freight. This table maintains uniqueness based on vendor and PO number.
- The sales table captures actual sales transactions, detailing the brands purchased by vendors, the quantity sold, the selling price, and the revenue earned.

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

```
[13]: freight_summary = pd.read_sql_query("""select VendorNumber, sum(Freight) as FreightCose
from vendor_invoice
group by VendorNumber""",conn)
```

```
[14]: freight_summary
```

```
[14]:
```

	VendorNumber	FreightCose
0	2	27.08
1	54	0.48
2	60	367.52
3	105	62.39
4	200	6.19
..
121	98450	856.02
122	99166	130.09
123	172662	178.34
124	173357	202.50
125	201359	0.09

[126 rows x 2 columns]

```
[15]: pd.read_sql_query("""select
p.VendorNumber,
p.VendorName,
p.Brand,
p.PurchasePrice,
pp.Volume,
pp.Price as ActualPrice,
sum(Quantity) as TotalPurchaseQuantity,
sum(Dollars) as TotalPurchaseDollars
from purchases p
join purchase_prices pp
on p.brand = pp.brand
where p.PurchasePrice > 0
group by p.VendorName = pp.VendorName, p.Brand
order by TotalPurchaseDollars""",conn)
```

```
[15]:
```

	VendorNumber	VendorName	Brand	PurchasePrice	Volume	\
0	7245	PROXIMO SPIRITS INC.	3065	0.71	50	
1	3960	DIAGEO NORTH AMERICA INC	3775	0.73	50	
2	9815	WINE GROUP INC	22407	2.25	750	
3	8004	SAZERAC CO INC	5683	0.39	50	
4	9815	WINE GROUP INC	8527	1.32	750	
...	
8507	3960	DIAGEO NORTH AMERICA INC	3545	21.89	1750	
8508	17035	PERNOD RICARD USA	8068	18.24	1750	

	ActualPrice	TotalPurchaseQuantity	TotalPurchaseDollars
0	0.99	1	0.71
1	0.99	1	0.73
2	3.29	1	2.25
3	0.49	6	2.34
4	4.99	2	2.64
...
8507	29.99	58783	1286759.87
8508	24.99	75385	1375022.40
8509	28.99	62385	1446708.15
8510	22.99	96073	1553500.41
8511	36.99	60320	1584606.40

```
[16]: 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)
```

	TotalExciseTax
0	0.10
1	0.05
2	0.05

3	0.05
4	0.10
...	...
7653	16909.12
7654	20557.97
7655	17598.14
7656	21833.58
7657	37163.76

[7658 rows x 6 columns]

```
[17]: 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)

```

[18]: vendor_sales_summary

```

[18]:
      VendorNumber      VendorName  Brand  \
0          1128  BROWN-FORMAN CORP      1233
1          3960  DIAGEO NORTH AMERICA INC  4261
2          4425      MARTIGNETTI COMPANIES  3405
3          17035  PERNOD RICARD USA      8068
4           3960  DIAGEO NORTH AMERICA INC  3545
...          ...          ...    ...
8507          9815  WINE GROUP INC      8527
8508          8004  SAZERAC CO INC      5683
8509          9815  WINE GROUP INC     22407
8510          3960  DIAGEO NORTH AMERICA INC  3775
8511          7245  PROXIMO SPIRITS INC.    3065

      Description  PurchasePrice  ActualPrice  Volume  \
0  Jack Daniels No 7 Black      26.27      36.99    1750
1  Capt Morgan Spiced Rum      16.17      22.99    1750
2  Tito's Handmade Vodka      23.19      28.99    1750

```

3	Absolut 80 Proof	18.24	24.99	1750
4	Ketel One Vodka	21.89	29.99	1750
...
8507	Concannon Glen Ellen Wh Zin	1.32	4.99	750
8508	Dr McGillicuddy's Apple Pie	0.39	0.49	50
8509	Three Wishes Chard	2.25	3.29	750
8510	Smirnoff Sorbet Pine/Coconut	0.73	0.99	50
8511	Three Olives Grape Vodka	0.71	0.99	50

	TotalPurchaseQuantity	TotalPurchaseDollars	TotalSalesQuantity	\
0	60320	1584606.40	9578.0	
1	96073	1553500.41	20226.0	
2	62385	1446708.15	9203.0	
3	75385	1375022.40	11189.0	
4	58783	1286759.87	11883.0	
...	
8507	2	2.64	3.0	
8508	6	2.34	128.0	
8509	1	2.25	1.0	
8510	1	0.73	NaN	
8511	1	0.71	81.0	

	TotalSalesDollars	TotalSalesPrice	TotalExciseTax	FreightCost
0	344712.22	64889.97	17598.14	68601.68
1	444810.74	43304.31	37163.76	257032.07
2	275162.97	52289.50	16909.12	144929.24
3	288135.11	48202.30	20557.97	123780.22
4	357759.17	52774.51	21833.58	257032.07
...
8507	5.97	5.97	0.33	27100.41
8508	62.72	0.98	6.72	50293.62
8509	3.29	3.29	0.11	27100.41
8510	NaN	NaN	NaN	257032.07
8511	80.19	29.70	4.21	38994.78

[8512 rows x 14 columns]

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.
- 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 quickly from vendor_sales_summary.

```
[19]: vendor_sales_summary.dtypes
```

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

```
[20]: vendor_sales_summary.isnull().sum()
```

```
[20]: VendorNumber      0
      VendorName       0
      Brand            0
      Description      0
      PurchasePrice    0
      ActualPrice      0
      Volume           0
      TotalPurchaseQuantity  0
      TotalPurchaseDollars  0
      TotalSalesQuantity  1633
      TotalSalesDollars    1633
      TotalSalesPrice     1633
      TotalExciseTax      1633
      FreightCost        0
      dtype: int64
```

```
[21]: vendor_sales_summary['VendorName'].unique()
```

```
[21]: array(['BROWN-FORMAN CORP', 'DIAGEO NORTH AMERICA INC',
      'MARTIGNETTI COMPANIES', 'PERNOD RICARD USA',
      'BACARDI USA INC', 'JIM BEAM BRANDS COMPANY',
      'ULTRA BEVERAGE COMPANY LLP', 'PROXIMO SPIRITS INC.',
      'MAJESTIC FINE WINES', 'STOLI GROUP,(USA) LLC',
      'CAMPARI AMERICA', 'SAZERAC CO INC',
      'MOET HENNESSY USA INC', 'M S WALKER INC',
      'CONSTELLATION BRANDS INC', 'SAZERAC NORTH AMERICA INC.',
      'REMY COINTREAU USA INC', 'SIDNEY FRANK IMPORTING CO',
```


'WILLIAM GRANT & SONS INC	',	'PALM BAY INTERNATIONAL INC	',
'E & J GALLO WINERY	',	'HEAVEN HILL DISTILLERIES	',
'CASTLE BRANDS CORP.	',	'SOUTHERN WINE & SPIRITS NE	',
'DISARONNO INTERNATIONAL LLC'	,	'EDRINGTON AMERICAS	',
'TRINCHERO FAMILY ESTATES	',	'STE MICHELLE WINE ESTATES	',
'WINE GROUP INC	',	'PERFECTA WINES	',
'LUXCO INC	',	'SHAW ROSS INT L IMP LTD	',
'TREASURY WINE ESTATES	',	'MHW LTD	',
'PHILLIPS PRODUCTS CO.	',	'KOBRA BRAND CORPORATION	',
'BANFI PRODUCTS CORP	',	'STATE WINE & SPIRITS	',
'DIAGEO CHATEAU ESTATE WINES'	,	'CALEDONIA SPIRITS INC	',
'PINE STATE TRADING CO	',	'VINEYARD BRANDS INC	',
'FABRIZIA SPIRITS LLC	',	'LATITUDE BEVERAGE COMPANY	',
'DELICATO VINEYARDS INC	',	'OLE SMOKY DISTILLERY LLC	',
'Serralles Usa LLC	',	'DUGGANS DISTILLED PRODUCTS	',
'KLIN SPIRITS LLC	',	'LAIRD & CO	',
'SEA HAGG DISTILLERY LLC	',	'Dunn Wine Brokers	',
'WESTERN SPIRITS BEVERAGE CO'	,	'CHARLES JACQUIN ET CIE INC	',
'FLAG HILL WINERY & VINEYARD'	,	'FREDERICK WILDMAN & SONS	',
'MCCORMICK DISTILLING CO	',	'MARSALLE COMPANY	',
'MANGO BOTTLING INC	',	'SWEET BABY VINEYARD	',
'AMERICAN VINTAGE BEVERAGE	',	'LABELLE VYDS AND WINERY	',
'VRANKEN AMERICA	',	'PREMIUM PORT WINES, INC.	',
'ATLANTIC IMPORTING COMPANY	',	'PREMIER DISTRIBUTORS	',
'Russian Standard Vodka	',	'NICHE W & S	',
'WEIN BAUER INC	',	'BULLY BOY DISTILLERS	',
'VINILANDIA USA	',	'TAKARA SAKE USA INC	',
'PSP WINES	',	'ADAMBA IMPORTS INTL INC	',
'TY KU LLC	',	'MARTIGNETTI COMPANIES	',
'ZORVINO VINEYARDS	',	'SMOKY QUARTZ DISTILLERY LLC'	,
'CRUSH WINES	',	'TAMWORTH DISTILLING	',
'MOONLIGHT MEADERY	',	'ALISA CARR BEVERAGES	',
'POVERTY LANE ORCHARDS	',	'SEA BREEZE CELLARS LLC	',
'VINEXTRA INC	',	'SWEETWATER FARM	',
'FORTUNE WINE BROKERS LLC	',	'DJINN SPIRITS LLC	',
'JEWELL TOWNE VINEYARDS	',		
'IRA GOLDMAN AND WILLIAMS, LLP	',		
'Circa Wines	',	'HOOD RIVER DISTILLERS, Inc.'	',
'INCREDIBREW INC	',	'ALTAMAR BRANDS LLC	',
'CANDIA VINEYARDS	',	'SURVILLE ENTERPRISES CORP	',
'TALL SHIP DISTILLERY LLC	',	'BLACK PRINCE DISTILLERY INC'	,
'STELLAR IMPORTING CO LLC	',	'VINEDREA WINES LLC	',
'FULCHINO VINEYARD INC	',	'STAR INDUSTRIES INC.	',
'WALPOLE MTN VIEW WINERY	',	'HAUNTING WHISPER VYDS	',
'R.P.IMPORTS INC	',	'THE PIERPONT GROUP LLC	',
'THE IMPORTED GRAPE LLC	',	'BLACK ROCK SPIRITS LLC	',
'CENTEUR IMPORTS LLC	',	'BRONCO WINE COMPANY	',

```
'PARK STREET IMPORTS LLC      ', 'APPOLO VINEYARDS LLC      ',
'HIGHLAND WINE MERCHANTS LLC', 'MILTONS DISTRIBUTING CO   ',
'SILVER MOUNTAIN CIDERS      ', 'FANTASY FINE WINES CORP   ',
'CAPSTONE INTERNATIONAL      ', 'UNCORKED                  ', dtype=object)
```

```
[22]: vendor_sales_summary['Description'].unique()
```

```
[22]: array(['Jack Daniels No 7 Black', 'Capt Morgan Spiced Rum',
        "Tito's Handmade Vodka", ..., 'Smirnoff Light Strawberry',
        'Concannon Glen Ellen Wh Zin', 'Three Wishes Chard'], dtype=object)
```

```
[23]: vendor_sales_summary['Volume'] = vendor_sales_summary['Volume'].
        ↪astype('float64')
```

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

```
[25]: vendor_sales_summary['VendorName'] = vendor_sales_summary['VendorName'].str.
        ↪strip()
```

```
[26]: vendor_sales_summary['GrossProfit'] = vendor_sales_summary['TotalSalesDollars']_
        ↪- vendor_sales_summary['TotalPurchaseDollars']
```

```
[27]: vendor_sales_summary['ProfitMargin'] = (vendor_sales_summary['GrossProfit'] /_
        ↪vendor_sales_summary['TotalSalesDollars'])*100
```

```
[28]: vendor_sales_summary['StockTurnover'] =_
        ↪vendor_sales_summary['TotalSalesQuantity']/
        ↪vendor_sales_summary['TotalPurchaseQuantity']
```

```
[29]: vendor_sales_summary['SalesPurchaseRatio'] =_
        ↪vendor_sales_summary['TotalSalesDollars']/
        ↪vendor_sales_summary['TotalPurchaseDollars']
```

```
[ ]:
```

```
[30]: cursor = conn.cursor()
```

```
[31]: 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(10,2),
        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)
);
"""

```

```

-----
OperationalError                                Traceback (most recent call last)
Cell In[31], 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(10,2),
      9     TotalPurchaseQuantity INT,
     10     TotalPurchaseDollars DECIMAL(15,2),
     11     TotalSalesQuantity INT,
     12     TotalSalesDollars DECIMAL(15,2),
     13     TotalSalesPrice DECIMAL(15,2),
     14     TotalExciseTax DECIMAL(15,2),
     15     FreightCost DECIMAL(15,2),
     16     GrossProfit DECIMAL(15,2),
     17     ProfitMargin DECIMAL(15,2),
     18     StockTurnover DECIMAL(15,2),
     19     SalesToPurchaseRatio DECIMAL(15,2),
     20     PRIMARY KEY (VendorNumber, Brand)
     21 );
     22 """)

OperationalError: table vendor_sales_summary already exists

```

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

```
[ ]: vendor_sales_summary.to_sql('vendor_sales_summary',conn, if_exists = 'replace',
    ↪index = False)
```

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

```
[ ]: #Creating vendor_summary table

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

log_path = 'Logs'

# Ensure the log directory exists
os.makedirs(log_path, exist_ok=True)

# Clear existing logging handlers
for handler in logging.root.handlers[:]:
    logging.root.removeHandler(handler)

# Configure logging to log only to file
from logging import FileHandler

# Clear existing handlers
for handler in logging.root.handlers[:]:
    logging.root.removeHandler(handler)

# Set up file handler with UTF-8 encoding
file_handler = FileHandler(os.path.join(log_path, 'get_vendor_summary.txt'),
    mode='w', encoding='utf-8')

# Configure logging
logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s - %(levelname)s - %(message)s',
    handlers=[file_handler]
)

def create_vendor_summary(conn):
    """This function merges tables to get the overall vendor summary"""
    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 pd.read_sql_query(query,conn)

def clean_data(df):
    """This function cleans and enhances the vendor sales data."""
    df['Volume'] = df['Volume'].astype(float)
    df.fillna(0, inplace=True)
    df['VendorName'] = df['VendorName'].str.strip()
    df['Description'] = df['Description'].str.strip()

    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().to_string())

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

    logging.info(clean_df.head().to_string())

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

```

```
[ ]: vendor_sales_summary
```

```
[ ]: import sqlite3
import pandas as pd
import logging

# Configure logging
```

```

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - \n
↳ %(message)s')

# Connect to SQLite database (File 1)
try:
    conn = sqlite3.connect("inventory.db") # Adjust path to your database file
    logging.info('Database connection established.')
except sqlite3.Error as e:
    logging.error(f"Database connection error: {str(e)}")
    raise

# Load the sales summary table
try:
    sales_summary_df = pd.read_sql_query("SELECT * FROM vendor_sales_summary", \n
↳ conn)
    logging.info('Sales summary table loaded successfully.')
    print(sales_summary_df.head())
except Exception as e:
    logging.error(f"Error loading table: {str(e)}")
    raise
finally:
    conn.close()
    logging.info('Database connection closed.')

```

[40]: *#exporting vendor_sales_summary*

```

df_to_export = vendor_sales_summary

# Export to CSV
output_filename = "vendor_sales_summary.csv"
df_to_export.to_csv(output_filename, index=False) # Corrected line

print(f"Data successfully exported to {output_filename}")

```

Data successfully exported to vendor_sales_summary.csv

[]:

[]: