Data-Driven Strategies for optimizing E-commerce Sales Performance

Presented to

Digital Egypt Pioneers Initiative (DEPI)

Report by

Abdelrahman Hassan Mahmoud
Amany Alaraby
Mahmoud Mohammed Shoshan
Maysan Salah Elghanam

Table of Contents

Roles and Responsibilities

THE PROJECT WAS CARRIED OUT BY A TEAM OF FOUR MEMBERS, WITH TASKS DIVIDED TO LEVERAGE EACH INDIVIDUAL'S STRENGTHS AND ENSURE EFFICIENT COMPLETION OF THE OBJECTIVES.

- Team Leader: Abdelrahman Hassan Abdelrahman led the project, ensuring the team was aligned with the goals and managed overall coordination. He also contributed by working on SQL tasks alongside Mahmoud.
- SQL and Python Development: Abdelrahman Hassan and Mahmoud Shoshan. Abdelrahman and Mahmoud were responsible for all SQL and Python tasks. They worked together on data management, database querying, and implementing necessary Python scripts to support the project's data needs.
- Power BI and Documentation: Maysan Salah and Amany Alaraby Maysan and Amany handled the data visualization and reporting aspects of the project using Power BI. Together, they also worked on creating comprehensive documentation to capture the project's progress, findings, and deliverables.

Data source

The data source for our E-COMMERCE database originates from Kaggle, a prominent online community for data scientists and machine learning practitioners. Kaggle provides a vast collection of datasets contributed by data enthusiasts from around the world.

In our pursuit of creating a comprehensive E-COMMERCE database, we have leveraged a dataset obtained from Kaggle to ensure the accuracy and relevance of our information.

Dataset components

THE DATASET CONSISTS OF 6 CSV FILES EACH FILE HAS A TABLE OF THE DATA: LET'S BEGIN BY UNDERSTANDING THE TABLES AND THEIR RELATIONSHIPS:

1. customers table: It contains information about customers including their personal details like. The table has five columns: "id", "join date", “city_id” (foreign key linking to the city_id table),” gender_id” (foreign key linking to the gender_id table).

2. genders table: It lists different types of genders. The table has two columns: ("id" and "gender_type").

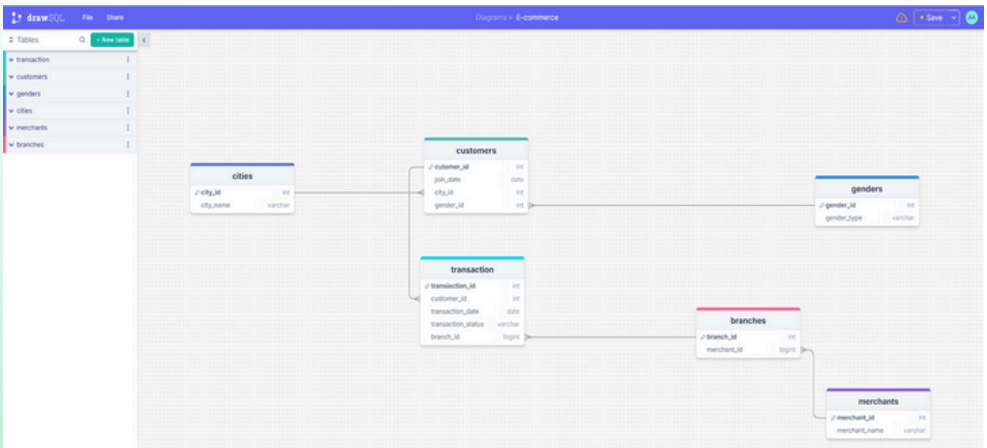
3. cities table: This table stores information about customer’s cities. It has two columns (“city_id”, “city”)

4. merchants table: It holds data about the merchants ("id", “merchant_name”)

5). branches table: This table stores the branch info like (“branch_id” , “merchant_id”(foreign key linking to the merchant_id table)

6. transaction: It stores the details of transactions and it is the fact table of data . it is contain all details like (“id”, “customer_id ” (foreign key linking to the customer_id table), “date” of transaction , “burn_date” if that burned, “status “ ,”branch_id” (foreign key linking to the branch_id table)

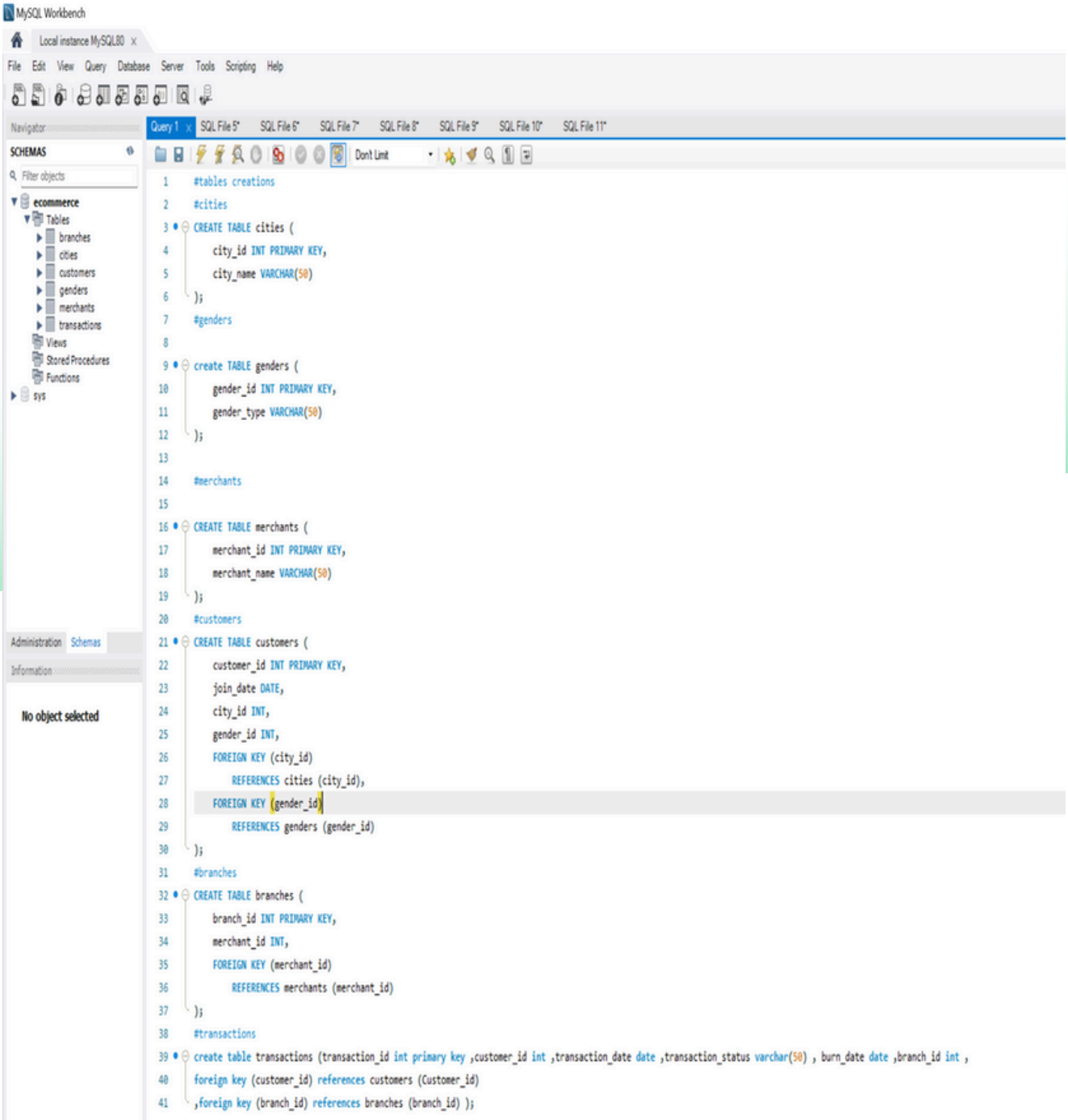
Relational database: In order to understand the relationship between our tables we created schema using drawsql.app.



https://drawsql.app/teams/depi-3/diagrams/e-commerce

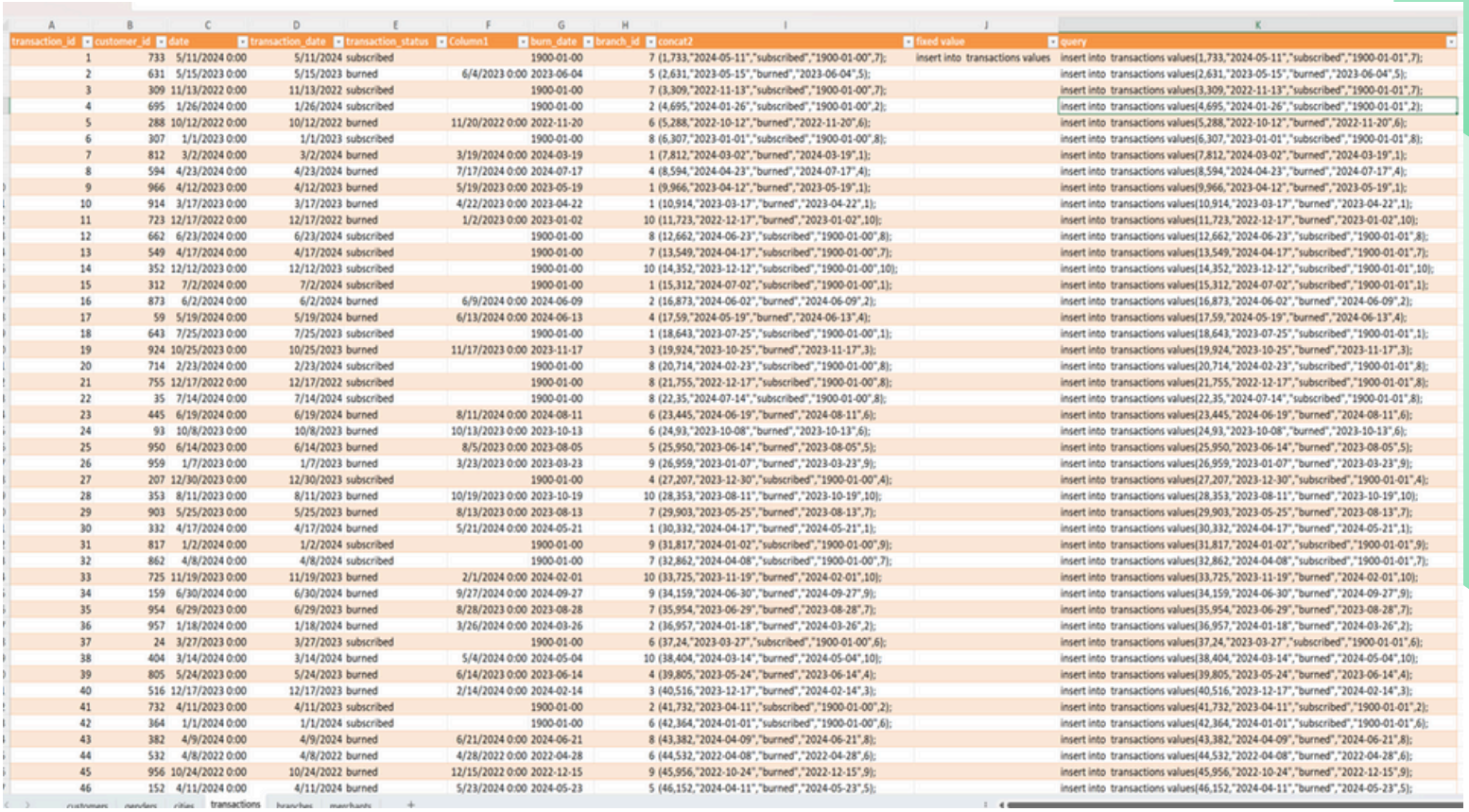
Table creation

Its time to create the database we used SQL SERVER to make the tables using the following queries:



Data handling

As our data were large some of the tables has more than 5000, so we needed to find a methodology to repeat the queries instead of writing them manually each time: We used EXCEL to repeat the queries like the following sample



transaction_id	customer_id	date	transaction_date	transaction_status	Column1	burn_date	branch_id	concat2	fixed value	query
1	733	5/11/2024 0:00	5/11/2024	subscribed		1900-01-00		7 (1,733,"2024-05-11","subscribed","1900-01-00",7);	insert into transactions values	insert into transactions values(1,733,"2024-05-11","subscribed","1900-01-01",7);
2	631	5/15/2023 0:00	5/15/2023	burned	6/4/2023 0:00	2023-06-04		5 (2,631,"2023-05-15","burned","2023-06-04",5);		insert into transactions values(2,631,"2023-05-15","burned","2023-06-04",5);
3	309	11/13/2022 0:00	11/13/2022	subscribed		1900-01-00		7 (3,309,"2022-11-13","subscribed","1900-01-00",7);		insert into transactions values(3,309,"2022-11-13","subscribed","1900-01-01",7);
4	695	1/26/2024 0:00	1/26/2024	subscribed		1900-01-00		2 (4,695,"2024-01-26","subscribed","1900-01-00",2);		insert into transactions values(4,695,"2024-01-26","subscribed","1900-01-01",2);
5	288	10/12/2022 0:00	10/12/2022	burned	11/20/2022 0:00	2022-11-20		6 (5,288,"2022-10-12","burned","2022-11-20",6);		insert into transactions values(5,288,"2022-10-12","burned","2022-11-20",6);
6	307	1/1/2023 0:00	1/1/2023	subscribed		1900-01-00		8 (6,307,"2023-01-01","subscribed","1900-01-00",8);		insert into transactions values(6,307,"2023-01-01","subscribed","1900-01-01",8);
7	812	3/2/2024 0:00	3/2/2024	burned	3/19/2024 0:00	2024-03-19		1 (7,812,"2024-03-02","burned","2024-03-19",1);		insert into transactions values(7,812,"2024-03-02","burned","2024-03-19",1);
8	594	4/23/2024 0:00	4/23/2024	burned	7/17/2024 0:00	2024-07-17		4 (8,594,"2024-04-23","burned","2024-07-17",4);		insert into transactions values(8,594,"2024-04-23","burned","2024-07-17",4);
9	966	4/12/2023 0:00	4/12/2023	burned	5/19/2023 0:00	2023-05-19		1 (9,966,"2023-04-12","burned","2023-05-19",1);		insert into transactions values(9,966,"2023-04-12","burned","2023-05-19",1);
10	914	3/17/2023 0:00	3/17/2023	burned	4/22/2023 0:00	2023-04-22		1 (10,914,"2023-03-17","burned","2023-04-22",1);		insert into transactions values(10,914,"2023-03-17","burned","2023-04-22",1);
11	723	12/17/2022 0:00	12/17/2022	burned	1/2/2023 0:00	2023-01-02		10 (11,723,"2022-12-17","burned","2023-01-02",10);		insert into transactions values(11,723,"2022-12-17","burned","2023-01-02",10);
12	662	6/23/2024 0:00	6/23/2024	subscribed		1900-01-00		8 (12,662,"2024-06-23","subscribed","1900-01-00",8);		insert into transactions values(12,662,"2024-06-23","subscribed","1900-01-01",8);
13	549	4/17/2024 0:00	4/17/2024	subscribed		1900-01-00		7 (13,549,"2024-04-17","subscribed","1900-01-00",7);		insert into transactions values(13,549,"2024-04-17","subscribed","1900-01-01",7);
14	352	12/12/2023 0:00	12/12/2023	subscribed		1900-01-00		10 (14,352,"2023-12-12","subscribed","1900-01-00",10);		insert into transactions values(14,352,"2023-12-12","subscribed","1900-01-01",10);
15	312	7/2/2024 0:00	7/2/2024	subscribed		1900-01-00		1 (15,312,"2024-07-02","subscribed","1900-01-00",1);		insert into transactions values(15,312,"2024-07-02","subscribed","1900-01-01",1);
16	873	6/2/2024 0:00	6/2/2024	burned	6/9/2024 0:00	2024-06-09		2 (16,873,"2024-06-02","burned","2024-06-09",2);		insert into transactions values(16,873,"2024-06-02","burned","2024-06-09",2);
17	59	5/19/2024 0:00	5/19/2024	burned	6/13/2024 0:00	2024-06-13		4 (17,59,"2024-05-19","burned","2024-06-13",4);		insert into transactions values(17,59,"2024-05-19","burned","2024-06-13",4);
18	643	7/25/2023 0:00	7/25/2023	subscribed		1900-01-00		1 (18,643,"2023-07-25","subscribed","1900-01-00",1);		insert into transactions values(18,643,"2023-07-25","subscribed","1900-01-01",1);
19	924	10/25/2023 0:00	10/25/2023	burned	11/17/2023 0:00	2023-11-17		3 (19,924,"2023-10-25","burned","2023-11-17",3);		insert into transactions values(19,924,"2023-10-25","burned","2023-11-17",3);
20	714	2/23/2024 0:00	2/23/2024	subscribed		1900-01-00		8 (20,714,"2024-02-23","subscribed","1900-01-00",8);		insert into transactions values(20,714,"2024-02-23","subscribed","1900-01-01",8);
21	755	12/17/2022 0:00	12/17/2022	subscribed		1900-01-00		8 (21,755,"2022-12-17","subscribed","1900-01-00",8);		insert into transactions values(21,755,"2022-12-17","subscribed","1900-01-01",8);
22	35	7/14/2024 0:00	7/14/2024	subscribed		1900-01-00		8 (22,35,"2024-07-14","subscribed","1900-01-00",8);		insert into transactions values(22,35,"2024-07-14","subscribed","1900-01-01",8);
23	445	6/19/2024 0:00	6/19/2024	burned	8/11/2024 0:00	2024-08-11		6 (23,445,"2024-06-19","burned","2024-08-11",6);		insert into transactions values(23,445,"2024-06-19","burned","2024-08-11",6);
24	93	10/8/2023 0:00	10/8/2023	burned	10/13/2023 0:00	2023-10-13		6 (24,93,"2023-10-08","burned","2023-10-13",6);		insert into transactions values(24,93,"2023-10-08","burned","2023-10-13",6);
25	950	6/14/2023 0:00	6/14/2023	burned	8/5/2023 0:00	2023-08-05		5 (25,950,"2023-06-14","burned","2023-08-05",5);		insert into transactions values(25,950,"2023-06-14","burned","2023-08-05",5);
26	959	1/7/2023 0:00	1/7/2023	burned	3/23/2023 0:00	2023-03-23		9 (26,959,"2023-01-07","burned","2023-03-23",9);		insert into transactions values(26,959,"2023-01-07","burned","2023-03-23",9);
27	207	12/30/2023 0:00	12/30/2023	subscribed		1900-01-00		4 (27,207,"2023-12-30","subscribed","1900-01-00",4);		insert into transactions values(27,207,"2023-12-30","subscribed","1900-01-01",4);
28	353	8/11/2023 0:00	8/11/2023	burned	10/19/2023 0:00	2023-10-19		10 (28,353,"2023-08-11","burned","2023-10-19",10);		insert into transactions values(28,353,"2023-08-11","burned","2023-10-19",10);
29	903	5/25/2023 0:00	5/25/2023	burned	8/13/2023 0:00	2023-08-13		7 (29,903,"2023-05-25","burned","2023-08-13",7);		insert into transactions values(29,903,"2023-05-25","burned","2023-08-13",7);
30	332	4/17/2024 0:00	4/17/2024	burned	5/21/2024 0:00	2024-05-21		1 (30,332,"2024-04-17","burned","2024-05-21",1);		insert into transactions values(30,332,"2024-04-17","burned","2024-05-21",1);
31	817	1/2/2024 0:00	1/2/2024	subscribed		1900-01-00		9 (31,817,"2024-01-02","subscribed","1900-01-00",9);		insert into transactions values(31,817,"2024-01-02","subscribed","1900-01-01",9);
32	862	4/8/2024 0:00	4/8/2024	subscribed		1900-01-00		7 (32,862,"2024-04-08","subscribed","1900-01-00",7);		insert into transactions values(32,862,"2024-04-08","subscribed","1900-01-01",7);
33	725	11/19/2023 0:00	11/19/2023	burned	2/1/2024 0:00	2024-02-01		10 (33,725,"2023-11-19","burned","2024-02-01",10);		insert into transactions values(33,725,"2023-11-19","burned","2024-02-01",10);
34	159	6/30/2024 0:00	6/30/2024	burned	9/27/2024 0:00	2024-09-27		9 (34,159,"2024-06-30","burned","2024-09-27",9);		insert into transactions values(34,159,"2024-06-30","burned","2024-09-27",9);
35	954	6/29/2023 0:00	6/29/2023	burned	8/28/2023 0:00	2023-08-28		7 (35,954,"2023-06-29","burned","2023-08-28",7);		insert into transactions values(35,954,"2023-06-29","burned","2023-08-28",7);
36	957	1/18/2024 0:00	1/18/2024	burned	3/26/2024 0:00	2024-03-26		2 (36,957,"2024-01-18","burned","2024-03-26",2);		insert into transactions values(36,957,"2024-01-18","burned","2024-03-26",2);
37	24	3/27/2023 0:00	3/27/2023	subscribed		1900-01-00		6 (37,24,"2023-03-27","subscribed","1900-01-00",6);		insert into transactions values(37,24,"2023-03-27","subscribed","1900-01-01",6);
38	404	3/14/2024 0:00	3/14/2024	burned	5/4/2024 0:00	2024-05-04		10 (38,404,"2024-03-14","burned","2024-05-04",10);		insert into transactions values(38,404,"2024-03-14","burned","2024-05-04",10);
39	805	5/24/2023 0:00	5/24/2023	burned	6/14/2023 0:00	2023-06-14		4 (39,805,"2023-05-24","burned","2023-06-14",4);		insert into transactions values(39,805,"2023-05-24","burned","2023-06-14",4);
40	516	12/17/2023 0:00	12/17/2023	burned	2/14/2024 0:00	2024-02-14		3 (40,516,"2023-12-17","burned","2024-02-14",3);		insert into transactions values(40,516,"2023-12-17","burned","2024-02-14",3);
41	732	4/11/2023 0:00	4/11/2023	subscribed		1900-01-00		2 (41,732,"2023-04-11","subscribed","1900-01-00",2);		insert into transactions values(41,732,"2023-04-11","subscribed","1900-01-01",2);
42	364	1/1/2024 0:00	1/1/2024	subscribed		1900-01-00		6 (42,364,"2024-01-01","subscribed","1900-01-00",6);		insert into transactions values(42,364,"2024-01-01","subscribed","1900-01-01",6);
43	382	4/9/2024 0:00	4/9/2024	burned	6/21/2024 0:00	2024-06-21		8 (43,382,"2024-04-09","burned","2024-06-21",8);		insert into transactions values(43,382,"2024-04-09","burned","2024-06-21",8);
44	532	4/8/2022 0:00	4/8/2022	burned	4/28/2022 0:00	2022-04-28		6 (44,532,"2022-04-08","burned","2022-04-28",6);		insert into transactions values(44,532,"2022-04-08","burned","2022-04-28",6);
45	956	10/24/2022 0:00	10/24/2022	burned	12/15/2022 0:00	2022-12-15		9 (45,956,"2022-10-24","burned","2022-12-15",9);		insert into transactions values(45,956,"2022-10-24","burned","2022-12-15",9);
46	152	4/11/2024 0:00	4/11/2024	burned	5/23/2024 0:00	2024-05-23		5 (46,152,"2024-04-11","burned","2024-05-23",5);		insert into transactions values(46,152,"2024-04-11","burned","2024-05-23",5);

This is a photo of data after editing

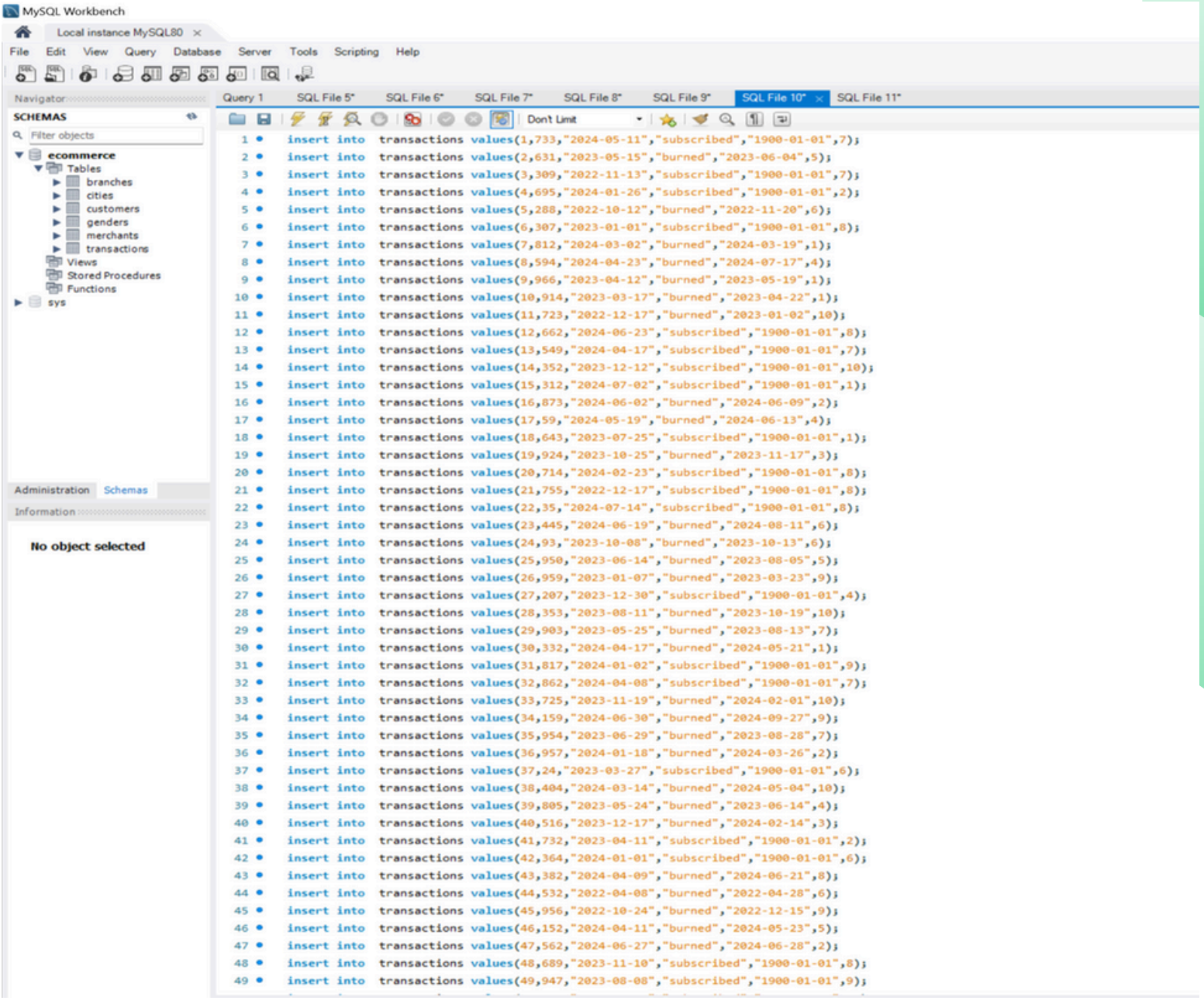
We repeated the queries but with different values. The codes used here are:

The main statement to insert data into a data base is :
INSERT INTO [Table name]([Columns name]) values1([Values]);
concat column > [Values]:
In this query our target was to concatenate the values in each column to create the “([Values])” part of the query
=CHAR(40)&
CONCAT(A2,",",B2,",",D2,",",E2,",",G2,",",H2)&CHAR(41)&CHAR(59)
Notes: -the text is inserted in the SQL between quotations -the time also is added as text and between quotations [‘date’ column] In order to solve the first.

we created a new column for each text to be between quotations like “column” By the following code: =CHAR(39) &C2 & CHAR(39) [Date to text column]
And to solve the timing issues we casted it to text using the following code:
=TEXT(B2, “yyyy – mm – dd”)

2 - Fixed part > INSERT INTO [Table name] ([Columns name]) values: This part constant in each row so we didn`t repeat it instead, we referred to it by dollar sign in the final statement to be static. INSERT INTO transactions values (id, customer_id, transaction_date,status,burn_date,branch_id)
Final syntax- > concatenation between 1, 2: Here we concatenated the fixed part in every syntax with the dynamic part of values = CONCAT (j2, K2)

Insert data into tables SQL sample:



Trend Analysis

We used Python to identify trends in sales and marketing performance over time while also using historical sales data to create forecasting models in python

Install necessary libraries:

In [1]:
```
import warnings
warnings.filterwarnings('ignore')

# import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#import sheets
file_path = 'C:/Users/abdelrahman  hassan/Desktop/E-commerce_data.xlsx'

customers = pd.read_excel(file_path, sheet_name='customers')
genders = pd.read_excel(file_path, sheet_name='genders')
cities = pd.read_excel(file_path, sheet_name='cities')
transactions = pd.read_excel(file_path, sheet_name='transactions')
branches = pd.read_excel(file_path, sheet_name='branches')
merchants = pd.read_excel(file_path, sheet_name='merchants')
```

In [3]:
```
# Merge Sheets To one File
merged_data = pd.merge(transactions, customers, how='left', on='customer_id')

merged_data = pd.merge(merged_data, genders, how='left', on='gender_id')

merged_data = pd.merge(merged_data, cities, how='left', on='city_id')

merged_data = pd.merge(merged_data, branches, how='left', on='branch_id')

merged_data = pd.merge(merged_data, merchants, how='left', on='merchant_id')

print(merged_data.head())
```

```
   transaction_id  customer_id transaction_date transaction_status  \
0               1          733       2024-05-11         subscribed
1               2          631       2023-05-15             burned
2               3          309       2022-11-13         subscribed
3               4          695       2024-01-26         subscribed
4               5          288       2022-10-12             burned

  coupon_name  burn_date  branch_id  join_date  city_id  gender_id  \
0     uNY-568        NaT          7 2023-05-16       15          2
1     kBx-219 2023-06-04          5 2022-10-10       14          2
2     PLG-122        NaT          7 2022-05-30        2          1
3     Wzf-897        NaT          2 2023-11-27       15          2
4     qGb-428 2022-11-20          6 2021-09-04       14          1

  gender_name   city_name  merchant_id                  merchant_name
0      Female       Aswan            7                 Berry-Anderson
1      Female    Damietta            9  Campbell, Shaffer and Martinez
2        Male  Alexandria            7                 Berry-Anderson
3      Female       Aswan           18                      Lewis LLC
4        Male    Damietta           15                    Butler-Gomez
```

In [5]:
```
# Rename File
data = merged_data
```

Trend Analysis

data description

In [6]:
```
data.head(10)
```

Out[6]:

	transaction_id	customer_id	transaction_date	transaction_status	coupon_name	burn_date	branch_id	join_date	city_id	gender_id	gender_name	city_name	merchant_id	merchant_name
0	1	733	2024-05-11	subscribed	uNY-568	NaT	7	2023-05-16	15	2	Female	Aswan	7	Berry-Anderson
1	2	631	2023-05-15	burned	kBx-219	2023-06-04	5	2022-10-10	14	2	Female	Damietta	9	Campbell, Shaffer and Martinez
2	3	309	2022-11-13	subscribed	PLG-122	NaT	7	2022-05-30	2	1	Male	Alexandria	7	Berry-Anderson
3	4	695	2024-01-26	subscribed	Wzf-897	NaT	2	2023-11-27	15	2	Female	Aswan	18	Lewis LLC
4	5	288	2022-10-12	burned	qGb-428	2022-11-20	6	2021-09-04	14	1	Male	Damietta	15	Butler-Gomez
5	6	307	2023-01-01	subscribed	gke-047	NaT	8	2021-12-20	2	1	Male	Alexandria	20	Griffin-Leblanc
6	7	812	2024-03-02	burned	Xrl-052	2024-03-19	1	2023-10-27	8	1	Male	El-Mahalla El-Kubra	11	Smith, Lawson and Hernandez
7	8	594	2024-04-23	burned	ogZ-906	2024-07-17	4	2024-01-30	3	1	Male	Giza	15	Butler-Gomez
8	9	966	2023-04-12	burned	dwn-619	2023-05-19	1	2022-10-13	12	1	Male	Fayyum	11	Smith, Lawson and Hernandez
9	10	914	2023-03-17	burned	Wjs-081	2023-04-22	1	2022-12-09	20	2	Female	Sohag	11	Smith, Lawson and Hernandez

In [7]:
```
data.tail(10)
```

Out[7]:

	transaction_id	customer_id	transaction_date	transaction_status	coupon_name	burn_date	branch_id	join_date	city_id	gender_id	gender_name	city_name	merchant_id	merchant_name
4990	4991	159	2024-07-05	subscribed	MhW-567	NaT	1	2024-06-13	9	1	Male	Tanta	11	Smith, Lawson and Hernandez
4991	4992	201	2023-12-23	subscribed	PIZ-597	NaT	6	2023-08-09	3	1	Male	Giza	15	Butler-Gomez
4992	4993	487	2024-06-26	subscribed	Yih-583	NaT	9	2022-03-13	12	2	Female	Fayyum	13	Thomas-Nelson
4993	4994	557	2024-02-13	subscribed	WUa-348	NaT	8	2022-08-28	11	1	Male	Ismailia	20	Griffin-Leblanc
4994	4995	962	2023-03-29	subscribed	lll-187	NaT	3	2021-10-17	15	2	Female	Aswan	8	Medina-Foster
4995	4996	776	2024-03-20	subscribed	OSq-518	NaT	1	2022-01-08	12	1	Male	Fayyum	11	Smith, Lawson and Hernandez
4996	4997	583	2024-06-27	subscribed	FsJ-607	NaT	2	2023-08-03	5	2	Female	Port Said	18	Lewis LLC
4997	4998	504	2022-11-08	burned	WcY-330	2022-12-06	2	2022-03-12	10	2	Female	Asyut	18	Lewis LLC
4998	4999	876	2024-05-05	burned	bgx-731	2024-07-29	7	2022-11-19	5	2	Female	Port Said	7	Berry-Anderson
4999	5000	371	2022-08-29	subscribed	vYJ-493	NaT	6	2022-02-11	16	1	Male	Minya	15	Butler-Gomez

In [12]:
```
data.isnull().sum()
```

Out[12]:
```
transaction_id          0
customer_id             0
transaction_date        0
transaction_status      0
coupon_name             0
burn_date            2484
branch_id               0
join_date               0
city_id                 0
gender_id               0
gender_name             0
city_name               0
merchant_id             0
merchant_name           0
dtype: int64
```

In [13]:
```
data.shape
```

Out[13]:
```
(5000, 14)
```

statistical summary

In [14]:
```
data.describe()
```

Out[14]:

	transaction_id	customer_id	transaction_date	burn_date	branch_id	join_date	city_id	gender_id	merchant_id
count	5000.000000	5000.000000	5000	2516	5000.000000	5000	5000.000000	5000.00000	5000.000000
mean	2500.500000	509.551600	2023-10-22 04:38:47.040000	2023-12-01 20:39:06.581876224	5.529800	2023-01-18 14:41:34.080000	10.091800	1.48660	12.882600
min	1.000000	1.000000	2021-08-05 00:00:00	2021-08-14 00:00:00	1.000000	2021-07-18 00:00:00	1.000000	1.00000	7.000000
25%	1250.750000	262.000000	2023-05-31 18:00:00	2023-07-12 00:00:00	3.000000	2022-04-10 00:00:00	5.000000	1.00000	9.000000
50%	2500.500000	514.000000	2024-01-02 00:00:00	2024-02-11 00:00:00	6.000000	2023-01-11 00:00:00	10.000000	1.00000	13.000000
75%	3750.250000	758.000000	2024-05-08 00:00:00	2024-06-18 00:00:00	8.000000	2023-10-25 00:00:00	15.000000	2.00000	15.000000
max	5000.000000	1000.000000	2024-07-14 00:00:00	2024-10-08 00:00:00	10.000000	2024-07-13 00:00:00	20.000000	2.00000	20.000000
std	1443.520003	287.065172	NaN	NaN	2.895788	NaN	5.860703	0.49987	4.000452

Trend Analysis

Data analysis

In [16]:
```
# Merchant Activity (Number of Transactions per Merchant)
merchant_activity = data['merchant_name'].value_counts().reset_index()
merchant_activity
```

Out[16]:

	merchant_name	count
0	Thomas-Nelson	1032
1	Butler-Gomez	1001
2	Smith, Lawson and Hernandez	527
3	Berry-Anderson	496
4	Griffin-Leblanc	496
5	Campbell, Shaffer and Martinez	489
6	Medina-Foster	487
7	Lewis LLC	472

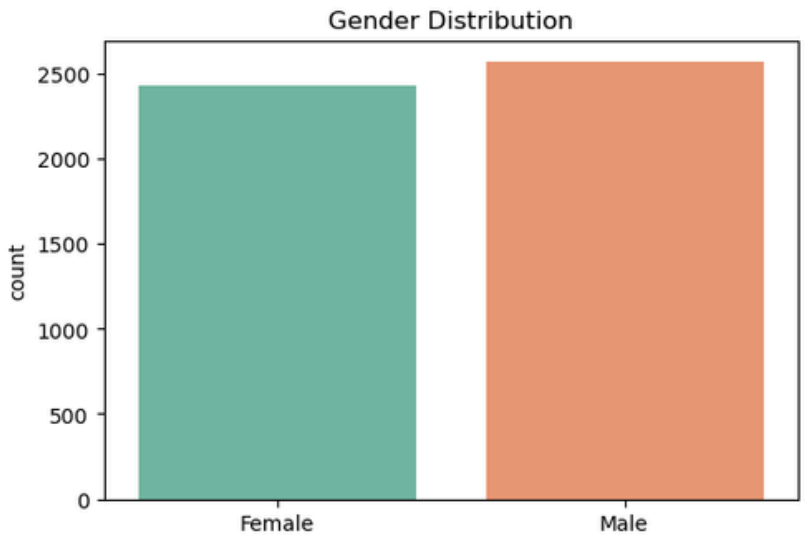
In [23]:
```
# Average Number of Transactions per Customer
avg_transactions_per_customer = data.groupby('customer_id').size().mean()
print("\nAverage Number of Transactions per Customer:\n", avg_transactions_per_customer)
```

```
Average Number of Transactions per Customer:
 5.055611729019211
```

In [24]:
```
# Coupon Burn Rate (ratio of burned to subscribed)
burned_coupons = data[data['transaction_status'] == 'burned'].shape[0]
subscribed_coupons = data[data['transaction_status'] == 'subscribed'].shape[0]
coupon_burn_rate = burned_coupons / subscribed_coupons if subscribed_coupons > 0 else 0
print("\nCoupon Burn Rate (burned/subscribed):\n", coupon_burn_rate)
```

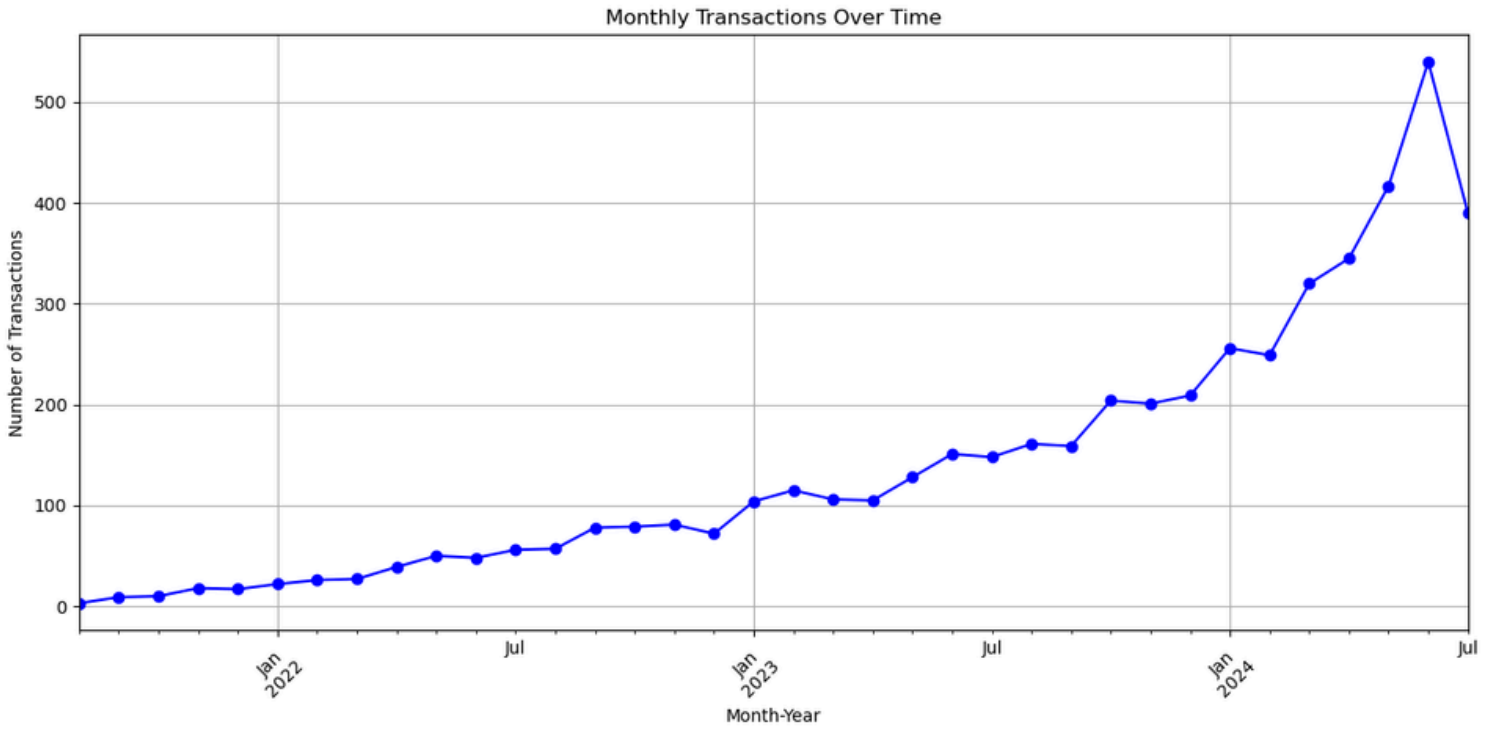
```
Coupon Burn Rate (burned/subscribed):
 1.0128824476650564
```

In [25]:
```
# Distribution of gender

plt.figure(figsize=(6, 4))
sns.countplot(data=data, x='gender_name', palette='Set2')
plt.title('Gender Distribution')
plt.show()
```

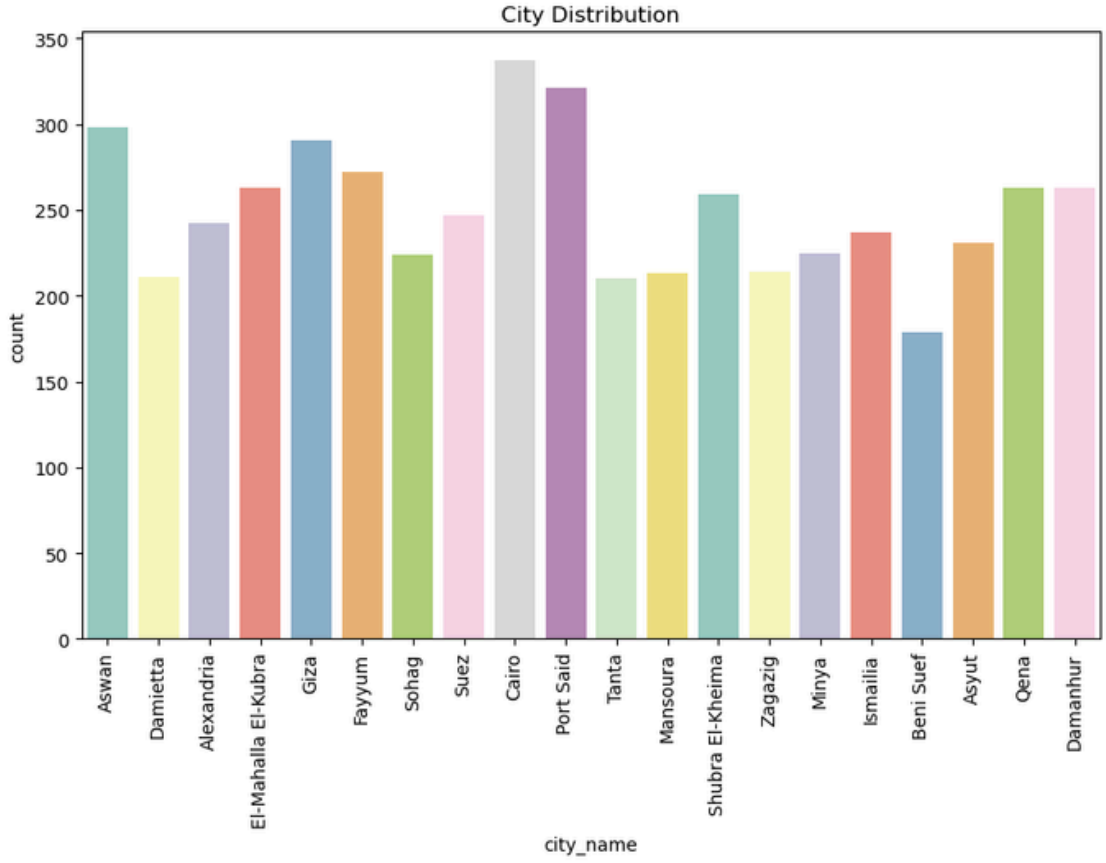


Trend Analysis

In [26]:
```
plt.figure(figsize=(12, 6))
monthly_trends = data.groupby(data['transaction_date'].dt.to_period('M')).size()
monthly_trends.plot(kind='line', marker='o', color='b')
plt.title("Monthly Transactions Over Time")
plt.xlabel("Month-Year")
plt.ylabel("Number of Transactions")
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

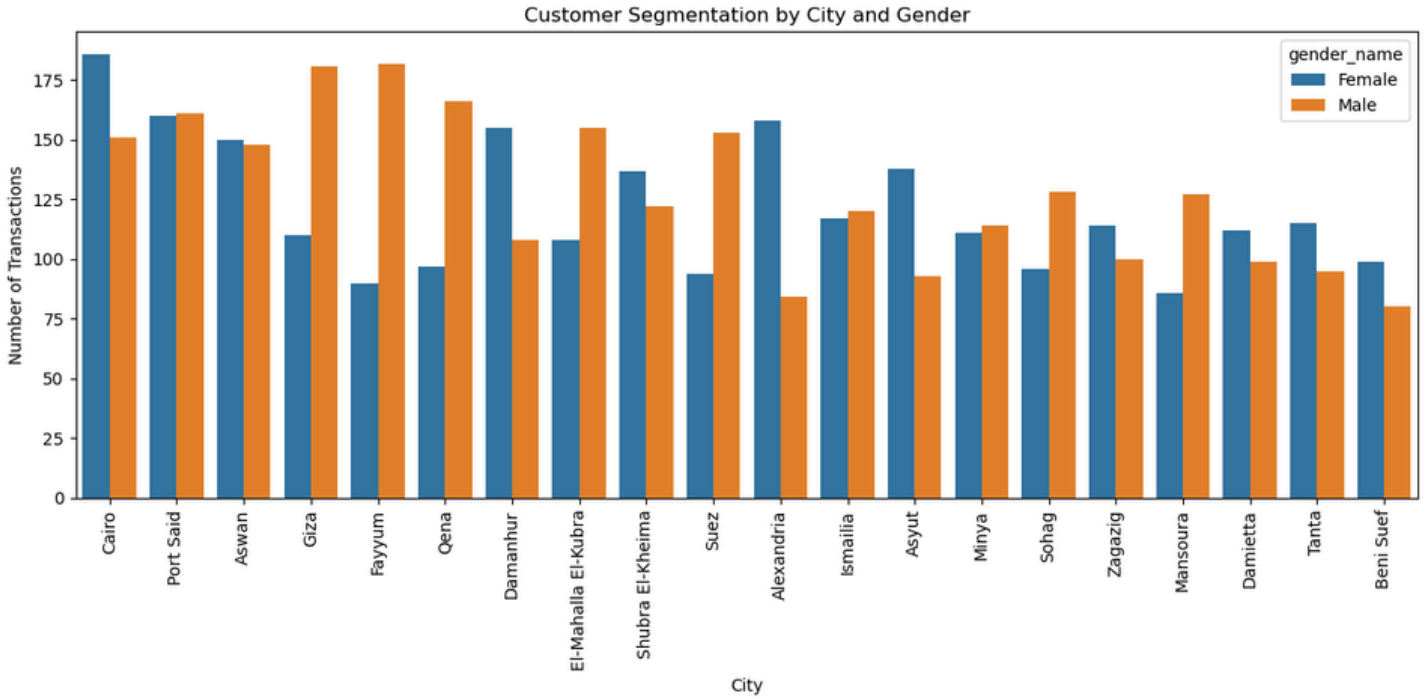


In [27]:
```
# Distribution of cities

plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='city_name', palette='Set3')
plt.title('City Distribution')
plt.xticks(rotation=90)
plt.show()
```

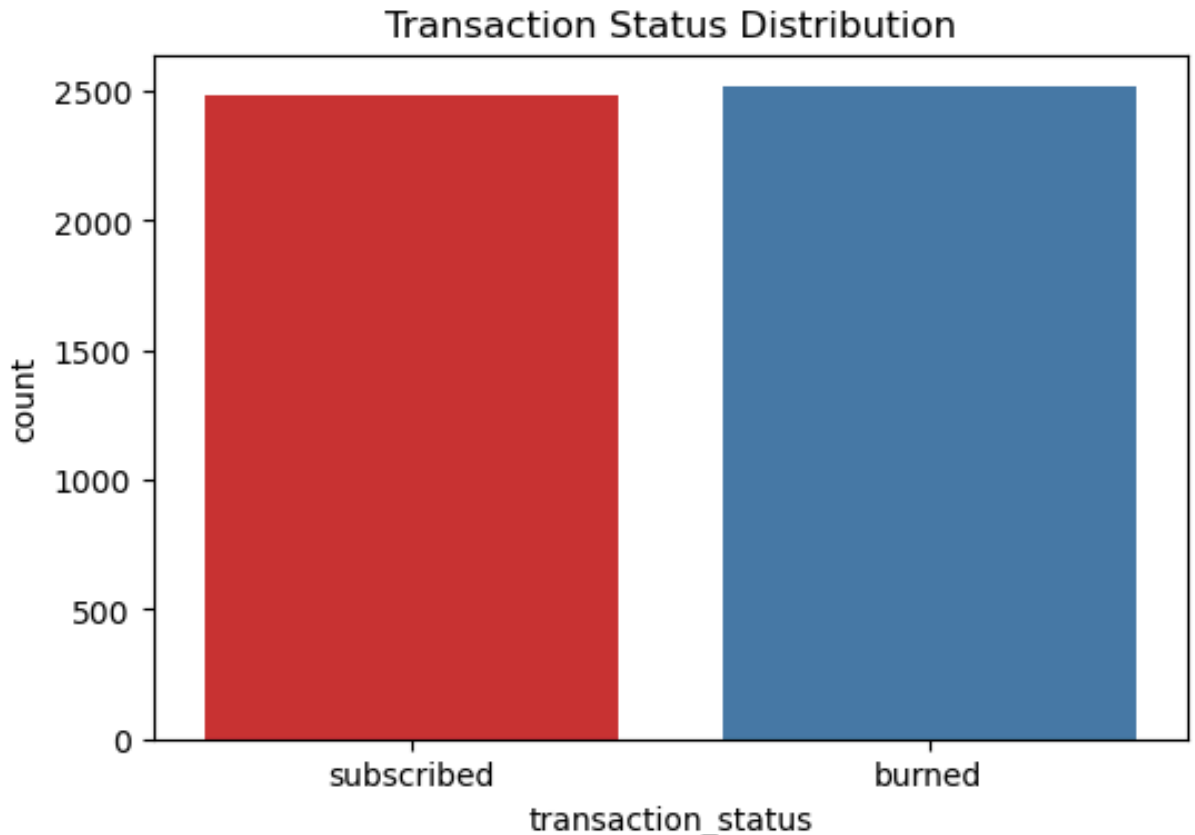


Trend Analysis

In [28]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='city_name', hue='gender_name', data=data, order=data['city_name'].value_counts().index)
plt.title('Customer Segmentation by City and Gender')
plt.xlabel('City')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

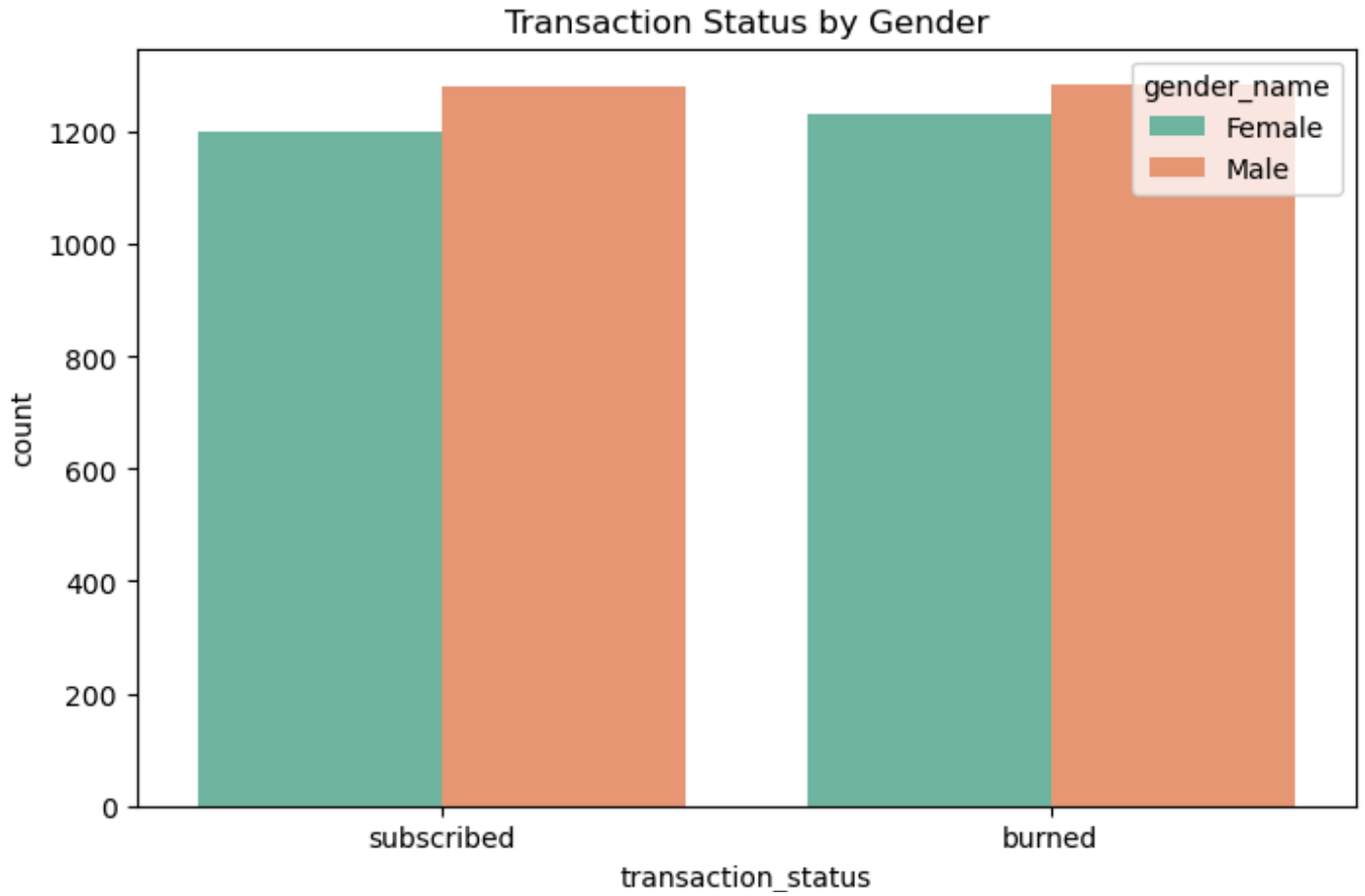


In [29]:
```
# Transaction status distribution

plt.figure(figsize=(6, 4))
sns.countplot(data=data, x='transaction_status', palette='Set1')
plt.title('Transaction Status Distribution')
plt.show()
```

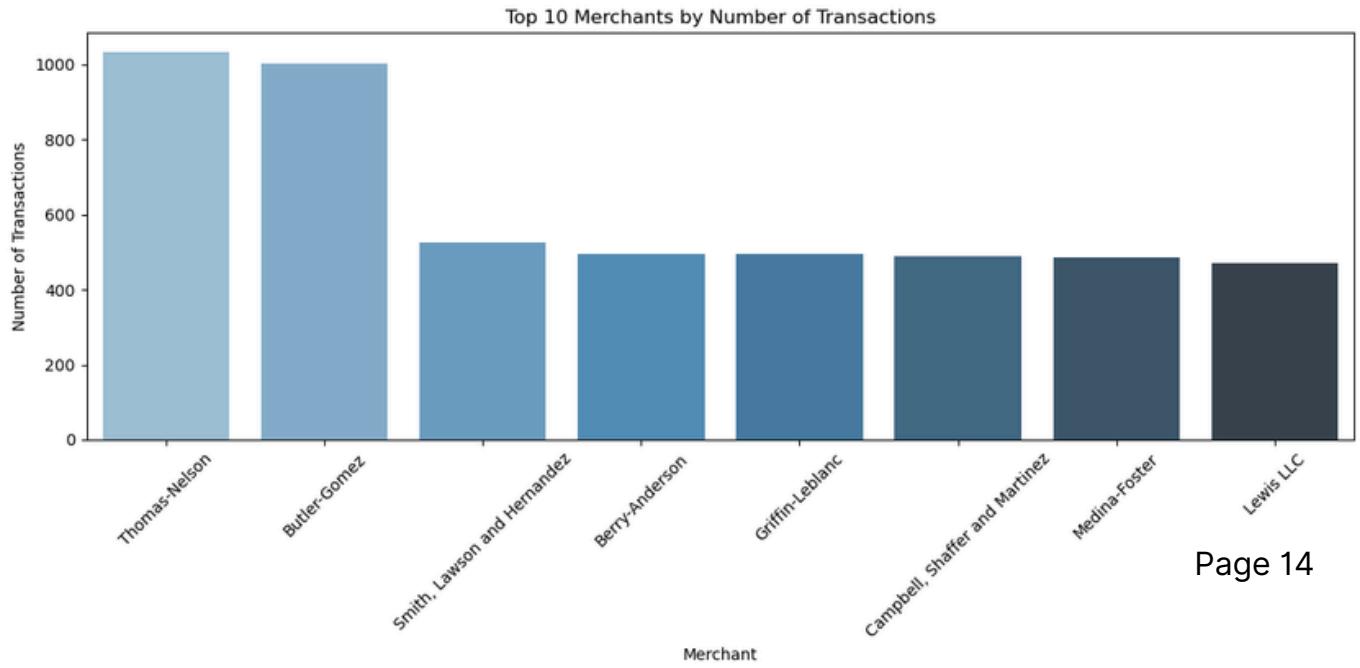


Trend Analysis

In [30]:
```
# The relationship between gender and transaction status

plt.figure(figsize=(8, 5))
sns.countplot(data=data, x='transaction_status', hue='gender_name', palette='Set2')
plt.title('Transaction Status by Gender')
plt.show()
```

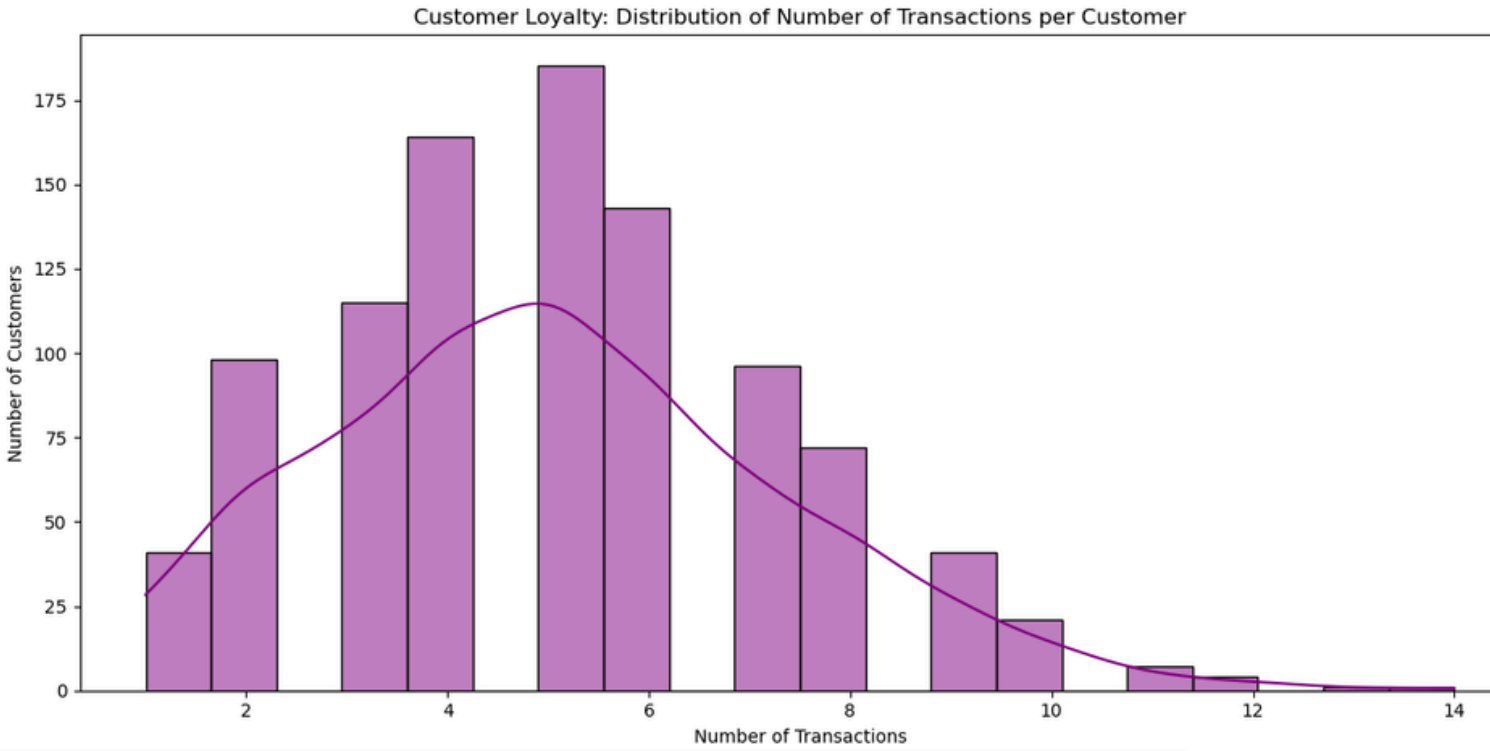


In [31]:
```
#Merchant Performance Analysis (Top 10 Merchants by Transaction Count)

plt.figure(figsize=(12, 6))
top_merchants = data['merchant_name'].value_counts().head(10)
sns.barplot(x=top_merchants.index, y=top_merchants.values, palette="Blues_d")
plt.title('Top 10 Merchants by Number of Transactions')
plt.xlabel('Merchant')
plt.ylabel('Number of Transactions')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

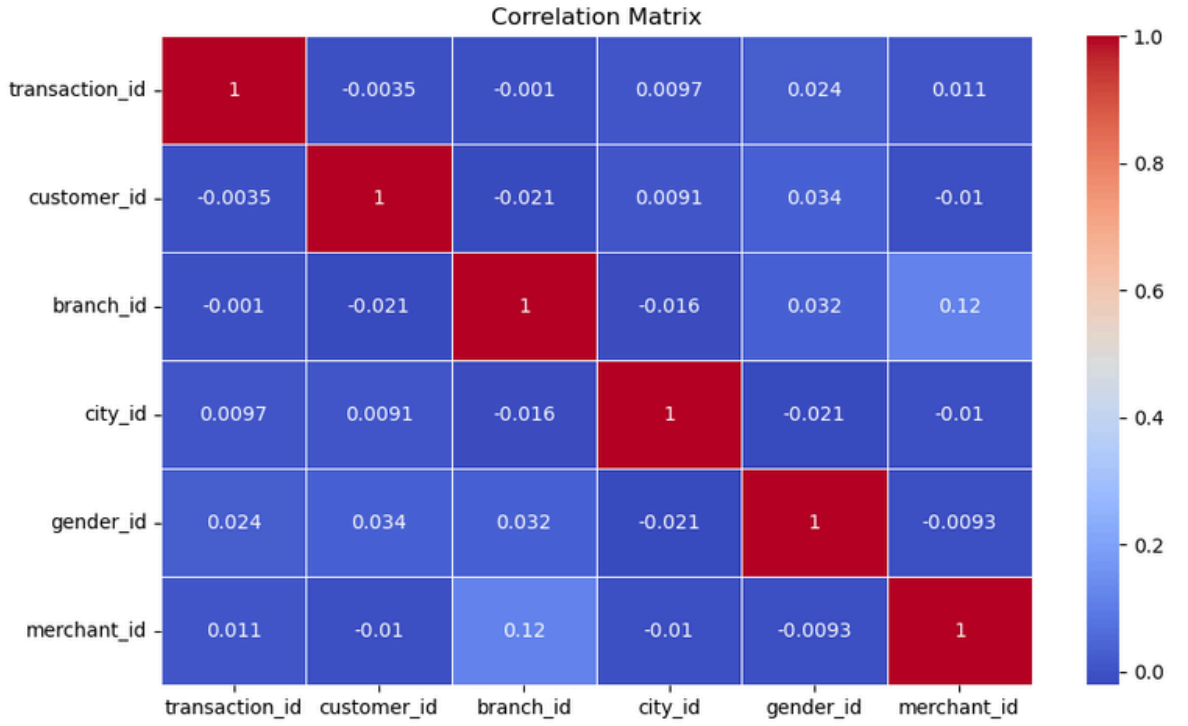


Trend Analysis

In [33]:
```
# Customer Loyalty Analysis
plt.figure(figsize=(12, 6))
customer_loyalty = data.groupby('customer_id').size()
sns.histplot(customer_loyalty, bins=20, kde=True, color='purple')
plt.title('Customer Loyalty: Distribution of Number of Transactions per Customer')
plt.xlabel('Number of Transactions')
plt.ylabel('Number of Customers')
plt.tight_layout()
plt.show()
```



In [34]:
```
# Analyzing the correlation between numerical features
import matplotlib.pyplot as plt
import seaborn as sns

# Selecting only the numerical columns
numerical_data = data.select_dtypes(include='number')

# Plotting the correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(numerical_data.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()
```



Creating Interactive Dashboards

E-commerce analysis using power bi

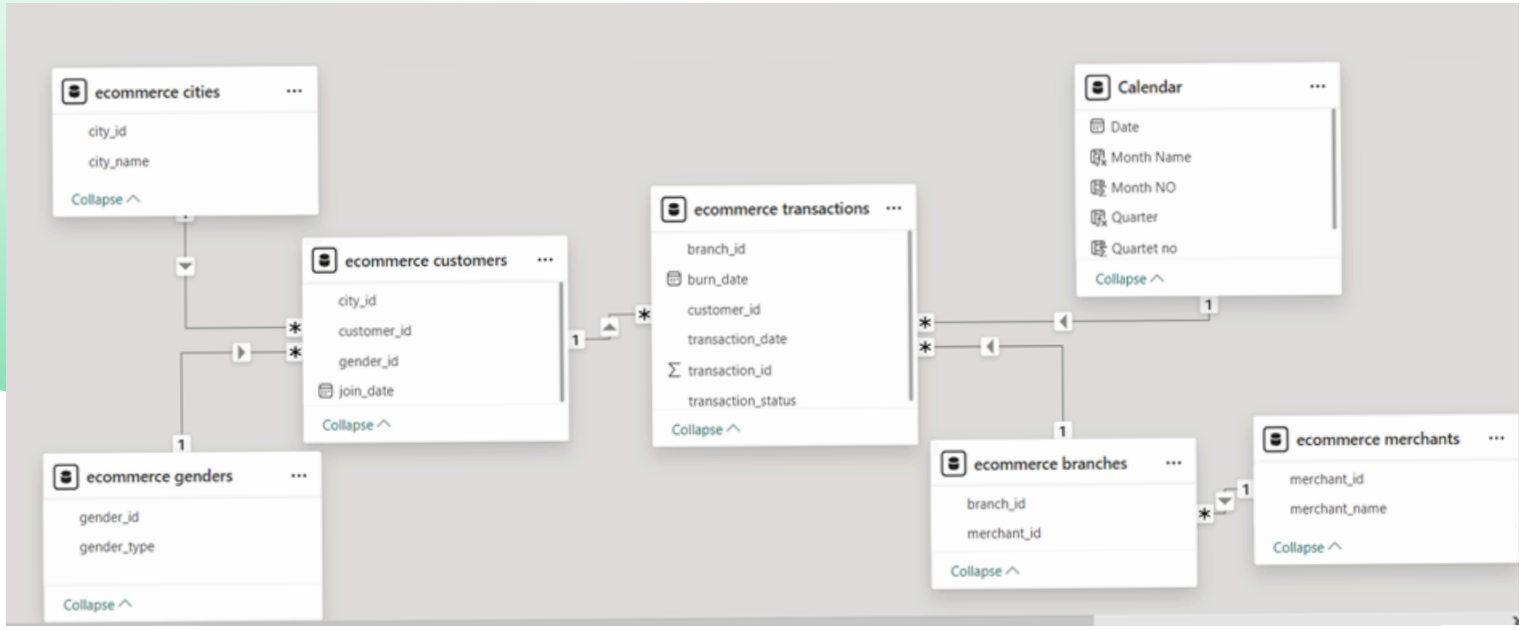
1. Import Data into Power BI from SQL
2. Clean and Transform the Data
3. Create a calendar table to facilitate accurate and flexible time-based analysis

```
Calendar = CALENDAR("1/1/2021","14/7/2024")
Month Number = MONTH(Calendar[Date])
Month Name =
FORMAT(Calendar[Date],"mmm")
Year = YEAR(Calendar[Date])
Quarter = "Q" & QUARTER(Calendar[Date])
Quartet Number = QUARTER(Calendar[Date])
```

4. Create Relationships

5. Create Measures Using DAX such as:

```
Total Orders = COUNT('ecommerce transactions'[customer_id])

Active Customers = DISTINCTCOUNT('ecommerce transactions' [customer_id])

Orders Subscribed = CALCULATE( COUNTROWS('ecommerce transactions'), 'ecommerce transactions'[transaction_status] = "subscribed")

Orders Burned = CALCULATE( COUNTROWS('ecommerce transactions'),
        'ecommerce transactions'[transaction_status] = "burned")

Customer Status Difference = [Orders Subscribed]-[Orders Burned]

Average Orders Per Customer =
VAR TotalOrders = COUNTROWS('ecommerce transactions')
VAR TotalCustomers = DISTINCTCOUNT('ecommerce transactions' [customer_id])
RETURN DIVIDE(TotalOrders, TotalCustomers)
```



6.Build Visualizations

1.KPIS (KEY PERFORMANCE INDICATORS) AT THE TOP:

Total Orders: Displays the total number of orders (5,000).

Active Customers: Shows the number of active customers (989).

Orders Subscribed: Indicates the total number of subscribed orders (2,484).

Orders Burned: Represents orders that were canceled or failed (2,516).

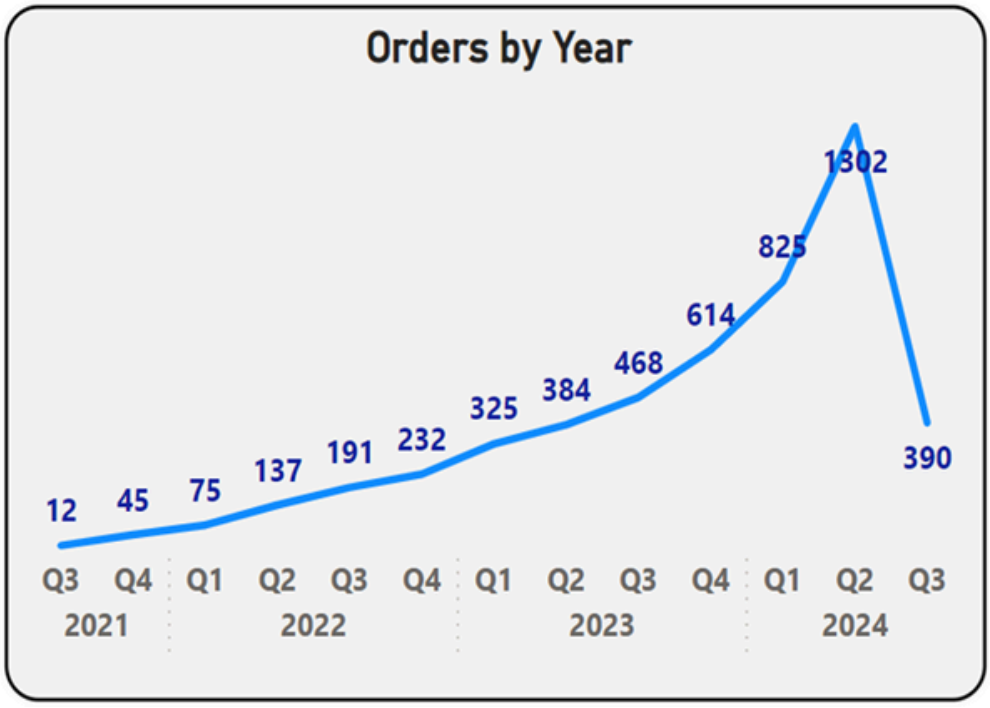
Customer Status Difference: Shows the net change in customer status (-32). A positive status difference in customer subscriptions indicates good retention and growth, while a negative difference might signal issues with customer satisfaction or engagement.

Average Orders Per Customer: Displays the average number of orders per customer (5.06).

6.Build Visualizations

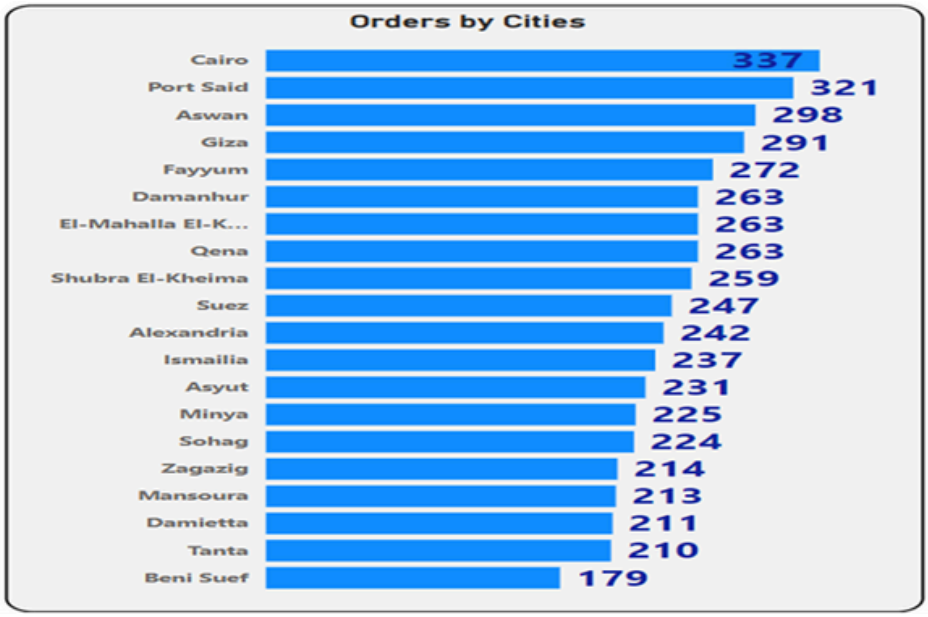
2. LINE CHARTS

Orders by Year: Plots the number of orders over time, segmented by quarters and years (2021-2024). This visual shows the trend of orders increasing each quarter.



3. BAR CHARTS

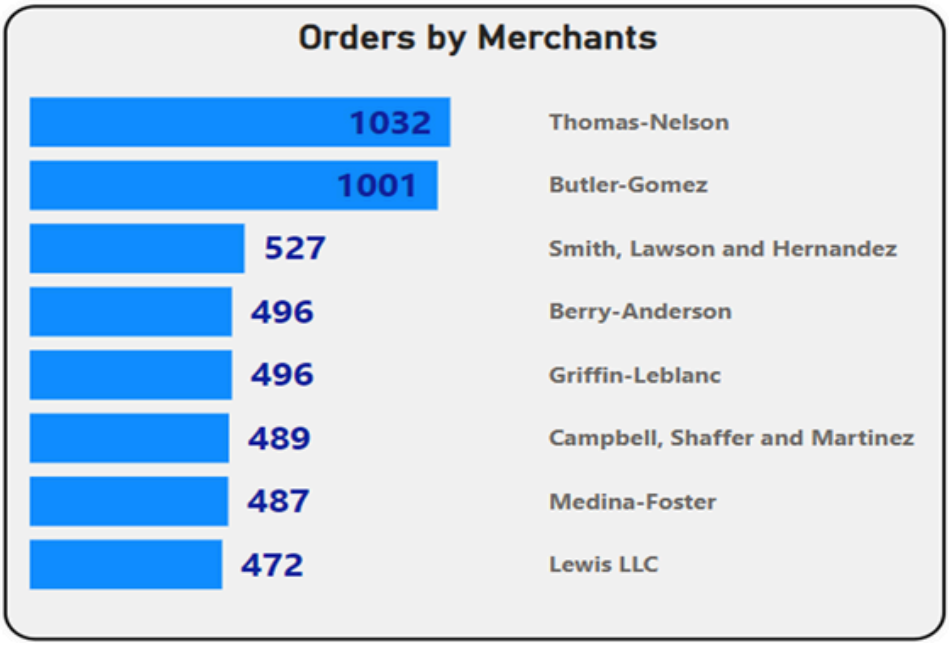
Orders by Cities: This shows the number of orders from different cities, with Cairo having the highest number of orders (337).



6.Build Visualizations

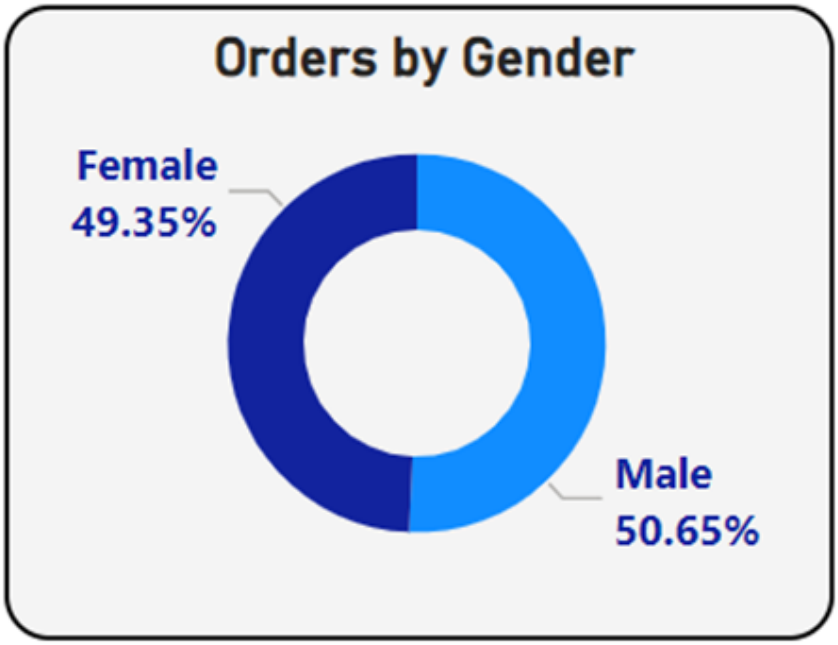
3. BAR CHARTS

Orders by Merchants: Displays the number of orders processed by various merchants, with "Thomas-Nelson" having the most orders (1,032).



4. PIE CHARTS

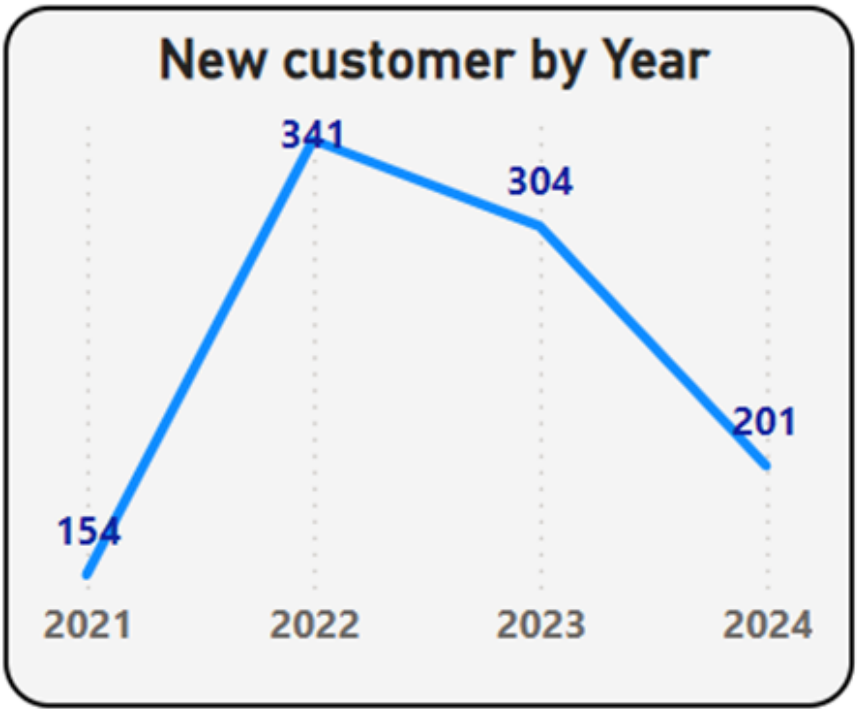
Orders by Gender: Illustrates the distribution of orders by gender, showing nearly equal percentages (Female: 49.35%, Male: 50.65%).



6.Build Visualizations

5. LINE CHARTS

New Customers by Year: This shows the number of new customers acquired yearly from 2021 to 2024. It peaks in 2022 (341) and declines in subsequent years



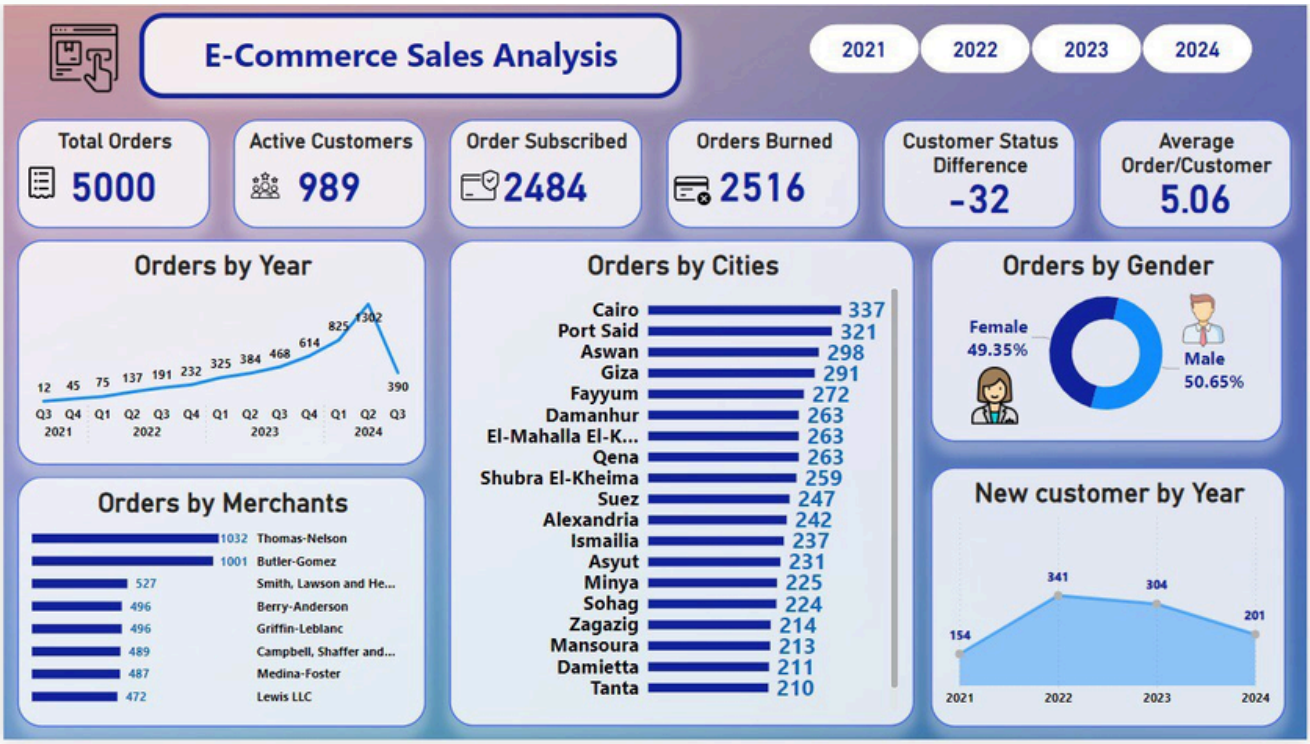
6. YEAR SLICER

A slicer on the left side allows data to be filtered based on the selected year (2021, 2022, 2023, or 2024).

Final Dashboard

After developing interactive dashboards that track KPIs in real time, Which Includes:
Sales Dashboard: Displays total sales, sales by product category, sales by region, and sales trends.
Marketing Performance Dashboard: Tracks campaign effectiveness (CTR, conversions, ROAS), marketing spend by channel, and customer acquisition cost.
Customer Segmentation Dashboard: Visualizes customer demographics, behavior, and purchase history to identify high-value segments.
Drill-Down Capabilities: Allow users to drill down into specific time periods, campaigns, or customer segments for more detailed insight

Based on the insights from the dashboards, it provides recommendations on which marketing channels to focus on, which campaigns to optimize, and where to cut costs



Improving Our E-commerce Performance

These steps can help improve this e-commerce platform's performance across multiple fronts.

1. Customer Retention & Engagement

-Active Customers vs. Orders Burned: We have 2,516 orders burned, which indicates potential issues with order fulfilment or customer churn.

*Focus on:

- Enhanced customer support to resolve issues quickly.
- Re-engagement campaigns through emails, SMS, or social media to win back lost customers.
- Customer feedback surveys to understand why customers stop purchasing and how to improve.

2. Increase Subscriptions

-Order Subscribed (2,484): This number is quite high, but we can boost it further by:

- Loyalty programs that offer discounts or exclusive content for subscribers.
- Personalized product recommendations based on browsing or purchasing behavior.
- Subscription incentives like free trials or exclusive deals to encourage more subscriptions.

3. Optimizing Average Order Value (AOV)

-Average Order/Customer (5.06).

*To raise this:

- Upselling and Cross-selling: Use product bundles, recommended products, and limited-time offers during checkout.
- Free Shipping Thresholds: Encourage larger orders by offering free shipping for purchases above a certain amount.
- Discounts on Bulk Purchases: Implement volume-based discounts for customers who buy more.

4. Target High-Performance Cities

-Orders by Cities: Cairo, Port Said, and Aswan are our top-performing cities.

*We can:

- Localized marketing efforts: Run targeted ads, promotions, or events specifically for top-performing cities.
- Improve shipping and fulfillment services in these cities to strengthen customer loyalty.
- Expand operations into nearby areas of high-performing cities to capture potential customers.

5. Boost Underperforming Regions

Cities like Mansoura, Damietta, and Tanta have lower order numbers.

*Consider:

- Localized promotions and offers in these cities to boost sales.
- Partnerships with local influencers or businesses to increase brand visibility.
- Tailored campaigns for these regions, focusing on their specific needs or preferences.

6. Customer Gender Insights

-Orders by Gender: There's a nearly equal split between male (50.65%) and female (49.35%) customers. This presents an opportunity to:

- Personalize campaigns targeting both genders, such as product recommendations based on gender-specific preferences.
- Gender-specific promotions like women's fashion sales or tech gadgets for men to cater to different segments.

7. Improve Merchant Relations

-Orders by Merchants: Thomas-Nelson and Butler-Gomez are the top merchants, but there's a steep drop-off after them. Focus on:

- Strengthening relationships with smaller merchants to increase their performance. This could include offering them better deals, support, or promotions on your platform.
- Onboarding new merchants with high-quality products to diversify your offerings.

8. Monitor Yearly Performance

-Orders by Year: There's a noticeable drop from 825 orders in Q2 2024 to 390 in Q3 2024.

*Investigate reasons for this dip:

- Analyze seasonal trends and identify if it's a temporary seasonal decline.
- Boost marketing efforts during low-performance quarters, especially with limited-time offers and holiday campaigns.
- Leverage data to predict upcoming low points and counteract with stronger campaigns and promotions.

Improving Our E-commerce Performance

9. Acquire New Customers

-New Customer Acquisition: There's a decline in new customer numbers in 2024.

*To reverse this trend:

- Run acquisition-focused marketing campaigns on social media, Google ads, or influencer partnerships.
- Referral programs where existing customers are rewarded for bringing in new customers.
- Improve SEO to ensure our e-commerce site ranks higher in search results.

10. Use Data to Predict Trends

-Future Planning: Use our historical sales data to:

- Forecast sales trends and plan inventory accordingly.
- Anticipate busy seasons and ensure the platform is ready to handle spikes in traffic or demand.
- Invest in data analytics tools to get deeper insights into customer behavior and preferences.

By focusing on customer retention, merchant relations, and regional targeting, we can drive growth and address potential challenges effectively.

Conclusion

This proposal outlines a comprehensive solution for tracking and optimizing E-commerce marketing and sales performance using Excel, SQL,python and Power BI. By centralizing data, automating reporting, and providing actionable insights, we aim to help Client make data-driven decisions that lead to higher sales, improved marketing efficiency, and overall business growth.