In [2]:

```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
import matplotlib.ticker as  mticks
from matplotlib.ticker import FuncFormatter
from sqlalchemy import create_engine
```

In [3]:

```
hostname = 'localhost'
username = 'root'
password = '############'
port = 3306
database = 'sales'

ce = create_engine('mysql+pymysql://'+username+':'+password+'@'+hostname+':'+str(port)+'/'+database)
```

In [4]:

```
conn = ce.connect()
```

In [5]:

```
# List all the unique customer
query = pd.read_sql_query('select distinct customer_name from customers',conn)
df = pd.DataFrame(query)
df
```

Out[5]:

| | customer_name |
|---|---|
| 0 | Surge Stores |
| 1 | Nomad Stores |
| 2 | Excel Stores |
| 3 | Surface Stores |
| 4 | Premium Stores |
| 5 | Electricalsara Stores |
| 6 | Info Stores |
| 7 | Acclaimed Stores |

| | customer_name |
|---|---|
| 8 | Electricalsquipo Stores |
| 9 | Atlas Stores |
| 10 | Flawless Stores |
| 11 | Integration Stores |
| 12 | Unity Stores |
| 13 | Forward Stores |
| 14 | Electricalsbea Stores |
| 15 | Logic Stores |
| 16 | Epic Stores |
| 17 | Electricalslance Stores |
| 18 | Electricalsopedia Stores |
| 19 | Nixon |
| 20 | Modular |
| 21 | Electricalslytical |
| 22 | Sound |
| 23 | Power |
| 24 | Path |
| 25 | Insight |
| 26 | Control |
| 27 | Sage |
| 28 | Electricalsocity |
| 29 | Synthetic |
| 30 | Zone |
| 31 | Elite |
| 32 | All-Out |
| 33 | Expression |
| 34 | Relief |
| 35 | Novus |
| 36 | Propel |
| 37 | Leader |

In [14]:

```
# Get the total revenue
query = pd.read_sql_query('select sum(sales_amount) as total_revenue from transactions',conn)
df = pd.DataFrame(query)
print(f'The total revenue : {int(df.loc[0,'total_revenue']):,}')
```

```
The total revenue : 984,812,713
```

In [7]:

```
# Get the total revenue for each year
query = pd.read_sql_query('''
                            select d.year,sum(sales_amount) as total_revenue
                            from transactions t
                            join date d
                            on t.order_date = d.date
                            group by d.year
                            ''',conn)

df = pd.DataFrame(query)
print(f'The revenue for the year {df.loc[0,'year']} was {int(df.loc[0,'total_revenue']):,}')
print(f'The revenue for the year {df.loc[1,'year']} was {int(df.loc[1,'total_revenue']):,}')
print(f'The revenue for the year {df.loc[2,'year']} was {int(df.loc[2,'total_revenue']):,}')
print(f'The revenue for the year {df.loc[3,'year']} was {int(df.loc[3,'total_revenue']):,}')
```

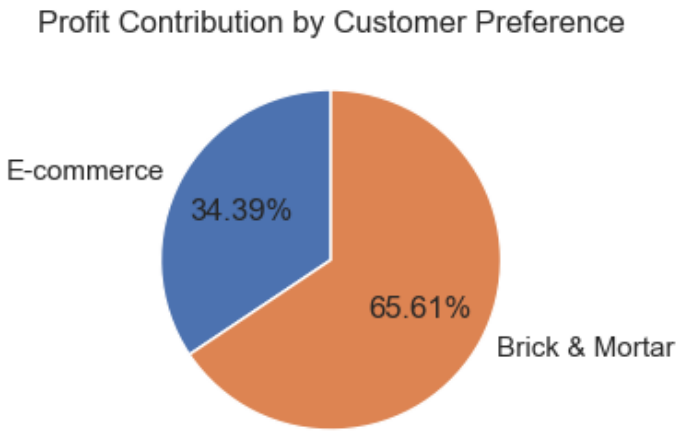
```
The revenue for the year 2017 was 92,881,903
The revenue for the year 2018 was 413,687,163
The revenue for the year 2019 was 336,019,102
The revenue for the year 2020 was 142,224,545
```

In [8]:

```
# Profit contribution by customer type
query = pd.read_sql_query('''
                          select customer_type,
                          round((profit / total_profit)*100,2) as profit_perecntage
                          from
                            (select c.customer_type,sum(profit_margin) as profit,total_profit
                            from transactions t
                            join customers c
                            on t.customer_code = c.customer_code
                            cross join
                              (select round(sum(profit_margin),2) as total_profit
                              from transactions) a
                          group by c.customer_type,total_profit)x
                          ''',conn)
df = pd.DataFrame(query)
print(df)

sns.set(style='dark')
plt.figure(figsize=(5,3))
```

```
plt.pie(df['profit_perecntage'],labels=['E-commerce','Brick & Mortar'],autopct='%1.2f%%',startangle=90)
plt.title('Profit Contribution by Customer Preference')
plt.show()
```

```
    customer_type  profit_perecntage
0      E-Commerce              34.39
1  Brick & Mortar              65.61
```

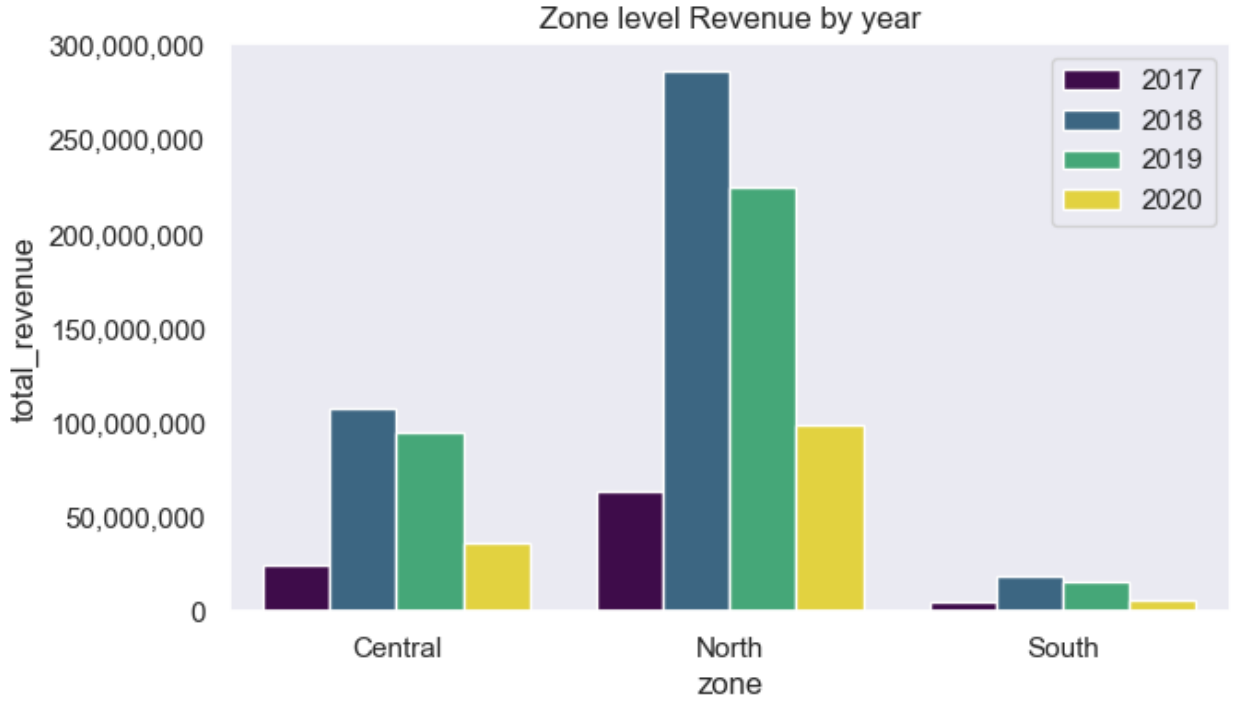



In [9]:

```
# Revenue by zone level for each year
query = pd.read_sql_query('''
                        select d.year,m.zone,round(sum(sales_amount),0) as total_revenue
                          from transactions t
                          join markets m
                          on t.market_code = m.markets_code
                          join date d
                          on t.order_date = d.date
                          group by d.year,m.zone
                          order by d.year asc''',conn)

df = pd.DataFrame(query)
print(df)
zone_grp = df.groupby(['year','zone'])[['total_revenue']].sum()

plt.figure(figsize=(7,4))
sns.set(style='dark')
plot = sns.barplot(data=zone_grp,x='zone',y='total_revenue',hue='year',errorbar=None,palette='viridis')

formatter = FuncFormatter(lambda x,pos:f'{x:,.0f}')
plot.yaxis.set_major_formatter(formatter)

plt.title('Zone level Revenue by year')
```

```
plt.legend(loc='best',bbox_to_anchor=(1,1))
plt.show()
```

| | year | zone | total_revenue |
|---|---|---|---|
| 0 | 2017 | Central | 24420633.0 |
| 1 | 2017 | North | 63726649.0 |
| 2 | 2017 | South | 4734621.0 |
| 3 | 2018 | Central | 107839601.0 |
| 4 | 2018 | North | 287037445.0 |
| 5 | 2018 | South | 18810117.0 |
| 6 | 2019 | Central | 95362521.0 |
| 7 | 2019 | North | 225201876.0 |
| 8 | 2019 | South | 15454705.0 |
| 9 | 2020 | Central | 36098228.0 |
| 10 | 2020 | North | 99565797.0 |
| 11 | 2020 | South | 6560520.0 |




In [10]:

```
# Retrieve the customer_name and total_revenue who has done the sales more than the average sales
query = pd.read_sql_query('''
                        with sales as
                          (select c.customer_name,sum(t.sales_amount) as total_revenue
                          from transactions t
                          join customers c
                          on t.customer_code = c.customer_code
```

```
                          group by c.customer_name),
                        avg_sales as
                          (select round(avg(total_revenue),0) as average_revenue from sales)
                        select customer_name,total_revenue,average_revenue
                          from sales s
                          join avg_sales a
                          on s.total_revenue > a.average_revenue
                        ''',conn)

df = pd.DataFrame(query)
df
```

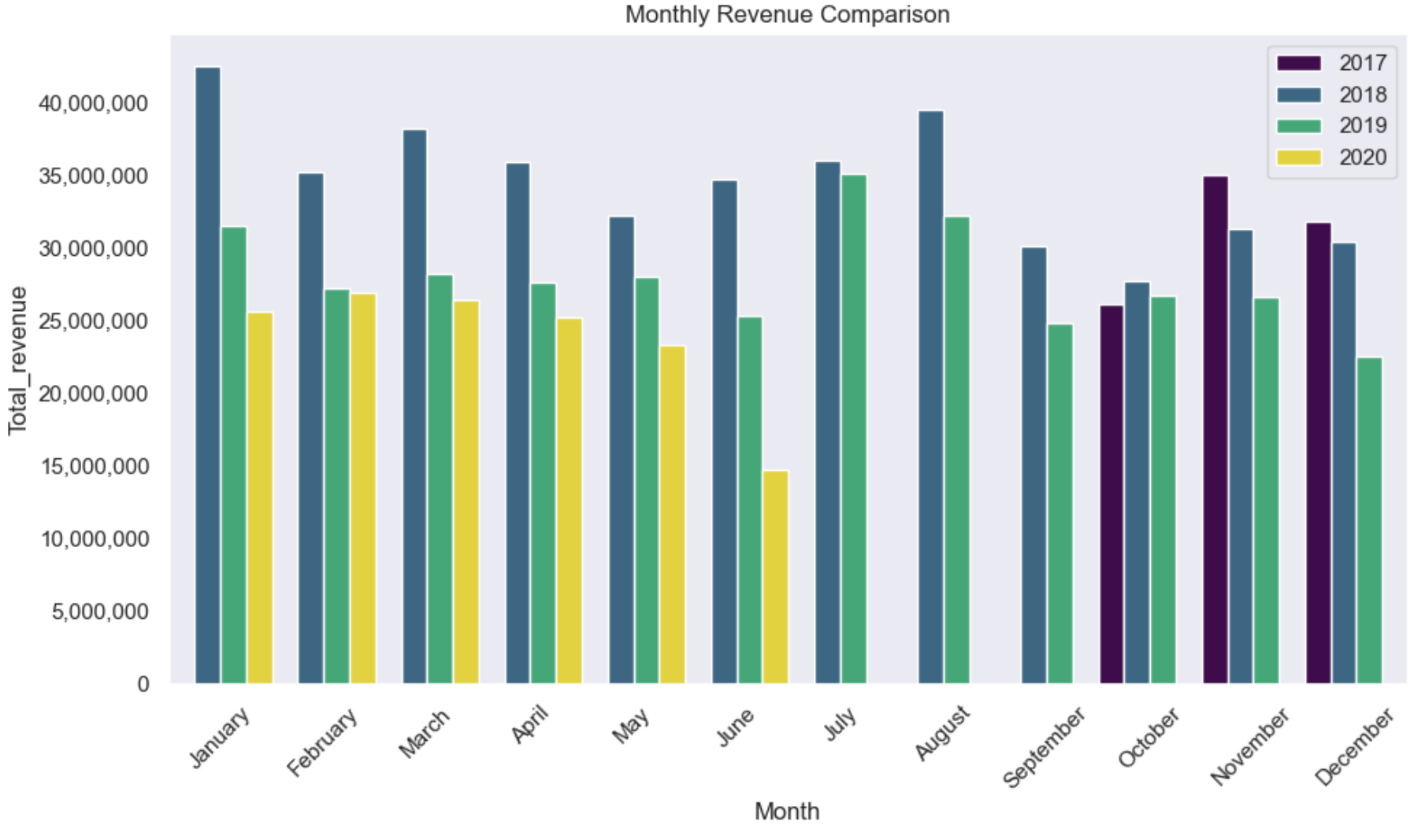
Out[10]:

| | customer_name | total_revenue | average_revenue |
|---|---|---|---|
| 0 | Nixon | 43893083.0 | 25916124.0 |
| 1 | Electricalslytical | 49644189.0 | 25916124.0 |
| 2 | Info Stores | 35100033.0 | 25916124.0 |
| 3 | Electricalsara Stores | 413333588.0 | 25916124.0 |
| 4 | Premium Stores | 44905916.0 | 25916124.0 |
| 5 | Excel Stores | 49115620.0 | 25916124.0 |
| 6 | Control | 31771997.0 | 25916124.0 |
| 7 | Surge Stores | 28648916.0 | 25916124.0 |

In [11]:

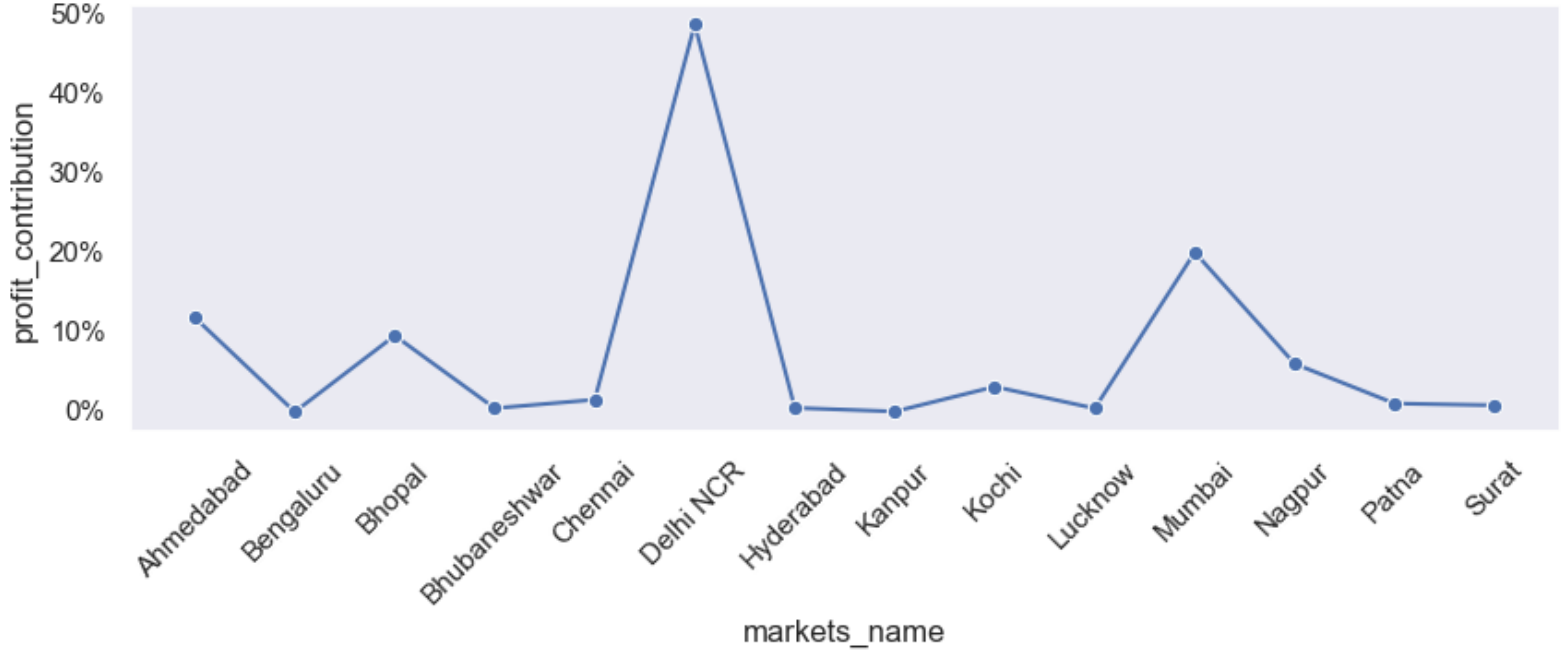
```
# Monthly revenue comparison for the each year
query = pd.read_sql_query('''
                    select year,month,total_revenue
                    from
                        (select d.year,month(d.date) as month_num,
                        monthname(d.date) as month,sum(t.sales_amount) as total_revenue
                        from transactions t
                        join date d
                        on t.order_date = d.date
                        group by d.year,month(d.date),monthname(d.date)
                        order by d.year,month_num)x
                        ''',conn)
df = pd.DataFrame(query)

month_order = ['January','February','March','April','May','June','July','August','September',
               'October','November','December']

df['month'] = pd.Categorical(df['month'],categories=month_order,ordered=True)
```

```
sns.set(style='dark')
plt.figure(figsize=(10,6))
plot = sns.barplot(data=df,x='month',y='total_revenue',hue='year',errorbar=None,palette='viridis',width=1)

plot.yaxis.set_major_formatter(FuncFormatter(lambda x,pos:f'{x:,.0f}'))
plt.xticks(rotation=45)
plt.title('Monthly Revenue Comparison')
plt.legend(loc='best',bbox_to_anchor=(1,1))
plt.xlabel('Month')
plt.ylabel('Total_revenue')
plt.tight_layout()
plt.show()
```



In [12]:

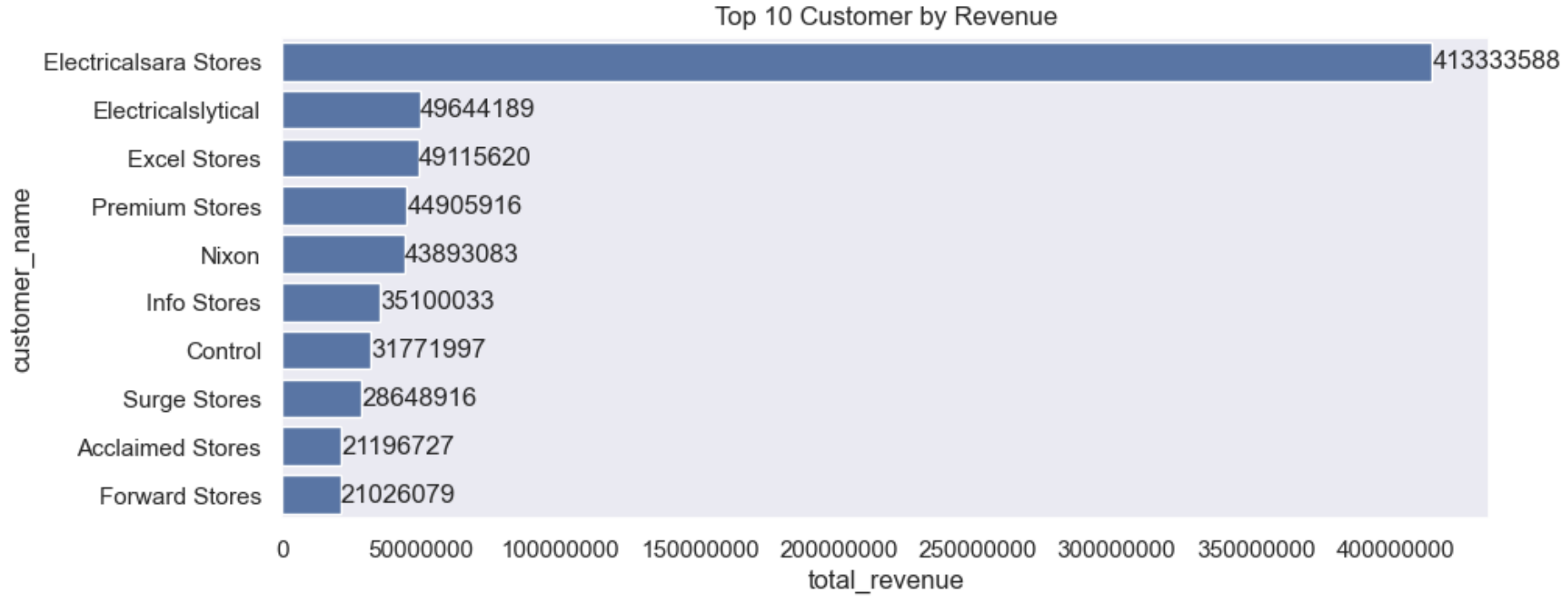
```
# Overall Market profit insights
query = pd.read_sql_query('''
                          select m.markets_name,sum(profit_margin) as profit_contribution
                          from
                          (select * from transactions) t
                          join
                          (select * from markets) m
                          on t.market_code = m.markets_code
                          group by m.markets_name''',conn)

df = pd.DataFrame(query)
perc_grp = df.groupby('markets_name')[['profit_contribution']].sum()
perc_contribution = perc_grp / perc_grp.sum()*100

plt.figure(figsize=(10,3))
plot = sns.lineplot(data = perc_contribution,x='markets_name',y='profit_contribution',marker='o')
plot.yaxis.set_major_formatter(mticks.PercentFormatter())


plt.xticks(rotation=45)
plt.show()
```




In [13]:

```
# Top ten customers with best revenue
query = pd.read_sql_query('''
```

```
                            select c.customer_name,sum(t.sales_amount) as total_revenue
                            from transactions t
                            join customers c
                            on c.customer_code = t.customer_code
                            group by c.customer_name
                            order by total_revenue desc
                            limit 10
                            ''',conn)
df = pd.DataFrame(query)

plt.figure(figsize=(10,4))
plot = sns.barplot(data=df,x='total_revenue',y='customer_name')
plot.bar_label(plot.containers[0],fmt='%.0f')
plot.xaxis.set_major_formatter(FuncFormatter(lambda x,pos:f'{x:.0f}'))
plot.set(title='Top 10 Customer by Revenue')

plt.show()
```




In [15]:

```
conn.close()
```

In [ ]:

```
```