1. Corporation wants to investigate the best-selling item across stores 2. The Best selling item in All Stores 3. Best selling bundled items in stores 4. Best performing stores via sales 5. The best selling items in the best selling stores 6. Worse performing products and its corresponding stores 7. Worse performing stores by sales

In [4]:
```
import pandas as pd
import random
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder
```

In [6]:
```
df = pd.read_csv('sales_data.csv') #Load csv file
baskets_per_store = df.groupby(['StoreID', 'OrderID'])['Product Name'].apply(list)
```

In [120…
```
def get_best_selling_item_per_store(df, store_id):
    store_data = df[df['StoreID'] == store_id]
    best_selling_item = store_data['Product Name'].value_counts().head(1)
    return best_selling_item

random_stores = random.sample(df['StoreID'].unique().tolist(), 10)

best_selling_items_per_store = []

for store_id in random_stores:
    best_selling_item = get_best_selling_item_per_store(df, store_id)
    best_selling_items_per_store.append((store_id, best_selling_item.index[0], best

best_selling_df = pd.DataFrame(best_selling_items_per_store, columns=['StoreID', 'B

top_10_stores = best_selling_df.sort_values(by='SalesCount', ascending=False).head(

for idx, row in top_10_stores.iterrows():
    print(f"Store: {row['StoreID']}    Best selling item: {row['BestSellingItem']}
```

```
Store: 267    Best selling item: Tropicana Orange Juice ..  Sold 79 times
Store: 619    Best selling item: Toll House Cookie Dough ..  Sold 77 times
Store: 28    Best selling item: Apple AirPods Pro ..  Sold 48 times
Store: 587    Best selling item: Organic 2% Milk ..  Sold 42 times
Store: 282    Best selling item: Daves Killer Bread ..  Sold 42 times
Store: 435    Best selling item: Driscolls Blueberries ..  Sold 37 times
Store: 212    Best selling item: Organic 2% Milk ..  Sold 26 times
Store: 707    Best selling item: Daves Killer Bread ..  Sold 22 times
Store: 533    Best selling item: Apple AirPods Pro ..  Sold 16 times
Store: 654    Best selling item: Tropicana Orange Juice ..  Sold 9 times
```

In [144…
```
def get_best_selling_items_across_organization(df):
    return df['Product Name'].value_counts().head(10)

top_selling_items = get_best_selling_items_across_organization(df)

header = "Best Selling Items Across All Stores:"
separator = "=" * 37
print(f"{header}\n{separator}")
```

```
for idx, (item, count) in enumerate(top_selling_items.items(), start=1):
    print(f"{idx}. {item} - Sold {count} times")
```

```
Best Selling Items Across All Stores:
====================================
1. Driscolls Blueberries - Sold 12641 times
2. Goodfellow Grey T-shirt - Sold 12621 times
3. Afflux Type-C - Sold 12575 times
4. Daves Killer Bread - Sold 12564 times
5. Organic 2% Milk - Sold 12531 times
6. Apple AirPods Pro - Sold 12519 times
7. Mobil 1 5W30 Oil - Sold 12458 times
8. Tropicana Orange Juice - Sold 12399 times
9. Alisan Kitchen Mats - Sold 12374 times
10. Toll House Cookie Dough - Sold 12318 times
```

In [149…

```
random_stores = random.sample(baskets_per_store.index.get_level_values(0).unique().

report = []

for store_id in random_stores:
    transactions = baskets_per_store[store_id].tolist()

    te = TransactionEncoder()
    store_df = pd.DataFrame(te.fit_transform(transactions), columns=te.columns_)

    frequent_itemsets = apriori(store_df, min_support=0.05, use_colnames=True)

    bundled_itemsets = frequent_itemsets[frequent_itemsets['itemsets'].apply(lambda
    top_itemsets = bundled_itemsets.nlargest(5, 'support')

    report.append(f"Top 5 Bundled Items for Store {store_id}:")
    report.extend(
        f"{', '.join(row['itemsets'])} - Purchased together in {row['support']*100:
        for _, row in top_itemsets.iterrows()
    )
    report.append("".center(50, "-"))

print("\n".join(report))
```

Top 5 Bundled Items for Store 580:
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 70.00% of orders.
Goodfellow Grey T-shirt, Driscolls Blueberries - Purchased together in 60.00% of ord
ers.
Goodfellow Grey T-shirt, Toll House Cookie Dough - Purchased together in 60.00% of o
rders.
Tropicana Orange Juice, Goodfellow Grey T-shirt - Purchased together in 60.00% of or
ders.
Tropicana Orange Juice, Toll House Cookie Dough - Purchased together in 60.00% of or
ders.

Top 5 Bundled Items for Store 675:
Apple AirPods Pro, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Daves Killer Bread, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Driscolls Blueberries, Alisan Kitchen Mats - Purchased together in 100.00% of order
s.
Goodfellow Grey T-shirt, Alisan Kitchen Mats - Purchased together in 100.00% of orde
rs.
Alisan Kitchen Mats, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.

Top 5 Bundled Items for Store 775:
Daves Killer Bread, Apple AirPods Pro - Purchased together in 100.00% of orders.
Apple AirPods Pro, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.
Apple AirPods Pro, Organic 2% Milk - Purchased together in 100.00% of orders.
Tropicana Orange Juice, Apple AirPods Pro - Purchased together in 100.00% of orders.
Daves Killer Bread, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.

Top 5 Bundled Items for Store 413:
Goodfellow Grey T-shirt, Toll House Cookie Dough - Purchased together in 90.00% of o
rders.
Daves Killer Bread, Goodfellow Grey T-shirt - Purchased together in 80.00% of order
s.
Daves Killer Bread, Toll House Cookie Dough - Purchased together in 80.00% of order
s.
Goodfellow Grey T-shirt, Driscolls Blueberries - Purchased together in 80.00% of ord
ers.
Toll House Cookie Dough, Driscolls Blueberries - Purchased together in 80.00% of ord
ers.

Top 5 Bundled Items for Store 769:
Apple AirPods Pro, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Driscolls Blueberries, Alisan Kitchen Mats - Purchased together in 100.00% of order
s.
Goodfellow Grey T-shirt, Alisan Kitchen Mats - Purchased together in 100.00% of orde
rs.
Alisan Kitchen Mats, Organic 2% Milk - Purchased together in 100.00% of orders.
Toll House Cookie Dough, Alisan Kitchen Mats - Purchased together in 100.00% of orde
rs.

Top 5 Bundled Items for Store 123:
Afflux Type-C, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Afflux Type-C, Apple AirPods Pro - Purchased together in 100.00% of orders.
Afflux Type-C, Driscolls Blueberries - Purchased together in 100.00% of orders.
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 100.00% of orders.
Afflux Type-C, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.

```
Top 5 Bundled Items for Store 771:
Afflux Type-C, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Afflux Type-C, Apple AirPods Pro - Purchased together in 100.00% of orders.
Daves Killer Bread, Afflux Type-C - Purchased together in 100.00% of orders.
Afflux Type-C, Driscolls Blueberries - Purchased together in 100.00% of orders.
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 100.00% of orders.
--------------------------------------------------
Top 5 Bundled Items for Store 827:
Goodfellow Grey T-shirt, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 90.00% of orders.
Afflux Type-C, Mobil 1 5W30 Oil - Purchased together in 90.00% of orders.
Goodfellow Grey T-shirt, Alisan Kitchen Mats - Purchased together in 90.00% of order
s.
Alisan Kitchen Mats, Mobil 1 5W30 Oil - Purchased together in 90.00% of orders.
--------------------------------------------------
Top 5 Bundled Items for Store 64:
Afflux Type-C, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Daves Killer Bread, Afflux Type-C - Purchased together in 100.00% of orders.
Afflux Type-C, Driscolls Blueberries - Purchased together in 100.00% of orders.
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 100.00% of orders.
Afflux Type-C, Mobil 1 5W30 Oil - Purchased together in 100.00% of orders.
--------------------------------------------------
Top 5 Bundled Items for Store 465:
Afflux Type-C, Alisan Kitchen Mats - Purchased together in 100.00% of orders.
Afflux Type-C, Apple AirPods Pro - Purchased together in 100.00% of orders.
Daves Killer Bread, Afflux Type-C - Purchased together in 100.00% of orders.
Afflux Type-C, Driscolls Blueberries - Purchased together in 100.00% of orders.
Afflux Type-C, Goodfellow Grey T-shirt - Purchased together in 100.00% of orders.
--------------------------------------------------
```

In [150…

```python
df['Product Price'] = df['Product Price'].replace({'$': '', ',': ''}, regex=True).a

total_sales_per_order = df.groupby(['StoreID', 'OrderID'])['Product Price'].sum().r

total_sales_per_store = total_sales_per_order.groupby('StoreID')['Product Price'].s

top_10_stores_by_sales = total_sales_per_store.sort_values(by='Product Price', asce

top_10_stores_by_sales['Top'] = range(1, len(top_10_stores_by_sales) + 1)

for idx, row in top_10_stores_by_sales.iterrows():
    print(f"Top {int(row['Top'])}: Store {int(row['StoreID'])}   Total Sales: ${row
```

```
Top 1: Store 576   Total Sales: $34648.27
Top 2: Store 39   Total Sales: $32312.80
Top 3: Store 165   Total Sales: $30216.95
Top 4: Store 681   Total Sales: $30139.26
Top 5: Store 779   Total Sales: $29844.75
Top 6: Store 692   Total Sales: $29309.16
Top 7: Store 611   Total Sales: $28150.82
Top 8: Store 803   Total Sales: $25567.10
Top 9: Store 816   Total Sales: $25512.17
Top 10: Store 103   Total Sales: $25087.08
```

In [151…

```python
best_selling_items_per_store = df.groupby(['StoreID', 'Product Name']).size().reset
```

```
best_selling_items_per_store = best_selling_items_per_store.sort_values(by=['StoreI

top_best_selling_items = best_selling_items_per_store.drop_duplicates('StoreID', ke

top_10_stores = top_best_selling_items.sort_values(by='Count', ascending=False).hea

top_10_stores['Rank'] = range(1, len(top_10_stores) + 1)

for idx, row in top_10_stores.iterrows():
    print(f"Top {row['Rank']}: Store {row['StoreID']}   Best-Selling Item: {row['Pr
```

```
Top 1: Store 576   Best-Selling Item: Mobil 1 5W30 Oil with 158 purchases
Top 2: Store 39   Best-Selling Item: Mobil 1 5W30 Oil with 145 purchases
Top 3: Store 681   Best-Selling Item: Toll House Cookie Dough with 139 purchases
Top 4: Store 165   Best-Selling Item: Alisan Kitchen Mats with 122 purchases
Top 5: Store 779   Best-Selling Item: Apple AirPods Pro with 118 purchases
Top 6: Store 692   Best-Selling Item: Alisan Kitchen Mats with 118 purchases
Top 7: Store 611   Best-Selling Item: Organic 2% Milk with 116 purchases
Top 8: Store 729   Best-Selling Item: Afflux Type-C with 111 purchases
Top 9: Store 761   Best-Selling Item: Afflux Type-C with 109 purchases
Top 10: Store 776   Best-Selling Item: Goodfellow Grey T-shirt with 108 purchases
```

In [152…

```
product_report = []
product_report.append("\nBottom 3 Performing Products by Sales:")
product_report.append("-" * 50)

worst_products = df['Product Name'].value_counts().tail(3)
for product, sales in worst_products.items():
    stores_with_product = df[df['Product Name'] == product]['StoreID'].unique()[:5]
    stores_list = ", ".join(map(str, stores_with_product))
    product_report.append(f"{product} - Sold {sales} times in stores: {stores_list}

store_report = []
store_report.append("\nTop 10 Worst-Performing Stores by Total Sales:")
store_report.append("-" * 50)

total_sales_per_store = df.groupby('StoreID').size()
worse_stores = total_sales_per_store.nsmallest(10)

for store, sales in worse_stores.items():
    store_report.append(f"Store {store} - Total Sales: {sales}")

print("\n".join(product_report))
```

```
Bottom 3 Performing Products by Sales:
--------------------------------------------------
Tropicana Orange Juice - Sold 12399 times in stores: 460, 649, 178, 819, 743
Alisan Kitchen Mats - Sold 12374 times in stores: 460, 649, 178, 819, 743
Toll House Cookie Dough - Sold 12318 times in stores: 460, 649, 178, 819, 743
```

In [153…

```
product_report = []
product_report.append("\nBottom 3 Performing Products by Sales:")
product_report.append("=" * 50)

worst_products = df['Product Name'].value_counts().tail(3)
for product, sales in worst_products.items():
```

```
    stores_with_product = df[df['Product Name'] == product]['StoreID'].unique()[:5]
    stores_list = ", ".join(map(str, stores_with_product))
    product_report.append(f"{product} - Sold {sales} times in stores: {stores_list}")
    product_report.append("-" * 50)

store_report = []
store_report.append("\nTop 10 Worst-Performing Stores by Total Sales:")
store_report.append("-" * 50)

total_sales_per_store = df.groupby('StoreID').size()
worse_stores = total_sales_per_store.nsmallest(10)

for store, sales in worse_stores.items():
    store_report.append(f"Store {store} - Total Sales: {sales}")


print("\n".join(store_report))
```

```
Top 10 Worst-Performing Stores by Total Sales:
--------------------------------------------------
Store 174 - Total Sales: 42
Store 178 - Total Sales: 42
Store 529 - Total Sales: 47
Store 35 - Total Sales: 48
Store 825 - Total Sales: 49
Store 50 - Total Sales: 50
Store 222 - Total Sales: 52
Store 3 - Total Sales: 54
Store 654 - Total Sales: 54
Store 302 - Total Sales: 58
```