In [1]:
```
#Razat Siwakoti (A00046635)
#DMV302 - Assessment 2
#Shopping.ipynb created on Jupyter notebook


#source: Nguyen C. (2022)
#https://towardsdatascience.com/introduction-to-simple-association-rules-mining-for-mark
```

In [2]:
```
#import necessary libraries
import numpy as np
import csv
```

In [3]:
```
# Load the dataset from shoppingtransactions.csv
def load_dataset():
    with open('shoppingtransactions.csv', 'r') as file:
        reader = csv.reader(file)
        dataset = [row for row in reader]
    return dataset
```

In [4]:
```
# Calculate and print support of an item or set of two items
def calculate_support(dataset, items):
    items_count = 0
    # Count transactions containing all specified items
    for transaction in dataset:
        if all(item in transaction for item in items):
            items_count += 1
    support = items_count / len(dataset)
    print(f"Support of {', '.join(items)}: {support:.2%}")
```

In [5]:
```
# Calculate and print confidence of an item or two items
def calculate_confidence(dataset, items, condition):
    items_count = 0
    condition_count = 0
    # Count transactions containing all specified items
    for transaction in dataset:
        if all(item in transaction for item in items):
            items_count += 1
             # Count transactions containing both specified items and the condition item
            if all(cond_item in transaction for cond_item in condition):
                condition_count += 1
    confidence = condition_count / items_count if items_count > 0 else 0
    print(f"Confidence of {', '.join(items)} --> {', '.join(condition)}: {confidence:.2%
```

In [6]:
```
# Main application loop
def main():
    #Load the dataset
    dataset = load_dataset()
    # Continuous loop until the user chooses to exit
    while True:
        print("\nCommands:")
        print("1. sup item[,item]")
        print("2. con item[,item] --> item[,item]")
        print("3. exit")

        # Get user input
        user_input = input("Enter your command: ").strip().lower()

        # Process user input
        if user_input == 'exit' or user_input == '3':
            print("Exiting the application. Goodbye!")
            break
        elif user_input.startswith('sup'):
            items = user_input[4:].strip().split(',')
            calculate_support(dataset, items)
        elif user_input.startswith('con'):
            items_condition = user_input[4:].strip().split('-->')
            items = items_condition[0].strip().split(',')
            condition = items_condition[1].strip().split(',')
            calculate_confidence(dataset, items, condition)
        else:
            print("Invalid command. Please try again.")
```

In [ ]:

```
# Entry point for the script
if __name__ == "__main__":
    main()
```

```
Commands:
1. sup item[,item]
2. con item[,item] --> item[,item]
3. exit
```

In [ ]: