**DECISION TABLE TESTING**

**PROBLEM STATEMENT**

Suppose a software component COMMISION has been implemented to automatically compute a commission for salespersons in XYZ-store. If a non-salaried salesperson sells an item that is neither standard nor bonus to someone other than a regular customer, he/she receives a 7% commission, unless the item costs more than $5,000, in which case the commission is 3%. For all salespeople, if a standard item is sold, or if any item is sold to a regular customer, no commission is given. If a salaried salesperson sells a bonus item, he/she receives a 3% commission, unless the item sells for more than $500, in which case he/she receives a flat $30 commission. If a non-salaried salesman sells a bonus item to someone other than a regular customer, he/she receives a 7% commission, unless the item sells for more than $500, in which case he/she receives a flat commission of $65. In all other cases, a salesman receives a 2% commission.

The component accepts five inputs:

* Salesman name
* Salesman type
* Item type
* Customer type
* Item price

**Assumptions:**

• The maximum size of Salesman name is 20 characters.

• Salesman type values: Salaried (S), Non-salaried (NS)

• Item type values: Standard (ST), Bonus (B), General (G)

• Customer type values: Regular (R), Walk-in (W)

• Item price is an integer.

**Sample test cases for the component:**

Test #1: Salesman name=Smith, Salesman type=NS, Item type=ST, Customer type=W, Item price=1400

Test #2: Salesman name=Brown, Salesman type=S, Item type=B, Customer type=R, Item price=200

**Use decision-table based testing to design test cases to test the COMMISION program. Provide a decision table and test cases derived from the decision table.**

Note: In your solution conditions cannot be complex logical expressions.

For example: (Item price ≤ 500) and (Item type=B) is not acceptable as a condition in the decision table.

However, “Item price ≤ 500” is a condition; “Item type=B” is a condition.