

# Decision Tables

## Introduction

A decision table is an excellent tool to use in both testing and requirements management. Essentially it is a structured exercise to formulate requirements when dealing with complex business rules. Decision tables are used to model complicated logic. They can make it easy to see that all possible combinations of conditions have been considered and when conditions are missed, it is easy to see this.

Decision tables can be used in all situations where the outcome depends on the combinations of different choices, and that is usually very often. In many systems there are tons of business rules where decision tables add a lot of value.

Decision tables should best be constructed during system design, since they become useful to both developers and testers. The requirements specialist also becomes more confident that everything that is important is actually documented. If there are no decision tables, testers can create them during test design to be able to write better test cases.

## Decision table

Decision table is a brief visual representation for specifying which actions to perform depending on given conditions. The information represented in decision tables can also be represented as decision trees or in a programming language using if-then-else and switch-case statements.

A decision table is a good way to settle with different combination inputs with their corresponding outputs and also called cause-effect table. Reason to call cause-effect table is a related logical diagramming technique called cause-effect graphing that is basically used to obtain the decision table.

## Importance of Decision Table:

- Decision tables are very much helpful in test design technique.
- It helps testers to search the effects of combinations of different inputs and other software states that must correctly implement business rules.
- It provides a regular way of stating complex business rules, that is helpful for developers as well as for testers.
- It assists in development process with developer to do a better job. Testing with all combination might be impractical.

- A decision table is basically an outstanding technique used in both testing and requirements management.

## Steps to create decision tables

### Step 1 – Analyze the requirement and create the first column

Requirement: “Withdrawal is granted if requested amount is covered by the balance or if the customer is granted credit to cover the withdrawal amount”.

Express conditions and resulting actions in a list so that they are either TRUE or FALSE. In this case there are two conditions, “withdrawal amount  $\leq$  balance” and “credit granted”. There is one result, the withdrawal is granted.

<b>Conditions</b>
Withdrawal Amount $\leq$ Balance
Credit granted
<b>Actions</b>
Withdrawal granted

### Step 2: Add Columns

Calculate how many columns are needed in the table. The number of columns depends on the number of conditions and the number of alternatives for each condition. If there are two conditions and each condition can be either true or false, you need 4 columns. If there are three conditions there will be 8 columns and so on.

Mathematically, the number of columns is 2 conditions. In this case  $2^2 = 4$  columns.

Number of columns that is needed:

Number of Conditions	Number of Columns
1	2
2	4
3	8
4	16
5	32

The bottom line is that you should create more smaller decision tables instead of fewer larger ones, otherwise you run the risk of the decision tables being so large as to be unmanageable. Test the technique by picking areas with fewer business rules.

Now is the time to fill in the T (TRUE) and F (FALSE) for the conditions. How do you do that? The simplest is to say that it should look like this:

Row 1: TF

Row 2: TTFF

Row 3: TTTTFFFF

For each row, there is twice as many T and F as the previous line.

Repeat the pattern above from left to right for the entire row. In other words, for a table with 8 columns, the first row will read TFTFTFTF, the second row will read TTFFTTFF and the third row will read TTTTFFFF.

Conditions				
Withdrawal Amount <= Balance	T	F	T	F
Credit granted	T	T	F	F
Actions				
Withdrawal granted				

Tip: There are Excel templates to fill in decision tables if you need help.

### Step 3: Reduce the table

Mark insignificant values with “-“. If the requested amount is less than or equal to the account balance it does not matter if credit is granted. In the next step, you can delete the columns that have become identical.

Conditions				
Withdrawal Amount <= Balance	T	F	T	F
Credit granted	-	T	-	F
Actions				
Withdrawal granted				

Check for invalid combinations. Invalid combinations are those that cannot happen, for example, that someone is both an infant and senior. Mark them somehow, e.g. with “X”. In this example, there are no invalid combinations.

Finish by removing duplicate columns. In this case, the first and third column are equal, therefore one of them is removed.

#### Step 4: Determine actions

Enter actions for each column in the table. You will be able to find this information in the requirement. Name the columns (the rules). They may be named R1/Rule 1, R2/Rule 2 and so on, but you can also give them more descriptive names.

Conditions			
Withdrawal Amount <= Balance	T	F	F
Credit granted	-	T	F
Actions			
Withdrawal granted	T	T	F

#### Step 5: Write test cases

Write test cases based on the table. At least one test case per column gives full coverage of all business rules.

Test case for R1: balance = 200, requested withdrawal = 200. Expected result: withdrawal granted.

Test case for R2: balance = 100, requested withdrawal = 200, credit granted. Expected result: withdrawal granted.

Test case for R3: balance = 100, requested withdrawal = 200, no credit. Expected Result: withdrawal denied.

#### Advantages of Decision Tables

1. When the conditions are numerous, then the decision table helps to visualize the outcomes of a situation.
2. Decision tables summarize all the outcomes of a situation & suggest suitable actions
3. They provide more compact documentation

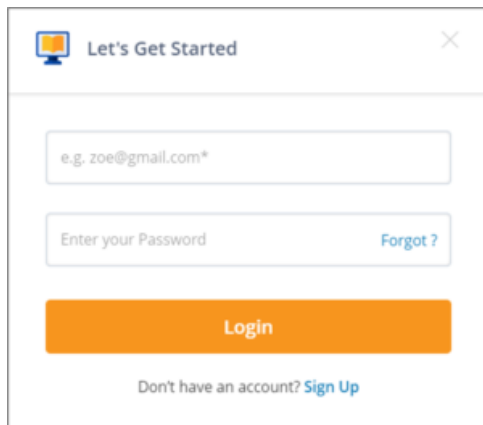
4. Decision tables can be changed easily.
5. Decision table has a standard format.

## Disadvantages of Decision Tables

1. Decision tables only present a partial solution
2. Do not depict the flow of logic of a solution
3. Decision tables are quite far away from high-level languages

## Example (Way to use Decision Table):

A Decision Table is a tabular representation of inputs versus rules, cases or test conditions. Let's take an example and see how to create a decision table for a login screen:



The condition states that if the user provides the correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Username	F	T	F	T
Password	F	F	T	T
Output	E	E	E	H

In the above example,

T – Correct username/password

F – Wrong username/password

E – Error message is displayed

H – Home screen is displayed

Now let's understand the interpretation of the above cases:

Case 1 – Username and password both were wrong. The user is shown an error message.

Case 2 – Username was correct, but the password was wrong. The user is shown an error message.

Case 3 – Username was wrong, but the password was correct. The user is shown an error message.

Case 4 – Username and password both were correct, and the user is navigated to the homepage.

So, this was an example of building a decision table in software testing.