let

// Step 1: Create Month × Scenario cross join

MonthRange = List.Numbers(1, 60, 1),

MonthTable = Table.FromList(MonthRange, Splitter.SplitByNothing(), null, null, ExtraValues.Error),

#"Renamed Month Column" = Table.RenameColumns(MonthTable,{{"Column1", "Month"}}),

// Get scenarios

ScenariosData = Scenarios,

// Cross join months and scenarios

CrossJoin = Table.AddColumn(#"Renamed Month Column", "Scenario", each ScenariosData),

#"Expanded Scenarios" = Table.ExpandTableColumn(CrossJoin, "Scenario", {"ScenarioName", "ScenarioDisplay", "DefaultMultiplier"}, {"ScenarioName", "ScenarioDisplay", "DefaultMultiplier"}),

// Step 2: Account Calculations

#"Added Accounts" = Table.AddColumn(#"Expanded Scenarios", "Accounts", each

let

CurrentMonth = [Month],

BaseAccounts = Table.SelectRows(InterpolatedInputs, each [InputType] = "Accounts" and [Month] = CurrentMonth){0}[Value],

ScenarioMultiplier = if [ScenarioName] = "Custom" then 1.0 else [DefaultMultiplier]

in

Number.Round(BaseAccounts \* ScenarioMultiplier, 0)

),

#"Added Active Share" = Table.AddColumn(#"Added Accounts", "ActiveShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "ActiveShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added Active Accounts" = Table.AddColumn(#"Added Active Share", "Active\_Accounts", each

Number.Round([Accounts] \* [ActiveShare], 0)

),

#"Added Checking Share" = Table.AddColumn(#"Added Active Accounts", "CheckingShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "CheckingShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added Checking Accounts" = Table.AddColumn(#"Added Checking Share", "Checking\_Accounts", each

Number.Round([Active\_Accounts] \* [CheckingShare], 0)

),

#"Added Saving Share" = Table.AddColumn(#"Added Checking Accounts", "SavingShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "SavingShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added Savings Accounts" = Table.AddColumn(#"Added Saving Share", "Savings\_Accounts", each

Number.Round([Active\_Accounts] \* [SavingShare], 0)

),

// Step 3: INFLOW CALCULATIONS

// ACH Inflows

#"Added ACHinPerActive" = Table.AddColumn(#"Added Savings Accounts", "ACHinPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "ACHinPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added ACHinQuantity" = Table.AddColumn(#"Added ACHinPerActive", "ACHinQuantity", each

[Active\_Accounts] \* [ACHinPerActive]

),

#"Added ACHinRate" = Table.AddColumn(#"Added ACHinQuantity", "ACHinRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "ACHinRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added ACHinAmount" = Table.AddColumn(#"Added ACHinRate", "ACHinAmount", each

[ACHinQuantity] \* [ACHinRate]

),

// RTP Inflows

#"Added RTPinPerActive" = Table.AddColumn(#"Added ACHinAmount", "RTPinPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "RTPinPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added RTPinQuantity" = Table.AddColumn(#"Added RTPinPerActive", "RTPinQuantity", each

[Active\_Accounts] \* [RTPinPerActive]

),

#"Added RTPinRate" = Table.AddColumn(#"Added RTPinQuantity", "RTPinRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "RTPinRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added RTPinAmount" = Table.AddColumn(#"Added RTPinRate", "RTPinAmount", each

[RTPinQuantity] \* [RTPinRate]

),

// Wire Inflows

#"Added WireInPerActive" = Table.AddColumn(#"Added RTPinAmount", "WireInPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "WireInPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added WireInQuantity" = Table.AddColumn(#"Added WireInPerActive", "WireInQuantity", each

[Active\_Accounts] \* [WireInPerActive]

),

#"Added WireInRate" = Table.AddColumn(#"Added WireInQuantity", "WireInRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "WireInRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added WireInAmount" = Table.AddColumn(#"Added WireInRate", "WireInAmount", each

[WireInQuantity] \* [WireInRate]

),

// Total Inflows

#"Added Total Inflows" = Table.AddColumn(#"Added WireInAmount", "Total\_Inflows", each

[ACHinAmount] + [RTPinAmount] + [WireInAmount]

),

// Step 4: OUTFLOW CALCULATIONS

// ACH Outflows

#"Added ACHoutPerActive" = Table.AddColumn(#"Added Total Inflows", "ACHoutPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "ACHoutPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added ACHoutQuantity" = Table.AddColumn(#"Added ACHoutPerActive", "ACHoutQuantity", each

[Active\_Accounts] \* [ACHoutPerActive]

),

#"Added ACHoutShare" = Table.AddColumn(#"Added ACHoutQuantity", "ACHoutShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "ACHoutShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added ACHoutAmount" = Table.AddColumn(#"Added ACHoutShare", "ACHoutAmount", each

[Total\_Inflows] \* [ACHoutShare]

),

#"Added ACHoutRate" = Table.AddColumn(#"Added ACHoutAmount", "ACHoutRate", each

if [ACHoutQuantity] > 0 then [ACHoutAmount] / [ACHoutQuantity] else 0

),

// RTP Outflows

#"Added RTPoutPerActive" = Table.AddColumn(#"Added ACHoutRate", "RTPoutPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "RTPoutPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added RTPoutQuantity" = Table.AddColumn(#"Added RTPoutPerActive", "RTPoutQuantity", each

[Active\_Accounts] \* [RTPoutPerActive]

),

#"Added RTPoutShare" = Table.AddColumn(#"Added RTPoutQuantity", "RTPoutShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "RTPoutShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added RTPoutAmount" = Table.AddColumn(#"Added RTPoutShare", "RTPoutAmount", each

[Total\_Inflows] \* [RTPoutShare]

),

#"Added RTPoutRate" = Table.AddColumn(#"Added RTPoutAmount", "RTPoutRate", each

if [RTPoutQuantity] > 0 then [RTPoutAmount] / [RTPoutQuantity] else 0

),

// Wire Outflows

#"Added WireOutPerActive" = Table.AddColumn(#"Added RTPoutRate", "WireOutPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "WireOutPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added WireOutQuantity" = Table.AddColumn(#"Added WireOutPerActive", "WireOutQuantity", each

[Active\_Accounts] \* [WireOutPerActive]

),

#"Added WireOutShare" = Table.AddColumn(#"Added WireOutQuantity", "WireOutShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "WireOutShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added WireOutAmount" = Table.AddColumn(#"Added WireOutShare", "WireOutAmount", each

[Total\_Inflows] \* [WireOutShare]

),

#"Added WireOutRate" = Table.AddColumn(#"Added WireOutAmount", "WireOutRate", each

if [WireOutQuantity] > 0 then [WireOutAmount] / [WireOutQuantity] else 0

),

// Debit Card Outflows

#"Added DebitCardTransactionsPerActive" = Table.AddColumn(#"Added WireOutRate", "DebitCardTransactionsPerActive", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "DebitCardTransactionsPerActive" and [Month] = CurrentMonth){0}[Value]

),

#"Added DebitCardTransactionsQuantity" = Table.AddColumn(#"Added DebitCardTransactionsPerActive", "DebitCardTransactionsQuantity", each

[Active\_Accounts] \* [DebitCardTransactionsPerActive]

),

#"Added DebitCardTransactionShare" = Table.AddColumn(#"Added DebitCardTransactionsQuantity", "DebitCardTransactionShare", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "DebitCardTransactionShare" and [Month] = CurrentMonth){0}[Value]

),

#"Added DebitCardTransactionAmount" = Table.AddColumn(#"Added DebitCardTransactionShare", "DebitCardTransactionAmount", each

([Total\_Inflows] - [ACHoutAmount] - [RTPoutAmount] - [WireOutAmount]) \* [DebitCardTransactionShare]

),

#"Added DebitCardTransactionRate" = Table.AddColumn(#"Added DebitCardTransactionAmount", "DebitCardTransactionRate", each

if [DebitCardTransactionsQuantity] > 0 then [DebitCardTransactionAmount] / [DebitCardTransactionsQuantity] else 0

),

// Step 5: FINAL CALCULATIONS

// Total Outflows

#"Added Total Outflows" = Table.AddColumn(#"Added DebitCardTransactionRate", "Total\_Outflows", each

[ACHoutAmount] + [RTPoutAmount] + [WireOutAmount] + [DebitCardTransactionAmount]

),

// Net Remaining for Savings

#"Added Net Remaining For Savings" = Table.AddColumn(#"Added Total Outflows", "Net\_Remaining\_For\_Savings", each

[Total\_Inflows] - [Total\_Outflows]

),

// Savings Transfer Rate and Amount

#"Added SavingsTransferRate" = Table.AddColumn(#"Added Net Remaining For Savings", "SavingsTransferRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "SavingsTransferRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added SavingsTransfers" = Table.AddColumn(#"Added SavingsTransferRate", "SavingsTransfers", each

[Net\_Remaining\_For\_Savings] \* [SavingsTransferRate]

),

// Monthly Flows

#"Added Monthly Checking" = Table.AddColumn(#"Added SavingsTransfers", "Monthly\_Checking", each

[Net\_Remaining\_For\_Savings] - [SavingsTransfers]

),

#"Added Monthly Savings" = Table.AddColumn(#"Added Monthly Checking", "Monthly\_Savings", each

[SavingsTransfers]

),

// Usage Rates

#"Added CheckingUsageRate" = Table.AddColumn(#"Added Monthly Savings", "CheckingUsageRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "CheckingUsageRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added SavingsUsageRate" = Table.AddColumn(#"Added CheckingUsageRate", "SavingsUsageRate", each

let CurrentMonth = [Month] in

Table.SelectRows(InterpolatedInputs, each [InputType] = "SavingsUsageRate" and [Month] = CurrentMonth){0}[Value]

),

#"Added CheckingUsage" = Table.AddColumn(#"Added SavingsUsageRate", "CheckingUsage", each

[Monthly\_Checking] \* [CheckingUsageRate]

),

#"Added SavingsUsage" = Table.AddColumn(#"Added CheckingUsage", "SavingsUsage", each

[Monthly\_Savings] \* [SavingsUsageRate]

),

// Final data type cleanup

#"Changed Types" = Table.TransformColumnTypes(#"Added SavingsUsage",{

{"Month", Int64.Type},

{"Accounts", type number},

{"Active\_Accounts", type number},

{"Checking\_Accounts", type number},

{"Savings\_Accounts", type number},

{"ACHinAmount", type number},

{"RTPinAmount", type number},

{"WireInAmount", type number},

{"Total\_Inflows", type number},

{"Total\_Outflows", type number},

{"Net\_Remaining\_For\_Savings", type number},

{"Monthly\_Checking", type number},

{"Monthly\_Savings", type number},

{"CheckingUsage", type number},

{"SavingsUsage", type number}

})

in

#"Changed Types"