let

Source = Folder.Files("[redacted]"),

#"Added Custom" = Table.AddColumn(Source, "Custom", each Csv.Document([Content])),

#"Removed Other Columns" = Table.SelectColumns(#"Added Custom",{"Custom"}),

#"Expanded Custom" = Table.ExpandTableColumn(#"Removed Other Columns", "Custom", {"Column1", "Column2", "Column3", "Column4", "Column5", "Column6", "Column7", "Column8", "Column9", "Column10", "Column11", "Column12", "Column13", "Column14", "Column15", "Column16", "Column17", "Column18", "Column19", "Column20", "Column21", "Column22", "Column23", "Column24", "Column25", "Column26"}, {"Column1", "Column2", "Column3", "Column4", "Column5", "Column6", "Column7", "Column8", "Column9", "Column10", "Column11", "Column12", "Column13", "Column14", "Column15", "Column16", "Column17", "Column18", "Column19", "Column20", "Column21", "Column22", "Column23", "Column24", "Column25", "Column26"}),

#"Promoted Headers" = Table.PromoteHeaders(#"Expanded Custom", [PromoteAllScalars=true]),

//--Data cleaning: changed empty cells to zeroes

#"Replaced Value16" = Table.ReplaceValue(#"Promoted Headers","","0",Replacer.ReplaceValue,{"ScheduleElapsedTime"}),

#"Replaced Value" = Table.ReplaceValue(#"Replaced Value16","","0",Replacer.ReplaceValue,{"ActualDepartureTime"}),

#"Replaced Value1" = Table.ReplaceValue(#"Replaced Value","","0",Replacer.ReplaceValue,{"DepartureDelayInMinutes"}),

#"Replaced Value2" = Table.ReplaceValue(#"Replaced Value1","","0",Replacer.ReplaceValue,{"TaxiOutTime"}),

#"Replaced Value7" = Table.ReplaceValue(#"Replaced Value2","","0",Replacer.ReplaceValue,{"TaxiInTime"}),

#"Replaced Value9" = Table.ReplaceValue(#"Replaced Value7","","0",Replacer.ReplaceValue,{"ActualArrivalTime"}),

#"Replaced Value10" = Table.ReplaceValue(#"Replaced Value9","","0",Replacer.ReplaceValue,{"ArrivalDelayInMinutes"}),

#"Replaced Value11" = Table.ReplaceValue(#"Replaced Value10","","0",Replacer.ReplaceValue,{"ActualElapsedTime"}),

#"Replaced Value12" = Table.ReplaceValue(#"Replaced Value11","","0",Replacer.ReplaceValue,{"CarrierDelayInMinutes"}),

#"Replaced Value13" = Table.ReplaceValue(#"Replaced Value12","","0",Replacer.ReplaceValue,{"WeatherDelayInMinutes"}),

#"Replaced Value14" = Table.ReplaceValue(#"Replaced Value13","","0",Replacer.ReplaceValue,{"NASDelayInMinutes"}),

#"Replaced Value15" = Table.ReplaceValue(#"Replaced Value14","","0",Replacer.ReplaceValue,{"SecurityDelayInMinutes"}),

#"Changed Type" = Table.TransformColumnTypes(#"Replaced Value15",{{"DimAirlineKey", Int64.Type}, {"DimOriginAirportKey", Int64.Type}, {"DimArrivalAirportKey", Int64.Type}, {"DimCancellationReasonKey", Int64.Type}, {"DimDelayLengthKey", Int64.Type}, {"DimDepartureBlockKey", Int64.Type}, {"DimArrivalBlockKey", Int64.Type}, {"DimDistanceGroupKey", Int64.Type}, {"FlightDateKey", Int64.Type}, {"FlightNumber", Int64.Type}, {"ScheduleDepartureTime", Int64.Type}, {"ActualDepartureTime", Int64.Type}, {"DepartureDelayInMinutes", Int64.Type}, {"TaxiOutTime", Int64.Type}, {"TaxiInTime", Int64.Type}, {"ScheduleArrivalTime", Int64.Type}, {"ActualArrivalTime", Int64.Type}, {"ArrivalDelayInMinutes", Int64.Type}, {"ScheduleElapsedTime", Int64.Type}, {"ActualElapsedTime", Int64.Type}, {"DistanceInMiles", Int64.Type}, {"CarrierDelayInMinutes", Int64.Type}, {"WeatherDelayInMinutes", Int64.Type}, {"NASDelayInMinutes", Int64.Type}, {"SecurityDelayInMinutes", Int64.Type}, {"FlightDate", type date}}),

#"Changed Type1" = Table.TransformColumnTypes(#"Changed Type",{{"DimCancellationReasonKey", type text}}),

//--Data cleaning: changed null values to zeroes

#"Replaced Value3" = Table.ReplaceValue(#"Changed Type1",null,0,Replacer.ReplaceValue,{"CarrierDelayInMinutes"}),

#"Replaced Value4" = Table.ReplaceValue(#"Replaced Value3",null,0,Replacer.ReplaceValue,{"WeatherDelayInMinutes"}),

#"Replaced Value5" = Table.ReplaceValue(#"Replaced Value4",null,0,Replacer.ReplaceValue,{"NASDelayInMinutes"}),

#"Replaced Value6" = Table.ReplaceValue(#"Replaced Value5",null,0,Replacer.ReplaceValue,{"SecurityDelayInMinutes"}),

#"Replaced Value8" = Table.ReplaceValue(#"Replaced Value6",null,0,Replacer.ReplaceValue,{"ArrivalDelayInMinutes"}),

//Total delay is the subtotal of the specified delay categories

#"Added Custom2" = Table.AddColumn(#"Replaced Value8", "TotalDelay", each [CarrierDelayInMinutes] + [WeatherDelayInMinutes] + [NASDelayInMinutes] + [SecurityDelayInMinutes]),

//Elasped Variance is the difference between the scheduled length of flight time and the actual length of flight time.

//This shows how much of the delay is due to travel related incidents.

//Negative variance denotes lateness and positive variance denotes being early.

#"Added Custom3" = Table.AddColumn(#"Added Custom2", "ElapsedVar", each [ScheduleElapsedTime]-[ActualElapsedTime]),

//Departure variane is the difference between the scheduled departure time and the actual departure time.

//This shows how much of the flight delay can be explained by slower take off times.

#"Added Custom4" = Table.AddColumn(#"Added Custom3", "DepartVar", each [ScheduleDepartureTime]-[ActualDepartureTime]),

//Arrival variance is the difference between the scheduled arrival time and the actual arrival time.

//Thos shows how much of the flight delay can be explained by other unrelated flight challenges.

#"Added Custom5" = Table.AddColumn(#"Added Custom4", "ArriveVar", each [ScheduleArrivalTime]-[ActualArrivalTime]),

//Total variance is the subtotal of both variance types.

//This shows the difference in expected performance and the actual performance in minutes.

#"Added Custom6" = Table.AddColumn(#"Added Custom5", "TotalVar", each [DepartVar] + [ArriveVar]),

//Flights with no arrival times or elapsed times are marked as cancelled

#"Added Custom8" = Table.AddColumn(#"Added Custom6", "Cancelled", each if [ActualArrivalTime] = 0 or [ActualElapsedTime] = 0 or [ScheduleElapsedTime] = 0 then "Y" else "N"),

//Conditional status is assigned based on delays and variance. No delays or variance are labeled perfect.

#"Added Custom1" = Table.AddColumn(#"Added Custom8", "Status", each if [TotalDelay] = 0 and [TotalVar] = 0 and [DepartVar] = 0 and [ArriveVar] = 0

and [ScheduleDepartureTime] = [ActualDepartureTime] and [ScheduleArrivalTime] = [ActualArrivalTime]

and [ScheduleElapsedTime] = [ActualElapsedTime] and [Cancelled] = "N"

then "Perfect"

else if

//On time flights are flights with differences in scheduled and actual times but no variance

([ScheduleDepartureTime] > [ActualDepartureTime] or [ScheduleDepartureTime] < [ActualDepartureTime]

or [ScheduleArrivalTime] > [ActualArrivalTime] or [ScheduleArrivalTime] < [ActualArrivalTime])

and [TotalVar] = 0 and [Cancelled] = "N"

then "On Time"

else if

//Early flights are flights with no delays and a positive total variance.

([ScheduleDepartureTime] > [ActualDepartureTime] or [ScheduleArrivalTime] > [ActualArrivalTime])

and [TotalDelay] = 0 and [ElapsedVar] >= 0 and [TotalVar] > 0 and [Cancelled] = "N"

then "Early"

else if

//Lateness is determined by negative variance or positive delays.

([ScheduleDepartureTime] < [ActualDepartureTime] or [ScheduleArrivalTime] < [ActualArrivalTime])

and ([TotalDelay] > 0 or [TotalVar] < 0) and [Cancelled] = "N"

then "Late"

else if [Cancelled] = "Y"

then "Cancelled"

else "Early"),

//Conditional delay type is assigned based on delay minutes by category

#"Added Custom7" = Table.AddColumn(#"Added Custom1", "DelayType", each if [CarrierDelayInMinutes] > 0 and [Cancelled] = "N" then "Carrier"

else if [WeatherDelayInMinutes] > 0 and [Cancelled] = "N" then "Weather"

else if [NASDelayInMinutes] > 0 and [Cancelled] = "N" then "NASD"

else if [SecurityDelayInMinutes] > 0 and [Cancelled] = "N" then "Security"

else if ([Status] = "Early" or [Status] = "Perfectly On Time" or[Status] = "On Time" )then "No Interference Delay"

else if ([TotalVar] < 0 or [ElapsedVar] < 0) and ([CarrierDelayInMinutes] = 0 and [WeatherDelayInMinutes] = 0 and [NASDelayInMinutes] = 0 and [SecurityDelayInMinutes] = 0) and [Cancelled] = "N" then "Flight Delay"

else if [Cancelled] = "Y" then "Cancelled"

else

"Not Specified"),

#"Changed Type2" = Table.TransformColumnTypes(#"Added Custom7",{{"DepartureDelayInMinutes", Int64.Type}, {"ArrivalDelayInMinutes", Int64.Type}, {"DistanceInMiles", Int64.Type}, {"CarrierDelayInMinutes", Int64.Type}, {"WeatherDelayInMinutes", Int64.Type}, {"NASDelayInMinutes", Int64.Type}, {"SecurityDelayInMinutes", Int64.Type}, {"TotalDelay", Int64.Type}, {"ElapsedVar", Int64.Type}, {"DepartVar", Int64.Type}, {"ArriveVar", Int64.Type}, {"TotalVar", Int64.Type}, {"DimAirlineKey", type text}, {"DimOriginAirportKey", type text}, {"DimArrivalAirportKey", type text}}),

#"Replaced Value17" = Table.ReplaceValue(#"Changed Type2","1342","1334",Replacer.ReplaceText,{"DimAirlineKey"}),

//--Data cleaning: DimCancellationKey is incorrectly assigned to some rows where the flight was indeed cancelled.

//This is corrected by using a conditional statement to check if a flight was cancelled, has no elasped time, and cancel key of -1. These will be replaced with zeroes instead.

#"Added Custom9" = Table.AddColumn(#"Replaced Value17", "Custom", each if [Cancelled] = "Y" and [DimCancellationReasonKey] = "-1" and [ActualElapsedTime] = 0 then "0" else [DimCancellationReasonKey]),

#"Removed Columns" = Table.RemoveColumns(#"Added Custom9",{"DimCancellationReasonKey"}),

#"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"Custom", "DimCancellationReasonKey"}}),

#"Changed Type3" = Table.TransformColumnTypes(#"Renamed Columns",{{"DimCancellationReasonKey", type text}})

//After creating the new column, the original column is removed and the new one is renamed to keep the same structure.

in

#"Changed Type3"

//if actual elapsed time = 0 then dimcancellation key should equal unknown or zero