[

{

"op": "core/text-transform",

"description": "Text transform on cells in column calltime using expression value.toDate()",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "calltime",

"expression": "value.toDate()",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column timeclose using expression value.toDate()",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "timeclose",

"expression": "value.toDate()",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/column-addition",

"description": "Create column timeSinceCall at index 3 based on column calltime using expression grel:diff(cells['timeclose'].value, value, 'minutes')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "timeSinceCall",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(cells['timeclose'].value, value, 'minutes')",

"onError": "set-to-blank"

},

{

"op": "core/column-rename",

"description": "Rename column timeSinceCall to lengthOfCall",

"oldColumnName": "timeSinceCall",

"newColumnName": "lengthOfCall"

},

{

"op": "core/column-addition",

"description": "Create column days\_since\_jan\_1 at index 3 based on column calltime using expression grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'days')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "days\_since\_jan\_1",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'days')",

"onError": "set-to-blank"

},

{

"op": "core/column-addition",

"description": "Create column hours\_since\_jan\_1 at index 3 based on column calltime using expression grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'hours')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "hours\_since\_jan\_1",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'hours')",

"onError": "set-to-blank"

},

{

"op": "core/column-addition",

"description": "Create column weeks\_since\_jan1 at index 3 based on column calltime using expression grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'weeks')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "weeks\_since\_jan1",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'weeks')",

"onError": "set-to-blank"

},

{

"op": "core/column-rename",

"description": "Rename column lengthOfCall to lengthOfCall(min)",

"oldColumnName": "lengthOfCall",

"newColumnName": "lengthOfCall(min)"

},

{

"op": "core/column-addition",

"description": "Create column months\_since\_jan1 at index 3 based on column calltime using expression grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'months')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "months\_since\_jan1",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(cells['calltime'].value, toDate('01/01/2010'), 'months')",

"onError": "set-to-blank"

},

{

"op": "core/column-rename",

"description": "Rename column months\_since\_jan1 to monthsSinceJan1",

"oldColumnName": "months\_since\_jan1",

"newColumnName": "monthsSinceJan1"

},

{

"op": "core/column-rename",

"description": "Rename column weeks\_since\_jan1 to weeksSinceJan1",

"oldColumnName": "weeks\_since\_jan1",

"newColumnName": "weeksSinceJan1"

},

{

"op": "core/column-rename",

"description": "Rename column hours\_since\_jan\_1 to hoursSinceJan1",

"oldColumnName": "hours\_since\_jan\_1",

"newColumnName": "hoursSinceJan1"

},

{

"op": "core/column-rename",

"description": "Rename column days\_since\_jan\_1 to daysSinceJan1",

"oldColumnName": "days\_since\_jan\_1",

"newColumnName": "daysSinceJan1"

},

{

"op": "core/column-reorder",

"description": "Reorder columns",

"columnNames": [

"﻿X",

"Y",

"calltime",

"timeclose",

"lengthOfCall(min)",

"callsource",

"nature",

"closecode",

"monthsSinceJan1",

"weeksSinceJan1",

"hoursSinceJan1",

"daysSinceJan1",

"block",

"streetname",

"zip",

"crossroad1",

"crossroad2"

]

},

{

"op": "core/column-addition",

"description": "Create column Month at index 3 based on column calltime using expression grel:datePart(value, 'month')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "Month",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:datePart(value, 'month')",

"onError": "set-to-blank"

},

{

"op": "core/column-addition",

"description": "Create column Weekday at index 3 based on column calltime using expression grel:value.datePart('weekday')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "Weekday",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:value.datePart('weekday')",

"onError": "set-to-blank"

},

{

"op": "core/column-addition",

"description": "Create column TimeOfDay at index 3 based on column calltime using expression grel:value.datePart('h')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "TimeOfDay",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:value.datePart('h')",

"onError": "set-to-blank"

},

{

"op": "core/column-addition",

"description": "Create column timeOfDay at index 3 based on column calltime using expression grel:value.datePart('h')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "timeOfDay",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:value.datePart('h')",

"onError": "set-to-blank"

},

{

"op": "core/column-removal",

"description": "Remove column TimeOfDay",

"columnName": "TimeOfDay"

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"JERVEY AVE",

"JERVEY AV"

],

"to": "JERVEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LITTLE JOHN DR",

"LITTLEJOHN DR"

],

"to": "LITTLE JOHN DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DEVEREAUX AVE",

"DEVEREAUX AV"

],

"to": "DEVEREAUX AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"SUMMERVILLE AVE",

"SUMMERVILLE AV"

],

"to": "SUMMERVILLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WALLERTON AVE",

"WALLERTON AV"

],

"to": "WALLERTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MCLERNON TRCE",

"MCLERNON TRC"

],

"to": "MCLERNON TRCE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HINSON AVE",

"HINSON AV"

],

"to": "HINSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PONCE DE LEON AVE",

"PONCE DE LEON AV"

],

"to": "PONCE DE LEON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LINDENDALE AVE",

"LINDENDALE AV"

],

"to": "LINDENDALE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COMSEE LN",

"COMESEE LN"

],

"to": "COMSEE LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LITTLE AV",

"LITTLE AVE"

],

"to": "LITTLE AV"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PATTERSON AVE",

"PATTERSON AV"

],

"to": "PATTERSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MCLEOD RD",

"MCCLEOD RD"

],

"to": "MCCLEOD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"S ENSTON AVE",

"S ENSTON AV"

],

"to": "S ENSTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PENN AVE",

"PENN AV"

],

"to": "PENN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BROCKINGTON AV",

"BROCKINGTON AVE"

],

"to": "BROCKINGTON AV"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HYDRANGEA TRL",

"HYDRANGEA TR"

],

"to": "HYDRANGEA TRL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MOHAWK AVE",

"MOHAWK AV"

],

"to": "MOHAWK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HOFF AVE",

"HOFF AV"

],

"to": "HOFF AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"S ANDERSON AVE",

"S ANDERSON AV"

],

"to": "S ANDERSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAINT JOHNS WOODS PKWY",

"SAINT JOHN WOODS PKWY"

],

"to": "SAINT JOHNS WOODS PKWY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BALFOUR DR",

"BALFOURE DR"

],

"to": "BALFOUR DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HOWLE AVE",

"HOWLE AV"

],

"to": "HOWLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SYCAMORE AVE",

"SYCAMORE AV"

],

"to": "SYCAMORE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WAMBAW AVE",

"WAMBAW AV"

],

"to": "WAMBAW AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"STILES BEE AVE",

"STILES BEE AV"

],

"to": "STILES BEE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AUSTIN AVE",

"AUSTIN AV"

],

"to": "AUSTIN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FORT ROYAL AVE",

"FORT ROYAL AV"

],

"to": "FORT ROYAL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MACQUEEN AVE",

"MACQUEEN AV"

],

"to": "MACQUEEN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PIEDMONT AVE",

"PIEDMONT AV"

],

"to": "PIEDMONT AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GUNN AV",

"GUNN AVE"

],

"to": "GUNN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"THREE OAKS AVE",

"THREE OAKS AV"

],

"to": "THREE OAKS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HAGOOD AVE",

"HAGOOD AV"

],

"to": "HAGOOD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FORBES AVE",

"FORBES AV"

],

"to": "FORBES AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BRAXTON AVE",

"BRAXTON AV"

],

"to": "BRAXTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FARMFIELD AVE",

"FARMFIELD AV"

],

"to": "FARMFIELD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ASHLEY AVE",

"ASHLEY AV"

],

"to": "ASHLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MACLAURA HALL AVE",

"MACLAURA HALL AV"

],

"to": "MACLAURA HALL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWANSON AVE",

"SWANSON AV"

],

"to": "SWANSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MARY ADER AVE",

"MARY ADER AV"

],

"to": "MARY ADER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEMMINGER AVE",

"MEMMINGER AV"

],

"to": "MEMMINGER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CHAMPAIGNE LN",

"CHAMPAIGN LN"

],

"to": "CHAMPAIGNE LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"KINLOCK CT",

"KINLOCH CT"

],

"to": "KINLOCK CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GAMECOCK AVE",

"GAMECOCK AV"

],

"to": "GAMECOCK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AVONDALE AVE",

"AVONDALE AV"

],

"to": "AVONDALE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AUDUBON AV",

"AUDUBON AVE"

],

"to": "AUDUBON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DAVIDSON AVE",

"DAVIDSON AV"

],

"to": "DAVIDSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RUTLEDGE AVE",

"RUTLEDGE AV"

],

"to": "RUTLEDGE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TALL OAK AVE",

"TALL OAK AV"

],

"to": "TALL OAK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OTIS AVE",

"OTIS AV"

],

"to": "OTIS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"12TH AVE",

"12TH AV"

],

"to": "12TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"10TH AVE",

"10TH AV"

],

"to": "10TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HUNLEY AVE",

"HUNLEY AV"

],

"to": "HUNLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WAGENER AVE",

"WAGENER AV"

],

"to": "WAGENER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BELGRADE AVE",

"BELGRADE AV"

],

"to": "BELGRADE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CARLIN AVE",

"CARLIN AV"

],

"to": "CARLIN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CASTLE AVE",

"CASTLE AV"

],

"to": "CASTLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"5TH AVE",

"5TH AV"

],

"to": "5TH AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"FLAT BLOOM DR",

"FLAT BLOOM"

],

"to": "FLAT BLOOM DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAM RITTENBERG BLVD",

"SAM RITTENBERG BLVD RAMP"

],

"to": "SAM RITTENBERG BLVD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WESTMORELAND AVE",

"WESTMORELAND AV"

],

"to": "WESTMORELAND AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"AQUARIUM WHF",

"AQUARIUM WHARF"

],

"to": "AQUARIUM WHARF"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COURT HOUSE SQUARE",

"COURT HOUSE SQ"

],

"to": "COURT HOUSE SQ"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"ACKERMANS CT",

"ACKERMAN CT"

],

"to": "ACKERMAN CT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad1",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad1",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"BARRACUDA ST",

"BARACUDA ST"

],

"to": "BARRACUDA ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RAFER ALY",

"RAFERS ALY"

],

"to": "RAFERS ALY"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column streetname",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "streetname",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"WOODALL CT",

"WOODALL CT/"

],

"to": "WOODALL CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"STONO WATCH DR",

"STONO WATCH DR/"

],

"to": "STONO WATCH DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MARSH GRASS WAY",

"MARSH GRASS WAY/"

],

"to": "MARSH GRASS WAY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RIVER REACH WAY",

"RIVER REACH WAY/"

],

"to": "RIVER REACH WAY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ASHLEY RIVER RD/ASHLEY HALL PLANTATION RD",

"ASHLEY HALL PLANTATION RD/ASHLEY RIVER RD"

],

"to": "ASHLEY RIVER RD/ASHLEY HALL PLANTATION RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"VILLAGE CASTLE PL",

"VILLAGE CASTLE PL/"

],

"to": "VILLAGE CASTLE PL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWYGERT BLVD",

"SWYGERT BLVD/"

],

"to": "SWYGERT BLVD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"KING EDWARD DR",

"KING EDWARD DR/"

],

"to": "KING EDWARD DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"EXECUTIVE HALL RD",

"EXECUTIVE HALL RD/"

],

"to": "EXECUTIVE HALL RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WANDO LANDING ST/WANDO VIEW ST",

"WANDO VIEW ST/WANDO LANDING ST"

],

"to": "WANDO LANDING ST/WANDO VIEW ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ELTON CT",

"ELTON CT/"

],

"to": "ELTON CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"POULNOT LN",

"POULNOT LN/"

],

"to": "POULNOT LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WAMBAW CREEK RD",

"WAMBAW CREEK RD/"

],

"to": "WAMBAW CREEK RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"THOMAS JONES RD",

"THOMAS JONES RD/"

],

"to": "THOMAS JONES RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"APPLEBEE WAY",

"APPLEBEE WAY/"

],

"to": "APPLEBEE WAY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"H ST",

"H ST/"

],

"to": "H ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"KING CHARLES CT",

"KING CHARLES CT/"

],

"to": "KING CHARLES CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CREEK BACK ST",

"CREEK BACK ST/"

],

"to": "CREEK BACK ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"E BAY ST/E BATTERY ST",

"E BATTERY ST/E BAY ST"

],

"to": "E BAY ST/E BATTERY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWAMP FOX LN",

"SWAMP FOX LN/"

],

"to": "SWAMP FOX LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PEELE PL",

"PEELE PL/"

],

"to": "PEELE PL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HONEY HILL RD",

"HONEY HILL RD/"

],

"to": "HONEY HILL RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CITADEL HAVEN DR",

"CITADEL HAVEN DR/"

],

"to": "CITADEL HAVEN DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TURKEY PEN RD",

"TURKEY PEN RD/"

],

"to": "TURKEY PEN RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MAMIE ST",

"MAMIE ST/"

],

"to": "MAMIE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CONE ST",

"CONE ST/"

],

"to": "CONE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ORANGE GROVE RD/ORANGE BRANCH RD",

"ORANGE BRANCH RD/ORANGE GROVE RD"

],

"to": "ORANGE GROVE RD/ORANGE BRANCH RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HIGH ST",

"HIGH ST/"

],

"to": "HIGH ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LITTLE JOHN DR",

"LITTLE JOHN DR/"

],

"to": "LITTLE JOHN DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DEREEF CT",

"DEREEF CT/"

],

"to": "DEREEF CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CERTIFICATE CT",

"CERTIFICATE CT/"

],

"to": "CERTIFICATE CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BATTERY AVE",

"BATTERY AVE/"

],

"to": "BATTERY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OAK POINT RD",

"OAK POINT RD/"

],

"to": "OAK POINT RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"JUSTIN AVE",

"JUSTIN AVE/"

],

"to": "JUSTIN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"JACKWOOD CT",

"JACKWOOD CT/"

],

"to": "JACKWOOD CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"IMMIGRATION ST",

"IMMIGRATION ST/"

],

"to": "IMMIGRATION ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MULMAR ST",

"MULMAR ST/"

],

"to": "MULMAR ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEETING ST RAMP/I26 RAMP",

"MEETING ST RAMP/I26 MEETING ST RAMP"

],

"to": "MEETING ST RAMP/I26 RAMP"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HOWLE AVE",

"HOWLE AVE/"

],

"to": "HOWLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DILLS BLUFF CT",

"DILLS BLUFF CT/"

],

"to": "DILLS BLUFF CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COMMERCIAL ROW",

"COMMERCIAL ROW/"

],

"to": "COMMERCIAL ROW"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"N ROMNEY ST",

"N ROMNEY ST/"

],

"to": "N ROMNEY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BENEFITFOCUS PKWY",

"BENEFITFOCUS PKWY/"

],

"to": "BENEFITFOCUS PKWY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PEAS ISLAND RD",

"PEAS ISLAND RD/"

],

"to": "PEAS ISLAND RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"EXCHANGE LANDING RD",

"EXCHANGE LANDING RD/"

],

"to": "EXCHANGE LANDING RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PHYSICIANS DR",

"PHYSICIANS DR/"

],

"to": "PHYSICIANS DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"E ELLIOTT ST/E BAY ST",

"E BAY ST/E ELLIOTT ST"

],

"to": "E ELLIOTT ST/E BAY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"POINSETT ST",

"POINSETT ST/"

],

"to": "POINSETT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HARBOR COVE LN",

"HARBOR COVE LN/"

],

"to": "HARBOR COVE LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HOPEWELL LN",

"HOPEWELL LN/"

],

"to": "HOPEWELL LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GREENLEAF ST",

"GREENLEAF ST/"

],

"to": "GREENLEAF ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TALL SAIL DR",

"TALL SAIL DR/"

],

"to": "TALL SAIL DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SPORTMAN ISLAND DR",

"SPORTMAN ISLAND DR/"

],

"to": "SPORTMAN ISLAND DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FOURTH ST/",

"FOURTH ST"

],

"to": "FOURTH ST/"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ROPER RD",

"ROPER RD/"

],

"to": "ROPER RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAN MIGUEL RD",

"SAN MIGUEL RD/"

],

"to": "SAN MIGUEL RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ASHLEY HALL RD/ASHLEY RIVER RD",

"ASHLEY RIVER RD/ASHLEY HALL RD"

],

"to": "ASHLEY HALL RD/ASHLEY RIVER RD"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column streetname",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "streetname",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"SPRING ST/SIRES ST",

"SIRES ST/SPRING ST"

],

"to": "SPRING ST/SIRES ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DELAHOW ST/DALTON ST",

"DALTON ST/DELAHOW ST"

],

"to": "DELAHOW ST/DALTON ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BRADY ST/BURNHAM ST",

"BURNHAM ST/BRADY ST"

],

"to": "BRADY ST/BURNHAM ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"E BAY ST/ELLIOTT ST",

"ELLIOTT ST/E BAY ST"

],

"to": "E BAY ST/ELLIOTT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AMHERST ST/AMERICA ST",

"AMERICA ST/AMHERST ST"

],

"to": "AMHERST ST/AMERICA ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OLD FOLLY RD/FOLLY RD",

"OLD FOLLY RD"

],

"to": "OLD FOLLY RD/FOLLY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAROLINA BAY DR/CORNSILK DR",

"CORNSILK DR/CAROLINA BAY DR"

],

"to": "CAROLINA BAY DR/CORNSILK DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DOGWOOD RD/DOUBLE OAK RD",

"DOUBLE OAK RD/DOGWOOD RD"

],

"to": "DOGWOOD RD/DOUBLE OAK RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SUMTER ST/S TRACY ST",

"S TRACY ST/SUMTER ST"

],

"to": "SUMTER ST/S TRACY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RIVERLAND DR/LUCKY RD",

"LUCKY RD/RIVERLAND DR"

],

"to": "RIVERLAND DR/LUCKY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DOUBLE FOX RD/RADCLIFFE PLACE DR",

"RADCLIFFE PLACE DR/DOUBLE FOX RD"

],

"to": "DOUBLE FOX RD/RADCLIFFE PLACE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MARGINAL RD/MCCLEOD RD",

"MCCLEOD RD/MARGINAL RD"

],

"to": "MARGINAL RD/MCCLEOD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CHARLOTTE ST/CONCORD ST",

"CONCORD ST/CHARLOTTE ST"

],

"to": "CHARLOTTE ST/CONCORD ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CARTRIGHT ST/COOKE ST",

"COOKE ST/CARTRIGHT ST"

],

"to": "CARTRIGHT ST/COOKE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HARRIS ST/HANOVER ST",

"HANOVER ST/HARRIS ST"

],

"to": "HARRIS ST/HANOVER ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COURTENAY DR/CHARLESTON CENTER DR",

"CHARLESTON CENTER DR/COURTENAY DR"

],

"to": "COURTENAY DR/CHARLESTON CENTER DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CHADWICK DR/CULBERTSON DR",

"CULBERTSON DR/CHADWICK DR"

],

"to": "CHADWICK DR/CULBERTSON DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WESTWOOD DR/WESTRIDGE CIR",

"WESTRIDGE CIR/WESTWOOD DR"

],

"to": "WESTWOOD DR/WESTRIDGE CIR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SHEFFIELD ST/SALISBURY ST",

"SALISBURY ST/SHEFFIELD ST"

],

"to": "SHEFFIELD ST/SALISBURY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BERESFORD CREEK ST/BELLINGER ST",

"BELLINGER ST/BERESFORD CREEK ST"

],

"to": "BERESFORD CREEK ST/BELLINGER ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MONTAGU ST/MONTAGU CT",

"MONTAGU CT/MONTAGU ST"

],

"to": "MONTAGU ST/MONTAGU CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEETING ST/MOUNT PLEASANT ST",

"MOUNT PLEASANT ST/MEETING ST"

],

"to": "MEETING ST/MOUNT PLEASANT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CANNON ST/CARRERE CT",

"CARRERE CT/CANNON ST"

],

"to": "CANNON ST/CARRERE CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SMYTHE ST/SETTLERS ST",

"SETTLERS ST/SMYTHE ST"

],

"to": "SMYTHE ST/SETTLERS ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ITHECAW CREEK ST/ITHECAW CT",

"ITHECAW CT/ITHECAW CREEK ST"

],

"to": "ITHECAW CREEK ST/ITHECAW CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"I26/I526",

"I526/I26"

],

"to": "I26/I526"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SPRING ST/SAINT PHILIP ST",

"SAINT PHILIP ST/SPRING ST"

],

"to": "SPRING ST/SAINT PHILIP ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CLEMENTS FERRY RD/CAINHOY VILLAGE RD",

"CAINHOY VILLAGE RD/CLEMENTS FERRY RD"

],

"to": "CLEMENTS FERRY RD/CAINHOY VILLAGE RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DOWNING ST/DERBY ST",

"DERBY ST/DOWNING ST"

],

"to": "DOWNING ST/DERBY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"S MARKET ST/STATE ST",

"STATE ST/S MARKET ST"

],

"to": "S MARKET ST/STATE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CHURCH ST/CUMBERLAND ST",

"CUMBERLAND ST/CHURCH ST"

],

"to": "CHURCH ST/CUMBERLAND ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BARRE ST/BULL ST",

"BULL ST/BARRE ST"

],

"to": "BARRE ST/BULL ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SANTA ELENA WAY/SAINT JOHNS PARRISH WAY",

"SAINT JOHNS PARRISH WAY/SANTA ELENA WAY"

],

"to": "SANTA ELENA WAY/SAINT JOHNS PARRISH WAY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CLEMENTS FERRY RD/CAINHOY LANDING RD",

"CAINHOY LANDING RD/CLEMENTS FERRY RD"

],

"to": "CLEMENTS FERRY RD/CAINHOY LANDING RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PALM COVE DR/PINE HARBOR DR",

"PINE HARBOR DR/PALM COVE DR"

],

"to": "PALM COVE DR/PINE HARBOR DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RIVER RD/RUSHLAND LANDING RD",

"RUSHLAND LANDING RD/RIVER RD"

],

"to": "RIVER RD/RUSHLAND LANDING RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CANNON ST/COMING ST",

"COMING ST/CANNON ST"

],

"to": "CANNON ST/COMING ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TRAILMORE DR/TRAILEE DR",

"TRAILEE DR/TRAILMORE DR"

],

"to": "TRAILMORE DR/TRAILEE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OCEAN NEIGHBORS BLVD/OLD MILITARY RD",

"OLD MILITARY RD/OCEAN NEIGHBORS BLVD"

],

"to": "OCEAN NEIGHBORS BLVD/OLD MILITARY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PARKDALE DR/PARK STONE DR",

"PARK STONE DR/PARKDALE DR"

],

"to": "PARKDALE DR/PARK STONE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FOLLY ROAD BLVD/FOLLY RD",

"FOLLY RD/FOLLY ROAD BLVD"

],

"to": "FOLLY ROAD BLVD/FOLLY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEETING ST/MAPLE ST",

"MAPLE ST/MEETING ST"

],

"to": "MEETING ST/MAPLE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HANOVER ST/HAMPDEN CT",

"HAMPDEN CT/HANOVER ST"

],

"to": "HANOVER ST/HAMPDEN CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BEAUFAIN ST/BARRE ST",

"BARRE ST/BEAUFAIN ST"

],

"to": "BEAUFAIN ST/BARRE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"S GRIMBALL RD/SCOTT HILL RD",

"SCOTT HILL RD/S GRIMBALL RD"

],

"to": "S GRIMBALL RD/SCOTT HILL RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MAIN RD/MARSHFIELD RD",

"MARSHFIELD RD/MAIN RD"

],

"to": "MAIN RD/MARSHFIELD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MURRAY WOOD RD",

"MURRAYWOOD RD"

],

"to": "MURRAY WOOD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CLEMENTS FERRY RD/CAINHOY RD",

"CAINHOY RD/CLEMENTS FERRY RD"

],

"to": "CLEMENTS FERRY RD/CAINHOY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BENNETT STOCK LN/BOILLAT LN",

"BOILLAT LN/BENNETT STOCK LN"

],

"to": "BENNETT STOCK LN/BOILLAT LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWEETLEAF LN/SUGARBERRY LN",

"SUGARBERRY LN/SWEETLEAF LN"

],

"to": "SWEETLEAF LN/SUGARBERRY LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RIVERLAND DR/GEORGE GRIFFITH BLVD",

"GEORGE GRIFFITH BLVD/RIVERLAND DR"

],

"to": "RIVERLAND DR/GEORGE GRIFFITH BLVD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LEGARE ST/LAMBOLL ST",

"LAMBOLL ST/LEGARE ST"

],

"to": "LEGARE ST/LAMBOLL ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAROLINA ST/COMING ST",

"COMING ST/CAROLINA ST"

],

"to": "CAROLINA ST/COMING ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWEETLEAF LN/SILVERBELL LN",

"SILVERBELL LN/SWEETLEAF LN"

],

"to": "SWEETLEAF LN/SILVERBELL LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CALHOUN ST/COMING ST",

"COMING ST/CALHOUN ST"

],

"to": "CALHOUN ST/COMING ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SOMERSET CIR/SOUTHGATE DR",

"SOUTHGATE DR/SOMERSET CIR"

],

"to": "SOMERSET CIR/SOUTHGATE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BEE ST/BRAVO ST",

"BRAVO ST/BEE ST"

],

"to": "BEE ST/BRAVO ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BENNETT STOCK LN/BUSHELL LN",

"BUSHELL LN/BENNETT STOCK LN"

],

"to": "BENNETT STOCK LN/BUSHELL LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CORINNE ST/COMING ST",

"COMING ST/CORINNE ST"

],

"to": "CORINNE ST/COMING ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"JERVEY AVE/JOY AVE",

"JOY AVE/JERVEY AVE"

],

"to": "JERVEY AVE/JOY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TRAPMAN ST/TRUMBO ST",

"TRUMBO ST/TRAPMAN ST"

],

"to": "TRAPMAN ST/TRUMBO ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TALL SAIL DR/TREESCAPE DR",

"TREESCAPE DR/TALL SAIL DR"

],

"to": "TALL SAIL DR/TREESCAPE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CANNON ST/CHERRY ST",

"CHERRY ST/CANNON ST"

],

"to": "CANNON ST/CHERRY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BEAR SWAMP RD/BEES FERRY RD",

"BEES FERRY RD/BEAR SWAMP RD"

],

"to": "BEAR SWAMP RD/BEES FERRY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BARRE ST/BENNETT ST",

"BENNETT ST/BARRE ST"

],

"to": "BARRE ST/BENNETT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GLENN MCCONNELL PKWY/GOODWILL WAY",

"GOODWILL WAY/GLENN MCCONNELL PKWY"

],

"to": "GLENN MCCONNELL PKWY/GOODWILL WAY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CORN PLANTERS ST/CAPE ST",

"CAPE ST/CORN PLANTERS ST"

],

"to": "CORN PLANTERS ST/CAPE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"W WILDCAT BLVD/WILLIAM E MURRAY BLVD",

"WILLIAM E MURRAY BLVD/W WILDCAT BLVD"

],

"to": "W WILDCAT BLVD/WILLIAM E MURRAY BLVD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SHEPPARD ST/SAINT PHILIP ST",

"SAINT PHILIP ST/SHEPPARD ST"

],

"to": "SHEPPARD ST/SAINT PHILIP ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TANGLEWOOD AVE/TALL OAK AVE",

"TALL OAK AVE/TANGLEWOOD AVE"

],

"to": "TANGLEWOOD AVE/TALL OAK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SUNNYSIDE AVE/RIVERSIDE DR",

"RIVERSIDE DR/SUNNYSIDE AVE"

],

"to": "SUNNYSIDE AVE/RIVERSIDE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WHITMARSH DR/S SHERWOOD DR",

"S SHERWOOD DR/WHITMARSH DR"

],

"to": "WHITMARSH DR/S SHERWOOD DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PURCELL LN/PUBLISHERS LN",

"PUBLISHERS LN/PURCELL LN"

],

"to": "PURCELL LN/PUBLISHERS LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DUPONT RD/DULSEY RD",

"DULSEY RD/DUPONT RD"

],

"to": "DUPONT RD/DULSEY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CONGRESS ST/CARONDOLET ST",

"CARONDOLET ST/CONGRESS ST"

],

"to": "CONGRESS ST/CARONDOLET ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PARTNER RD/PROSPERITY RD",

"PROSPERITY RD/PARTNER RD"

],

"to": "PARTNER RD/PROSPERITY RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BERESFORD RUN/BLUE SKY LN",

"BLUE SKY LN/BERESFORD RUN"

],

"to": "BERESFORD RUN/BLUE SKY LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SHELDON RD/SAINT ANDREWS BLVD",

"SAINT ANDREWS BLVD/SHELDON RD"

],

"to": "SHELDON RD/SAINT ANDREWS BLVD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GRAND COUNCIL ST/GIBBON ST",

"GIBBON ST/GRAND COUNCIL ST"

],

"to": "GRAND COUNCIL ST/GIBBON ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FOXCROFT RD",

"FOX CROFT RD"

],

"to": "FOXCROFT RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MAIN RD/MARY ANN POINT RD",

"MARY ANN POINT RD/MAIN RD"

],

"to": "MAIN RD/MARY ANN POINT RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BLAKEWAY ST/BALTIMORE ST",

"BALTIMORE ST/BLAKEWAY ST"

],

"to": "BLAKEWAY ST/BALTIMORE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BRADY ST/BOUNTY ST",

"BOUNTY ST/BRADY ST"

],

"to": "BRADY ST/BOUNTY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CALHOUN ST/CONCORD ST",

"CONCORD ST/CALHOUN ST"

],

"to": "CALHOUN ST/CONCORD ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEETING ST/MARKET ST",

"MARKET ST/MEETING ST"

],

"to": "MEETING ST/MARKET ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAVANNAH HWY RAMP/SAINT ANDREWS BLVD RAMP",

"SAINT ANDREWS BLVD RAMP/SAVANNAH HWY RAMP"

],

"to": "SAVANNAH HWY RAMP/SAINT ANDREWS BLVD RAMP"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"E BAY ST/EXCHANGE ST",

"EXCHANGE ST/E BAY ST"

],

"to": "E BAY ST/EXCHANGE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WATEREE DR/WIMBEE DR",

"WIMBEE DR/WATEREE DR"

],

"to": "WATEREE DR/WIMBEE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GEORGE ST/GLEBE ST",

"GLEBE ST/GEORGE ST"

],

"to": "GEORGE ST/GLEBE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ALBERTA AVE/AMELIA AVE",

"AMELIA AVE/ALBERTA AVE"

],

"to": "ALBERTA AVE/AMELIA AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MAIN ST/MINNIE ST",

"MINNIE ST/MAIN ST"

],

"to": "MAIN ST/MINNIE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CHURCH ST/CHALMERS ST",

"CHALMERS ST/CHURCH ST"

],

"to": "CHURCH ST/CHALMERS ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WELLINGTON DR/WINBORN DR",

"WINBORN DR/WELLINGTON DR"

],

"to": "WELLINGTON DR/WINBORN DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CESSNA AVE/CULVER AVE",

"CULVER AVE/CESSNA AVE"

],

"to": "CESSNA AVE/CULVER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HANOVER ST/HUGER ST",

"HUGER ST/HANOVER ST"

],

"to": "HANOVER ST/HUGER ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CONCORD ST/CUMBERLAND ST",

"CUMBERLAND ST/CONCORD ST"

],

"to": "CONCORD ST/CUMBERLAND ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RIVERDALE DR/ROSEDALE DR",

"ROSEDALE DR/RIVERDALE DR"

],

"to": "RIVERDALE DR/ROSEDALE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FOLLY RD/FORT JOHNSON RD",

"FORT JOHNSON RD/FOLLY RD"

],

"to": "FOLLY RD/FORT JOHNSON RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEETING ST/MARY ST",

"MARY ST/MEETING ST"

],

"to": "MEETING ST/MARY ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DAWN HILL DR/DONCASTER DR",

"DONCASTER DR/DAWN HILL DR"

],

"to": "DAWN HILL DR/DONCASTER DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BOHICKET RD/BERRYHILL RD",

"BERRYHILL RD/BOHICKET RD"

],

"to": "BOHICKET RD/BERRYHILL RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAROLINA BAY DR/CANDELA GROVE DR",

"CANDELA GROVE DR/CAROLINA BAY DR"

],

"to": "CAROLINA BAY DR/CANDELA GROVE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CORN PLANTERS ST/CARTRIGHT ST",

"CARTRIGHT ST/CORN PLANTERS ST"

],

"to": "CORN PLANTERS ST/CARTRIGHT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ANCRUM HILL RD/ASHLEY RIVER RD",

"ASHLEY RIVER RD/ANCRUM HILL RD"

],

"to": "ANCRUM HILL RD/ASHLEY RIVER RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAN MIGUEL RD/SAVAGE RD",

"SAVAGE RD/SAN MIGUEL RD"

],

"to": "SAN MIGUEL RD/SAVAGE RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AMERICA ST/ALEXANDER ST",

"ALEXANDER ST/AMERICA ST"

],

"to": "AMERICA ST/ALEXANDER ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AKERS RD/ASHLEY RIVER RD",

"ASHLEY RIVER RD/AKERS RD"

],

"to": "AKERS RD/ASHLEY RIVER RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAMPBELL DR/CULBERTSON DR",

"CULBERTSON DR/CAMPBELL DR"

],

"to": "CAMPBELL DR/CULBERTSON DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BROAD ST/BARRE ST",

"BARRE ST/BROAD ST"

],

"to": "BROAD ST/BARRE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BARFIELD ST/BALTIMORE ST",

"BALTIMORE ST/BARFIELD ST"

],

"to": "BARFIELD ST/BALTIMORE ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WESTWAY DR/WESTWOOD DR",

"WESTWOOD DR/WESTWAY DR"

],

"to": "WESTWAY DR/WESTWOOD DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAINHOY RD/CHARITY CHURCH RD",

"CHARITY CHURCH RD/CAINHOY RD"

],

"to": "CAINHOY RD/CHARITY CHURCH RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OCEAN VIEW RD",

"OCEANVIEW RD"

],

"to": "OCEAN VIEW RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ORANGE GROVE RD/RIVER FRONT DR",

"RIVER FRONT DR/ORANGE GROVE RD"

],

"to": "ORANGE GROVE RD/RIVER FRONT DR"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column streetname",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "streetname",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"BOEING AVE",

"BOEING AV"

],

"to": "BOEING AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"INLAND AVE",

"INLAND AV"

],

"to": "INLAND AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BURLINGTON AVE",

"BURLINGTON AV"

],

"to": "BURLINGTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PIERPONT AVE",

"PIERPONT AV"

],

"to": "PIERPONT AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DARLINGTON AVE",

"DARLINGTON AV"

],

"to": "DARLINGTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"N LANDER LN",

"N LANDERS LN"

],

"to": "N LANDER LN"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAVANNAH HWY/MCLEOD RD",

"SAVANNAH HWY/MCCLEOD RD"

],

"to": "SAVANNAH HWY/MCLEOD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HOWLE AVE",

"HOWLE AV"

],

"to": "HOWLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BALFOUR DR",

"BALFOURE DR"

],

"to": "BALFOUR DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ARMSTRONG AVE",

"ARMSTRONG AV"

],

"to": "ARMSTRONG AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COKER AVE",

"COKER AV"

],

"to": "COKER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SYCAMORE AVE",

"SYCAMORE AV"

],

"to": "SYCAMORE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"JOY AVE",

"JOY AV"

],

"to": "JOY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AUSTIN AVE",

"AUSTIN AV"

],

"to": "AUSTIN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ETIWAN AVE",

"ETIWAN AV"

],

"to": "ETIWAN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GUNN AVE",

"GUNN AV"

],

"to": "GUNN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"THREE OAKS AVE",

"THREE OAKS AV"

],

"to": "THREE OAKS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TANNER TRL",

"TANNER TR"

],

"to": "TANNER TRL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HAGOOD AVE",

"HAGOOD AV"

],

"to": "HAGOOD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FORBES AVE",

"FORBES AV"

],

"to": "FORBES AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CURTISS AVE",

"CURTISS AV"

],

"to": "CURTISS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FARMFIELD AVE",

"FARMFIELD AV"

],

"to": "FARMFIELD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ASHLEY AVE",

"ASHLEY AV"

],

"to": "ASHLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CARNEGIE AVE",

"CARNEGIE AV"

],

"to": "CARNEGIE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PITTSBURGH AVE",

"PITTSBURGH AV"

],

"to": "PITTSBURGH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"GAMECOCK AVE",

"GAMECOCK AV"

],

"to": "GAMECOCK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SPRUILL AVE",

"SPRUILL AV"

],

"to": "SPRUILL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RUTLEDGE AVE",

"RUTLEDGE AV"

],

"to": "RUTLEDGE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FOLLY RD/HUNLEY AVE",

"FOLLY RD/HUNLEY AV"

],

"to": "FOLLY RD/HUNLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DEVONS AV",

"DEVONS AVE"

],

"to": "DEVONS AV"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BARRACUDA ST",

"BARACUDA ST"

],

"to": "BARRACUDA ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SWIFT AVE",

"SWIFT AV"

],

"to": "SWIFT AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HUNLEY AVE",

"HUNLEY AV"

],

"to": "HUNLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CARLIN AV",

"CARLIN AVE"

],

"to": "CARLIN AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad2",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad2",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"JERVEY AVE",

"JERVEY AV"

],

"to": "JERVEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CRAVEN AVE",

"CRAVEN AV"

],

"to": "CRAVEN AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad2",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad2",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"SUMMERVILLE AVE",

"SUMMERVILLE AV"

],

"to": "SUMMERVILLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HIGHLAND AVE",

"HIGHLAND AV"

],

"to": "HIGHLAND AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LIMEBAKER ST",

"LIMBAKER ST"

],

"to": "LIMBAKER ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"KRUGER AVE",

"KRUGER AV"

],

"to": "KRUGER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"S ENSTON AVE",

"S ENSTON AV"

],

"to": "S ENSTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PENN AVE",

"PENN AV"

],

"to": "PENN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HYDRANGEA TRL",

"HYDRANGEA TR"

],

"to": "HYDRANGEA TRL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PIERPONT AVE",

"PIERPONT AV"

],

"to": "PIERPONT AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DARLINGTON AVE",

"DARLINGTON AV"

],

"to": "DARLINGTON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PAULINE AVE",

"PAULINE AV"

],

"to": "PAULINE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAINT JOHNS WOODS PKWY",

"SAINT JOHN WOODS PKWY"

],

"to": "SAINT JOHNS WOODS PKWY"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BALFOURE DR",

"BALFOUR DR"

],

"to": "BALFOURE DR"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CAVALIER AVE",

"CAVALIER AV"

],

"to": "CAVALIER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"6TH AVE",

"6TH AV"

],

"to": "6TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HASELL ST",

"HASWELL ST"

],

"to": "HASELL ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SYCAMORE AVE",

"SYCAMORE AV"

],

"to": "SYCAMORE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WAMBAW AVE",

"WAMBAW AV"

],

"to": "WAMBAW AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"STILES BEE AVE",

"STILES BEE AV"

],

"to": "STILES BEE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MCHENRY AVE",

"MCHENRY AV"

],

"to": "MCHENRY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BATTLEWOOD AV",

"BATTLEWOOD AVE"

],

"to": "BATTLEWOOD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FORT ROYAL AVE",

"FORT ROYAL AV"

],

"to": "FORT ROYAL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ETIWAN AVE",

"ETIWAN AV"

],

"to": "ETIWAN AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PIEDMONT AVE",

"PIEDMONT AV"

],

"to": "PIEDMONT AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"THREE OAKS AV",

"THREE OAKS AVE"

],

"to": "THREE OAKS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"11TH AVE",

"11TH AV"

],

"to": "11TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HAGOOD AVE",

"HAGOOD AV"

],

"to": "HAGOOD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ERSKINE AV",

"ERSKINE AVE"

],

"to": "ERSKINE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FARMFIELD AVE",

"FARMFIELD AV"

],

"to": "FARMFIELD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CEDARHURST AVE",

"CEDARHURST AV"

],

"to": "CEDARHURST AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ASHLEY AVE",

"ASHLEY AV"

],

"to": "ASHLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"FRUITWOOD AVE",

"FRUITWOOD AV"

],

"to": "FRUITWOOD AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MACLAURA HALL AVE",

"MACLAURA HALL AV"

],

"to": "MACLAURA HALL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MARY ADER AVE",

"MARY ADER AV"

],

"to": "MARY ADER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"MEMMINGER AVE",

"MEMMINGER AV"

],

"to": "MEMMINGER AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"WESTMORELAND AVE",

"WESTMORELAND AV"

],

"to": "WESTMORELAND AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AVONDALE AVE",

"AVONDALE AV"

],

"to": "AVONDALE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"LAUREL AVE",

"LAUREL AV"

],

"to": "LAUREL AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AUDUBON AVE",

"AUDUBON AV"

],

"to": "AUDUBON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"DAVIDSON AV",

"DAVIDSON AVE"

],

"to": "DAVIDSON AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"RUTLEDGE AVE",

"RUTLEDGE AV"

],

"to": "RUTLEDGE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"TALL OAK AVE",

"TALL OAK AV"

],

"to": "TALL OAK AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"OTIS AVE",

"OTIS AV"

],

"to": "OTIS AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PLYMOUTH AVE",

"PLYMOUTH AV"

],

"to": "PLYMOUTH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BARRACUDA ST",

"BARACUDA ST"

],

"to": "BARRACUDA ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"PRESIDENT PL",

"PRESIDENTS PL"

],

"to": "PRESIDENT PL"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"10TH AVE",

"10TH AV"

],

"to": "10TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"HUNLEY AVE",

"HUNLEY AV"

],

"to": "HUNLEY AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"9TH AVE",

"9TH AV"

],

"to": "9TH AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"ACKERMANS CT",

"ACKERMAN CT"

],

"to": "ACKERMANS CT"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"BELGRADE AVE",

"BELGRADE AV"

],

"to": "BELGRADE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"CASTLE AV",

"CASTLE AVE"

],

"to": "CASTLE AVE"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"5TH AVE",

"5TH AV"

],

"to": "5TH AVE"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad2",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad2",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"MCLEOD RD",

"MCCLEOD RD"

],

"to": "MCCLEOD RD"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"KINLOCK CT",

"KINLOCH CT"

],

"to": "KINLOCK CT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column crossroad2",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "crossroad2",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"MIDDLE ATLANTIC WHF",

"MIDDLE ATLANTIC WHARF"

],

"to": "MIDDLE ATLANTIC WHF"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"COURT HOUSE SQ",

"COURT HOUSE SQUARE"

],

"to": "COURT HOUSE SQ"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"AQUADUCT ST",

"AQUADUCT"

],

"to": "AQUADUCT ST"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"E HAMPSTEAD SQUARE",

"N HAMPSTEAD SQ"

],

"to": "E HAMPSTEAD SQ"

},

{

"fromBlank": false,

"fromError": false,

"from": [

"SAINT DENIS DR",

"SAINT DENNIS DR"

],

"to": "SAINT DENNIS DR"

}

]

},

{

"op": "core/column-addition",

"description": "Create column category at index 10 based on column nature using expression grel:value",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "category",

"columnInsertIndex": 10,

"baseColumnName": "nature",

"expression": "grel:value",

"onError": "set-to-blank"

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('HANGUP'), value.replace(value, \"HANGUP\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('HANGUP'), value.replace(value, \"HANGUP\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ABDUCTION'), value.replace(value, \"ABDUCTION\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ABDUCTION'), value.replace(value, \"ABDUCTION\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ACCIDENT'), value.replace(value, \"ACCIDENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ACCIDENT'), value.replace(value, \"ACCIDENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ANIMAL'), value.replace(value, \"ANIMAL\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ANIMAL'), value.replace(value, \"ANIMAL\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('FIRE'), value.replace(value, \"FIRE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('FIRE'), value.replace(value, \"FIRE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ARSON'), value.replace(value, \"FIRE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ARSON'), value.replace(value, \"FIRE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BURGLARY'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BURGLARY'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ROBBERY'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ROBBERY'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LARCENY'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LARCENY'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ASSAULT'), value.replace(value, \"ASSAULT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ASSAULT'), value.replace(value, \"ASSAULT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DRUNK DRIVER'), value.replace(value, \"ACCIDENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DRUNK DRIVER'), value.replace(value, \"ACCIDENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"LARCENCY OCCURED EARLIER"

],

"to": "LARCENY OCCURED EARLIER"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"LARCENCY IN PROGRESS"

],

"to": "LARCENY IN PROGRESS"

}

]

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('HANGUP'), value.replace(value, \"HANGUP\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('HANGUP'), value.replace(value, \"HANGUP\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DEATH'), value.replace(value, \"DEATH\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DEATH'), value.replace(value, \"DEATH\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('RESCUE'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('RESCUE'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DOG'), value.replace(value, \"ANIMAL\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DOG'), value.replace(value, \"ANIMAL\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TRAFFIC'), value.replace(value, \"TRAFFIC\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TRAFFIC'), value.replace(value, \"TRAFFIC\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('HOMICIDE'), value.replace(value, \"DEATH\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('HOMICIDE'), value.replace(value, \"DEATH\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LARCENY'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LARCENY'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('OVERDOSE'), value.replace(value, \"NARCOTICS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('OVERDOSE'), value.replace(value, \"NARCOTICS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('SHOOTING'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('SHOOTING'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('STABBING'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('STABBING'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('GUN'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('GUN'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('VEHICLE'), value.replace(value, \"VEHICLE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('VEHICLE'), value.replace(value, \"VEHICLE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('VIOLENCE'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('VIOLENCE'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TRESPASSER'), value.replace(value, \"TRESPASS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TRESPASSER'), value.replace(value, \"TRESPASS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TOWING'), value.replace(value, \"TRAFFIC\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TOWING'), value.replace(value, \"TRAFFIC\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('SHOPLIFTER'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('SHOPLIFTER'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('VANDALISM'), value.replace(value, \"VANDALISM\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('VANDALISM'), value.replace(value, \"VANDALISM\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('GRAFFITI'), value.replace(value, \"VANDALISM\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('GRAFFITI'), value.replace(value, \"VANDALISM\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('SECURED'), value.replace(value, \"TRESPASS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('SECURED'), value.replace(value, \"TRESPASS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DRIVEWAY'), value.replace(value, \"TRAFFIC\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DRIVEWAY'), value.replace(value, \"TRAFFIC\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BICYCLE'), value.replace(value, \"ACCIDENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BICYCLE'), value.replace(value, \"ACCIDENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('CHILD'), value.replace(value, \"UNDERAGE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('CHILD'), value.replace(value, \"UNDERAGE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DELINQUENT'), value.replace(value, \"UNDERAGE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DELINQUENT'), value.replace(value, \"UNDERAGE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('WEAPON'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('WEAPON'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PEDESTRIAN'), value.replace(value, \"TRAFFIC\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PEDESTRIAN'), value.replace(value, \"TRAFFIC\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"DELIQUENT FINE AND BOOT REMOVE"

],

"to": "DELINQUENT FINE AND BOOT REMOVE"

}

]

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DELINQUENT'), value.replace(value, \"UNDERAGE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DELINQUENT'), value.replace(value, \"UNDERAGE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BOAT'), value.replace(value, \"ACCIDENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BOAT'), value.replace(value, \"ACCIDENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('CAR'), value.replace(value, \"ACCIDENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('CAR'), value.replace(value, \"ACCIDENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ESCORT'), value.replace(value, \"ESCORT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ESCORT'), value.replace(value, \"ESCORT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('STOLEN'), value.replace(value, \"ROBBERY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('STOLEN'), value.replace(value, \"ROBBERY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('INJURY'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('INJURY'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('WIRES'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('WIRES'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('HOSTAGE'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('HOSTAGE'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('CASUALTY'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('CASUALTY'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ASSIST'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ASSIST'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TRUANCY'), value.replace(value, \"UNDERAGE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TRUANCY'), value.replace(value, \"UNDERAGE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BREACH'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BREACH'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('FRAUD'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('FRAUD'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('HARRASSMENT'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('HARRASSMENT'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('STALKING'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('STALKING'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DISTRESS'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DISTRESS'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ALARM'), value.replace(value, \"TRESSPASS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ALARM'), value.replace(value, \"TRESSPASS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('CASULTY'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('CASULTY'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DROWNING'), value.replace(value, \"DEATH\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DROWNING'), value.replace(value, \"DEATH\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('EXPLOSION'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('EXPLOSION'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('THREAT'), value.replace(value, \"THREAT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('THREAT'), value.replace(value, \"THREAT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LOUD MUSIC'), value.replace(value, \"NUISANCE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LOUD MUSIC'), value.replace(value, \"NUISANCE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('INJURED'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('INJURED'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PERSON STOP'), value.replace(value, \"SUSPECT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PERSON STOP'), value.replace(value, \"SUSPECT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('SUSP PERSON'), value.replace(value, \"SUSPECT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('SUSP PERSON'), value.replace(value, \"SUSPECT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ASSAULT'), value.replace(value, \"VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ASSAULT'), value.replace(value, \"VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BOLO'), value.replace(value, \"SUSPECT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BOLO'), value.replace(value, \"SUSPECT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DISPUTE VERBAL'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DISPUTE VERBAL'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('EVENT'), value.replace(value, \"EVENT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('EVENT'), value.replace(value, \"EVENT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ROAD RAGE'), value.replace(value, \"TRAFFIC\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ROAD RAGE'), value.replace(value, \"TRAFFIC\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LOITERING'), value.replace(value, \"VAGRANT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LOITERING'), value.replace(value, \"VAGRANT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PANHANDL'), value.replace(value, \"VAGRANT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PANHANDL'), value.replace(value, \"VAGRANT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LOCKOUT'), value.replace(value, \"RESCUE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LOCKOUT'), value.replace(value, \"RESCUE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PROPERTY'), value.replace(value, \"PROPERTY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PROPERTY'), value.replace(value, \"PROPERTY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ITEM'), value.replace(value, \"PROPERTY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ITEM'), value.replace(value, \"PROPERTY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PEDDLER'), value.replace(value, \"VAGRANT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PEDDLER'), value.replace(value, \"VAGRANT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('DROW'), value.replace(value, \"DEATH\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('DROW'), value.replace(value, \"DEATH\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('EXPLOSIVE'), value.replace(value, \"THREAT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('EXPLOSIVE'), value.replace(value, \"THREAT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TRESSPASS'), value.replace(value, \"TRESPASS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TRESSPASS'), value.replace(value, \"TRESPASS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ABDUCTION'), value.replace(value, \"MISSING PERSON\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ABDUCTION'), value.replace(value, \"MISSING PERSON\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PERSON MISSING'), value.replace(value, \"MISSING PERSON\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PERSON MISSING'), value.replace(value, \"MISSING PERSON\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('BRIDGE'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('BRIDGE'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('CRIME SCENE'), value.replace(value, \"OFFICER\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('CRIME SCENE'), value.replace(value, \"OFFICER\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('OFFICER'), value.replace(value, \"OFFICER\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('OFFICER'), value.replace(value, \"OFFICER\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('MESSAGE'), value.replace(value, \"NOTIFICATION\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('MESSAGE'), value.replace(value, \"NOTIFICATION\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('NOTIFICATION'), value.replace(value, \"NOTIFICATION\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('NOTIFICATION'), value.replace(value, \"NOTIFICATION\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('FLOODING'), value.replace(value, \"NATURE\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('FLOODING'), value.replace(value, \"NATURE\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TEST CALL'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TEST CALL'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('FOOT'), value.replace(value, \"SUSPECT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('FOOT'), value.replace(value, \"SUSPECT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TELEPHONE'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TELEPHONE'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('NUISANCE'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('NUISANCE'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PICKETERS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PICKETERS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('VEHICLE'), value.replace(value, \"PROPERTY\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('VEHICLE'), value.replace(value, \"PROPERTY\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('ESCORT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('ESCORT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PRISONER'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PRISONER'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('NOTIFICATION'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('NOTIFICATION'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('MENTAL SUBJECT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('MENTAL SUBJECT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PUBLIC WORKS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PUBLIC WORKS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('LOST TOU'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('LOST TOU'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('EVENT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('EVENT'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('TEMPORARY DETENTION'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('TEMPORARY DETENTION'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('VAGRANT'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('VAGRANT'), value.replace(value, \"NON VIOLENT CRIME\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PICKUP DEBRIS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PICKUP DEBRIS'), value.replace(value, \"MISCELLANEOUS\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('PACKAGE'), value.replace(value, \"THREAT\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('PACKAGE'), value.replace(value, \"THREAT\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(value.contains('FOUND'), value.replace(value, \"MISSING PERSON\"), value)",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "grel:if(value.contains('FOUND'), value.replace(value, \"MISSING PERSON\"), value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/column-addition",

"description": "Create column daysSinceCall at index 3 based on column calltime using expression grel:diff(now(), cells['calltime'].value, 'days')",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "daysSinceCall",

"columnInsertIndex": 3,

"baseColumnName": "calltime",

"expression": "grel:diff(now(), cells['calltime'].value, 'days')",

"onError": "set-to-blank"

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"VANDALISM"

],

"to": "PROPERTY"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"TRESPASS"

],

"to": "PROPERTY"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"KEEP A CHECK"

],

"to": "OFFICER INITIATED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"WALK AND TALK"

],

"to": "OFFICER INITIATED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"OFFICER"

],

"to": "OFFICER REQUIRED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"NATURE"

],

"to": "MISCELLANEOUS"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"FUGITIVE"

],

"to": "MISCELLANEOUS"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"ROBBERY"

],

"to": "NON VIOLENT CRIME"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"NARCOTICS"

],

"to": "NON VIOLENT CRIME"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"HANGUP"

],

"to": "MISCELLANEOUS"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"TRAFFIC"

],

"to": "TRAFFIC/ACCIDENT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"ACCIDENT"

],

"to": "TRAFFIC/ACCIDENT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"VIOLENT CRIME"

],

"to": "VIOLENT CRIME/THREAT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"THREAT"

],

"to": "VIOLENT CRIME/THREAT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"SUSPECT"

],

"to": "SUSPECT/MISSING PERSON"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"MISSING PERSON"

],

"to": "SUSPECT/MISSING PERSON"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"OFFICER INITIATED"

],

"to": "OFFICER INITIATED/REQUIRED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"OFFICER REQUIRED"

],

"to": "OFFICER INITIATED/REQUIRED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"RESCUE"

],

"to": "OFFICER INITIATED/REQUIRED"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"DEATH"

],

"to": "VIOLENT CRIME"

}

]

},

{

"op": "core/text-transform",

"description": "Text transform on cells in column category using expression grel:if(cells['nature'].value.contains('ARSON'), \"VIOLENT CRIME\", value)",

"engineConfig": {

"mode": "row-based",

"facets": [

{

"omitError": false,

"expression": "value",

"selectBlank": false,

"selection": [

{

"v": {

"v": "FIRE",

"l": "FIRE"

}

}

],

"selectError": false,

"invert": false,

"name": "category",

"omitBlank": false,

"type": "list",

"columnName": "category"

}

]

},

"columnName": "category",

"expression": "grel:if(cells['nature'].value.contains('ARSON'), \"VIOLENT CRIME\", value)",

"onError": "keep-original",

"repeat": false,

"repeatCount": 10

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"VIOLENT CRIME"

],

"to": "VIOLENT CRIME/THREAT"

}

]

},

{

"op": "core/mass-edit",

"description": "Mass edit cells in column category",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"columnName": "category",

"expression": "value",

"edits": [

{

"fromBlank": false,

"fromError": false,

"from": [

"UNDERAGE"

],

"to": "UNDERAGE COMMITED/VICTIM"

}

]

},

{

"op": "core/column-addition",

"description": "Create column numberOfCallsOnWeekdays at index 6 based on column Weekday using expression grel:facetCount(cells[\"Weekday\"].value, \"value\", \"Weekday\")",

"engineConfig": {

"mode": "row-based",

"facets": []

},

"newColumnName": "numberOfCallsOnWeekdays",

"columnInsertIndex": 6,

"baseColumnName": "Weekday",

"expression": "grel:facetCount(cells[\"Weekday\"].value, \"value\", \"Weekday\")",

"onError": "set-to-blank"

}

]