

Data Cleaning Recipe

```
[
  {
    "op": "core/column-removal",
    "columnName": "quality_dimension",
    "description": "Remove column quality_dimension"
  },
  {
    "op": "core/row-reorder",
    "mode": "row-based",
    "sorting": {
      "criteria": [
        {
          "valueType": "string",
          "column": "id",
          "blankPosition": 2,
          "errorPosition": 1,
          "reverse": false,
          "caseSensitive": false
        }
      ]
    },
    "description": "Reorder rows"
  },
  {
    "op": "core/blank-down",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "id",
    "description": "Blank down cells in column id"
  },
  {
    "op": "core/row-removal",
    "engineConfig": {
      "facets": [
        {
          "type": "list",
          "name": "id",
          "expression": "isBlank(value)",
          "columnName": "id",
          "invert": false,
          "omitBlank": false,
          "omitError": false,
          "selection": [
            {
              "v": {
                "v": true,
                "l": "true"
              }
            }
          ],
          "selectBlank": false,
          "selectError": false
        }
      ],
      "mode": "row-based"
    }
  }
]
```

```

    },
    "description": "Remove rows"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "release_date",
    "expression": "value.toDate()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column release_date using
expression value.toDate()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "release_date",
    "expression": "grel:value.toString(\"dd-MM-yyyy\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column release_date using
expression grel:value.toString(\"dd-MM-yyyy\")"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "value.toTitlecase()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression value.toTitlecase()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "album",
    "expression": "value.toTitlecase()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression value.toTitlecase()"
  },
  {

```

```

      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "artists",
      "expression": "value.toTitlecase()",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column artists using
expression value.toTitlecase()"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "album",
      "expression": "grel:value.replace('_', '')",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column album using
expression grel:value.replace('_', '')"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "album",
      "expression": "grel:value.replace(';', '')",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column album using
expression grel:value.replace(';', '')"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "album",
      "expression": "grel:value.replace('...', '')",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column album using
expression grel:value.replace('...', '')"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"

```

```

    },
    "columnName": "album",
    "expression": "grel:value.replace('...', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression grel:value.replace('...', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "album",
    "expression": "grel:value.replace(\"'\", '\"\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression grel:value.replace(\"'\", '\"\")"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "album",
    "expression": "grel:value.replace('\\\"', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression grel:value.replace('\\\"', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [
        {
          "type": "list",
          "name": "album",
          "expression": "value",
          "columnName": "album",
          "invert": false,
          "omitBlank": false,
          "omitError": false,
          "selection": [
            {
              "v": {
                "v": "(not Adam)",
                "l": "(not Adam)"
              }
            }
          ],
          "selectBlank": false,
          "selectError": false
        }
      ]
    }
  }
}

```

```

    ],
    "mode": "row-based"
  },
  "columnName": "album",
  "expression": "grel:value.replace('(not Adam)','Not Adam')",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column album using
expression grel:value.replace('(not Adam)','Not Adam') "
},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [
      {
        "type": "list",
        "name": "album",
        "expression": "value",
        "columnName": "album",
        "invert": false,
        "omitBlank": false,
        "omitError": false,
        "selection": [
          {
            "v": {
              "v": "##retribution",
              "l": "##retribution"
            }
          }
        ],
        "selectBlank": false,
        "selectError": false
      }
    ],
    "mode": "row-based"
  },
  "columnName": "album",
  "expression": "grel:value.replace('##','')",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column album using
expression grel:value.replace('##','') "
},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [
      {
        "type": "list",
        "name": "album",
        "expression": "value",
        "columnName": "album",
        "invert": false,
        "omitBlank": false,
        "omitError": false,
        "selection": [
          {
            "v": {

```

```

        "v": "##retributionv",
        "l": "##retributionv"
    }
}
],
"selectBlank": false,
"selectError": false
}
],
"mode": "row-based"
},
"columnName": "album",
"expression": "grel:value.replace('##retributionv','Retribution')",
"onError": "keep-original",
"repeat": false,
"repeatCount": 10,
"description": "Text transform on cells in column album using
expression grel:value.replace('##retributionv','Retribution')"
},
{
    "op": "core/row-removal",
    "engineConfig": {
        "facets": [
            {
                "type": "list",
                "name": "album",
                "expression": "value",
                "columnName": "album",
                "invert": false,
                "omitBlank": false,
                "omitError": false,
                "selection": [
                    {
                        "v": {
                            "v": "#hollywood",
                            "l": "#hollywood"
                        }
                    }
                ],
                "selectBlank": false,
                "selectError": false
            }
        ],
        "mode": "row-based"
    },
    "description": "Remove rows"
},
{
    "op": "core/mass-edit",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "album",
    "expression": "value",
    "edits": [
        {
            "from": [
                "4v"
            ],

```

```

        "fromBlank": false,
        "fromError": false,
        "to": "4"
    }
],
"description": "Mass edit cells in column album"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "album",
    "expression": "value.toTitlecase()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression value.toTitlecase()"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "album_id",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album_id using
expression value.trim()"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "album",
    "expression": "grel:if(value[-1]==\"v\",value[0,-1],value)",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression grel:if(value[-1]==\"v\",value[0,-1],value)"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('-', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,

```

```

    "description": "Text transform on cells in column name using
expression grel:value.replace('-', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace(';', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace(';', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('?', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('?', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('... ', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('... ', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('... ', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('... ', '')"
  },
  {

```



```

    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('...', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('...', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace(\"'\", '\"\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace(\"'\", '\"\")"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('\\\"', '')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('\\\"', '')"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace(\".\", '\"\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace(\".\", '\"\")"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"

```

```

    },
    "columnName": "name",
    "expression": "grel:value.replace('(', '' )",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('(', '' )"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('[', '' )",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('[', '' )"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace(')', '' )",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace(')', '' )"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace(']', '' )",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace(']', '' )"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [
        {
          "type": "list",
          "name": "explicit",
          "expression": "value",
          "columnName": "explicit",

```

```

        "invert": false,
        "omitBlank": false,
        "omitError": false,
        "selection": [
            {
                "v": {
                    "v": "0",
                    "l": "0"
                }
            }
        ],
        "selectBlank": false,
        "selectError": false
    }
],
    "mode": "row-based"
},
    "columnName": "explicit",
    "expression": "grel:value.replace('0','FALSE')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column explicit using
expression grel:value.replace('0','FALSE')"
},
{
    "op": "core/column-addition",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "baseColumnName": "release_date",
    "expression": "grel:value",
    "onError": "set-to-blank",
    "newColumnName": "year_new",
    "columnInsertIndex": 16,
    "description": "Create column year_new at index 16 based on column
release_date using expression grel:value"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "year_new",
    "expression": "value.toDate()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column year_new using
expression value.toDate()"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "year_new",

```

```

    "expression": "grel:value.toString(\"yyyy\")",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column year_new using
expression grel:value.toString(\"yyyy\")"
  },
  {
    "op": "core/column-removal",
    "columnName": "year",
    "description": "Remove column year"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [
        {
          "type": "list",
          "name": "last_updated",
          "expression": "value",
          "columnName": "last_updated",
          "invert": false,
          "omitBlank": false,
          "omitError": false,
          "selection": [
            {
              "v": {
                "v": "14-03-2021",
                "l": "14-03-2021"
              }
            }
          ],
          "selectBlank": false,
          "selectError": false
        }
      ],
      "mode": "row-based"
    },
    "columnName": "last_updated",
    "expression": "jython:return '16-03-2021'",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column last_updated using
expression jython:return '16-03-2021'"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [
        {
          "type": "list",
          "name": "artists",
          "expression": "value",
          "columnName": "artists",
          "invert": false,
          "omitBlank": false,
          "omitError": false,
          "selection": [],
          "selectBlank": true,

```

```

        "selectError": false
    }
],
    "mode": "row-based"
},
    "columnName": "artists",
    "expression": "jython:return 'N/A'",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression jython:return 'N/A'"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "artists",
    "expression":
"grel:value.replace('(','')\nvalue.replace(',')'\nvalue.replace('\\\"\\\",\\
\\\"\\)\nvalue.replace('[','')\nvalue.replace('@','')\nvalue.replace('*',')\n
\nvalue.replace('$','')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression
grel:value.replace('(','')\nvalue.replace(',')'\nvalue.replace('\\\"\\\",\\
\\\"\\)\nvalue.replace('[','')\nvalue.replace('@','')\nvalue.replace('*',')\n
\nvalue.replace('$','')"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "artists",
    "expression": "grel:value.replace('[','')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression grel:value.replace('[','')"
},
{
    "op": "core/text-transform",
    "engineConfig": {
        "facets": [],
        "mode": "row-based"
    },
    "columnName": "artists",
    "expression": "grel:value.replace(']','')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression grel:value.replace(']','')"
}

```

```

},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [],
    "mode": "row-based"
  },
  "columnName": "artists",
  "expression": "grel:value.replace('@','')",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column artists using
expression grel:value.replace('@','')",
},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [],
    "mode": "row-based"
  },
  "columnName": "artists",
  "expression": "grel:value.replace('*', '')",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column artists using
expression grel:value.replace('*', '')",
},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [],
    "mode": "row-based"
  },
  "columnName": "artists",
  "expression": "grel:value.replace('$','')",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column artists using
expression grel:value.replace('$','')",
},
{
  "op": "core/text-transform",
  "engineConfig": {
    "facets": [],
    "mode": "row-based"
  },
  "columnName": "artists",
  "expression": "grel:value.replace(\"'\", '\"\")",
  "onError": "keep-original",
  "repeat": false,
  "repeatCount": 10,
  "description": "Text transform on cells in column artists using
expression grel:value.replace(\"'\", '\"\")",
},
{
  "op": "core/column-move",
  "columnName": "year_new",

```

```

    "index": 14,
    "description": "Move column year_new to position 14"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('#','')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('#','') "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "grel:value.replace('$','')",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression grel:value.replace('$','') "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression value.trim() "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "album_id",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album_id using
expression value.trim() "
  },
  {

```

```

    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "artists",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression value.trim() "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "artist_ids",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artist_ids using
expression value.trim() "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "track_number",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column track_number using
expression value.trim() "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "explicit",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column explicit using
expression value.trim() "
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },

```



```

    },
    "columnName": "year_new",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column year_new using
expression value.trim()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "release_date",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column release_date using
expression value.trim()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "last_updated",
    "expression": "value.trim()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column last_updated using
expression value.trim()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "track_number",
    "expression": "value.toNumber()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column track_number using
expression value.toNumber()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "Index",
    "expression": "value.toNumber()",
    "onError": "keep-original",

```

```

    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column Index using
expression value.toNumber()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "id",
    "expression": "value.toString()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column id using expression
value.toString()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "name",
    "expression": "value.toString()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column name using
expression value.toString()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "album",
    "expression": "value.toString()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column album using
expression value.toString()"
  },
  {
    "op": "core/text-transform",
    "engineConfig": {
      "facets": [],
      "mode": "row-based"
    },
    "columnName": "artists",
    "expression": "value.toString()",
    "onError": "keep-original",
    "repeat": false,
    "repeatCount": 10,
    "description": "Text transform on cells in column artists using
expression value.toString()"
  }

```

```

    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "explicit",
      "expression": "value.toString()",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column explicit using
expression value.toString()"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "duration_ms",
      "expression": "value.toNumber()",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column duration_ms using
expression value.toNumber()"
    },
    {
      "op": "core/text-transform",
      "engineConfig": {
        "facets": [],
        "mode": "row-based"
      },
      "columnName": "time_signature",
      "expression": "value.toNumber()",
      "onError": "keep-original",
      "repeat": false,
      "repeatCount": 10,
      "description": "Text transform on cells in column time_signature
using expression value.toNumber()"
    }
  ]

```