Example A: SDTH with SPARQL Queries and Output

March 11, 2025

Contents

March 11, 2025 1

1. Example Python Script (Example_A_Python_code.py) 2

2. Example SPARQL Queries and Output 3

2.1 What variables affected the values of variable HHcateg? 3

2.2 What variables were affected by variable PPHHSIZE? 4

2.3 What commands affected the values of variable HHcateg? 5

2.4 What commands were affected by variable PPHHSIZE? 7

3. SDTL version of the Python Script (Example_A_SDTL.json) 9

4: SDTH version of Example in JSON-LD RDF (SDTH_RDF_Example_A.json) 14

5: SDTH of Example in Turtle RDF (SDTH_RDF_Example_A_turtle.ttl) 20

1. Example Python Script (Example_A_Python_code.py)

```
import pandas as pd
PoliticalData = pd.read_csv("SmallTestPolitical.csv")
PersonalData = pd.read_csv("SmallTestPersonal.csv")
PersonalData   = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'] )
PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999], include_lowest=True,
        right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'] )
MergedData = PersonalData.merge(PoliticalData, on="ID", how="inner")
MergedData.to_csv("SmallTestMerged.csv")
```

This Python code does the following:

1. The import command loads the pandas library of commands and functions
2. The file "SmallTestPolitical.csv" is loaded into the ‘PoliticalData’ dataframe
3. The file " SmallTestPersonal.csv" is loaded into the ‘PersonalData’ dataframe
4. Variable ‘HHsize’ is set equal to variable 'PPHHSIZE' and added to the ‘PersonalData dataframe
5. Variable 'HHcateg' is created by classifying ‘HHsize’ into categories and adding it to the ‘PersonalData dataframe. 'HHcateg' is a factor variable, and value labels are assigned.
6. Rows from dataframes ‘PoliticalData’ and ‘PersonalData’ are merged using variable ‘ID’ as a key. The resulting dataframe is named ‘MergedData’
7. The ‘MergedData’ dataframe is saved to a CSV file named "SmallTestMerged.csv"

2. Example SPARQL Queries and Output

2.1 What variables affected the values of variable HHcateg?

SPARQL:

```
PREFIX sdth: <http://DDI/SDTH/>
PREFIX sdtest: <http://test/#>

SELECT distinct  ?sname ?oname
WHERE {
      ?s  sdth:wasDerivedFrom+ ?o .
                  ?s  sdth:hasName ?sname .
                  ?o  sdth:hasName ?oname .
   FILTER (?sname = "HHcateg")
   }
```

Output:

Subject variable name (?sname)	**Object variable name(?oname)**
HHcateg	HHsize
HHcateg	PPHHSIZE

2.2 What variables were affected by variable PPHHSIZE?

SPARQL:

```
PREFIX sdth: <http://DDI/SDTH/>
PREFIX sdtest: <http://test/#>

SELECT distinct  ?sname ?oname
WHERE {
   ?s  sdth:wasDerivedFrom+ ?o .
                ?s  sdth:hasName ?sname .
                ?o  sdth:hasName ?oname .
   FILTER (?oname = "PPHHSIZE")
   }
```

Output:

Subject variable name (?sname)	**Object variable name(?oname)**
HHsize	PPHHSIZE
HHcateg	PPHHSIZE

2.3 What commands affected the values of variable HHcateg?

This query is a union of 3 subqueries

- The 1st subquery finds program steps where the target variable (HHcateg) wasDerivedFrom or elaborationOf.
- The 2nd subquery finds program steps that changed variables that the target variable (HHcateg) wasDerivedFrom or elaborationOf.
- The 3rd subquery finds program steps with subprograms that changed the target variable (HHcateg)

SPARQL:

```
PREFIX sdth: <http://DDI/SDTH/>
PREFIX sdtest: <http://test/#>
   SELECT DISTINCT ?sname ?oname  ?pscode
   WHERE {
      {?x  ( sdth:wasDerivedFrom |sdth:elaborationOf) ?y .
            ?x  sdth:hasName ?xname .
            ?y  sdth:hasName ?yname .
      ?pstep sdth:assignsVariable ?x .
      ?pstep sdth:hasSourceCode  ?pscode.}
      UNION
       {?x  ( sdth:wasDerivedFrom+ |sdth:elaborationOf+) ?y .
            ?x  sdth:hasName ?xname .
            ?y  sdth:hasName ?yname .
      ?pstep sdth:assignsVariable ?y .
      ?pstep sdth:hasSourceCode  ?pscode.
         MINUS { ?pstep sdth:usesVariable ?y . }
          }
      UNION
      {?pstep sdth:hasProgramStep ?psub.
      ?psub sdth:assignsVariable ?x .
      ?pstep sdth:hasSourceCode ?pscode.
      ?x  sdth:hasName ?xname .
      ?x  ( sdth:wasDerivedFrom+ |sdth:elaborationOf+) ?y .
      ?y  sdth:hasName ?yname . }
   FILTER (?xname = "HHcateg")
   }
   ORDER BY ?pstep
```

Output:

Target variable (?xname)	**Other variable (?yname)**	**Command (?pscode)**
HHcateg	PPHHSIZE	PersonalData = pd.read_csv("SmallTestPersonal.csv")
HHcateg	HHsize	[PersonalData = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'])]
HHcateg	HHsize	[PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'])]
HHcateg	PPHHSIZE	[PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'])]
HHcateg	HHcateg	[PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'])]
HHcateg	HHcateg	[MergedData = PersonalData.merge(PoliticalData, on="ID", how="inner")]

2.4 What commands were affected by variable PPHHSIZE?

This query is a union of 3 subqueries:

- The first query finds program steps that use the target variable.
- The second query finds program steps that use a variable derived from the target variable.
- The third query finds program steps with source code that have subprograms that use a variable derived from the target variable.

SPARQL:

```
PREFIX sdth: <http://DDI/SDTH/>
PREFIX sdtest: <http://test/#>
SELECT distinct ?pstep  ?psource ?oname ?tname
                WHERE {
                {
                        ?pstep  sdth:usesVariable+ ?o .
                        ?pstep  sdth:hasSourceCode ?psource .
                        ?o  sdth:hasName ?tname .
                        ?o  sdth:hasName ?oname .
                        FILTER (?tname = "PPHHSIZE")
                        }
                UNION
                {
                        ?pstep  sdth:usesVariable+ ?s2 .
                        ?pstep  sdth:hasSourceCode ?psource .
                        {SELECT distinct  ?s2   ?oname   ?tname
                        WHERE {
                                ?s2  (sdth:wasDerivedFrom+  | sdth:elaborationOf+ )+ ?o2 .
                                ?s2  sdth:hasName ?oname .
                                ?o2  sdth:hasName ?tname .
                                FILTER (?tname = "PPHHSIZE" )
                                }
                                }
                        }
        UNION
                {     ?pstep sdth:hasProgramStep ?psub.
            ?pstep sdth:hasSourceCode ?psource.
            {  SELECT DISTINCT    ?s   ?tname  ?t  ?oname ?psub
                        WHERE {
                            ?s   (sdth:wasDerivedFrom+ | sdth:elaborationOf+ )+ ?t .
                            ?s  sdth:hasName ?oname .
                            ?t  sdth:hasName ?tname .
                            ?psub (sdth:usesVariable+ | sdth:assignsVariable+) ?s .
                            FILTER (?tname = "PPHHSIZE" )
                        }
                  }
            }
                }
ORDER BY ?pstep
```

Output:

Target variable ?oname	Variable derived from target variable ?tname	Command in source code (?psource)
PPHHSIZE	PPHHSIZE	PersonalData = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'])
PPHHSIZE	HHcateg	PersonalData = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'])
PPHHSIZE	HHsize	PersonalData = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'])
PPHHSIZE	PPHHSIZE	MergedData = PersonalData.merge(PoliticalData, on="ID", how="inner")
PPHHSIZE	HHcateg	MergedData = PersonalData.merge(PoliticalData, on="ID", how="inner")
PPHHSIZE	HHsize	MergedData = PersonalData.merge(PoliticalData, on="ID", how="inner")
PPHHSIZE	PPHHSIZE	MergedData.to_csv("SmallTestMerged.csv")
PPHHSIZE	HHcateg	MergedData.to_csv("SmallTestMerged.csv")
PPHHSIZE	HHsize	MergedData.to_csv("SmallTestMerged.csv")

3. SDTL version of the Python Script (Example_A_SDTL.json)


```
{"commands": [
{"$type": "NoTransformOp", "command": "NoTransformOp",
        "sourceInformation": [{"$type": "SourceInformation", "lineNumberStart": 1, "lineNumberEnd": 1,
                "sourceStartIndex": 1, "sourceStopIndex": 19,
                "originalSourceText": "import pandas as pd"}]
        },
{"$type": "Load", "command": "Load",
                "fileName": "SmallTestPolitical.csv",
                "fileFormat": "csv",
                "producesDataframe": [{"dataframeName": "PoliticalData", "variableInventory": [ "Q3",
"Q244_NEW", "Q356", "Q330A", "Q330C", "Q27", "Q1010", "ID"]}],
        "sourceInformation": [{"$type": "SourceInformation", "lineNumberStart": 3, "lineNumberEnd": 3,
                "sourceStartIndex": 22, "sourceStopIndex": 74,
                "originalSourceText": "PoliticalData = pd.read_csv(\"SmallTestPolitical.csv\")"}]
        },

{"$type": "Load", "command": "Load",
                "fileName": "SmallTestPersonal.csv",
                "fileFormat": "csv",
                "producesDataframe": [{"dataframeName": "PersonalData", "variableInventory": [ "PPEDUCAT",
"PPHHSIZE", "PPRENT", "ID"]}],
        "sourceInformation": [{"$type": "SourceInformation", "lineNumberStart": 5, "lineNumberEnd": 5,
                "sourceStartIndex": 77, "sourceStopIndex": 127,
                "originalSourceText": "PersonalData = pd.read_csv(\"SmallTestPersonal.csv\")"}]
        },

{"$type": "Compute", "command": "Compute",
    "variable": {
      "$type": "VariableSymbolExpression",
      "variableName": "HHsize" },
    "expression": {
      "$type": "VariableSymbolExpression",
      "variableName": "PPHHSIZE" },
    "consumesDataframe": [ {
                        "dataframeName": "PersonalData",
      "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID"]
      } ],
    "producesDataframe": [ {
      "dataframeName": "PersonalData",
      "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize" ]
      } ],
    "sourceInformation": [{
                        "$type": "SourceInformation", "lineNumberStart": 7, "lineNumberEnd": 7,
```

```
                    "sourceStartIndex": 130, "sourceStopIndex": 199,
    "originalSourceText": "PersonalData   = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'] )"
                    }]
},
     {"$type": "Compute",
            "command": "Compute",
            "consumesDataframe": [ {
    "dataframeName": "PersonalData",
    "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize" ]
    } ],
            "variable": {
                    "$type": "VariableSymbolExpression",
                    "variableName": "HHcateg" },
            "expression": {
                    "$type": "FunctionCallExpression",
                    "function": "cut_list",
                    "isSdtlName": true,
                    "arguments":[
                            {       "$type": "FunctionArgument",
                                    "argumentName": "EXP1",
                                    "argumentValue":
                                    {       "$type": "VariableSymbolExpression",
                                            "variableName": "HHsize"        }
                            },
                            {       "$type": "FunctionArgument",
                                    "argumentName": "EXP2",
                                    "argumentValue":
                                    {       "$type": "ValueListExpression",
                                            "values": [
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "1" },
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "2" },
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "3" },
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "5" },
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "7" },
                                                    {       "$type": "NumericConstantExpression",
                                                            "numericType": "Integer",
                                                            "value": "10" },
                                                    {       "$type": "NumericConstantExpression",
```

```
                                                    "numericType": "Integer",
                                                    "value": "999" }
                                            ]
                                    }
                            },
                            {       "$type": "FunctionArgument",
                                    "argumentName": "EXP3",
                                    "argumentValue": {
                                            "$type": "StringConstantExpression",
                                            "value": "Left" }
                                    },
                            {       "$type": "FunctionArgument",
                                    "argumentName": "EXP4",
                                    "argumentValue": {
                                            "$type": "BooleanConstantExpression",
                                            "booleanValue": false }
                                    },
                            {       "$type": "FunctionArgument",
                                    "argumentName": "EXP5",
                                    "argumentValue": {
                                            "$type": "StringConstantExpression",
                                            "value": "Int_code" }
                                    }
                            ]
                    },
                    "producesDataframe": [ { "dataframeName": "PersonalData",
                            "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]}
                            ],
                    "sourceInformation": [{
                            "$type": "SourceInformation", "lineNumberStart": 9, "lineNumberEnd": 9,
                            "sourceStartIndex": 202, "sourceStopIndex": 365,
                            "originalSourceText": "PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3,
                            5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+']
                            )"}]
                            },
{"$type": "SetDataType",
                    "command": "SetDataType",
                    "dataType": "Factor",
                    "consumesDataframe": [  {
        "dataframeName": "PersonalData",
        "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]
        } ],
                    "producesDataframe": [ { "dataframeName": "PersonalData",
                    "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]
        } ],
                    "subType": "ordered",
                    "subTypeSchema": "https://pandas.pydata.org/pandas-
docs/stable/reference/api/pandas.Categorical.html",
```

```
                "variables": [
                        {       "$type": "VariableSymbolExpression",
                                "variableName": "HHcateg"}
                        ],
                "sourceInformation": [{
                        "$type": "SourceInformation", "lineNumberStart": 9, "lineNumberEnd": 9,
                        "sourceStartIndex": 202, "sourceStopIndex": 365,
                        "originalSourceText": "PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3,
                        5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+']
                        )"}]
          },
{"$type": "SetValueLabels",
                "command": "SetValueLabels",
                "consumesDataframe": [ {
        "dataframeName": "PersonalData",
        "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]
        } ],
                "labels": [
                        { "$type": "ValueLabel", "value": "0", "label": "1" },
                        { "$type": "ValueLabel", "value": "1", "label": "2" },
                        { "$type": "ValueLabel", "value": "2", "label": "3-7" },
                        { "$type": "ValueLabel", "value": "2", "label": "5-6" },
                        { "$type": "ValueLabel", "value": "2", "label": "7-9" },
                        { "$type": "ValueLabel", "value": "2", "label": "10+" }
                ],
                "producesDataframe": [ { "dataframeName": "PersonalData",
                "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]
                } ],
                "variables": [
                        {       "$type": "VariableSymbolExpression",
                                "variableName": "HHcateg"       }
                ],
                "sourceInformation": [{
                        "$type": "SourceInformation", "lineNumberStart": 9, "lineNumberEnd": 9,
                        "sourceStartIndex": 202, "sourceStopIndex": 365,
                        "originalSourceText": "PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3,
5, 7, 10, 999], include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'] )"}]
          },
{"$type": "MergeDatasets",
                "command": "MergeDatasets",
                "consumesDataframe": [
                        { "dataframeName": "PersonalData",
                        "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg" ]
                                } ,
                        { "dataframeName": "PoliticalData",
                        "variableInventory": [ "Q3", "Q244_NEW", "Q356", "Q330A", "Q330C", "Q27", "Q1010",
                        "ID"]
                                }
```

```
                ],
            "mergeByVariables": {
                "$type": "VariableSymbolExpression",
                "variableName": "ID"},
            "mergeFiles":    [
                {       "$type": "MergeFileDescription",
                        "fileName": "PersonalData",
                        "mergeType": "Cartesian",
                        "update": "Master",
                        "newRow": true
                        },
                {       "$type": "MergeFileDescription",
                        "fileName": "PoliticalData",
                        "mergeType": "Cartesian",
                        "update": "FillNew",
                        "newRow": false
                        }
                ],
            "producesDataframe": [
                {       "dataframeName": "MergedData ",
                        "variableInventory":
                        ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg", "Q3",
                        "Q244_NEW", "Q356", "Q330A", "Q330C", "Q27", "Q1010" ]
                        }
                ],
            "sourceInformation": [{
                "$type": "SourceInformation", "lineNumberStart": 11, "lineNumberEnd": 11,
                "sourceStartIndex": 368, "sourceStopIndex": 435,
                "originalSourceText": "MergedData = PersonalData.merge(PoliticalData, on=\"ID\",
                how=\"inner\")"}]
        },
{"$type": "Save",
            "command": "Save",
            "fileName": "SmallTestMerged.csv",
            "fileFormat": "CSV",
            "consumesDataframe": [
                {       "dataframeName": "MergedData ",
                "variableInventory": ["PPEDUCAT", "PPHHSIZE", "PPRENT", "ID", "HHsize", "HHcateg",
                "Q3", "Q244_NEW", "Q356", "Q330A", "Q330C", "Q27", "Q1010" ]
                        }
                ],
        "sourceInformation": [{
            "$type": "SourceInformation", "lineNumberStart": 13, "lineNumberEnd": 13,
            "sourceStartIndex": 438, "sourceStopIndex": 477,
            "originalSourceText": "MergedData.to_csv(\"SmallTestMerged.csv\")"
            }]
        }
        ]}
```

4: SDTH version of Example in JSON-LD RDF (SDTH_RDF_Example_A.json)


```
{       "@context": {
                "sdth": "http://DDI/SDTH/",
                "sdtest": "http://test/#",
                "ddi-l": "http://ddi-l/",
                "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                },

"@id" : "sdtest:ProgramEx5",
        "rdf:type" : {"@id": "sdth:Program"},
        "sdth:hasProgramStep": [ {"@id": "sdtest:ProgStep001"},
         {"@id": "sdtest:ProgStep002"},
         {"@id": "sdtest:ProgStep003"},
         {"@id": "sdtest:ProgStep004"},
         {"@id": "sdtest:ProgStep005"},
         {"@id": "sdtest:ProgStep006"},
         {"@id": "sdtest:ProgStep007" } ],

"sdth:ProgramStep": [

        {"@id" : "sdtest:ProgStep001",
                "rdf:type" : {"@id": "sdth:ProgramStep"},
                "sdth:hasSourceCode": "import pandas as pd" ,
                "sdth:hasSDTL": "\"command\": \"NoTransformOp...\"" }           ,

        {"@id": "sdtest:ProgStep002",
                "rdf:type" : {"@id": "sdth:ProgramStep"},
                "sdth:hasSourceCode": "PoliticalData = pd.read_csv(\"SmallTestPolitical.csv\")",
                "sdth:hasSDTL": "\"command\": \"Load...\"",
                "sdth:loadsFile": [{"@id": "sdtest:FileIns001"}],
                "sdth:producesDataframe": {"@id": "sdtest:DFIns001"},
                "sdth:assignsVariable": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"} ] },

        {"@id": "sdtest:ProgStep003",
                "rdf:type" : {"@id": "sdth:ProgramStep"},
                "sdth:hasSourceCode": "PersonalData = pd.read_csv(\"SmallTestPersonal.csv\")",
                "sdth:hasSDTL": "\"command\": \"Load...\"",
                "sdth:loadsFile": [{"@id": "sdtest:FileIns002"}],
                "sdth:producesDataframe": {"@id": "sdtest:DFIns002"},
                "sdth:assignsVariable": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id":
                        "sdtest:VarIns011"}, {"@id": "sdtest:VarIns012"} ] },
```

```
{"@id": "sdtest:ProgStep004",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSourceCode": "PersonalData  = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'] )",
        "sdth:hasSDTL": "\"command\": \"Compute...\"",
        "sdth:consumesDataframe": [{"@id": "sdtest:DFIns002"}],
        "sdth:producesDataframe": {"@id": "sdtest:DFIns003"},
        "sdth:assignsVariable": [{"@id": "sdtest:VarIns013"}],
        "sdth:usesVariable": [ {"@id": "sdtest:VarIns010" } ] },


{"@id": "sdtest:ProgStep005",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSourceCode": "PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999],
                include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'] )",
        "sdth:hasProgramStep": [
                {"@id": "sdtest:ProgStep005a"}, {"@id": "sdtest:ProgStep005b"}, {"@id": "sdtest:ProgStep005c"
}
                ] },

{"@id": "sdtest:ProgStep005a",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSDTL": "\"command\": \"Compute...\"",
        "sdth:consumesDataframe": [{"@id": "sdtest:DFIns003"} ],
        "sdth:producesDataframe": {"@id": "sdtest:DFIns004a"},
        "sdth:assignsVariable": [{"@id": "sdtest:VarIns014a"} ],
        "sdth:usesVariable": [ {"@id": "sdtest:VarIns013" } ] },


{"@id": "sdtest:ProgStep005b",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSDTL": "\"command\": \"SetDataType...\"",
        "sdth:consumesDataframe": [{"@id": "sdtest:DFIns004a"} ],
        "sdth:producesDataframe": {"@id": "sdtest:DFIns004b"},
        "sdth:assignsVariable": [{"@id": "sdtest:VarIns014b"} ],
        "sdth:usesVariable": [ {"@id": "sdtest:VarIns014a" } ] },


{"@id": "sdtest:ProgStep005c",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSDTL": "\"command\": \"SetValueLabels...\"",
        "sdth:consumesDataframe": [{"@id": "sdtest:DFIns004b"} ],
        "sdth:producesDataframe": {"@id": "sdtest:DFIns004c"},
        "sdth:assignsVariable": [{"@id": "sdtest:VarIns014c"} ],
        "sdth:usesVariable": [ {"@id": "sdtest:VarIns014b" } ] },

{"@id": "sdtest:ProgStep006",
        "rdf:type" : {"@id": "sdth:ProgramStep"},
        "sdth:hasSourceCode": "MergedData = PersonalData.merge(PoliticalData, on=\"ID\", how=\"inner\")",
```

```
                "sdth:consumesDataframe": [{"@id": "sdtest:DFIns001"}, {"@id": "sdtest:DFIns004c"}],
                "sdth:producesDataframe": {"@id": "sdtest:DFIns005"},
                "sdth:assignsVariable": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}, {"@id":
                        "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"}, {"@id":
                        "sdtest:VarIns012"}, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ]          ,
                "sdth:usesVariable": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}, {"@id":
                        "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"}, {"@id":
                        "sdtest:VarIns012"}, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ]   },

        {"@id": "sdtest:ProgStep007",
                "rdf:type" : {"@id": "sdth:ProgramStep"},
                "sdth:hasSourceCode": "MergedData.to_csv(\"SmallTestMerged.csv\")",
                "sdth:consumesDataframe": {"@id": "sdtest:DFIns005"},
                "sdth:savesFile": {"@id": "sdtest:FileIns003"},
                "sdth:usesVariable": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}, {"@id":
                        "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"}, {"@id":
                        "sdtest:VarIns012"}, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ]
                         }
        ] ,

"sdth:FileInstance": [
                {"@id": "sdtest:FileIns001",
                "rdf:type" : {"@id": "sdth:FileInstance" },
                "sdth:hasName": "SmallTestPolitical.csv",
                "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"} ]              },

                {"@id": "sdtest:FileIns002",
                "rdf:type" : {"@id": "sdth:FileInstance"},
                "sdth:hasName": "SmallTestPersonal.csv",
                "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id":
                        "sdtest:VarIns011"}, {"@id": "sdtest:VarIns012" } ]
                },

                {"@id": "sdtest:FileIns003",
                "rdf:type" : {"@id": "sdth:FileInstance"},
                "sdth:hasName": "SmallTestMerged.csv",
                "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id":
                        "sdtest:VarIns003"}, {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id":
                        "sdtest:VarIns006"}, {"@id": "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}, {"@id":
```

```
                    "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"}, {"@id":
                    "sdtest:VarIns012"}, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ]  ,
            "sdth:wasDerivedFrom": {"@id": "sdtest:DFIns005"}
            }
            ],

"sdth:DataframeInstance": [
        {"@id": "sdtest:DFIns001",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PoliticalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id": "sdtest:VarIns003"},
                {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id": "sdtest:VarIns006"}, {"@id":
                "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}] ,
        "sdth:wasDerivedFrom": {"@id": "sdtest:FileIns001"}     },

        {"@id": "sdtest:DFIns002",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PersonalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"},
                {"@id": "sdtest:VarIns012" } ] ,
        "sdth:wasDerivedFrom": {"@id": "sdtest:FileIns002"}     },

        {"@id": "sdtest:DFIns003",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PersonalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"},
                {"@id": "sdtest:VarIns012" }, {"@id": "sdtest:VarIns013"} ],
        "sdth:wasDerivedFrom": {"@id": "sdtest:DFIns002"}      },

        {"@id": "sdtest:DFIns004a",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PersonalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"},
                {"@id":"sdtest:VarIns012" }, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014a"} ],
        "sdth:wasDerivedFrom": {"@id": "sdtest:DFIns003"}      },

        {"@id": "sdtest:DFIns004b",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PersonalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"},
                {"@id":"sdtest:VarIns012" }, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014b"} ],
        "sdth:elaborationOf": {"@id": "sdtest:DFIns004a"}         },

        {"@id": "sdtest:DFIns004c",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "PersonalData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns009"}, {"@id": "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"},
                {"@id":"sdtest:VarIns012" }, {"@id": "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ],
```

```
        "sdth:elaborationOf": {"@id": "sdtest:DFIns004b"}          },

        {"@id": "sdtest:DFIns005",
        "rdf:type" : {"@id": "sdth:DataframeInstance"},
        "sdth:hasName": "MergedData" ,
        "sdth:hasVarInstance": [ {"@id": "sdtest:VarIns001"}, {"@id": "sdtest:VarIns002"}, {"@id": "sdtest:VarIns003"},
                {"@id": "sdtest:VarIns004"}, {"@id": "sdtest:VarIns005"}, {"@id": "sdtest:VarIns006"}, {"@id":
                "sdtest:VarIns007"}, {"@id": "sdtest:VarIns008"}, {"@id": "sdtest:VarIns009"}, {"@id":
                "sdtest:VarIns010"}, {"@id": "sdtest:VarIns011"}, {"@id": "sdtest:VarIns012"}, {"@id":
                "sdtest:VarIns013"}, {"@id": "sdtest:VarIns014c"} ]          ,
        "sdth:wasDerivedFrom": [{"@id": "sdtest:DFIns001"}, {"@id": "sdtest:DFIns004c"} ]  }
        ],

"sdth:VariableInstance" : [
        {"@id": "sdtest:VarIns001",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q3" },
        {"@id": "sdtest:VarIns002",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q244_NEW" },
        {"@id": "sdtest:VarIns003",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q356" },
        {"@id": "sdtest:VarIns004",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q330A" },
        {"@id": "sdtest:VarIns005",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q330C" },
        {"@id": "sdtest:VarIns006",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q27" },
        {"@id": "sdtest:VarIns007",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "Q1010" },
        {"@id": "sdtest:VarIns008",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "ID" },
        {"@id": "sdtest:VarIns009",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "PPEDUCAT" },
        {"@id": "sdtest:VarIns010",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "PPHHSIZE" },
        {"@id": "sdtest:VarIns011",
                "rdf:type" : {"@id": "sdth:VariableInstance"},
                "sdth:hasName": "PPRENT" },
        {"@id": "sdtest:VarIns012",
```

```
            "rdf:type" : {"@id": "sdth:VariableInstance"},
            "sdth:hasName": "ID" },
        {"@id": "sdtest:VarIns013",
            "rdf:type" : {"@id": "sdth:VariableInstance"},
            "sdth:hasName": "HHsize",
            "sdth:wasDerivedFrom": [{"@id": "sdtest:VarIns010"}] },
        {"@id": "sdtest:VarIns014a",
            "rdf:type" : {"@id": "sdth:VariableInstance"},
            "sdth:hasName": "HHcateg",
            "sdth:wasDerivedFrom": [{"@id": "sdtest:VarIns013"}] },
        {"@id": "sdtest:VarIns014b",
            "rdf:type" : {"@id": "sdth:VariableInstance"},
            "sdth:hasName": "HHcateg",
            "sdth:elaborationOf": [{"@id": "sdtest:VarIns014a"}],
            "ddi-l:NumericDomain":"<NumericDomain> <classificationLevel>\"Ordinal\"</classificationLevel>
                <type>\"Integer\"</type> </NumericDomain>"},
        {"@id": "sdtest:VarIns014c",
            "rdf:type" : {"@id": "sdth:VariableInstance"},
            "sdth:hasName": "HHcateg",
            "sdth:elaborationOf": [{"@id": "sdtest:VarIns014b"}],
            "ddi-l:NumericDomain":"<NumericDomain> <classificationLevel>\"Ordinal\"</classificationLevel>
                <type>\"Integer\"</type> </NumericDomain>",
            "ddi-l:CodeList":"<CodeList> <Code> <Value>1</Value> <Label>\"1\"</Label> </Code> <Code>
                <Value>2</Value> <Label>\"2\"</Label> </Code> <Code> <Value>3</Value> <Label>\"3-
                4\"</Label> </Code> <Code> <Value>5</Value> <Label>\"5-6\"</Label> </Code> <Code>
                <Value>7</Value> <Label>\"7-9\"</Label> </Code> <Code> <Value>10</Value>
                <Label>\"10+\"</Label> </Code> </CodeScheme>"}
        ]
}
```

5: SDTH of Example in Turtle RDF (SDTH_RDF_Example_A_turtle.ttl)

```
@prefix ddi-l: <http://ddi-l/> .
@prefix sdtest: <http://test/#> .
@prefix sdth: <http://DDI/SDTH/> .

sdtest:ProgramEx5 a sdth:Program ;
   sdth:DataframeInstance sdtest:DFIns001,
      sdtest:DFIns002,
      sdtest:DFIns003,
      sdtest:DFIns004a,
      sdtest:DFIns004b,
      sdtest:DFIns004c,
      sdtest:DFIns005 ;
   sdth:FileInstance sdtest:FileIns001,
      sdtest:FileIns002,
      sdtest:FileIns003 ;
   sdth:ProgramStep sdtest:ProgStep001,
      sdtest:ProgStep002,
      sdtest:ProgStep003,
      sdtest:ProgStep004,
      sdtest:ProgStep005,
      sdtest:ProgStep005a,
      sdtest:ProgStep005b,
      sdtest:ProgStep005c,
      sdtest:ProgStep006,
      sdtest:ProgStep007 ;
   sdth:VariableInstance sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014a,
      sdtest:VarIns014b,
      sdtest:VarIns014c ;
   sdth:hasProgramStep sdtest:ProgStep001,
      sdtest:ProgStep002,
```

```
      sdtest:ProgStep003,
      sdtest:ProgStep004,
      sdtest:ProgStep005,
      sdtest:ProgStep006,
      sdtest:ProgStep007 .

sdtest:FileIns003 a sdth:FileInstance ;
   sdth:hasName "SmallTestMerged.csv" ;
   sdth:hasVarInstance sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c ;
   sdth:wasDerivedFrom sdtest:DFIns005 .

sdtest:ProgStep001 a sdth:ProgramStep ;
   sdth:hasSDTL "\"command\": \"NoTransformOp...\"" ;
   sdth:hasSourceCode "import pandas as pd" .

sdtest:ProgStep002 a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008 ;
   sdth:hasSDTL "\"command\": \"Load...\"" ;
   sdth:hasSourceCode "PoliticalData = pd.read_csv(\"SmallTestPolitical.csv\")" ;
   sdth:loadsFile sdtest:FileIns001 ;
   sdth:producesDataframe sdtest:DFIns001 .

sdtest:ProgStep003 a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012 ;
   sdth:hasSDTL "\"command\": \"Load...\"" ;
```

```
   sdth:hasSourceCode "PersonalData = pd.read_csv(\"SmallTestPersonal.csv\")" ;
   sdth:loadsFile sdtest:FileIns002 ;
   sdth:producesDataframe sdtest:DFIns002 .

sdtest:ProgStep004 a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns013 ;
   sdth:consumesDataframe sdtest:DFIns002 ;
   sdth:hasSDTL "\"command\": \"Compute...\"" ;
   sdth:hasSourceCode "PersonalData  = PersonalData.assign(HHsize=PersonalData['PPHHSIZE'] )" ;
   sdth:producesDataframe sdtest:DFIns003 ;
   sdth:usesVariable sdtest:VarIns010 .

sdtest:ProgStep005 a sdth:ProgramStep ;
   sdth:hasProgramStep sdtest:ProgStep005a,
      sdtest:ProgStep005b,
      sdtest:ProgStep005c ;
   sdth:hasSourceCode "PersonalData['HHcateg'] = pd.cut(PersonalData['HHsize'], [1, 2, 3, 5, 7, 10, 999],
include_lowest=True, right=False, labels=['1', '2', '3-4', '5-6', '7-9', '10+'] )" .

sdtest:ProgStep005a a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns014a ;
   sdth:consumesDataframe sdtest:DFIns003 ;
   sdth:hasSDTL "\"command\": \"Compute...\"" ;
   sdth:producesDataframe sdtest:DFIns004a ;
   sdth:usesVariable sdtest:VarIns013 .

sdtest:ProgStep005b a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns014b ;
   sdth:consumesDataframe sdtest:DFIns004a ;
   sdth:hasSDTL "\"command\": \"SetDataType...\"" ;
   sdth:producesDataframe sdtest:DFIns004b ;
   sdth:usesVariable sdtest:VarIns014a .

sdtest:ProgStep005c a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns014c ;
   sdth:consumesDataframe sdtest:DFIns004b ;
   sdth:hasSDTL "\"command\": \"SetValueLabels...\"" ;
   sdth:producesDataframe sdtest:DFIns004c ;
   sdth:usesVariable sdtest:VarIns014b .

sdtest:ProgStep006 a sdth:ProgramStep ;
   sdth:assignsVariable sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
```

```
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c ;
   sdth:consumesDataframe sdtest:DFIns001,
      sdtest:DFIns004c ;
   sdth:hasSourceCode "MergedData = PersonalData.merge(PoliticalData, on=\"ID\", how=\"inner\")" ;
   sdth:producesDataframe sdtest:DFIns005 ;
   sdth:usesVariable sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c .

sdtest:ProgStep007 a sdth:ProgramStep ;
   sdth:consumesDataframe sdtest:DFIns005 ;
   sdth:hasSourceCode "MergedData.to_csv(\"SmallTestMerged.csv\")" ;
   sdth:savesFile sdtest:FileIns003 ;
   sdth:usesVariable sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c .

sdtest:FileIns001 a sdth:FileInstance ;
   sdth:hasName "SmallTestPolitical.csv" ;
   sdth:hasVarInstance sdtest:VarIns001,
```

```
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008 .

sdtest:FileIns002 a sdth:FileInstance ;
   sdth:hasName "SmallTestPersonal.csv" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012 .

sdtest:DFIns001 a sdth:DataframeInstance ;
   sdth:hasName "PoliticalData" ;
   sdth:hasVarInstance sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008 ;
   sdth:wasDerivedFrom sdtest:FileIns001 .

sdtest:DFIns002 a sdth:DataframeInstance ;
   sdth:hasName "PersonalData" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012 ;
   sdth:wasDerivedFrom sdtest:FileIns002 .

sdtest:DFIns003 a sdth:DataframeInstance ;
   sdth:hasName "PersonalData" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013 ;
   sdth:wasDerivedFrom sdtest:DFIns002 .

sdtest:DFIns004a a sdth:DataframeInstance ;
   sdth:hasName "PersonalData" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
```

```
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014a ;
   sdth:wasDerivedFrom sdtest:DFIns003 .

sdtest:DFIns004b a sdth:DataframeInstance ;
   sdth:elaborationOf sdtest:DFIns004a ;
   sdth:hasName "PersonalData" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014b .

sdtest:DFIns004c a sdth:DataframeInstance ;
   sdth:elaborationOf sdtest:DFIns004b ;
   sdth:hasName "PersonalData" ;
   sdth:hasVarInstance sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c .

sdtest:DFIns005 a sdth:DataframeInstance ;
   sdth:hasName "MergedData" ;
   sdth:hasVarInstance sdtest:VarIns001,
      sdtest:VarIns002,
      sdtest:VarIns003,
      sdtest:VarIns004,
      sdtest:VarIns005,
      sdtest:VarIns006,
      sdtest:VarIns007,
      sdtest:VarIns008,
      sdtest:VarIns009,
      sdtest:VarIns010,
      sdtest:VarIns011,
      sdtest:VarIns012,
      sdtest:VarIns013,
      sdtest:VarIns014c ;
   sdth:wasDerivedFrom sdtest:DFIns001,
      sdtest:DFIns004c .

sdtest:VarIns014a a sdth:VariableInstance ;
   sdth:hasName "HHcateg" ;
   sdth:wasDerivedFrom sdtest:VarIns013 .
```

```
sdtest:VarIns014b a sdth:VariableInstance ;
   sdth:elaborationOf sdtest:VarIns014a ;
   sdth:hasName "HHcateg" ;
   ddi-l:NumericDomain "<NumericDomain> <classificationLevel>\"Ordinal\"</classificationLevel>
<type>\"Integer\"</type> </NumericDomain>" .

sdtest:VarIns014c a sdth:VariableInstance ;
   sdth:elaborationOf sdtest:VarIns014b ;
   sdth:hasName "HHcateg" ;
   ddi-l:CodeList "<CodeList> <Code> <Value>1</Value> <Label>\"1\"</Label> </Code> <Code>
<Value>2</Value> <Label>\"2\"</Label> </Code> <Code> <Value>3</Value> <Label>\"3-4\"</Label> </Code>
<Code> <Value>5</Value> <Label>\"5-6\"</Label> </Code> <Code> <Value>7</Value> <Label>\"7-9\"</Label>
</Code> <Code> <Value>10</Value> <Label>\"10+\"</Label> </Code> </CodeScheme>" ;
   ddi-l:NumericDomain "<NumericDomain> <classificationLevel>\"Ordinal\"</classificationLevel>
<type>\"Integer\"</type> </NumericDomain>" .

sdtest:VarIns001 a sdth:VariableInstance ;
   sdth:hasName "Q3" .
sdtest:VarIns002 a sdth:VariableInstance ;
   sdth:hasName "Q244_NEW" .
sdtest:VarIns003 a sdth:VariableInstance ;
   sdth:hasName "Q356" .
sdtest:VarIns004 a sdth:VariableInstance ;
   sdth:hasName "Q330A" .
sdtest:VarIns005 a sdth:VariableInstance ;
   sdth:hasName "Q330C" .
sdtest:VarIns006 a sdth:VariableInstance ;
   sdth:hasName "Q27" .
sdtest:VarIns007 a sdth:VariableInstance ;
   sdth:hasName "Q1010" .
sdtest:VarIns008 a sdth:VariableInstance ;
   sdth:hasName "ID" .
sdtest:VarIns009 a sdth:VariableInstance ;
   sdth:hasName "PPEDUCAT" .
sdtest:VarIns011 a sdth:VariableInstance ;
   sdth:hasName "PPRENT" .
sdtest:VarIns012 a sdth:VariableInstance ;
   sdth:hasName "ID" .
sdtest:VarIns013 a sdth:VariableInstance ;
   sdth:hasName "HHsize" ;
   sdth:wasDerivedFrom sdtest:VarIns010 .
sdtest:VarIns010 a sdth:VariableInstance ;
   sdth:hasName "PPHHSIZE" .
```