



Mapeamento RDF para Ontologia dados de Entrada

- julho 05, 2021

Criar um arquivo RDF para representar cada carga de trabalho (ENTRADA):

Para cada DML criar o seguinte conjunto de triplas:

```
<tuning:AtomicStatement rdf:ID=DML_TPOH_01>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
  <tuning:hasDMLDescription>SELECT COLUNA WHERE TABLE</tuning:hasDMLDescription>
  <rdf:type rdf:resource="#DMLStatement"/>
</tuning:AtomicStatement>
```

Os campos em verde são o que possuem variação que são:

DML_TPOH_01 => ID do DML, ou seja, nome que identifique essa query.

SELECT COLUNA WHERE TABLE => conteúdo da query (DML).

Para cada Coluna criar o seguinte conjunto de triplas:

```
<tuning:Column rdf:ID="Column_R_NAME">
  <tuning:hasColumnName>R_NAME</tuning:hasColumnName>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
</tuning:Column>
```

Column_R_NAME => nome do ID da coluna, normalmente um nome parecido com o nome da Coluna no final.

R_NAME => nome da coluna no banco, da forma que ela aparece na query (DML).

Para cada Tabela criar o seguinte conjunto de tripla:

```
<tuning:Table rdf:ID= Table_Region >
  <tuning:hasTableName>REGION</tuning:hasTableName>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#NamedIndividual"/>
  <tuning:constitutedOf rdf:resource= #Column_R_NAME >
  <tuning:constitutedOf rdf:resource= #Column_R_REGIONKEY />
</tuning:Table>
```

Table_Region => nome do ID da tabela, normalmente um nome parecido com o nome da Tabela no final.

REGION => nome da tabela no banco, da forma que ela aparece na query (DML).

#Column_R_NAME => nome do ID da Coluna que pertence a essa tabela.

#Column_R_REGIONKEY => nome do ID da Coluna que pertence a essa tabela.

Para cada Índice Físico, criar o seguinte conjunto de tripla:

```
<tuning:Index rdf:ID= LINEITEM_L_PARTKEY_FKEY >
  <tuning:SecondColumn rdf:resource= #Column_L_SUPPKEY />
  <tuning:FirstColumn rdf:resource= #Column_L_PARTKEY />
</tuning:Index>
```

LINEITEM_L_PARTKEY_FKEY => nome do índice físico no banco.

#Column_L_PARTKEY => nome do ID da Coluna que pertence a esse índice, nesse exemplo é o índice na Ordem 1.

#Column_L_SUPPKEY => nome do ID da Coluna que pertence a esse índice, nesse exemplo é o índice na Ordem 2, porque é composto, índice constituídos de só 1

coluna, não terão essa propriedade.

Para cada Índice Hipotético, criar o seguinte conjunto de triplas:

```
<tuning:HypotheticalIndex rdf:ID= HYP_C_NATIONKEY_C_CUSTKEY >
  <tuning:SecondColumn rdf:resource= #Column_C_CUSTKEY />
  <tuning:hyplIndexesColumn rdf:resource= #Column_C_CUSTKEY />
  <tuning:hyplIndexesColumn rdf:resource= #Column_C_NATIONKEY />
<tuning:hasHypotheticalIndexName>HYP_C_NATIONKEY_C_CUSTKEY</tuning:hasHypo
theticalIndexName>
  <tuning:FirstColumn rdf:resource= #Column_C_NATIONKEY />
</tuning:HypotheticalIndex>
```

HYP_C_NATIONKEY_C_CUSTKEY => nome do ID do índice hipotético.

#Column_C_CUSTKEY => nome do ID da Coluna que pertence a esse índice, nesse exemplo é o índice hipotético na Ordem 2, porque é composto, índice hipotético constituídos de só 1 coluna, não terão essa propriedade.

#Column_C_CUSTKEY => nome do ID da Coluna que pertence a esse índice hipotético ,

#Column_C_NATIONKEY => nome do ID da Coluna que pertence a esse índice hipotético ,

HYP_C_NATIONKEY_C_CUSTKEY => nome do índice hipotético.

#Column_C_NATIONKEY => nome do ID da Coluna que pertence a esse índice, nesse exemplo é o índice hipotético na Ordem 1.

OBS: nesse caso não está sendo considerados as propriedades que identificam o DML de origem (dmlOriginatesHypInd) , e por qual regra foi gerada (isGeneratedBy), pois entendem-se que foi algo que já existia no banco e que não foi sugerido pela Ontologia.