Desenvolvimento de um Sistema de Informação Baseado na Web

Exploring a Netflix Catalog

Web Semântica – TP1

2023 / 2024

Grupo 5

Bruno Nunes, 80614 | David Raposo, 93395 | Rafael Gil, 118377 | Salomé Dias, 118163

1. Introdução ao tema

Dataset:

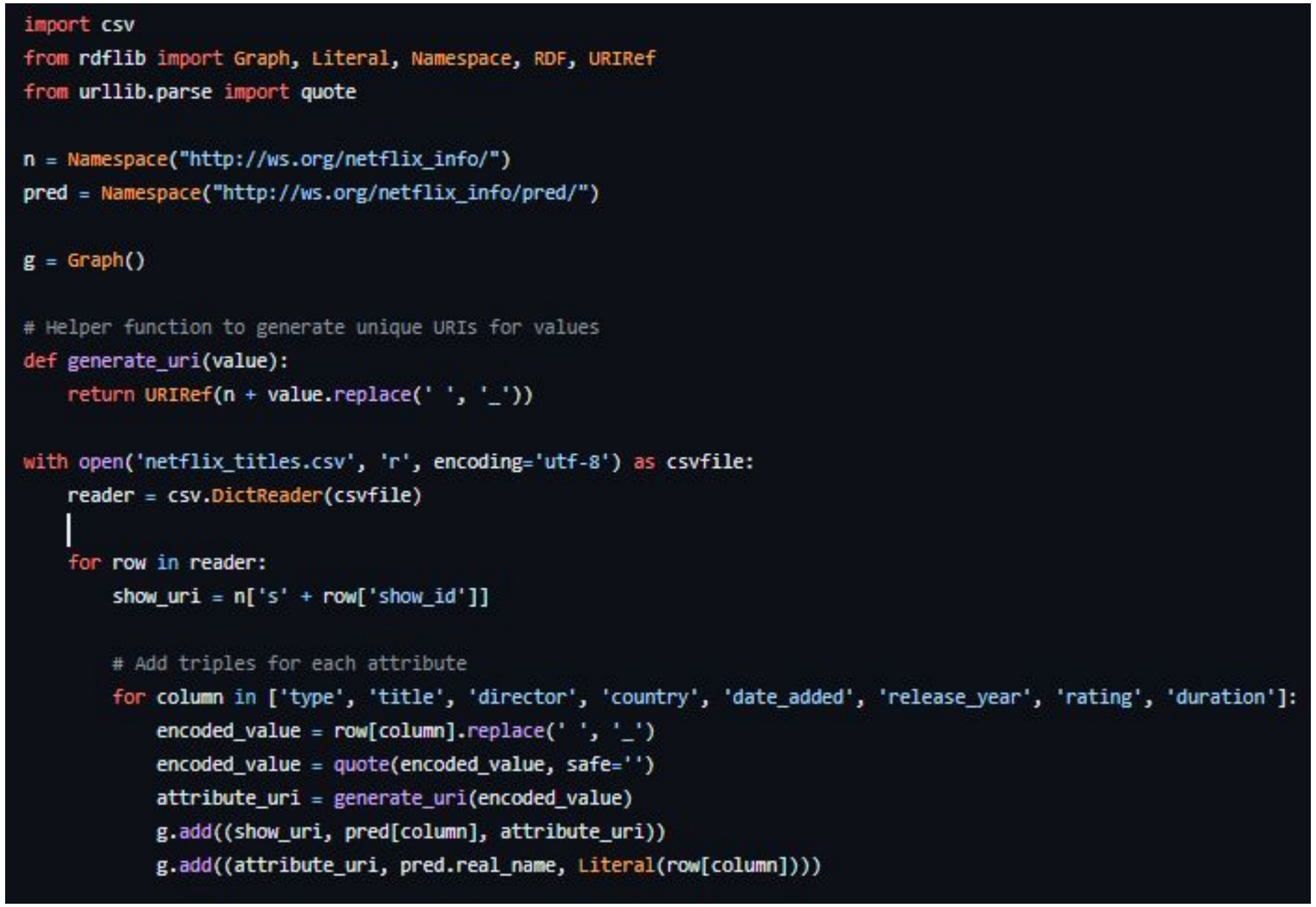
- Filmes e Séries Netflix - Kaggle

Tecnologias utilizadas:

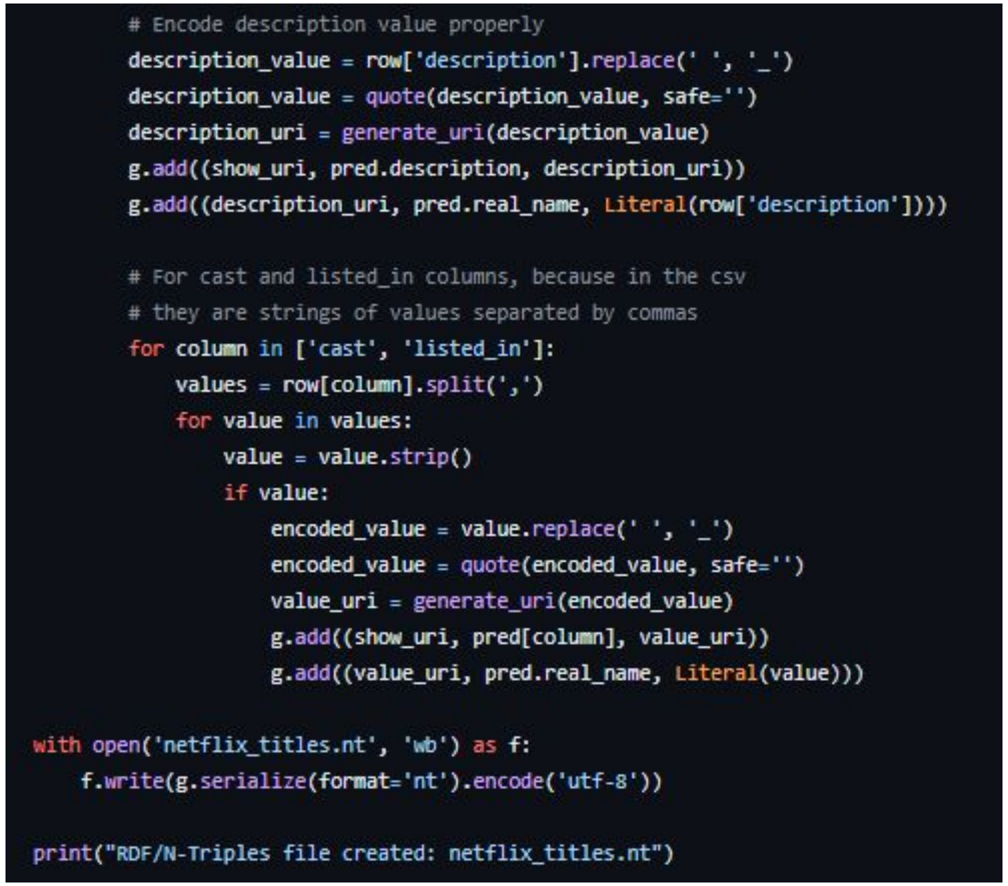
- Python/Django - para a programação da aplicação;
- RDF/NT - para o formato de dados;
- Triplestore GraphDB - como repositório de dados;
- SPARQL - para pesquisa e alteração dos dados na Triplestore;

2. Transformação dos dados

- CSV para RDF/NT

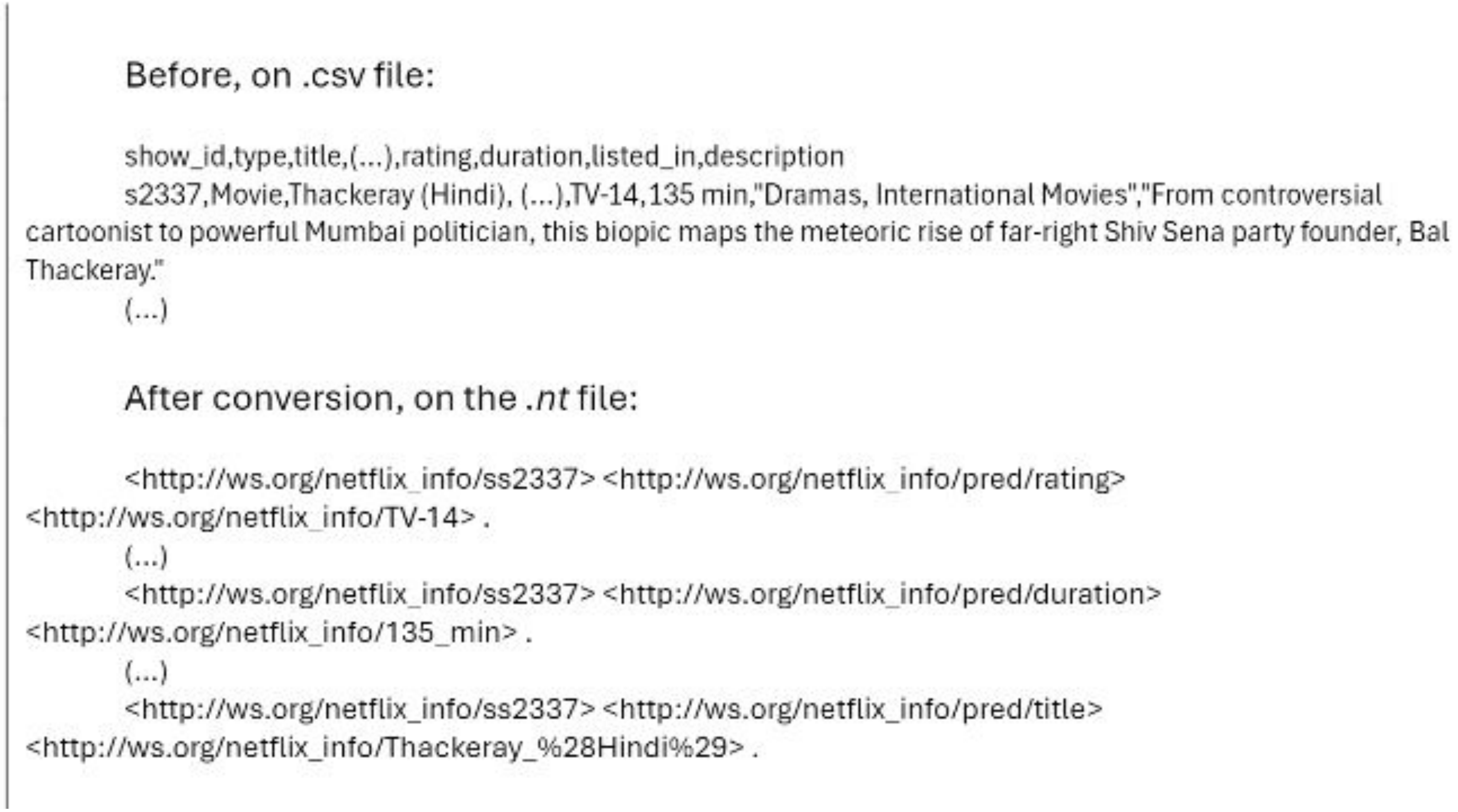


```python
import csv
from rdflib import Graph, Literal, Namespace, RDF, URIRef
from urllib.parse import quote

n = Namespace("http://ws.org/netflix_info/")
pred = Namespace("http://ws.org/netflix_info/pred/")

g = Graph()

# Helper function to generate unique URIs for values
def generate_uri(value):
    return URIRef(n + value.replace(' ', '_'))

with open('netflix_titles.csv', 'r', encoding='utf-8') as csvfile:
    reader = csv.DictReader(csvfile)

    for row in reader:
        show_uri = n['s' + row['show_id']]

        # Add triples for each attribute
        for column in ['type', 'title', 'director', 'country', 'date_added', 'release_year', 'rating', 'duration']:
            encoded_value = row[column].replace(' ', '_')
            encoded_value = quote(encoded_value, safe='')
            attribute_uri = generate_uri(encoded_value)
            g.add((show_uri, pred[column], attribute_uri))
            g.add((attribute_uri, pred.real_name, Literal(row[column])))
```



```python
        # Encode description value properly
        description_value = row['description'].replace(' ', '_')
        description_value = quote(description_value, safe='')
        description_uri = generate_uri(description_value)
        g.add((show_uri, pred.description, description_uri))
        g.add((description_uri, pred.real_name, Literal(row['description'])))

        # For cast and listed_in columns, because in the csv
        # they are strings of values separated by commas
        for column in ['cast', 'listed_in']:
            values = row[column].split(',')
            for value in values:
                value = value.strip()
                if value:
                    encoded_value = value.replace(' ', '_')
                    encoded_value = quote(encoded_value, safe='')
                    value_uri = generate_uri(encoded_value)
                    g.add((show_uri, pred[column], value_uri))
                    g.add((value_uri, pred.real_name, Literal(value)))

with open('netflix_titles.nt', 'wb') as f:
    f.write(g.serialize(format='nt').encode('utf-8'))

print("RDF/N-Triples file created: netflix_titles.nt")
```

3. Transformação dos dados



Before, on .csv file:

```
show_id,type,title,(...),rating,duration,listed_in,description
s2337,Movie,Thackeray (Hindi), (...),TV-14,135 min,"Dramas, International Movies","From controversial cartoonist to powerful Mumbai politician, this biopic maps the meteoric rise of far-right Shiv Sena party founder, Bal Thackeray."
(...)
```

After conversion, on the *.nt* file:

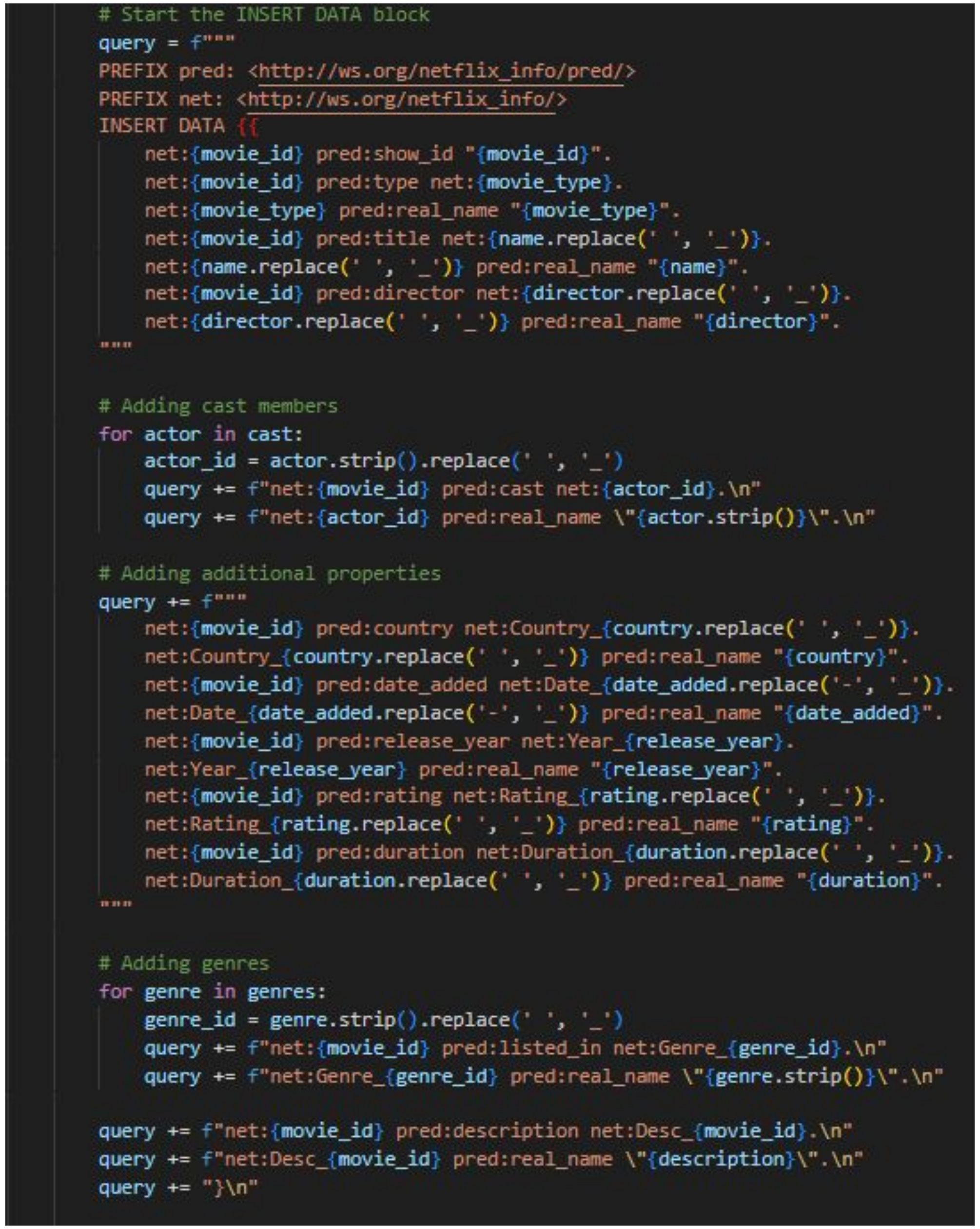
```
<http://ws.org/netflix_info/ss2337> <http://ws.org/netflix_info/pred/rating> <http://ws.org/netflix_info/TV-14> .
(...)
<http://ws.org/netflix_info/ss2337> <http://ws.org/netflix_info/pred/duration> <http://ws.org/netflix_info/135_min> .
(...)
<http://ws.org/netflix_info/ss2337> <http://ws.org/netflix_info/pred/title> <http://ws.org/netflix_info/Thackeray_%28Hindi%29> .
```

4. Operações sobre os dados (SPARQL)

Operações utilizando SPARQL

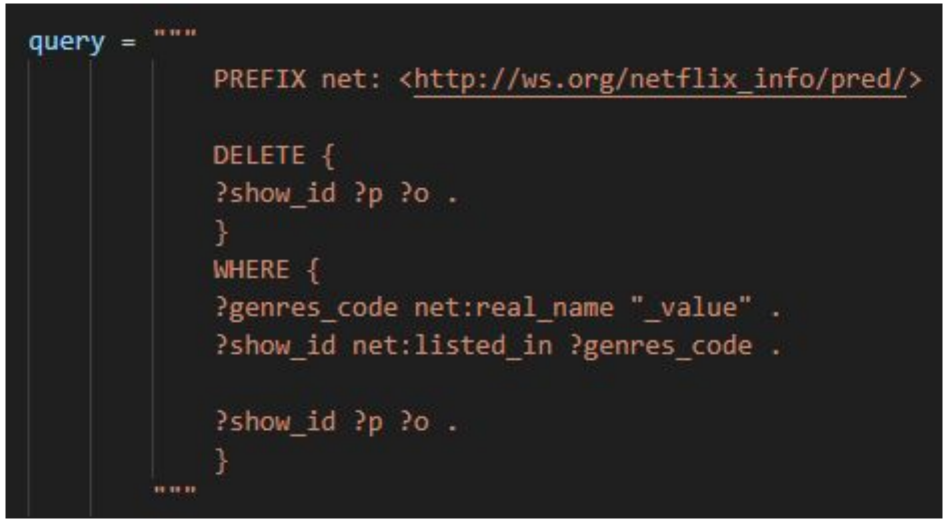
- Procura de dados
- Procura genérica
- Procura por um filme específico
- Procura de um membro em específico do elenco
- Procura por um diretor específico
- Procura de filmes lançados entre datas
- Procura de filmes lançados num dado ano
- Procura de filmes baseados em gêneros
- Inserção e exclusão de dados
- Inserção de um novo filme/série
- Exclusão de um nome

- Insert Query



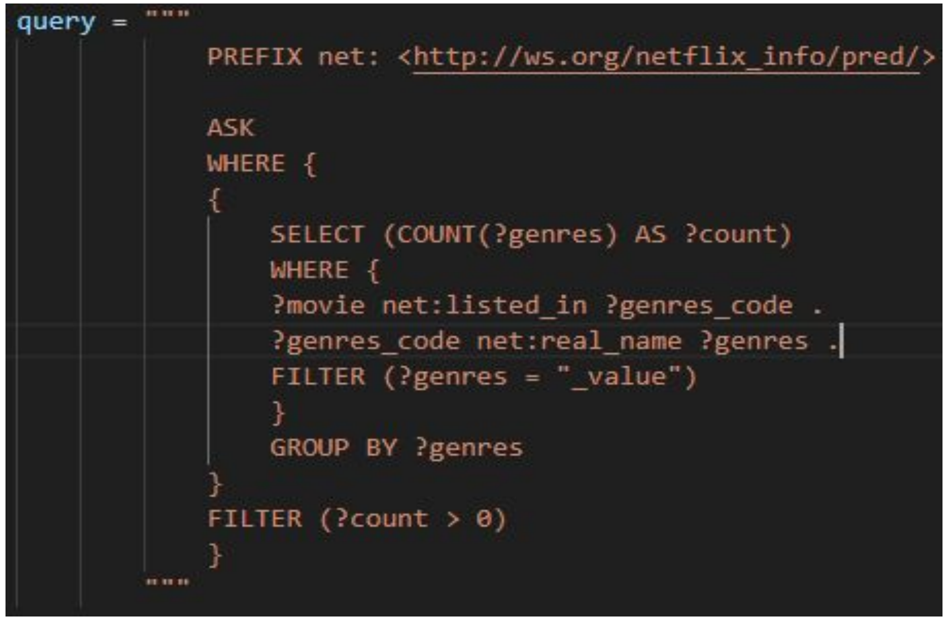
```
# Start the INSERT DATA block
query = f"""
PREFIX pred: <http://ws.org/netflix_info/pred/>
PREFIX net: <http://ws.org/netflix_info/>
INSERT DATA {{
    net:{movie_id} pred:show_id "{movie_id}".
    net:{movie_id} pred:type net:{movie_type}.
    net:{movie_type} pred:real_name "{movie_type}".
    net:{movie_id} pred:title net:{name.replace(' ', '_')}.
    net:{name.replace(' ', '_')} pred:real_name "{name}".
    net:{movie_id} pred:director net:{director.replace(' ', '_')}.
    net:{director.replace(' ', '_')} pred:real_name "{director}".
"""

# Adding cast members
for actor in cast:
    actor_id = actor.strip().replace(' ', '_')
    query += f"net:{movie_id} pred:cast net:{actor_id}.\n"
    query += f"net:{actor_id} pred:real_name \"{actor.strip()}\".\n"

# Adding additional properties
query += f"""
    net:{movie_id} pred:country net:Country_{country.replace(' ', '_')}.
    net:Country_{country.replace(' ', '_')} pred:real_name "{country}".
    net:{movie_id} pred:date_added net:Date_{date_added.replace('-', '_')}.
    net:Date_{date_added.replace('-', '_')} pred:real_name "{date_added}".
    net:{movie_id} pred:release_year net:Year_{release_year}.
    net:Year_{release_year} pred:real_name "{release_year}".
    net:{movie_id} pred:rating net:Rating_{rating.replace(' ', '_')}.
    net:Rating_{rating.replace(' ', '_')} pred:real_name "{rating}".
    net:{movie_id} pred:duration net:Duration_{duration.replace(' ', '_')}.
    net:Duration_{duration.replace(' ', '_')} pred:real_name "{duration}".
"""

# Adding genres
for genre in genres:
    genre_id = genre.strip().replace(' ', '_')
    query += f"net:{movie_id} pred:listed_in net:Genre_{genre_id}.\n"
    query += f"net:Genre_{genre_id} pred:real_name \"{genre.strip()}\".\n"

query += f"net:{movie_id} pred:description net:Desc_{movie_id}.\n"
query += f"net:Desc_{movie_id} pred:real_name \"{description}\".\n"
query += "}\n"
```

- Delete Query



```
query = """
            PREFIX net: <http://ws.org/netflix_info/pred/>

            DELETE {
            ?show_id ?p ?o .
            }
            WHERE {
            ?genres_code net:real_name "_value" .
            ?show_id net:listed_in ?genres_code .

            ?show_id ?p ?o .
            }
        """
```

- Delete Query Validation



```
query = """
            PREFIX net: <http://ws.org/netflix_info/pred/>

            ASK
            WHERE {
            {
                SELECT (COUNT(?genres) AS ?count)
                WHERE {
                ?movie net:listed_in ?genres_code .
                ?genres_code net:real_name ?genres .
                FILTER (?genres = "_value")
                }
                GROUP BY ?genres
            }
            FILTER (?count > 0)
            }
        """
```

DEMO