# Sample Query Graph2SAPRQL

## 1. Prime 10 triple che appartengono alla classe `Film` e che sono interpretate da `John Wayne`

**SPARQL**

```
SELECT ?f WHERE {
  ?f rdf:type dbo:Film .
  ?f dbo:starring dbr:John_Wayne .
} LIMIT 10
```

**GraphQL**

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    filters: [
      {
        property: "http://dbpedia.org/ontology/starring"
        value: "http://dbpedia.org/resource/John_Wayne"
      }
    ]
    limit: 10
  )
}
```

## 2. Primi 10 film in lingua italiana (descritti in lingua inglese)

**SPARQL**

```
SELECT ?f WHERE {
  ?f a dbo:Film .
  ?f dbp:language "Italian"@en .
} LIMIT 10
```

## GraphQL

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    filters: [
      {
        property: "http://dbpedia.org/property/language"
        value: "Italian@en"
        operator: "="
      }
    ]
    limit: 10
  )
}
```

## 3. Primi 10 Film interpretati da un attore nato dopo il 1 Gennaio 1980

### SPARQL

```
SELECT ?film WHERE {
  ?film a dbo:Film;
        dbp:starring ?actor .
  ?actor dbp:birthDate ?birthDate .
  FILTER(?birthDate>=xsd:date("1980-01-01"))
} LIMIT 10
```

### GraphQL

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    filters: [
      {
        path: [
          { property: "http://dbpedia.org/property/starring" },
          { property: "http://dbpedia.org/property/birthDate",
              value: "1980-01-01", operator: ">=" }
        ]
      }
    ]
    limit: 10
  )
}
```

## 4. Tutti i Film scritti da autori ancora in vita

### SPARQL

```
SELECT ?film WHERE {
  ?film a dbo:Film ;
        dbp:writer ?author .
  FILTER NOT EXISTS{?author dbp:deathDate ?deathDate}
}
```

**GraphQL**

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    filters: [
      {
        path: [
          { property: "http://dbpedia.org/property/writer" }
        ]
      }
    ]
    notExistsFilters: [
      {
        path: [
          { property: "http://dbpedia.org/property/writer" },
          { property: "http://dbpedia.org/property/deathDate" }
        ]
      }
    ]
  )
}
```

**5. Lista di Person nate in un certo luogo (`birthPlace`) la cui nazione (`dbo:country`) è l'Italia**

**SPARQL**

```
SELECT ?person WHERE {
  ?person a dbo:Person ;
          dbp:birthPlace ?bp .
  ?bp dbo:country dbr:Italy .
}
```

**GraphQL**

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Person"
    filters: [
      {
        path: [
          { property: "http://dbpedia.org/property/birthPlace" },
          { property: "http://dbpedia.org/ontology/country",
                value: "http://dbpedia.org/resource/Italy" }
        ]
      }
    ]
  )
}
```

## 6. Primi 10 Film con budget ordinati in modo ascendente

### SPARQL

```
SELECT ?film ?budget WHERE {
  ?film a dbo:Film ;
        dbo:budget ?budget .
}
ORDER BY ASC(?budget)
LIMIT 10
```

### GraphQL

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    selectFields: [
      "http://dbpedia.org/ontology/budget"
    ]
    orderBy: [
      { property: "http://dbpedia.org/ontology/budget", order: "ASC" }
    ]
    limit: 10
  )
}
```

## 7. Film scritti da autori morti, con budget opzionale

### SPARQL

```
SELECT * WHERE {
  ?film a dbo:Film;
        dbp:writer ?author .
  ?author dbp:deathDate ?deathDate .
  OPTIONAL { ?film dbo:budget ?budget }
}
```

## GraphQL

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/Film"
    filters: [
      {
        path: [
          { property: "http://dbpedia.org/property/writer" },
          { property: "http://dbpedia.org/property/deathDate" }
        ]
      }
    ]
    optionalFilters: [
      { property: "http://dbpedia.org/ontology/budget" }
    ]
    selectFields: [
      "http://dbpedia.org/ontology/budget"
    ]
  )
}
```

## 8. Top 10 città con la popolazione più alta

### SPARQL

```
SELECT ?s ?pt WHERE {
  ?s a dbo:City .
  ?s dbp:populationTotal ?pt .
}
ORDER BY DESC(?pt)
LIMIT 10
```

### GraphQL

```
query {
  searchEntity(
    className: "http://dbpedia.org/ontology/City"
    selectFields: [
      "http://dbpedia.org/property/populationTotal"
    ]
    orderBy: [
      {
        property: "http://dbpedia.org/property/populationTotal"
        order: "DESC"
      }
    ]
    limit: 10
  )
}
```