Parsing XML / HTML documents

Gaston Sanchez

About

In these slides we describe how to parse XML / HTML content with the R package **xml2**

We'll cover a variety of situations

- Navigating the XML structure
- Main functions in package xml2
- Basics of XPath

Parsing XML / HTML content

- Getting data from the Web often involves reading and processing content from XML and HTML documents. This is known as **parsing**.
- R provides two main packages for parsing XML documents: **XML** (by Duncan Temple Lang) and **xml2** by (Wickham et al)

Parsing Functions in **“xml2”**

Functions read_xml() and read_html()

xml2 comes with a main parsing function **read_xml()**

xml2 also comes with an HTML parsing function **read_html()**

The main input is either

- a file,
- a complete URL,
- or a string

What does read_xml() do?

It reads an XML document into a hierarchical structure representation

It returns an object of class `"xml_document"`

```
<movie>
  <title>
    Good Will Hunting
  </title>
  <director>
    <first_name>Gus</first_name>
    <last_name>Van Sant</last_name>
  </director>
  <year>1997</year>
  <genre>drama</genre>
</movie>
```

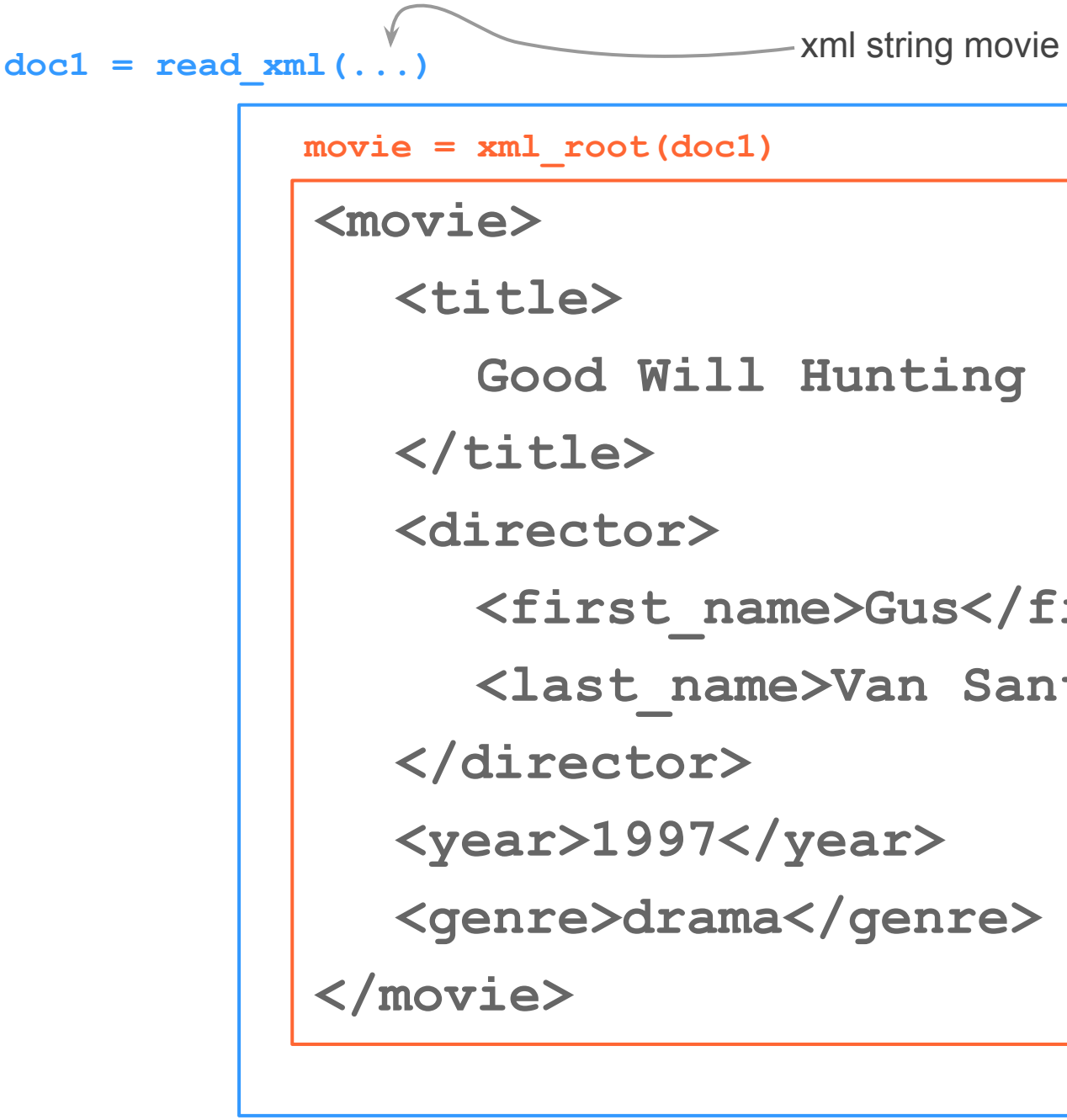
```
# toy example with xml string
movie <- read_xml(
   "<movie>
   <title>Good Will Hunting</title>
   <director>
   <first_name>Gus</first_name>
   <last_name>Van Sant</last_name>
   </director>
   <year>1997</year>
   <genre>drama</genre>
   </movie>")

class(movie)
"xml_document" "xml_node"
```

Working with parsed documents

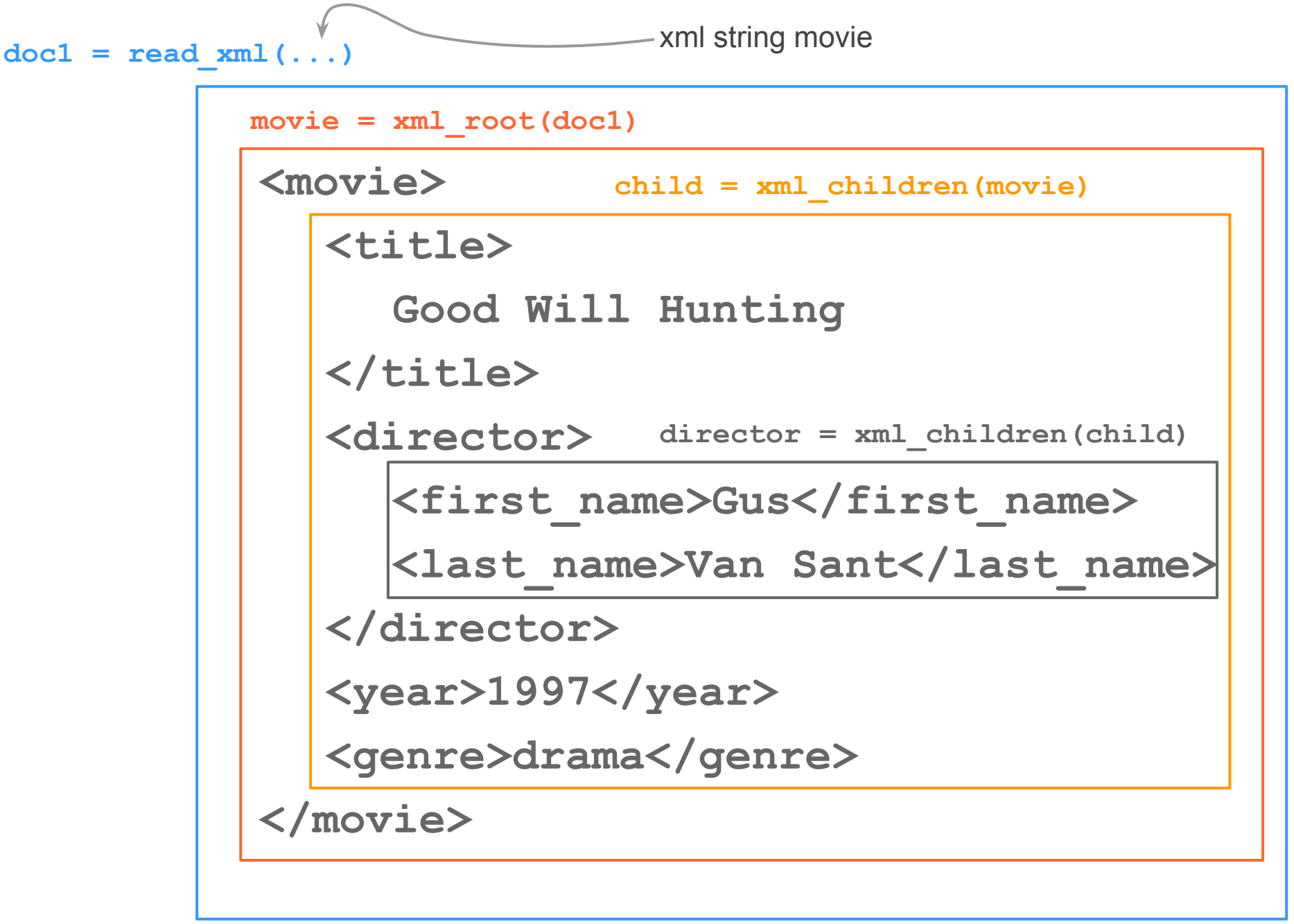
```
<movie>
   <title>
      Good Will Hunting
   </title>
   <director>
      <first_name>Gus</first_name>
      <last_name>Van Sant</last_name>
   </director>
   <year>1997</year>
   <genre>drama</genre>
</movie>
```

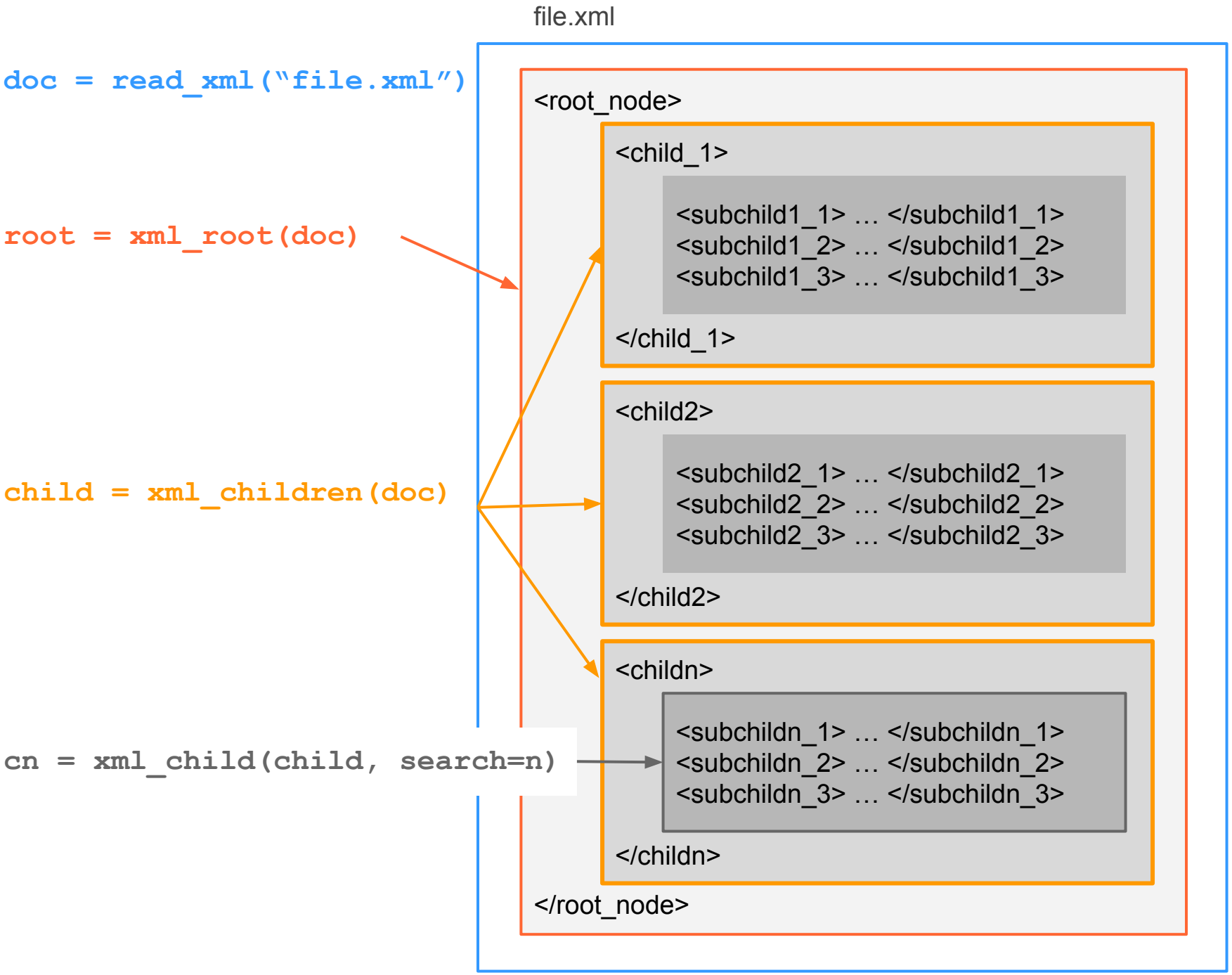
```
doc1 = read_xml(...)
```


```
<movie>
   <title>
      Good Will Hunting
   </title>
   <director>
      <first_name>Gus</first_name>
      <last_name>Van Sant</last_name>
   </director>
   <year>1997</year>
   <genre>drama</genre>
</movie>
```


doc1 = read_xml(...)
xml string movie
movie = xml_root(doc1)
<movie>
<title>
Good Will Hunting
</title>
<director>
<first_name>Gus</first_name>
<last_name>Van Sant</last_name>
</director>
<year>1997</year>
<genre>drama</genre>
</movie>


doc1 = read_xml(...)
xml string movie
movie = xml_root(doc1)
<movie>
child = xml_children(movie)
<title>
Good Will Hunting
</title>
<director>
<first_name>Gus</first_name>
<last_name>Van Sant</last_name>
</director>
<year>1997</year>
<genre>drama</genre>
</movie>


doc1 = read_xml(...)
xml string movie
movie = xml_root(doc1)
<movie>
child = xml_children(movie)
<title>
Good Will Hunting
</title>
<director>
director = xml_children(child)
<first_name>Gus</first_name>
<last_name>Van Sant</last_name>
</director>
<year>1997</year>
<genre>drama</genre>
</movie>

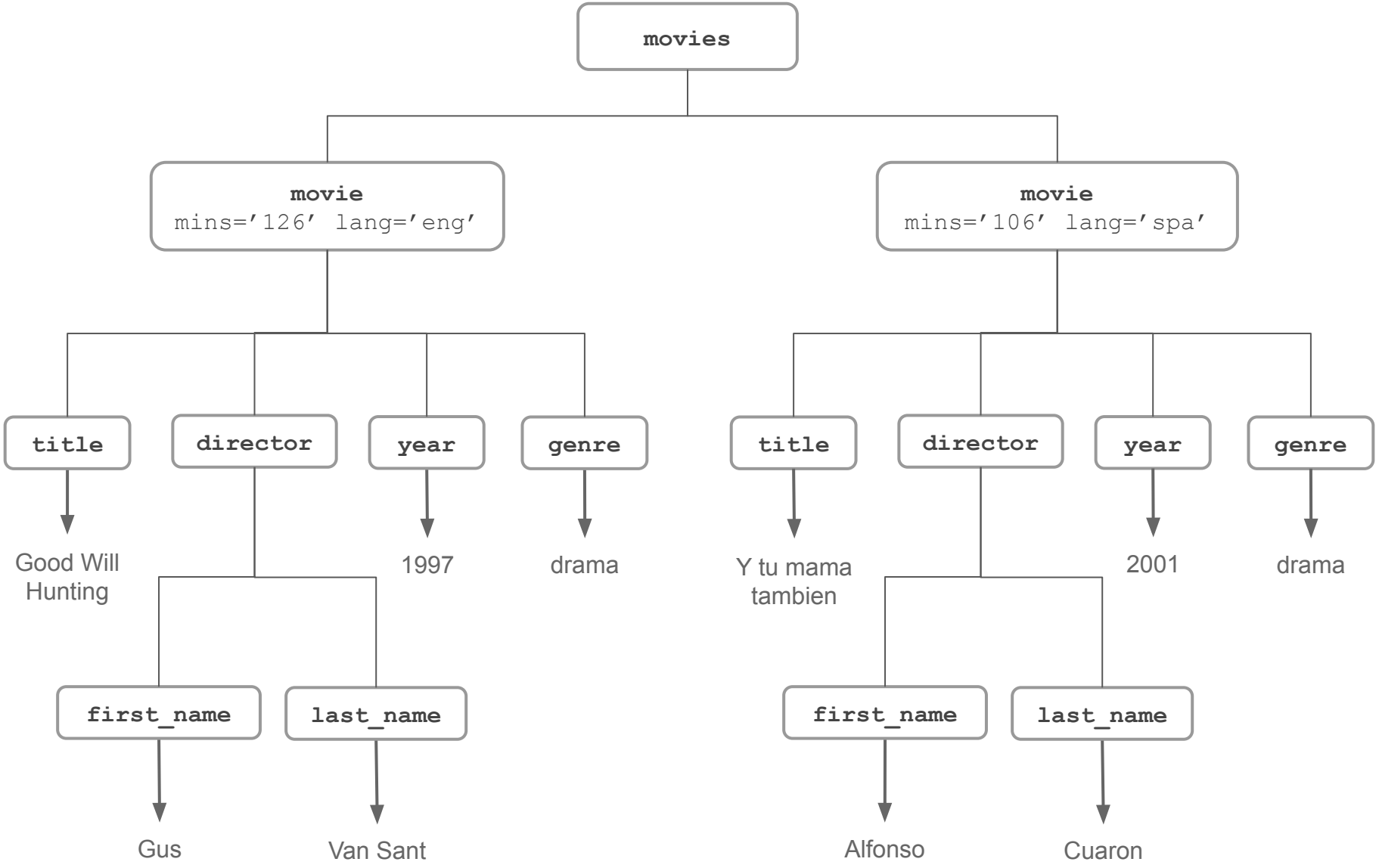

file.xml
doc = read_xml("file.xml")
root = xml_root(doc)
child = xml_children(doc)
cn = xml_child(child, search=n)
<root_node>
<child_1>
<subchild1_1> … </subchild1_1>
<subchild1_2> … </subchild1_2>
<subchild1_3> … </subchild1_3>
</child_1>
<child2>
<subchild2_1> … </subchild2_1>
<subchild2_2> … </subchild2_2>
<subchild2_3> … </subchild2_3>
</child2>
<childn>
<subchildn_1> … </subchildn_1>
<subchildn_2> … </subchildn_2>
<subchildn_3> … </subchildn_3>
</childn>
</root_node>

Navigation of XML /HTML tree

Function	Description
`xml_root`	Returns root node
`xml_children`	Returns children nodes
`xml_child`	Returns specified children number
`xml_name`	Returns name of a node
`xml_contents`	Returns contents of a node
`xml_text`	Returns text
`xml_length`	Returns number of children nodes
`xml_parents`	Returns set of parent nodes
`xml_siblings`	Returns set of sibling nodes

Example

```
<movies>
   <movie mins="126" lang="eng">
      <title>Good Will Hunting</title>
      <director>
         <first_name>Gus</first_name>
         <last_name>Van Sant</last_name>
      </director>
      <year>1997</year>
      <genre>drama</genre>
   </movie>
   <movie mins="106" lang="spa">
      <title>Y tu mama tambien</title>
      <director>
         <first_name>Alfonso</first_name>
         <last_name>Cuaron</last_name>
      </director>
      <year>2001</year>
      <genre>drama</genre>
   </movie>
</movies>
```


movies
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

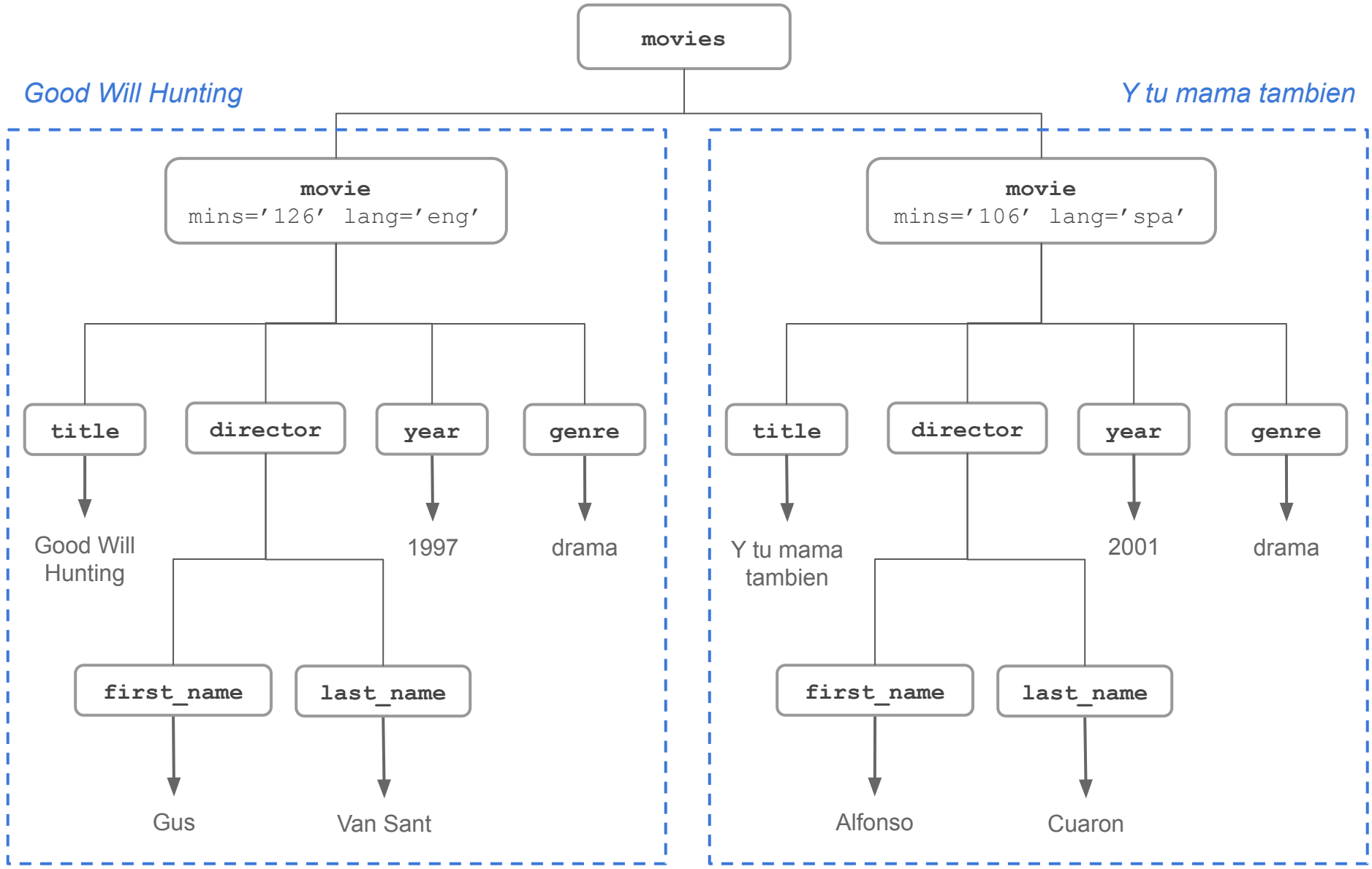
```
# toy example with xml string
xml_string <- c(
  '<?xml version="1.0" encoding="UTF-8"?>',
  '<movies>',
  '<movie mins="126" lang="eng">',
  '<title>Good Will Hunting</title>',
  '<director>',
  '<first_name>Gus</first_name>',
  '<last_name>Van Sant</last_name>',
  '</director>',
  '<year>1997</year>',
  '<genre>drama</genre>',
  '</movie>',
  '<movie mins="106" lang="spa">',
  '<title>Y tu mama tambien</title>',
  '<director>',
  '<first_name>Alfonso</first_name>',
  '<last_name>Cuaron</last_name>',
  '</director>',
  '<year>2001</year>',
  '<genre>drama</genre>',
  '</movie>',
  '</movies>')
```

Good Will Hunting

Y tu mama tambien

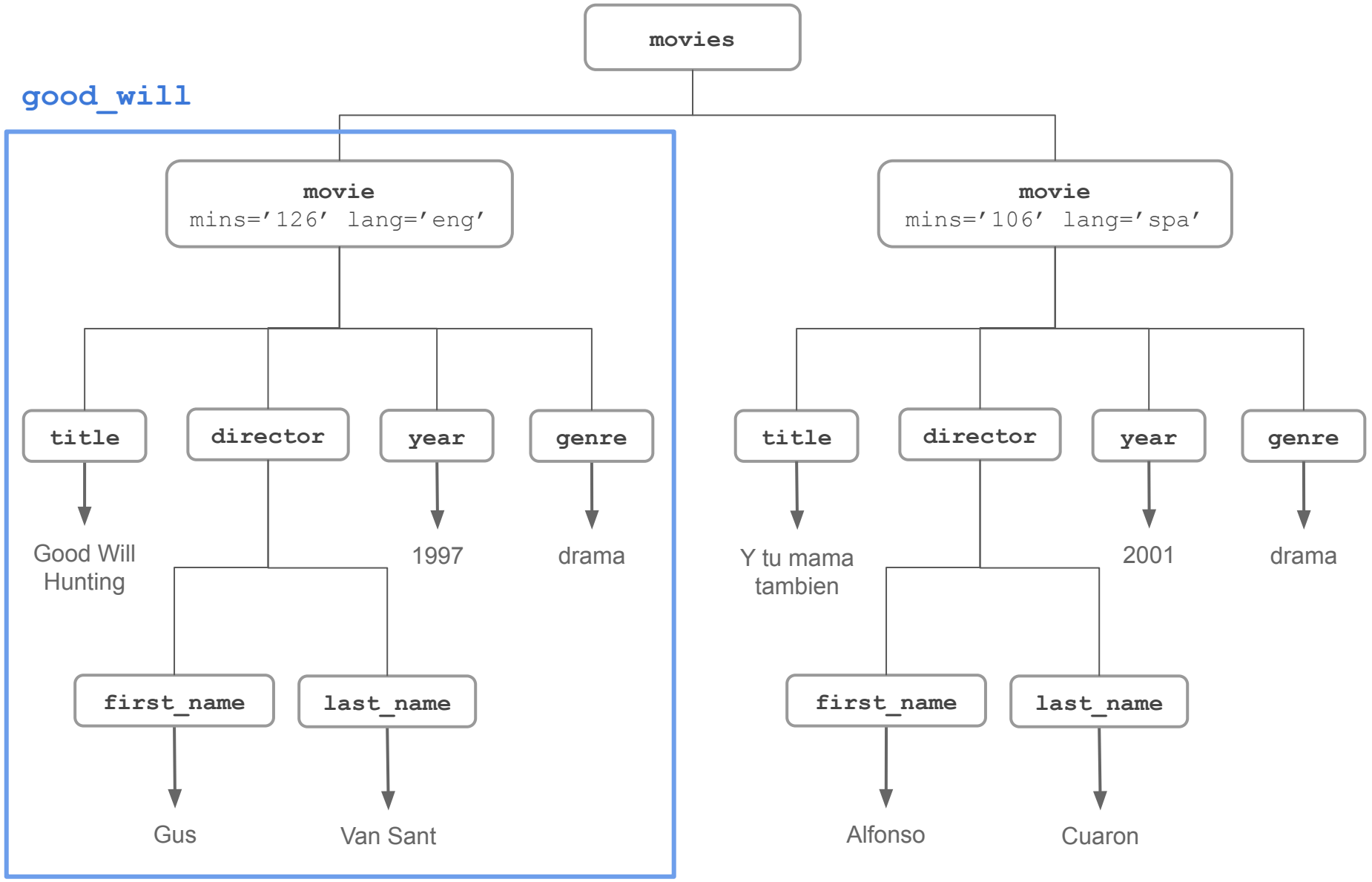
```
# parsing xml string
doc <- read_xml(paste(xml_string, collapse = ''))

doc
{xml_document}
<movies>
[1] <movie mins="126" lang="eng">\n  <title>Good Will
Hunting ...
[2] <movie mins="106" lang="spa">\n  <title>Y tu mama
tambien ...

class(doc)
[1] "xml_document" "xml_node"

# root node
movies <- xml_root(doc)

identical(doc, movies)
[1] TRUE
```

```
# parsing xml string
xml_length(doc)
[1] 2

xml_children(doc)
{xml_nodeset (2)}
[1] <movie mins="126" lang="eng">\n  <title>Good Will
Hunting ...
[2] <movie mins="106" lang="spa">\n  <title>Y tu mama
tambien ...


xml_child(doc, search = 1)
{xml_node}
<movie mins="126" lang="eng">
[1] <title>Good Will Hunting</title>
[2] <director>\n  <first_name>Gus</first_name>\n
<last_name> ...
[3] <year>1997</year>
[4] <genre>drama</genre>
```


movies
Good Will Hunting
Y tu mama tambien
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

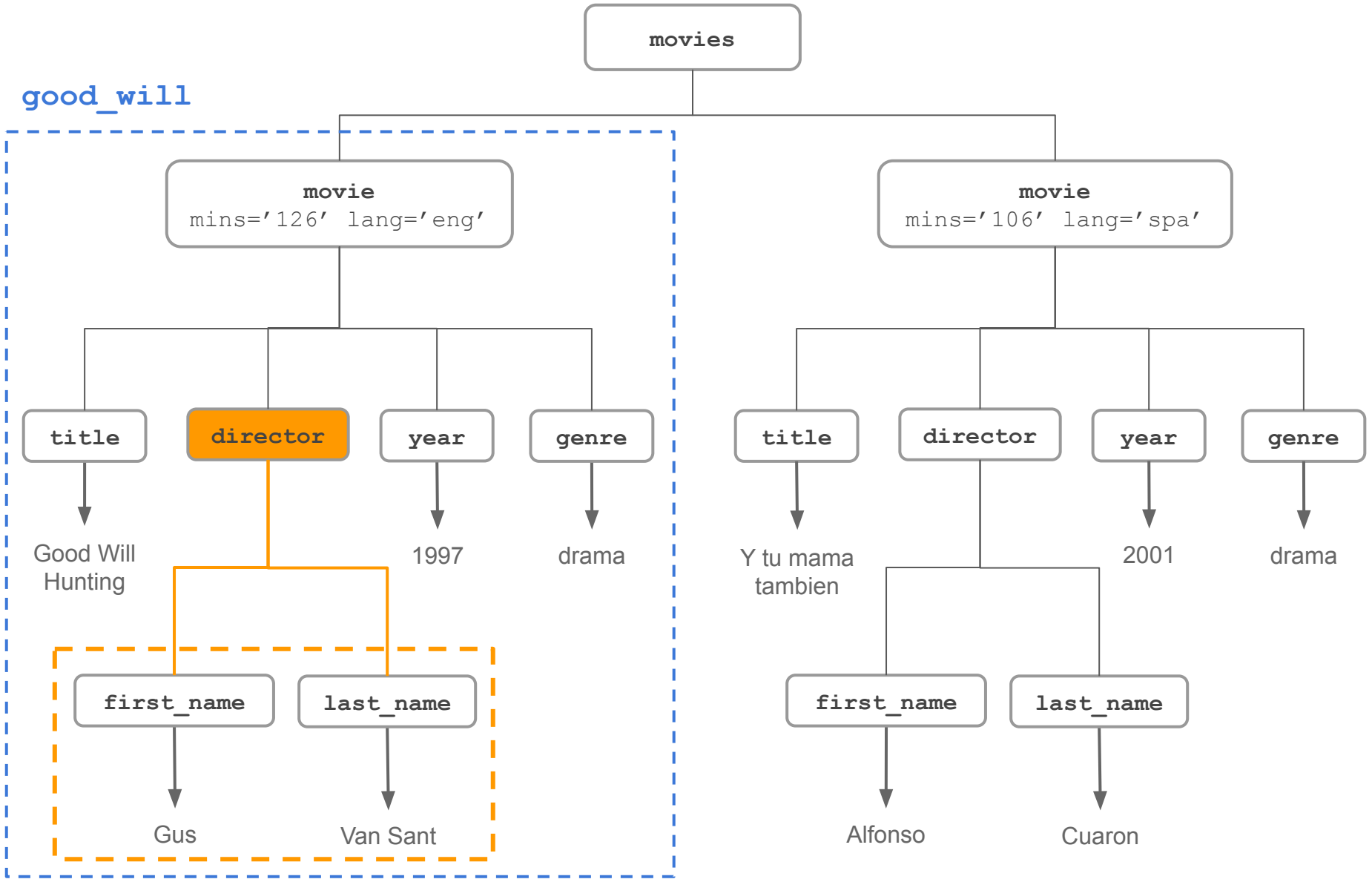
```
# first child
good_will <- xml_child(doc, search = 1)
good_will
{xml_node}
<movie mins="126" lang="eng">
[1] <title>Good Will Hunting</title>
[2] <director>\n  <first_name>Gus</first_name>\n
<last_name> ...
[3] <year>1997</year>
[4] <genre>drama</genre>


# second child
tu_mama <- xml_child(doc, search = 2)
tu_mama
{xml_node}
<movie mins="106" lang="spa">
[1] <title>Y tu mama tambien</title>
[2] <director>\n  <first_name>Alfonso</first_name>\n
<last_n ...
[3] <year>2001</year>
[4] <genre>drama</genre>
```

```
# children of good_will
xml_children(good_will)
{xml_nodeset (4)}
[1] <title>Good Will Hunting</title>
[2] <director>\n  <first_name>Gus</first_name>\n
<last_name> ...
[3] <year>1997</year>
[4] <genre>drama</genre>


# children of tu_mama
xml_children(tu_mama)
{xml_nodeset (4)}
[1] <title>Y tu mama tambien</title>
[2] <director>\n  <first_name>Alfonso</first_name>\n
<last_n ...
[3] <year>2001</year>
[4] <genre>drama</genre>
```


movies
good_will
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

```
# name of an element
xml_name(good_will)
[1] "movie"

# attributes
xml_attrs(good_will)
 mins  lang
"126" "eng"

# how many children
xml_length(good_will)
[1] 4

# name of children (of good_will)
xml_name(xml_children(good_will))
[1] "title"    "director" "year"     "genre"

# good_will title
xml_child(good_will, "title")
{xml_node}
<title>
```

```
# good_will title
title1 <- xml_child(good_will, "title")
title1
{xml_node}
<title>


# content good_will title
xml_contents(title1)
{xml_nodeset (1)}
[1] Good Will Hunting


# text good_will title
xml_text(title1)
[1] "Good Will Hunting"
```


movies
good_will
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

```
# good_will director
dir1 <- xml_child(good_will, "director")
dir1
{xml_node}
<director>
[1] <first_name>Gus</first_name>
[2] <last_name>Van Sant</last_name>


xml_children(dir1)
{xml_nodeset (2)}
[1] <first_name>Gus</first_name>
[2] <last_name>Van Sant</last_name>


xml_text(dir1)
[1] "GusVan Sant"
```

XPath Language

XPath for querying trees

The real parsing power comes from the ability to **locate nodes and extract information** from them.

To do this, we need to be able to perform queries on the parsed content.

The solution is provided by **XPath**, which is a language to navigate through the elements and attributes in an XML / HTML document.

XPath for querying trees

XPath is a language for finding information in an XML document

Works by identifying patterns to match data or content

Uses path expressions to select nodes, based on:

- node names
- node content (attributes and values)
- a node's relationship to other nodes

XPath for querying trees

The key concept is knowing how to write XPath expressions, which have a syntax similar to the way files are located in a hierarchy of directories/folders in a computer file system. For instance:

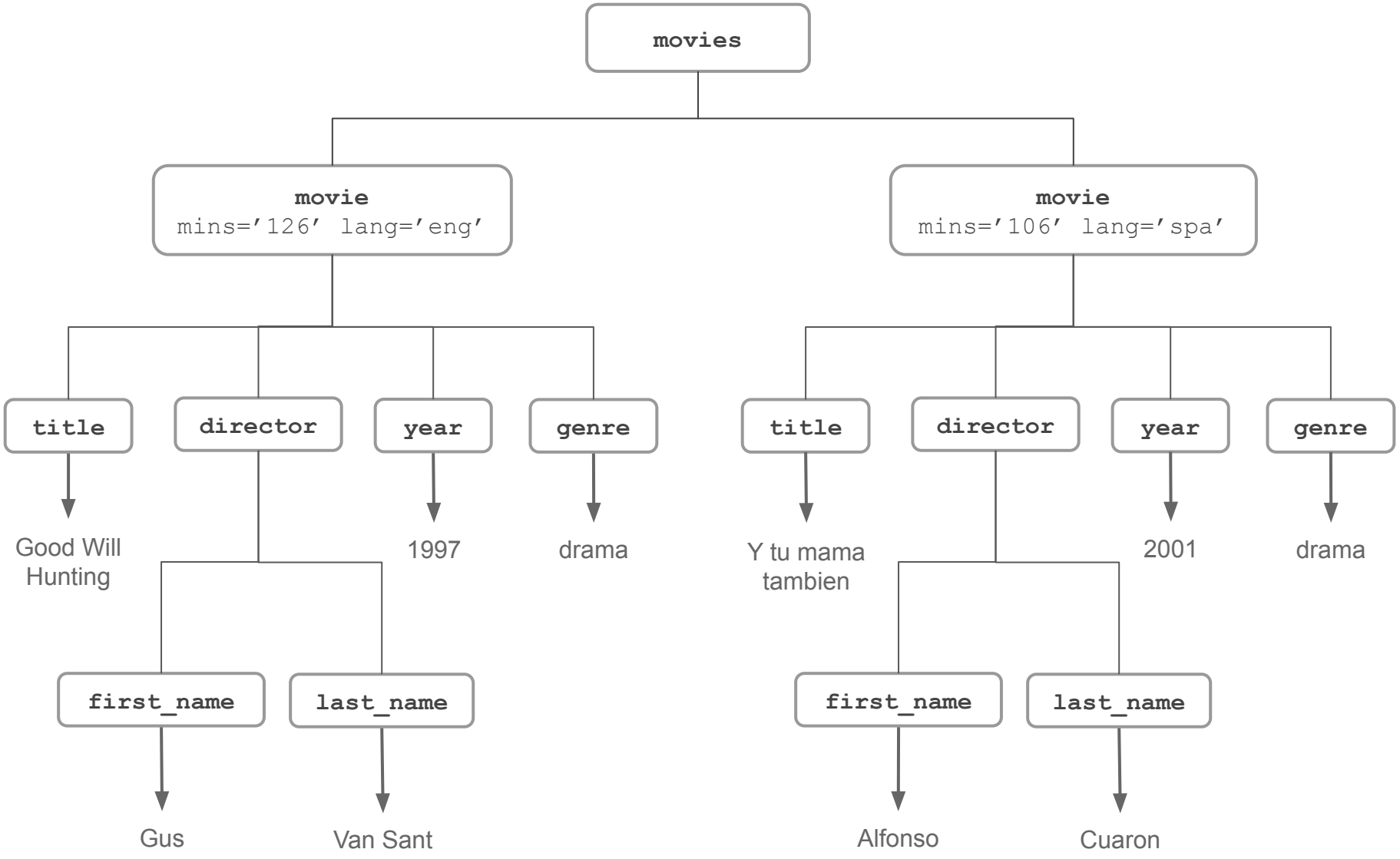
```
/movies/movie
```

is the XPath expression to locate the movie children in the movies (root) element

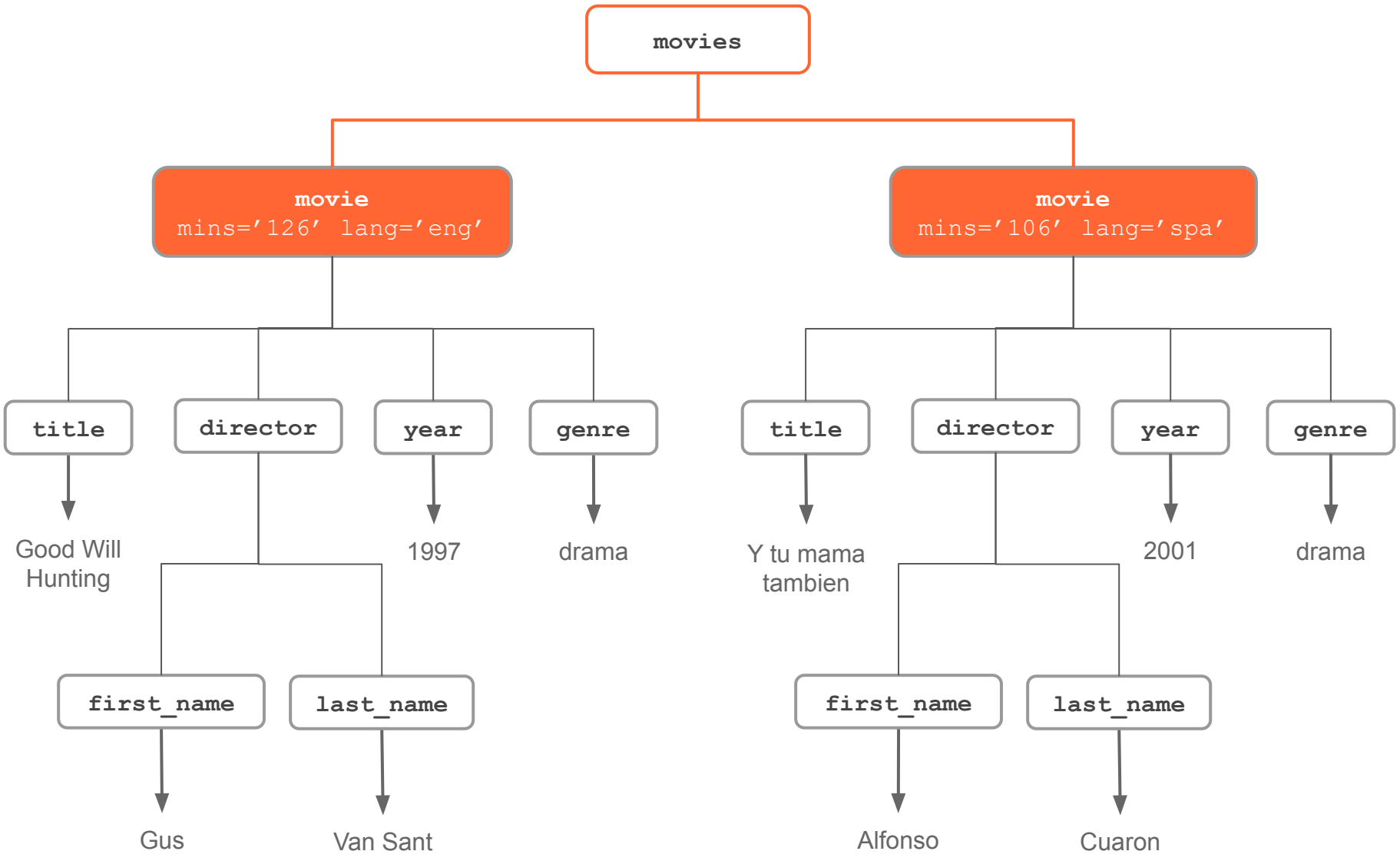
Symbol	Description
`/`	Starting slash indicates root
`//`	Double slash indicates anywhere
`.`	Selects the current node
`..`	Selects the parent of the current node
`@`	Selects attributes
`[]`	Square brackets to indicate attributes
`*`	Wildcard that matches any element
`@*`	Matches any attribute node

Example	Description
`/node`	Selects top level node
`//node`	Select nodes at any level
`node[@attr]`	Node that has an attribute named **`attr`**
`node[@attr="abc"]`	Node that has an attribute named **`attr`** with value **`"abc"`**
`node/@attr`	Value of an attribute **`attr`** in node with such attribute
`node/*`	Any (child) element in node
`node/@*`	Value of any attribute in node

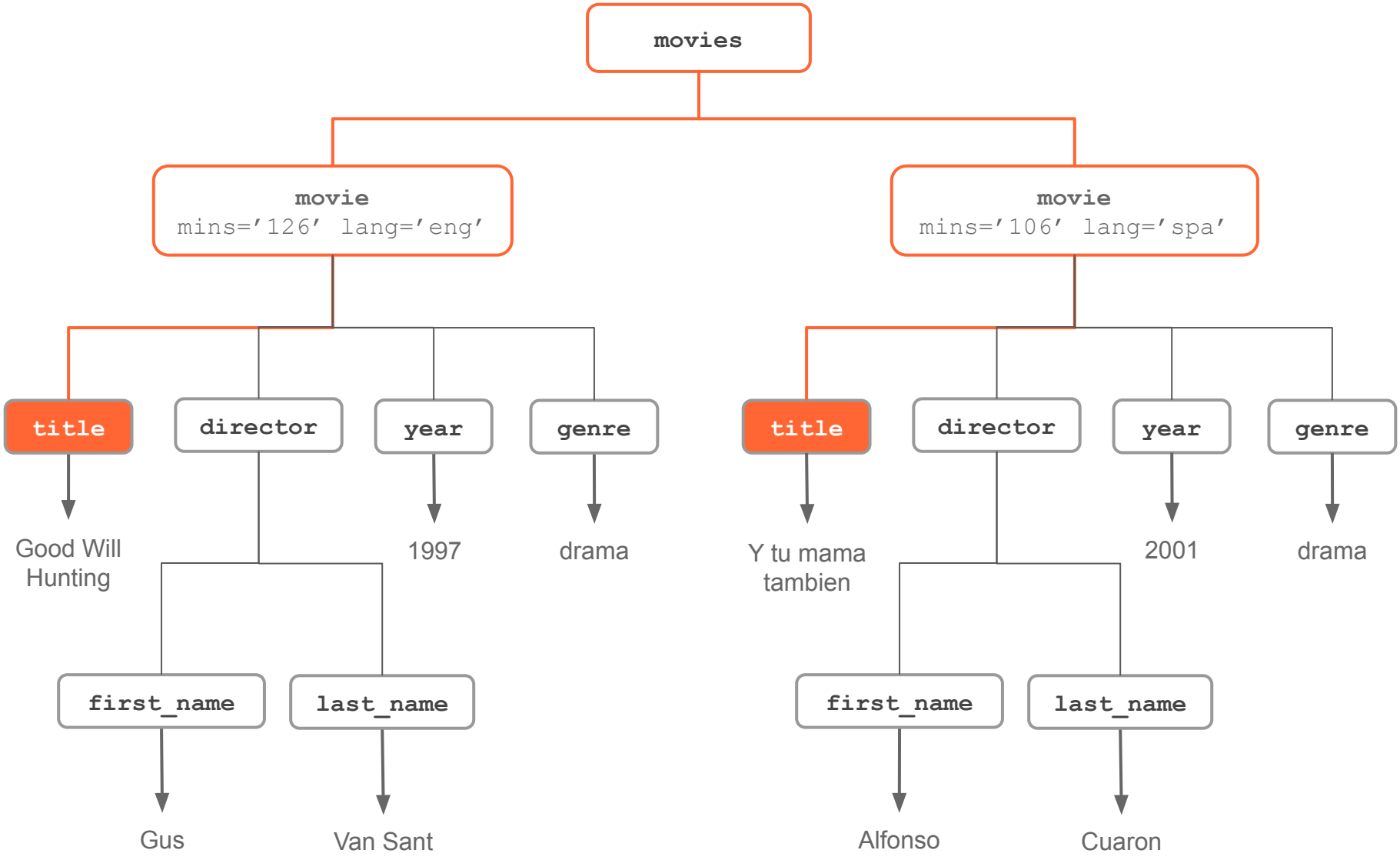
```
<movies>
   <movie mins="126" lang="eng">
      <title>Good Will Hunting</title>
      <director>
         <first_name>Gus</first_name>
         <last_name>Van Sant</last_name>
      </director>
      <year>1997</year>
      <genre>drama</genre>
   </movie>
   <movie mins="106" lang="spa">
      <title>Y tu mama tambien</title>
      <director>
         <first_name>Alfonso</first_name>
         <last_name>Cuaron</last_name>
      </director>
      <year>2001</year>
      <genre>drama</genre>
   </movie>
</movies>
```


movies
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

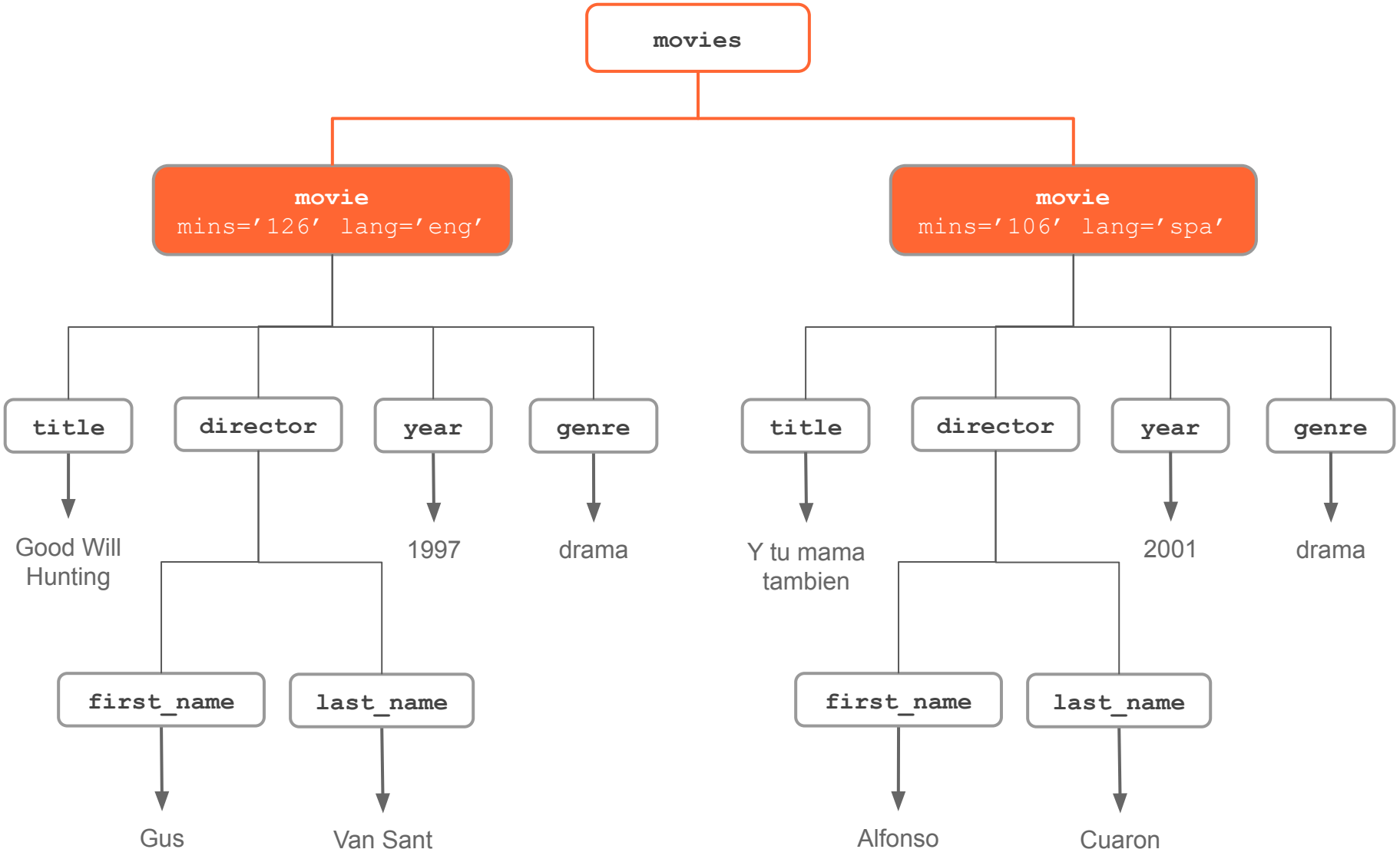
/movies/movie



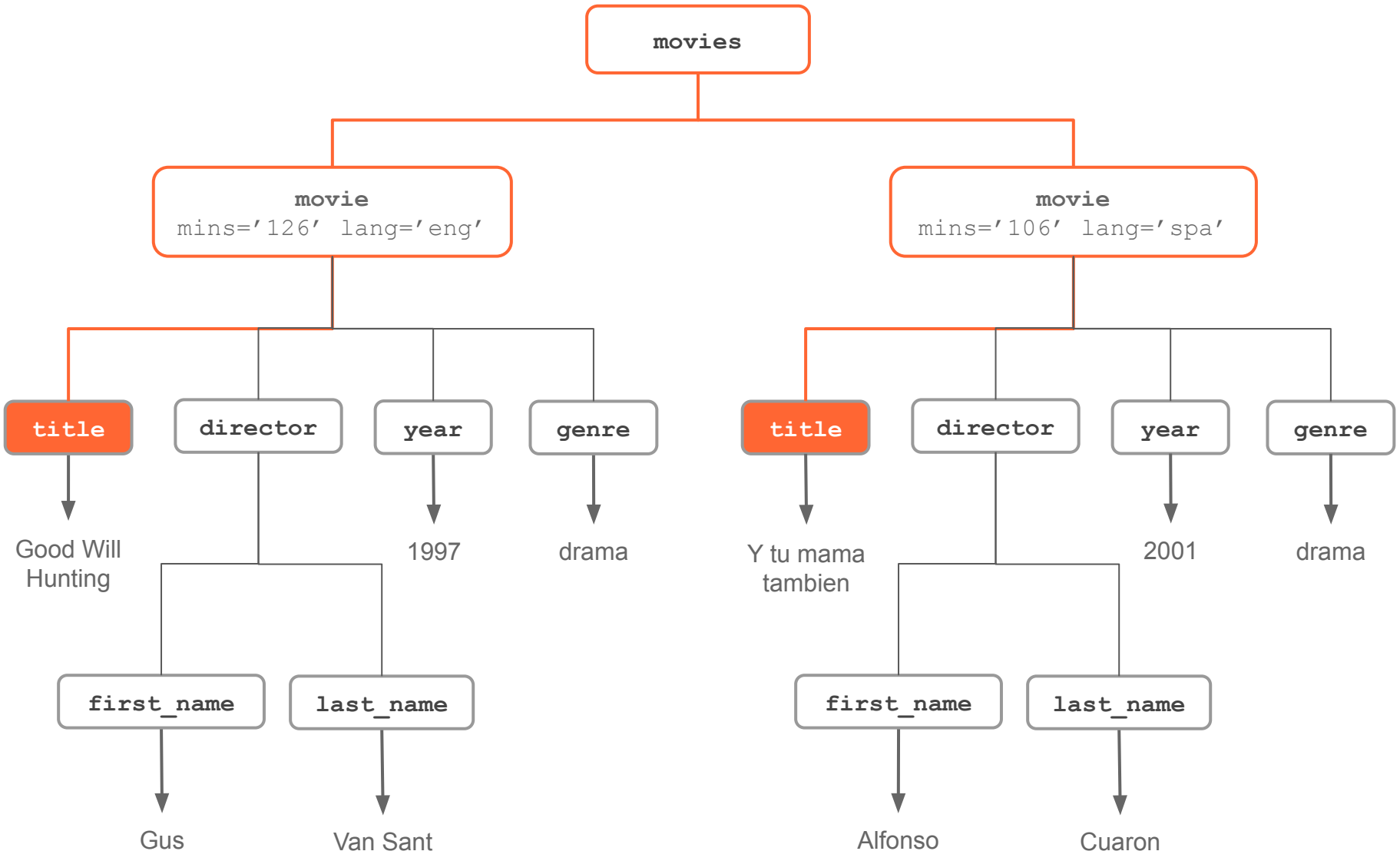
/movies/movie/title



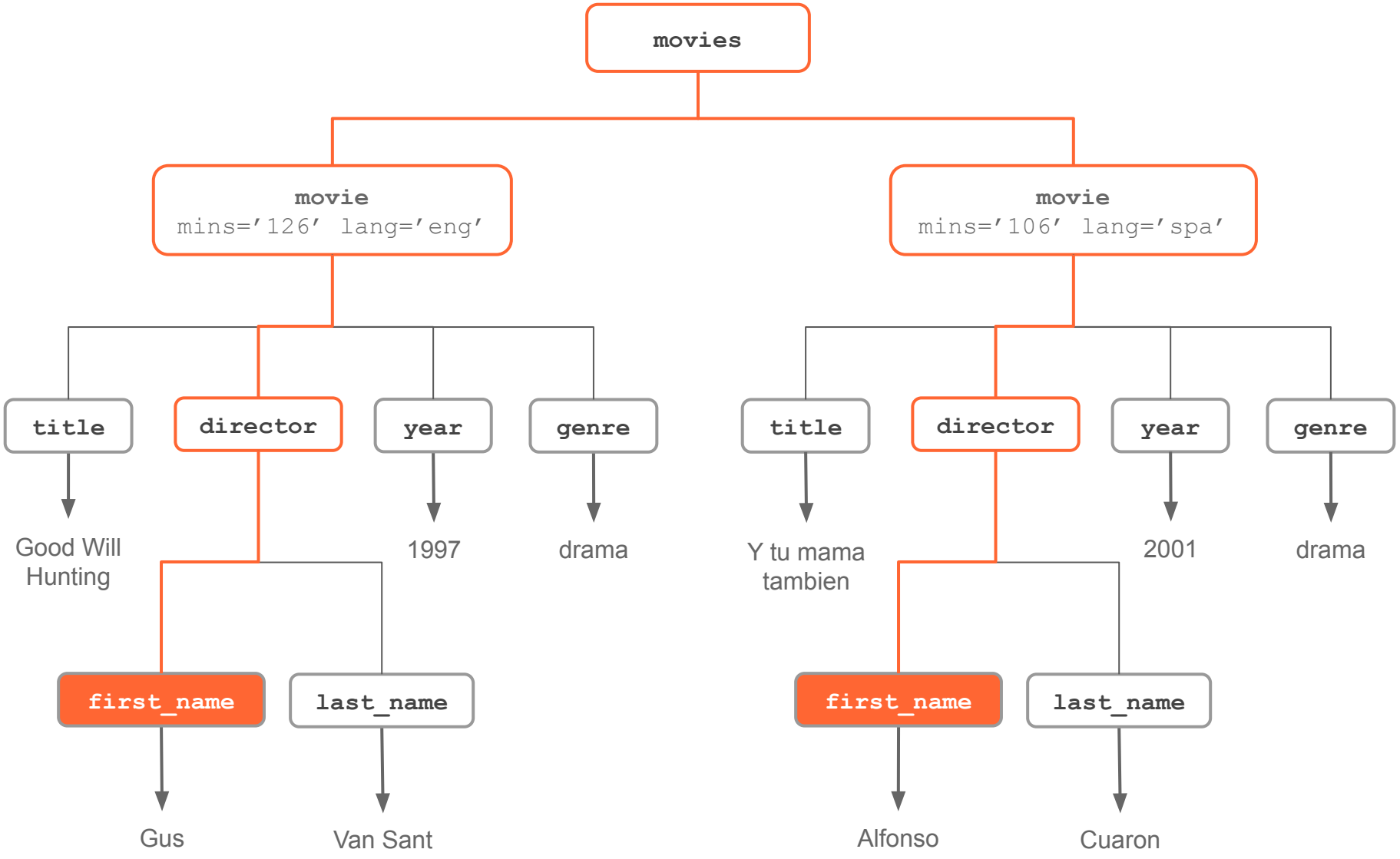
/movies/*


movies
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

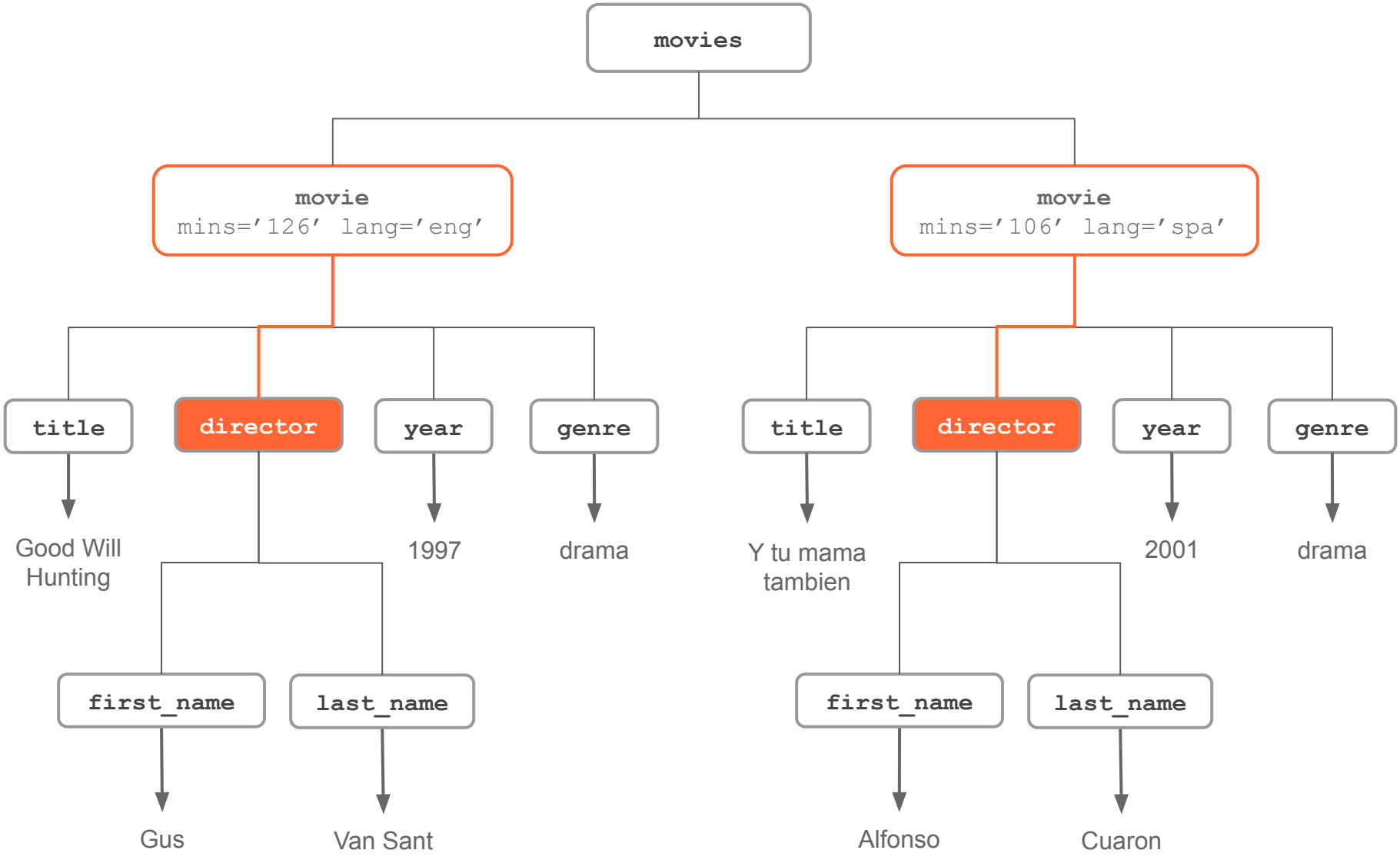
/movies/*/title



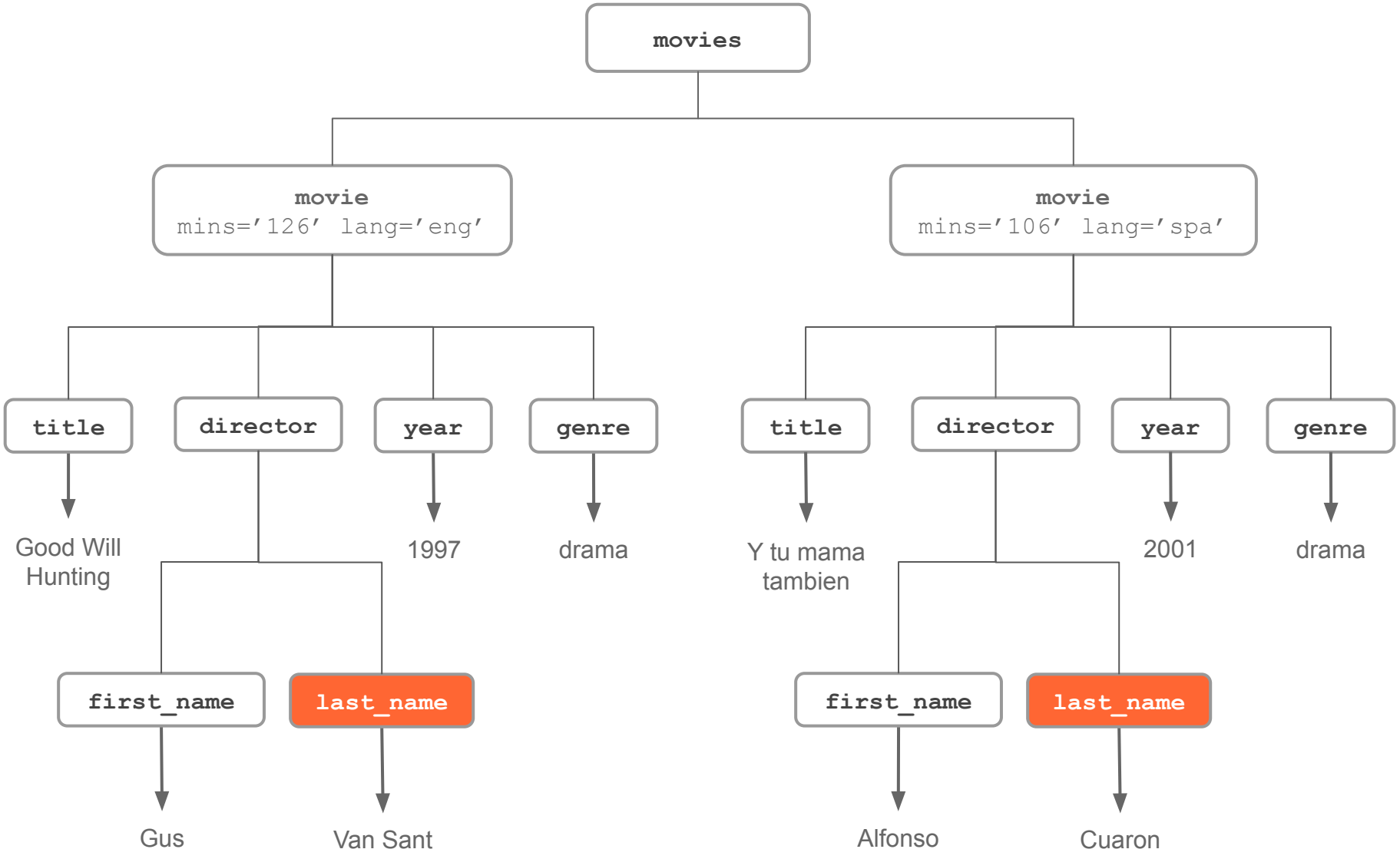
/movies/movie/director/first_name



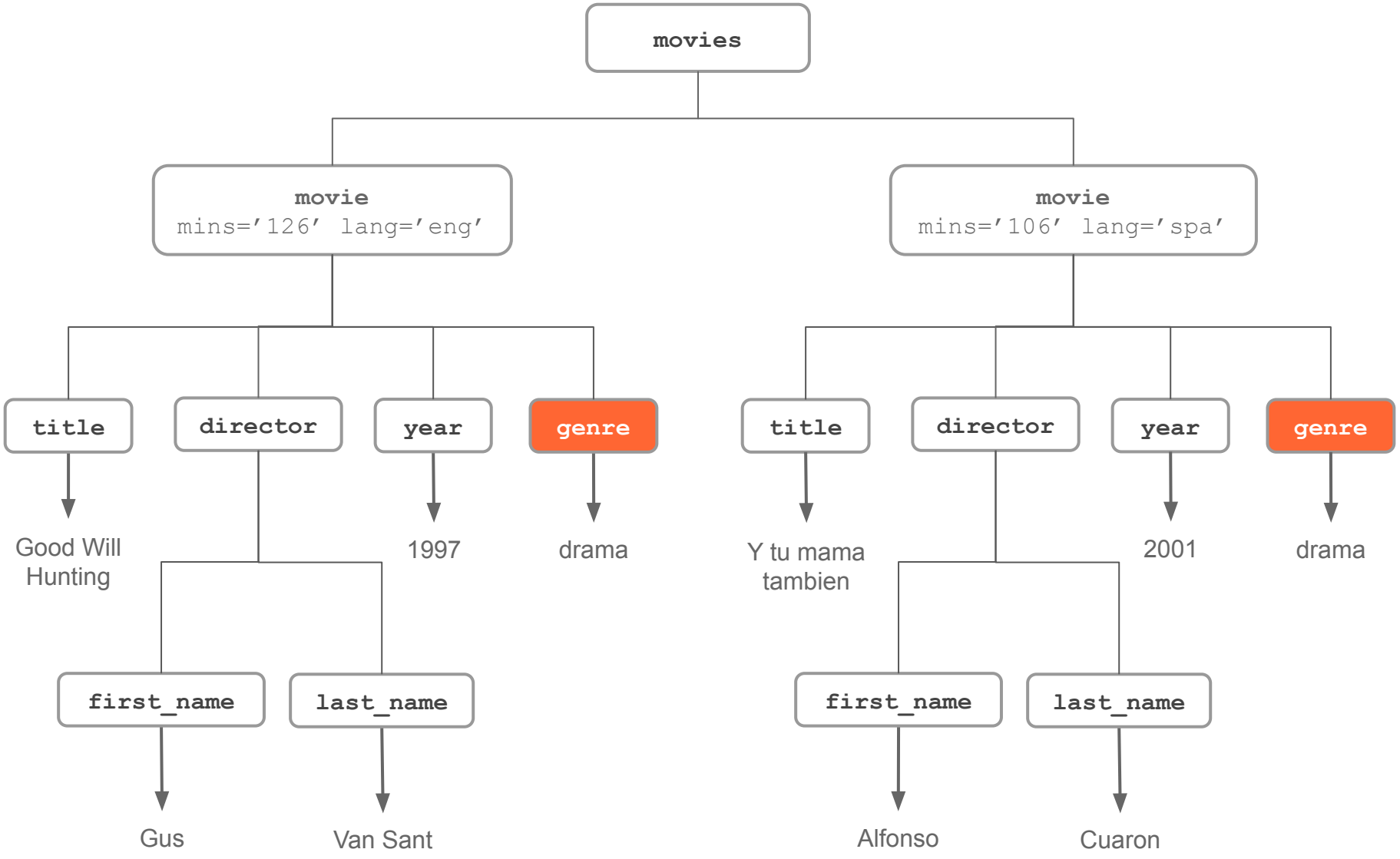
//movie/director



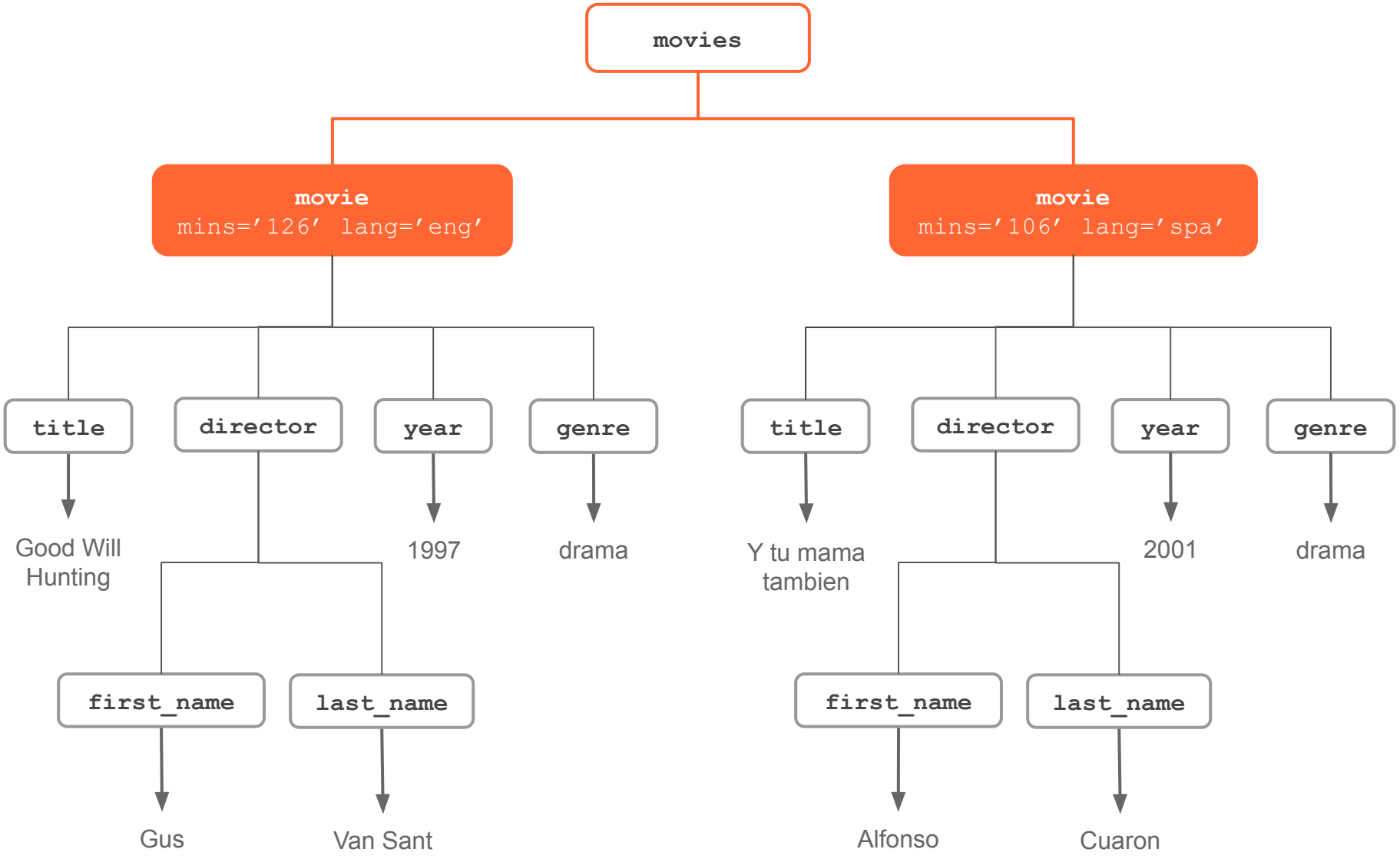
//last_name


movies
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

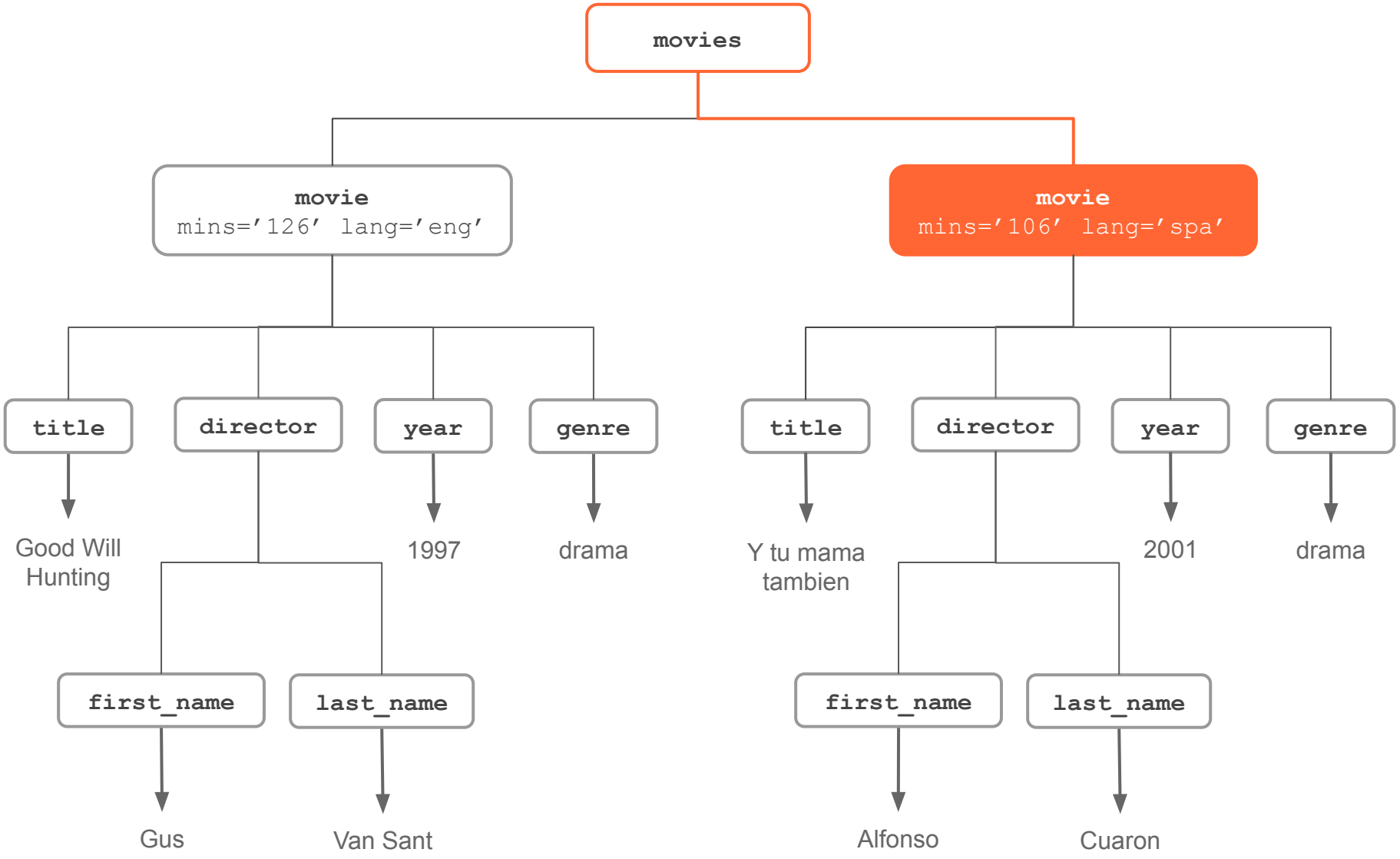
//genre


movies
movie
mins='126' lang='eng'
movie
mins='106' lang='spa'
title
director
year
genre
title
director
year
genre
Good Will Hunting
1997
drama
Y tu mama tambien
2001
drama
first_name
last_name
first_name
last_name
Gus
Van Sant
Alfonso
Cuaron

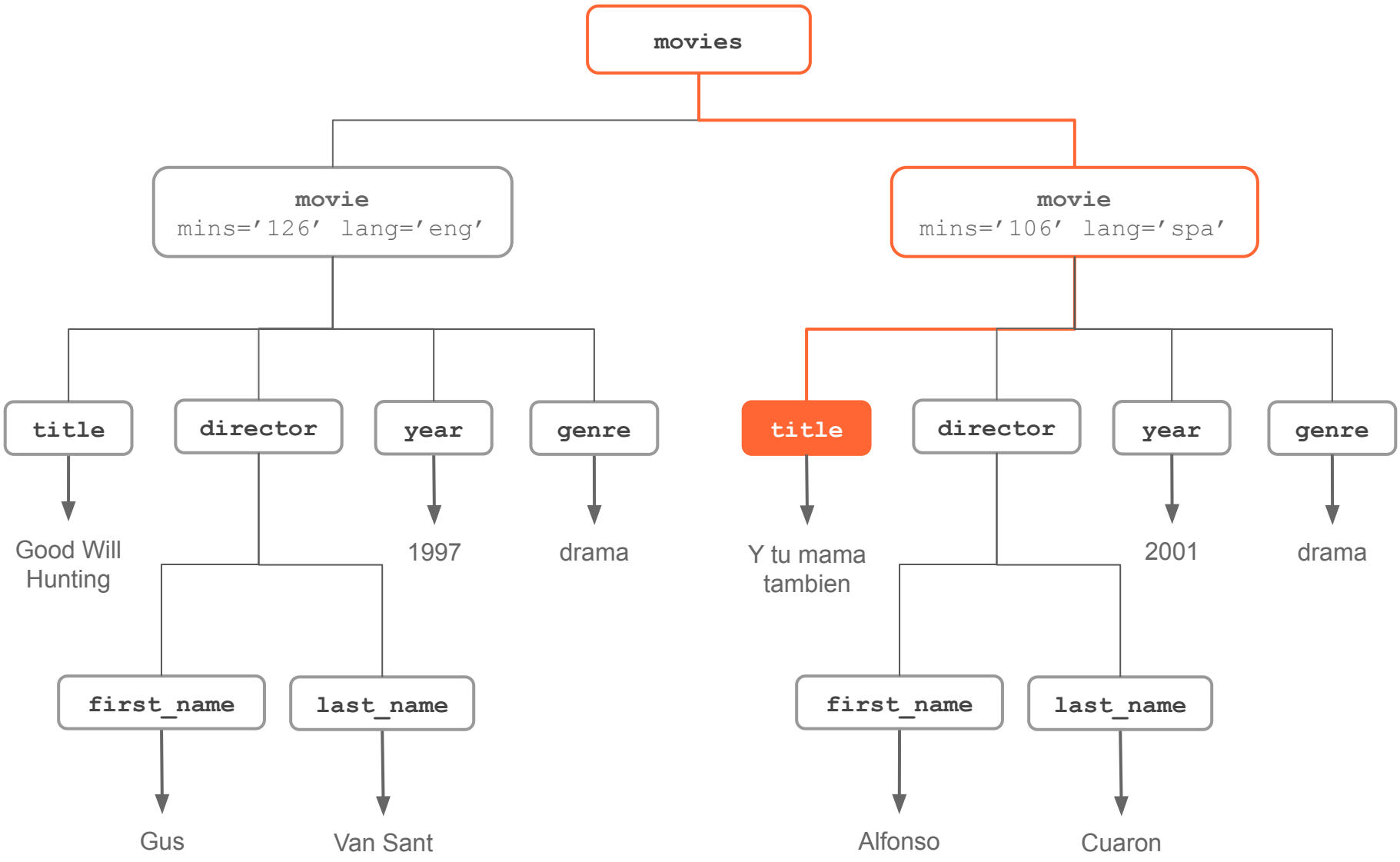
/movies/movie[@lang]



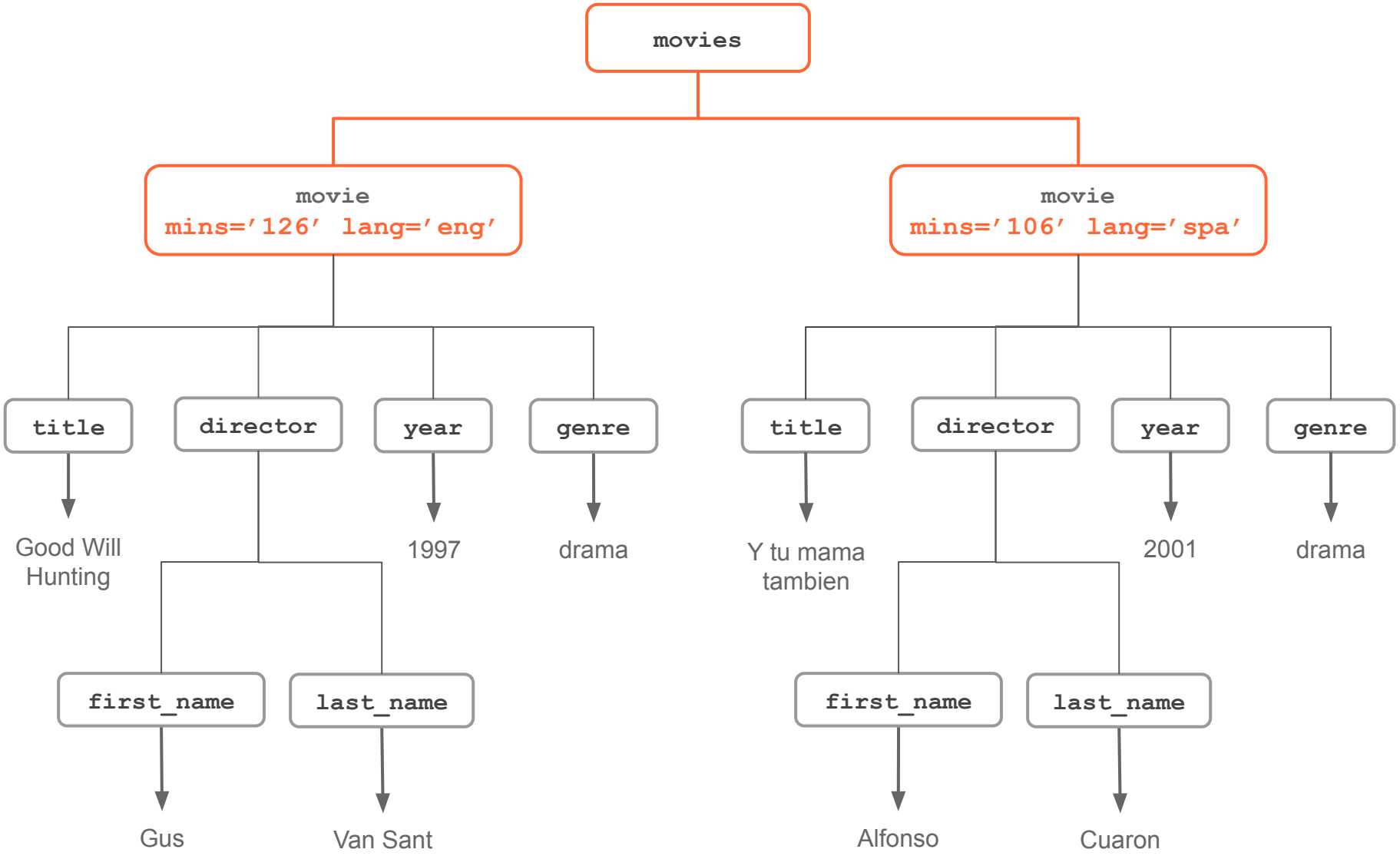
/movies/movie[@lang='spa']



/movies/movie[@lang='spa']/title



/movies/movie/@*



```
# movie children (from root node)
xml_find_all(doc, "/movies/movie")
{xml_nodeset (2)}
[1] <movie mins="126" lang="eng">\n  <title>Good Will
Hunting ...
[2] <movie mins="106" lang="spa">\n  <title>Y tu mama
tambien ...


# title children (from root node)
xml_find_all(doc, "/movies/movie/title")
{xml_nodeset (2)}
[1] <title>Good Will Hunting</title>
[2] <title>Y tu mama tambien</title>


# first_name children (from root node)
xml_text(xml_find_all(doc, "/movies/movie/title"))
[1] "Good Will Hunting" "Y tu mama tambien"
```

```
# director children (from any movie element)
xml_find_all(doc, "//movie/director")
{xml_nodeset (2)}
[1] <director>\n  <first_name>Gus</first_name>\n
<last_name> ...
[2] <director>\n  <first_name>Alfonso</first_name>\n
<last_n …

xml_text(xml_find_all(doc, "//movie/director"))
[1] "GusVan Sant"   "AlfonsoCuaron"


# first_name children (from root node)
xml_find_all(doc, "/movies/movie/director/first_name")
{xml_nodeset (2)}
[1] <first_name>Gus</first_name>
[2] <first_name>Alfonso</first_name>

xml_text(
  xml_find_all(doc, "/movies/movie/director/first_name"))
[1] "Gus"     "Alfonso"
```

```
# last_name (from anywhere in the tree)
xml_find_all(doc, "//last_name")
{xml_nodeset (2)}
[1] <last_name>Van Sant</last_name>
[2] <last_name>Cuaron</last_name>

# text of last_name (from anywhere in the tree)
xml_text(xml_find_all(doc, "//last_name"))
[1] "Van Sant" "Cuaron"


# title of movie with attribute lang='spa'
xml_find_all(doc, "/movies/movie[@lang='spa']/title")
{xml_nodeset (1)}
[1] <title>Y tu mama tambien</title>

xml_text(
  xml_find_all(doc, "/movies/movie[@lang='spa']/title"))
[1] "Y tu mama tambien"
```