目录

[Odoo Domain(过滤规则)使用教程 2](#__RefHeading___Toc373_1888487417)

[为什么要用Domain？ 2](#__RefHeading___Toc375_1888487417)

[什么是Domain？ 2](#__RefHeading___Toc377_1888487417)

[实例 5](#__RefHeading___Toc379_1888487417)

[odoo开发domain的使用 8](#__RefHeading___Toc381_1888487417)

[Odoo domain写法及运用 9](#__RefHeading___Toc433_4187943547)

[Domain的写法 9](#__RefHeading___Toc435_4187943547)

[字段名 9](#__RefHeading___Toc437_4187943547)

[操作符 9](#__RefHeading___Toc439_4187943547)

[值 10](#__RefHeading___Toc441_4187943547)

[条件间的逻辑前缀 10](#__RefHeading___Toc443_4187943547)

[Domain的使用场景 10](#__RefHeading___Toc445_4187943547)

[1.关联字段 10](#__RefHeading___Toc447_4187943547)

[2.过滤出想要显示的记录 11](#__RefHeading___Toc449_4187943547)

[3.权限过滤 11](#__RefHeading___Toc451_4187943547)

[4.搜索过滤 11](#__RefHeading___Toc453_4187943547)

# **Odoo Domain(过滤规则)使用教程**

## **为什么要用Domain？**

ERP系统里角色众多，权限交错，常见的需求会有不同角色的用户登陆进来看到不一样的的数据或者不同菜单看到不同的记录。在Odoo里面实现各种的“过滤”都是通过**Domain**来实现的。Domain中文叫做“域”，它起到的作用是“过滤规则”。

Odoo里面很多功能都需要用到Domain，操作层面的记录规则、过滤器、动作；代码层面的search方法等都需要用到Domain。

## **什么是Domain？**



Domain是个多条件的列表，每个条件是一个三元表达式：[(字段名，操作符，值), (字段名，操作符，值)]。

**字段名**

当前模型的字段或者是通过点操作符访问当前模型的Many2one / Many2Many对象。如果左边是Many2Many对象的时候，则表示左边只要任意一个值符合条件则符合条件。例如是：你的Domain是用在sale.order上的，然后你想根据客户的国家编码来进行筛选可以这样写partner\_id.country\_id.code。

**操作符**

=,!=,>,>=,<,<=,

这些就是我们平常用的“等于”，“不等于”，“大于”，“大于等于”，“小于”，“小于等于“。

**=**?

未设置或者等于，未设置就是当值是None或者是False，其余和“=”一样。

=like

可以使用模式匹配：下划线“\_”匹配一个字符，百分号“%”匹配零或者多个字符。 这里默认的匹配模式是：value（不加其他通配符）。

like

通过%value%匹配。 常用于模糊搜索（例如：搜索名字包含“123”的记录）

not like

通过%value%不匹配的。

ilike

类似like，忽略大小写。

=ilike

类似=like，忽略大小写。

not ilike

类似not like，忽略大小写。

in

判断value是否在元素的列表里面。

not in

判断value是否不再元素的列表里面。

child\_of

判断是否value的子记录，通过\_parent\_name实现。

**值**

进行筛选的变量，必须能通过操作符来和字段进行比较。结合上面的例子，如果我们想过滤中国客户的订单可以这样[('partner\_id.country\_id.code', '=', 'CN')]。

**条件间的逻辑**



看完每个条件里面的写法，那么现在来看看各个条件之间的组合关系。Odoo是使用了[波兰表示法](https://link.zhihu.com/?target=https%3A//zh.wikipedia.org/wiki/%25E6%25B3%25A2%25E5%2585%25B0%25E8%25A1%25A8%25E7%25A4%25BA%25E6%25B3%2595)，简单来说就是一种无需括号的表达方法，可以看看小节图里就是波兰表示法表示3+4。

条件间使用的逻辑前缀：

**&** ：逻辑 AND，条件间的默认关系。2个参数（后2个条件或者条件组合） ，例如:

['&', ('partner\_id.coutnry\_id.code', '=', 'CN'), ('partner\_id.coutry\_id.code', '=', 'US')]

这里的 & 就是把后面的2个条件通过AND组合起来也就是 A AND B，但是注意到这里还有“条件组合”的情况，所以还有可能是

['&', ('partner\_id.coutnry\_id.code', '=', 'CN'),

'&',('partner\_id.coutry\_id.code', '=', 'US'), ('partner\_id.coutry\_id.code', '=', 'GB')]

转换为一般的表示方法则是 A AND (B AND C)，但是因为'&'是默认的逻辑关系，所以我们其实可以不用显式表示

[('partner\_id.coutnry\_id.code', '=', 'CN'),('partner\_id.coutry\_id.code', '=', 'US'),

('partner\_id.coutry\_id.code', '=', 'GB')]

**|**：逻辑 OR，2个参数。   
**!**：逻辑 NOT，1个参数。

## **实例**

“名字为 ABC”，就是一个最简单的单条件Domain。

[('name','=','ABC')]

“名字为 ABC 而且 语言编码不为en\_US”，Domain里条件默认逻辑关系就是and，所以如下。

[('name','=','ABC'),

('language.code','!=','en\_US')]

“名字为 ABC 而且语言编码不为 en\_US 而且国家的编码为 be 或者 de”。or的逻辑关系是用"|"表示。

[('name','=','ABC'),

('language.code','!=','en\_US'),

'|',('country\_id.code','=','be'),

('country\_id.code','=','de')]

当然还有简单一点的写法

[('name','=','ABC'),

('language.code','!=','en\_US'),

('country\_id.code','in', ['be', 'de'])]

条件里面丰富的操作符可以方便地让我们转换条件，例如 = -> !=, 1< -> >=。注意如果需要在xml使用Domain，大于，小于因为和标签符号一样所以需要转义大于=>&gt;，小于=>&lt;

如果我们要做到这个效果

A and (B or C) and D and E

先从里面开始，把or提前

A and (or B C) and D and E

把里面的and提前，去掉括号

and A or B C and D E

所以最后的domain可以这样写

A, '|', B, C, D, E

当然了，我们为什么不写得让自己也容易看一点呢，如下：

A, D, E, '|', B, C

通过Domain我们可以管理我们的权限，定制我们需要的过滤器，让Odoo成为更贴合我们需求的ERP系统。

[编辑于 2016-10-23](http://zhuanlan.zhihu.com/p/22991458)

# odoo开发domain的使用

2016年05月05日

odoo开发中，domain的使用可以在view中，可以在model中，因此说说自己用到的一些地方。

首先domain得使用方法就不说了，可以参考http://www.chinamaker.net/html/2014/study\_1211/183.html，写的还是挺详细的，说说在view中的使用。

1、view中字段关联的过滤

在view视图中，字段之间的关联使用，比如说在地址信息中，有省、市、县信息，因此在选择省的时候就在市中列出该省的市来，那么首先字段中要设置市的字段，比如：

city 的model中要设置和省得多对一的字段

state\_id=fields.Man2one('state',u'省名称',required=True,select=True)

那么在odoo的view视图中要设置省字段，市字段，设置domain过滤，可以参考http://www.chinamaker.net/html/2015/study\_0929/219.html

<field name="city\_id" placeholder="城区" domain="[('state\_id','=',state\_id)]"/>

2、过滤只显示需要显示的数据

在view视图中，哟时候需要根据不同的权限显示不同的数据，因此需要对数据进行过滤，我看到资料可以根据用户组来显示，但是没有用到过，没有尝试，可以参考这篇博客http://blog.sina.com.cn/s/blog\_7cb52fa80102v4z0.html，

比如我遇到一个需求，只显示某一字段非非空的数据，在act-window视图中添加过滤条件

<field name="domain" >[('home','!=',False)]</field>

就可以显示该字段为非空的情况。

（注意：在实践中发现如果设置了该过滤条件，那么再更改过滤条件升级模块会发现显示的数据还是之前过滤条件，怎么解决这个bug呢？需要把该菜单隐藏后升级，然后再放出菜单再升级才会实现过滤条件的更改，当时弄了半天才发现这个bug）

# [**Odoo domain写法及运用**](https://segmentfault.com/a/1190000014758701)

2018-05-07

# **Domain的写法**

**Domain是一个元组，由3个元素组成 (字段名, 操作符， 值)**

## **字段名**

如果是Many2one/Many2Many/One2many字段，可以通过'.'取出对应的属性，如：

('routing\_id.operation\_ids.workcenter\_id','=', active\_id)

## **操作符**

| 符号 | 说明 |
| --- | --- |
| =,>.<,>=,<=,!= | 比较运算，等于，不等于，大于，大于等于，小于，小于等于 |
| like | 模糊匹配，通过%value%匹配 |
| =like | 可以使用模式匹配，下划线-匹配一个字符，百分号%匹配零或者多个字符 |
| ilike | 类似like，但是忽略大小写 |
| =ilike | 类似=like，但是忽略大小写 |
| not like | 通过%value%不匹配的 |
| not ilike | 类似not like，但是忽略大小写 |
| =? | 未设置或者等于，未设置表示当值是None或者是False，其余和=一样 |
| in | 判断value是否在元素的列表里面 |
| not in | 判断value是否不再元素的列表里面 |
| child\_of | 判断是否value的子记录 |

示例：

('company\_id','child\_of',[user.company\_id.id])

('company\_level','not in',('common\_company'))

有一个要注意的地方,'in'可以反向表示，比如：

('user\_ids', 'in', user.id)

应用场景是当一个model中的字段是user\_ids = field.Many2many('res.users')，我们可以将这个model的权限赋给这些用户，只有这些用户可以访问

<record id="mrp\_dashboard\_rule" model="ir.rule">

<field name="name">mrp\_dashboard\_rule</field>

<field name="model\_id" **ref**="model\_mrp\_workcenter"/>

<field name="global" eval="True"/>

<field name="domain\_force">[('user\_ids', '**in**', user.id)]</field>

</record>

## **值**

包括字符串，True或False，数字等  
在xml中，能取出当前登录用户user对象，你可以取出user.id,user.company\_id.id等

## **条件间的逻辑前缀**

| 符号 | 说明 |
| --- | --- |
| & | 'AND'或者'与',表示后面2个条件同时满足。默认关系，可以省略 |
| | | 'OR'或者'或'，表示后面2个条件满足一个就行 |
| ！ | 'NOT'或者'非'，将后面一个条件反转，如果是True，就变成False |

示例：

[('name', '=', 'Allen'), '|', ('language.code', '!=', 'en\_US'), ('company\_id', '=', user.company\_id.id)]

# **Domain的使用场景**

## **1.关联字段**

(Many2one/One2many/Many2many)过滤出不同的选择结果

default\_debit\_account\_id = fields.Many2one('account.account', string='Default Debit Account', domain=[('deprecated', '=', False)])

如上：只有当account.account中字段deprecated=False的记录才能被选择显示

## **2.过滤出想要显示的记录**

在ir.actions.act\_windon中添加Domain，当某个地方调用这个动作时，只显示符合domain条件的记录

<record id="relate\_partner\_opportunities" model="ir.actions.act\_window">

<field name="name">Opportunities</field>

<field name="res\_model">crm.lead</field>

<field name="view\_mode">kanban,tree,form,graph,calendar</field>

<field name="domain">[('type','=','opportunity')]</field>

<field name="context">{

'search\_default\_partner\_id': active\_id,

'default\_type': 'opportunity'

}</field>

<field name="view\_id" eval="False"/>

<field name="search\_view\_id" ref="crm.view\_crm\_case\_opportunities\_filter"/>

</record>

## **3.权限过滤**

<record id="account\_move\_comp\_rule" model="ir.rule">

<field name="name">Account Entry</field>

<field name="model\_id" ref="model\_account\_move"/>

<field name="global" eval="True"/>

<field name="domain\_force">['|',('company\_id','=',False),('company\_id','child\_of',[user.company\_id.id])]</field>

</record>

## **4.搜索过滤**

<record id="view\_crossovered\_budget\_search" model="ir.ui.view">

<field name="name">crossovered.budget.search</field>

<field name="model">crossovered.budget</field>

<field name="arch" type="xml">

<search string="Budget">

<field name="name" filter\_domain="[('name','ilike',self)]" string="Budget"/>

<field name="date\_from"/>

<field name="date\_to"/>

<filter string="Draft" domain="[('state','=','draft')]" help="Draft Budgets"/>

<filter string="To Approve" domain="[('state','=','confirm')]" help="To Approve Budgets" />

<field name="state"/>

</search>

</field>

</record>

Domain的使用场景还有许多，我就不一一列举了。  
Domain的介绍就到此结束了，有什么问题，请大家多多指教。