

一、查詢:

1. 查詢**單筆**資料，若查詢出多筆資料程式會發生錯誤，其語法如下：

```
資料表.objects.get(欄位 1="條件",欄位 2="條件",....)
```

2. 查詢**多筆**資料，其語法如下：

```
資料表.objects.filter(欄位 1="條件",欄位 2="條件",....)
```

符合條件的資料

```
資料表.objects.exclude(欄位 1="條件",欄位 2="條件",....)
```

不符合條件的資料

3. 查詢**所有**資料，其語法如下：

```
資料表.objects.all()
```

4. 對查詢結果進行**排序**，其語法如下：

```
資料表.objects.all().order_by("欄位")
```

遞增排序

```
資料表.objects.filter(...).order_by("-欄位")
```

遞減排序

5. 查詢時，可於欄位後方加上「__關鍵字」（雙底線+關鍵字），以加強查詢的精準度，如下表：

語法	功能說明	範例
<code>__contains</code>	包含搜尋	姓名 <code>__contains="林"</code> (類似萬用字元功能： <code>*林*</code>)
<code>__icontains</code>	包含搜尋(不分大小寫)	姓名 <code>__icontains="林"</code>
<code>__exact</code>	精確搜尋	姓名 <code>__exact="蔡依林"</code>
<code>__iexact</code>	精確搜尋(不分大小寫)	姓名 <code>__iexact="蔡依林"</code>
<code>__gt</code>	大於: >	生日 <code>__gt=datetime.date(2022,1,28)</code>
<code>__gte</code>	大於等於: >=	生日 <code>__gte=datetime.date(2022,1,28)</code>
<code>__lt</code>	小於: <	生日 <code>__lt=datetime.date(2022,1,28)</code>
<code>__lte</code>	小於等於: <=	生日 <code>__lte=datetime.date(2022,1,28)</code>
<code>__startswith</code>	以...開頭	姓名 <code>__startswith="林"</code> (類似萬用字元功能： <code>林*</code>)
<code>__istartswith</code>	以...開頭 忽略大小寫	姓名 <code>__istartswith="林"</code>
<code>__endswith</code>	以...結尾	姓名 <code>__endswith="林"</code> (類似萬用字元功能： <code>*林</code>)
<code>__iendswith</code>	以...結尾，忽略大小寫	姓名 <code>__iendswith="林"</code>
<code>__range</code>	在...範圍內	生日 <code>__range=(datetime.date(2022,1,25),datetime.date(2022,1,31))</code>
<code>__year</code>	日期欄位的年份	生日 <code>__year=2022</code> 生日 <code>__year__gt=2022</code>
<code>__month</code>	日期欄位的月份	生日 <code>__month=5</code> 生日 <code>__month__gt=5</code>
<code>__day</code>	日期欄位的日	生日 <code>__day=20</code> 生日 <code>__day__gt=20</code>

範例一：查詢單筆資料

myweb\view\p1.py

```
from django.shortcuts import redirect, render
from db.models import student

def selectOne(request):
    data = student.objects.get(學號="1091301")
    return render(request, "class.html", locals())
```

myweb\myweb\urls.py

```
from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('class/', selectOne),
]
```

myweb\template\class.html

```
{% extends 'base.html' %}
{% block mainbody %}
<table class="table table-striped table-hover w-50">
  <thead>
    <tr>
      <th>學號</th>
      <th>姓名</th>
      <th>性別</th>
      <th>生日</th>
      <th>信箱</th>
      <th>自傳</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>{{data.學號}}</td>
      <td>{{data.姓名}}</td>
      <td>{{data.性別}}</td>
      <td>{{data.生日}}</td>
      <td>{{data.信箱}}</td>
      <td>{{data.自傳}}</td>
    </tr>
  </tbody>
</table>
{% endblock mainbody %}
```

Navbar Home Link Dropdown ▾

Search

Search

學號	姓名	性別	生日	信箱	自傳
1091301	王劭謙	男	2022年1月23日	1091301@mail.slhs.tp.edu.tw	測試

範例二：查詢所有資料**myweb\view\p1.py**

```
from django.shortcuts import redirect, render
from db.models import student

def selectAll(request):
    data = student.objects.all().order_by("學號")
    return render(request, "class.html", locals())
```

myweb\myweb\urls.py

```
from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('class/', selectAll),
]
```

myweb\template\class.html

```
{% extends 'base.html' %}

{% block mainbody %}
<table class="table table-striped table-hover w-50 ">
  <thead>
    <tr>
      <th>學號</th>
      <th>姓名</th>
      <th>性別</th>
      <th>生日</th>
      <th>信箱</th>
      <th>自傳</th>
    </tr>
  </thead>
  <tbody>
    {% for i in data %}
      <tr>
        <td>{{i.學號}}</td>
        <td>{{i.姓名}}</td>
        <td>{{i.性別}}</td>
        <td>{{i.生日}}</td>
        <td>{{i.信箱}}</td>
        <td>{{i.自傳}}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
{% endblock mainbody %}
```

學號	姓名	性別	生日	信箱	自傳
1091301	王劭謙	男	2022年1月23日	1091301@mail.slhs.tp.edu.tw	測試
1091302	王麒竣	男	2022年1月24日	1091302@mail.slhs.tp.edu.tw	測試
1091303	林子靖	男	2022年1月25日	1091303@mail.slhs.tp.edu.tw	測試
1091304	林佑亦	男	2022年1月26日	1091304@mail.slhs.tp.edu.tw	測試
1091305	張庭璋	男	2022年1月27日	1091305@mail.slhs.tp.edu.tw	測試
1091306	曾浚翔	男	2022年1月28日	1091306@mail.slhs.tp.edu.tw	測試
1091307	黃喧智	男	2022年1月29日	1091307@mail.slhs.tp.edu.tw	測試
1091308	裘承軒	男	2022年1月30日	1091308@mail.slhs.tp.edu.tw	測試
1091309	蔡使勳	男	2022年1月31日	1091309@mail.slhs.tp.edu.tw	測試
1091310	方品雯	女	2022年1月23日	1091310@mail.slhs.tp.edu.tw	測試
1091311	古歆翎	女	2022年1月24日	1091311@mail.slhs.tp.edu.tw	測試
1091312	李宜蕓	女	2022年1月25日	1091312@mail.slhs.tp.edu.tw	測試
1091313	李嫻瑢	女	2022年1月26日	1091313@mail.slhs.tp.edu.tw	測試
1091314	周芷璇	女	2022年1月27日	1091314@mail.slhs.tp.edu.tw	測試
1091315	林昕瑩	女	2022年1月28日	1091315@mail.slhs.tp.edu.tw	測試
1091316	姜采綺	女	2022年1月29日	1091316@mail.slhs.tp.edu.tw	測試
1091317	徐書彤	女	2022年1月30日	1091317@mail.slhs.tp.edu.tw	測試
1091318	莊凡蕓	女	2022年1月31日	1091318@mail.slhs.tp.edu.tw	測試
1091319	郭甄庭	女	2022年1月23日	1091319@mail.slhs.tp.edu.tw	測試
1091320	陳詣涵	女	2022年1月24日	1091320@mail.slhs.tp.edu.tw	測試
1091321	蔡忻婷	女	2022年1月25日	1091321@mail.slhs.tp.edu.tw	測試
1091322	譚筠馨	女	2022年1月26日	1091322@mail.slhs.tp.edu.tw	測試

範例三：查詢多筆資料(生日在 28 號之後且為男生，以生日遞減排序)

myweb\view\p1.py

```
from django.shortcuts import redirect, render
from db.models import student

def selectAll(request):
    data = student.objects.filter(生日__day__gt=28,性別="男").order_by("-生日")
    return render(request, "class.html", locals())
```

myweb\myweb\urls.py

```
from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('class/',selectfilter),
]
```

myweb\template\class.html

```
{% extends 'base.html' %}

{% block mainbody %}
<table class="table table-striped table-hover w-50 ">
  <thead>
    <tr>
      <th>學號</th>
      <th>姓名</th>
      <th>性別</th>
      <th>生日</th>
      <th>信箱</th>
      <th>自傳</th>
    </tr>
  </thead>
  <tbody>
    {% for i in data %}
    <tr>
      <td>{{i.學號}}</td>
      <td>{{i.姓名}}</td>
      <td>{{i.性別}}</td>
      <td>{{i.生日}}</td>
      <td>{{i.信箱}}</td>
      <td>{{i.自傳}}</td>
    </tr>
    {% endfor %}
  </tbody>
</table>
{% endblock mainbody %}
```

學號	姓名	性別	生日	信箱	自傳
1091309	蔡使勳	男	2022年1月31日	1091309@mail.slhs.tp.edu.tw	測試
1091308	蔡承軒	男	2022年1月30日	1091308@mail.slhs.tp.edu.tw	測試
1091307	黃曜智	男	2022年1月29日	1091307@mail.slhs.tp.edu.tw	測試

二、新增:

1. 利用 save() 方法新增資料，其語法如下：

```
變數名稱 = 資料表.objects.create(欄位 1=值, 欄位 2=值, 欄位 3=值,....)
變數名稱.save()
```

範例**myweb\view\p1.py**

```
def add(request):
    message = "新增成功!"
    try:
        data = student.objects.create(學號="1091323",姓名="王小美",性別="女",生日="2000-3-5",信箱
        ="1091323@mail.slhs.tp.edu.tw",自傳="Hello!!")
        data.save()
    except:
        message = "已有相同的學號"
    return render(request, "result.html", locals())
```

myweb\myweb\urls.py

```

from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('result/',add),
]

```

myweb\template\result.html

```

{% extends 'base.html' %}

{% block mainbody %}
<h1>{{message}}</h1>
{% endblock mainbody %}

```

<input type="checkbox"/>	1091318	莊凡堯	女	2022年1月31日	1091318@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091319	郭甄庭	女	2022年1月23日	1091319@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091320	陳詣涵	女	2022年1月24日	1091320@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091321	蔡忻妤	女	2022年1月25日	1091321@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091322	譚筠馨	女	2022年1月26日	1091322@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091323	王小美	女	2000年3月5日	1091323@mail.slhs.tp.edu.tw	Hello!!	2022年3月14日 18:37	2022年3月14日 18:37

三、修改:

1. 首先查詢欲修改之資料，再利用 update() 方法更新修改結果，其語法如下：

```

變數名稱 = 資料表.objects.filter 或 exclude 或 get(欄位 1="條件",欄位 2="條件",...)
變數名稱.update(欄位 1="修改值",欄位 2="修改值",...)

```

範例**myweb\view\p1.py**

```

def edit(request):
    message = "更新成功!"
    try:
        data = student.objects.filter(性別="男",生日__day__lte=26)
        data.update(自傳="未來目標為台科大資訊管理系!")
    except:
        message = "找不該資料"
    return render(request, "result.html", locals())

```

myweb\myweb\urls.py

```

from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('result/',edit),
]

```

myweb\template\result.html

```
{% extends 'base.html' %}

{% block mainbody %}
<h1>{{message}}</h1>
{% endblock mainbody %}
```

<input type="checkbox"/>	學號	1 ▲ 姓名	2 ▲ 性別	生日	信箱	自傳	註冊時間	最後更新
<input type="checkbox"/>	1091301	王劭謙	男	2022年1月23日	1091301@mail.slhs.tp.edu.tw	未來目標為台科大資訊管理系!	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091302	王麒竣	男	2022年1月24日	1091302@mail.slhs.tp.edu.tw	未來目標為台科大資訊管理系!	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091303	林于靖	男	2022年1月25日	1091303@mail.slhs.tp.edu.tw	未來目標為台科大資訊管理系!	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091304	林佑亦	男	2022年1月26日	1091304@mail.slhs.tp.edu.tw	未來目標為台科大資訊管理系!	2022年2月21日 19:53	2022年2月21日 19:53
<input type="checkbox"/>	1091305	張庭璋	男	2022年1月27日	1091305@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年2月21日 19:53

四、刪除:

1. 首先查詢欲刪除之資料，再利用 delete() 方法儲存刪除結果，其語法如下：

```
變數名稱 = 資料表.objects.filter 或 exclude 或 get(欄位 1="條件", 欄位 2="條件")
變數名稱.delete()
```

範例**myweb\view\p1.py**

```
def delete(request):
    message = "刪除成功!"
    try:
        data = student.objects.filter(姓名__startswith="林", 生日__day__lte=26)
        data.delete()
    except:
        message = "找不該資料"
    return render(request, "result.html", locals())
```

myweb\myweb\urls.py

```
from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('result/', delete),
]
```

myweb\template\result.html

```
{% extends 'base.html' %}

{% block mainbody %}
<h1>{{message}}</h1>
{% endblock mainbody %}
```

ID	姓名	性別	生日	電子郵件	狀態	最後登入時間	下次登入時間
<input type="checkbox"/> 1091302	王麒竣	男	2022年1月24日	1091302@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年3月16日 19:45
<input type="checkbox"/> 1091303	林于晴	男	2022年1月25日	1091303@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年3月16日 19:45
<input type="checkbox"/> 1091304	林佑亦	男	2022年1月26日	1091304@mail.slhs.tp.edu.tw	測試	2022年2月21日 19:53	2022年3月16日 19:45

五、表單

在網頁當中，經常會使用表單的方式來傳遞資料，其傳遞的方式分兩種：(1) POST 與 (2) GET。由於 GET 傳遞資料時，會將傳遞資料顯示於網址(URL)當中，因此若注重資料安全性則必須使用 POST；反之可使用 GET。

方法	網址(傳遞資料時)	安全性	CSRF 防護語法
GET	www.slhs.tp.edu.tw/dp/?user=alex&pwd=1234	低	不需要
POST	www.slhs.tp.edu.tw/dp	高	{% csrf_token %}

在之前的範例中，view 中的 py 檔中，所有函數的第一個參數都必須是 HttpRequest 資料型別，預設變數名稱為 request，該變數名稱便是儲存 POST 及 GET 所傳遞的資料。

1. request 常用的方法如下：

方法	用途	語法
GET	取得以 GET 方式傳送的表單欄位內容。	request.GET["欄位名稱"]
POST	取得以 POST 方式傳送的表單欄位內容。	request.POST["欄位名稱"]
method	取得表單傳送方式。	if request.method=="POST":

範例一：利用 POST 傳遞資料 (POST.html => ans.html)

```
myweb\view\p1.py
from django.shortcuts import redirect, render

def POST(request):
    if request.method == "POST":
        sid = request.POST["sid"]
        name = request.POST["name"]
        sex = request.POST["sex"]
        birth = request.POST["birth"]
        email = request.POST["email"]
        auto = request.POST["auto"]
        return render(request, "ans.html", locals())
    else:
        sid = name = sex = birth = email = auto = ""
        return render(request, "POST.html", locals())
```

```
myweb\myweb\urls.py
from django.contrib import admin
from django.urls import path
from view.p1 import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('POST/',POST),
    path('POST/ans/',POST),
]
```


myweb\template\POST.html

```
{% extends 'base.html' %}

{% block mainbody %}
<form action="ans/" method="POST" name="form1">
  {% csrf token %}
  <table class="table table-striped table-hover w-50 ">
    <tr>
      <th>學號</th>
      <td><input type="text" name="sid"></td>
    </tr>
    <tr>
      <th>姓名</th>
      <td><input type="text" name="name"></td>
    </tr>
    <tr>
      <th>性別</th>
      <td>
        <input type="radio" name="sex" value="男" checked> 男
        <input type="radio" name="sex" value="女"> 女
      </td>
    </tr>
    <tr>
      <th>生日</th>
      <td><input type="date" name="birth"></td>
    </tr>
    <tr>
      <th>信箱</th>
      <td><input type="email" name="email"></td>
    </tr>
    <tr>
      <th>自傳</th>
      <td><textarea rows="5" cols="50" name="auto"></textarea></td>
    </tr>
  </table>
  <input type="submit" vlaue="送出">
  <input type="reset" vlaue="清空">
</form>
{% endblock mainbody %}
```

myweb\template\ans.html

```
{% extends 'base.html' %}

{% block mainbody %}
<ol>
  <li>{{sid}}</li>
  <li>{{name}}</li>
  <li>{{sex}}</li>
  <li>{{birth}}</li>
  <li>{{email}}</li>
  <li>{{auto}}</li>
</ol>
{% endblock mainbody %}
```

127.0.0.1:8000/POST/

Navbar Home Link Dropdown ▾

學號

姓名

性別 男 女

生日

信箱

自傳

127.0.0.1:8000/POST/ans/

Navbar Home Link Dropdown ▾

1. 1081401
2. Mary
3. 女
4. 2022-03-11
5. mary@gmail.com
6. I'm Mary.

範例二：利用 GET 傳遞資料 (GET.html => ans.html)

myweb\view\p1.py

```
from django.shortcuts import redirect, render
```

```
def GET(request):
```

```
    if "sid" in request.GET and request.GET["sid"] != "":
```

```
        sid = request.GET["sid"]
```

```
        name = request.GET["name"]
```

```
        sex = request.GET["sex"]
```

```
        birth = request.GET["birth"]
```

```
        email = request.GET["email"]
```

```
        auto = request.GET["auto"]
```

```
        return render(request, "ans.html", locals())
```

```
    else:
```

```
        sid = name = sex = birth = email = auto = ""
```

```
        return render(request, "GET.html", locals())
```

myweb\myweb\urls.py

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from view.p1 import *
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    path('GET/', GET),
```

```
    path('GET/ans/', GET),
```

```
]
```

myweb\template\GET.html

```
{% extends 'base.html' %}

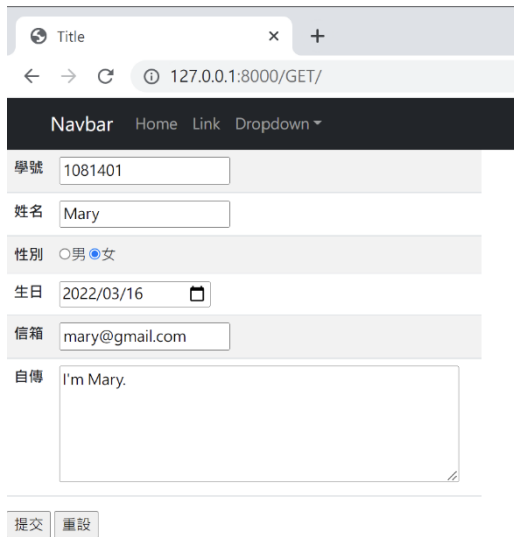
{% block mainbody %}
<form action="ans/" method="GET" name="form1">
  <table class="table table-striped table-hover w-50">
    <tr>
      <th>學號</th>
      <td><input type="text" name="sid"></td>
    </tr>
    <tr>
      <th>姓名</th>
      <td><input type="text" name="name"></td>
    </tr>
    <tr>
      <th>性別</th>
      <td>
        <input type="radio" name="sex" value="男" checked> 男
        <input type="radio" name="sex" value="女"> 女
      </td>
    </tr>
    <tr>
      <th>生日</th>
      <td><input type="date" name="birth"></td>
    </tr>
    <tr>
      <th>信箱</th>
      <td><input type="email" name="email"></td>
    </tr>
    <tr>
      <th>自傳</th>
      <td><textarea rows="5" cols="50" name="auto"></textarea></td>
    </tr>
  </table>
  <input type="submit" value="送出">
  <input type="reset" value="清空">
</form>
{% endblock mainbody %}
```

myweb\template\ans.html

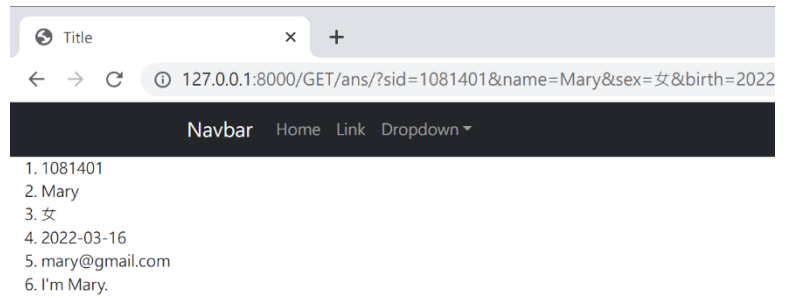
```
{% extends 'base.html' %}

{% block mainbody %}
<ol>
  <li>{{sid}}</li>
  <li>{{name}}</li>
  <li>{{sex}}</li>
  <li>{{birth}}</li>
  <li>{{email}}</li>
  <li>{{auto}}</li>
</ol>

{% endblock mainbody %}
```



A screenshot of a web browser showing a Django form. The browser address bar shows '127.0.0.1:8000/GET/'. The form has a dark blue header with 'Navbar' and links for 'Home', 'Link', and 'Dropdown'. The form fields are: '學號' (ID) with value '1081401', '姓名' (Name) with value 'Mary', '性別' (Gender) with radio buttons for '男' (Male) and '女' (Female), '生日' (Date of Birth) with value '2022/03/16' and a calendar icon, '信箱' (Email) with value 'mary@gmail.com', and '自傳' (Bio) with value 'I'm Mary.'. At the bottom are '提交' (Submit) and '重設' (Reset) buttons.



A screenshot of a web browser showing the rendered HTML output of the form. The browser address bar shows '127.0.0.1:8000/GET/ans/?sid=1081401&name=Mary&sex=女&birth=2022'. The rendered HTML is a list of six items: 1. 1081401, 2. Mary, 3. 女, 4. 2022-03-16, 5. mary@gmail.com, 6. I'm Mary.

六、表單模型化與驗證

為了避免使用者漏填或格式錯誤，因此會設計表單驗證機制，檢查使用者所輸入的資料。而 Django 提供 **form.Form** 類別，將表單模型化，不僅縮短程式碼加快表單設計的速度，並提供驗證功能。