

# Evidencias de Consultas ORM y SQL

---

Proyecto: Chilwe Actividad 5 Módulo 7 Base de datos: PostgreSQL Framework: Django Autor: Ximena Garrido

---

## 1. Recuperación de registros

---

### 1.1 Recuperar todos los libros registrados

Código:

```
Libro.objects.all()
```

```
# Todos los libros
print("\nTodos los libros registrados:")
todos = Libro.objects.all()
mostrar_libros(todos)
```

Explicación:

Obtiene todos los registros almacenados en la tabla Libro.

Evidencia:

```
Todos los libros registrados:
01. 1984 | George Orwell | 328 páginas | Disponible: True
02. Cien Años de Soledad | Gabriel García Marquez | 417 páginas | Disponible: True
03. Cronica de una Muerte Anunciada | Gabriel García Marquez | 122 páginas | Disponible: False
04. El Hobbit | J.R.R. Tolkien | 310 páginas | Disponible: True
05. El Principito | Antoine de Saint Exupery | 96 páginas | Disponible: True
06. Harry Potter y el Caliz de Fuego | J.K. Rowling | 636 páginas | Disponible: False
07. Harry Potter y el Prisionero de Azkaban | J.K. Rowling | 317 páginas | Disponible: True
08. La Comunidad del Anillo | J.R.R. Tolkien | 423 páginas | Disponible: True
09. Rebellion en la Granja | George Orwell | 112 páginas | Disponible: True
10. Vuelo Nocturno | Antoine de Saint Exupery | 192 páginas | Disponible: True
```

---

### 1.2 Recuperar libros cuyo autor sea "Gabriel Garcia Marquez"

Código:

```
Libro.objects.filter(autor__nombre="Gabriel Garcia Marquez")
```

```
# Libros de Gabriel Garcia Marquez
print("\nLibros de Gabriel Garcia Marquez:")
gabo = Libro.objects.filter(autor__nombre="Gabriel Garcia Marquez")
mostrar_libros(gabo)
```

Explicación:

Filtra los libros utilizando la relación ForeignKey hacia Autor.

Evidencia:

```
Libros de Gabriel Garcia Marquez:
01. Cien Años de Soledad | Gabriel Garcia Marquez | 417 páginas | Disponible: True
02. Cronica de una Muerte Anunciada | Gabriel Garcia Marquez | 122 páginas | Disponible: False
```

---

## 1.3 Recuperar libros con más de 200 páginas

Código:

```
Libro.objects.filter(paginas__gt=200)
```

```
# Libros con más de 200 páginas
print("\nLibros con más de 200 páginas:")
mas_200 = Libro.objects.filter(paginas__gt=200)
mostrar_libros(mas_200)
```

Explicación:

Usa un filtro comparativo (\_\_gt) para obtener libros mayores a 200 páginas.

Evidencia:

```
Libros con más de 200 páginas:
01. 1984 | George Orwell | 328 páginas | Disponible: True
02. Cien Años de Soledad | Gabriel Garcia Marquez | 417 páginas | Disponible: True
03. El Hobbit | J.R.R. Tolkien | 310 páginas | Disponible: True
04. Harry Potter y el Caliz de Fuego | J.K. Rowling | 636 páginas | Disponible: False
05. Harry Potter y el Prisionero de Azkaban | J.K. Rowling | 317 páginas | Disponible: True
06. La Comunidad del Anillo | J.R.R. Tolkien | 423 páginas | Disponible: True
```

---

## 2. Filtros y exclusiones

### 2.1 Mostrar solo libros disponibles

Código:

```
Libro.objects.filter(disponible=True)
```

```
# Libros disponibles
print("\nLibros disponibles:")
disponibles = Libro.objects.filter(disponible=True)
mostrar_libros(disponibles)
```

Explicación:

Filtra libros cuyo campo booleano disponible sea True.

Evidencia:

```
Libros disponibles:
01. 1984 | George Orwell | 328 páginas | Disponible: True
02. Cien Años de Soledad | Gabriel Garcia Marquez | 417 páginas | Disponible: True
03. El Hobbit | J.R.R. Tolkien | 310 páginas | Disponible: True
04. El Principito | Antoine de Saint Exupery | 96 páginas | Disponible: True
05. Harry Potter y el Prisionero de Azkaban | J.K. Rowling | 317 páginas | Disponible: True
06. La Comunidad del Anillo | J.R.R. Tolkien | 423 páginas | Disponible: True
07. Rebellion en la Granja | George Orwell | 112 páginas | Disponible: True
08. Vuelo Nocturno | Antoine de Saint Exupery | 192 páginas | Disponible: True
```

---

## 2.2 Excluir libros con menos de 100 páginas

Código:

```
Libro.objects.exclude(paginas__lt=100)
```

```
# Excluir libros con menos de 100 páginas
print("\nLibros con 100 páginas o más:")
mayores_100 = Libro.objects.exclude(paginas__lt=100)
mostrar_libros(mayores_100)
```

Explicación:

Excluye todos los libros cuya cantidad de páginas sea menor a 100.

Evidencia:

```
Libros con 100 páginas o más:  
01. 1984 | George Orwell | 328 páginas | Disponible: True  
02. Cien Años de Soledad | Gabriel García Marquez | 417 páginas | Disponible: True  
03. Cronica de una Muerte Anunciada | Gabriel García Marquez | 122 páginas | Disponible: False  
04. El Hobbit | J.R.R. Tolkien | 310 páginas | Disponible: True  
05. Harry Potter y el Caliz de Fuego | J.K. Rowling | 636 páginas | Disponible: False  
06. Harry Potter y el Prisionero de Azkaban | J.K. Rowling | 317 páginas | Disponible: True  
07. La Comunidad del Anillo | J.R.R. Tolkien | 423 páginas | Disponible: True  
08. Rebelion en la Granja | George Orwell | 112 páginas | Disponible: True  
09. Vuelo Nocturno | Antoine de Saint Exupery | 192 páginas | Disponible: True
```

---

## 3. Consultas SQL personalizadas

---

### 3.1 Consulta SQL con raw() ordenada por título

Código:

```
Libro.objects.raw("SELECT * FROM biblioteca_libro ORDER BY titulo ASC")
```

```
# RAW SQL  
print("\nLibros ordenados por título (SQL RAW):")  
raw_query = Libro.objects.raw(  
    "SELECT * FROM biblioteca_libro ORDER BY titulo ASC"  
)  
  
for i, libro in enumerate(raw_query, start=1):  
    print(f"{i:02d}. {libro.titulo} | {libro.autor.nombre}")
```

Explicación:

Ejecuta una consulta SQL directa utilizando el método raw().

Evidencia:

```
Libros ordenados por título (SQL RAW):  
01. 1984 | George Orwell  
02. Cien Años de Soledad | Gabriel García Marquez  
03. Cronica de una Muerte Anunciada | Gabriel García Marquez  
04. El Hobbit | J.R.R. Tolkien  
05. El Principito | Antoine de Saint Exupery  
06. Harry Potter y el Caliz de Fuego | J.K. Rowling  
07. Harry Potter y el Prisionero de Azkaban | J.K. Rowling  
08. La Comunidad del Anillo | J.R.R. Tolkien  
09. Rebelion en la Granja | George Orwell  
10. Vuelo Nocturno | Antoine de Saint Exupery
```

---

### 3.2 Consulta SQL usando connection.cursor()

Código:

```
with connection.cursor() as cursor:
    cursor.execute("""
        SELECT a.nombre, COUNT(l.id)
        FROM biblioteca_libro l
        JOIN biblioteca_autor a ON l.autor_id = a.id
        GROUP BY a.nombre
        ORDER BY COUNT(l.id) DESC
    """)
    resultados = cursor.fetchall()
```

```
# Cursor SQL
print("\nConteo de libros por autor (SQL directo con cursor):")

with connection.cursor() as cursor:
    cursor.execute("""
        SELECT a.nombre, COUNT(l.id)
        FROM biblioteca_libro l
        JOIN biblioteca_autor a ON l.autor_id = a.id
        GROUP BY a.nombre
        ORDER BY COUNT(l.id) DESC
    """)
    resultados = cursor.fetchall()

for i, fila in enumerate(resultados, start=1):
    print(f"{i:02d}. {fila[0]:30} | Total libros: {fila[1]}")
```

Explicación:

Ejecuta una consulta SQL manual utilizando JOIN y GROUP BY para contar libros por autor.

Evidencia:

```
Conteo de libros por autor (SQL directo con cursor):
01. Antoine de Saint Exupery      | Total libros: 2
02. George Orwell                 | Total libros: 2
03. J.K. Rowling                  | Total libros: 2
04. J.R.R. Tolkien                 | Total libros: 2
05. Gabriel Garcia Marquez        | Total libros: 2
```

---

## 4. Campos específicos y anotaciones

---

### 4.1 Recuperar solo los títulos

Código:

```
Libro.objects.values("titulo")
```

```
# Solo títulos
print("\nSolo títulos (values):")
titulos = Libro.objects.values("titulo")

for i, titulo in enumerate(titulos, start=1):
    print(f"{i:02d}. {titulo['titulo']}")
```

Explicación:

Obtiene únicamente el campo titulo usando values().

Evidencia:

```
Solo títulos (values):
01. 1984
02. Cien Años de Soledad
03. Cronica de una Muerte Anunciada
04. El Hobbit
05. El Principito
06. Harry Potter y el Caliz de Fuego
07. Harry Potter y el Prisionero de Azkaban
08. La Comunidad del Anillo
09. Rebelion en la Granja
10. Vuelo Nocturno
```

---

## 4.2 Conteo de libros por autor usando annotate()

Código:

```
Libro.objects.values("autor__nombre").annotate(total=Count("id"))
```

```
# Conteo por autor con annotate
print("\nCantidad de libros por autor (ORM annotate):")

conteo = Libro.objects.values("autor__nombre").annotate(total=Count("id"))

for i, item in enumerate(conteo, start=1):
    print(f"{i:02d}. {item['autor__nombre']:30} | Total libros: {item['total']}")
```

Explicación:

Agrupar los libros por autor y calcular el total utilizando annotate().

Evidencia:

```
Cantidad de libros por autor (ORM annotate):
01. Antoine de Saint Exupery      | Total libros: 2
02. George Orwell                 | Total libros: 2
03. J.K. Rowling                  | Total libros: 2
04. J.R.R. Tolkien                | Total libros: 2
05. Gabriel Garcia Marquez       | Total libros: 2
```

