

# Django Inicial

Clase 08

**django**

**CONQUER BLOCKS**

# Consultas entre modelos

# Consultas entre modelos

ForeignKey

Consultas inversas

# Consultas entre modelos

ManyToManyField

# Consultas entre modelos

OneToOneField

# **Agregaciones y anotaciones**

# Aggregaciones y anotaciones

count

avg

sum

min

max

# Aggregaciones y anotaciones

annotate



# **Manager y modelos personalizados**

# Managers

get\_queryset

# Consultas complejas

# Consultas complejas

## **Q objects**

Los objetos Q en Django permiten realizar consultas complejas que involucren operadores lógicos como AND y OR

# Consultas complejas

OR

# Consultas complejas

AND

# Optimización

# Optimización

## **`select_related`**

es una función en Django que se utiliza para optimizar consultas SQL cuando se accede a datos relacionados en una relación de tipo ForeignKey o OneToOneField. Específicamente, `select_related` realiza una única consulta SQL y “carga previa” (eager loading) los objetos relacionados, lo que reduce la cantidad de consultas realizadas a la base de datos.



# Optimización

## Beneficios

- Rendimiento Mejorado: Reduce significativamente el número de consultas realizadas a la base de datos.
- Eficiencia: Especialmente útil cuando se accede a datos relacionados de muchos objetos en un bucle.
- Simplicidad: Facilita la escritura de código más limpio y eficiente al manejar relaciones en Django.

# Optimización

## **prefetch\_related**

Hace que Django realice una consulta separada para obtener todas las familias relacionadas y luego las une a las personas en Python, optimizando así el acceso a relaciones ManyToMany.

# Optimización

Ambos métodos (`select_related` y `prefetch_related`) son herramientas poderosas para optimizar consultas en Django, cada uno adecuado para diferentes tipos de relaciones.

# Buenas prácticas

**<Despedida>**

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**