**1. ¿Cuáles son las bases de datos soportadas por Django y en qué se diferencian?**

Django es compatible con varios motores de bases de datos, incluyendo PostgreSQL, MySQL, SQLite y Oracle. Cada uno de estos motores tiene sus propias características y funcionalidades. Por ejemplo, PostgreSQL es conocido por su robustez y soporte para características avanzadas, mientras que SQLite es una opción ligera y fácil de usar para aplicaciones más pequeñas. La elección del motor de base de datos depende de los requisitos específicos de tu proyecto.

**2. ¿Qué es una migración en Django y para qué se utiliza?**

Una migración en Django es un proceso que permite gestionar y aplicar cambios en la estructura de la base de datos a lo largo del tiempo, de forma automatizada. Las migraciones se utilizan para crear, modificar o eliminar tablas, columnas y otros elementos de la base de datos de manera controlada y sin perder los datos existentes. Esto es especialmente útil cuando se desarrollan aplicaciones en equipo o cuando se necesita actualizar la base de datos en producción sin interrumpir el servicio.

**3. ¿Cuál es la diferencia entre usar consultas SQL y consultas ORM en Django?**

En Django, puedes utilizar tanto consultas SQL directamente como consultas ORM (Object-Relational Mapping). Las consultas SQL te permiten escribir consultas en el lenguaje específico del motor de base de datos que estés utilizando, como PostgreSQL o MySQL. Por otro lado, las consultas ORM te permiten interactuar con la base de datos utilizando el lenguaje de programación Python, abstrayendo los detalles del motor de base de datos subyacente. Esto facilita el desarrollo al proporcionar una sintaxis más legible y una capa de abstracción que simplifica las operaciones comunes.

**4. ¿Cómo se instalan los paquetes de base de datos en Django y cuál es su importancia?**

Los paquetes de base de datos en Django se instalan a través de las bibliotecas de conexión específicas de cada motor de base de datos. Por ejemplo, para utilizar PostgreSQL, puedes instalar el paquete `psycopg2` o `psycopg2-binary`. Para MySQL, puedes instalar `mysqlclient` o `PyMySQL`. La importancia de instalar estos paquetes es que proporcionan a Django la capacidad de comunicarse con el motor de base de datos elegido y ejecutar las consultas necesarias.

**5. ¿Qué ventajas ofrece Django como ORM para la integración con una base de datos?**

Django ofrece varias ventajas como ORM (Object-Relational Mapping) para la integración con una base de datos:

- Abstracción de la base de datos: Django proporciona una capa de abstracción que permite trabajar con diferentes motores de bases de datos sin tener que preocuparse por las diferencias en su sintaxis SQL.

- Sintaxis legible: Las consultas ORM de Django utilizan una sintaxis legible en Python, lo que facilita el desarrollo y el mantenimiento del código.

- Seguridad: Django se encarga automáticamente de evitar las vulnerabilidades comunes de seguridad en las consultas a la base de datos, como la inyección SQL.

- Portabilidad: Utilizando el ORM de Django, puedes escribir código que sea independiente del motor de base de datos subyacente. Esto facilita la migración de una base de datos a otra sin tener que modificar el código de la aplicación.

- Funcionalidades adicionales: Django ofrece características adicionales, como la gestión de relaciones entre modelos, la generación automática de esquemas de base de datos y la capacidad de realizar operaciones en la base de datos de forma transaccional.

Estas ventajas hacen que Django sea una opción popular para la integración con bases de datos en el desarrollo web.