## ¿Qué ventajas observas de la separación del código de las clases que acceden a la BBDD de las clases que implementan la lógica de negocio, en este caso, solo la clase principal?

Desde mi punto de vista, siempre que las cosas en un programa estén mas organizadas es algo positivo. En este caso, al tenerlo separado, en caso de tener algún problema con la propia conexión no habría que cambiar nada mas que lo que ejecuta la conexión, dejando los métodos sin tocar.

Por lo tanto nos permite reutilizar gran parte del código entre clases al estar completamente separadas.

## ¿Se te ocurre alguna propuesta para no tener que crear el objeto Connection en la clase principal?.

Creo que seria mas optimo abrir la conexión cuando se va a realizar alguna consulta o modificación, ya que dejar la conexión abierta en este tipo de programas puede resultar mas optimo, pero si no todos los métodos accediesen a la BBDD, sino que solo lo hiciese por ejemplo al arrancar el programa y al cerrarlo estaríamos consumiendo recursos de forma absurda.

Otra opción seria crear el objeto Connection en una clase, y que, mediante herencia, se fuese pasando a los métodos necesarios.

## Mejoras añadidas

He añadido una mejora que consiste en obtener la cantidad de vehículos y de propietarios.

De manera que se puede saber el alcance de la BBDD sin tener que contar los vehículos manualmente.

También adjunto el SQL generado para crear las tablas, crear las tuplas y diversas pruebas.