

ACTIVIDAD

Tenemos una base de datos llamada conserionario con dos tabals relacionadas propietario y coches

- Propietario
 - o Dni : varchar(10) : clave primaria
 - o Nombre: varchar(40)
 - o Edad int
- Coche
 - o Matricula : varchar(10) : clave primaria
 - o Marca : varchar(20)
 - o Precio : int
 - o Dni varchar(10) : clave ajena

Tareas:

- Cargar el driver JDBC.
- Conectarse a la Base de Datos utilizando la clase Connection.
- Crear sentencias SQL, utilizando objetos de tipo Statement.
- Ejecutar las sentencias SQL a través de los objetos de tipo Statement.
- En caso que sea necesario, procesar el conjunto de registros resultante utilizando la clase ResultSet.

1.- Vamos a crearnos un proyectop Ant para poder incluir el driver mediante un jar

2.- Conectarse a la base de datos

La conexión a la base de datos concesionario se desarrollará mediante a clase java AccesoDatos.java. Las pruebas las realizaremos sobre la

clase PruebaAccesoDatos que solo contendrá una función main desde donde se llamará a todas las funciones de AccesoDatos.

3.- Consultas a la base de datos

Obtener todos los registros de las tablas propietario y coche de la base de datos concesionario.

Para recuperar la información de la base de datos se utilizará la clase Statement, cuyos objetos se crean a partir de una conexión.

Para ejecutar cualquier consulta SQL de tipo SELECT utilizamos el método executeQuery que retorna un conjunto de registros en un objeto de la clase ResultSet

4.- Consultas de actualización

Para actualizar la base de datos con sentencia SQL tipo UPDATE, INSERT o DELETE, es necesario ejecutar el método executeUpdate que no devuelve un ResultSet como resultado, sino la cantidad de registros afectados

4.1

Método para modificar la tabla coches pasando la matrícula del coche que se quiere modificar y el nuevo precio. OJO las cadenas en la condición es necesario ponerlas entre comillas simples

4.2

Método para borrar el coche cuyo matricula se pasa como argumento

4.3

Método que permite insertar un nuevo registro en la tabla coches pasándole como argumento la matrícula, marca y precio así como el dni del propietario

5.- Ejecución mediante la clase PreparedStatement

Implementar un método inserta2 que se encargue de insertar registros en la tabla propietario de forma parametrizada

Cuando se realiza la misma operación varias veces, es mejor utilizar la clase `PreparedStatement` para una ejecución eficiente.

Esta eficiencia está dada porque la consulta que se ejecute a través de un objeto `PreparedStatement` será recompilada por el motor de SQL de la fuente de datos a la cual se accede.

Esta clase permite de forma más cómoda ejecutar consultas las cuales tienen muchos parámetros

Los signos de interrogación representan los parámetros. Para sustituir los signos de interrogación utilizamos los métodos `setXXX` (pos, valor) cuyo nombre depende del tipo de parámetro

6.- Obtener la información de la base de datos mediante la clase `DatabaseMetaData`

7.- Obtener la información de las tablas propietario y coche

Implementamos un único método que recibe como argumento el nombre de la tabla.