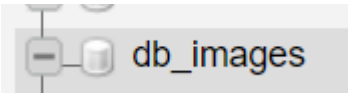
Compte rendu TD6 SLAM 5

Dans cette partie, en tant que stagiaire dans une entreprise, votre but sera de concevoir une application java client lourd en SWING qui permet de récupérer une image dans un dossier local et de l'insérer dans une base de donnée.

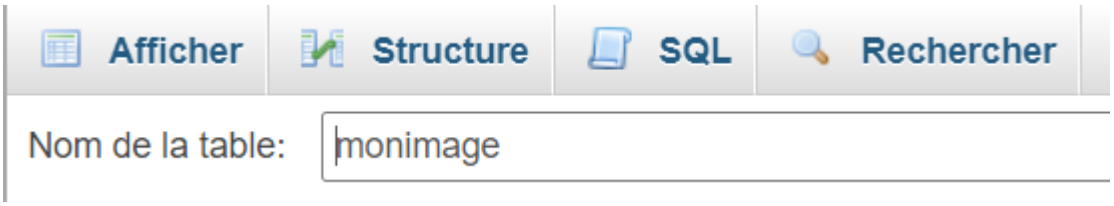
Pour cela dans un premier temps, nous allons créer une base de donnée qui s'intitule "**db_images**" et qui contiendra la table "**monimage**".

Création base de donnée :

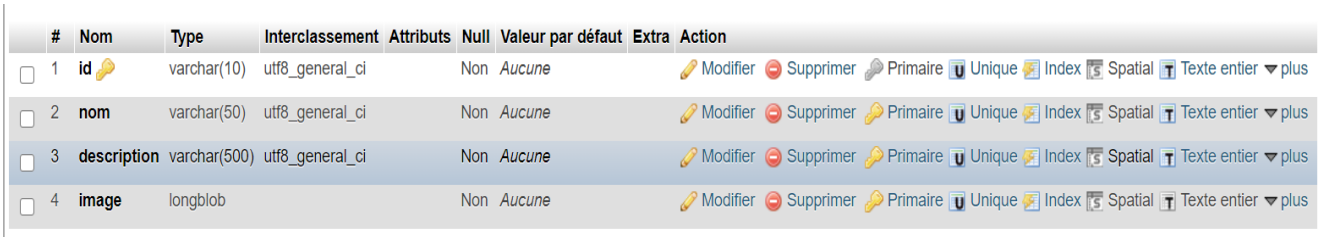
et on appuie sur "**créer**" :



Maintenant que ceci a été fait, créons la table "**monimage**" :

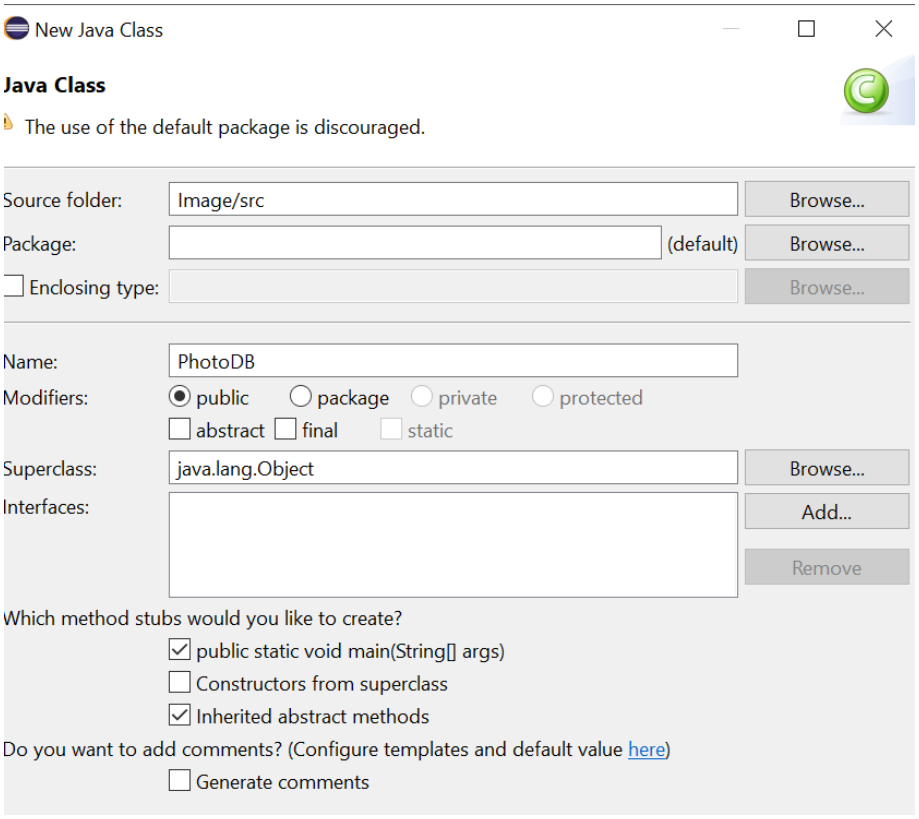


Et ainsi que ces colonnes :



#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Extra	Action
1	id	varchar(10)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Primaire Unique Index Spatial Texte entier plus
2	nom	varchar(50)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Primaire Unique Index Spatial Texte entier plus
3	description	varchar(500)	utf8_general_ci		Non	*Aucune*		Modifier Supprimer Primaire Unique Index Spatial Texte entier plus
4	image	longblob			Non	*Aucune*		Modifier Supprimer Primaire Unique Index Spatial Texte entier plus

Une fois que ceci a été fait, passons maintenant à la partie code. Pour cela nous allons créer une classe qui va se nommer "PhotoDB" comme ci-dessous :



Voici le programme :

```java
public class PhotoDB extends JFrame{
    JButton btnAjouter;
    JButton btnParcourir;
    JButton btnSupprimer;
    JLabel label;
    JTextField txtId;
    JTextField textNom;
    JTextArea area;
    String s;

    public PhotoDB() {
        super ("Insérer une photo dans une base de donnée");
        this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        btnAjouter= new JButton("Ajouter");
        btnAjouter.setBounds(200, 300, 100, 40);

        btnParcourir = new JButton("Parcourir");
        btnParcourir.setBounds(80, 300, 100, 40);

        btnSupprimer= new JButton("Supprimer");
        btnSupprimer.setBounds(80, 300, 100, 40);

        txtId= new JTextField("ID");
        txtId.setBounds(320, 290, 100, 20);

        textNom= new JTextField("Nom");
        textNom.setBounds(320, 330, 100, 20);

        area= new JTextArea("Description",100,100);

        JScrollPane pane= new JScrollPane(area);
        pane.setBounds(450, 270, 200, 100);


        label = new JLabel();
        label.setBounds(10, 10, 670, 250);


        btnParcourir.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JFileChooser fileChooser = new JFileChooser();
                fileChooser.setCurrentDirectory(new File (System.getProperty("user.dir")));
                FileNameExtensionFilter filter = new FileNameExtensionFilter("*.IMAGE","jpg","gif","png");
                fileChooser.addChoosableFileFilter(filter);
                int result = fileChooser.showSaveDialog(null);
                if(result==fileChooser.APPROVE_OPTION) {
                    File selectedFile = fileChooser.getSelectedFile();
                    String path = selectedFile.getAbsolutePath();
                    label.setIcon(ResizeImage(path));
                    s = path;
                }else if(result==JFileChooser.CANCEL_OPTION) {
                    System.out.println("Pas de données");
```

Dans cette première partie du code, nous avons défini les variables des boutons, des champs de texte,etc…

Par la suite, nous avons commencé à créer des boutons avec leurs noms (par exemple : ajouter, supprimer, parcourir) et leurs attribuer des tailles. Et pareil pour les champs de texte.

Après, nous avons géré l'événement pour le bouton "**parcourir**" avec "**ActionListener**". Puis nous avons défini les extensions des images.

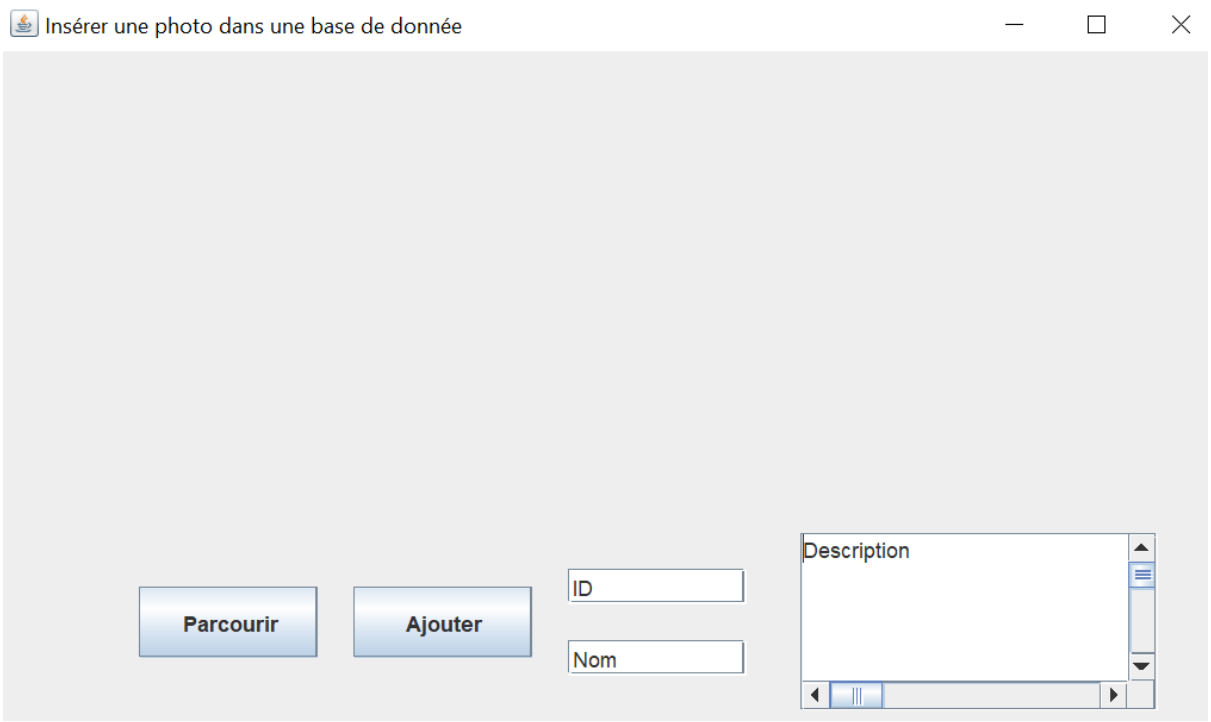
```
77
78              }
79          });
80
81          btnAjouter.addActionListener(new ActionListener() {
82              @Override
83              public void actionPerformed(ActionEvent e) {
84                  try {
85                      Connection con =DriverManager.getConnection("jdbc:mysql://localhost/db_images","root","root");
86                      PreparedStatement ps= con.prepareStatement("insert into monimage(id,nom,description,image)values(?,?,?,?)")
87                      InputStream is= new FileInputStream(new File(s));
88
89                      ps.setString(1,txtId.getText());
90                      ps.setString(2,textNom.getText());
91                      ps.setString(3,area.getText());
92                      ps.setBlob(4,is);
93                      ps.executeUpdate();
94                      JOptionPane.showMessageDialog(null, "Engistrement effectué");
95                  }catch (Exception ex) {
96                      ex.printStackTrace();
97                  }
98
99
100             }
101         });
102
103         this.add(label);
104         this.add(txtId);
105         this.add(textNom);
106         this.add(pane);
107         this.add(btnAjouter);
108         this.add(btnParcourir);
109         this.add(btnSupprimer);
110         this.setLayout(null);
111         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
112         this.setSize(700, 420);
113         this.setVisible(true);
114     }
115
116     public ImageIcon ResizeImage(String imgPath) {
117         ImageIcon MyImage = new ImageIcon(imgPath);
118         Image img = MyImage.getImage();
119         Image newImage = img.getScaledInstance(label.getWidth(),label.getHeight(), Image.SCALE_SMOOTH);
120         ImageIcon image = new ImageIcon(newImage);
121         return image;
122     }
123
124
125     public static void main(String[] args) {
126         new PhotoDB();
127
128     }
```

Dans cette deuxième partie du code, nous avons géré l'événement pour le bouton "**Ajouter**".

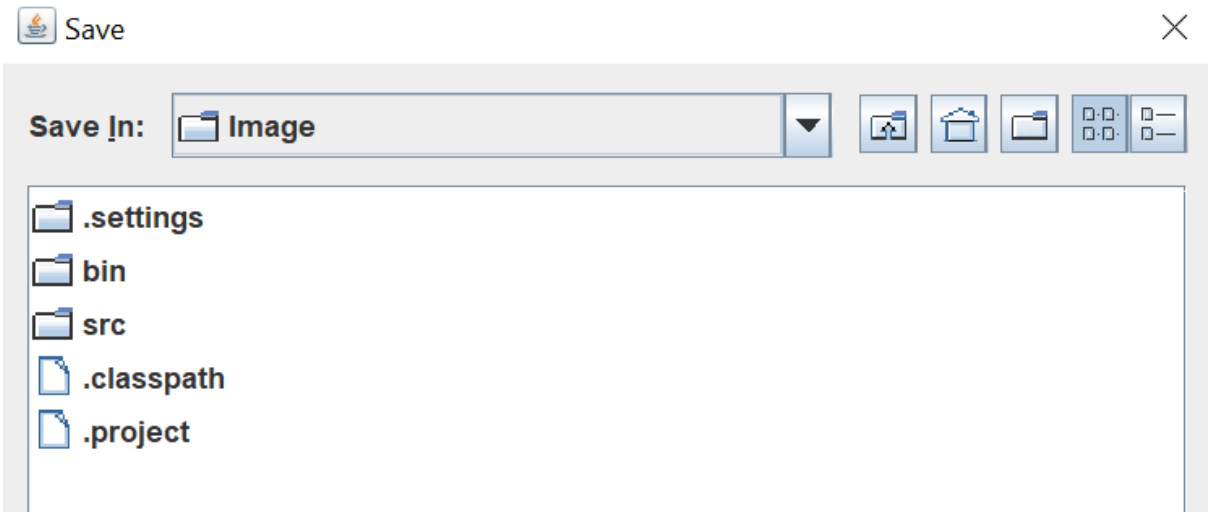
Par la suite, nous avons fait la connexion avec la base de données que nous avons créé précédemment notamment avec la requête "**INSERT**" qui va introduire les descriptions et les id de nos futures images dans la base de données.

Ensuite nous avons défini la taille de nos images.

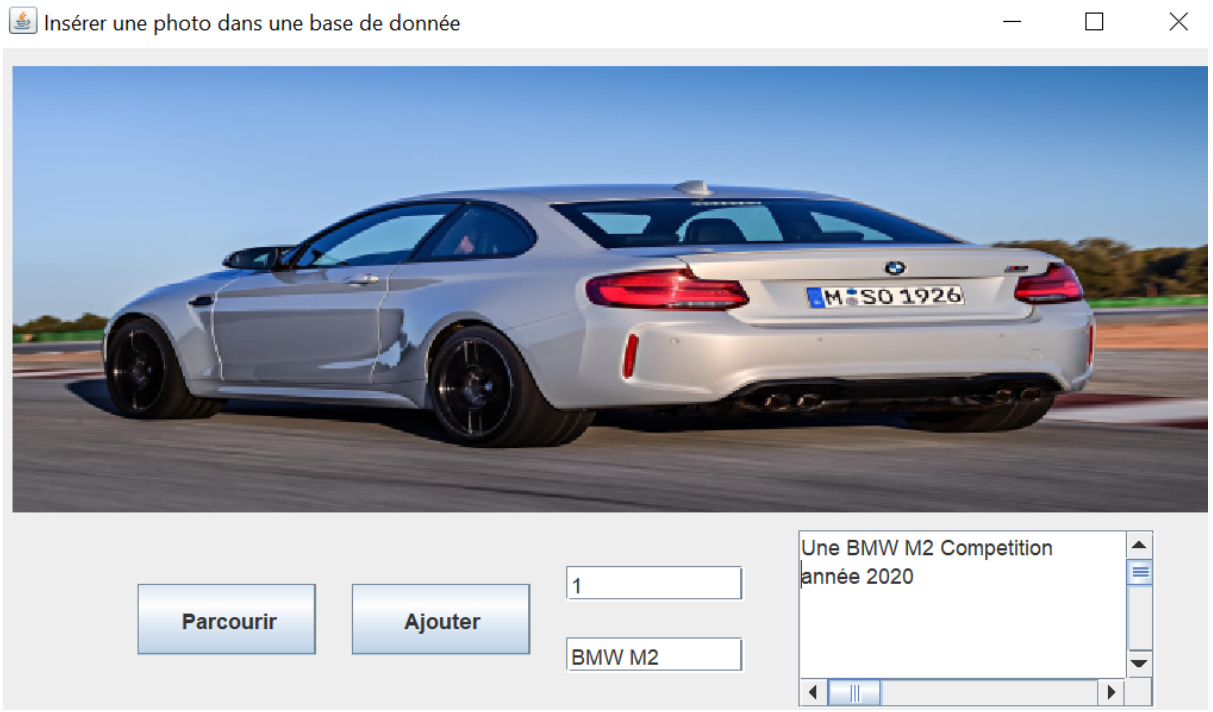
Voici le résultat :



Appuyons sur "Parcourir" :



Allons chercher nos images :



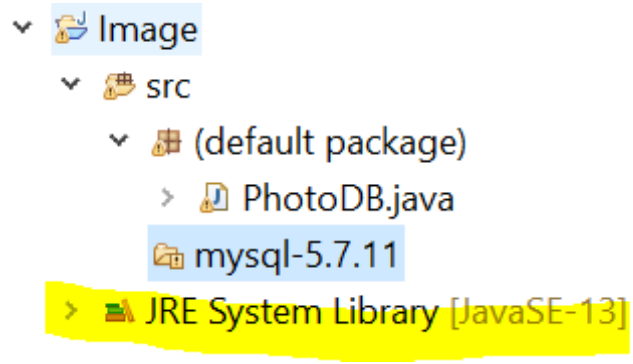
Maintenant que nous avons l'image, nous allons l'ajouter dans la base de donnée :

Oups… la BMW ne veut pas s'ajouter quand j'appuie sur le bouton !

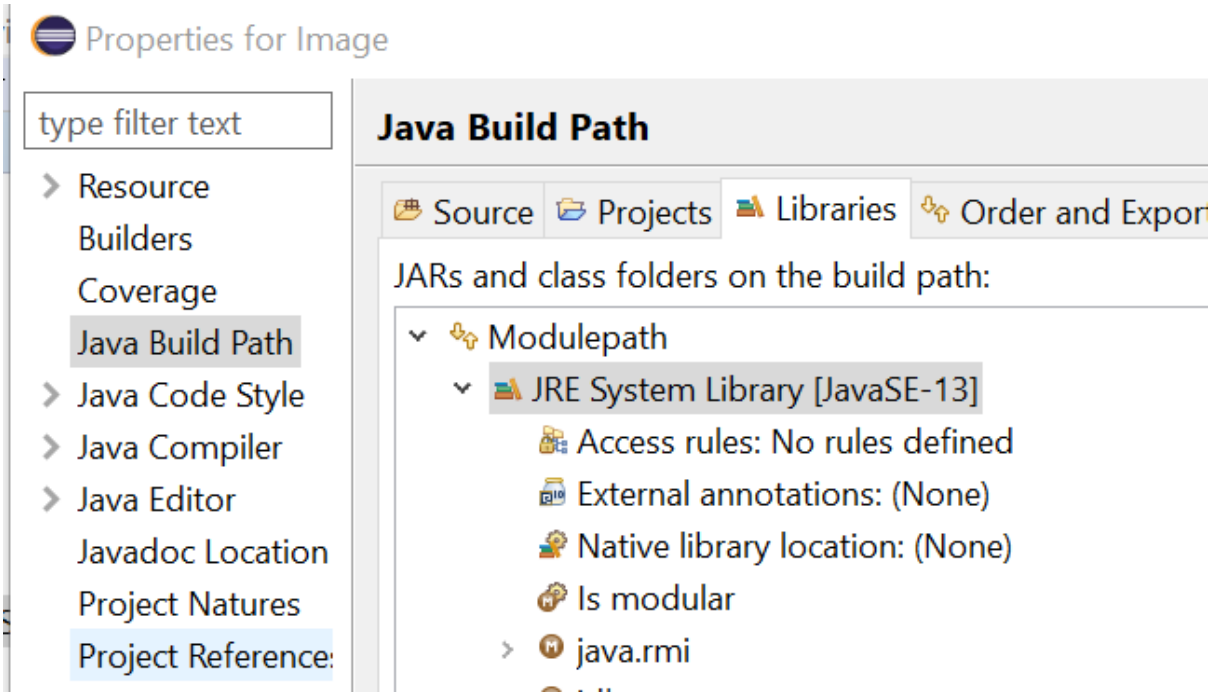
Voyons ce qu'on peut faire !

Nous avons toute une liste d'erreurs ! Pour cela nous devons ajouter le connecteur dans la librairie comme ci-dessous :

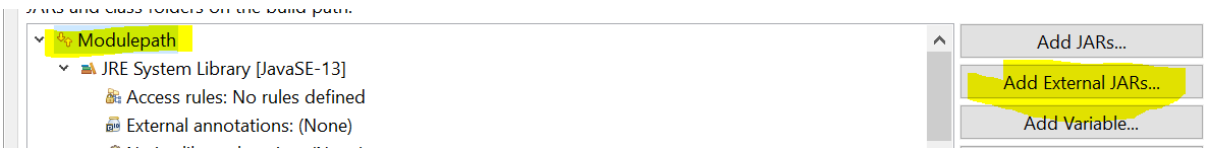
Nous devons dans un premier temps aller à la librairie comme ci-dessous :



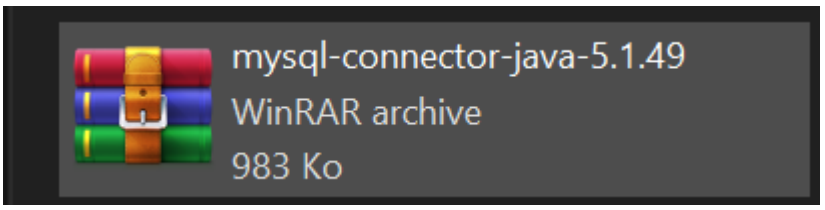
Ensuite nous allons sur "**Built path**" en faisant clic droit comme ci-dessous :



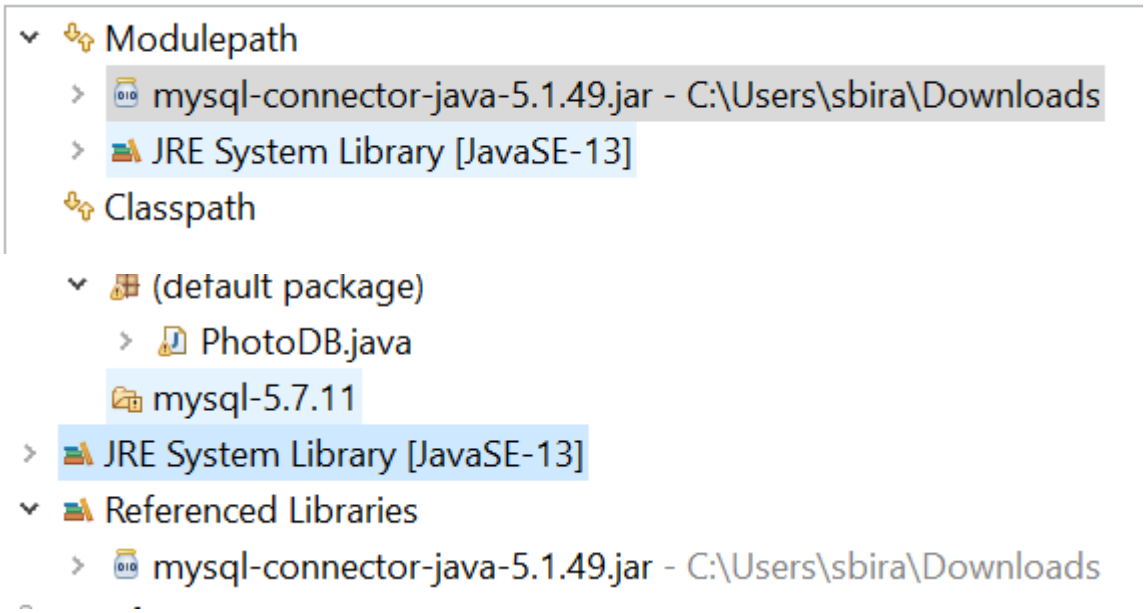
On appuis sur "module" et on clique sur "**add External JAR's**" comme ci-dessous :



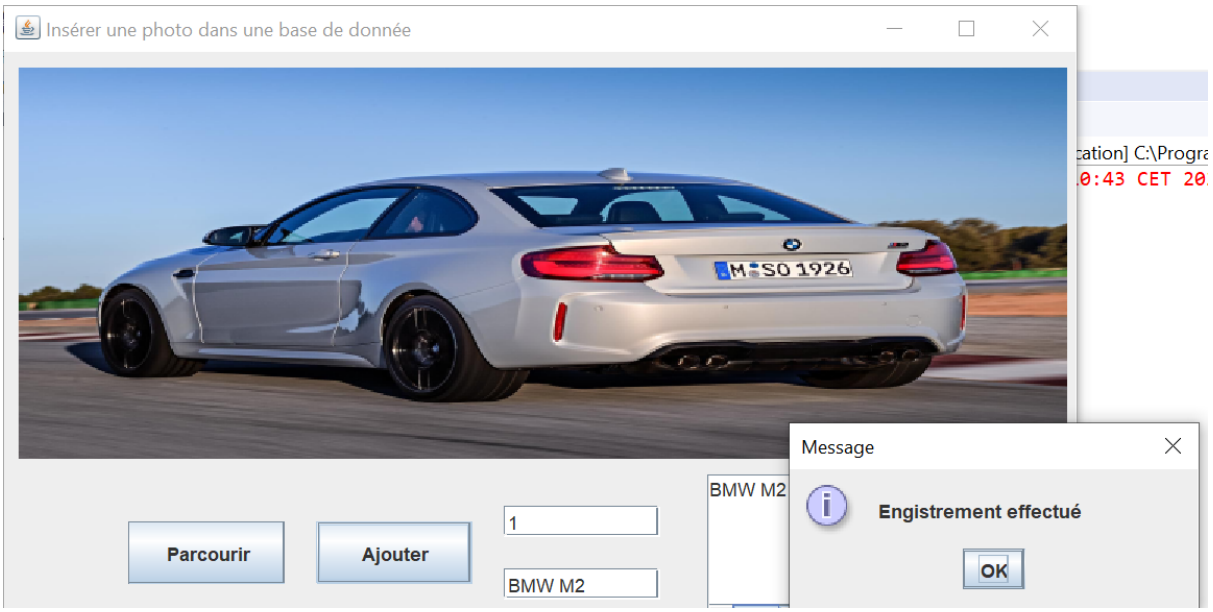
Et ensuite on met le connecteur :



Voici une fois ajouté :



Maintenant cliquons "**apply and close**" et allons tester à nouveau si ca fonctionne :



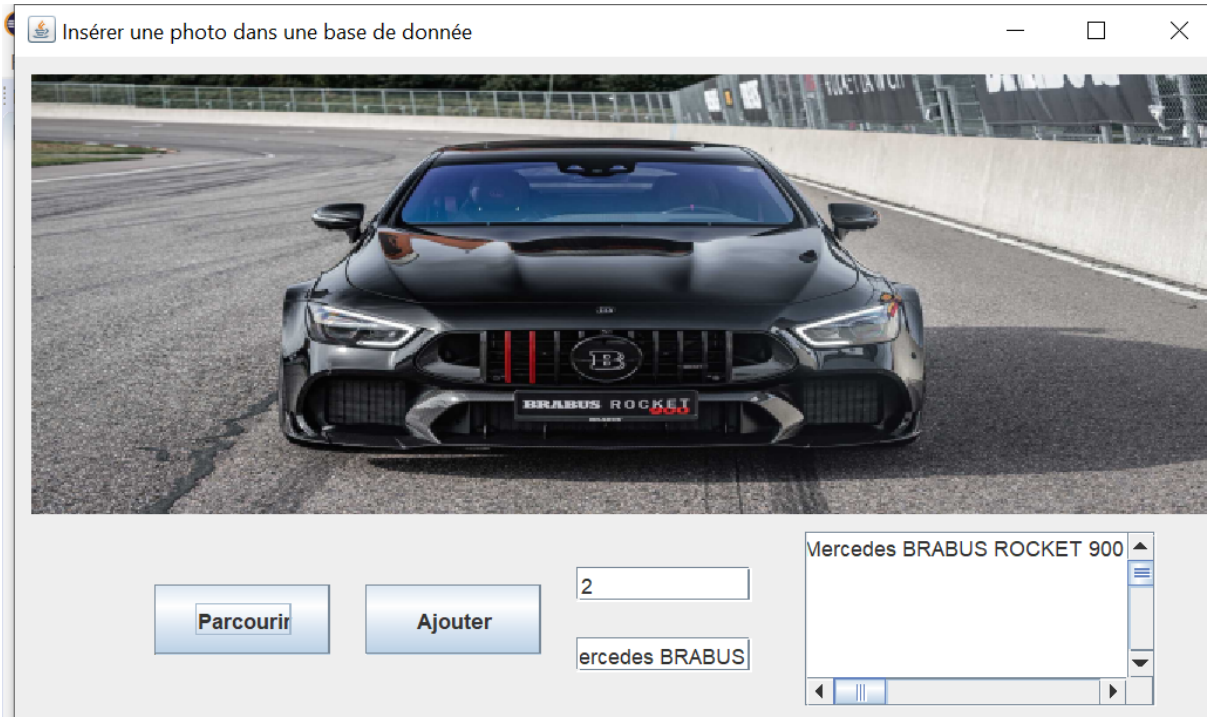
Ca fonctionne la M2 a repris son chemin vers la base de donnée avec le message de confirmation. Mais voyons si elle est bien arrivée à destination (base de donnée **"db_images"**) :

+ Options

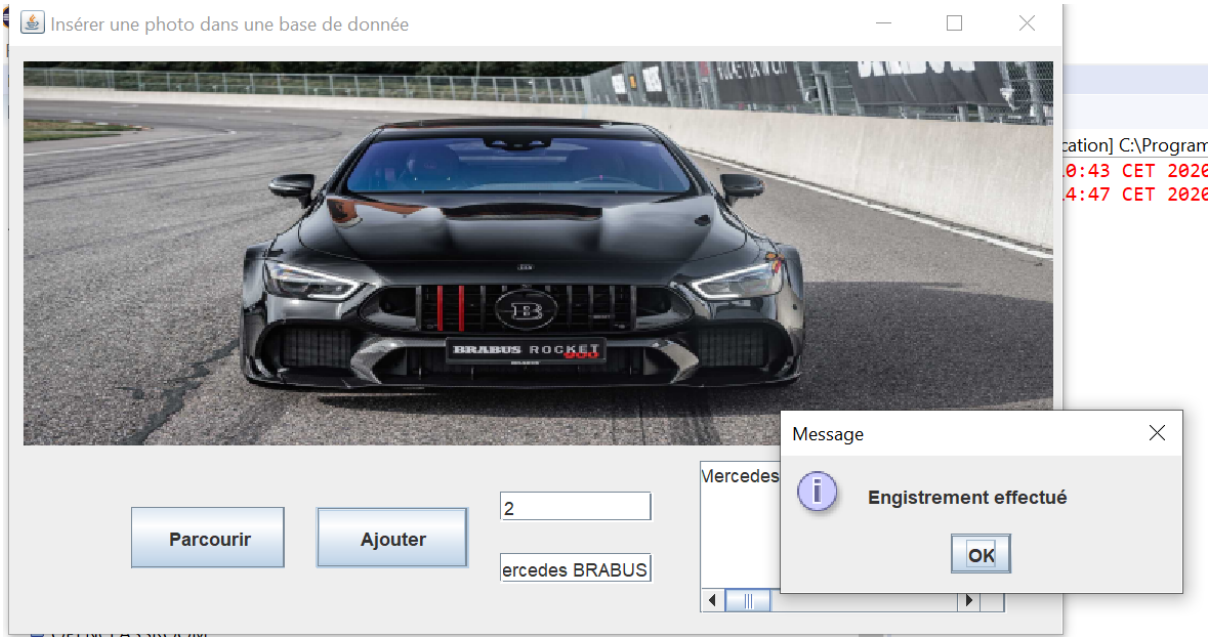
←T→	id	nom	description	image
Modifier Copier Effacer	1	BMW M2	BMW M2 Competition	[BLOB - 169,3 Kio]

La voiture est arrivée à destination.

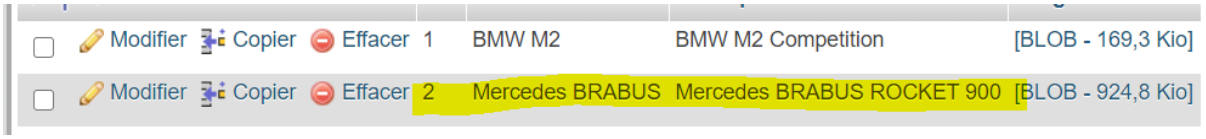
Testons pour une autre image…



Testons !



Regardons la base de donnée :



☐	Modifier	Copier	Effacer	1	BMW M2	BMW M2 Competition	[BLOB - 169,3 Kio]
☐	Modifier	Copier	Effacer	2	Mercedes BRABUS	Mercedes BRABUS ROCKET 900	[BLOB - 924,8 Kio]

Elle aussi est arrivée à destination !

Les éléments ont été ajoutés correctement à la base de donnée avec le id, le nom et la description.