

RAPPORT EXAMEN BIG DATA

INFORMATIQUE ET INGENIERIE DES DONNES II

APPLICATION JAVA PERMETTANT MANIPULATION

BD HBASE

AMADOU BARRY DJOULDE

KAPULA DAVID DAVID

Professeur Soussi Nassima

AVRIL 2022

Partie 1. Interface Graphique

Nous avons commencé à implémenter l'application grâce à Swing

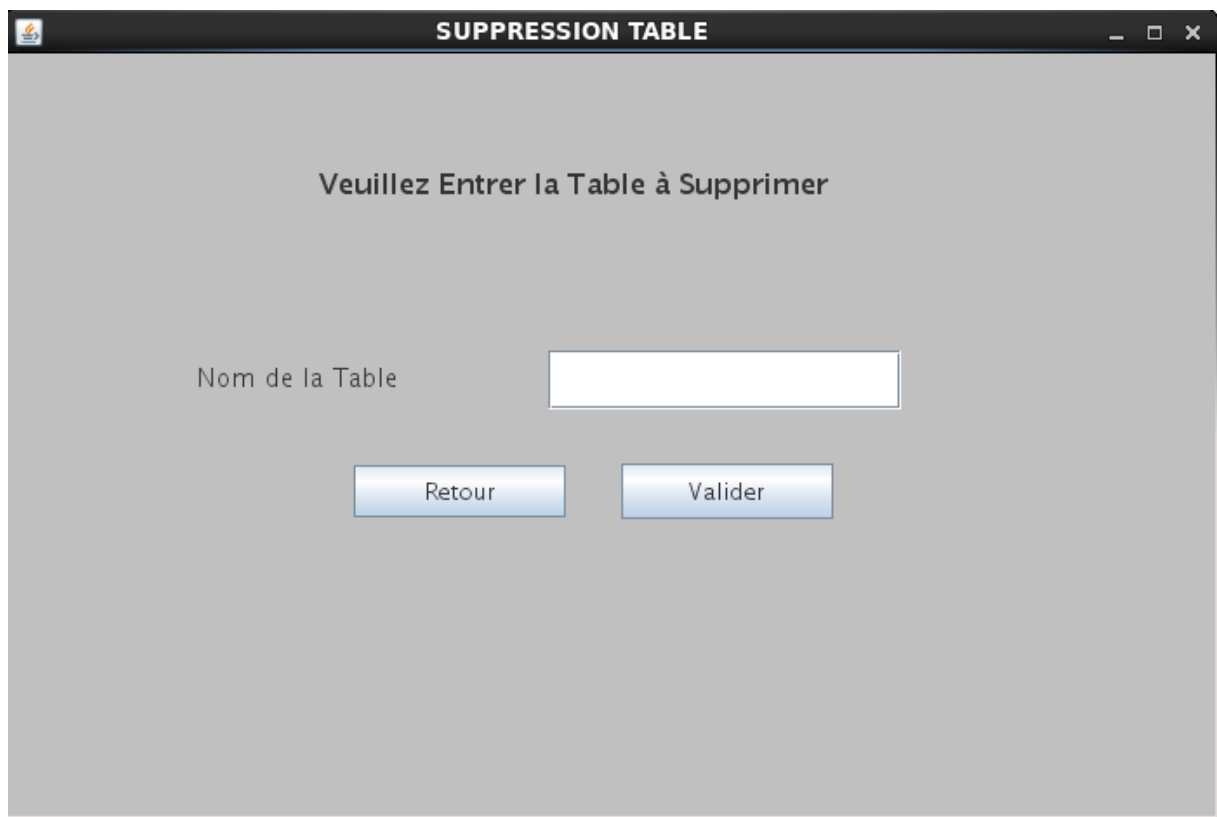
A partir d'un menu principal l'utilisateur aura certaines options telle que : La création d'une table, affichage d'une table, insertion valeur, suppression table, affichage d'une ligne



a. Création Table :

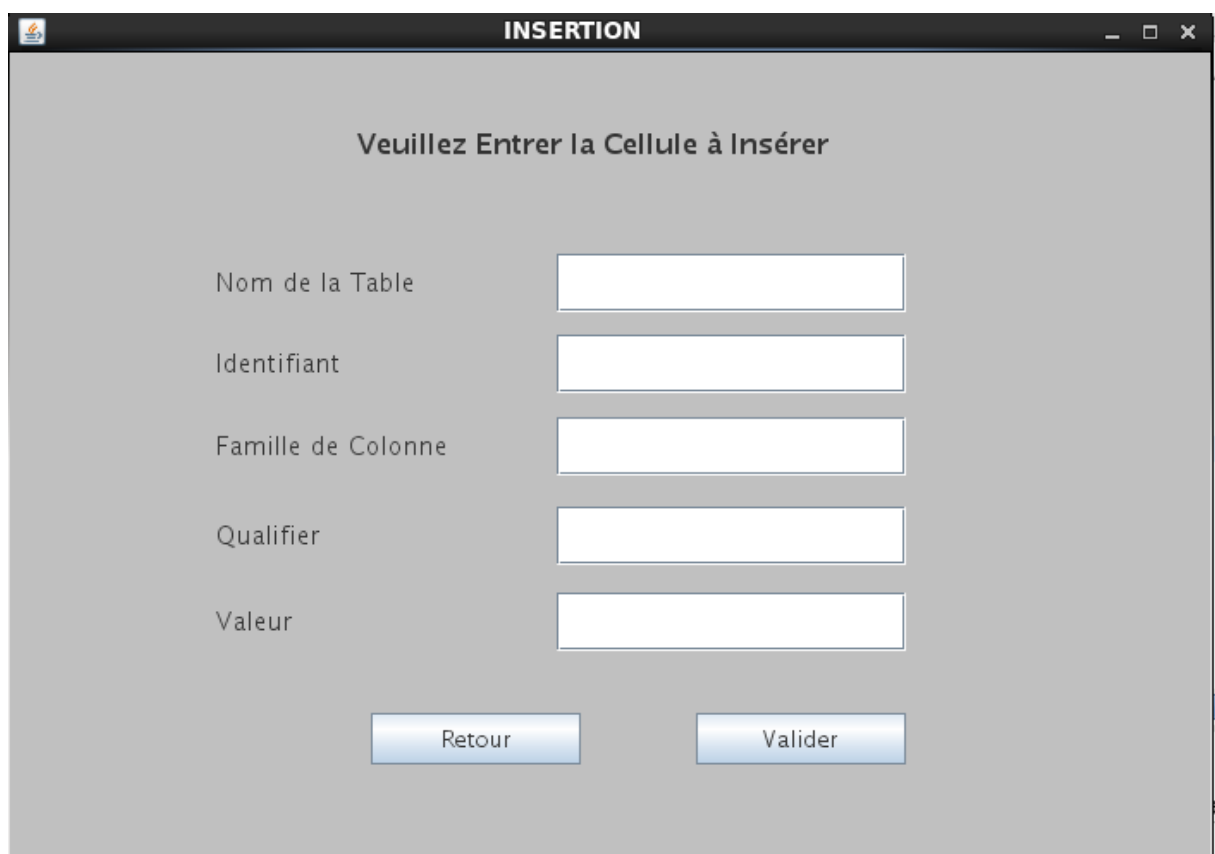


b. Supprimer une table



The screenshot shows a window titled "SUPPRESSION TABLE" with a standard Windows-style title bar. The main content area has a light gray background. At the top, centered, is the text "Veuillez Entrer la Table à Supprimer". Below this, on the left, is the label "Nom de la Table". To the right of the label is a white rectangular text input field. Below the input field, there are two blue buttons with white text: "Retour" on the left and "Valider" on the right.

c. Insertion



The screenshot shows a window titled "INSERTION" with a standard Windows-style title bar. The main content area has a light gray background. At the top, centered, is the text "Veuillez Entrer la Cellule à Insérer". Below this, there are five labels on the left, each followed by a white rectangular text input field on the right: "Nom de la Table", "Identifiant", "Famille de Colonne", "Qualifier", and "Valeur". At the bottom of the form, there are two blue buttons with white text: "Retour" on the left and "Valider" on the right.

d. Affichage Ligne

AFFICHAGE LIGNE

Veuillez Entrer la Ligne à Afficher

Nom de la Table

Identifiant

Famille de Colonne

Qualifier

e. Affichage Table

AFFICHAGE TABLE

Veuillez Entrer la Table à Afficher

Nom de la Table

Partie 2. Code Source

Note : Pour la connexion avec HBase nous instancions premièrement la classe de configuration ensuite nous créons la connexion grâce.

a. Création Table

Pour la création de la table nous avons rencontré la difficulté de placer un nombre de famille sur précision de l'utilisateur et donc nous avons imposé un nombre de famille à deux.

```
// Pour creation table
public boolean createHbaseTable(String tableName,String family1,String family2) throws IOException {
    Configuration config = HBaseConfiguration.create();
    Connection connection = ConnectionFactory.createConnection(config);
    Admin admin = connection.getAdmin();

    HTableDescriptor ht = new HTableDescriptor(tableName.valueOf(tableName));
    ht.addFamily(new HColumnDescriptor(family1));
    ht.addFamily(new HColumnDescriptor(family2));
    System.out.println("connecting");

    if(!admin.tableExists(tableName.valueOf(tableName))){
        System.out.println("Creating Table");
        admin.createTable(ht);
        System.out.println("Done.");
        return true;
    }else{
        System.out.println("La table existe deja");
        return false;
    }
}

JButton btnValiderCreation = new JButton("Valider");
btnValiderCreation.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        HBaseFonctions t=new HBaseFonctions();
        try {
            Creation frame = new Creation();
            if(t.createHbaseTable(tableName.getText(),FC1.getText(),FC2.getText())){

                JOptionPane.showMessageDialog(frame, "Creation Table Reussie");
            }else
                JOptionPane.showMessageDialog(frame, "Table existe deja");

        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
```

Les données entrées par l'utilisateur sont récupérer ensuite la table entrée subit une vérification d'existence si elle n'existe pas la fonction retourne true spécifiant que la classe peut être créée sinon false qui sera donc traitée lors de l'action « valider »

que posera l'utilisateur, un popup sera donc afficher avec le message Création Table Réussie ou dans le cas contraire Table existe déjà

b. Supprimer une table

Pour la suppression d'une table il faut que la table soit désactivée avant qu'elle soit supprimer et donc l'utilisateur entrera le nom de la table et si la table n'existe pas il lui sera retourné un message lui disant que la table n'existe pas.

```
//Supprimer Une table

public boolean supprimerUser(String tableName) throws IOException{
    Configuration conf = HBaseConfiguration.create();
    HBaseAdmin admin = new HBaseAdmin(conf);
    if(admin.tableExists(tableName)){
        admin.disableTable(tableName);
        admin.deleteTable(tableName);
        return true;
    }else{
        System.out.println("Table n'existe pas");
        return false;
    }
}

JButton btnValiderSuppression = new JButton("Valider");
btnValiderSuppression.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        HBaseFonctions t=new HBaseFonctions();
        try {
            Suppression frame = new Suppression();
            if(t.supprimerUser(tableName.getText())){

                JOptionPane.showMessageDialog(frame, "Suppression Reussie");
            }else
                JOptionPane.showMessageDialog(frame, "La Table n'existe pas");

        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
```

Pareil pour la Suppression dans ce cas ci lorsque la Table existe la fonction retourne donc True pour pouvoir passer à l'action supprimer et afficher le popup Suppression Réussi et dans le cas contraire la fonction retourne False et l'action ne pourra pas s'effectuer ce qui affichera le popu avec message la Table n'existe pas

c. Afficher une table

Pour l'affichage, l'utilisateur entre le nom de la table et les données seront afficher grâce à la fonction scan

```
public void affiche() throws IOException{

    HBaseFonctions t = new HBaseFonctions();
    ResultScanner res = t.afficherUser(tableName.getText());
    String[] columnNames = { "Famille Colonne", "Colonne", "Valeur"};
    DefaultTableModel model = new DefaultTableModel(columnNames, 0);
    for (Result r = res.next(); r != null; r = res.next()){

        byte [] value = r.getValue(Bytes.toBytes("info_maj"), Bytes.toBytes("ename"));
        byte [] value1 = r.getValue(Bytes.toBytes("info_sup"), Bytes.toBytes("manager"));
        String valueStr = Bytes.toString(value);
        String fam = "info_maj";
        String col = "ename";
        String valueStr1 = Bytes.toString(value1);
        String fam1 = "info_sup";
        String col1 = "manager";
        model.addRow(new Object[]{fam, col, valueStr});
        model.addRow(new Object[]{fam1, col1, valueStr1});

    }

    JTable j = new JTable();
    j.setModel(model);
    JScrollPane scrollPane = new JScrollPane(j);
    scrollPane.setBounds(0, 250, 800, 1000);
    add(scrollPane);
}
```

```
//Pour afficher une table
public ResultScanner afficherUser(String tableName) throws IOException{

    Configuration config = HBaseConfiguration.create();

    Connection connection = ConnectionFactory.createConnection(config);
    Table table = connection.getTable(TableName.valueOf(tableName));
    Scan s = new Scan();

    // Scanning the required columns
    s.addColumn(Bytes.toBytes("info_maj"), Bytes.toBytes("ename"));
    s.addColumn(Bytes.toBytes("info_sup"), Bytes.toBytes("manager"));

    ResultScanner scanner = table.getScanner(s);
    return scanner;
    //scanner.close();

}
```

d. Insertion Valeur

La valeur à insérer sera placée grâce à la précision de la cellule contenant donc la table dans laquelle il veut placer sa valeur, l'identifiant correspondant, la famille de colonne et la colonne

```
//Pour ajouter une ligne
public boolean addUser(String tableName,String id,String family1,String col,String valeur) throws IOException{

    Configuration config = HBaseConfiguration.create();
    Connection connection = ConnectionFactory.createConnection(config);
    Admin admin = connection.getAdmin();
    Table table = connection.getTable(TableName.valueOf(tableName));
    if(admin.tableExists(TableName.valueOf(tableName))){
        byte[] row1 = Bytes.toBytes(id);
        Put p = new Put(row1);
        p.addColumn(family1.getBytes(), col.getBytes(), Bytes.toBytes(valeur));
        table.put(p);
        return true;
    }else
        return false;
    }

    JButton btnValiderInsertion = new JButton("Valider");
    btnValiderInsertion.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            HBaseFonctions t=new HBaseFonctions();
            try {
                Insertion frame = new Insertion();
                if(t.addUser(tableName.getText(),id.getText(),FC.getText(),Q.getText(),valeur.getText()))
                    JOptionPane.showMessageDialog(frame, "Insertion Reussie");
                else
                    JOptionPane.showMessageDialog(frame, "Table Inexistante");
            } catch (IOException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
        }
    });
}
```

L'utilisateur entre la donnée en précisant la cellule ce qui est récupéré et placé dans la table si elle existe sinon un message d'erreur se déclenche Table Inexistante.

e. Affichage Ligne

Enfin pour l'affichage l'utilisateur précise la cellule afin de recevoir la ligne à afficher

```
//Pour afficher une ligne
public String recuperer(String tableName,String id,String fam1,String col) throws IOException{

    Configuration config = HBaseConfiguration.create();
    HTable table = new HTable(config, tableName); //mettre tablename
    Get g = new Get(Bytes.toBytes(id)); //mettre id

    Result result = table.get(g);

    byte [] value = result.getValue(Bytes.toBytes(fam1),Bytes.toBytes(col)); //mettre famille et colonne

    // afficher les valeurs |
    String valeur = Bytes.toString(value);
    System.out.println("valeur: " + valeur );
    return valeur;
}

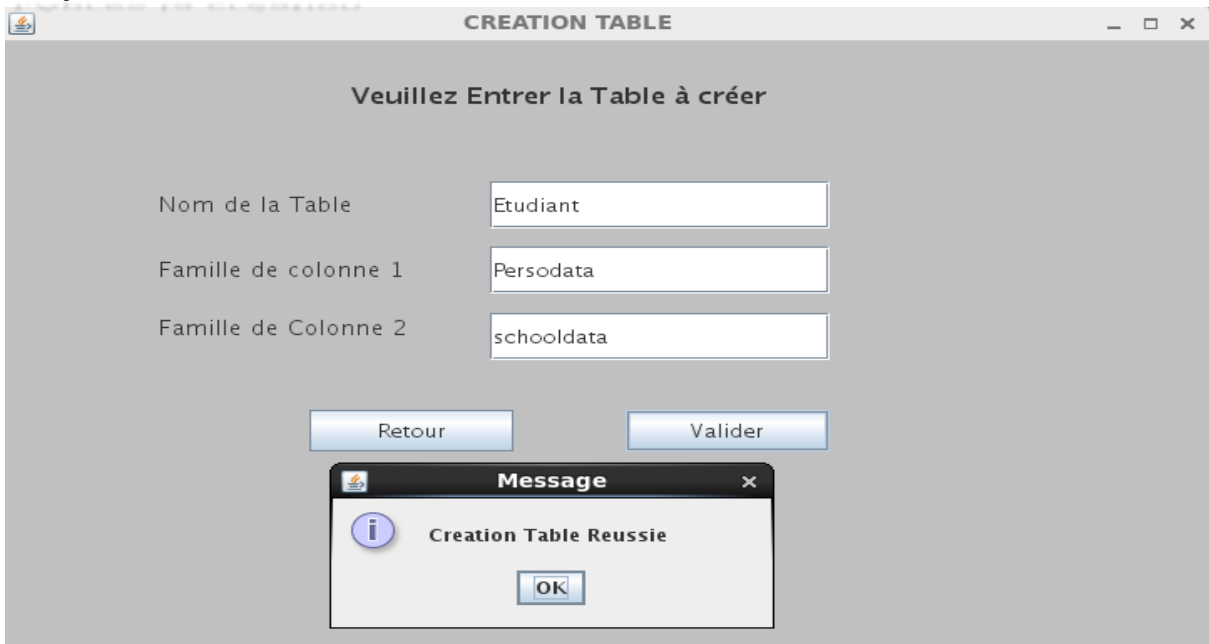
JButton btnValiderCreation = new JButton("Valider");

JButton btnValiderAl = new JButton("Valider");
btnValiderAl.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        HBaseFonctions t=new HBaseFonctions();
        try {
            AffichageLigne frame = new AffichageLigne();
            String val=t.recuperer(tableName.getText(),id.getText(),FC.getText(),qualifier.getText());
            text.setText(id.getText()+" "+FC.getText()+" "+qualifier.getText()+"="+val);
        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
});
```


Pour Afficher la ligne, l'utilisateur entre la table, l'id, la famille de colonne et la colonne, ceux-ci sont récupérés et la valeur correspondante est retrouvée afin d'être affichée sur un espace dédié sur la fenêtre

Partie 3. Exécution

1. Après la création



2. Après suppression

a. cas où la table n'existe pas




b.table existe

SUPPRESSION TABLE

Veuillez Entrer la Table à Supprimer

Nom de la Table

Message

 **Suppression Reussie**

3.affichage ligne

AFFICHAGE LIGNE

Veuillez Entrer la Ligne à Afficher

Nom de la Table

Identifiant

Famille de Colonne

Qualifier

insertion

INSERION

Veuillez Entrer la Cellule à Insérer

Nom de la Table

Etudiant

Identifiant

id1

Famille de Colonne

Personnaldata

Qualifier

nom

Valeur

David

Retour

Valider

Message

✕

i

Insertion Reusiie

OK

/ [JavaSE-1.7]

RAY);

Veuillez Entrer la Table à Afficher		
Nom de la Table	employees	
Famille Colonne	Retour	Valider
		Valeur
nfo_maj	ename	SMITH
nfo_sup	manager	7902
nfo_maj	ename	ALLEN
nfo_sup	manager	7698
nfo_maj	ename	WARD
nfo_sup	manager	7698
nfo_maj	ename	TURNER
nfo_sup	manager	7839
nfo_maj	ename	MARTIN
nfo_sup	manager	7698
nfo_maj	ename	MILLER
nfo_sup	manager	7839
nfo_maj	ename	CLARK
nfo_sup	manager	7839
nfo_maj	ename	SCOTT
nfo_sup	manager	7566
nfo_maj	ename	KING
nfo_sup	manager	NULL
nfo_maj	ename	TURNER
nfo_sup	manager	7698
nfo_maj	ename	ADAMS
nfo_sup	manager	7788
nfo_maj	ename	JAMES
nfo_sup	manager	7698
nfo_maj	ename	FORD
nfo_sup	manager	7566
nfo_maj	ename	MILLER
nfo_sup	manager	7782
nfo_maj	ename	ename
nfo_sup	manager	manager

Partie 4. Conclusion

Le travail n'a pas été de tout repos mais nous avons pu faire ce qui était en notre pouvoir et donc nous avons bénéficié de plus de temps que prévu pour ce le travail est rendu avec plus de sérieux et de rigueur à notre niveau.