

Petit lab JDBC :

Objectifs :

- Établir une connexion à la base de données.
- Exécuter une requête Update avec un Statement
- Obtenir les résultats dans un ResultSet
- Examiner quelques méthodes du ResultSet.
- Exécuter des requêtes paramétrées

Exécuter le script suivant :

```
CREATE TABLE EMPLOYESBIDON (NUMEMP NUMBER PRIMARY KEY, NOMEMP
VARCHAR2(30), PRENOMEMP VARCHAR2(30), EMPLOI VARCHAR2(30) , SALAIRE
NUMBER(8,2));

insert into employesbidon values (1,'Bidon','Dindon','PROGRAMMEUER',21000);

insert into employesbidon values (2,'Lepine','David','PROGRAMMEUER',45000);

insert into employesbidon values (3,'Valente','Olivier','PROGRAMMEUER',45000);

insert into employesbidon values (4,'Patoche','Alain','PROGRAMMEUER',45000);

insert into employesbidon values (5,'Proulx','William','PROGRAMMEUER',45000);

insert into employesbidon values (8,'Paquette','Nathalie','FINANCES',35000);

insert into employesbidon values (6,'Poitras','Clara','FINANCES',35000);

insert into employesbidon values (7,'Lechat','Remi','FINANCES',35000);
```

Exercice No1

Écrire un petit code Java-JDBC avec IntelliJ Idea qui va vous permettre de :

- 1- Charger le driver jdbc pour Oracle
- 2- Établir la connexion à la base de données Oracle
- 3- D'effectuer une mise à jour des salaires de David Lépine à 50 000 et celui de Olivier Valente à 52000. Afficher le nombre de lignes mises à jour.
- 4- Afficher les noms, prénom et les salaires de tous les employés. L'affichage se fera dans la console.
- 5- D'insérer un enregistrement. Utiliser une séquence.

Exercice No2 : Écrire un petit code Java-JDBC avec IntelliJ Idea qui va vous permettre de :

1. Écrire une requête paramétrée qui permet de faire une insertion dans la table Employesbidon. Utiliser une séquence.
2. Écrire une requête paramétrée qui permet de faire une mise à jour des salaires des employés selon le numéro d'employés;
3. Écrire une requête paramétrée qui affiche les employés selon leur emploi (paramètre).
Afficher les résultats
 - a. du début jusqu'à la fin
 - b. de la fin jusqu'au début.

Attention ! Pour la question 3, votre ResultSet doit être Scrollable

Exercice No3 :

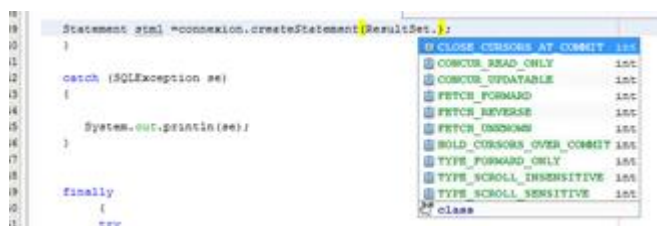
Écrire un programme Java-Jdbc avec IntelliJ Idea qui permet et qui utilise la table employesbidon pour afficher le nom, le prénom et le salaire.

- Afficher le premier enregistrement
- Le dernier enregistrement ;
- Le troisième enregistrement;
- Le troisième enregistrement à partir de l'enregistrement courant;
- À quoi correspond `Resultat.absolute(1)` ? (Resultat étant le nom du ResultSet)
- À quoi correspond `Resultat.absolute(-1)` ?
- Exécuter l'instruction `Resultat.beforeFirst()`;
- Exécuter l'instruction `Resultat.afterLast()`;
- Quel est l'enregistrement précédent du premier enregistrement ?
- Donner le programme qui permet de faire la lecture inverse (de la fin jusqu'au début)

Pour pouvoir exécuter certaines méthodes du ResultSet, vous aurez besoin de le parcourir dans les deux sens.

Définir le type de votre ResultSet au moment de la création du Statement avec l'instruction :

```
Statement stm1 = connexion.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
```



The screenshot shows a code editor with the following code:

```
10 Statement stm1 = connexion.createStatement(ResultSet.);  
11 }  
12 catch (SQLException se)  
13 {  
14     System.out.println(se);  
15 }  
16 }  
17 }  
18 }  
19 }  
20 }  
21 }  
22 }  
23 }  
24 }  
25 }  
26 }  
27 }  
28 }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 }  
35 }  
36 }  
37 }  
38 }  
39 }  
40 }  
41 }  
42 }  
43 }  
44 }  
45 }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }
```

A dropdown menu is open, showing the following options:

- TYPE_SCROLL_SENSITIVE
- TYPE_SCROLL_INSENSITIVE
- TYPE_FORWARD_ONLY
- HOLD_CURSORS_OVER_COMMIT
- FETCH_REVERSE
- FETCH_FORWARD
- CONCUR_UPDATABLE
- CONCUR_READ_ONLY
- CLOSE_CURSORS_AT_COMMIT

La correction se fera sur place