




Apprenez à programmer en Java

40 heures  Difficile

Mis à jour le 05/08/2019



Accéder à des sources de données

Bravo ! Vous avez réussi cet exercice !

Compétences évaluées



Accéder à des sources de données avec Java Database Connectivity (JDBC)

Question 1

De quoi avez-vous besoin pour permettre à Java de communiquer avec une base de données via JDBC ?

- ☐ De rien en particulier
- ☒ Du pilote JDBC de la base de données en question
- ☐ D'une bibliothèque spéciale et des pilotes JDBC de la base de données
- ☐ D'un serveur

Tout comme pour les imprimantes, les bases de données utilisent des drivers afin de pouvoir être utilisées. Dans le monde Java, les moteurs de base de données fournissent des pilotes JDBC afin de permettre au langage de communiquer avec elles. Ce pilote JDBC est en fait un fichier .jar contenant tout le nécessaire pour la communication.

Question 2

Quelle forme a l'URL de connexion à une base de données JDBC ?

- ✓ ☒ jdbc:typeDePilote://cheminReseau:portDeDestination/NomDeLaBaseDeDonnées
- ☐ typeDePilote://cheminReseau:portDeDestination/NomDeLaBaseDeDonnées
- ☐ jdbc:typeDePilote:/NomDeLaBaseDeDonnées//cheminReseau:portDeDestination
- ☐ typeDePilote:/NomDeLaBaseDeDonnées//cheminReseau:portDeDestination

La syntaxe à utiliser est à retenir, mais elle est aussi facilement retrouvable... La bonne réponse était :

jdbc:typeDePilote://cheminReseau👉ortDeDestination/NomDeLaBaseDeDonnées Ce qui, pour une base de données PostgreSQL hébergée sur le serveur 10.20.20.1, écoutant sur le port 5000 et hébergeant une base de données openclassrooms, nous donnerait : jdbc👉ostgresql://10.20.20.1:5000/openclassrooms

Question 3

Quel couple d'objets est souvent utilisé pour interroger une base de données avec JDBC ?

- ☐ Statement et Result
- ✓ ☒ Statement et ResultSet
- ☐ Statement et Set
- ☐ Statement et ResultMap
- ☐ Statement et Map

Java met à disposition le couple Statement et ResultSet afin de permettre à vos programmes Java d'interagir avec les différentes bases de données existantes.

Question 4

À quel indice commence le compte des en-têtes de colonnes via l'objet ResultSet ?

- ☐ 0
- ✓ ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4

Attention, contrairement aux indices de tableaux, les indices de colonnes SQL commencent à 1 !

Question 5

Comment créer un Statement permettant à l'objet ResultSet de pouvoir être lu d'avant en arrière avec possibilité de modification des données lues, mais aussi mise à jour des

données récupérées en cas de modification en base par une autre personne ?

- ☐ En spécifiant `ResultSet.TYPE_SCROLL_INSENSITIVE` et `ResultSet.CONCUR_UPDATABLE` lors de la création de l'objet `ResultSet`
- ✓ ☒ En spécifiant `ResultSet.TYPE_SCROLL_SENSITIVE` et `ResultSet.CONCUR_UPDATABLE` lors de la création de l'objet `Statement`
- ☐ En spécifiant `ResultSet.TYPE_SCROLL_SENSITIVE` et `ResultSet.CONCUR_READONLY` lors de la création de l'objet `Statement`
- ☐ En spécifiant `ResultSet.TYPE_SCROLL_INSENSITIVE` et `ResultSet.CONCUR_UPDATABLE` lors de la création de l'objet `Statement`
- ☐ En spécifiant `ResultSet.TYPE_SCROLL_INSENSITIVE` et `ResultSet.CONCUR_READONLY` lors de la création de l'objet `Statement`

Pour avoir ce fonctionnement, vous devrez créer votre `Statement` de cette façon :

```
Statement state = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,  
ResultSet.CONCUR_UPDATABLE);
```

`ResultSet.TYPE_SCROLL_SENSITIVE` : permet de parcourir les résultats dans les deux sens mais aussi d'avoir les modifications apportées en base directement. `ResultSet.CONCUR_UPDATABLE` : permet la modification des données lues.

Question 6

Quels sont les deux grands intérêts des requêtes préparées ?

- ☐ Elles sont plus rapides et plus simples d'utilisation.
- ✓ ☒ Elles sont plus rapides et acceptent des requêtes à trous que vous pouvez manipuler.
- ☐ Elles sont plus simples et acceptent des requêtes à trous que vous pouvez manipuler.
- ☐ Elles n'ont rien de particulier, ce sont d'anciens objets Java.

Les requêtes préparées ont une première particularité : elles sont précompilées, ce qui leur permet d'être un peu plus rapides que leurs homologues non préparées. Elles peuvent aussi contenir des requêtes à trous, symbolisés par le caractère « ? ». Ces trous sont ensuite manipulables afin d'y insérer des valeurs souhaitées.

Question 7

Comment s'assurer que le mécanisme de connexion à une base de données ne s'effectue qu'une seule fois dans un programme ?

- ☐ En utilisant un constructeur synchronisé.
- ✓ ☒ En utilisant le pattern singleton.
- ☐ En utilisant un thread pour créer votre connexion.

- ☐ En vous connectant dans l'EDT.

Afin de vous assurer que votre connexion entre Java et la base de données ne s'effectue qu'une fois (je vous rappelle que le mécanisme de connexion est assez long et coûteux en ressources), il est de bon ton d'utiliser le pattern singleton.

Question 8

Qu'est-ce qui permet de s'assurer que la connexion est instanciée au lancement du programme ?

- ☐ En instanciant la connexion depuis la méthode main().
- ☐ En instanciant la connexion depuis un constructeur privé.
- ☐ En instanciant la connexion depuis un objet prévu à cet effet.
- ✓ ☒ En instanciant la connexion depuis la déclaration de votre classe de connexion.

En instanciant la connexion depuis la déclaration de votre classe de connexion, l'effet sera de laisser le soin à la JVM de créer votre instance dès le chargement du programme en mémoire. Ceci vous assure que deux threads différents ne seront pas en concurrence pour la création de votre instance.

Question 9

À quoi sert le pattern DAO ?

- ✓ ☒ À faire le lien entre les tables de la base de données et une représentation objet de ces tables (et vice-versa).
- ☐ À permettre une meilleure performance de votre application.
- ☐ À lire dans votre base de données.
- ☐ À gérer la création d'entrées dans votre base de données.

Le pattern DAO permet de faire l'interface entre les données stockées dans votre base de données et des objets Java représentant ces données. Il permet donc de manipuler les tables de votre base de données depuis des objets Java.

Question 10

Quel est le caractère que vous devez utiliser dans une requête préparée pour symboliser les futures données que vous allez utiliser ?

- ✓ ☒ ?
- ☐ #
- ☐ ~

- !
- ^
- \$

Dans une requête préparée, les futures données que vous allez renseigner afin de compléter la requête sont symbolisées par des “?”.

◀ **LIEZ VOS TABLES AVEC DES OBJETS JAVA :
LE PATTERN DAO**

**ACTIVITÉ : RÉALISEZ UN GARAGE AVEC UNE
BASE DE DONNÉES EMBARQUÉE** ▶

Le professeur

Cyrille Herby

Spécialiste en développement Java et curieux insatiable d'informatique et de programmation web.
Actuellement auditeur en sécurité.

Découvrez aussi ce cours en...



Livre



PDF

OpenClassrooms

L'entreprise

Alternance

Forum

Blog

Nous rejoindre

Entreprises

Employeurs

En plus

Devenez mentor

Aide et FAQ

Conditions Générales d'Utilisation

Politique de Protection des Données Personnelles

Nous contacter

 Français 

