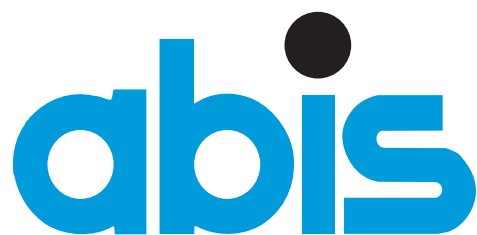


Auto-évaluation La programmation database en JDBC

Document: f1216test.fm

3 septembre 2019

ABIS Training & Consulting
Diestsevest 32 / 4b
B-3000 Leuven
Belgium

The logo for ABIS Training & Consulting. The word "abis" is written in a bold, blue, lowercase sans-serif font. A solid black circle is positioned above the letter 'i'. Below the text is a thick, horizontal grey bar.

TRAINING & CONSULTING

INTRODUCTION AUTO-ÉVALUATION LA PROGRAMMATION DATABASE EN JDBC

Ce test est basé sur des sujets traités dans le cours ABIS [La programmation data base avec JDBC](#).

Ce test comporte des questions à choix multiples. Pour certaines questions, plusieurs réponses correctes sont possibles. Cela sera alors clairement indiqué. Cochez les solutions qui vous semblent bonnes, et comparez vos réponses avec les solutions.

Ce test ne contient pas de question piège (du moins pas consciemment). Une réponse est considérée comme correcte si toutes les réponses correctes sont données. Comptez environ 15 minutes pour effectuer ce test.

Les réponses correctes sont disponibles à la fin de ce document.

QUESTIONS AUTO-ÉVALUATION LA PROGRAMMATION DATABASE EN JDBC

1. Quelles affirmations sont vraies à propos de JDBC? (2 réponses)
 - [a] JDBC est un API pour se connecter à des sources de données relationnelles, objets ou XML.
 - [b] JDBC vient de Java DataBase Connectivity.
 - [c] JDBC est un API qui permet d'accéder à des bases de données relationnelles, des feuilles de tableurs et des fichiers en format texte.
 - [d] JDBC est un API pour contourner la discordance objet-relationnel entre les programmes OO et les bases de données relationnelles.

2. Quels packages contiennent les classes JDBC?
 - (a) java.jdbc et javax.jdbc
 - (b) java.jdbc et java.jdbc.sql
 - (c) java.sql et javax.sql
 - (d) java.rdb et javax.rdb

3. Quel type de driver convertit les appels JDBC dans le protocole réseau qui peut-être directement compris par un système de gestion de base de données?
 - (a) Type 1 driver
 - (b) Type 2 driver
 - (c) Type 3 driver
 - (d) Type 4 driver

4. Quel type de Statement peut exécuter une requête paramétrable?
 - (a) PreparedStatement
 - (b) ParameterizedStatement
 - (c) ParameterizedStatement et CallableStatement
 - (d) Toute les sortes de Statement (i.e. sous interfaces de Statement)

5. Comment peut-on voir les données qui sont dans un ResultSet?
- (a) En appelant la méthode `get(..., String type)` sur le ResultSet, où type sera le type de la base de données
 - (b) En appelant la méthode `get(..., Type type)` sur le ResultSet, où Type est un objet qui représente un type de la base de données
 - (c) En appelant la méthode `getValue (...)`, et en convertissant le résultat vers le type Java souhaité
 - (d) En appelant les méthodes spéciales "getter" sur le ResultSet: `getString(...)`, `getBoolean(...)`, `getClob(...)`,...
6. Comment peut-on exécuter des requêtes DML (p.e. insert, delete, update) dans une base de données?
- (a) En utilisant les classes `InsertStatement`, `DeleteStatement` ou `UpdateStatement`
 - (b) En appelant les méthodes `execute(...)` ou `executeUpdate(...)` depuis un objet `Statement` quelconque ou un objet d'un sous-interface de cette interface
 - (c) En appelant les méthodes `executeInsert(...)`, `executeDelete(...)` ou `executeUpdate(...)` depuis un objet `DataModificationStatement`.
 - (d) En utilisant la méthode `execute(...)` depuis un objet `DataModificationStatement`
7. Comment peut-on savoir, dans un programme Java, que l'exécution d'une requête dans une base de données a renvoyé un SQL Warning?
- (a) Il faut gérer l'exception `SQLException` qui est activée par la méthode qui a exécuté la requête
 - (b) Il faut capter l'exception non vérifiée `SQLWarningException` qui est générée par la méthode qui a exécuté l'instruction
 - (c) Il faut appeler la méthode `getWarnings()` sur l'objet `Statement` (ou une sous interface de celui-ci)
 - (d) Il faut questionner l'objet `ResultSet` à propos des warnings susceptibles d'avoir été générés
8. Qu'est-ce, en termes JDBC, qu'un DataSource?
- (a) Une `DataSource` est un service de base pour gérer un ensemble de drivers JDBC
 - (b) Une `DataSource` est une représentation Java d'une source de données physique
 - (c) Une `DataSource` est un "registry point" pour les JNDI-services
 - (d) Une `DataSource` est une "factory" pour les connexions vers des sources de données physiques

9. Quelle est la signification du `ResultSet.TYPE_SCROLL_INSENSITIVE`
- (a) Cela signifie que ce `ResultSet` n'est pas sensible au défilement
 - (b) Cela signifie que ce `ResultSet` est sensible au défilement, mais pas sensible aux mis-à-jours et qu'il n'est donc pas modifiable
 - (c) Cela signifie que le `ResultSet` est sensible au défilement mais qu'il est insensible aux changements faits par d'autres
 - (d) La signification dépend de la source de données, ainsi que du type et de la version du driver utilisé avec la source de données
10. Les `ResultSet`s sont-ils modifiables?
- (a) Oui, mais seulement si la méthode `openCursor()` est appelée sur le `ResultSet`, et si le driver supporte cette option
 - (b) Oui, mais seulement l'on indique la stratégie de concurrence au moment où l'on exécute l'instruction, et si le driver supporte cette option
 - (c) Oui, mais seulement si le `ResultSet` est un objet de la classe `UpdateableResultSet`, et si le driver supporte cette option
 - (d) Non, les `ResultSet`s ne sont jamais modifiables. Il faut explicitement exécuter des instructions DML (`insert`, `delete` et `update`) afin de modifier des données dans la base de données concernée
11. Quelles affirmations à propos des transactions JDBC sont-elles vraies? (2 réponses correctes)
- [a] Une transaction est un ensemble d'instruction exécutées avec succès dans une base de données
 - [b] Une transaction est terminée si `commit()` ou `rollback()` sont appelés sur l'objet `Connection`
 - [c] Une transaction est terminée si `commit()` ou `rollback()` sont appelés sur l'objet `Transaction`
 - [d] Une transaction est terminée si `close()` est appelée sur un objet `Connection`.
12. Comment est-il possible de démarrer une transaction dans une base de données?
- (a) En demandant un objet `Transaction` à la `Connection`, et en appelant la méthode `begin()`
 - (b) En demandant un objet `Transaction` à la `Connection`, et en mettant la propriété `autoCommit` de l'objet `Transaction` sur `faux`
 - (c) En appelant la méthode `beginTransaction()` sur l'objet `Connection`
 - (d) En mettant la propriété `autoCommit` de la connexion sur la position `faux` et en exécutant une instruction dans la base de données

13. Quelle est la signification du niveau d'isolation d'une transaction `TRANSACTION_REPEATABLE_READ`
- (a) Les dirty reads, non-repeatable reads et phantom reads peuvent avoir lieu
 - (b) Les dirty reads sont évités; non-repeatable reads et phantom reads peuvent avoir lieu
 - (c) Les dirty reads et non-repeatable reads sont évités; les phantom reads peuvent avoir lieu
 - (d) Les dirty reads, non-repeatable reads et phantom reads sont évités
14. Comment utiliser un 'savepoint'?
- (a) Un 'savepoint' est réalisé en appelant la méthode `setAutoCommit(true)` sur la connexion.
 - (b) Un 'savepoint' est activé en appelant la méthode `setSavePoint("mysavepoint")` sur la transaction
 - (c) Un 'savepoint' est utilisé pour marquer des points intermédiaires dans une transaction, afin d'obtenir un meilleur contrôle. Transactions peuvent être annulées (rollback) vers un 'savepoint' précédent sans influence sur les étapes précédentes.
 - (d) Un savepoint déclenche une synchronisation automatique avec la base de données.
15. Quelles affirmations sont vraies en parlant des updates positionnés (curseur update) dans les ResultSets? (2 réponses)
- [a] La technique du curseur est à l'heure actuelle la seule utilisable pour modifier les données de la ligne courante dans un ResultSet
 - [b] Les instructions insert sont supportées que avec les "scrollable" curseurs
 - [c] Seulement les "scrollable updateable ResultSets" peuvent utiliser cette technique pour modifier les données d'une ligne courante dans un ResultSet
 - [d] Le nom du curseur est spécifié par la méthode `setCursorName(String name)` dans l'objet Statement.
16. Comment est-il possible d'exécuter des procédures stockées dans une bases de données?
- (a) En appelant la méthode `execute()` sur un objet CallableStatement
 - (b) En appelant la méthode `executeProcedure()` sur un objet Statement
 - (c) En appelant la méthode `execute()` sur un objet StoredProcedure
 - (d) En appelant la méthode `run()` sur un objet ProcedureCommand

17. Que se passe-t-il si l'on appelle la méthode `close()` sur un objet `ResultSet`?
- (a) la méthode `close()` n'existe pas pour un `ResultSet`. Seulement les connexions peuvent être fermées
 - (b) les ressources de la base de données et de JDBC sont libérées
 - (c) on reçoit une exception `SQLException`, et cela parce que un `ResultSet` peut seulement être fermé par des objets `Statement`
 - (d) le `ResultSet`, en même temps que le `Statement` qui a créé le `ResultSet` ainsi que la connexion à partir de laquelle le `Statement` avait été créé vont être fermés et toutes les ressources de la base de données et de JDBC vont être libérées
18. Que se passe-t-il si l'on appelle `deleteRow()` sur un objet `ResultSet`?
- (a) La ligne sur laquelle on est positionné est détruite dans le `ResultSet`, mais pas celle de la base de données elle-même
 - (b) La ligne sur laquelle on est positionné est détruite du `ResultSet` ainsi que dans la base de données
 - (c) Le résultat dépend de la propriété `synchronizeWithDataSource` qui peut être définie à vrai ou faux
 - (d) On reçoit une erreur de compilation: la méthode n'existe pas puisqu'il est impossible de détruire des lignes d'un `ResultSet`
19. Quelles propositions à propos d'insert et update en batch sont elles correctes? (2 réponses)
- [a] Pour faire un batch d'instructions insert et delete, il faut faire un objet de type `Batch`, et appeler la méthode `addStatement(String statement)` pour chaque instruction à exécuter en batch
 - [b] Des insert et update en batch sont uniquement possible lorsque l'on utilise des requêtes paramétrées.
 - [c] Pour exécuter des update/insert en batch, il faut appeler `addBatch(String statement)` pour chaque objet pour chaque instruction que l'on veut exécuter
 - [d] Pour exécuter une update/insert en batch, il faut appeler la méthode `executeBatch()` sur un objet `Statement`
20. Quelles propositions sont vraies à propos des instructions DDL (create, grant,...)?
- (a) Les instructions DDL doivent être traitées comme des instructions SQL normales et sont exécutées en lançant la méthode DDL `execute()` sur un `Statement` objet (ou une sous-interface de celle-ci)
 - (b) Pour pouvoir exécuter des instructions DDL, il faut installer les fichiers de support additionnels
 - (c) Les instructions DDL ne peuvent pas être exécutées à l'aide de JDBC, il faut utiliser pour cela les outils natifs de la base de données
 - (d) Le support des instructions DDL sera implémenté dans les releases futures de JDBC

EVALUATION.

Ici sont les réponses correctes:

1. b c
2. c
3. d
4. a
5. d
6. b
7. c
8. d
9. c
10. b
11. b d
12. d
13. c
14. c
15. b d
16. a
17. b
18. b
19. c d
20. a

Comptez 1 point par bonne réponse. Pour les questions avec plusieurs bonnes réponses, comptez 1 point seulement si vous avez coché toutes les bonnes alternatives.

Si votre score atteint 80% ou plus, il ne vous est pas nécessaire de suivre ce cours [La programmation data base avec JDBC](#).

Si votre score est entre 50% et 80%, ce cours [La programmation data base avec JDBC](#) vous permettra de compléter vos connaissances.

Si votre score est inférieur à 50%, il vous est vivement conseillé de suivre ce cours [La programmation data base avec JDBC](#).