

---

## Série 3

Testez! Utilisez! Soyez curieux...

Nom de la BD (SID) :ens

Nom de la machine : ne-ege-letto.ig.he-arc.ch

### Exercice 1 et 2– OLE DB et ODBC

Veillez configurer OLEDB et/ou ODBC sur votre machine pour accéder à la base de données SOUK. Testez via OLEDB depuis Excel.

### Exercice 3 – JDBC

A l'aide de Netbeans, veuillez créer une classe « main » qui se connecte à votre compte Oracle (ens) et qui affiche des informations concernant la base de données :

- Nom du fournisseur de la BD
- Version de la BD (11.1 par exemple)
- Nom du driver
- Nom de l'utilisateur connecté
- URL de la BD

Exemple

- Oracle
- 11.1
- Oracle JDBC driver
- PROF\_BAUDET
- jdbc:oracle:thin:@ne-ege-letto.ig.he-arc.ch:1521:ens

## Exercice 4 – Statement et PreparedStatement

A l'aide de Netbeans, veuillez créer une classe « main » qui se connecte à votre compte Oracle (ens) et qui exécute des instructions SQL.

- Veuillez tester des SELECT, INSERT, UPDATE et DELETE.
- Veuillez exécuter des SELECT en mode « static »
- Veuillez exécuter des SELECT en mode « dynamique » (essayer de changer la valeur des paramètres)

Conseils :

- lors de l'exécution d'une requête SQL, veuillez afficher le nom de la classe du curseur : `System.out.println(myCursor.getClass());` Nous verrons lors du prochain cours comment manipuler ces données à l'aide de curseurs.
- Lors de l'exécution d'un INSERT, UPDATE ou DELETE, veuillez vérifier le résultat depuis SQLDeveloper après avoir appelé la méthode `commit()` sur la connexion.

## Exercice 5 – Curseur « statique en avant seulement »

A l'aide de Netbeans, veuillez créer une classe « main » qui se connecte à votre compte Oracle (ens) et qui affiche la liste des élèves d'une classe.

## Exercice 6 – Curseur navigable

A l'aide de Netbeans, veuillez créer une classe « main » qui se connecte à votre compte Oracle (ens) et qui affiche la liste des élèves d'une classe. Veuillez cette fois parcourir votre curseur de différentes manières.

## Exercice 7 – Curseur modifiable

A l'aide de Netbeans, veuillez créer une classe « main » qui se connecte à votre compte Oracle (ens) et qui affiche la liste des élèves d'une classe. Veuillez effectuer des mises à jour sur votre curseur puis de nouveau afficher la liste des élèves d'une classe.

Tester des mises à jour, des insertions et des suppressions de données.

Vérifiez les données dans votre base de données à l'aide de SQL Developer.

## Exercice 8 – RowSet

Veuillez tester les fonctionnalités principales des RowSet : connecté, déconnecté et ses propriétés de JavaBean...

Pensez à la gestion des conflits en mode déconnecté !

## Exercice 9 – Transaction et verrous

Veuillez modifier votre exercice 7 afin de gérer correctement les transactions et les verrous.

## Exercice 10 – Classe OracleConnection

Afin de rendre votre code plus réutilisable, veuillez créer une classe de connexion « OracleConnection ».

Contrat de classe :

- Permettre la connexion à la BD
- Permettre la déconnexion à la BD

Veuillez tester le bon fonctionnement de cette nouvelle classe à l'aide d'un des précédents exercices.

## Exercice 11 – Pool de connexions manuel

Réfléchissez à comment développer un pool de connexion manuellement. Vous pouvez créer un diagramme de classe pour expliquer le concept. Du pseudo code peut aussi être intéressant.

Je ne demande pas un pool de connexions très performant. Il ne s'agit que de voir le concept global → concept académique...

**Attention** : Veuillez me faire parvenir votre diagramme de classe (avec des explications si nécessaire) et / ou votre pseudo code en format PDF avant le cours de la semaine prochaine à l'adresse [cedric.baudet@he-arc.ch](mailto:cedric.baudet@he-arc.ch).

## Exercice 12 – DataSource et pool de connexion

Veuillez utiliser le pool de connexion DBCP de Apache et tester 2 fonctionnalités (vous pouvez réutiliser un ancien exercice et l'adapter) :

- Test de l'utilisation d'une OracleDataSource. N'utilisez pas encore de pool de connexion mais seulement une DataSource.
- Test de l'utilisation d'une BasicDataSource du package `org.apache.commons.dbcp`. N'utilisez pas encore de pool de connexion mais seulement une DataSource.
  - Conseil... Pensez à utiliser la librairie commons pool pour résoudre les conflits (<http://commons.apache.org/pool>)
- Test du pool de connexion de DBCP (`PoolingDataSource` du package `org.apache.commons.dbcp`)

Documents de référence et éléments à downloader sur :

- [http://www.oracle.com/technology/docs/tech/java/sqlj\\_jdbc/doc\\_library/javadoc/oracle.jdbc.pool.OracleDataSource.html](http://www.oracle.com/technology/docs/tech/java/sqlj_jdbc/doc_library/javadoc/oracle.jdbc.pool.OracleDataSource.html)
- <http://commons.apache.org/dbcp/>

Ceux qui ont un peu d'avance ou de temps peuvent tester le pool de connexion Oracle (`OracleConnectionPoolDataSource`).

## Exercice 13 – Améliorations finales

Veuillez réutiliser un exercice précédent et y intégrer les différents éléments vus en classe dans le chapitre « Améliorations et limites de JDBC » (pas SQLj... sauf si vous avez du temps !).