

Guide Pratique Oracle et MongoDB

Ecole Centrale Paris - Cours électif IS1230

Oracle Express & SQLDeveloper & MongoDB
Installation, Connexion, Mise en pratique

version 1.0
6 septembre 2015

Nicolas Travers

TABLE DES MATIÈRES

1	Installation	3
1.1	Oracle Express	3
1.2	SQLDeveloper	3
2	SQLDeveloper - Connexion	4
2.1	Lancement de l'application	4
2.2	Connexion : serveur local	4
2.2.1	Connexion administrateur système	5
2.2.2	Création de l'espace de Travaux Pratiques - TEST	5
2.2.3	Connexion au compte test_USER	5
2.3	Connexion au serveur de l'Ecole Centrale Paris	6
3	Mise en pratique	7
3.1	DataModeler	7
3.2	Exécuter des scripts SQL	8
3.3	Stocker une procédure, une fonction ou un déclencheur	8
3.3.1	Créer une procédure	8
3.3.2	Créer une fonction	8
3.3.3	Créer un Trigger	8
4	MongoDB	9
4.1	Installation	9
4.1.1	Windows	9
4.1.2	Linux et MacOS	9
4.2	Console Mongo	10
4.2.1	Gestion de collections	10
4.2.2	API MongoDB	11
4.2.3	Map/Reduce	11

Dans le cadre du cours IS1230 d'introduction aux bases de données, l'installation de Oracle Express n'est pas obligatoire. En effet, un serveur Oracle est déjà installée à l'Ecole Centrale Paris et une simple connexion à celui-ci suffit pour effectuer l'ensemble des travaux pratiques requis.

Toutefois, si vous souhaitez avoir une utilisation locale sur votre ordinateur personnel, vous pourrez suivre les instructions ci-dessous pour faire son installation.

Si vous n'avez pas accès aux installateurs sous Claroline vous pouvez les télécharger sur le site de Oracle. Toutefois, la création d'un compte est nécessaire (aucun frais).

1.1 Oracle Express

Oracle Express© est une solution logicielle pour installer un SGBD Oracle simplifié et libre d'accès. Vous pourrez y faire l'ensemble des travaux pratiques de IS1230.

L'installation de Oracle Express est assez simple pour les environnements Windows et Linux. Il vous suffit de télécharger les installateurs déposés sur claroline (ou lien direct ¹).

- **Windows** Décompresser le fichier, aller dans *Disk1*, exécuter *setup.exe*
Suivre les instructions par défaut. **!!! Bien noter le mot de passe "SYSTEM" !!!** (utile pour la connexion et l'administration)
- **Linux** Installer le RPM et **!!! Bien noter le mot de passe "SYSTEM" !!!**
- **MacOS** Pas d'installateur disponible. Pour contourner le problème, une machine virtuelle est nécessaire :
Télécharger et installer **VirtualBox** de Oracle, suivre les instructions par défaut.
Télécharger la machine virtuelle sous Linux "*OTN_Developer_Day_VM.ova*"² qui contient un serveur Oracle Express et SQLDeveloper. Attention, 25 Go sont nécessaires pour tout installer.
Le login / mot de passe système par défaut "oracle" et "oracle".

1.2 SQLDeveloper

SQLDeveloper est une solution logicielle Oracle pour vous permettre de manipuler la base de données, aussi bien avec la modélisation, la création d'une base de données, les requêtes, les procédures stockées et les déclencheurs.

Télécharger l'installateur sur Claroline (ou lien direct ³). Il est disponible sur toutes les systèmes d'exploitation standards. Décompresser le fichier et c'est installer. Idéalement, il faut déplacer le répertoire créé à l'endroit les programmes sont placés (*Program Files* ou */etc/local*). En effet, l'exécutable disponible dans le répertoire SQLDeveloper est un script de lancement d'application Java (une JVM est nécessaire dans votre système d'exploitation, mais c'est une installation par défaut).

1. <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

2. <http://www.oracle.com/technetwork/database/enterprise-edition/databaseappdev-vm-161299.html>

3. <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

2 SQLDEVELOPER - CONNEXION

2.1 Lancement de l'application

Lancer l'exécutable dans le répertoire "SQLDeveloper", une fenêtre de progression s'ouvre :



FIGURE 2.1 – Lancement de SQLDeveloper

Ne pas importer les préférences :

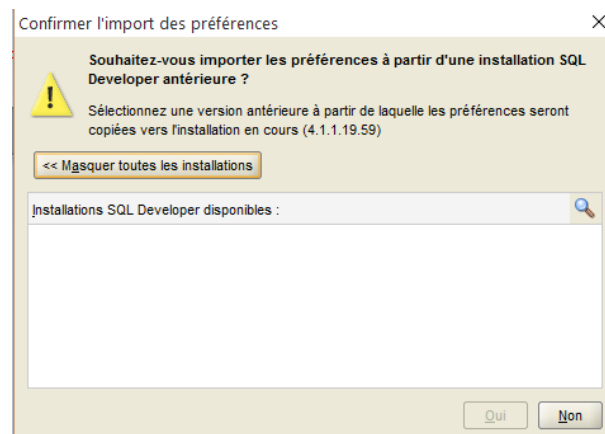



FIGURE 2.2 – Ne pas importer les préférences

Une fois l'interface utilisateur ouverte, il est nécessaire de créer une première connexion.

2.2 Connexion : serveur local

Les instructions ci-dessous ne concernent qu'une connexion à un serveur local dont vous avez le mot de passe administrateur. Si vous utilisez le serveur de Centrale, vous pouvez passer à la section suivante.

2.2.1 Connexion administrateur système

Pour créer une nouvelle connexion, le symbole  (en haut à gauche) permet d'en créer une nouvelle. Remplissez le formulaire de connexion avec les instructions de connexions suivantes :

- **Nom de connexion** : locale (Nom affiché pour la connexion)
- **Nom de l'utilisateur** : SYSTEM (login administrateur) - *oracle* (si VM VirtualBox)
- **Mot de passe** : ... (celui que vous avez donné lors de l'installation) - *oracle* (si VM VirtualBox)
- **Nom de l'hôte** : localhost (serveur local)
- **Port** : 1521 (port d'écoute de Oracle Express)
- **SID** : xe (Identifiant de base oracle par défaut pour Oracle Express)

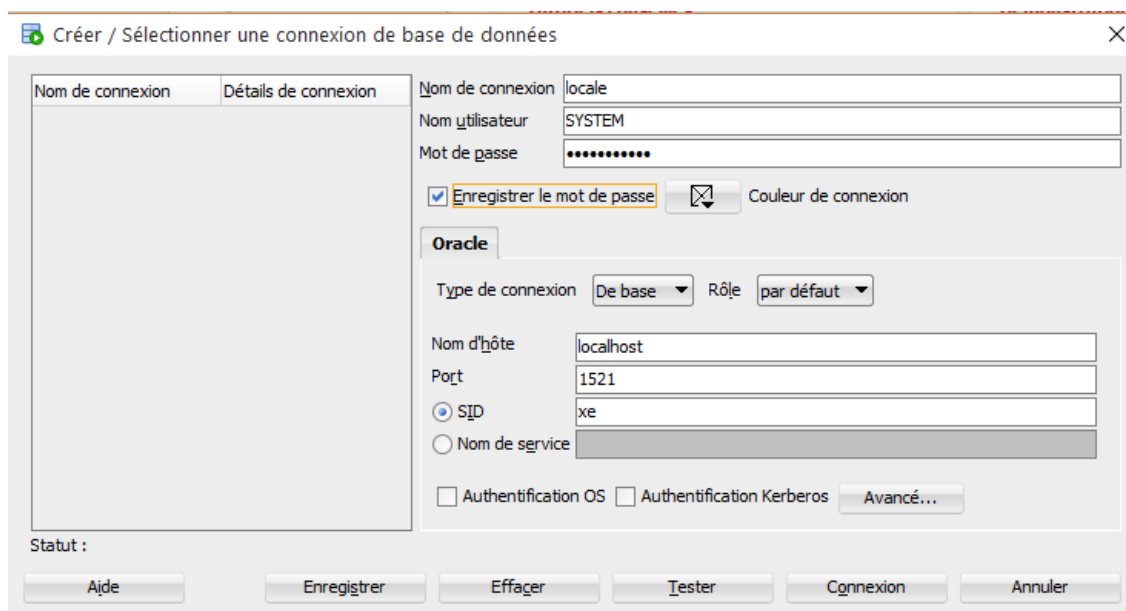



FIGURE 2.3 – Formulaire de création de connexion

2.2.2 Création de l'espace de Travaux Pratiques - TEST

Télécharger le script 'creation.sql' sous claroline pour créer l'espace de travail qui sera utilisé pour les travaux pratiques. Pour exécuter le script, aller dans le menu Fichier/Ouvrir, un onglet s'ouvre, cliquer sur le bouton 'Exécuter un script'  (ou touche F5). L'ensemble des requêtes SQL sont exécutées, en particulier la création d'un compte "test_USER".


2.2.3 Connexion au compte test_USER

Suivre les mêmes instructions que la connexion SYSTEM avec les informations suivantes :

- **Nom de connexion** : test
- **Nom de l'utilisateur** : test_USER
- **Mot de passe** : test

Déconnecter la connexion “locale” pour faire en sorte que l’ensemble des requêtes effectuées soient bien sur “test” et non “locale”. Pour ce faire, clique droit sur la connexion “locale” (à gauche) et “déconnecter”.

2.3 Connexion au serveur de l’Ecole Centrale Paris

Pour créer une nouvelle connexion, le symbole  (en haut à gauche) permet d’en créer une nouvelle. Remplissez le formulaire de connexion avec les instructions de connexions suivantes :




- **Nom de connexion** : IS1230 (Nom affiché pour la connexion)
- **Nom de l’utilisateur** : <votre login centrale>
- **Mot de passe** : ...
- **Nom de l’hôte** : dev-ora11g.cti.ecp.fr (serveur ECP)
- **Port** : 1521 (port d’écoute du serveur Oracle)
- **SID** : BDTP (Identifiant de base oracle utilisé)

3.1 DataModeler

Afin de modéliser la base de données, il faut utiliser l'outil DataModeler :

- Télécharger le fichier “test_export.zip” sur claroline et le décompresser ;
- Sous SQLDeveloper, menu “Fichier/DataModeler/Ouvrir” ;
- Chercher le fichier “test_export.dmd” qui a été décompressé précédemment ;
- Ne cocher que “Relationnel” (pas “physique”) ;

La fenêtre de modélisation s'ouvre alors avec une première relation “Etudiant” à partir de laquelle vous allez pouvoir modéliser votre base de données. Les commandes qui vous seront utiles sont :

- **Nouvelle relation**  (Définir le nom, chaque attribut, la clé primaire)
- **Clé étrangère**  (relier deux relations, à partir de la clé primaire de la table destination)
- **Créer les requêtes de création (DDL)** . Une fenêtre intégrée apparaît :
 - À faire une fois que le modèle est terminé ;
 - Cliquer sur “Générer” (en haut de la fenêtre intégrée) ;
 - Décocher toutes les informations de l'arborescence comme dans l'image ci-dessous ;

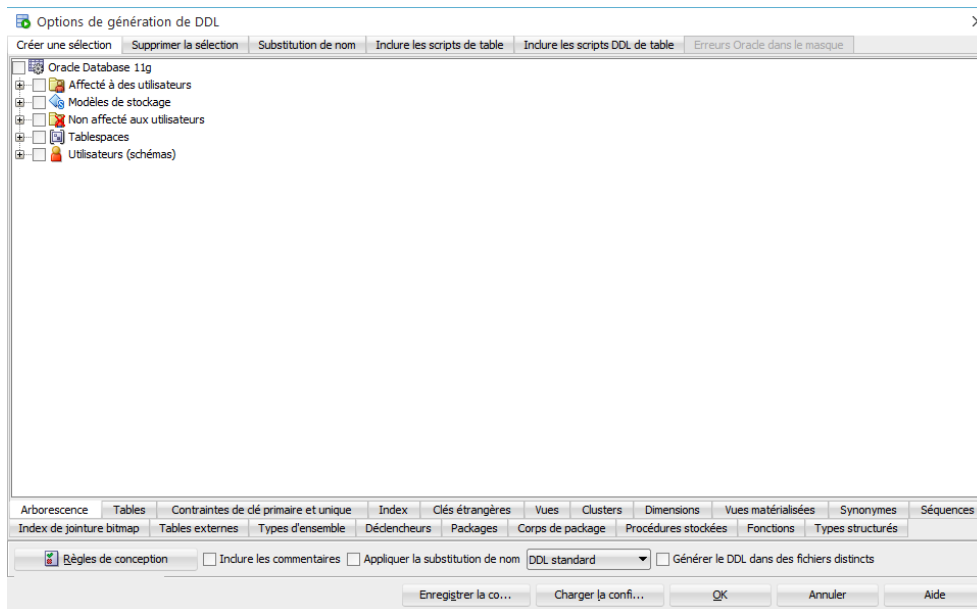



FIGURE 3.1 – Génération des scripts DDL de la base de données

- Dans l'onglet “Tables” (en bas de la fenêtre intégrée), cocher l'ensemble des tables ;
- Cliquer sur “OK” ;
- Les requêtes SQL générées peuvent copiées (ou enregistrées) pour être exécutées dans la console SQLDeveloper pour créer la base de données.

3.2 Exécuter des scripts SQL

L'exécution de requêtes SQL, procédures stockées, triggers est très simple. Pour cela, il faut lancer une console SQL  (ou utilisée une déjà ouverte), tout en spécifiant quelle connexion utiliser pour cela (**Utiliser la connexion "test" ou "IS1230" pour que cela reste sur votre espace**).

Une fois sur la console, la requête écrite, cliquer sur 'F5' vous permettra d'exécuter l'ensemble des requêtes présentes dans la console (Ctrl+Entrée pour exécuter la requête sélectionnée par le curseur de la souris). Le résultat s'affiche en dessous de la console.

3.3 Stocker une procédure, une fonction ou un déclencheur

Pour PL/SQL et les Triggers, des dossiers correspondants sont disponibles dans la connexion à la base de données (arborescence à gauche).

3.3.1 Créer une procédure

- Cliquez droit sur le dossier "Procédures" et cliquez sur "Nouvelle procédure"
- Donner un nom à celle-ci. Ce sera le nom utilisé à chaque instruction "call"
- Ajouter les paramètres d'entrée : nom et type de données (mode IN par défaut)
- Modifier le script pour y intégrer votre code de procédure
- Cliquez sur le bouton "Compiler" (roue crantée), cela enregistre la procédure

3.3.2 Créer une fonction

- Cliquez droit sur le dossier "Fonction" et cliquez sur "Nouvelle fonction"
- Donner un nom à celle-ci. Ce sera le nom utilisé dans les requêtes SQL
- Préciser le type de valeur renvoyée par la fonction
- Ajouter les paramètres d'entrée : nom et type de données (mode IN par défaut)
- Modifier le script pour y intégrer votre code de fonction
- Cliquez sur le bouton "Compiler" (roue crantée), cela enregistre la procédure

3.3.3 Créer un Trigger

- Cliquez droit sur le dossier "Déclencheur" et cliquez sur "Nouveau déclencheur"
- Donner un nom à celle-ci
- Préciser la table ciblée : Object de base
- Choisir quand le trigger agit : BEFORE / AFTER
- Choisir le/les types d'événement traités (mettre dans la colonne de droite)
- Modifier le script pour y intégrer votre code de trigger
- Cliquez sur le bouton "Compiler" (roue crantée), cela enregistre la procédure

Le logiciel MongoDB¹ peut être installé et utilisé dans un environnement local ou distribué. Il est disponible sur claroline, et également en ligne : <http://www.mongodb.org/downloads/>

Le Travaux Pratique associé à ce guide est dédié à une utilisation en mode Console pour bien intégrer les possibilités de l'API mongoDB et de MapReduce.

Sur Claroline est également disponible 2 documents PDF aidant à l'utilisation de MongoDB :

- MongoDB In Action
- MongoDB in Praticce

4.1 Installation

Téléchargez et installer le logiciel MongoDB² ;

4.1.1 Windows

Ouvrir un PREMIER shell cmd

```
> wmic os get osarchitecture           (pour connaître l'architecture de son
                                       système d'exploitation (32 ou 64 bits))
```

Extraire l'archive dans « C:\ » par exemple

```
> cd /
> md data                               (crée un répertoire data)
> md data\db                             (crée un répertoire db)
```

Lancement du Serveur : (ne pas fermer la fenêtre)

```
> C: \mongodb\bin\mongod.exe
```

Ou

```
> C:\mongodb\bin\mongod.exe --dbpath "dossier data"
   (en spécifiant le dossier si c'est différent de c:\data\db)
```

Lancement de la console cliente dans un SECOND shell :

```
> C: \mongodb\bin\mongo.exe
```

4.1.2 Linux et MacOS

Dans une PREMIERE console :

```
> tar zxvf mongodb-xxx.tgz           (xxx:version choisie)
```

1. <http://www.mongodb.org/>

2. Documentation : <http://docs.mongodb.org/manual/contents/>

```
> mv mongodb-xxx /votredossier
> ln -s mongodb-xxx mongodb      (crée un lien symbolique
                                vers le dossier contenant mongodb)
> sudo mkdir /data/db           (par défaut, mongodb utilise ce dossier,
                                il faut donc le créer)
Lancement du Serveur : (ne pas fermer la fenêtre)

> bin/mongod                    (par défaut la base est /data/db)
Ou
> bin/mongod --dbpath /votreDossier (Dossier perso)

Lancement du client dans une SECONDE console :

> bin/mongo
```

4.2 Console Mongo

Dans la console client (voir section précédente), le langage de manipulation de données de MongoDB est inspiré de JavaScript avec des objets. Pour faciliter l'utilisation, il vous est possible de :

- Commandes précédente : *flèche vers le haut*
- Compléter automatiquement une instruction : *tabulation*
- Utiliser des variables

4.2.1 Gestion de collections

Création de la base :

Une base de données MongoDB est créée dès sa première utilisation avec la commande `use <nom de la base>`

Une base contient un ensemble de collections de documents.

Création d'une collection :

Les documents que l'on stocke, sont placés dans une collection que l'on crée à l'aide de l'instruction : `db.createCollection('<nom de la collection>')` ; Dans ce guide, nous utiliserons la collection 'publis' : `db.createCollection('publis')` ;

Toute opération sur cette collection devra alors commencer par : `db.publis.xxx`

Création d'un document :

Un document est un *JSON document* qui est encapsulé dans des accolades. Chaque attribut à une clé et une valeur. Une valeur peut être une chaîne de texte, un nombre, une liste de valeur ou un document JSON. Il est possible d'intégrer des listes de valeurs à l'aide d'un tableau avec des crochets. Voici un exemple de document :

```
{"type": "Book", "title": "Modern Database Systems: The Object Model, Interoperability, and Beyond.", "year": 1995, "publisher": "ACM Press and Addison-Wesley", "authors": ["Won Kim"], "source": "DBLP"}
```

L'insertion du document se fait à l'aide de l'instruction : `db.publis.save(<Document JSON>)` ; Il est possible également de sauvegarder un document dans une variable, et ensuite de l'utiliser : `w = {"type": "Book", "title": "Modern Database Systems: The Object Model, Interoperability, and Beyond.", "year": 1995, "publisher": "ACM Press and Addison-Wesley", "authors": ["Won`

```
Kim"], "source": "DBLP"}
db.publis.save(w);
```

Consulter le contenu d'une collection :

```
db.publis.find();
```

Seule les 20 premiers éléments sont affichés. Utilisez la commande `lt` pour afficher les résultats suivants.

Importer un grand nombre de documents :

```
mongoimport -host localhost:27017 -db DBLP -collection publis < dblp.json
```

Le chemin vers l'exécutable "mongoimport" est sans doute nécessaire, ou une variable d'environnement ferait l'affaire.

L'opération peut prendre quelques secondes

4.2.2 API MongoDB

Pour des recherches simples, il faut créer des "documents requêtes". Exemple de recherche de tous les documents ayant un attribut 'type' ayant pour valeur 'Book': `db.publis.find({"type": "Book"});`

Il est également possible de d'enlever les doublons d'un attribut donné: `db.publis.distinct("title");`

Afin de faire des traitements sur résultats (projection, tri, groupement), la fonction 'aggregate' est nécessaire. Une liste de paramètres doivent être traité, chaque paramètre est précédé d'un '\$'. Exemple :

```
db.publis.aggregate([
  {$match : {'type' : 'Book'}},
  {$group : { _id : "$publisher", number : { $sum : 1 } }},
  {$project : {number : 1}},
  {$sort : {number : -1}}
]);
```

Opérations disponibles :

- `$match` : recherche simple de l'API MongoDB
- `$project` : projection des attributs demandés pour le résultat
- `$group` : attribut à grouper sur valeur, l'opération d'agrégat peut préciser le nom de l'attribut et la fonction (`$sum`).
- `$sort` : tri les documents sur les valeurs de l'attribut spécifié
- `$unwind` : désimbrique une liste de valeurs

4.2.3 Map/Reduce

Pour faire une requête MapReduce, il faut :

- Définir le map: `var mapFunction = function () {if(this.type == "Book") emit(this.title, 1);};`
- Définir le reduce: `var reduceFunction = function (key, values) {return Array.sum(values);};`
- Définir les paramètres d'interrogation: `var queryParam = {query : {}, out : "result_set"}`
- Lancer la requête: `db.publis.mapReduce(mapFunction, reduceFunction, queryParam);`
- Consulter le résultat: `db.result_set.find();`