TD1 — BD et Ingénierie des SI

Master 1 MIAGE - année 2021-2022

Exercice 1 - Manipulation de Docker

Vous devrez lire les supports de cours pour pouvoir effectuer les opérations suivantes :

1. Récupérez et lancez un conteneur **httpd** version **2.4**. Il s'agit d'un serveur apache.

```
docker run -d --name="httpd" httpd:2.4
```

1. Ouvrez un navigateur et tapez l'adresse http://localhost:8080. Que constatez vous ? Expliquez le pourquoi de la chose.

Erreur "La connexion a échoué".
Sur le host (localhost), le port 8080 n'est associé à aucun service.

1. Cherchez l'adresse IP du conteneur (inspect)

```
172.17.0.x
```

1. Ouvrez le navigateur et saisissez l'adresse http://IP_DU_CONTENEUR où IP_DU_CONTENEUR est l'IP que vous avez trouvée à la question précédente. Que constatez vous ?

Cette fois, la connexion au serveur **httpd** dans le conteneur fonctionne.
On accède directement au conteneur via son IP, comme s'il s'agissait d'une machine distante, et dont le port 80 est ouvert (configuration par défaut pour **httpd**).

1. Stoppez le conteneur et vérifiez avec le navigateur que l'URL n'est plus accessible.

```
docker stop httpd
```

Le conteneur n'est plus accessible (puisqu'il est arrêté).

1. Supprimez le.

```
docker rm httpd
```

1. Pour des raisons de simplicité d'utilisation, il peut être intéressant de mapper les ports du conteneur sur localhost. Une façon simple de le faire est de le faire lors de la première exécution grâce à l'option **-p**. Mappez le port 80 du conteneur sur le port 8080 de localhost.

```
docker run -d --name="httpd" -p 8080:80 httpd:2.4
```

1. Après avoir lancé le conteneur vérifiez qu'effectivement que **http://localhost:8080** donne bien le résultat attendu.

La page d'accueil du serveur **httpd** (dans le conteneur) s'affiche.

1. Stoppez maintenant et supprimez de nouveau le conteneur créé

```
docker rm -f httpd
```

1. Une option intéressante proposée par docker permet de partager une partie de votre disque avec le conteneur lancé. Créez un dossier **monserveur** et relancez le conteneur en prenant soin de mapper votre répertoire courant (en utilisant le chemin absolu) avec le dossier **/usr/local/apache2/htdocs/** de votre conteneur. Ajoutez une page index.html dans votre dossier **monserveur** et vérifiez que le serveur affiche bien votre page.

```
mkdir ./monserveur
docker run -d --name="httpd" --publish 8080:80
--volume="$(pwd)/monserveur":/usr/local/apache2/htdocs httpd:2.4
echo "<html><body>Hello, World!</body></html>" > ./monserveur/index.html
```

En se connectant à `localhost:8080`, on voit bien la page d'accueil du serveur **httpd** qui est dans le conteneur.

1. Stoppez le conteneur et supprimez le.

```
docker stop httpd
docker rm httpd
```

Exercice 2 - une nuit au Musée

Vous devez modéliser le SI du nouveau muséum d'histoire naturelle d'Orléans, le MOBE. Ce musée possède une collection d'objets, qu'il enrichie chaque année en achetant de nouvelles oeuvres. Ces

oeuvres peuvent être exposées dans l'une des 20 salles du musée pendant une certaine période, sinon elles sont remisées dans la réserve jusqu'à la prochaine exposition.

Un exemple : le musée a acheté le 15 septembre 2015 un tyrannosaure, Rex, stocké au début dans la réserve ; le 1er janvier 2016, il a été exposé dans le grand hall du musée ; le 1er septembre 2016, Rex a été déplacé dans la grande salle 1, pour l'exposition "les dinosaures". le 15 mars 2017, il est revenu dans le grand hall. Du 20 décembre 2018 au 1 avril 2019, il est retourné en réserve pour être restauré. Il est à nouveau exposé dans la grande salle 1 depuis.

1. Pour la version 1 de votre SI, vous décidez d'utiliser MySQL ; donnez les tables permettant de modéliser les données, et les données de ces tables pour l'exemple du tyrannosaure.

```
OEUVRE(_id_, nomOeuvre, dateArrivee)
SALLE(_id_, nomSalle)
DEPLACEMENT(_id_, idOeuvre, idSalle, dateDeplacement)
```

1. Lancez une instance de mysql (nom du conteneur) tout en prenant soin de mapper le port 3306 du conteneur avec le port 3306 de localhost.

```
docker run -d --name=mysql -e MYSQL_ROOT_PASSWORD=foobar mysql:5.7
```

Au lancement, le choix d'un mot de passe pour le compte admin de MySQL est demandé.

1. Le but maintenant est de se rendre à l'intérieur du conteneur et de prendre le contrôle en bash.

```
docker exec -it mysql bash
```

1. Une fois à l'intérieur, lancez la commande **mysql --password**. Normalement, l'invite de commande vous demande le mot de passe que vous avez dû définir lors du lancement du conteneur.

Le prompt `mysql>` s'affiche.

1. Créez les tables et ajoutez les données correspondant au scénario.

```
CREATE DATABASE IF NOT EXISTS musee;

# liste des databases présentes dans le SI
SHOW DATABASES;

# choix d''une database pour les requêtes à suivre
USE musee;

# database actuellement activée pour les requêtes
SELECT DATABASE();

CREATE TABLE oeuvre (id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, nom VARCHAR(100),
dateArrivee DATE);
CREATE TABLE salle (id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, nom VARCHAR(100));
CREATE TABLE deplacement (id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, idOeuvre INT,
idSalle INT, dateDeplacement DATE, FOREIGN KEY (idOeuvre) REFERENCES oeuvre(id),
FOREIGN KEY (idSalle) REFERENCES salle(id));

# liste des tables présentes dans la database
SHOW TABLES;

INSERT INTO salle(nom) VALUES ('Grand hall');
INSERT INTO salle(nom) VALUES ('Grande salle 1');
INSERT INTO salle(nom) VALUES ('Reserve');

INSERT INTO oeuvre(nom, dateArrivee) VALUES ('T-Rex', '2015-09-15');

INSERT INTO deplacement(idOeuvre,idSalle,dateDeplacement) VALUES (1,3,'2015-09-15');
INSERT INTO deplacement(idOeuvre,idSalle,dateDeplacement) VALUES (1,2,'2016-01-01');
INSERT INTO deplacement(idOeuvre,idSalle,dateDeplacement) VALUES (1,1,'2016-09-01');

SELECT D.id, O.nom, S.nom, D.dateDeplacement
FROM oeuvre O
INNER JOIN deplacement D
   ON D.idOeuvre = O.id
INNER JOIN salle S
   ON S.id = D.idSalle
ORDER BY D.dateDeplacement DESC
;
```

Exercice 3 - Client Java

Le but de cet exercice est de faire un client Java permettant de requêter la base de données MySql grâce au connecteur MySql récupérable via la dépendance Maven suivante :

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.25</version>
</dependency>
```

Une fois ceci fait, vérifiez que vous pouvez accèder à la base de données en cliquant sur l'onglet **Database** dans la marge à droite. La procédure à suivre est la suivante :

Database → + → Data source → MySQL

Après avoir renseigné correctement les champs, cliquez sur le bouton **Test Connection**. Mettez à jour le connecteur si éventuellement IntellIJ vous le demande.

Nous supposons dans l'exemple ci-dessous que la base de données se nomme **musee** et qu'il existe une table **salles** composée de deux champs.

Créez un package **application** puis une classe **Connexion** à l'intérieur de ce dernier. Le programme ci-dessous permet d'afficher le contenu de la table **salles**.

```
package application;
import java.sql.*;
class Connexion{
public static void main(String args[]){
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection connexion=DriverManager.getConnection(
        "jdbc:mysql://localhost:3306/musee","root","root");
        //musee est ici le nom de notre base de données
        Statement stmt=con.createStatement();
        ResultSet rs=stmt.executeQuery("select * from salles");
        // stmt.executeAndUpdate("Insert ...") pour l'ajout de données
        while(rs.next())
        System.out.println(rs.getInt(1)+"  "+rs.getString(2));
        con.close();
    }
    catch(Exception e){ System.err.println(e);}
    }
}
```

Questions :

1. Faites un programme permettant d'afficher le détail de tous les mouvements d'oeuvres enregistrés dans le SI.
2. Un tricératops vient d'arriver aujourd'hui. Il est exposé dans le grand hall. Faites un programme permettant de mettre à jour les tables pour ajouter ces données.
3. Refactorez le code de telle façon à ce que l'on puisse offrir aux utilisateurs des possibilités

a. d’ajout de salles/oeuvres/déplacements
b. de récupération de l’ensemble des déplacements pour une oeuvre donnée
c. de récupérer toutes les oeuvres exposées dans une salle à une date donnée.

Exercice 4 - autre cas d’étude

Nous voulons mettre en place une application pour la saisie de notes à l’université. Les étudiants sont identifiés par leur numéro d’étudiant. Ils sont inscrits dans une filière pour une année universitaire donnée. Chaque filière est composée de deux semestres et chaque semestre est composé d’un certains nombre de modules. Un module a un libellé et un certain nombres d’ECTS.

- Nous voulons faire une application permettant aux responsables de modules d’envoyer les notes obtenues par les étudiants dans leurs modules.
- Lorsque toutes les notes sont saisies, nous voulons être capable d’extraire le PV de chaque semestre pour l’ensemble de la promotion
- Nous voulons pouvoir extraire l’ensemble des notes d’un étudiant donné

Questions :

1. Proposez un modèle relationnel permettant de capturer l’ensemble des données de notre cas d’étude.
2. Créez cette base de données sous MySQL
3. Développez une façade Java et un modèle en Java qui proposeront les fonctionnalités vues précédemment