



UFR DE SCIENCES ET TECHNIQUES  
UNIVERSITÉ DE CERGY-PONTOISE  
SITE DE SAINT-MARTIN

2, RUE ADOLPHE-CHAUVIN  
BP 222, PONTOISE 95011 CERGY-PONTOISE CEDEX.

L3 INFORMATIQUE

RAPPORT DE STAGE

---

**Sujet:**

**PENTAHO MONDRIAN AUX  
ENSEIGNEMENTS OLAP**

---

*Stagiaires:*

Zouleye BACHARD

Eric RUKUNDO

Mohammad Kabir ABDULSALAM

*Tuteurs de stage:*

Tianxiao LIU

JenTAO-YUAN

*Maître de stage:*

Dominique LAURENT

June 4, 2012

# Contents

<b>I</b>	<b>Introduction</b>	<b>3</b>
0.1	Le rôle de stage . . . . .	4
0.2	Présentation de l' Université de Cergy-Pontoise . . . . .	4
0.3	Le sujet . . . . .	5
<b>II</b>	<b>Développement</b>	<b>6</b>
0.4	Contexte du stage . . . . .	7
0.5	Le projet . . . . .	7
0.5.1	Présentation . . . . .	7
0.5.2	Etude du problématique . . . . .	7
0.5.3	Installations des Logiciels . . . . .	7
0.5.4	Utilisations des logiciels . . . . .	22
0.5.5	Interface graphique en java . . . . .	24
0.5.6	Manuel d'utilisation . . . . .	28
0.6	Planning d'activités . . . . .	28
<b>III</b>	<b>Conclusion</b>	<b>30</b>
<b>IV</b>	<b>Références</b>	<b>32</b>

# Remerciements

Avant tout développement sur cette expérience au sein de l' Université de Cergy-Pontoise, il paraît tout naturel de débiter ce rapport de stage par des remerciements.

Nous remercions donc dans un premier temps Mr Dominique Laurent et Jen Tao-Yuan, pour nous avoir offert l'opportunité d'intégrer l'Université en tant que stagiaire.

Ensuite, Nous tenons tout particulièrement à remercier Mr Liu Tianxao et Jen Tao-Youn, nos tuteurs de stages, pour nous avoir accompagné tout au long de cette expérience ainsi que pour leurs nombreux conseils qu'ils ont pu nous prodiguer, le tout avec beaucoup de pédagogie.

Et pour terminer, Nous remercions également l'ensemble de l'Université de Cergy-Pontoise , et en particulier le département Informatique, ceci permettant de travailler dans de très bonnes conditions, et pour leur gentillesse de nous avoir permis de travailler dans leurs locaux .

**Part I**

**Introduction**

# Introduction

## 0.1 Le rôle de stage

Le stage final de la licence informatique joue un rôle très important pour la poursuite des études ou pour l'insertion dans le monde du travail, car Il ne s'agit pas d'un simple stage de découverte du milieu de l'entreprise, mais bel et bien d'un stage où il est impératif de réaliser les tâches confiées tout en mettant en oeuvre l'ensemble des connaissances acquises lors de ces trois années de licence à l'université.

Etant potentiellement intéressé d'obtenir une expérience professionnelle dans un domaine de base de donnée qui nous captive,nous nous sommes donc tout naturellement tourné vers l'université qui nous offre l'opportunité de réaliser un stage plaisant et faisant appel à nos connaissances.

## 0.2 Présentation de l' Université de Cergy-Pontoise

L'université de Cergy-Pontoise est une université française située dans le département du Val-d'Oise et la région île-de-France. Elle est implantée principalement dans les villes de Cergy, de Pontoise et de Neuville-sur-Oise, et dispose d'un total de douze sites.

C'est l'une des cinq universités de l'académie de Versailles. Elle est créée en 1991 dans le cadre des universités nouvelles à partir d'implantations universitaires plus anciennes, ouvertes à partir de 1989.

L'établissement intègre le pôle de recherche et d'enseignement supérieur (PRES) de Cergy-Pontoise Val-d'Oise lors de sa création en 2006, et passe aux compétences élargies en 2009.

L'établissement dispense des formations dans les domaines des arts-lettres-langues, du droit-économie-gestion, des sciences humaines et sociales et des sciences et technologie, ainsi que des préparations à des concours à quelques 11 338 étudiants.

Ses activités de recherche sont assurées par quelque 500 enseignants-chercheurs et 350 doctorants, et elles sont structurées autour de quatre pôles :

- Etude des matériaux et sciences expérimentales
- Analyse, modélisation et management de systèmes complexes
- Technologies de l'intelligence embarquée

- Transferts culturels et transmission des savoirs.

### 0.3 Le sujet

**Le sujet qui nous a été donné:**

"Intégration le projet Pentaho Mondrian aux enseignements OLAP".

**Activités demandés:**

Installer et vérifier des logiciels nécessaires(Mysql,TomCat,Mondrian workbench,etc)et la base de donnée démo(Foodmart) et ensuite exploiter le système avec des différentes bases de données et réaliser un GUI pour cette exploitation.

# Part II

## Développement

# Développement

## 0.4 Contexte du stage

Avant de commencer le stage, Mr Jen Tao-Yun maître de stage, nous a fait une réunion et nous a expliqué le sujet du stage en nous donnant les bases et des références pour pouvoir commencer, la durée et d'autres conseils dont certains relatifs à la sécurité (particulièrement pour l'accès au réseau informatique), de travailler en groupe et de s'entraider entre nous.

## 0.5 Le projet

### 0.5.1 Présentation

Le "projet Pentaho Mondrian aux enseignements OLAP" consiste à installer sous windows et Unix des logiciels comme Mysql, TomCat, Mondrian Workbench et une base de donnée Foodmart et puis de réaliser une interface graphique (GUI) en java cette exploitation. Cette exploitation doit être réalisée en langage de programmation java. Son principal intérêt est de pouvoir installer et vérifier ces logiciels en se connectant sur la base de donnée et puis de réaliser une interface graphique en java facilitant la connexion sur la base de donnée sans avoir besoin de modifier chaque fois des fichiers concernés.

### 0.5.2 Etude du problème

Après avoir bien réfléchi et analyser notre sujet de stage, on a commencé par se documenter sur internet car sont des nouveaux logiciels et la plus part sont rédigés en anglais. Après ces recherches, on a commencé l'installation.

### 0.5.3 Installations des Logiciels

Avant commencer l'installation, c'est quoi mondrian OLAP et ça sert à quoi?

Mondrian est un open source OLAP (Online Analytical Processing) serveur, écrit en Java. Il prend en charge le langage de requête MDX (expressions multidimensionnelles)



et le XML for Analysis et les spécifications d'interface olap4j. Il lit à partir de sources de données SQL et d'autres.

Mondrian est utilisé pour:

- Haute performance, l'analyse interactive de grands ou petites quantités d'information
- L'exploration des dimensions des données, par exemple l'analyse des ventes par ligne de produits, par région, par période de temps
- Analyse de la langue MDX en Structured Query Language (SQL) pour récupérer les réponses aux requêtes dimensionnelles
- Haute-vitesse des requêtes grâce à l'utilisation des tableaux de données agrégées dans le SGBDR
- Calculs avancés en utilisant les expressions de calcul du langage MDX

### **Installation de Mondrian Olap WorkBench:**

Pour installer Mondrian Olap WorkBench sous Unix il faut d'abord installer Apache Tomcat Server et MySQL Server. Une mise à jour de Java (JDK et JRE) est nécessaire avant de commencer l'installation. La version de notre Java est Java 6 update 31.

#### **1. Mise à jour de Java**

Téléchargement de la dernière version du JDK (qui contient aussi la dernière version de JRE) dans le site d'Oracle.

Le fichier télécharger est: *jdk-6u31-linux-i586.bin*.

Maintenant, nous devons donner des autorisations d'exécution:

```
sudo chmod 755 jdk-6u31-linux-i586.bin
```

et installer:

```
sudo ./jdk-6u31-linux-i586.bin
```

Maintenant, on déplace le dossier généré dans un répertoire plus approprié:

```
sudo mv jdk1.6.0_31/ /usr/local
```

On met la nouvelle Java comme une des alternatives "java" avec la commande:

```
sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/jdk1.6.0_31/bin/java" 1
```

Maintenant on met la "nouvelle alternative" comme le réel Java.  
Cette étape permet que la nouvelle version de java soit utilisée par défaut:

```
sudo update-alternatives --set java /usr/local/jdk1.6.0_31/bin/java
```

Maintenant, nous faisons les mêmes étapes ci-dessus pour javac:

```
sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/local/jdk1.6.0_31/bin/javac"
sudo update-alternatives --set javac /usr/local/jdk1.6.0_31/bin/javac
```

Pour vérifier que tout fonctionne correctement on tape sur le Terminal:

```
java -version
```

Et on trouve ceci :

```
java version "1.6.0_31"
Java (TM) SE Runtime Environment (build 1.6.0_31-B06)
Java HotSpot (TM) Client VM (build 17.0-b16, mixed mode)
```

Pour javac:

```
javac -version
```

on trouve :

```
javac 1.6.0_31
```

Pour définir le chemin d'installation de JAVA et JRE comme variables d'environnement, on ouvre le fichier

```
" /etc/environment " :  
sudo gedit /etc/environment
```

Ici, nous enregistrons les routes où nous avons installé Java :

```
JAVA_HOME = "/usr/local/jdk1.6.0_31/bin/"  
JRE_HOME = "/usr/local/jdk1.6.0_31/jre"  
PATH = "(autres path): JAVA_HOME : JRE_HOME"
```

Et sous Windows on met à jour "java JDK 1.6.\_31 " et on installe le fichier exécutable, on suit le chemin d'installation en cliquant sur suivant.  
Fin de mise à jour de Java.

## 2.Installation du serveur MySQL

Installation du MySQL client et MySQL server avec les commandes suivantes :

```
sudo apt-get install mysql-client  
sudo apt-get install mysql-server
```

Pendant l'installation du mysql-server on a créé un mot de passe pour l'utilisateur " root ". Après l'installation on a créé aussi un utilisateur " foodmart " avec mot de passe " foodmart " pour l'importation de la base de données FoodMart :

```
mysql -u root -p " password
```

```
mysql> create database foodmart;  
mysql> create user "foodmart"@"localhost' identified by "foodmart";  
mysql> grant all privileges on *.* to "foodmart"@"localhostâ identified by "foodmart";
```

Et sous Windows on télécharge la dernière version Mysql sur le site de Mysql, et puis on clique sur le fichier exécutable et on suit les instructions.

## 3. Installation d'Apache Tomcat Server

Téléchargement de la dernière version d'Apache Tomcat " *apache-tomcat-7.0.27.tar.gz*"

à partir du site d'Apache.

On le décompresse avec la commande suivante:

```
tar -xvzf apache-tomcat-7.0.27.tar.gz
```

Et puis nous mettons dans un répertoire approprié, dans notre cas: */usr/local/tomcat7*. Nous faisons cela avec la commande:

```
sudo mv apache-tomcat-7.0.27 /usr/local/tomcat7 (le répertoire tomcat7 sera créé).
```

Nous avons eu quelques problèmes dans la définition de ces variables d'environnement, quelque fois le tomcat ne reconnaît pas, mais un moyen infaillible de reconnaître tomcat, est de définir les chemins de fichiers à l'intérieur "catalina.sh" situé dans tomcat7/bin.

Pour modifier ce fichier, utilisez la commande suivante:

```
"sudo gedit /usr/local/tomcat7/bin/catalina.sh"
```

Maintenant il suffit d'insérer le JAVA\_HOME et JRE\_HOME après la première ligne, de sorte que le fichier soit comme suit:

```
!/bin/sh  
JAVA_HOME="/usr/local/jdk1.6.0_31/bin"  
JRE_HOME="/usr/local/jdk1.6.0_31/jre"  
Licensed to the Apache Software Foundation (ASF)...
```

```
...  
...  
....
```

On configure les utilisateurs Tomcat, cela se fait dans le fichier " *tomcat-users.xml* " situé dans le répertoire *tomcat7/conf*.

```
"sudo gedit /usr/local/tomcat7/conf/tomcat-users.xml"
```

Contrairement aux versions précédentes, où l'administrateur doit posséder rôle "manager" Il devrait maintenant être "manager-gui" à opérer sur le web d'administration tomcat7. Le fichier se présente comme suit:

```
<? xml version='1.0' encoding='utf-8'?>
<tomcat-users>
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="manager"/>
<role rolename="admin-gui"/>
<role rolename="admin-script"/>
<role rolename="admin"/>
<user username="username" password="password"
roles="manager-gui,admin-gui,manager,admin,manager-script,admin-script"/>
</tomcat-users>
```

On met les droits d'exécution pour les scripts " *startup.sh* " et " *shutdown.sh* " qui permettent respectivement de lancer et arrêter le serveur d'Apache.

```
sudo chmod 755 /usr/local/tomcat7/bin/startup.sh
/usr/local/tomcat7/bin/shutdown.sh
```

On lance le serveur avec la commande suivante:

```
"sudo /usr/local/tomcat7/bin/startup.sh"
```

Et sous Windows on extrait apache dans le répertoire qu'on veut et on tape dans la console: run startup.bat

Avec ceci, nous obtenons le résultat suivant sur la console:

```
Using CATALINA_BASE: /usr/local/tomcat7
Using CATALINA_HOME: /usr/local/tomcat7
Using JRE_HOME : /usr/local/jdk1.6.0_31/jre
Using CLASSPATH:
/usr/local/tomcat7/bin/bootstrap.jar:/usr/local/tomcat7/bin/tomcatjuli.jar
```

On vérifie que le JRE\_HOME est l'endroit où nous avons défini pendant la mise à jour de Java. Changement du port par défaut du serveur d'Apache de 8080 à 8181 (si le port 8080 est déjà utilisé dans l'ordinateur). On ouvre le fichier " server.xml " qui se trouve dans le répertoire /usr/local/tomcat7/conf/ :

```
"sudo gedit /usr/local/tomcat7/conf/server.xml"
```

On cherche un endroit dans le fichier similaire à:

```
<Connector port="8080" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />
```

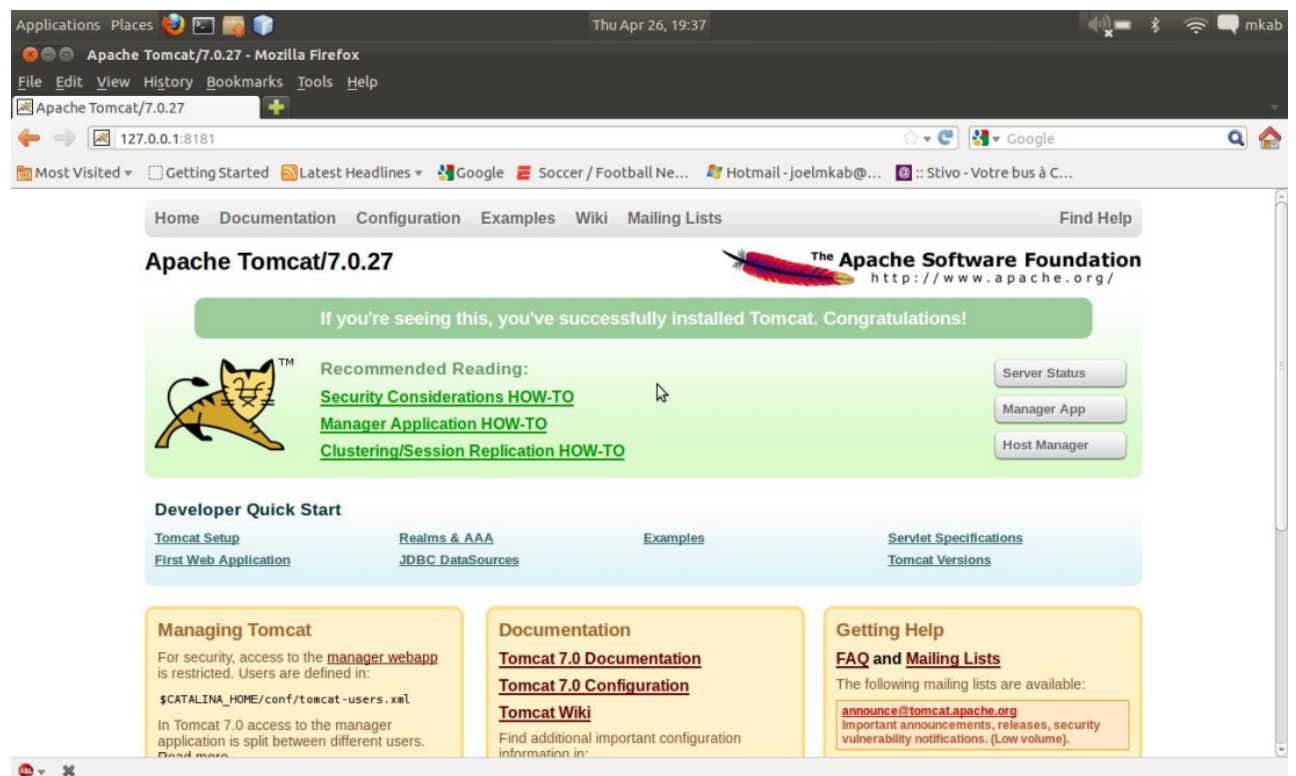
Et on change le port à 8181 :

```
<Connector port="8181" protocol="HTTP/1.1"  
connectionTimeout="20000"  
redirectPort="8443" />
```

Maintenant, ouvrez votre navigateur Web et saisissez l'URL suivante:

<http://127.0.0.1:8181/>

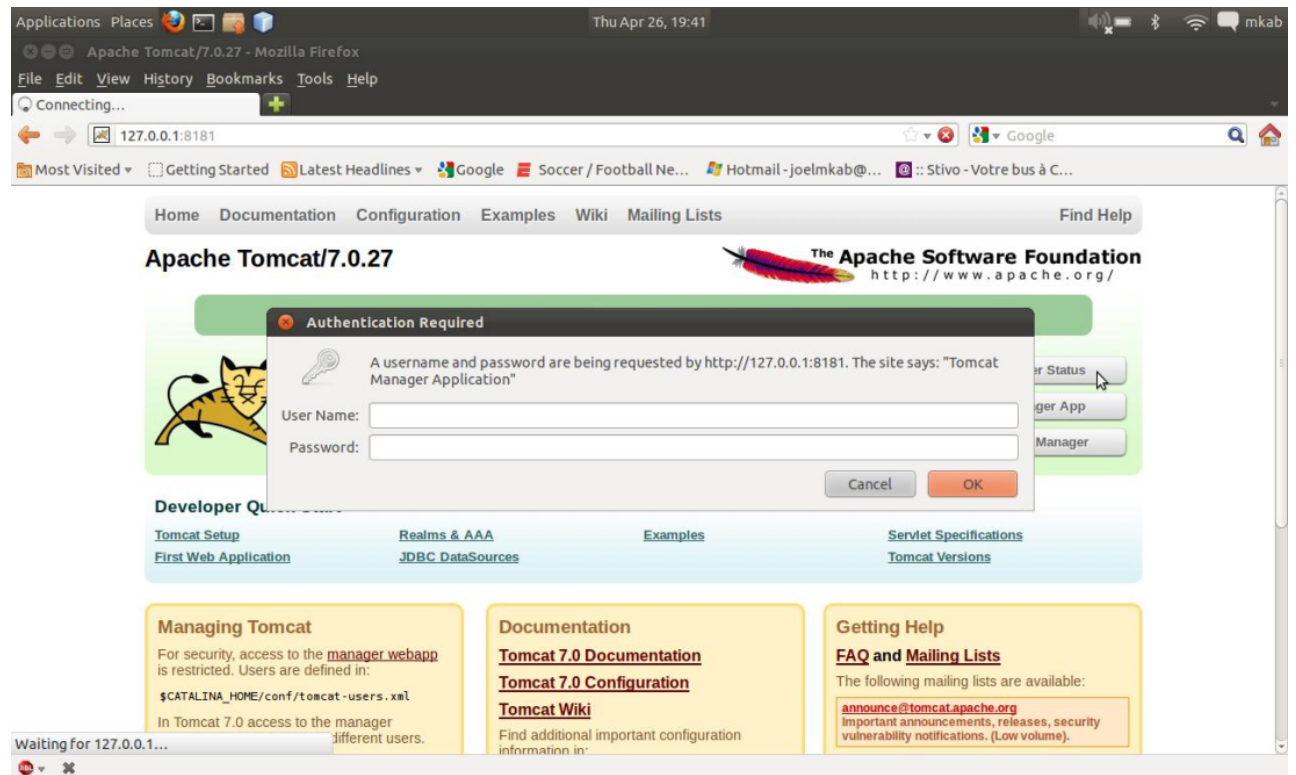
Ainsi nous obtenons la page suivante:



## Apache Tomcat

Si nous entrons dans le gestionnaire de l'administration de Tomcat, nous cliquons sur le menu ou directement à l'adresse URL: *http://127.0.0.1:8181/manager/html*

Un nom d'utilisateur et mot de passe sera demandé et on entre les données qu'on avait configuré dans le fichier " *tomcat-users.xml* "



## Authentication Requis



Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/mondrian	None specified	Mondrian	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/test	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

**Deploy**  
 Deploy directory or WAR file located on server  
 Context Path (required):   
 XML Configuration file URL:

On a testé l'échantillon pour s'assurer que tout fonctionne bien, ils sont dans la section "Miscellaneous" dans le menu ou à l'adresse URL:

*<http://127.0.0.1:8181/examples/>*

## 2.1 Commandes

*Démarrer le serveur:*

*`sudo /usr/local/tomcat7/bin/startup.sh`*

*Arrêter le serveur:*

```
sudo /usr/local/tomcat7/bin/shutdown.sh
```

## 2.2 Démarrage automatique

Pour que tomcat se démarre automatiquement lorsque nous avons démarré l'ordinateur, on a ajouté un script pour faire le démarrage automatique et aussi l'arrêt. Voici les étapes :

```
sudo gedit /etc/init.d/tomcat7
```

Puis dans ce fichier :

```
Tomcat auto-start
description: Auto-starts tomcat
processname: tomcat
pidfile: /var/run/tomcat.pid
case 1 in
start) starts tomcat server
sh /usr/share/tomcat7/bin/startup.sh
;;
stop) stops tomcat server
sh /usr/share/tomcat7/bin/shutdown.sh
;;
restart) restarts (starts and stops) tomcat server
sh /usr/share/tomcat7/bin/shutdown.sh
sh /usr/share/tomcat7/bin/startup.sh
;;
esac
exit 0
```

On rend le script exécutable en exécutant la commande chmod:

```
sudo chmod 755 /etc/init.d/tomcat7
```

La dernière étape est en fait de lier ce script pour les dossiers de démarrage avec un lien symbolique.

Exécuter ces deux commandes et nous devrions être sur notre chemin.

```
sudo ln -s / etc/init.d/tomcat7/etc/rc1.d/K99tomcat
sudo ln -s / etc/init.d/tomcat7/etc/rc2.d/S99tomcat
```

Tomcat est maintenant complètement installé et opérationnel.

#### **4.Installation de Mondrian Olap WorkBench**

Téléchargement de la dernière version de Mondrian sur le site de Sourceforge et on le décompresse dans `/usr/local/mondrian`. Le résultat sera un nouveau répertoire contenant tous les fichiers de Mondrian.

On a créé un répertoire mondrian dans `/usr/local/tomcat/webapps/` et déplacez "`mondrian.war`" à l'existence.

Le résultat devrait ressembler à ceci:

```
/usr/local/tomcat/webapps /mondrian /mondrian.war.
```

On décompresse mondrian.war.

On télécharge le mysql connector "`mysql-connector-java-5.1.19.tar.gz`" et puis on déplace le fichier jar dans `/usr/local/tomcat/webapps/mondrian/WEB-INF/lib`. Puis exécution de l'application FoodMartLoader pour charger la base de données avec les données Foodmart.

La commande utilisé est :

```
java -cp "/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/mondrian.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/log4j-1.2.8.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/eigenbase-xom.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/eigenbase-resgen.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/eigenbase-properties.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/commons-logging-1.0.4.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/commons-collections-3.1.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/olap4j.jar:/usr/local/tomcat7/webapps/mondrian/WEB-INF/lib/mysql-connector-java-5.1.19-bin.jar" mondrian.test.loader.MondrianFoodMartLoader -verbose -tables -data -indexes -jdbcDrivers=com.mysql.jdbc.Driver -inputFile=/usr/local/mondrian/mondrian-3.3.0.14703/demo/FoodMartCreateData.sql -outputJdbcURL="jdbc:mysql://localhost/foodmart?user=foodmartpassword=foodmart"
```

On copie le fichier " xalan.jar " à " *tomcat/common/endorsed* "

```
cp /usr/local/tomcat/webapps/mondrian/WEB-INF/lib/xalan.jar/usr/local/tomcat/common/endorsed
```

Allez dans /usr/local/tomcat/webapps/mondrian/WEB-INF et modifier les fichiers web.xml et datasources.xml, le cas échéant.

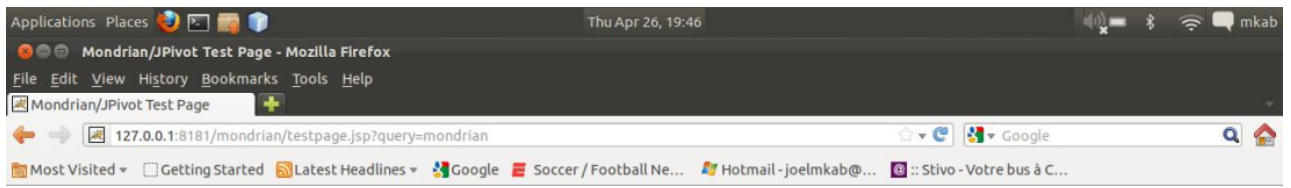
La valeur conductrice doit être:

*JdbcDrivers = com.mysql.jdbc.Driver*

On teste l'installation de Mondrian avec :

*http://localhost:8080/mondrian*

et puis on a:



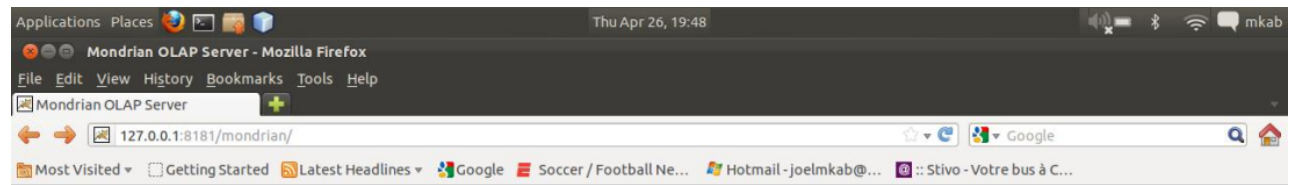
## Test Query uses Mondrian OLAP



		Measures		
Promotion Media	Product	Unit Sales	Store Cost	Store Sales
+All Media	+All Products	266,773	225,627.23	565,238.13

Slicer: [Year=1997]

[back to index](#)

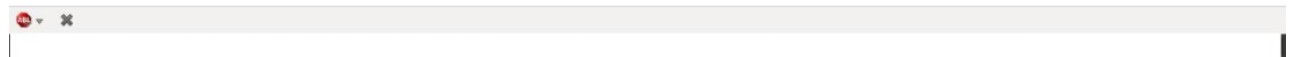


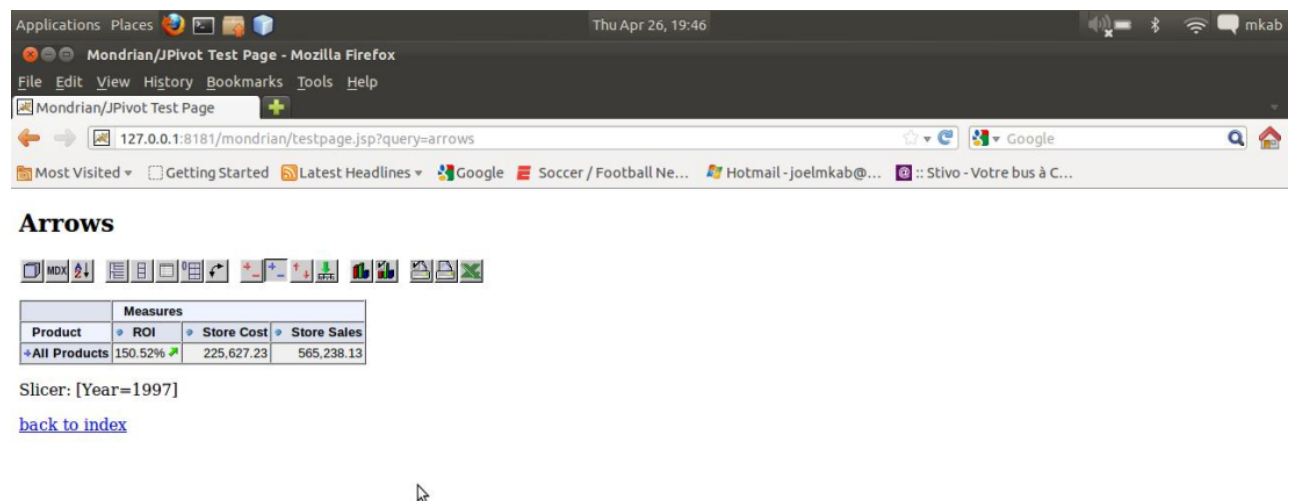
Mondrian examples:

- [JPivot pivot table](#)
- [JPivot pivot table by XMLA](#)
- [JPivot with 4 hierarchies](#)
- [JPivot with role 'California Manager' set](#)
- [JPivot with arrows](#)
- [JPivot with colors](#)
- [Various queries formatted using the Mondrian tag-library](#)
- [Basic interface for ad hoc queries](#)
- [XML for Analysis tester](#)

Other links:

- [Mondrian home page](#)
- [Mondrian project page](#)
- [JPivot home page](#)
- [JPivot project page](#)





## 0.5.4 Utilisations des logiciels

### Utilisation de Mondrian schema-workbench

Après avoir téléchargé sur le site de *mondrian.pentaho.com* la dernière version de mondrian qui est *workbench-2.3.2.9247.zip* et le décompresser dans le répertoire approprié.

Puis télécharger sur le site de Java la dernière version *mysqlconnector* qui est *mysql-connector-java-5.1.2.0.tar.gz* le décompresser et puis mettre le fichier *mysql-connector-java-5.1.20-bin.jar* dans le dossier *drivers* qui se trouve dans le dossier *schema-workbench*. S'il n'y est pas on le crée pour pouvoir mettre le fichier *mysql-connector-java-5.1.20-bin.jar* parce qu'il y'a des anciennes version de *schema-workbench* qui n'ont pas de dossier *drivers*.

Une fois tout est fait, on se place dans le dossier *schema-workbench* et on donne les droits aux fichiers *set-pentaho-env.sh* et *workbench.sh* en tapant ces ligne de commande:

```
"sudo chmod 755 workbench.sh" et
"sudo chmod 755 set-pentaho-env.sh"
```

Biensur sur ubuntu car sur windows on a pas besoin de donner les droits.

Pour lancer schema-workbench on doit se placer à l'endroit où on l'a installé et puis on l'exécute en tapant `./workbench.sh` sur *ubuntu* et `run workbench.bat` sur *Windows*. On verra apparaître par exemple des lignes de commandes semblables à cela:

```
DEBUG: Using JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/local/jdk1.6.0_31/
DEBUG: _PENTAHO_JAVA=/usr/local/jdk1.6.0_31/bin/java
```

et en même temps s'affiche la fenêtre du schema workbench qui est la suivante :



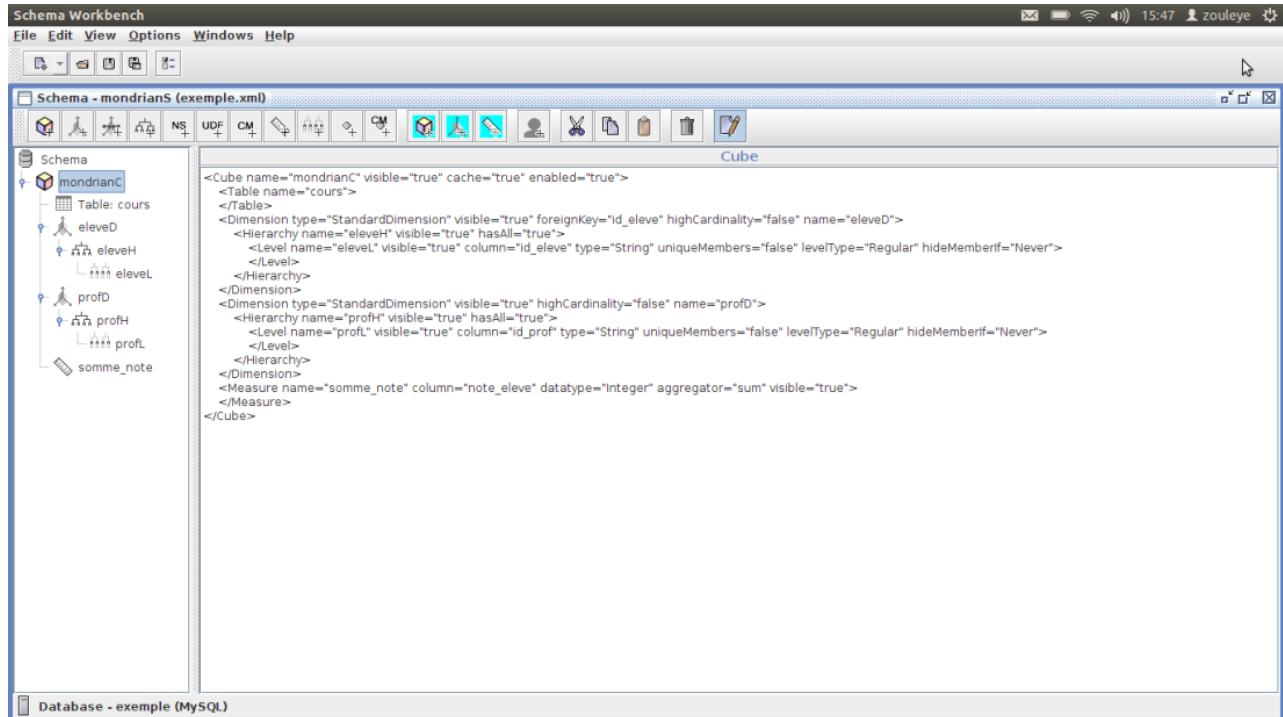
Donc quand le schema-workbench est bien installé, il va nous servir à créer un fichier xml qui correspond à la structure de notre base de donnée.

Pour ce fait on va sur la fenêtre du schema workbench qui est déjà ouverte, on clique sur Options puis sur connection pour créer une connexion entre notre base de donnée qu'on a créée dans mysql en remplissant les champs nécessaires, puis créer trois tables dans cette base de donnée dont deux avec chacune sa clef primaire et la troisième qui contiendra deux clefs étrangères feront appel aux deux clefs primaire des deux premières tables et il faut au moins une colonne qui contient des entiers.

Ensuite on a créé un cube avec schema workbench on l'a nommé mondrianC , qui génère automatiquement une table qu'on a renommé cours , on a créé une mesure et deux dimensions (eleveD et profD) sous lesquelles on a créé respectivement des hiérar-



chies (eleveH et profH) et sous les hiérarchie aussi on crée respectivement des levels (eleveL et profL). Maintenant on se place sous Measure on remplit les champs nécessaires pour générer automatiquement la requête MDX et le fichier XML. Pour voir ce dernier on clique sur view situé en haut de la fenêtre schema workbench puis view XML.



Pour connecter mondrian à une base de donnée il faut modifier les fichiers suivant :

- arrows.jsp
- colors.jsp
- mondrian.jsp
- fourhier.jsp

Se trouvant dans : *tomcat/Webapps/mondrian/WEB-INF/queries.*

Dans ces fichiers ".jsp" on change juste la connexion en modifiant les chemins de : *jdbcTemplate* , *jdbcUrl* , *catalogUri*

### 0.5.5 Interface graphique en java

#### MondrianConfig.java

Remarque : tout au long de rapport CATALINA\_HOME est le chemin de l'installation

de Tomcat.

Mondrian utilise certains fichiers de configuration pour la connexion à une base de donnée. Ces fichiers et leurs emplacements se trouvent dans le tableau ci-dessous :

Fichiers Mondrian	Emplacement
arrows.jsp	<i>CATALINA_HOME/webapps/mondrian/WEB-INF/queries</i>
mondrian.jsp	<i>CATALINA_HOME/webapps/mondrian/WEB-INF/queries</i>
colors.jsp	<i>CATALINA_HOME/webapps/mondrian/WEB-INF/queries</i>
hline fourhier.jsp	<i>CATALINA_HOME/webapps/mondrian/WEB-INF/queries</i>
adhoc.jsp	<i>CATALINA_HOME/webapps/mondrian</i>
datasources.xml	<i>CATALINA_HOME/webapps/mondrian/WEB-INF</i>

Pour que Mondrian se connecte à une base de donnée différent de FoodMart, il faut modifier les fichiers mentionnés dans le tableau ci-dessus et relancer Mondrian pour que les configurations soit prise en compte.

Il faut avoir déjà créé un fichier XML avec le Mondrian Schema Workbench et copier se fichier à l'endroit " CATALINA\_HOME/webapps/mondrian/WEB-INF/queries ".

On a un fichier ".jsp" qui ressemble à: jp:mondrianQuery id="query01"

```
jdbcDriver="on insère le driver jdbc de la base de donnée"
jdbcUrl="l'URL de la base de données"
catalogUri="/WEB-INF/queries/le fichier XML crée avec Workbench"
jdbcUser="nom d'utilisateur de la base de données"
jdbcPassword="mot de passe de la base de données"
connectionPooling="false"
role="California manager"
```

ON INSERE LES REQUETES MDX

</jp:mondrianQuery>

<c:set var="title01" scope="session">Test query uses Mondrian OLAP, using role 'California manager'</c:set>

Les endroits entre les guillemets en italiques sont à changer.

Le but est de créer une interface graphique pour changer automatiquement dans ces fichiers, le `jdbcDriver`, `jdbcUrl`, `catalogUri`, `jdbcUser` et `jdbcPassword` afin que Mondrian puisse accéder correctement à la base de données.

#### **4.Classes Utilisées**

##### ***1.EnvFactory.java***

Cette classe a une fonction statique qui permet d'avoir le chemin où Tomcat à été installé. Il renvoie un String qui contient le chemin d'installation de Tomcat.

##### ***2.ModrianData.java***

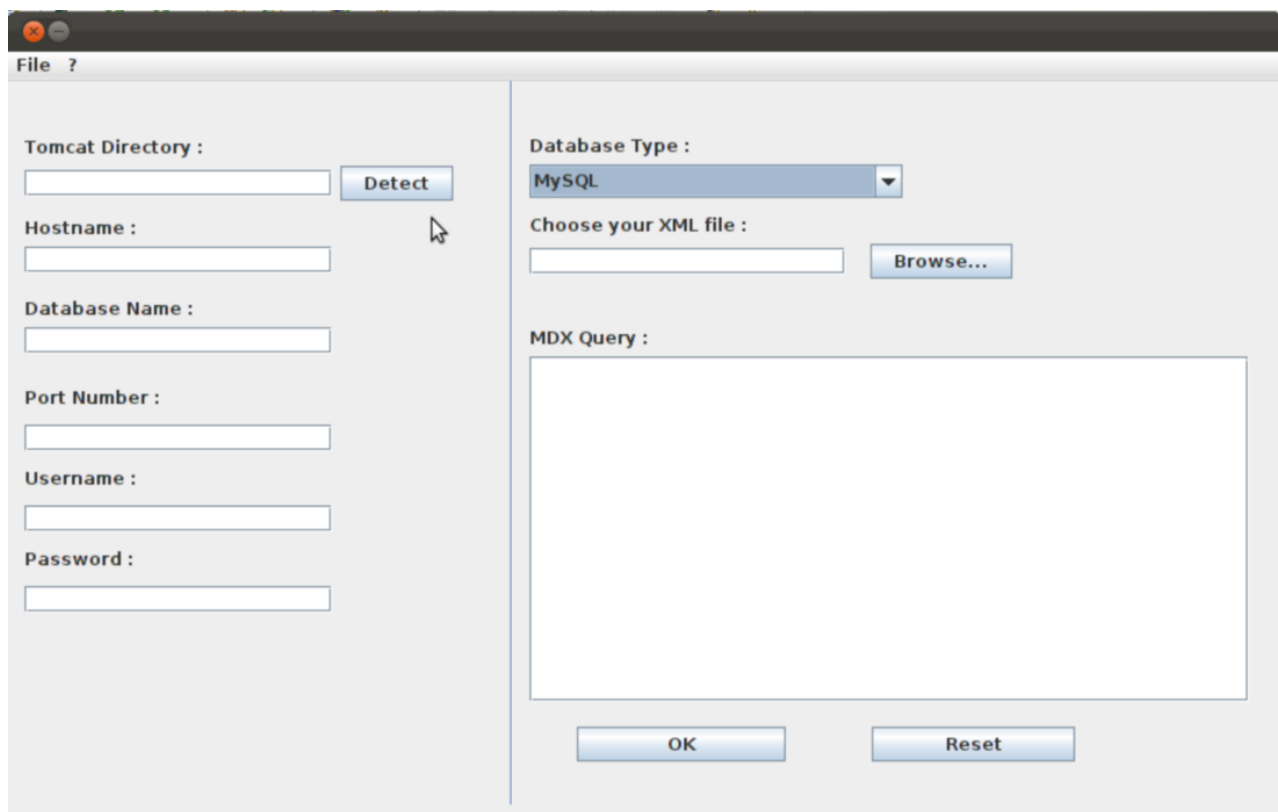
Cette classe est permet de contenir toutes les données nécessaire à insérer dans les fichiers JSP et XML.

##### ***3.MondrianConfig.java***

Cette classe contient des méthodes dédiées pour changer les configurations différentes dans les fichiers JSP et XML.

##### ***4.MondrianConfigGUI.java***

Cette classe dessine une interface graphique simple et conviviale pour les utilisateurs.



Le principe du fonctionnement de MondrianConfigGUI

### Quelques remarques importantes

1. Il faut lancer le programme en tant que root pour être sûr d'avoir toutes les privilèges nécessaires.
2. Copier et sauvegarder les fichiers arrows.jsp, mondrian.jsp, colors.jsp, fourhier.jsp, adhoc.jsp et datasources.xml dans un dossier à part pour avoir la possibilité de revenir à votre ancien configuration si nécessaire.

Pour modifier les fichiers de configuration, entrer les données caractéristiques de son base de donnée.

Les données à rentrer sont:

- Tomcat Directory - Le repertoire ou tomcat a été installé.
- Hostname - Le nom de l'hôte.
- Database Name - Le nom de la base de données.

- Port - Le numéro de port.
- Username - Le nom d'utilisateur.
- Password - Le mot de passe.
- Database type- Le type de la base de données (seulement MYSQL et ORACLE sont pris en compte dans cette version).
- XML File - Le chemin du fichier XML crée avec Mondrian Schema Workbench.
- MDX query - La requête MDX a été exécuté par mondrian. (facultatif)

Après avoir rentré ces valeurs, cliquer sur OK.

On change des fichiers de configuration nécessaires.

L'interface crée des fichiers temporaires (avec l'extension .tmp) à partir des fichiers originaux JSP et XML en changeant les données nécessaires (jdbcurl, jdbcUser, jdbcPassword etc.).

Lancer le script backup.sh en temps que root.

Le script crée un dossier avec le nom " backup " et copie ces fichiers originaux (JSP et XML) dans ce dossier et il renomme les fichiers temporaires avec les noms des fichiers backup.

Ces anciens fichiers sont sauvegardés dans le repertoire backup de CATALINA\_HOME/webapps/mondrian/WEBINF/ queries.

Relancer Mondrian pour tester votre nouvelle configuration.

### 0.5.6 Manuel d'utilisation

En construction car le projet n'est pas encore terminé, une partie d'interface java est en cours de réalisation.

## 0.6 Planning d'activités

Notre projet de stage nous a été donné au fur à mesure, en premier lieu, on nous a été demandé de se documenter sur le sujet et d'installer Mysql, Tomcat et Mondrian sous unix et sous windows. Nous avons fait ce travail pendant 2 semaines.

Deux semaines d'après, on nous a demandé de créer une base de données et de se connecter aux bases de données foodmart et puis de créer des cubes à partir de ces tables.

Ensuite depuis 2 semaines, on essaye de modifier quelques fichiers de Mondrian pour pouvoir se connecter à la base de données et de créer une interface graphique qui nous

facilitera de modifier les fichiers automatiquement sans être obligé de se déplacer chaque fois dans les fichiers.

# Part III

## Conclusion

# Conclusion

Comme vous avez pu le constater la première phase de notre projet nous a pris une grande partie du premier mois de stage, car nous ne connaissions pas les logiciels à installer, il nous a donc fallu apprendre à installer et à les utiliser.

Jusqu' à présent, on a réussi à installer des logiciels demandés sous Unix et sous Windows, on a réussi à se connecter sur la base de donnée foodmart , après on a crée notre base de donnée , nous avons établi une connection sur cette base sans aucun problème en modifiant quelques fichiers. On a fait aussi une interface graphique en java pour se connecter à la base de donnée sans modifier les fichiers chaque fois. Pour le moment on travaille sur cette interface et dans deux semaines qui nous restent on pourra tester sur une base de donnée de gros volume.

Nous pouvons donc dès à présent affirmer que ce stage est réellement formateur, particulièrement en terme de techniques de travail (organisation, autoformation, prise de recul, anticipation des problèmes éventuels), et a permis d'enrichir nos connaissances en nous faisant découvrir le monde de base donnée avancée.



# Part IV

## Références

# Références

## **Les Liens Utiles :**

### **Apache Tomcat (version 7.0.27) :**

<http://tomcat.apache.org/download-70.cgi#7.0.27>

### **Mondrian Olap WorkBench (Edition Pentaho) :**

<http://sourceforge.net/projects/mondrian/>

## **References:**

### **Installation d'Apache Tomcat Server:**

<http://diegobenna.blogspot.fr/2011/01/install-tomcat-7-in-ubuntu-1010.html>

### **Installation de Mondrian :**

<http://mondrian.pentaho.com/documentation/installation.php>

<http://computersight.com/software/how-to-install-mondrian-the-complete-guide>