

# Génération d'un rapport automatique de model de base de données PostgreSQL

## Analyse

### 1. Préambule

Le but du projet est à partir du descriptif d'une base de données (PostgreSQL...) de générer un document .PDF contenant le model de données avec le graphe et la liste des tables et leur détail.

Une recherche a été faite pour voir si des outils existent à cet effet et mettre en place une procédure avec ces derniers et compléter avec une partie en développement :

### 2. Outils

Les outils qui seront utilisés sont les suivants :

SchemaCrawler est un outil libre de documentation de bases de données qui permet d'extraire des informations sur la structure d'une base de données et de les afficher sous forme de diagrammes et de rapports. Il prend en charge plusieurs types de bases de données, notamment Oracle, PostgreSQL, MySQL, etc. Il prend en charge plusieurs formats de sortie, notamment HTML, CSV, JSON et GraphML. Il peut également être utilisé pour générer des scripts SQL pour la création de la base de données. Il peut être utilisé en ligne de commande.

Il est à télécharger via le lien suivant :

- <https://github.com/schemacrawler/SchemaCrawler/releases>

Jsoup est un outil libre conçu pour analyser, extraire et manipuler des données stockées dans des documents HTML.

Il est à télécharger via le lien suivant :

- <https://central.sonatype.com/artifact/org.jsoup/jsoup/1.15.4>

iText est un outil libre utilisé pour créer, modifier, traiter et extraire des fichiers PDF. Le module pdfHTML dans la bibliothèque iText va permettre de convertir le fichier HTML en PDF.

Il est à télécharger via le lien suivant :

- <https://central.sonatype.com/artifact/com.itextpdf/itext7-core/7.2.5>
- <https://central.sonatype.com/artifact/com.itextpdf/html2pdf/4.0.5>

log4j est libre qui permet de gérer des traces et des historiques d'applications.

- <https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core>
- <https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-api>

Développement en java avec l'outil de construction de projet Maven

### 3. Déroulement

Création d'un écran principal de saisie des paramètres de connexion à une base de données et autres.

Exécution de Schemacrawler pour récupérer des éléments du model de données de la base de données

Interprétation et mise en forme des éléments retournées par schemacrawler

Génération d'un fichier .PDF contenant le model de données de la base interrogée

### 4. Introduction

Un écran apparait avec des zones de saisie qui correspond aux éléments de connexion d'une base de données et des paramètres a passé au module en ligne de commande. Ce seront les paramètres d'une fonction qui va exécuter une ligne de commande. Cette ligne de commande exécute le programme d'un module nommé schemacrawler et génèrent un fichier. Un diagramme Entité Relation et un Descriptif des tables d'une base de données au format HTML.

### 5. Exécution de l'application

Description des composant de l'écran nommé « FEN\_Accueil »

Désignation	Libelle	Type	Format affiché	Obligatoire	Action sur le champ	Détail
< LST_Type_base>	Type de Base de Données	Liste	Libellé	OUI	Choix	Initialisation de la liste : (SQLite, oracle, SQL server, db2, MySQL, PostgreSQL, offline)
< SAI_Adresse_serveur >	Adresse du serveur	Text		OUI	Saisie	
< SAI_N_Port>	Numéro de Port	numérique		OUI	Saisie	

< SAI_Nom_Base_De_Donnees>	Nom Base de Donnée	text		OUI	Saisie	
< SAI_Description_Base_De_Donnees>	Description de la base de données :	text			Saisie	Ce texte libre apparaîtra sur le rapport final.
< SAI_Id_Utilisateur>	Identifiant			OUI	Saisie	Identifiant de connexion à la base de données
< SAI_Mdp_utilisateur >	Mot de passe		*****	OUI	saisie	Mot de passe de connexion à la base de données
<INT_MasqueMdp>	Masqué ?	Check			Choix	Affiche et masque le mot de passe
< SAI_Schema>	Nom du schéma	text		OUI	Saisie	= Valeur saisie à la création de la base de données (public,...)
<LST_Niveaux_informations>	Niveaux de Informations	Liste		OUI	Choix	Initialisation de la liste : ( detailed, minimum, standard, maximum )
< SAI_Chemin_fichier_final>	Chemin du fichier	text		OUI	Affichage	le composant prendra la chaîne de caractère de la variable CheminRepertoireModeleDonnees du fichier de configuration du projet
<BTN_LancerProcédure>	Lancer la procédure	Button			Clique	Action = Lance toutes les fonctions jusqu'à la génération du rapport
<BTN_Tester_Connexion>	Tester connexion	Button			Clique	Action = Lancer la fonction « TesterConnexion » pour tester la connexion à la base de données paramétrée.
<BTN_Aide>	Aide	Bouton			Clique	Action = Afficher une fenêtre pop-up pour l'utilisateur contenant des messages d'aide à l'utilisation.

## 6. Les règles de gestion

Code	Description	Composant concerné
RG_1	La sélection d'une option est obligatoire et un control est fait si aucune option n'est sélectionnée, on affichera un message d'alerte bloquant suivant : « Le type de base est obligatoire, compléter votre saisie » et ceci interrompt le traitement »	< Type_base>
RG_2	La saisie de ce champ est obligatoire et un control est fait si le champ est vide, on affichera un message d'alerte bloquant suivant : « L'adresse du serveur est obligatoire, compléter votre saisie » et ceci interrompt le traitement »	<Adresse_serveur>
RG_3	La saisie de ce champ est obligatoire et un control est fait si le champ est vide , on affichera un message d'alerte bloquant suivant : « Le numéro de port du serveur obligatoire, compléter votre saisie » et ceci interrompt le traitement. De plus le champ n'accepte que des chiffre	<N_Port>
RG_4	La saisie de ce champ est obligatoire et un control est fait si le champ est vide ou contient des doubles quotes, on affichera un message d'alerte bloquant suivant : « Le nom de base de données est obligatoire, compléter votre saisie » et ceci interrompt le traitement »	<Nom_Base_De_Donnees>
RG_5	La saisie de ce champ est obligatoire et un control est fait si le champ est vide ou contient des doubles quotes, on affichera un message d'alerte bloquant suivant : « L'identifiant de la base de la base de données est obligatoire, compléter votre saisie » et ceci interrompt le traitement »	<Id_Utilisateur>

RG_6	La saisie de ce champ est obligatoire et un control est fait si le champ est vide, on affichera un message d’alerte bloquant suivant : « Le mot de passe de la base de données est obligatoire, compléter votre saisie » et ceci interrompt le traitement	<Mdp_utilisateur>
RG_7	La saisie de ce champ est obligatoire et un control est fait si le champ est vide ou contient des caractères spéciaux , on affichera un message d’alerte bloquant suivant : « Le nom du schéma de la base de données est obligatoire, compléter votre saisie » et ceci interrompt le traitement	<Schema>
RG_8	Par défaut le niveau « standard » est sélectionné.	<Niveaux_informations>
RG_9	Le chemin d’accès aux répertoires dans le fichier config doit obligatoirement être renseigné et valide . Un control sera fait à l’exécution de l’application et en cas erreur, on affichera un message d’erreur et ceci interrompt le traitement.	Le fichier de configuration « config.properties »
RG_10	La saisie de ce champ n’est pas obligatoire et un control est fait si le champ contient des doubles quotes , on affichera un message d’alerte bloquant suivant : « La description de la base de donnée contient des doubles quotes, modifier votre saisie » et ceci interrompt le traitement	<SAI _Description_Base_De_Donnees>

## 7. L’action du bouton <btnTester\_Connexion>

### a. Vérifications des champs saisie

Une fonction nommée VerifChamps() va vérifier les champs avant de lancer la procédure et appliquer les règles de gestion nécessaires.

Si des champs ne respectent pas les règles de gestion un message d’alerte l’indiquera à l’utilisateur qui devra compléter sa saisie avant de lancer le traitement.

### b. Test de connexion

Cette action lance la fonction TesterConnexion() qui permet de tester la connexion à une base de données grâce aux éléments de connexion entré en paramètre et d’afficher à l’écran un message positif en cas de succès ou négatif en cas d’échec de connexion. La connexion se fera grâce au pilot JDBC du type de base sélectionné. Ces pilotes devront être installer grâce aux dépendances maven.

La fonction TesterConnexion() attend le passage des paramètres suivant :

TesterConnexion(< Type\_serveur> , <Nom\_Base\_De\_Donnees> , <Adresse\_serveur> , <N\_Port> , <Id\_Utilisateur> , <Mdp\_utilisateur>)

## 8. L'action du bouton :<btnAfficherAide>

Cette action affiche une nouvelle fenêtre qui permet de renseigner les détails concernant l'interface avec un exemple et comment utiliser l'application.

## 9. L'action du bouton :<btnLancerProcedure>

### a. Vérifications des champs saisie

Une fonction nommée VerifChamps() va vérifier tous les champs avant de lancer la procédure et appliquer les règles de gestion nécessaires.

Si des champs ne respectent pas les règles de gestion un message d'alerte l'indiquera à l'utilisateur qui devra compléter sa saisie avant de lancer le traitement.

Le bouton < btnLancerProcedure > va lancer plusieurs traitements décrit ci-dessous.

### b. Teste de connexion à la base de données

Un test de connexion à la base de données sera effectué avec les paramètres saisie avant de lancer la procédure. En cas d'échec un message indiquera une erreur « Echec de la connexion ». La connexion se fera grâce à des méthodes. Ces méthodes utiliseront des pilotes JDBC du type de base sélectionné. Ces pilotes devront être installés grâce aux dépendances Maven.

### c. Récupération des paramètres et constitution des lignes des commandes

### d. Lancement de SchemaCrawler en ligne de commande

La fonction ExexuterSchemacrawler( < Type\_serveur> , <Nom\_Base\_De\_Donnees> , <Adresse\_serveur> , <N\_Port> , <Schema> , <Id\_Utilisateur> , <Mdp\_utilisateur> , <Niveaux\_informations> ) va générer une ligne de commandes et l'exécute dans un terminal.

La syntaxe de la ligne de commande est la suivante :

```
schemacrawler --server=< Type_serveur> database=<Nom_Base_De_Donnees>
--host=<Adresse_serveur> --port=<N_Port> --schemas=<Schema> --user=<Id_Utilisateur>
--password=<Mdp_utilisateur> --command=schema --info-level= <Niveaux_informations>
--title=Rapport Modele de Données --load-row-counts --weak-associations
--output-format=htmlx --output-file=metadata.html
```

*Ligne de commande pour le diagramme Entité Relation et la description des tables*

Ce traitement permet de se connecter une base de données et générer un fichier de format html nommé avec le nom passer en paramètre (ici metadata.html).

Le fichier sera stocké dans un répertoire créé spécialement nommé « RepertoireTravail ». La connexion se fera automatique grâce au pilote JDBC du type de base sélectionné qui sont intégrés à l'outil. Aucune autre action n'est requise.

## e. Interprétation du fichier HTML

Une classe nommée « InterpretationHTML » disposera de plusieurs méthodes qui interpréteront un fichier HTML pour traduire le document, reformater les tables.

Le fichier HTML est composé d'un diagramme et d'une succession de tableaux qui représentent chaque table de la base de données. Une table est composée des éléments suivants :

- Chaque colonne de la table avec son type, son commentaire, si elle est nullable, si elle est auto-incrémenté, etc. Toutes les métadonnées de la colonne les uns à la suite des autres
- La clé primaire de la table
- Les clés étrangères de la table
- Les indexes de la table

Dans ce fichier HTML plus précisément dans chaque table il faut reformater les lignes pour avoir chaque métadonnée d'une colonne de la base de données dans des colonnes distinct du tableau. De plus au début du fichier quelques informations sont indiquées concernant le rapport : la date et quel outil a été utilisé. Dans le cas du projet ces informations sont inutiles. Elles seront remplacées par le nom de la base de données et la description de la base. Ces informations sont saisies par l'utilisateur.

La méthode nommée « FormatageHTML » va parcourir le fichier HTML pour interpréter le contenu et ensuite écrire dans le même fichier HTML en reformatant les tableaux pour que les métadonnées de chaque colonne d'une table de base de données soit plus espacées, plus lisibles. De plus il va aussi ajouter un entête à chaque colonne (Nom de la colonne, Type de Données, Nullable, Le commentaire).

Le traitement nommé « TraduireContenuHTML » va parcourir le fichier HTML et traduire en Français les mots en Anglais.

Exemple :

**Avant**

---

### Rapport Modèle de Données

#### System Information

generated by	SchemaCrawler 16.19.7
generated on	2023-03-22 11:03:50

public.categorie_cat		[table]
cat_id	serial not null	
	auto-incremented	
cat_code	varchar(20)	
cat_description	text not null	
	La description blablabla	
<b>Primary Key</b>		
categorie_cat_pk		[primary key]
cat_id		
<b>Foreign Keys</b>		
produit_pro_fk1		[foreign key, with no action]
cat_code ←(0..many)	public.produit_pro.pro_cat_code	
<b>Indexes</b>		
categorie_cat_pk		[unique index]
cat_id	ascending	
categorie_cat_cat_code_key		[unique index]
cat_code	ascending	

## Après

### Rapport Modèle de Données

#### Informations

Nom de la base de donnée	Nom Base De Données PostgreSQL
Description de la base de données	Lorem ipsum dolor sit amet consectetur adipiscing elit. Debitis accusamus aliquam fugit consequuntur repellendus sit nihil doloribus unde sint commodi, consectetur laboriosam quibusdam voluptates libero eligendi similique dolor, accusantium in.



**public.categorie\_cat** [table]

Nom	Type	Nullable	Commentaire
<b>cat_id</b>	serial	not null	
	auto-incremented		
cat_code	varchar(20)		
cat_description	text	not null	La description blablabla Le Lorem Ipsum est simple ment du faux texte emp is e en page avant impressio n. Le Lorem Ipsum est le faux texte standard de li mprimerie depuis le éa ise r un mcemment, par son in clue texte, comme Aldus P ageMaker.

**Primary Key**

<b>categorie_cat_pk</b>	[primary key]
cat_id	

**Foreign Keys**

<b>produit_pro_fk1</b>	[foreign key, with no action]
cat_code ←(0..many) public.produit_pro.pro_cat_code	

**Indexes**

f. Conversion du fichier HTML en PDF

Une fonction nommée « `ConversionHtmlPdf(<CheminRepertoireTravail>, <CheminRepertoireModeleDonnees> )` » va convertir un fichier HTML en PDF.

## 10. Fichier config à la base du projet

Un fichier config nommé « config.properties » à la base du projet sert à définir plusieurs paramètres à renseigner manuellement par l'administrateur ou autre :

- CheminRepertoireTravail : Le répertoire qui contiendra les fichiers retour du traitement de schemacrawler
- CheminRepertoireModeleDonnees : Le répertoire qui contiendra le fichier PDF résultat généré par le traitement.
- CheminRepertoireLogs : Le répertoire qui contiendra les fichiers logs contenant le déroulement du traitement

## 11. Gérer les logs de l'application

A chaque action, étape du programme une méthode nommée `EcritureLogs(Type, Message)` sera appelée pour écrire dans un fichier logs l'état d'avancement du programme. Pour écrire les logs dans un fichier, un fichier config doit être initialisé au format XML. Ce fichier permet de formater la sortie des informations. Il est possible de récupérer dans le fichier une variable externe. Une variable dans un fichier configuration par exemple. A chaque lancement de l'application un nouveau fichier logs est créé

Exemple :

*Le fichier config xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="DEBUG">
  <Properties>
    <Property name="CheminRepertoireLogs">Logs/</Property>
  </Properties>
  <Appenders>
    <File name="MyFile" fileName="${CheminRepertoireLogs}mylogfile-${date:yyyy-MM-dd-HH-mm-ss}.log">
      <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
    </File>
  </Appenders>
  <Loggers>
    <Root level="DEBUG">
      <AppenderRef ref="MyFile"/>
    </Root>
  </Loggers>
</Configuration>
```

*Le fichier log*

```
10:31:22.371 [main] INFO  Accueil - Procédure vérification fichier config
10:31:22.379 [main] INFO  Accueil - CheminRepertoireLogs du fichier config est correct
10:31:22.380 [main] INFO  Accueil - CheminRepertoireTravail du fichier config est correct
10:31:22.380 [main] INFO  Accueil - CheminRepertoireModeleDonnees du fichier config est correct
10:31:22.380 [main] INFO  Accueil - Chargement fichier config réussi
10:31:22.380 [main] INFO  Accueil - Procédure vérification fichier config terminé
```

*Heure | Type | Message*

### a. Au lancement de l'application

- Message : Procédure vérification fichier log4j2

*# Un des deux messages peut survenir*

Message : Fichier log4j2 introuvable

Message : Chargement du fichier log4j2 réussi

Message : Procédure vérification fichier log4j2 terminée

- Message : Procédure vérification fichier config

*# Un des deux messages peut survenir*

Message : Fichier config introuvable

Message : Chargement fichier config réussi

*# Un des deux messages peut survenir*

Message : Variable CheminRepertoireLogs du fichier config est manquante ou invalide

Message : Variable CheminRepertoireLogs du fichier config est correct

*# Un des deux messages peut survenir*

Message : Variable CheminRepertoireTravail du fichier config est manquante ou invalide

Message : Variable CheminRepertoireTravail du fichier config est correct

*# Un des deux messages peut survenir*

Message : Variable CheminRepertoireModeleDonnees du fichier config est correct

Message : Variable CheminRepertoireModeleDonnees du fichier config est manquante ou invalide

Message : Procédure vérification fichier config terminée

### b. Lors d'un clic sur le bouton Tester connexion

- Message : Procédure tester la connexion lancée

*# Suivant le résultat deux message peut survenir*

Message : Connexion à la base de données réussit

Message : Echec de la connexion à la base de données

Message : Procédure tester la connexion terminée

### c. Lors d'un clic sur le bouton Lancer la procédure

- Message : Procédure tester la connexion lancée

*# Suivant le résultat un des deux messages peut survenir*

Message : Connexion à la base de données réussit

Message : Echec de la connexion à la base de données

Message : Procédure tester la connexion terminée

- Message : Procédure Exécuter schemacrawler lancée

Message : Construction de la ligne de commande

Message : Exécution de la ligne de commande

*# Suivant le résultat un des deux messages peut survenir*

Message : Fichiers générés dans le répertoire Travail

Message : Echec de l'exécution de la ligne de commande

Message : Procédure Exécuter schemacrawler terminée

- Message : Procédure Interprétation HTML Lancée

Message : Début formatage HTML

*# Suivant le résultat un des deux messages peut survenir*

Message : Echec du formatage HTML

Message : Formatage HTML réussie

Message : Début Traduction HTML

*# Suivant le résultat un des deux messages peut survenir*

Message : Echec de la traduction HTML

Message : Traduction HTML réussie

Message : Procédure Interprétation HTML terminée

- Message : Procédure de conversion du fichier HTML en PDF lancée

*# Suivant le résultat un des deux messages peut survenir*

Message : Echec de la conversion

Message : Conversion réussie – Document PDF présent dans le répertoire ...

Message : Procédure de conversion du fichier HTML en PDF terminée

## 12. Annexe :

Les éléments de connexion à la base de données Test du projet

Type base : postgresql

Adresse du serveur : localhost

Port : 5432

Nom de la base : postgres

Identifiant : postgres

Mot de passe : superadmin

Schéma : public