

Haehnel Jonathan
Knobloch Thomas
Armand Julien

« Préparation à la fouille de données »

Tuteur : Nicolas Lachiche
Ref : T305ALAN

Cahier des charges



Année 2010-2011

IUT Robert Schumann – Informatique

Sommaire :

Introduction	Page 4
Etude des fonctionnalités du projet	Page 5-6
• Intégration des bases de données	Page 5
• Intégration des fichiers	Page 6
• Support	Page 6
Contraintes générales	Page 7
• Contraintes de temps	
• Contraintes techniques	
• Contraintes de portabilité	
• Environnement de développement	
Contraintes de réalisation	Page 8
Planning prévisionnel	Page 9-13
• Planning général	Page 9-11
• Diagramme de Gantt	Page 12-13
Maquette et analyse	Page 14-22
• Connexion à une base de données	Page 14
• Aperçu des différents menus	Page 15
• Extraction d'une sous-table	Pages 16-18
• Copie d'une base	Page 19
• Ouverture d'un fichier	Page 20
• Créer une base vierge	Page 20
• Enregistrement d'une sous-table dans une base	Page 21

Introduction:

Sujet :

Le projet consiste à réaliser une application java permettant de préparer les données pour plusieurs programmes de fouille de données. Cet utilitaire devra pouvoir manipuler et extraire des données de certaines bases de données et certains fichiers. Une interface graphique simple mais ergonomique est attendue.

Demandeur :

Mr Lachiche est le responsable du projet, mais également le maitre d'ouvrage.

Contexte :

Le projet s'insère dans le contexte de la fouille de données (recherche d'un modèle à partir d'un échantillon de données). Il va permettre de faciliter ce processus en sélectionnant à l'avance certaines données dans les bases et dans les fichiers. A la main, ce type d'opération est plutôt long et laborieux, il est donc important d'automatiser cette tâche.

Limite du sujet :

Le système permettra d'extraire des données pour l'élaboration d'un modèle, mais en aucun cas, l'apprentissage de ce modèle n'est attendu dans ce projet.

Type de sujet :

Ce projet demande une petite analyse sur la structure des fichiers et des bases de données à intégrer dans notre application, mais c'est surtout du développement. Il faut en effet développer une interface graphique permettant de se connecter à des bases de données, d'ouvrir des fichiers spécifiques pour générer de nouvelles bases ou fichiers.

Notre projet devra pouvoir s'intégrer dans un projet déjà existant, les bases de données et les fichiers d'entrés existent déjà.

Etudes des fonctionnalités :

Voici, une liste des principales fonctionnalités de notre application. Celle-ci ne sont pas définitives, des modifications pourront être apportées pendant le développement. Chaque fonctionnalités est classé par ordre d'importance :

P1 : fct. Essentielle > **P2** : fct. Complémentaire > **P3** : fct. De confort/Si possible

a) Intégration des bases de données :

Le cœur de notre projet est la manipulation de différentes bases de donnée et la création de nouvelles tables dans celle-ci.

Fonctionnalité	Priorité	Commentaires
Connexion à une/ou plusieurs base de données	P1	Connexion à une base de donnée MySQL ou PostgreSQL en utilisant la librairie JDBC. Il est possible d'intégrer les bases en entrée pour extraire des données ou en sortie pour enregistrer les données formatées.
Sélection d'une table à formater	P1	Les méta-données de la base ont été préalablement générer
Extraction d'une sous-table	P1	Il faudra générer les méta-données de la table sélectionnée pour en créer un aperçu, et permettre à l'utilisateur de sélectionner certaine ligne et champs de ce table.
Mise à jour automatique de l'aperçu de la sous-table	P1	Une mise à jour sera effectuée après chaque modification de l'utilisateur.
Ajouter un champs	P2	Possibilité de rajouter des champs dans la sous-table.
Renommer un champs	P2	Possibilité de renommer des champs dans le sous-ensemble de la table.
Enregistrement de la sous-table	P1	Enregistrement dans une base de sortie ouverte (étape 1)
Création d'une nouvelle table vide.	P1	Création avec connexion automatique sur celle ci.
Copie « Base à Base »	P1	Copie direct (sans formatage) après le choix de certaine tables d'une base d'entrée à une base de sortie
Exporter une base vers un fichiers de données	P2	Enregistrement dans un fichier formaté ACE, HiFi, Tertius.
Déconnexion	P1	Fermeture de la connexion à une base de donnée.

Etudes des fonctionnalités :

b) Intégration des fichiers de données :

En plus, de manipuler des bases de données, nous devons également intégrer trois types de fichiers (ACE, HiFi, Tertius) contenant des données formatées de façon différente.

Fonctionnalité	Priorité	Commentaires
Ouverture des fichiers	P1	/
Convertir en base de données	P1	Convertir le fichier ouvert en base de données (d' entrée) pour mieux pouvoir l'exploiter.
Extraction d'une sous-table	P1	Sur la base obtenue après conversion, il sera possible comme pour les base de données normal de sélectionner les lignes/colonnes et de rajouter/renommer des colonnes.
Enregistrement du nouveau contenu dans un fichier de même formatage.	P1	Le nouveau contenu signifie la nouvelle base de sortie créé à partir du fichier d'entrée.
Enregistrement dans un fichier de formatage différent.	P2	/
Enregistrement dans une base de données	P2	/
Génération des fichiers d'en-têtes	P3	Génération des fichiers d'entêtes contenant l'algorithme de formatage des 3 types de fichiers. (par exemple , fichier .S pour les fichier ACE)

c) Support:

En fin de projet, nous devons faire une support d'aide en ligne.

Fonctionnalité	Priorité	Commentaires
Aide en ligne	P1	Simple page de présentation du logiciel
Doc java	P3	Documentation java de nos propres classes
Support général en ligne	P3	Site plus évolué avec FAQ, questions courantes, présentation, tutoriels (vidéos ?), page contact.

Contraintes générales :

a) Contraintes de temps :

- Le logiciel devra respecter un certain calendrier (cf. Page 8), dont les principales échéances sont :
 - ➔ 15 octobre 2010 : Rendu du cahier des charges
 - ➔ 6 décembre 2010 : Rendu du premier prototype.
 - ➔ 10 au 12 Janvier 2011 : Développement
 - ➔ 13 Janvier 2011 : Livraison du projet
 - ➔ 14 Janvier 2011 : Soutenance
- Notre temps de travail individuel devra être organisé logiquement et équitablement durant toute la durée du projet. Nous allons essayer de maintenir un minimum de 5 heures par semaine par personne.

b) Contraintes techniques :

- Réalisation du logiciel en Java
- Interface simple mais ergonomique
- L'application doit pouvoir s'insérer dans un projet existant
- Les bases de données et les fichiers d'entrées sont existants
- Gérer les différents types de formatage des fichiers utilisés
- Générer toutes les méta-données

c) Contraintes de portabilités :

- Étant développé en Java, notre logiciel sera portable sur la majorité des systèmes d'exploitation ayant une machine virtuelle Java active.
- Notre logiciel devra néanmoins être compatible avec toute base de donnée MySQL et PostgreSQL, et avec tous les fichiers de données Ace, HiFi et Tertius.
- L'application devra être compatible avec l'outil de gestion déjà existant, car nous devons pouvoir les emboîter les uns dans les autres.

d) Environnement de développement :

- Type de base de donnée à utiliser : MySQL et PostgreSQL
- Langage de développement : Java
- Librairie de gestion des bases de données : JDBC
- Librairie Graphique utilisée : Swing , AWT
- Logiciel de développement : NetBeans , Éditeur de texte
- Type de logiciel : Utilitaire de gestion de base de données

Contraintes de réalisation :

- Les principaux acteurs de notre projet seront :
 - le **maître d'ouvrage** : *Mr. Lachiche* qui est également notre tuteur de projet.
 - le **maître d'œuvre** : *Jonathan Haehnel*, le chef du projet.
 - le **comité de pilotage** : *Mr Lachiche et Jonathan Haehnel*
 - l'**équipe de développement** : *Jonathan Haehnel, Julien Armand et Thomas Knobloch.*
- Les rendez-vous entre Tuteur et Groupe se programment après chaque entretien. Pour le moment ils se font le lundi à 15h30. Il est nécessaire de faire un compte-rendu après chaque réunion à faire pour la suivante afin que le tuteur puisse répondre à d'éventuelles questions ou incompréhensions du sujet.
- Le cycle de vie de notre projet est le cycle de vie **ICARE** :

Phase	Description
I nitialiser	Tout d'abord, chaque projet commence par être initialisé . Cette phase de son cycle de vie correspond à son lancement, au moment où le sujet est choisi. Nous avons dans cette phase formé notre trinôme, car nous avons tous un certain intérêt pour le sujet.
C ontractualiser	Ensuite, un projet est contractualisé. Durant cette phase, il est rédigé un cahier des charges. De plus, les tâches à effectuer pour la réalisation de ce dernier sont planifiées et réparties entre les différents membres du trinôme.
A nalysier	Vient ensuite la période d'analyse, durant laquelle s'effectuent la plupart des travaux de recherches et d'analyses. On va mettre au point un premier prototype, fonctionnel, afin d'avoir une base solide du projet.
R éaliser	Après l'analyse, le projet est réalisé , c'est-à-dire que c'est à ce moment là qu'est effectuée la programmation, ainsi que les tests associés. Une documentation explicative, si le projet en nécessite une, peut aussi être également fournie à ce moment là.
E xploiter	Pour finir, la phase qui clos le projet est celle consistant en l' exploitation de ce dernier. Le projet, finalisé, est livré au demandeur. A cette occasion, nous passerons une soutenance durant laquelle nous présenterons notre projet devant un jury.

Planning prévisionnel :

a) Planning général :

ETAPE 1: Lancement du projet – Réunions				
<i>Date</i>	<i>Objet - Tache</i>	<i>Travail attendu</i>	<i>Ressources</i>	<i>Temps estimé</i>
Vendredi 03 septembre 2010	Lancement du projet	- Choix des groupes - Choix du sujet - Premier contact avec le tuteur	X	X
Judi 09 septembre 2010	Première réunion avec le tuteur	- Explication du sujet et de la méthode de travail - Planification et estimation du temps de travail idéal.	X	X
Pour chaque réunion	Compte-rendu de la réunion précédente	- Résumer de ce qui a été dit lors de la réunion précédente avec le tuteur. - Trouver les questions à éclaircir lors de la prochaine réunion.	Tous (alternance)	1h/ compte-rendu
ETAPE 2: Cahier des charges et maquette graphique (15/09 - 15/10)				
<i>Date</i>	<i>Objet - Tache</i>	<i>Travail attendu</i>	<i>Ressources</i>	<i>Temps estimé</i>
15 septembre 2010	Réalisation du cahier des charges (lancement)	<u>Jonathan</u> : Partie 1 – Partie 2 – Partie 5 <u>Thomas</u> : Partie 4 – correction orthographique <u>Julien</u> : Partie 3 – Partie 2	Jonathan, Thomas, Julien	6 jours / personne
27 septembre 2010 – 15 octobre 2010	Réalisation d'une première interface graphique	- Réalisation de certaines pages par chaque membre - Mise en commun - Intégration de la maquette finale dans le cahier des charges	Jonathan, Thomas, Julien	5 jours / personne
15 octobre 2010	Rendu du cahier des charges	- Finalisation du cahier des charges - Interface graphique validé	X	X

par le tuteur

ETAPE 3: Prototype et développement (16/10 - 31/12)

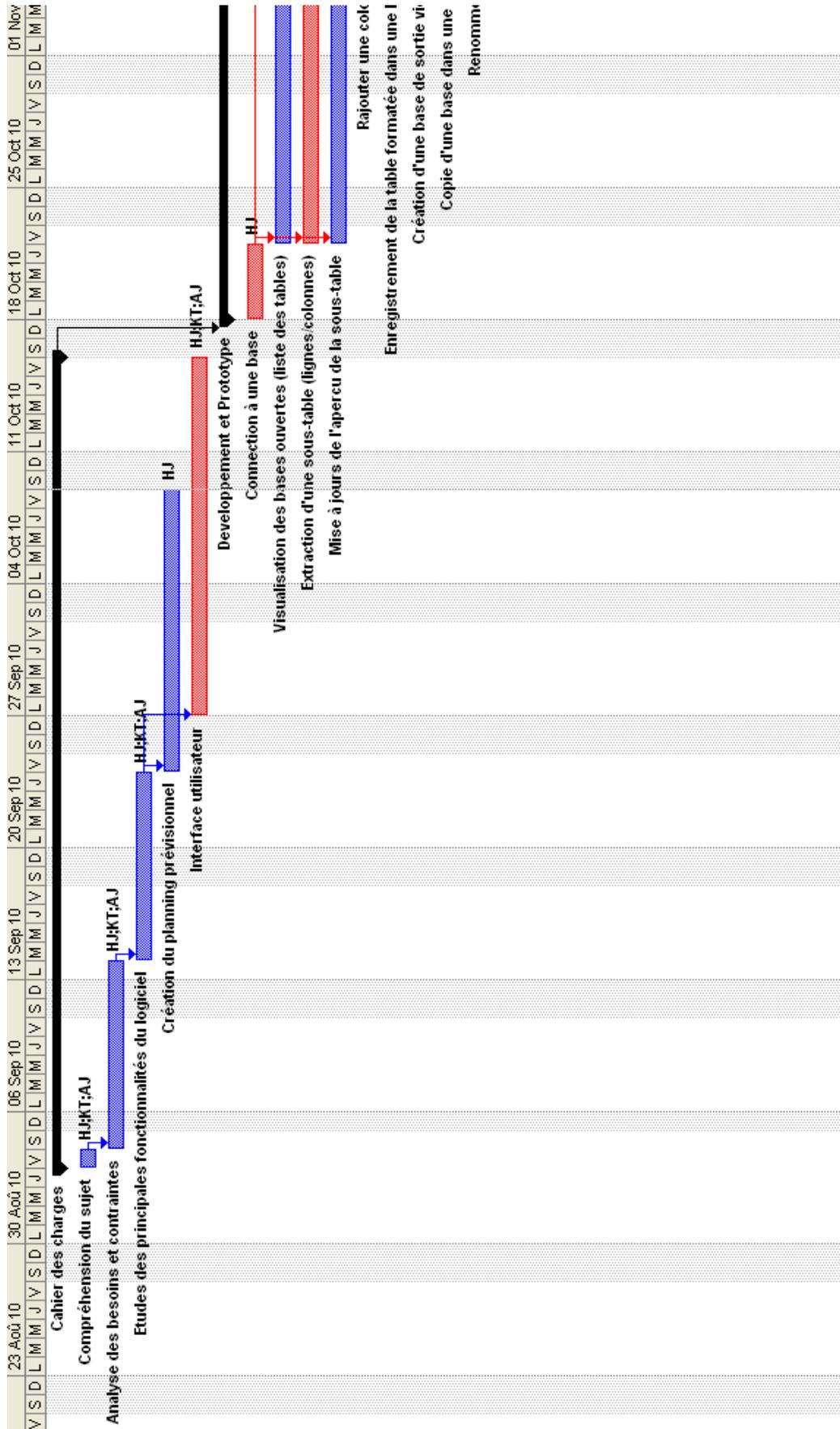
<i>Date</i>	<i>Objet - Tache</i>	<i>Travail attendu</i>	<i>Ressources</i>	<i>Temps estimé</i>
22 octobre 2010	Connexion aux bases de données	- Ouverture d'une ou plusieurs connexions - Mise en mémoire des connexions actives	Jonathan	3 jours
5 novembre 2010	Visualisation des tables des bases ouverte	- Lecture des tables des bases ouvertes et enregistrement - Aperçu de la liste des tables.	Jonathan	4 jours
	Sélection d'une table et extraction d'une sous-table	- Sélection d'une table de la liste. - Choix des colonnes à conserver. - Choix des lignes à conserver.	Julien	1 semaine
	Affichage de l'aperçu de la sous-table	X	Thomas	5 jours
10 novembre 2010	Ajout d'une colonne dans la table	- Ajout d'une colonne en fin de table	Thomas	3 jours
	Création d'une table vierge de sortie	X	Julien	2 jours
20 novembre 2010	Renommer un colonne de la sous-table	X	Thomas	4 jours
25 novembre 2010	Copie d'une base d'entrée dans une base de sortie	- Choix des tables à copier - Module de copie direct	Julien	6 jours
	Enregistrement de la table dans une base de sortie ouverte	- Enregistrement de la table extraite dans une base - Nommer la table avant enregistrement	Jonathan	7 jours
30 novembre 2010	Ouverture d'un fichier	- Ouverture d'un fichier (Ace, Tertius, HiFi)	Julien, Jonathan, Thomas	3 jours
5 décembre 2010	Transformer les fichier de données en données plus exploitables	- Transformer les fichiers en tables	Julien, Jonathan, Thomas	1 semaine / personne

	(tables)			
6 décembre 2010	Rendu du prototype	- Présenter l'avancement de notre projet et démonstration du prototype.	X	X
15 décembre 2010	Extraire des données d'un fichier	- Intégration des fichiers dans le système d'extraction des lignes / colonnes.	Jonathan et Julien	1 semaine
30 décembre 2010	Convertir une base en fichier de données	- Exporter une base de données en fichier de données. - Choix du répertoire et nom/type fichier de destination. - Génération du code formaté selon le type de fichier sélectionné	Julien, Jonathan, Thomas	1 semaine / personne

ETAPE 4 : Tests, Finalisation et Soutenance (01/01 - 14/01)

<i>Date</i>	<i>Objet - Tache</i>	<i>Travail attendu</i>	<i>Ressources</i>	<i>Temps estimé</i>
7 janvier 2011	Finalisation du développement	- Tests - Optimisation (saisie contrôlée...) et résolution d'éventuels bugs	Julien et Jonathan	3 jours
	Rédaction de l'aide en ligne	X	Thomas	3 jours
10 au 14 janvier 2011	Préparation à la soutenance (semaine bloquée)	- Création du PowerPoint - Entraînement à la présentation	Tous	4 jours
14 janvier 2011	Rendu final de l'application et présentation	X	X	X

b) Diagramme de Gantt :



Maquette et analyse :

a) Page de connexion à une base de données :



The screenshot shows a window titled "DBManager" with a menu bar containing "Fichier", "Edition", and "Aide". The main content area is titled "Connexion à une base de donnée :". It contains several input fields and dropdown menus:

- Type de base de donnée :** A dropdown menu with "MySQL" selected.
- Host du serveur :** A text field containing "localhost".
- Port du serveur :** A text field containing "3541".
- Nom de la base :** A text field containing "Entreprise".
- Login :** A text field containing "johnny".
- Password :** A password field with masked characters ".....".
- Définir comme :** A dropdown menu with "Entrée" selected.

At the bottom center, there is a button labeled "Se connecter à la base".

A l'ouverture du logiciel, cette page apparaît, elle permet la connexion à une base de données.

On peut apercevoir que l'on demande à l'utilisateur plusieurs informations. Tout d'abord, on choisit dans une `JcomboBox()` le **type de système de gestion** de la base à ouvrir. (*pour le moment, soit MySQL ou PostgreSQL*)

Puis, l'on saisit dans l'**adresse du serveur** de la base (host) dans un `JtextField()`.

Pour se connecter, il faut également ajouter le port d'accès à la base, chaque système de gestion possède **un port** par défaut, il sera automatiquement placé dans la zone de saisie, mais il sera possible de le modifier manuellement tout de même.

Dans un `JtextField()`, l'on saisit également le **nom de la base** et son **login** de connexion, le **mot de passe** sera quand à lui mis dans un `Jpasswordfield()` pour éviter d'être visible en clair.

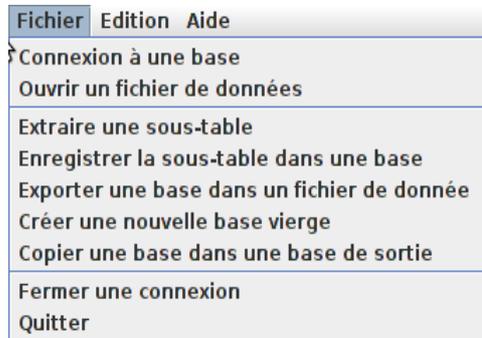
Après, il faut choisir dans une `JcomboBox()`, si la base est prise en compte comme **base d'entrée ou base de sortie** dans l'application. Nous avons choisi de mettre ce type de limitation pour éviter la modification/l'écrasement d'une base lors du formatage de celle-ci.

En cliquant sur le `Jbutton()` « Se connecter », soit la **connexion à réussie**, et alors on enregistre l'objet `BDConnexion()` dans un tableau (`Vector()`) qui contient toutes les connexions ouvertes et on affiche la page d'extraction d'une sous-table (cf. Point b). En revanche, en **cas d'échec**, l'on affiche une boîte de dialogue `JOptionPane(ERROR_MESSAGE)`.

Maquette et analyse :

b) Aperçu des différents menus:

Notre application possède trois différents menus dans sa JMenuBar(). Nous détaillerons leur contenu ci-dessous.



Le menu « Fichier »

Ce menu est le menu principal de notre application, toutes les actions possibles y sont référencées. Le premier JMenuItem() permet l'affichage de la page de **connexion à une base**. Cette page n'est soumise à aucune précondition, vous pouvez vous connecter autant de bases que vous voulez en même temps.

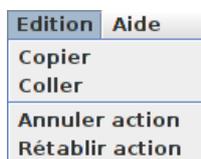
Dans un second temps, le menu suivant permet d'**ouvrir un fichier**. Comme son prédécesseur, il n'est soumis à aucune précondition. (plus d'information, partie d)

En revanche, par défaut, la **page d'extraction** et la **page de copie d'une base dans une base de sortie** sont grisées, elles ne peuvent que s'ouvrir s'il y a au moins une base d'entrée et une de sortie ouverte.

Plus complexe, la **page d'enregistrement** de la sous-table dans une base de sortie, grisée par défaut, pour les mêmes raisons que la précédente. Mais, en plus, elle ne s'ouvre que si une table a été formaté en amont.

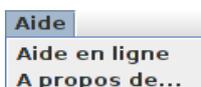
La page de **fermeture d'une connexion** nécessite qu'une connexion soit ouverte.

En revanche, les pages de **création d'une table de sortie vierge** et le **bouton quitter** l'application sont cliquable tout le temps. En cliquant sur « quitter », le programme ferme toutes les connexions en cours et termine l'exécution avec System.exit(0) ;



Le menu « Edition »

Ce menu contient quelques fonctions d'édition de base comme **Copier – Coller** et les fonction **Annuler-Rétablir** plus complexes à coder (Nécessitant un historique). Nous intégrerons ses deux dernières si possible.

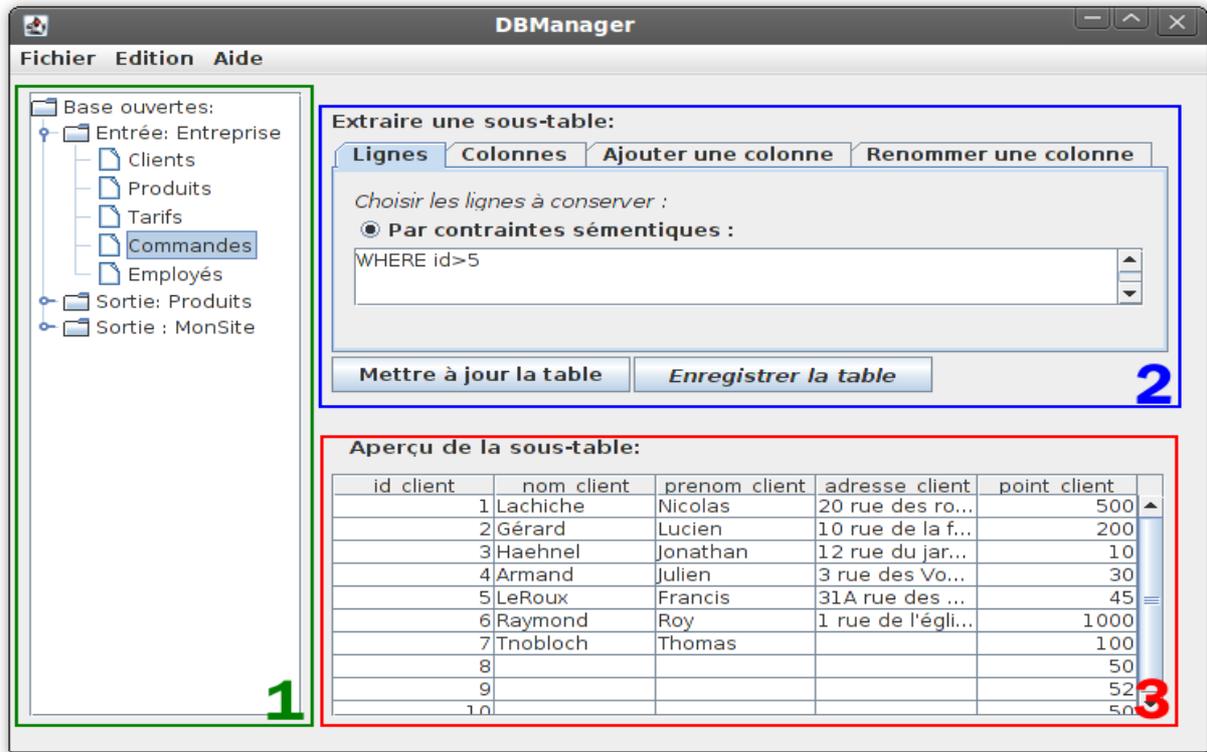


Le menu « Aide »

Ce menu contient deux éléments, le premier vous renvoie sur notre site d'**aide en ligne**. Et, le second ouvre une fenêtre de dialogue, avec les principales **informations techniques** de l'application (brève présentation, crédits, versions, logo).

Maquette et analyse :

c) Page d'extraction d'une sous-table:



Cette page a pour fonction principale de permettre d'extraire une sous-table à partir d'une table préexistante. (En annexe, elle permet aussi d'avoir un aperçu de l'ensemble des tables des bases que l'utilisateur a ouvertes)

Précondition : L'affichage de cette page ne peut se faire que si une base d'entrée et une base de sortie ont été préalablement ouvertes (se référer à la partie « Maquette et analyse » point « a ») pour plus de détails concernant l'ouverture d'une base en entrée ou en sortie).

On constate que la page d'extraction est composée de 3 parties :

- La 1^{ère} partie permet de **sélectionner une base parmi les bases ouvertes** et d'y **choisir une table**. Cette sélection est permise grâce à l'utilisation d'un objet JTree() qui affiche les bases et les tables sous forme hiérarchique.
- La 2nd (ci-dessous) permet, une fois une table sélectionnée, de :
 - **restreindre les lignes et les colonnes** à afficher.
 - **ajouter** une colonne.
 - **renommer** une colonne.Pour ce faire, on utilise un objet JTabbedPane() qui dessine un panneau contenant des onglets. (Plus d'information ci-dessous)

c) Page d'extraction d'une sous-table :

- La 3ème et dernière partie de l'interface affiche l'aperçu de la table actuellement sélectionné. Cette affichage est réalisé grâce à une `JTable()` qui est directement mise à jour par sélection d'une table ou action sur le `JButton()` « Mettre à jour la table ».

1- Sélection des lignes :



Cette onglet permet de restreindre l'**affichage des lignes d'une table à partir de contraintes sémantiques** saisies dans un objet `JTextArea()`.

Deux `JButton()` sont également présents l'un « Mettre à jour la table » met directement à jour la table dans l'aperçu , l'autre , « Enregistrer la table » ouvre la page d'enregistrement d'une table (Ces 2 `JButton()` sont présents sur tout les onglets de cette partie et conservent la même fonction).

2- Sélection des colonnes:



Cette onglet permet comme le précédent, de restreindre l'affichage mais cette-fois ci, **des colonnes**. Cette restriction est faite par l'intermédiaire de `JCheckBox()`. Une `JCheckBox()` est attribué à chaque colonne de la table et porte son nom. Le statut sélectionné ou non, conditionne les tables à conserver ou non.

3- Ajouter une colonne:



c) Page d'extraction d'une sous-table :

Cet onglet permet d'**ajouter une colonne**. D'abord l'utilisateur entre le nom de la nouvelle colonne grâce à un JTextField(). Ensuite, il sélectionne le type de cette colonne dans une ComboBox() contenant la liste des types de données possibles. Et enfin l'utilisateur clique sur le JButton() « Ajouter » , qui ajoute la nouvelle colonne en dernière position dans la table.

4- Ajouter une colonne:

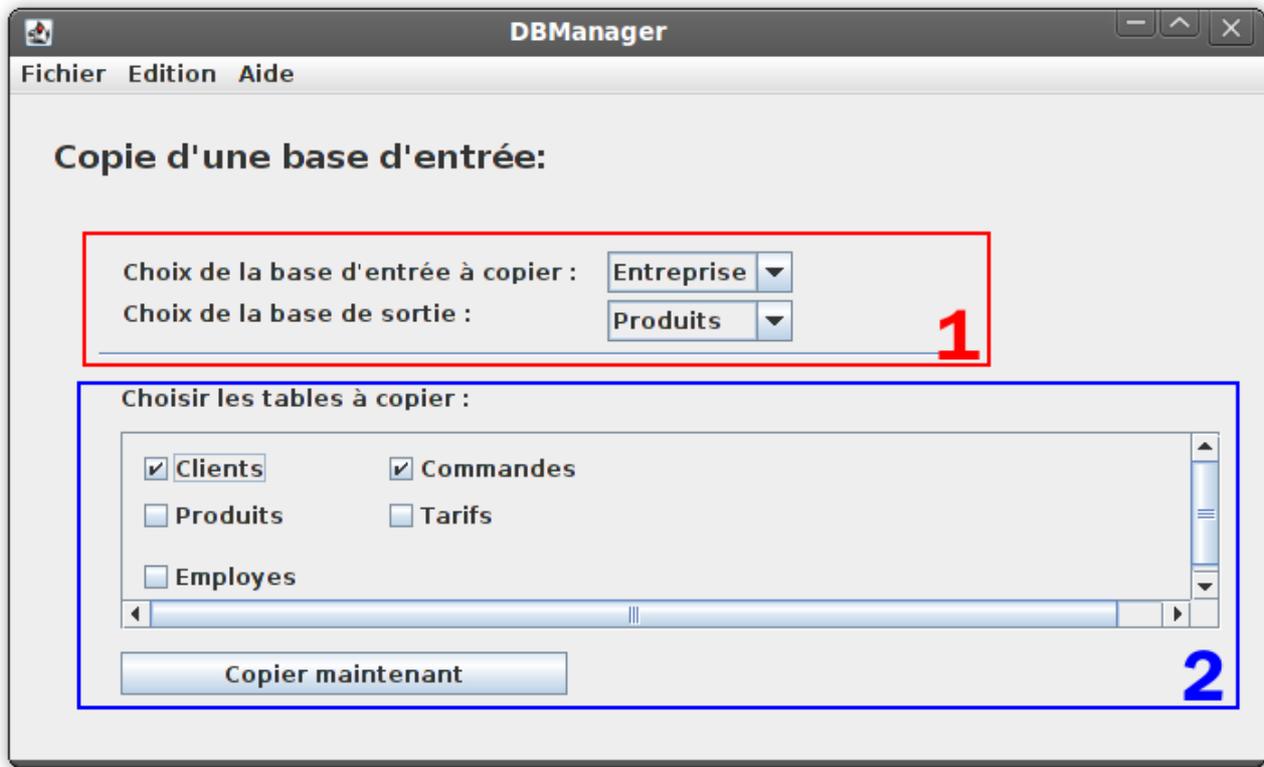


The screenshot shows a software interface with four tabs: 'Lignes', 'Colonnes', 'Ajouter une colonne', and 'Renommer une colonne'. The 'Renommer une colonne' tab is active. It contains a label 'Choisir la colonne à renommer :' followed by a dropdown menu with 'id_client' selected. Below this is a label 'Nouveau nom :' followed by an empty text input field. A 'Renommer' button is positioned below the input field. At the bottom of the interface, there are two buttons: 'Mettre à jour la table' and 'Enregistrer la table'.

Cette onglet permet de **renommer une colonne**. Dans un premier temps l'utilisateur sélectionne dans une JComboBox() la colonne (de la table actuellement ouverte) à renommer. Puis dans une JTextField() le nouveau nom de la colonne. Puis pour finir, il clique sur le JButton() « Renommer » , qui renomme la colonne.

Maquette et analyse :

d) Copie d'une base vers une autre:



La fenêtre permettant de **copier le contenu sélectionné d'une base d'entrée dans une base de sortie** est composé de deux JComboBox() permettant de choisir la base « source » et la base « cible ».

Lorsque ces deux éléments sont complétés la **seconde partie de la fenêtre se met à jour**. La seconde partie de la fenêtre permet de sélectionner des tables de la base « source » à la base « cible » à l'aide de JcheckBox() contenus dans un JScrollPane().

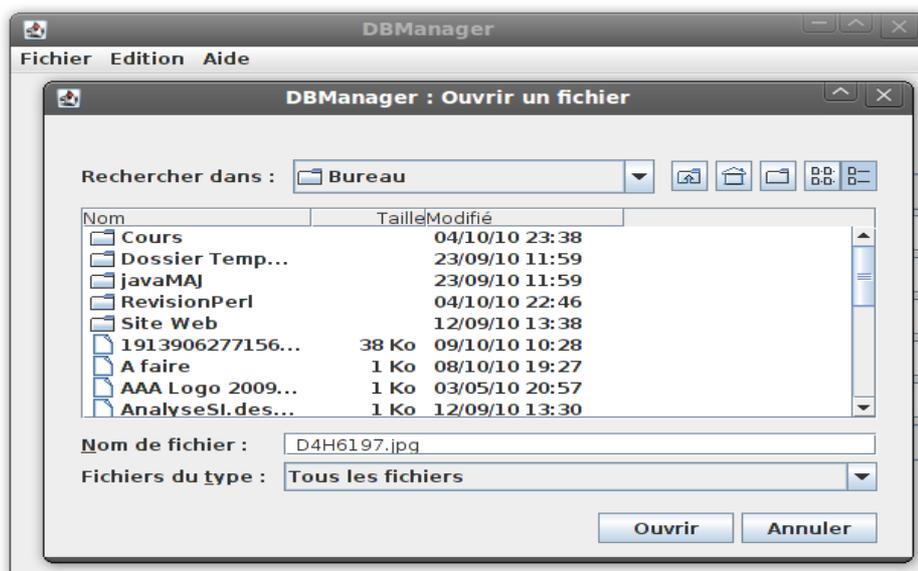
Maquette et analyse :

e) Création d'une base vierge :



La **création d'une base vierge** se fait dans une boîte de dialogue (Jdialog()) dans laquelle on saisit le nouveau nom dans un JTextField(). La création de nouvelles base se fera en utilisant JDBC et des requêtes SQL.

f) Ouverture d'un fichier de données :



L'**ouverture d'un fichier de données** se fera en cliquant sur « Ouvrir un fichier de données » présent dans la catégorie « Fichier » de la JmenuBar(). Cette action entrainera l'ouverture d'une Jdialog() dans laquelle un JfileChooser() permettra de rechercher le fichier souhaité et de l'ouvrir. L'ouverture du fichier se fait à partir de l'utilisation de la classe File et du constructeur `File(String parent, String child)`. Extrait de la documentation java de ce constructeur : « Creates a new File instance from a parent pathname string and a child pathname string. ».

Si l'ouverture du fichier échoue, un message d'erreur sera signalé à l'utilisateur dans une nouvelle boîte de dialogue, l'objet utilisé pour cette affichage d'erreur sera `JOptionPane(JOptionPane.ERROR_MESSAGE)`.

Maquette et analyse :

g) Enregistrement de la sous-table formaté dans une base de sortie :



Lorsque l'utilisateur a effectué une sélection sur les tables, en cliquant sur le bouton « Enregistrer la table », une `JDialog()` apparaît à l'écran. Celle-ci, permet à l'utilisateur d'**enregistrer la sous-table dans la base de sortie** de son choix en donnant le nom de la future table dans une `JTextField()`. Par défaut, le nom de l'ancienne table est choisie. Les bases de sorties disponibles sont représentées dans une `JComboBox()`.

Pour pouvoir enregistrer une sous-table, il faut obligatoirement avoir une base de sortie ouverte et logiquement avoir extrait une sous-table d'une base d'entrée.