



## UNIVERSITE DE FIANARANTSOA ECOLE NATIONALE D'INFORMATIQUE

---

Rapport de projet correspondant au cours de Système d'Information Décisionnel

Intitulé :

# BENETL

Présenté par :

- ANDRIARILALAO Johny Lino,
- RAFALIMANANTSOA Heritiana Lucka Erickson,
- RAKOTO ANDRIANAINA Ariela Cyrille,
- RAKOTOMALALA Mendrika Hajaina

Tuteur encadreur:

- Dr RAKOTOASIMBAHOAKA Cyprien Robert

Annee universitaire 2021 – 2022

## SOMMAIRE

SOMMAIRE .....	I
NOMENCLATURE .....	II
LISTE DES FIGURES.....	III
INTRODUCTION .....	1
PARTIE I. PRESENTATION .....	2
Chapitre 1. Présentation de l'outils .....	2
1.1. Entrepôt de données ou « Datawarehouse » .....	2
1.2. ETL ou « Extract Transform Load » .....	3
Chapitre 2. Description de l'application.....	5
2.1. Présentation de BenETL.....	5
2.2. Technologies utilisées.....	5
PARTIE II. ANALYSE .....	6
Chapitre 3. Analyse .....	6
3.1. Evaluation générale .....	6
3.2. Avantages .....	6
3.3. Inconvénients .....	6
PARTIE III. REALISATION.....	7
4.1. Installation et configuration .....	7
4.2. Mise en œuvre.....	8
CONCLUSION .....	15
BIBLIOGRAPHIE ET WEBOGRAPHIES .....	IV
TABLE DES MATIERES .....	V
RESUME .....	VI
ABSTRACT .....	VI

## NOMENCLATURE

BI: Business Intelligence

COBOL: COmmon Business Oriented Language

CSV: Comma-separated values

ERP: Entreprise Ressource Planning

ETL: Extract Transform Load

SAP: Systems, Applications and Products

SGBD : Système de Gestion de Base de Données

SQL: Structured Query Language

XML: Extensible Markup Language

## LISTE DES FIGURES

Figure 1 . Vue d'ensemble de l'architecture d'un entrepôt de données .....	3
Figure 2. Extract Transform Load .....	4
Figure 3. Fonctionnement du BenEtl.....	5
Figure 4. Choix de la langue d'installation et page de bienvenue.....	7
Figure 5. Conditions d'utilisation et taches supplementaires .....	7
Figure 6. Fin de l'installation de BenEtl .....	8
Figure 7. Les dossiers de configurations pour la connexion a la base de donnees.....	8
Figure 8. La librairie requise pour la connexion MySQL.....	8
Figure 9. Choix de la source de donnees.....	9
Figure 10. Interface de gestion des entites de BenEtl.....	9
Figure 11. Interface du choix de format de la date .....	10
Figure 12. interface de choix de la premier ligne à lire .....	10
Figure 13. Interface de configuration des champs.....	11
Figure 14. Interface de configuration des champs.....	12
Figure 15. Interface pour les lignes uniques .....	12
Figure 16. Interface de configuration du script de création .....	13
Figure 17. Interface de configuration du EtlTool .....	13

## INTRODUCTION

Suite à la grandiose évolution progressive de l'informatique, les grandes entreprises ont à gérer de plus en plus de données. L'usage de l'informatique décisionnelle (BI pour Business Intelligence) est alors adopté suite à la recherche des méthodes permettant d'extraire des informations sur ces données.

L'Informatique décisionnelle désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données d'une entreprise en vue d'offrir une aide à la décision et de permettre aux responsables de la stratégie d'entreprise d'avoir une vue d'ensemble de l'activité traitée. En outre, c'est l'informatique à l'usage des décideurs et des dirigeants d'entreprise.

Dans un business intelligence, les données de l'entreprise sont recueillies et sauvegarder dans un data warehouse ou entrepôt de données qui collecte et organise les informations issues des différentes applications des entreprises ainsi que des données économiques générales du secteur d'activité. Des outils informatiques, baptisés ETL ou Extract, Transform and Load permettent cette collecte. BenETL est justement un des outils les plus prisés pour les processus d'ETL.

Afin d'approfondir et d'en connaître davantage sur ce sujet, une étude en trois parties a été menée. La première partie est consacrée à la présentation du Datawarehouse, d'ETL et du BenETL. La deuxième partie est axée sur l'analyse et la troisième et dernière partie s'articule autour de la réalisation.

## PARTIE I. PRESENTATION

### Chapitre 1. Présentation de l'outils

#### 1.1. Entrepôt de données ou « Datawarehouse »

##### 1.1.1. Définition

[1] Un Entrepôt de données (Data Warehouse) est une base de données regroupant une partie ou l'ensemble des données fonctionnelles d'une entreprise. Il entre dans le cadre de l'informatique décisionnelle ; son but est de fournir un ensemble de données servant de référence unique, utilisée pour la prise de décisions dans l'entreprise par le biais de statistiques et de rapports réalisés via des outils de reporting. D'un point de vue technique, il sert surtout à 'délester' les bases de données opérationnelles des requêtes pouvant nuire à leurs performances.

Selon Bill Immon (1996), l'entrepôt de données se définit comme une « Collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support du processus d'aide à la décision ».

##### 1.1.2. Historique

Les principales dates à retenir construisant l'histoire de l'entrepôt de données sont les suivantes :

- Années 1960 - General Mills et l'Université Dartmouth, dans un projet conjoint, créent les termes faits et dimensions.
- 1983 - Teradata introduit dans sa base de données managériale un système exclusivement destiné à la prise de décision.
- 1988 - Barry Devlin et Paul Murphy publient l'article Une architecture pour les systèmes d'information financiers (An architecture for a business and information systems) où ils utilisent pour la première fois le terme Datawarehouse.
- 1990 - Red Brick Systems crée Red Brick Warehouse, un système spécifiquement dédié à la construction de l'entrepôt de données.
- 1991 - Bill Inmon publie Building the Data Warehouse (Construire l'entrepôt de données).
- 1995 - Le Data Warehousing Institute, une organisation à but lucratif destinée à promouvoir le data warehousing, est fondé.
- 1996 - Ralph Kimball publie The Data Warehouse Toolkit (La boîte à outils de l'entrepôt de données).

##### 1.1.3. Principe de fonctionnement

Dans les faits, les données alimentant l'Entrepôt de données sont hétérogènes, issues de différentes applications de production, voire de fichiers dits "plats" (fichiers Excel, fichiers texte, XML...) [1]. Il s'agit alors de les intégrer, de les homogénéiser et de leur donner un sens unique compréhensible par tous les utilisateurs. La transversalité recherchée sera d'autant plus efficace que le système d'information sera réellement intégré dans sa globalité. Cette intégration nécessite notamment :

- une forte activité de normalisation et de rationalisation, orientée vers la qualité ;
- une bonne gestion des référentiels, incluant une vérification constante de leur intégrité ;
- une parfaite maîtrise de la sémantique et des règles de gestion des métadonnées manipulées.

La problématique de l'intégration repose sur la standardisation de données internes à l'entreprise, mais aussi des données externes (provenant par exemple de clients ou de fournisseurs).

L'historisation d'un Datawarehouse repose sur le principe de conservation des données (ou de non-volatilité des données). Afin de conserver la traçabilité des informations et des décisions prises, les

données une fois entrées dans l'Entrepôt sont stables, en lecture seule, non modifiables par les utilisateurs [1]. Une même requête lancée plusieurs fois à différents moments doit ainsi restituer les mêmes résultats. Dès qu'une donnée est qualifiée pour être introduite dans l'Entrepôt de données, elle ne peut donc plus être altérée, modifiée ou supprimée (jusqu'à un certain délai de purge). Elle devient, de fait, partie intégrante de l'historique de l'entreprise.

Le principe de non-volatilité tranche avec la logique des systèmes de production, qui bien souvent remettent à jour les données par « annule et remplace » à chaque nouvelle transaction. Chaque donnée collectée se voit affecter une date ou un numéro de version pour éviter de recouvrir une information déjà présente dans la base de données, et permettre de suivre son évolution au cours du temps. Il y a de cette manière conservation de l'historique.

D'un point de vue fonctionnel, cette propriété permet de suivre dans le temps l'évolution des indicateurs et de réaliser des analyses comparatives (par exemple, les ventes d'une année sur l'autre). De ce fait, dans un entrepôt de données, un référentiel de temps unique est nécessaire.

La figure 1 montre une vue d'ensemble de l'architecture d'un entrepôt de données.

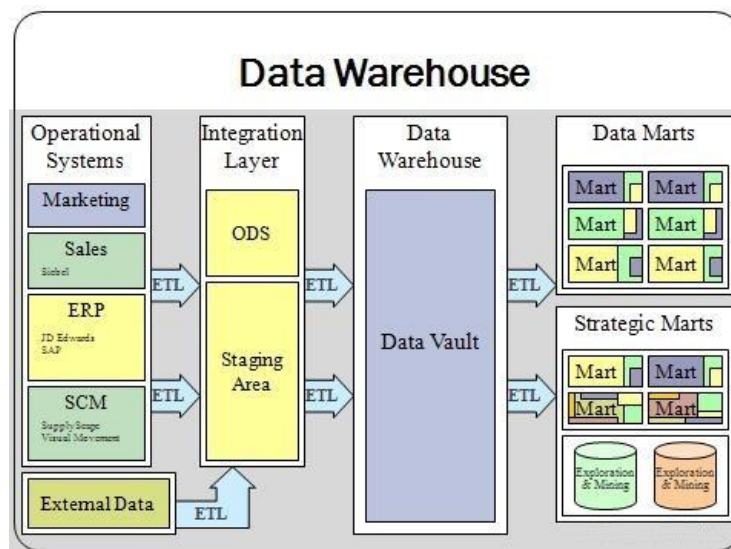


Figure 1 . Vue d'ensemble de l'architecture d'un entrepôt de données

## 1.2. ETL ou « Extract Transform Load »

### 1.2.1. Définition

[2] « Extract-transform-load » est connu sous le terme ETL, ou extracto-chargeur, (ou parfois : datapumping). Il s'agit d'une technologie informatique intergicielle permettant d'effectuer des synchronisations massives d'information d'une source de données (le plus souvent une base de données) vers une autre. Selon le contexte, on est amené à exploiter différentes fonctions, souvent combinées entre elles : « extraction », « transformation », « constitution » ou « conversion », « alimentation ». La figure 2 représente l'ETL.

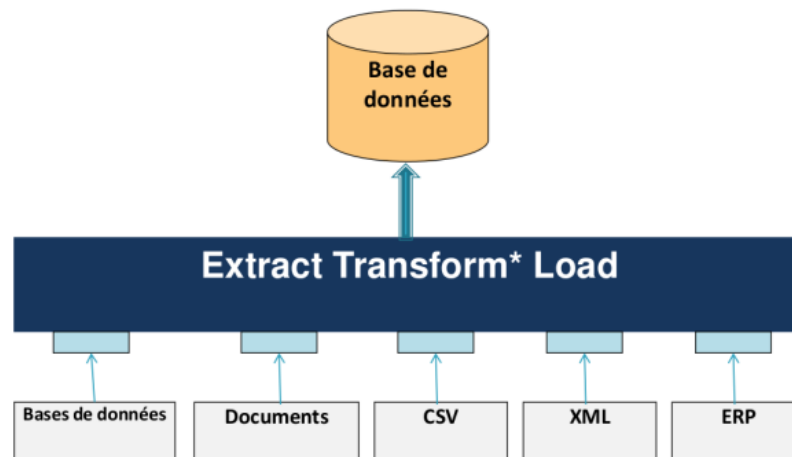


Figure 2. Extract Transform Load

### 1.2.2. Historique

[3] Les ETL sont apparus d'abord dans le monde de la Business Intelligence (ou informatique décisionnelle), leurs utilités étaient de relier les différentes sources de données pour les extraire vers une unique destination. L'adaptation était alors une priorité pour ne pas avoir à modifier les outils existants. Aujourd'hui, les ETL sont de plus en plus utilisés dans le domaine de l'informatique opérationnelle et remplissent un nombre quasi-illimité de possibilités. L'objectif des ETL est aujourd'hui de réduire le coût de développement et de maintenance lors d'un traitement de données.

Depuis l'apparition des ETL, plusieurs générations se sont succédées.

- Dans les années 1970 : Les premiers ETL sont apparus pour communiquer avec les Ordinateurs Centraux (ou mainframe : les premiers ordinateurs faisant la taille d'une salle). Ils permettaient de générer du COBOL, seul langage compréhensible par ces ordinateurs
- Dans les années 1980 : Première interface avec le langage SQL, ceci permet d'ouvrir les fonctionnalités des ETL à une plus large gamme d'outils. Cependant, des logiciels comme SAP n'étant pas ouvert, les connecteurs spécifiques n'existent pas encore. L'échange avec cet outil reste donc limité.
- Dans les années 1990 : Les premières interfaces graphiques apparaissent, l'usage des ETL est donc plus facile. Les connecteurs spécifiques font également leurs apparitions et donc les échanges avec SAP sont plus importants et plus poussés.
- Début des années 2000 : Il est désormais possible de gérer des flux de données en temps réel, ce qui apporte une fonctionnalité importante à cet outil. De plus, l'avènement du langage XML apporte son lot de nouveaux connecteurs.
- Depuis l'année 2005 : L'usage premier depuis cette date a été détourné pour être utilisé en plus de l'informatique décisionnelle dans l'informatique opérationnelle

### 1.2.3. Atouts majeurs des ETL

Les atouts majeurs de l'utilisation des ETL sont les suivantes :

- Initiation facile à des non-informaticiens
- Maintenance moins coûteuse
- Conception rapide
- Lecture/Ecriture des données sur un large choix de supports
- Pas besoin de Data Warehouse pour les utiliser



## Chapitre 2. Description de l'application

### 2.1. Présentation de BenETL

BenETL est un outil d'ETL gratuit développé par Benoît Carpentier depuis 2007 [4], écrit en JAVA et travaillant avec une base de données PostgreSQL ou MySQL. BenETL a été conçu pour fonctionner avec des fichiers source de données .csv ou .txt ou .xls.

Avec BenETL, on peut facilement récupérer des informations à partir de fichiers de données plates, les filtrer et les organiser en fonction de la date (horodatage) et de l'entité.

Pour récupérer des données, il faut définir une entité (quelque chose que l'on veut montrer, calculer), une date (c'est un timestamp, un moment où un événement apparaît).

Benetl propose également un moyen automatique de récupérer les données, en utilisant les tâches planifiées de votre système. La figure 3 illustre le fonctionnement du BenETL.

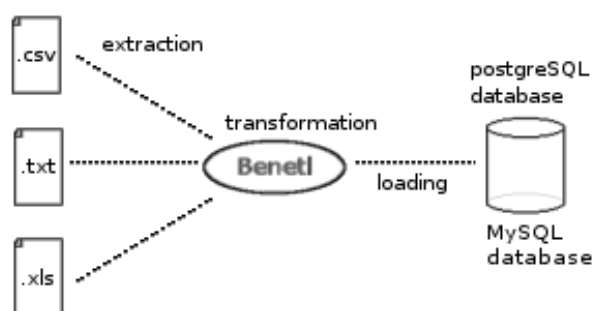


Figure 3. Fonctionnement du BenEtl

### 2.2. Technologies utilisées

Pour le bon fonctionnement du BenETL, les technologies suivantes sont nécessaires :

- Java 8
- Postgresql 9.6.\* (qui supporte plpgsql)
- MySQL 5.6 avec l'ancienne fonction mot de passe pour la connexion de l'utilisateur

Actuellement la version 4.8 peut être utiliser sur linux.

## PARTIE II. ANALYSE

Dans cette partie, nous assisterons à une évaluation générale des caractéristiques du BenETL puis une énumération de ses points forts et ses points faibles.

### Chapitre 3. Analyse

#### 3.1. Evaluation générale

BenETL est un outil d'ETL puissant permettant l'importation en base de données des fichiers de données de type txt, cvs et xls. Un outil qui peut s'avérer utile et indispensable aux entreprises pour le traitement de longs registres et documents car pleinement automatisable dans la création de tâches planifiées selon les systèmes d'exploitation.

Il est facilement téléchargeable sous Windows et linux, léger et peu consommateur de ressources et de mémoire.

#### 3.2. Avantages

BenETL est un outil puissant qui présente les forces suivantes. Tout d'abord, il favorise un gain de temps notable dans le traitement des informations ou données. En effet, aucune intervention manuelle n'est requise pour introduire les données .txt ou .cvs ou .xls dans la base de données.

Ensuite, sa capacité de lecture d'un grand volume de données est remarquable et rapide sans tenir compte de la capacité du matériel. Que votre fichier ait mille ou deux milles lignes, il les lira toujours sans problèmes.

Enfin, BenETL est très souple aux instructions que son administrateur lui soumet. Il est effectivement possible de le programmer à faire une lecture à telle ou telle fréquence souhaitée sur une source de données.

#### 3.3. Inconvénients

Toutes choses ayant ses revers, cet outil présente également des faiblesses. Primo, lors de l'intégration de n'importe quelle donnée dans une base de données, il impose l'inclusion de ses propres tables telles que « entities\_group », « entities\_to\_group » et « etl\_table » dans cette base de données.

Secundo, le type de format de fichier supporté est limité aux trois déjà cités antérieurement, à savoir le .txt, le .cvs et le .xls.

Tertio, malgré sa grande popularité, sa documentation semble rare et non traduites dans diverses langues.

## PARTIE III. REALISATION

### 4.1. Installation et configuration

BenETL est téléchargeable sur le site <https://www.benetl.net>, on nous propose deux liens de téléchargement, un pour la version Windows et un pour la version linux.

Une fois le téléchargement terminé, l'installation se fait en double cliquant sur le fichier binaire «BenetI5.5.0.exe». Les figure 4, 5 et 6 montrent les étapes à suivre pour l'installation de BenEtl.

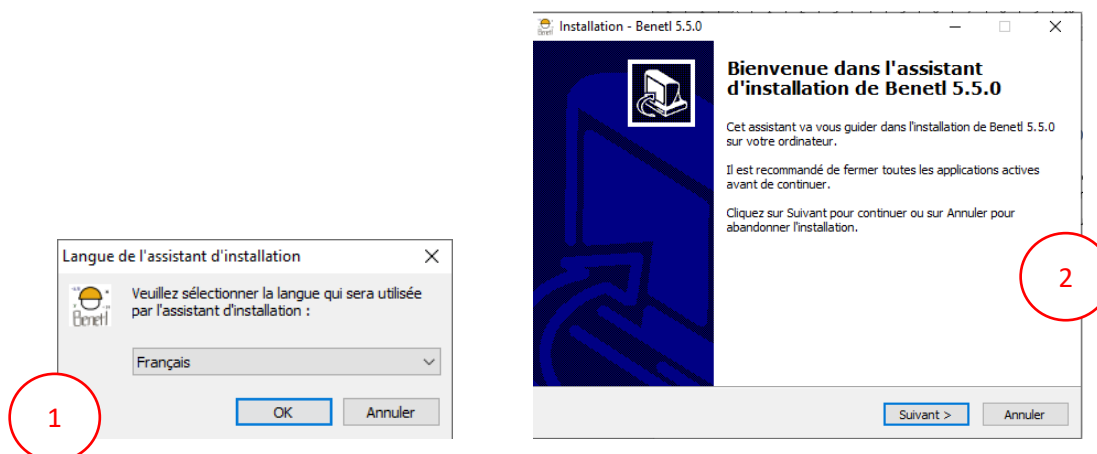


Figure 4. Choix de la langue d'installation et page de bienvenue

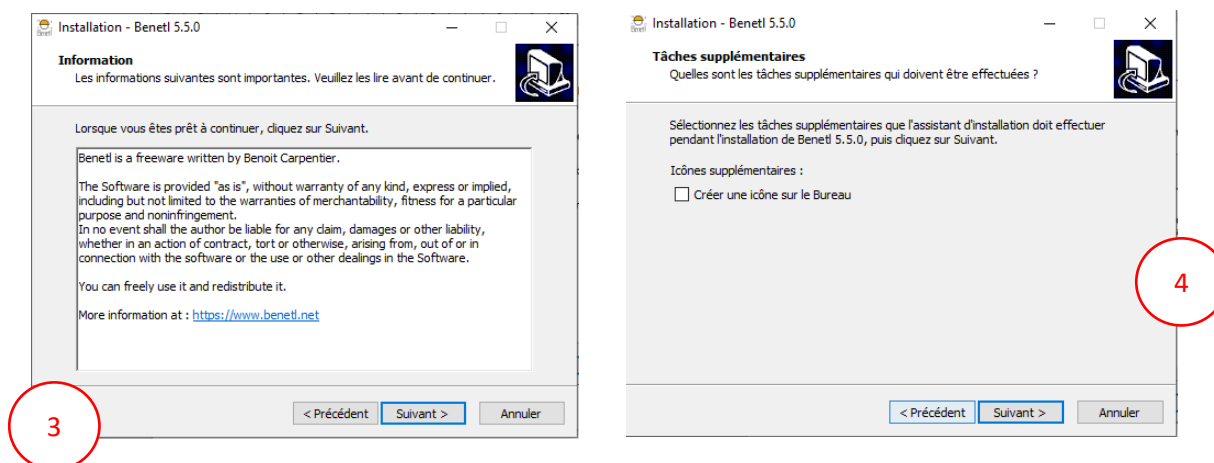


Figure 5. Conditions d'utilisation et tâches supplémentaires

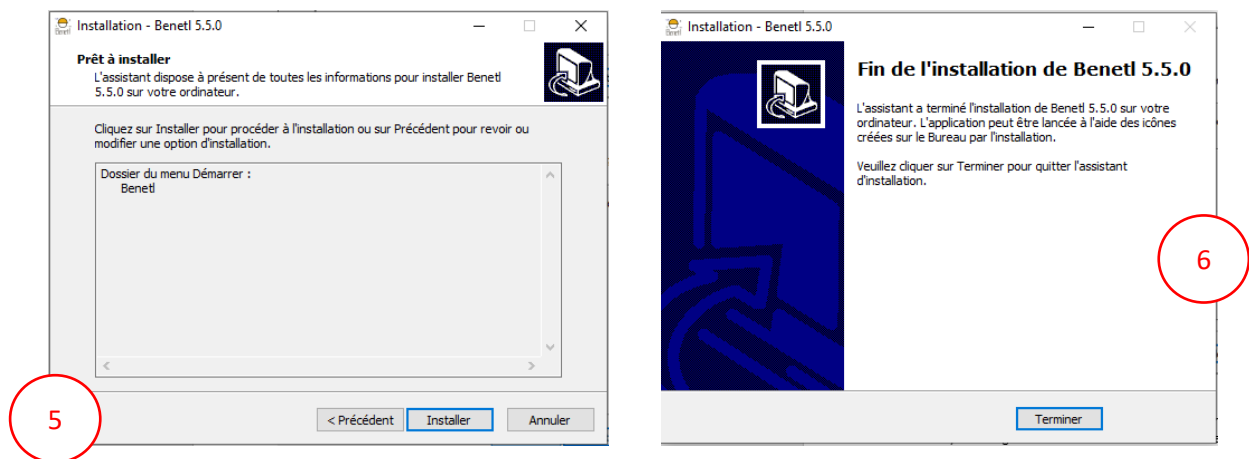


Figure 6. Fin de l'installation de BenETL

Une fois l'installation finie, on peut procéder à la configuration de BenETL. Pour la connexion à la base de données, les configurations sont effectuées dans les fichiers « benetl\_my.xml » pour MySQL et « benetl\_pg.xml » pour PostgreSQL. La figure 7 représente les dossiers de configurations pour la connexion à la base de données.

Ericson (C:) > Benetl >				
Nom	Modifié le	Type	Taille	
bin	21/03/2022 07:25	Dossier de fichiers		
lib	21/03/2022 07:25	Dossier de fichiers		
benetl_my.xml	01/11/2020 21:46	Fichier XML	1 Ko	
benetl_pg.xml	25/07/2020 00:16	Fichier XML	1 Ko	
etltool_gui.bat	20/02/2022 21:38	Fichier de comman...		
license.txt	10/01/2020 23:27	Document texte	1 Ko	
unins000.dat	21/03/2022 07:25	Fichier DAT	5 Ko	
unins000.exe	21/03/2022 07:23	Application	706 Ko	

Figure 7. Les dossiers de configurations pour la connexion à la base de données

Par défaut, BentETL travaille avec PostgreSQL mais si on veut travailler avec MySQL, il faut supprimer le fichier « benetl\_pg.xml » et ajouter une librairie pour la connexion à MySQL dans le dossier de librairie qui est le dossier « lib » d'installation. La figure 8 représente la librairie requise pour la connexion MySQL.

Ericson (C:) > Benetl > lib				
Nom	Modifié le	Type	Taille	
jackson-annotations-2.10.3.jar	25/03/2020 19:55	Executable Jar File	67 Ko	
jackson-core-2.10.3.jar	25/03/2020 20:01	Executable Jar File	341 Ko	
jackson-databind-2.10.3.jar	25/03/2020 20:01	Executable Jar File	1 372 Ko	
javax.activation-api-1.2.0.jar	10/01/2020 23:23	Executable Jar File	56 Ko	
jaxb-api-2.3.1.jar	10/01/2020 23:23	Executable Jar File	126 Ko	
jaxb-impl-2.3.4.jar	31/07/2021 09:52	Executable Jar File	1 109 Ko	
jxl-2.6.12.jar	09/01/2020 23:25	Executable Jar File	709 Ko	
log4j-api-2.17.1.jar	04/02/2022 23:14	Executable Jar File	295 Ko	
log4j-core-2.17.1.jar	04/02/2022 23:14	Executable Jar File	1 749 Ko	
mysql-connector-java-8.0.21.jar	18/03/2022 11:11	Executable Jar File	2 342 Ko	
postgresql-42.2.14.jar	28/07/2020 23:15	Executable Jar File	911 Ko	

Figure 8. La librairie requise pour la connexion MySQL

#### 4.2. Mise en œuvre

Après installation, on peut avoir un raccourci sur le bureau pour le lancement du BenETL, mais il est aussi possible de le lancer via le fichier « etltool\_gui.bat » qui contient la commande :

```
«java -Xms256m -Xmx512m -classpath
"C:\benetl\bin;C:\benetl\bin\etlpack;C:\benetl\lib\mysql-connector-java-
8.0.21.jar;C:\benetl\lib\postgresql-42.2.14.jar;C:\benetl\lib\jxl-
2.6.12.jar;C:\benetl\lib\jaxb-api-2.3.1.jar;C:\benetl\lib\log4j-api-
2.17.1.jar;C:\benetl\lib\jaxb-impl-2.3.4.jar;C:\benetl\lib\javax.activation-api-
1.2.0.jar;C:\benetl\lib\log4j-core-2.17.1.jar;C:\benetl\lib\jackson-core-
2.10.3.jar;C:\benetl\lib\jackson-databind-2.10.3.jar;C:\benetl\lib\jackson-
annotations-2.10.3.jar" etlpack.EtlRun»
```

L'exécution de cette commande ouvre la fenêtre que la figure 9 dans la première section présente, ainsi qu'une interface permettant de choisir le fichier qui sera la source de données, cette interface est illustrée par la figure 9.

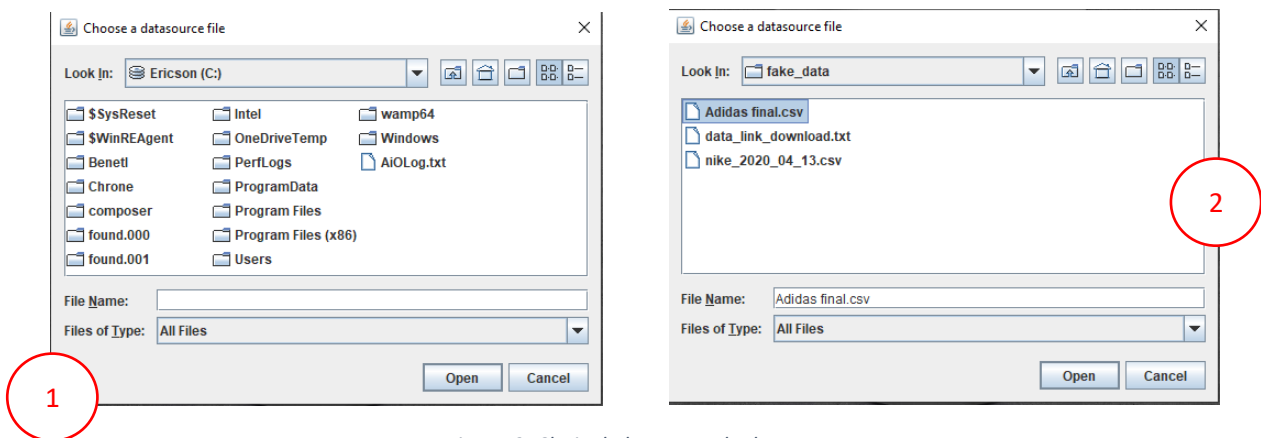


Figure 9. Choix de la source de données

Une fois la source de données choisie le processus d'ETL peut commencer. Avec BenETL 5.5, le processus commence sur l'interface de gestion des entités ainsi que des groupes d'entités vue à la figure 10. Si BenETL est connecté à une base de données qui possède déjà des données, cette option de gestion des entités sera accessible, dans le cas contraire la fonction est désactivée.

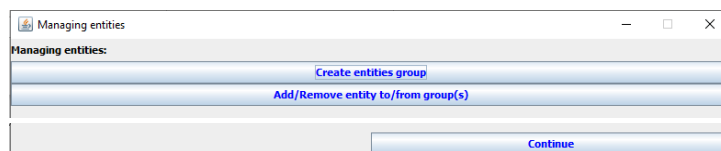


Figure 10. Interface de gestion des entités de BenEtl

Si on continue dans l'utilisation du BenETL, on arrive sur l'interface qui permet de choisir le format de la date qui se trouve dans la source de données. La figure 11 matérialise cette interface. Voici une liste de ces formats de date :

```
"dd/MM/yyyy", "MM/dd/yyyy", "dd/MM/yyyy HH:mm", "MM/dd/yyyy HH:mm",
"dd/MM/yyyy HH:mm:ss", "MM/dd/yyyy HH:mm:ss", "yyyy-MM-dd", "yyyy-MM-dd
HH:mm", "yyyy-MM-dd HH:mm:ss", "yyyy/MM/dd", "yyyy/MM/dd HH:mm", "yyyy/MM/dd
HH:mm:ss", "dd/MM/yy", "MM/dd/yy"
```

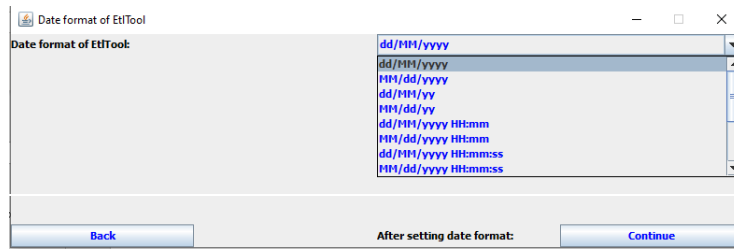


Figure 11. Interface du choix de format de la date

Quand le choix du format de la date est fait, on doit spécifier le numéro de la ligne où va commencer la lecture du fichier, sachant que la première ligne est la ligne 0. La figure 12 illustre cette interface.

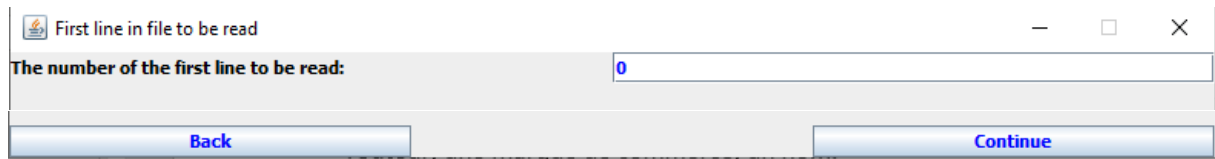


Figure 12. interface de choix de la premier ligne à lire

Ensuite, BenETL récupère les champs qu'il trouve dans le fichier source de données .txt ou .csv ou .xls que vous avez précédemment choisi.

On peut dire si ces champs (première ligne lue) sont des titres (noms de colonnes) ou non.

Si on dit "oui", cela signifie que BenETL conserve la valeur récupérée comme nom de colonne mais dans le cas contraire les champs de la première ligne lue ne sont pas des titres, les champs sont renommés comme "field\_1". La figure 13 représente cette interface.

Field name	Value
Field name 1:	URL
Field name 2:	Product_Name
Field name 3:	Product_ID
Field name 4:	Listing_Price
Field name 5:	Sale_Price
Field name 6:	Discount
Field name 7:	Brand
Field name 8:	Description
Field name 9:	Rating
Field name 10:	Reviews
Field name 11:	Images
Field name 12:	Last_Visited

Is this titles? yes

Add extra field

Back Continue

Figure 13. Interface de configuration des champs

Après avoir vue les champs trouver dans la source de données, on peut définir l'utilisation du champ comme : entité (entity), date(date), oui(yes), non(no).

Une entité est ce qu'on aimerait montrer, mesurer. Une entité, par exemple, peut être l'étiquette d'un routeur, une marque de commerce, un nom.

Toutes les lignes de la source de données contenant la même entité sont "liées" : BenETL utilise une table "entities", listant toutes les entités. Et toutes les lignes de source de données pointent vers une entité dans cette table.

On a besoin d'une zone d'entité et d'une zone de date par ligne de source de données, sinon on ne peut pas continuer.

Les noms des champs doivent tous être différents.

Si on met un champ à « yes », on définit un champ normal, c'est un champ qui contient une valeur.

Si on met un champ à « no », le champ sera exclu de l'extraction.

Pour une date, on peut définir un opérateur : none, gt, geq, lt, lt, leq, leq, like, unlike, part.

Pour une entité, on peut définir un opérateur : none, part, like, like, unlike.

Pour un champ normal (le champ avec utilisation est « yes ») : aucun, plus, moins, diviser, diviser, multiplier, partie, comme, contrairement.

L'opérateur "part" est probablement la capacité la plus puissante d'un EtlTool.

La figure 14 illustre l'interface permettant de faire toutes ces configurations.

Field name	Use	Type	Operator
Field name 1: URL	entity	text	none
Field name 2: Product_Name	yes	text	none
Field name 3: Product_ID	yes	text	none
Field name 4: Listing_Price	yes	text	none
Field name 5: Sale_Price	yes	text	none
Field name 6: Discount	yes	text	none
Field name 7: Brand	yes	text	none
Field name 8: Description	yes	text	none
Field name 9: Rating	yes	text	none
Field name 10: Reviews	yes	text	none
Field name 11: Images	yes	text	none
Field name 12: Last_Visited	date	date	none

Back After defining fields: Continue

Figure 14. Interface de configuration des champs

Dans l'interface suivant, qui est illustré par la figure 15, on n'a qu'à dire à BenETL si une ligne doit être unique ou non.

Unique data, read every x lines, save after every x lines

Are data lines unique: yes

Read line every x lines:

Batch saves after x lines:

Back After information about lines: Continue

Figure 15. Interface pour les lignes uniques

Si on répond "yes", cela signifie que si vous réexécutez un EtlTool à partir du début (par exemple), cela ne créera pas un nouvel enregistrement dans votre table EtlTool.

Cela signifie que on ne peut pas avoir deux fois la même ligne dans la table postgresQL ou MySQL pour votre EtlTool.

A l'inverse, si on répond "no", on peut créer plusieurs fois la même ligne dans le tableau EtlTool.

Dans interface présenté par la figure 16, on peut définir la création d'un script SQL pour les insertions. On peut laisser le nom de la table vide pour utiliser etltool comme nom de table.



On peut définir un nom de table pour le script de création, on peut aussi définir le mode de création du script,

Il y a 4 modes de création :

- normal,
- no date,
- no entity,
- no date and no entity.

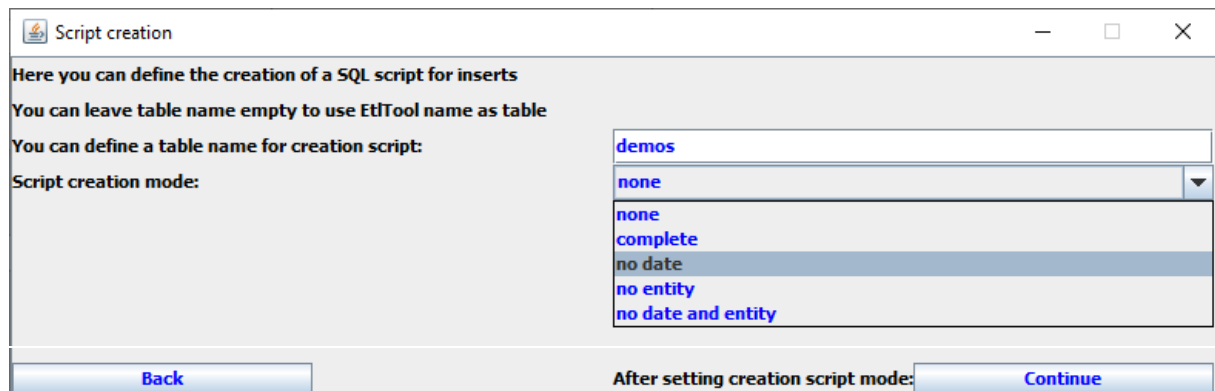


Figure 16. Interface de configuration du script de création

Ensuite, on arrive sur l'interface de configuration du « EtlTool » proprement dit qui est matérialiser par la figure 17. Ici, il faut définir le nom du EtlTool. Ce nom est aussi le nom de la table PostgreSQL ou de la table MySQL où sont stockées les lignes de données si on ne l'a pas défini auparavant.

On doit appuyer sur Entrée pour valider le nom du EtlTool, puis on peut sauvegarder les paramètres.

Un fichier appelé "EtlTool.bat" sera créé dans le dossier où se trouve la source de données.

Depuis Benetl 3.6, on peut définir une base de données spécifique (installée sur un autre système).

On peut entrer ici : l'url de l'installation MySQL / PostgreSQL, l'utilisateur MySQL / PostgreSQL et son mot de passe.

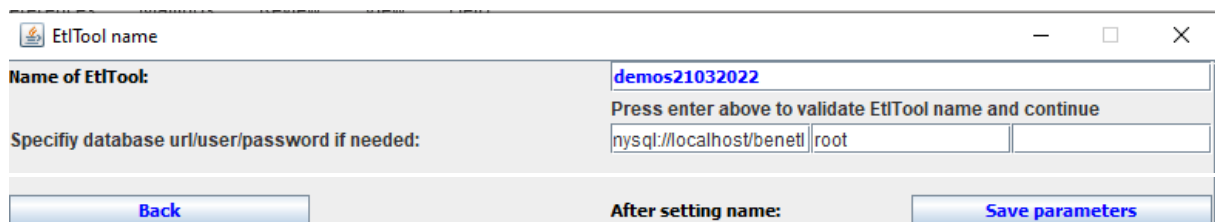


Figure 17. Interface de configuration du EtlTool

Après avoir défini le nom du EtlTool, on peut choisir de définir un ordonnancement ou non pour l'EtlTool. Si l'on choisit "aucun", il n'y aura pas de programmation. On peut affiner la programmation de l'outil EtlTool dans les "tâches planifiées" de Windows.

La tâche planifiée créée permet de contrôler la fréquence de lancement du fichier "EtlTool.bat" dans le dossier datasource.

Il y a 5 types de planification : aucun, par minute, par jour, par semaine, par mois

Ainsi s'achève la configuration du EtlTool sur BenETL, on peut désormais visualiser les données insérées dans Mysql et PostgreSQL.

## CONCLUSION

Pour conclure, la récupération des sources de données se fait via un outil informatique appelé ETL qui se charge d'extraire et de sauvegarder les données dans une base de données. BenETL est un outil qui permet la mise en œuvre de ce processus efficacement.

L'analyse nous a permis de constater que celle-ci est un outil complet, avec ses diverses panoplies d'options et de possibilités, et intéressant pour la mise en œuvre de la collecte des données malgré ses quelques limites.

La réalisation, quant à lui, nous a permis de nous familiariser sur la configuration et l'utilisation de l'outil. Ces derniers s'avèrent être assez simple en suivant les étapes prescrites.

A l'avenir, il nécessite encore quelques minime améliorations comme la traduction de la documentation officielle en français.



## BIBLIOGRAPHIE ET WEBOGRAPHIES

- [1] «Entrepôt de données»; [https://fr.wikipedia.org/wiki/Entrepot\\_de\\_donnees](https://fr.wikipedia.org/wiki/Entrepot_de_donnees). [Accès le 30 05 2018].
- [2] «Etl», <https://fr.wikipedia.org/wiki/Extract-transform-load>. [Accès le 29 05 2018].
- [3] «Les Etl», [http://igm.univ-mlv.fr/~dr/XPOSE2010/Extract\\_Transform\\_Load/etl.php](http://igm.univ-mlv.fr/~dr/XPOSE2010/Extract_Transform_Load/etl.php). [Accès le 01 06 2018].
- [4] «BenEtl», [www.benetl.net](http://www.benetl.net). [Accès le 29 05 2018].

## TABLE DES MATIERES

SOMMAIRE .....	I
NOMENCLATURE .....	II
LISTE DES FIGURES.....	III
INTRODUCTION .....	1
PARTIE I. PRESENTATION .....	2
Chapitre 1. Présentation de l'outils .....	2
1.1. Entrepôt de données ou « Datawarehouse » .....	2
1.1.1. Définition.....	2
1.1.2. Historique .....	2
1.1.3. Principe de fonctionnement.....	2
1.2. ETL ou « Extract Transform Load » .....	3
1.2.1. Définition.....	3
1.2.2. Historique .....	4
1.2.3. Atouts majeurs des ETL .....	4
Chapitre 2. Description de l'application.....	5
2.1. Présentation de BenETL.....	5
2.2. Technologies utilisées.....	5
PARTIE II. ANALYSE .....	6
Chapitre 3. Analyse .....	6
3.1. Evaluation générale .....	6
3.2. Avantages .....	6
3.3. Inconvénients .....	6
PARTIE III. REALISATION.....	7
4.1. Installation et configuration .....	7
4.2. Mise en œuvre.....	8
CONCLUSION .....	15
BIBLIOGRAPHIE ET WEBGRAPHIES .....	IV
TABLE DES MATIERES .....	V
RESUME .....	VI
ABSTRACT .....	VI

## RESUME

Afin de parfaire la prise de décision, l'utilisation de l'informatique décisionnelle est plus que nécessaire pour l'exploitation des données. BenETL est un outil ETL puissant, performant et souple pour faire la collecte des données de type .txt .csv et .xls. Il est gratuit, léger et hautement programmable pour la création de tâches bien que présentant des limites sur le nombre de format de source de données supportées et de système de gestion de base de données.

Mot clés : ETL, BenETL, SGBD, données, format

## ABSTRACT

In order to perfect decision making, the use of BI is more than necessary for data exploitation. BenETL is a powerful, powerful and flexible ETL tool for collecting.txt.csv and.xls data. It is free, lightweight and highly programmable for task creation although it has limitations on the number of supported data source format and database management system.

Keywords: ETL, BenETL, DBMS, data, format