Projet 1SIO
STESIO - Analyse de logs

Cahier des charges **2**
- Contexte 2
- Documents mis à disposition 3
- Documents à produire 3

Evaluation **4**

Les itérations du projet (étapes) **4**
- Étape 1: Auto formation : Lecture et écriture dans un fichier texte en Python 4
- Étape 2: Mise en place de la base de données en SQL 5
- Étape 3 : Analyse des fichiers textes de login 5
- Étape 4 : Réalisation d'une requête de validation du jeu d'essai 5
- Étape 5 : Réalisation du programme Python de génération du script SQL INSERT 5
- Étape 6 : Exécution des scripts SQL d'insertion puis test de requête SQL 6
- Étape 7 : Rédaction d'un mode opératoire 6

Cahier des charges

Contexte

Pour être en conformité avec les obligations légales concernant la mise à disposition d'Internet, la société STESIO a mis en place un proxy pour journaliser les accès au web réalisés par ses salariés. A partir de ce journal, le responsable du système d'information (S.I.) souhaite établir des statistiques comme :

- les sites les plus visités
- la liste des utilisateurs les plus consommateurs

et dans les cas où cela est nécessaire (enquête de police par exemple) être capable de répondre à une requête du type :

- qui a consulté tel site, tel jour, à telle heure ?

Le fichier de log du proxy est un simple fichier texte (log_proxy.txt) contenant des informations sur les accès au web comme l'adresse IP, la date, l'heure, la commande HTTP utilisée (GET ou POST) , l'URL des différents éléments constituant la page téléchargée (images, bandeau, …). Ce journal étant d'une part, un fichier texte et d'autre part étant très volumineux, il est difficile à utiliser directement pour répondre facilement à ces besoins.

Le responsable du SI vous demande de créer une base de données sur ORACLE qui contiendra les tables suivantes :
SALARIES (**num**, nom, prenom, adresseIP) - clef primaire : **num**
PROXY (i**d**, *adresseIP*, jourheure, URL) - clef primaire : i**d**

Précisions :

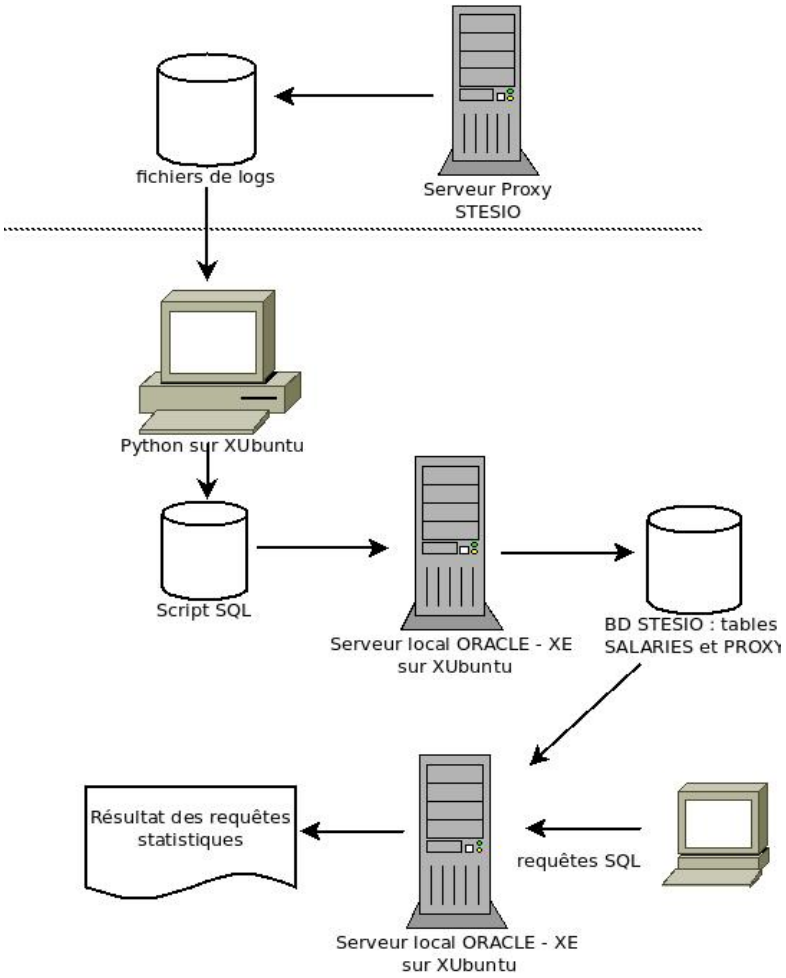
- *le champ adresseIP de la table SALARIES respecte une contrainte d'unicité*
- *le champ adresseIP de la table PROXY référence le champ adresseIP de la table SALARIES*
- *le champ **id** doit être auto-incrémenté (Cf. doc. jointe).*
- *le type DATE d'Oracle permet de mémoriser une date et une heure (Cf. doc. jointe) .*

Vous remplirez la table SALARIES avec les données suivantes :

Num	Nom	Prenom	IP
1	DUPOND	Marie	192.168.2.2
2	DUBOIS	Paul	192.168.2.1
3	DURANT	Quentin	192.168.2.3
4	LEJAUNE	Laurence	192.168.2.100

La table PROXY sera remplie par un script contenant des ordres SQL "INSERT INTO PROXY …" . Ce script sera généré par un programme, écrit en PYTHON, qui lira le fichier texte "log_proxy.txt" et qui, à partir des données lues, créera le script SQL.
Une fois le script généré, il suffira de l'exécuter dans ORACLE pour remplir la table PROXY. Il sera alors plus facile d'établir des statistiques.

Exemple indicatif de processus de traitement



Documents mis à disposition

- le présent cahier des charges
- des jeux d'essai : les fichiers texte log_proxy_2019-01-01.txt, log_proxy_2019-01-02.txt, log_proxy_2019-01-03.txt
- deux documentations :
    - PYTHON : la gestion de fichiers et le traitement des chaînes de caractères
    - ORACLE : la gestion des dates et des identifiants auto-incrémentés
- la grille d'évaluation du projet qui sera utilisée lors de la recette finale.

Document à produire

A chaque étape, vous devez compléter une documentation
Un travail a été ouvert à cet effet sur Moodle.

Environnement de travail

Vous pourrez réaliser et tester les scripts Python dans votre environnement Python favoris (par exemple sur xubuntu ou sur directement sur le poste Fedora).

Par contre, vous utiliserez la machine virtuelle « xubuntu » pour créer la base et tester les scripts SQL.

Les itérations du projet (étapes)

Étape 1: Lecture et écriture dans un fichier texte en Python

Cette étape a pour but de vous former aux commandes Python permettant de lire et d'écrire dans un fichier texte(Cf. doc. Python).
Enregistrez les lignes suivantes dans un fichier "testEntree.txt" :

```
Emile;ZOLA
Victor;HUGO
George;SAND
```

Réalisez ensuite un programme effectuant les actions suivantes :

- ouvrir le fichier testEntree.txt en lecture et un fichier testSortie.txt en écriture.
- pour chaque ligne du fichier testEntree.txt :
    - extraire la première lettre du premier mot et le deuxième mot pour les concaténer
    - convertir la chaîne obtenue en minuscules
    - écrire cette chaîne dans le fichier testSortie.txt

A documenter :

- le code source Python commenté,
- le contenu des fichiers testEntree.txt et testSortie.txt .

Étape 2: Mise en place de la base de données en SQL

Afin de pouvoir faire le projet sur 2 semaines sans que les mots de passe ne soient périmés la 2ème semaine :

- Connectez-vous en tant qu'administrateur système (*system* (mot de passe *system*)).
- Changez le mot de passe *system* en *system2*.
- Connectez-vous en tant qu'utilisateur *btssio* (mot de passe *btssio*).
- Changez le mot de passe *btssio* en *btssio2*.

Dans un l'environnement « réel » de la socité STESIO, nous pourrions créer un utilisateur spécifique pour STESIO, mais, le chapître SQL correspondant n'ayant pas encore été abordé à cette date du projet, vous vous connecterez ici en tant qu'utilisateur btssio (mot de passe btssio).

Vous créerez les tables SALARIES et PROXY, sans oublier les contraintes ni le contenu de la table SALARIES. Donnez un exemple d'ordre SQL pour ajouter un enregistrement dans la table PROXY. Attention au format des dates pour Oracle. L'identifiant devra être auto-incrémenté (Cf. doc. Oracle).

A documenter : un document contenant les ordres SQL permettant de créer les 2 tables, avec des explications.

Étape 3 : Analyse des fichiers textes de login

Vous analyserez les fichiers texte log_proxy_yyyy-mm-dd.txt : notez les informations disponibles sur chaque ligne et à quels champs de la base de données ils peuvent correspondre.

A documenter : un document décrivant votre analyse.

Étape 4 : Réalisation d'une requête de validation du jeu d'essai

Vous écrirez, sans la tester, le requête SQL correspondant à la **demande numéro 1** du cahier des charges à savoir permettant de dresser la liste des sites les plus visités : nombre d'utilisateurs par site, en ordre décroissant

A documenter : le code SQL de la requête permettant d'aboutir à ce résultat

```
select URL, count(*) as nb from proxy
group by URL
order by nb DESC;
```

Étape 5 : Réalisation du programme Python de génération du script SQL INSERT

Vous réaliserez une première version du programme Python demandé. Ce programme lira la liste des enregistrements du proxy dans le fichier texte "log_proxy_yyyy-mm-dd.txt" pour générer un fichier de script SQL nommé "insert_yyyy-mm-dd.sql" contenant les commandes SQL nécessaires pour créer dans la table PROXY les enregistrements correspondant à la date "yyyy-mm-dd".

Dans un premier temps, l'utilisateur saisira uniquement la date du fichier de log au format "yyyy-mm-dd" (exemple : 2017-01-03). Le programme générera les noms des fichiers à partir de cette saisie.

A documenter :

- le code source PYTHON **commenté**
- les scripts SQL générés à partir des 3 fichiers de log fournis

Étape 6 : Exécution des scripts SQL d'insertion puis test des requêtes SQL

Exécutez les scripts SQL générés à l'étape précédente
Suite à cela, la table PROXY doit être remplie (vérifier).
Vous testerez la requête conçue à l'étape 4.

A documenter : le compte-rendu du test, comprenant :

- le résultat de l'exécution des scripts d'insertion
- puis, pour la requête SQL :
    - le code SQL de la requête,
    - le résultat obtenu,
    - un commentaire.

Étape 7 : Rédaction d'un mode opératoire

Vous réaliserez à l'intention de l'administrateur système, un mode opératoire. Ce mode opératoire précisera :

- le format du fichier de log (structure)
- l'utilisation du programme Python

- l'utilisation des scripts SQL générés
- la procédure de test des requêtes de vérification des jeux d'essai

A documenter : le mode opératoire (bien) rédigé et mis en page.