Fiche de procédure

Stage en entreprise

1 : Objectif

Ce projet a pour but de synchroniser massivement d'anciennes données issues de Sage vers une base de données MySQL.

L'objectif principal est de permettre une migration en une seule opération, garantissant ainsi l'intégration de plusieurs années d'historique de données sans intervention manuelle récurrente. Cette synchronisation permet d'assurer une transition fluide vers un nouveau système tout en conservant les informations essentielles stockées dans Sage.

En centralisant ces anciennes données dans MySQL, il devient possible d'améliorer l'accès et la gestion des informations tout en optimisant les performances des applications qui exploitent ces données.

2 : Prérequis

Pour mettre en œuvre cette synchronisation, il est impératif de disposer d'un accès aux bases de données Sage et MySQL avec les privilèges nécessaires pour l'extraction et l'insertion de l'ensemble des données historiques.

Il est crucial de maîtriser la structure des anciennes données présentes dans Sage afin d'assurer une correspondance exacte avec les champs disponibles dans MySQL et d'éviter toute perte d'informations critiques lors de la migration.

Une gestion des entités avec l'EntityManagerInterface permettra de s'assurer que chaque enregistrement est correctement inséré sans duplication ni corruption de données.

Enfin, pour garantir la fiabilité de l'opération, un suivi via des logs et un système de gestion des erreurs doit être mis en place afin d'identifier rapidement tout problème survenu pendant l'importation massive des données historiques.

3 : Récupération des données depuis Sage

L'extraction des données depuis la base Sage se fait via la fonction `insertToSQL()`, qui exécute une requête SQL avancée. Cette requête récupère les informations essentielles des clients tout en filtrant les entrées indésirables.

```
function insertToSQL(): array
{
    $sql = "
        SELECT
            clt.CT_Num,
            clt.CT_Intitule,
            clt.cbCreation,
            clt.CT_Contact,
            clt.CT_Adresse,
            clt.CT_Complement,
            clt.CT_CodePostal,
            clt.CT_Ville,
            clt.CT_Pays,
            clt.CT_Siret,
            clt.CT_Telephone,
            clt.CT_EMail
        FROM F_COMPTET clt
            JOIN F_DOCENTETE doc ON clt.CT_Num = doc.CT_NumPayeur
        WHERE DO_Type = 7
        AND DO_Date >= DATEADD(YEAR, -4, GETDATE())
        AND clt.CT_Intitule NOT LIKE 'CORA%'
        AND clt.CT_Intitule NOT LIKE 'INTERMARCHE%'
        AND clt.CT_Intitule NOT LIKE 'CARREFOUR%'
        AND clt.CT_Num NOT IN ('006040001', '004700155', '006040002', 'MAISONBROUSSAUD')


        GROUP BY clt.CT_Num
```

```
                    clt.CT_Intitule,
                    clt.cbCreation,
                    clt.CT_Contact,
                    clt.CT_Adresse,
                    clt.CT_Complement,
                    clt.CT_CodePostal,
                    clt.CT_Ville,
                    clt.CT_Pays,
                    clt.CT_Siret,
                    clt.CT_Telephone,
                    clt.CT_EMail

        ORDER BY clt.CT_Intitule
    ";
    $result = $this->fetchAll($sql);
    $ctNumList = array();
    foreach($result as $row) $ctNumList[$row->CT_Num] = $row;
    return $ctNumList;
}
```

Cette requête permet d'exclure les clients non pertinents en filtrant les résultats selon plusieurs critères. Elle exclut les grands distributeurs dont les noms commencent par 'CORA', 'INTERMARCHÉ' ou 'CARREFOUR', ainsi que certains numéros clients spécifiques. L'optimisation de cette requête garantit que seules les données utiles sont récupérées, limitant ainsi la charge sur la base de données et améliorant les performances du processus.

4 : Traitement et préparation des données pour MySQL

Une fois les données récupérées depuis Sage, elles doivent être traitées et préparées avant leur insertion dans la base MySQL. Cette étape est réalisée par la méthode `addToSQL()`, qui structure les informations récupérées et les rend exploitables.

```php
1 usage  new *
public function addToSQL(): array
{
    $SageTiers = new SageTiers();
    $tList = $SageTiers->insertToSQL();
    return $tList;
}
```

Cette fonction facilite la gestion des données en les encapsulant dans une structure adaptée, qui sera ensuite transmise au gestionnaire de bases de données MySQL pour insertion.

4.1 : Insertion des données dans la base MySQL

La fonction `insert()` est responsable de l'injection des données Sage traitées dans MySQL. Elle assure la création d'entrées en évitant les doublons et en maintenant la cohérence des informations.

```php
#[Route('/dev/insert', name: 'dev_insert')]
public function insert(Sage $sage, SocietyManager $societyManager, EntityManagerInterface $entityManager): Response
{
    $societySAGE = $sage->addToSQL();
//      dd($societySAGE);

    foreach($societySAGE as $ctNum => $societies) {
        $society = new Society();
        $society
            ->setCode($societies->CT_Num)
            ->setName($societies->CT_Intitule)
            ->setCreatedAt(new \DateTimeImmutable($societies->cbCreation))
            ->setContact($societies->CT_Contact)
            ->setAddress($societies->CT_Adresse)
            ->setAddress1($societies->CT_Complement)
            ->setZip($societies->CT_CodePostal)
            ->setCity($societies->CT_Ville)
            ->setCountry($societies->CT_Pays)
            ->setSiret($societies->CT_Siret)
            ->setPhone($societies->CT_Telephone)
            ->setEmail($societies->CT_EMail)
            ->setSage( isSage: true);

        $entityManager->persist($society);
        $entityManager->flush();
//          dd($society);
    }
    die('OK');
```

Cette étape est cruciale pour garantir que les nouvelles données sont bien stockées dans MySQL tout en préservant leur intégrité.

5 : Affichage des données dans la vue Twig

Après la synchronisation des données entre Sage et MySQL, celles-ci doivent être accessibles et affichées correctement dans l'interface utilisateur.
L'affichage est géré à l'aide de Twig, permettant d'afficher dynamiquement les sociétés récupérées tout en maintenant une mise en page cohérente et responsive.
Le template ci-dessous permet de générer un tableau affichant toutes les informations nécessaires sur les sociétés importées depuis Sage :

```
<table class="table table-hover datatable-table" id="pc-dt-simple">
    <thead>
    <tr>
        <th>ID</th>
        <th>CLIENT</th>
        <th>CONTACT</th>
        <th>DATE</th>
        <th class="text-center">SAGE</th>
        <th class="text-center">ADRESSE</th>
        <th class="text-right">ACTIONS</th>
    </tr>
    </thead>
    <tbody>
```

```
{% for society in society_list %}
        <tr data-index="0">
            <td>{{ society.id }}</td>
            <td>
                {{ society.name }}
                <br>
                <em class="text-muted f-10">{{ society.code }}</em>
            </td>
            <td>{{ society.contact }}<br><span class="text-muted">{{ society.email }}</span></td>
            <td>
                <i class="ti ti-calendar-plus f-18"></i> créé le {{ society.createdAt|date('d/m/Y \\à H\\hi') }}
                <br>
                <i class="ti ti-calendar-time f-18"></i> modifié le {{ society.updatedat|date('d/m/Y \\à H\\hi') }}
            </td>
            <td class="text-center">
```

```
{% else %}
    <li>Aucune société trouvée.</li>
{% endfor %}
```

Explication du Code :

La boucle `{% for society in society_list %}` :

- Elle permet de parcourir la liste des sociétés récupérées depuis la base MySQL.
- Pour chaque société, un `<tr>` est créé avec ses informations détaillées.

Gestion des erreurs et des données absentes :

- La condition `{% else %}` est utilisée pour afficher un message lorsque la liste est vide.
- Cela permet d'éviter un affichage incomplet si la synchronisation n'a pas encore eu lieu.

Mise en page et structure :

- Le tableau est structuré pour une lecture claire avec des en-têtes bien définis.

ID	CLIENT	CONTACT	DATE	SAGE
1	MAISON BROUSSAUD *MAISONBROUSSAUD*	Aymeric Broussaud test@test.fr	créé le 02/09/2024 à 11h57 modifié le 21/02/2025 à 14h39	En anomali
2	100% GAILLARD / SPL BRIVE TOURISME *19100004*	AUDREY RODRIGO audrey.rodrigo@brive-tourisme.com	créé le 03/04/2018 à 00h00 modifié le 21/02/2025 à 14h39	À jour
3	A COEUR DE SOIE *44430001*	KARINE COLLIER karine.coeurdesoie@gmail.com	créé le 30/10/2024 à 11h22 modifié le 21/02/2025 à 14h39	À jour
4	A DEUX PAS D'ICI *47400001*	VALERIE BOTTECHIA adeuxpasdici47@gmail.com	créé le 07/11/2023 à 16h19 modifié le 21/02/2025 à 14h39	À jour
5	A L'AISE BREIZH SARL *29673001*	MARION LANNION marie@alaisebreizh.com; facturation@alaisebreizh.com	créé le 17/09/2010 à 00h00 modifié le 21/02/2025 à 14h39	À jour
6	A TISS B *49500001*	AUDREY BONET audrey.bonet@atissb.com	créé le 25/03/2022 à 14h16 modifié le 21/02/2025 à 14h39	À jour
7	ABY SPORT *94210001*	PATRICIA CANDOTTI pcandotti@abysport.com	créé le 21/02/2024 à 10h00 modifié le 21/02/2025 à 14h39	À jour
8	AD ASTRA CORPORATION LE SUPER ATELI *14000001*	PAUL COFFINIERES paul@lesuperatelier.fr	créé le 16/12/2014 à 00h00 modifié le 21/02/2025 à 14h39	À jour
9	ADEL *92130004*	FABIENNE DELAHAYE fdelahaye@adelexpo.fr	créé le 20/12/2023 à 16h42 modifié le 21/02/2025 à 14h39	À jour
10	ADN / AGENCE DROLEMENT NECESSAIRE *92300003*	VANESSA BROCHE contact@adrolementn.com; vanessa@adrolementn.com	créé le 23/03/2022 à 11h14 modifié le 21/02/2025 à 14h39	À jour

(Il s'agit de la Vue List faite la semaine d'avant, modifiée pour être entièrement dynamique.)

6 : Conclusion

Le système de synchronisation entre Sage et MySQL permet une mise à jour efficace et automatisée des données clients.
Grâce à une approche structurée incluant l'extraction, le traitement et l'insertion des données, cette solution améliore la fiabilité des informations et optimise les processus de gestion.
En cas d'anomalies, un système de détection et d'affichage d'erreurs facilite la correction et le suivi des intégrations.
Cette solution s'inscrit dans une démarche d'optimisation des systèmes d'information et constitue une base évolutive pouvant être améliorée pour intégrer de nouvelles fonctionnalités ou formats de données.