Fiche de TP N°3- Business intelligence:

ETL Avec Pentaho Data Integration

Exercice 1 :

Vous avez deux fichiers à disposition : **orders.csv** contient l'ensemble des commandes envoyées par le client ainsi que le transporteur que vous avez attribué à chaque commande. **realtime.csv** contient les commandes avec des horaires réels (pas toujours complets). On souhaite suivre les commandes en comparant les horaires de chargement et de livraison demandés par le client (**orders.csv**) et les horaires réels obtenus (**realtime.csv**). On veut procéder à la jointure des deux fichiers CSV. Pour cela, nous devons procéder en deux étapes car une jointure nécessite d'avoir trié les données au préalable.

1. Créez une nouvelle transformation : (**Fichier > Nouveau > Transformation)**
   a. Enregistrer et renommer la transformation (**ex1.ktr**)
2. Charger les deux fichiers CVS (**orders.csv, realtime.csv**)
   a. Aller dans l'onglet Palette de création : **Extraction > Extraction depuis fichier CSV**.
   b. Glisser-déposer deux de ces éléments vers la partie droite.
3. Configurer chaque élément
   a. Fixer les séparateurs de champs à **;**
   b. Cliquer sur **Récupérer Champs**
   c. Cliquer sur **Prévisualiser,** pour vérifier que toutes vos données sont importées correctement.
4. Faire un tri sur les deux fichiers (le tri est un préalable indispensable à la jointure).
   a. Insérer deux étapes **Transformation > Tri lignes**
5. Relier chaque étape d'extraction avec une étape de tri
   a. Sélectionner l'étape source et maintenir Shift enfoncée tout en glissant vers l'étape destination. Choisir **Sortie principale de l'étape**
6. Configurer le tri pour qu'il se fasse sur le champ OrderNumber (attribut de jointure)
   a. Double-cliquer sur l'élément ; Cliquer sur **Récupérer champs**; mettre **Ascendant à N** pour tous les champs sauf **OrderNumber**
7. Faire la jointure : Insérer une étape **Jointure lignes > Jointure comparaison de lignes**
8. Relier les deux étapes de tri à cette étape de jointure
   a. Dans **Première étape** sélectionner le tri correspondant au fichier Orders et sélectionner le second tri pour **Seconde étape**
9. Configurer une jointure externe gauche sur le champ **OrderNumber**
   a. Fixer **Type Jointure** à **Left Outer**
   b. Récupérer les champs clés pour les deux étapes ; Ne conserver que le champ **OrderNumber** dans les deux cas
10. En cliquant droit sur votre étape de jointure, vous pouvez choisir de prévisualiser votre flux de sortie.
11. Remplacez les valeurs nulles qui concernent les dates et les horaires réels avec une information de type String.
    a. Insérer une étape **Divers > Remplacer valeur nulle,** et relier avec la jointure
    b. Dans les étapes d'extraction, mettre les champs de type Date au type String
12. Configurer l'élément
    a. Double-cliquer sur l'élément et cocher **Sélection par champs**
    b. Récupérer les champs et ne conserver que les champs nommés Realxxxxxxxx dans la liste

c. Dans la colonne Remplacer par mettre la chaîne de caractères que vous voulez faire apparaître (inconnue)

13. Supprimez la colonne OrderNumber_1 de votre flux de données.
    a. Insérer une étape **Transformation -> Altération structure de flux**
    b. Configurer l'élément pour obtenir le résultat demandé
    c. Aller dans l'onglet **Retirer, Récupérer champs**, Retirer tous les champs de la liste sauf orderNumber_1
14. Exportez votre table dans un fichier Excel.
    a. Insérer une étape **Alimentation -> Alimentation fichier MS Excel**
    b. Relier cette étape avec la précédente
    c. Indiquer l'emplacement et le nom du fichier à sauvegarder
    d. Dans l'onglet Champs, cliquer sur Récupérer champ
15. Exécutez votre transformation.

Dans le menu principal **Actions -> Exécuter** puis cliquer sur **Démarrer**

Exercice 2 : Entity Resolution & Advanced Transformation

1. Importer les deux fichiers **dataset1.csv et dataset2.csv.**
2. Supprimer les doublons entre les deux datasets en se basant sur:
    a. L'opérateur **Calculateur** pour construire la clé de recherche [nom, prenom]
    b. L'opérateur **recherche approximative** (utilisé l'algorithme *similitude lettres pairs*)
    c. L'opérateur ***Filtrage des lignes*** en supprimant ceux avec le degré se similarité >= 0.5

3. Créer une nouvelle colonne (**email**) sous la forme "p.nom@esi-sba.dz", où ***p*** est la première lettre du prénom, à l'aide des opérateur suivants :
    a. **Extraction depuis chaînes de caractères** et **Calculateur**

4. Homogénéiser les valeurs de la colonne **sexe** en utilisant l'opérateur **Tableau de Correspondance.**

5. Ajouter une colonne **TypeSalaire** en se basant sur l'opérateur **Plage de Nombres**, tel que :
    a. Si 0<=salaire<50000, alors TypeSalaire=small
    b. Si 50000<=salaire<100000, alors TypeSalaire=medium
    c. Si 100000<=salaire, alors TypeSalaire=high

6. Supprimez les colonnes supplémentaires de votre flux de données.
    a. Insérer une étape **Altération structure de flux**

7. Sauvegarder la sortie dans la table **personne(nom, prenom, sexe, ville, code-postal, salaire, email, typesalaire)**
    a. Ajouter **Ajout séquence**
    b. Ajouter le fichier **ojdbc11.jar** dans le répertoire ***pdi-ce-9.1.0.0-324\data-integration\lib*** de **Pentaho** et cela pour pouvoir créer une connexion vers la base de données Oracle.
    c. Ajouter **insertion dans table** et indiquer les bonnes correspondances entre la table **Personne** et le flux de données (il faut indiquer les paramètres de votre connexion).

Exercice 3:

Créer une tâche qui a pour but d'exécuter la transformation de l'exercice précédent et de renvoyer un message en cas d'erreur.

1. Créer une nouvelle tâche : **Fichier > Nouveau > Tâche**
2. Glisser-déposer dans la partie droite un élément **Start**,
3. Vérifier la connexion bases de données
   a. Si oui, vider la table **Personne** (**Script SQL**),
   b. Sinon annuler la tâche (**Mise en échec tâche**) et afficher un message d'erreur
4. Rajouter un délai d'attente de 15 secondes
5. Ajouter un élément **Exécution Transformation**
6. Insérer deux éléments **Succès Tâche** et **Mise en échec tâche**.

Exercice 4 : Soit le schéma relationnel d'une base de données transactionnelle :

Magasin(id_magasin, nom_magasin, tel , #id_adr); **Adresse** (id_adr, rue, numero , ville, region);

Client id_client, nom_client, prenom_client, sexe, dateNaissance);

Produit (id_produit , nom_produit, prix_achat, prix_vente, #id_categorie); **Categorie**(id_categorie, categorie);

Ticket (id_ticket, id_magasin, idc, dateVente, montant); **Detail_ticket**(#id_ticket, #id_produit, qte);

Stock(#id_magasin,#id_produit, dateStock, qte_disponible);

❖ Partie 1 :

- On souhaite analyser la quantité vendue et le montant de vente par les dimensions suivantes:
  - **DIM_PRODUIT** (id_produit, nom_produit, categorie, prix_vente);
  - **DIM_MAGASIN**(id_magasin, nom_magasin, tel, ville, region);
  - **DIM_TEMPS** (id_temps, mois ,trimestre, annee);

1. Créer la BDD transactionnelle en exécutant le script ***creation_script_tp4_2025.sql***.

2. Modéliser le schéma en étoile, et créer les tables du datawarehouse.
   - Créer la table de fait et les tables dimension.

3. Modéliser la zone de préparation des données ***Staging Area*** et créer ces tables.

4. Réaliser le job ***ETL*** qui permet d'**alimenter** la BDD ***Staging Area.***
   - Utiliser l'«***Extraction Complète***» comme technique d'extraction

5. Réaliser le job ***ETL*** qui permet d'**alimenter** le ***DataWarehouse***
   - Utiliser l'«***Extraction Complète***» comme technique d'extraction

❖ Partie 2 :

- Refaire la **partie 1** en analysant
  - Le nombre de ventes & le bénéfice (**montant de vente-montant d'achat**) par : [Catégorie], [Ville , région] ,[semestre, année], [Sexe], et [groupeAge]
  - La qte de stocke moyenne : par catégorie, par région et par mois.