

Oracle® Data Integrator

Prise en Main d'un Projet ETL

10g Release 3 (10.1.3)

décembre 2006

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Table des matières

Introduction	1
Vue globale du produit	3
Qu'est-ce que Oracle Data Integrator ?	3
Problématique	3
Une solution unique	3
Architecture technique	4
Installation de Oracle Data Integrator	7
Vérification des pré-requis	7
Installation	7
Installation de Oracle Data Integrator à partir du CD	7
Installation de Oracle Data Integrator à partir d'un fichier téléchargé	7
Démarrer un projet	8
Le Projet ETL	9
Les exemples de ce document	9
L'environnement des exemples	9
Les modèles de données	9
La problématique	11
Premier démarrage	13
Premier démarrage	13
Démarrer l'environnement de démonstration	13
Démarrer Designer	13
L'Interface de Designer	14
Le contrôle de données	17
Introduction à l'intégration de données	17
Introduction au contrôle des données	17
Contrôle "SRC_CUSTOMER"	18
Objectif	18
Traduction du problème	18
Création des contraintes	18
Interpréter les résultats	24
Les interfaces	27
Interfaces pour l'intégration de données	27
Interface "Pop. TRG_CUSTOMER"	27
Interfaces pour l'intégration de données	40
Interface "Pop. TRG_SALES"	40
La mise en Production	49
Opération	49
Introduction à la mise en Production	49
Package "Load Sales Administration"	50

Objectif	50
Les développements livrés avec Oracle Data Integrator	50
Traduction du problème	50
Les étapes à suivre	51
Déploiement	55
Objectif.....	55
Les étapes à suivre.....	55
Création du Scénario	55
Exécution du Scénario à partir de Oracle Data Integrator	56
Suivre l'exécution du scénario	56
Exécuter le scénario à partir d'une Commande OS.....	57
Conclusion.....	59
Que peut-on faire d'autre avec Oracle Data Integrator ?	59
Aller plus loin avec Oracle Data Integrator	60
Contacter Oracle.....	60

Introduction

L'objet de ce document est de vous donner une vue d'ensemble des possibilités qui vous sont offertes avec Oracle Data Integrator.

Au travers de quelques exemples décrits dans ce document, vous serez à même de comprendre comment réaliser un projet ETL avec **Oracle Data Integrator**, visant au chargement d'un Data Warehouse depuis des systèmes hétérogènes, tout en contrôlant la qualité des données.

Ce projet est réalisable en 90 minutes environ. Vous pouvez sauvegarder votre travail et faire une pause à chaque instant.

Vue globale du produit

Qu'est-ce que Oracle Data Integrator ?

Produit d'intégration de données largement utilisé, Oracle Data Integrator propose une approche novatrice reposant sur les règles déclaratives pour la définition des processus d'intégration et de transformation de données, permettant une simplification et une accélération des développements et de la maintenance. Grâce à son architecture "E-LT", Oracle Data Integrator garantit non seulement les meilleures performances possibles pour l'exécution des processus de transformation et de validation des données, mais est aussi aujourd'hui la solution offrant le meilleur rapport performance /prix du marché.

Oracle Data Integrator fournit une infrastructure unifiée pour rationaliser tous les projets d'intégration de données et d'applications.

Problématique

Dans un souci permanent de compétitivité, les entreprises sont amenées à multiplier les solutions logicielles métier tout en les faisant coexister sur des plates-formes matérielles et systèmes hétérogènes. Les services informatiques se trouvent donc confrontés à la nécessité de fluidifier et de sécuriser le dialogue entre ces diverses solutions. Les projets qui implémentent ces besoins d'intégration doivent être réalisés dans les délais et en respectant les budgets.

Une solution unique

Oracle Data Integrator est une solution d'ETL qui repose d'une part sur une approche par les règles déclaratives et d'autre part sur une architecture "E-LT" unique tirant parti de la puissance de vos moteurs de SGBDR. Cette combinaison permet une forte productivité en développement comme en maintenance, et les meilleures performances pour l'exécution des processus de transformation et de validation de données, quels que soient les volumes de données à traiter.

Voici les raisons qui ont poussé plus de 500 sociétés à choisir Oracle Data Integrator pour leurs projets de data warehousing :

- **Développement et maintenance plus rapides et plus simples** : L'approche par les règles déclaratives réduit considérablement le temps d'apprentissage du produit et accroît la productivité du développeur, tout en facilitant la maintenance. Cette approche sépare la définition des processus de leur implémentation, et sépare les règles déclaratives (le "quoi") des flux de données (le "comment"). Cette approche par les règles déclaratives réduit drastiquement la charge de développement et de maintenance (jusqu'à cinq fois par rapport à des outils ETL traditionnels).
- **Contrôle de l'intégrité des données** : Oracle Data Integrator assure la détection automatique des données erronées et leur recyclage avant insertion dans l'application cible. Ces opérations sont effectuées sans programmation, en accord avec les règles et les contraintes définies à la fois dans l'application cible et dans Oracle Data Integrator.
- **Meilleure performance en exécution** : les solutions d'ETL traditionnelles sont basées sur des moteurs propriétaires qui effectuent les transformations de données enregistrement par enregistrement, ce qui limite les performances. L'architecture E-LT, basée sur vos moteurs de SGBDR, est capable d'exécuter les transformations de données sur le serveur cible en mode ensembliste, avec une performance bien supérieure.

- **Architecture plus simple et plus efficace** : l'architecture E-LT supprime le besoin d'un serveur ETL centralisé entre les sources et le serveur cible. Elle utilise le serveur cible et son SGBDR pour exécuter des transformations complexes (pour la plupart exécutées en batch quand le serveur n'est pas en train d'exécuter des requêtes utilisateur).
- **Indépendance de la plate-forme** : Oracle Data Integrator supporte toutes les plates-formes matérielles (du PC au Mainframe), avec le même logiciel.
- **Connectivité étendue** : Oracle Data Integrator supporte tous les SGBDR y compris les plates-formes principales de Data Warehousing comme Teradata, IBM DB2, Netezza, Oracle, Sybase IQ et de nombreuses autres technologies telles que les fichiers plats, les ERPs et CRMs, LDAP, XML, etc.
- **Économique** : L'élimination du serveur et du moteur ETL réduit les coûts initiaux d'acquisition du matériel et du logiciel, ainsi que les coûts de maintenance. La courbe d'apprentissage accélérée et la productivité accrue des développeurs réduisent quant à eux les coûts d'implémentation et de maintenance du projet.

Architecture technique

Oracle Data Integrator est construit autour de son référentiel modulaire accédé en mode client/serveur ou par des clients légers par des composants entièrement écrits en Java : les modules d'interface graphique et les agents d'exécution.

Le **Référentiel** est un référentiel de méta-données complet et totalement ouvert. Il contient les méta-données des applications accédées, les versions des projets développés ainsi que toutes les informations nécessaires à leur exploitation (planification, ordonnancement et compte-rendu d'exécution).

Topology Manager est l'interface graphique qui gère les données de description de l'architecture physique et logique du système d'information. La description des sites, des machines et des serveurs de données permettra à Oracle Data Integrator d'exécuter les mêmes interfaces dans des environnements différents.

Designer est l'outil du responsable des règles d'intégrité des données et du concepteur d'interfaces. Il offre notamment des fonctions de :

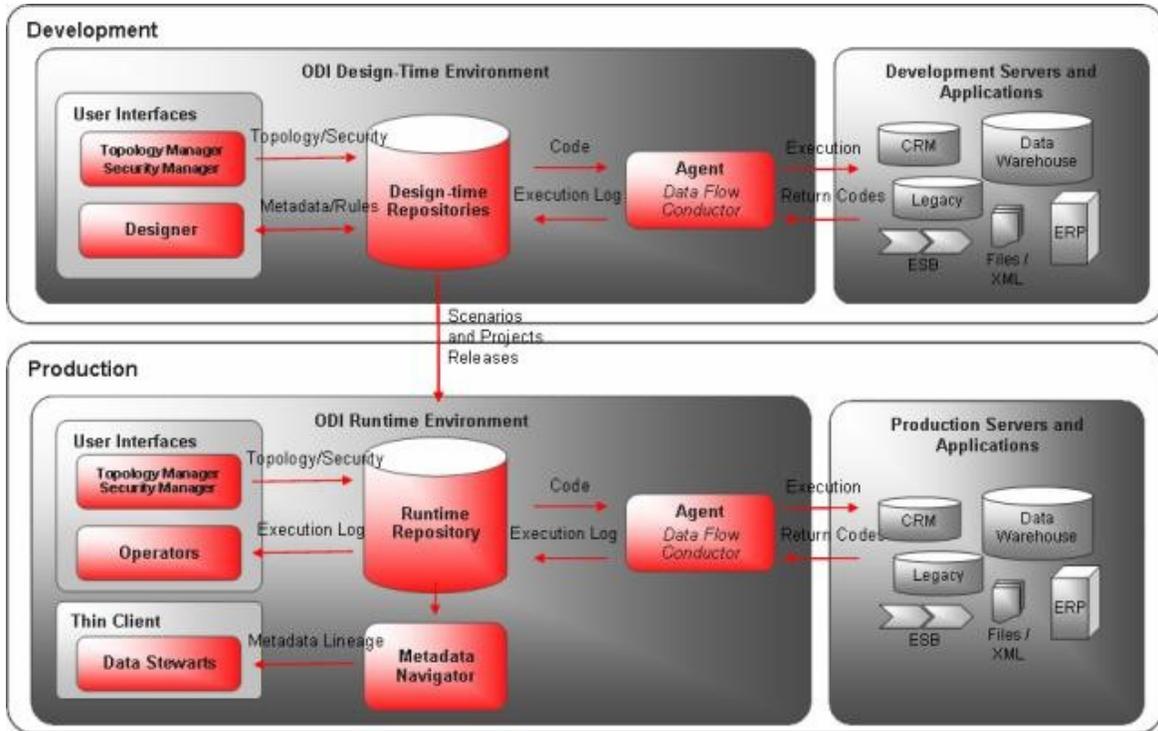
- Reverse-engineering automatisé des applications ou bases de données existantes
- Création et maintenance graphique des interfaces
- Visualisation des flux de données dans les interfaces
- Génération automatique de documentations
- Personnalisation et adaptation des méthodes de génération des traitements.

Security Manager est l'outil de gestion des utilisateurs et de la sécurité dans Oracle Data Integrator . Il permet de gérer les droits d'accès aux fonctions et aux composants de Oracle Data Integrator, par utilisateur et par profil.

Operator est le module de gestion et monitoring des interfaces en production. Il est dédié aux exploitants.

L'Agent permet l'exécution des interfaces inter-applicatives développées avec Designer. Il intègre un **Scheduler** pour exécuter les interfaces selon un planning pré-défini.

Metadata Navigator est une interface légère utilisable dans un navigateur web pour accéder le Référentiel afin de naviguer, chercher ou analyser son contenu, et tracer la généalogie des données, les références croisées, l'analyse d'impact, etc. Vous pouvez aussi utiliser cet outil pour gérer tout le processus de production.



Installation de Oracle Data Integrator

Pour tirer pleinement partie de ce document, vous devez avoir au préalable installé Oracle Data Integrator. La procédure d'installation est décrite ci-dessous :

Vérification des pré-requis

Votre poste de travail doit répondre aux pré-requis ci-dessous :

- Tout OS disposant de capacités Windows ou X/Windows (Windows, Linux, HP-UX, Solaris etc.)
- 256 Mo de RAM/ 512 recommandés
- 150 Mo Disque dur
- TCP/IP

Installation

Installation de Oracle Data Integrator à partir du CD

1. Insérez le CD-ROM de Oracle Data Integrator dans votre lecteur de CD-ROM.
2. Allez dans le répertoire `/setup/<plateforme>/` du CD, où `<plateforme>` est le nom de votre système.
3. Exécutez `setup.exe` sur une plate-forme Windows ou `sh ./setup.bin` sur une plate-forme Unix.
4. Suivez les instructions du programme d'installation.

Installation de Oracle Data Integrator à partir d'un fichier téléchargé

1. Exécutez `setup.exe` sur une plate-forme Windows ou `sh ./setup.bin` sur une plate-forme Unix.
2. Suivez les instructions du programme d'installation.

Note : Une Machine Virtuelle Java est comprise dans ce programme d'installation. Il est recommandé d'utiliser cette JVM.

Note: Pour utiliser les web services, Oracle Data Integrator requiert le Java Development Kit, qui n'est pas inclus avec le programme d'installation. Le message suivant peut apparaître lorsque vous démarrez des modules Oracle Data Integrator et doit être ignoré: "*A JDK is required to execute Web Services with Oracle Data Integrator. You are currently using a JRE.*"

Note : Ce programme d'installation vous offre la possibilité de créer des raccourcis vers les modules de Oracle Data Integrator que vous avez installés.

Attention : Si vous utilisez Microsoft **Windows 95 ou 98**, alors il vous faut effectuer les opérations indiquées dans Utiliser Oracle Data Integrator avec Windows 95/98 afin de modifier les raccourcis éventuellement créés.

Note: Pendant l'installation, vous aurez le choix de type d'installation. Vous pouvez choisir d'installer l'environnement *Évaluation* si vous ne voulez que les fichiers de base ou vous pouvez choisir une installation en environnement *Complet* si vous voulez installer les utilitaires de Oracle Data Integrator, notamment ceux permettant la création des référentiels. Les deux marcheront pour les exercices dans ce document. Ne choisissez pas une des autres options proposées

Démarrer un projet

Oracle Data Integrator est maintenant installé avec l'environnement d'évaluation.

Le Projet ETL

Ce projet **ETL** est un exemple permettant de comprendre comment échanger et contrôler les données des applications de votre système d'information.

Les exemples de ce document

L'environnement des exemples

L'environnement de prise en main choisi par Oracle Data Integrator reflète une problématique d'alimentation du suivi des ventes à partir de sources de données hétérogènes issues des systèmes de production.

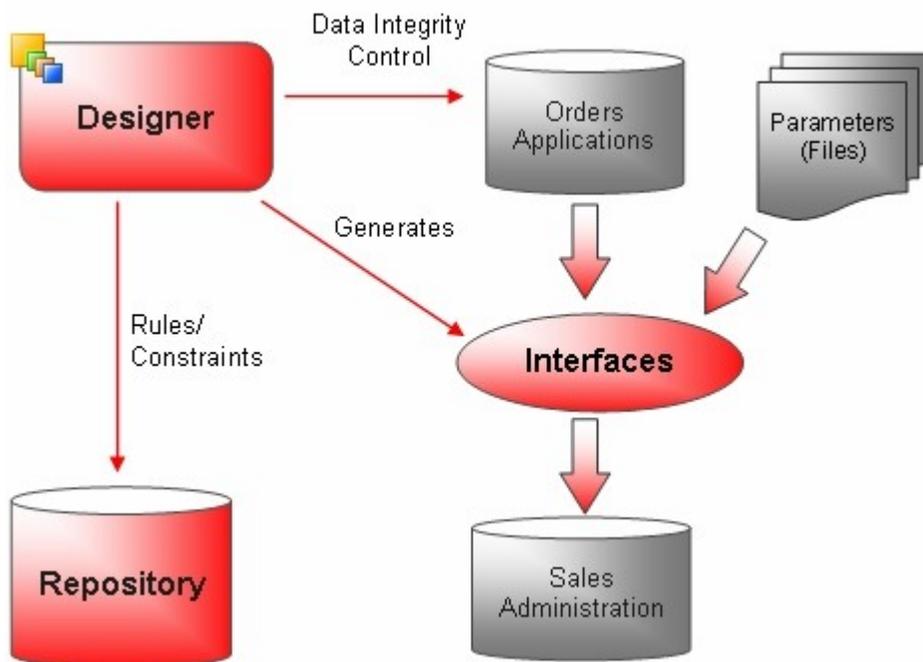
Référentiel : Le Référentiel contient l'ensemble des métadonnées nécessaires au fonctionnement des exemples de prise en main. Il est hébergé dans une base de données livrée en standard pour l'évaluation.

Orders Application : Application de suivi des commandes clients hébergée dans une base de données fournie (base exemple "srcdemo").

Parameters (File) : Fichiers plats (ASCII) issus du système de production contenant la liste des représentants et la segmentation des âges en tranches d'âges.

Sales Administration : Administration (Suivi) des ventes hébergée dans une autre base de données fournie (base exemple "trgdemo").

La vision globale de l'environnement est décrite ci-dessous :



Les modèles de données

Les figures ci-dessous illustrent les modèles de données des applications :

Orders Application

SRC_REGION			
<u>REGION_ID</u>	NUMERIC(10)	<pk>	not null
REGION	VARCHAR(50)		null
COUNTRY_ID	NUMERIC(10)		null
COUNTRY	VARCHAR(50)		null

SRC_CITY			
<u>CITY_ID</u>	NUMERIC(10)	<pk>	not null
CITY	VARCHAR(50)		null
REGION_ID	NUMERIC(10)		null
POPULATION	NUMERIC(10)		null

SRC_ORDER_LINES			
<u>ORDER_ID</u>	NUMERIC(10)	<pk>	not null
<u>LORDER_ID</u>	NUMERIC(10)	<pk>	not null
PRODUCT_ID	NUMERIC(10)		null
QTY	NUMERIC(10)		null
AMOUNT	NUMERIC(10,2)		null

SRC_CUSTOMER			
<u>CUSTID</u>	NUMERIC(10)	<pk>	not null
DEAR	NUMERIC(1)		null
LAST_NAME	VARCHAR(50)		null
FIRST_NAME	VARCHAR(50)		null
ADDRESS	VARCHAR(100)		null
CITY_ID	NUMERIC(10)		null
PHONE	VARCHAR(50)		null
AGE	NUMERIC(3)		null
SALES_PERS_ID	NUMERIC(10)		null

SRC_ORDERS			
<u>ORDER_ID</u>	NUMERIC(10)	<pk>	not null
STATUS	VARCHAR(3)		null
CUST_ID	NUMERIC(10)		null
ORDER_DATE	DATE		null
CUSTOMER	VARCHAR(35)		null

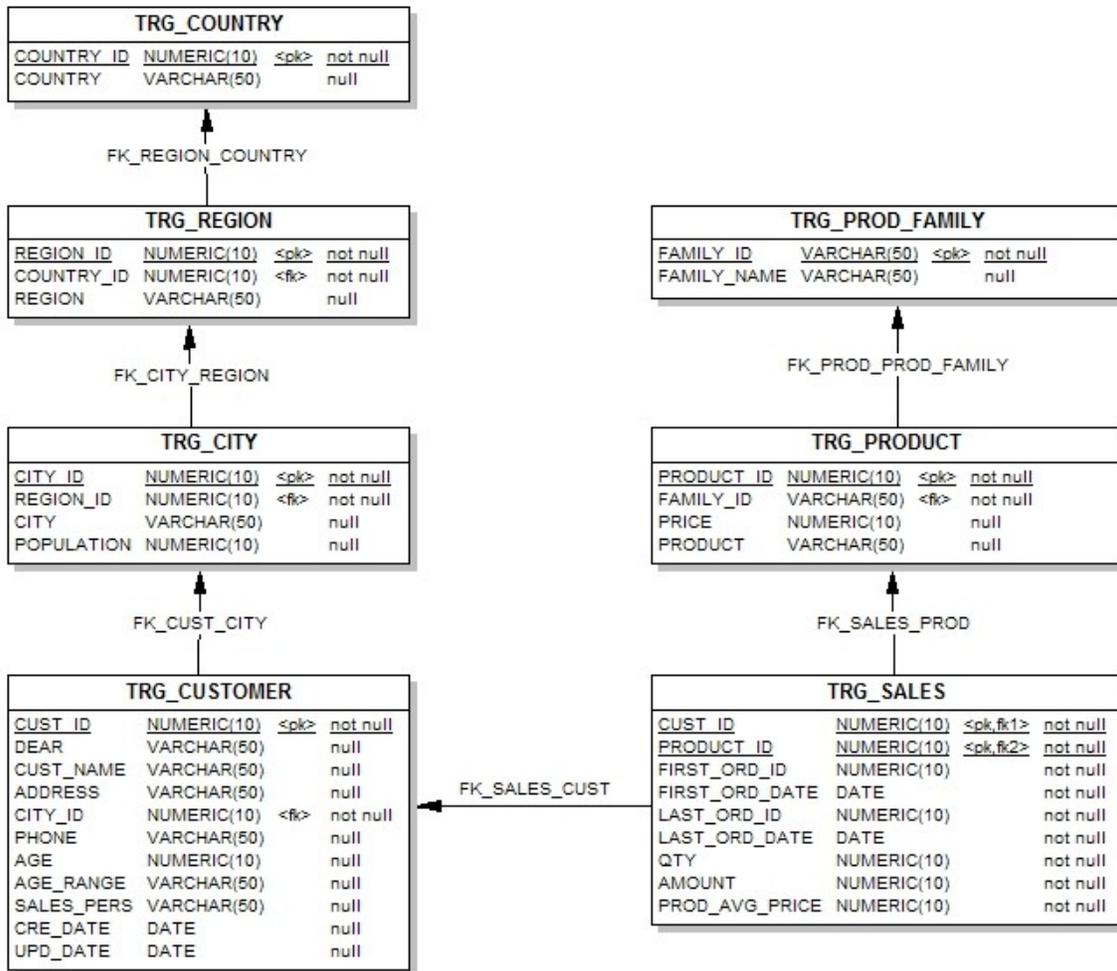
SRC_PRODUCT			
<u>PRODUCT_ID</u>	NUMERIC(10)	<pk>	not null
PRODUCT	VARCHAR(50)		null
PRICE	NUMERIC(10,2)		null
FAMILY_NAME	VARCHAR(50)		null

Parameters (File)

SRC_SALES_PERSON			
<u>SALES_PERSON_ID</u>	NUMERIC(10)	<pk>	not null
FIRST_NAME	VARCHAR(50)		null
LAST_NAME	VARCHAR(50)		null
HIRE_DATE	DATE		null

SRC_AGE_GROUP			
<u>AGE_MIN</u>	NUMERIC(3)	<pk>	not null
<u>AGE_MAX</u>	NUMERIC(3)	<pk>	not null
AGE_RANGE	VARCHAR(50)		null

Sales Administration



La problématique

La problématique commune à tout projet d'intégration et de transformation est double :

- Comprendre et connaître la qualité des données des applications.
- Échanger simplement les données entre les applications en respectant les règles métier du système d'information.

Nous adresserons ces problèmes dans des exemples qui couvriront les sujets suivants:

- Contrôler les données dans la base
- Créer et exécuter une transformation ETL
- Nettoyer les résultats de la transformation
- Automatiser le processus

Note: Les processus que nous allons examiner dans ce guide se focalisent sur l'ETL. Bien que ce point soit non traité dans ce document, la création d'un processus EAI en temps réel peut

être réalisée de la même manière. Pour plus d'informations, référez vous à la documentation en ligne de Oracle Data Integrator après avoir terminé la prise en main.

Le contrôle de données

Au travers d'exemples, vous allez apprendre comment Oracle Data Integrator vous permet d'auditer la qualité des données de vos applications. En effet, l'application "Order Application" contient un certain nombre d'incohérences que vous allez détecter :

L'alimentation de "Sales Administration"

Deux exemples simples vont vous montrer comment gagner en productivité dans l'alimentation des données de l'application "Sales Administration" à partir des applications "Orders Application" et "Parameters (File)":

La mise en production des développements réalisés avec Oracle Data Integrator

Ce chapitre a pour objectif de vous montrer comment ordonnancer l'exécution des développements et automatiser leur exécution. L'objectif étant d'arriver en un clic de souris à exécuter tous les traitements nécessaires à la bonne alimentation de l'application "Sales Administration" :

Premier démarrage

Premier démarrage

Démarrer l'environnement de démonstration

Oracle Data Integrator inclut les bases de données contenant les données des exemples ainsi que le référentiel initial. Avant de commencer à parcourir les différents exemples, **vous devez impérativement suivre les étapes suivantes** :

Sélectionnez **Démarrer > Programmes > Oracle Data Integrator > Exemples**, et sélectionnez **Démarrer Environnement de Démo** pour démarrer le serveur de base de données du Référentiel, les bases sources et cible.

Pour arrêter ces bases de données, vous devez utiliser le raccourci **Arrêter Environnement de Démo** qui se trouve dans le même dossier. **Vous ne devez pas arrêter les bases de données en utilisant le raccourci clavier [CTRL-C] ou en fermant leurs fenêtres d'exécution.**

Sous les systèmes Unix, vous pouvez lancer les scripts .sh appropriés dans une invite de commande en ajoutant "&" pour démarrer les serveurs en tâche de fond (par exemple : `startdemo.sh&` pour démarrer le serveur de base de données du Référentiel et `stopdemo.sh` pour l'arrêter).

Démarrer Designer

1. Sélectionnez **Démarrer > Programmes > Oracle Data Integrator > Designer**. La fenêtre suivante apparaît à l'écran, vous invitant à sélectionner le référentiel auquel vous désirez vous connecter :

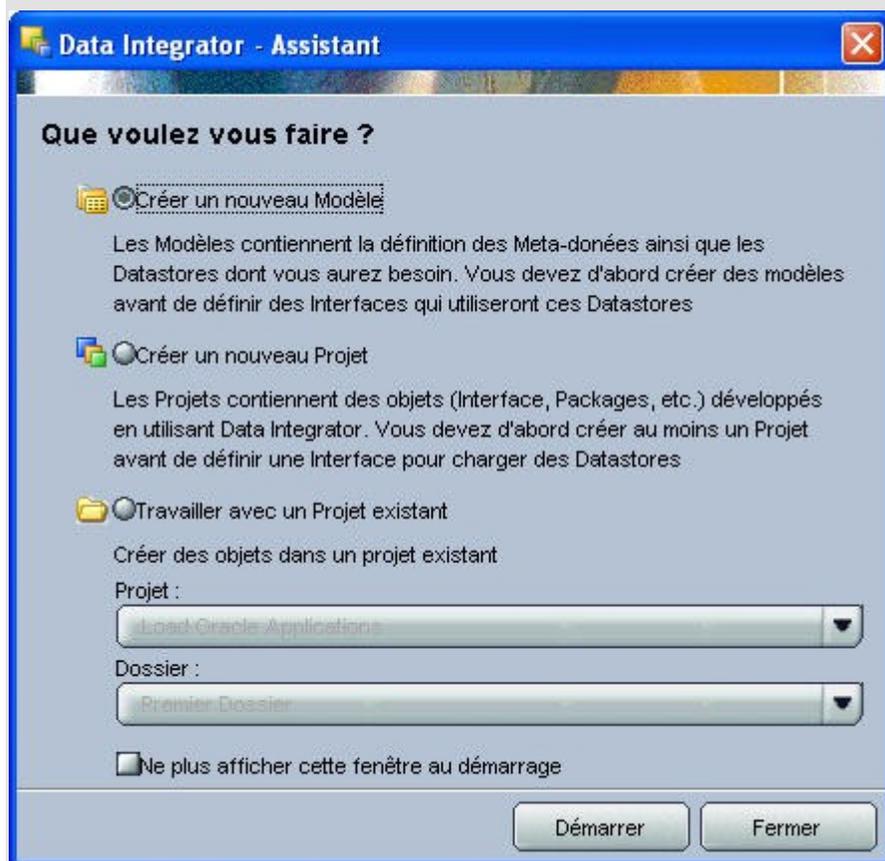


2. Sélectionnez le login approprié dans la liste **Nom de Login**. Choisissez *Getting Started – ETL Project*.
3. Cliquez sur **OK** pour vous connecter au Référentiel.

Note : Assurez-vous de bien avoir **démarré le serveur du Référentiel** comme indiqué dans le paragraphe précédent, ainsi que les serveurs de base de données des applications "Orders Application" et "Sales Administration".

4. **Designer** démarre.

Note : Quand Designer démarre, un assistant peut apparaître. Vous pouvez fermer cette fenêtre en cliquant sur le bouton Fermer.

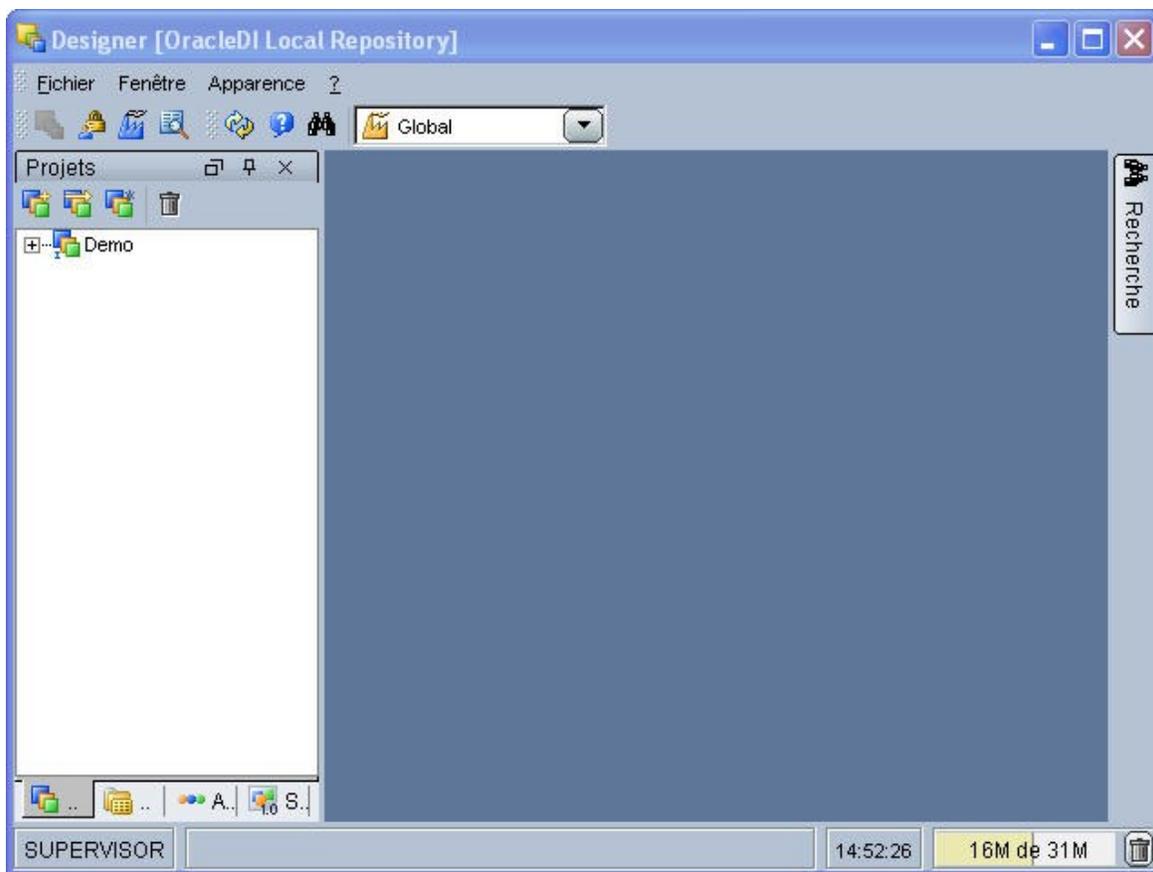


L'Interface de Designer

Designer vous permet de gérer :

- **Les Modèles de données** de vos applications : toutes les méta-données de vos serveurs de données (tables, colonnes, contraintes, descriptions, références croisées etc.)
- **Les Projets d'alimentation** : toutes les règles d'alimentations entre vos serveurs de données (interfaces, traitements, variables etc.)

L'interface graphique de Designer se présente comme suit :



Le Menu

Le **Menu** permet d'accéder aux fonctions générales :

- Import/Export;
- Assistants;
- Options d'affichage;
- Ouverture d'autres modules ou vues;
- Changement du mot de passe et des paramètres de l'utilisateur.

La Barre d'Outils

La **Barre d'Outils** permet :

- l'ouverture des autres modules de Oracle Data Integrator;
- de rafraîchir l'affichage des vues;
- d'ouvrir l'aide en ligne;
- de choisir le **contexte courant**. Le contexte que nous utiliseront pour notre projet est le contexte *Global*.

Les Vues

Les objets de Designer disponibles pour l'utilisateur connecté sont organisés sous forme d'arborescences dans les vues suivantes: **Projets**, **Modèles**, **Solutions** et **Autres** (Fonctions Utilisateur, Variables et Séquences Globales).

La majorité des actions que vous pouvez effectuer sur les objets se font au travers de menus contextuels (en sélectionnant l'objet, puis en cliquant avec le bouton droit pour faire apparaître un menu).

Chaque vue apparaît dans un cadre flottant qui peut être ancré sur les bords de la fenêtre principale de Designer. Ces cadres peuvent aussi être empilés. Quand plusieurs cadres de vue sont empilés, des onglets apparaissent au bas du cadre pour accéder à chaque vue.

Les cadres des vues peuvent être déplacés, ancrés ou empilés en sélectionnant et déplaçant le titre ou l'onglet correspondant au cadre. Si un cadre de vue n'apparaît pas dans la fenêtre principale de Designer ou a été fermé, il peut être ré-ouvert en utilisant le menu **Fenêtre > Afficher Vue**.

Certains objets existent déjà dans le référentiel de démonstration :

- Dans la vue **Modèles**, vous trouvez l'ensemble des modèles de données correspondant aux applications *Orders Application*, *Parameters* et *Sales Administration*.
- Dans la vue **Projets**, vous trouvez le projet *Demo* et le dossier *Sales Administration* qui contient déjà certaines interfaces. Vous développerez vos nouvelles interfaces dans ce dossier.

La Zone de Travail

Les fenêtres des objets en cours d'édition ou de consultation apparaissent dans la **zone de travail**.

Le contrôle de données

Introduction à l'intégration de données

Du fait que les applications dans votre organisation lisent et écrivent des données dans des formats hétérogènes, il est impératif d'avoir un processus en place qui vous permettra d'intégrer ces données de sorte qu'elles puissent être utilisées par n'importe qui dans votre entreprise. Il y a de nombreuses méthodes d'intégration de données à votre disposition – ETL, Réplication/Synchronisation de données, etc. Dans tous ces cas, la première étape consiste à assurer la qualité de vos données.

Introduction au contrôle des données

Le contrôle de données est une opération cruciale pour assurer la cohérence globale des applications du système d'information.

Il arrive parfois que les données des applications ne soient pas valides en fonction des contraintes dictées par le système d'information. En effet, la logique de contrôle est souvent remontée au niveau du code des applications, et de ce fait éloignée des données. Il n'est pas rare de trouver des commandes sans client, des lignes de commande sans produit, etc.

Oracle Data Integrator offre un cadre de travail qui permet, très tôt dans le cadre de vos projets, de détecter et reporter ces données dites "erronées".

Il y a deux types de contrôles: le **Contrôle Statique** et le **Contrôle de Flux**. Nous allons détailler leurs différences.

Contrôle statique

Le contrôle statique revient à définir les règles d'intégrité que doivent respecter les données de vos applications. Certaines de ces règles (dites contraintes) sont déjà implémentées directement dans vos serveurs de données (clés primaires, contraintes de référence etc.).

Oracle Data Integrator vous permet de définir des contraintes supplémentaires sans les déclarer directement dans vos serveurs, en vue de valider la cohérence de vos données.

Cette procédure s'appelle **contrôle statique** car elle permet d'opérer des contrôles directement sur les données existantes - statiques - .

Contrôle de flux

Les systèmes cibles à alimenter sont souvent porteurs de règles de contrôle à garantir. Le **contrôle de flux** permet de valider les données entrantes dans une application en fonction des contraintes exprimées. La procédure de contrôle de flux est détaillée dans le chapitre "Les interfaces".

Les avantages

Contrôler les données transférées procure beaucoup d'avantages dont voici les principaux :

- Gain de productivité pour l'utilisation de la base de données cible et ceci sur tout son cycle de vie. En effet, les erreurs sur les données ralentissent la programmation d'applications sur la base de données cible, ainsi que leur exploitation.
- Validation de la modélisation de la base de données cible. Les erreurs détectées ne relèvent pas toujours d'une insuffisance de qualité des données sources mais peuvent montrer une incomplétude du modèle cible. Migrer les données avant de réécrire une application permet donc de valider un nouveau modèle de données tout en fournissant une base de test collant à la réalité.
- Amélioration de la qualité du service à l'utilisateur final

L'assurance de la qualité des données n'est pas simple en terme de réalisation et d'exploitation d'interface. En effet, elle nécessite une gestion de recyclage et d'isolation des données en erreur impliquant une programmation plus complexe notamment lorsque la base de données cible possède un mécanisme de contrôle de contraintes d'intégrité activé. En terme d'exploitation, il convient de mettre en œuvre une démarche de correction des données erronées (sur la source, sur la cible ou sur les flux recyclés).

Exemple

Dans ce qui suit, un exemple va vous guider au travers de la procédure d'audit de la qualité de données (contrôle statique).

L'application "Orders Application - HSQL" présente des données erronées à différents niveaux. L'objectif est de vous faire découvrir de manière très simple les données de cette application qui ne vérifient pas les contraintes dictées par le système d'information.

Contrôle "SRC_CUSTOMER"

Objectif

Les données de la table "SRC_CUSTOMER" ne sont peut-être pas cohérentes par rapport à la règle suivante : "**Aucun client ne doit avoir moins de 21 ans**". Par ailleurs, cette table contient des identifiants de ville (colonne CITY_ID) n'existant peut-être pas dans la table "SRC_CITY", la table des villes.

Ces règles simples n'étaient pas contrôlées par l'application source. De ce fait, certaines données peuvent être erronées.

Traduction du problème

Ce type de règles se traduit par une contrainte de vérification (contrainte CHECK), appelée aussi condition, ainsi que par une référence entre la table "SRC_CITY" et "SRC_CUSTOMER".

Dans Oracle Data Integrator, il suffit d'ajouter une condition en spécifiant la règle à vérifier, et d'ajouter une référence en spécifiant les colonnes référencées et référençantes.

Création des contraintes

Contrainte sur l'âge

1. Dépliez le modèle "Orders Application - HSQL"
2. Dépliez la table "SRC_CUSTOMER"

3. Cliquez avec le bouton droit sur le noeud **Contraintes** et choisissez **Insérer Condition**



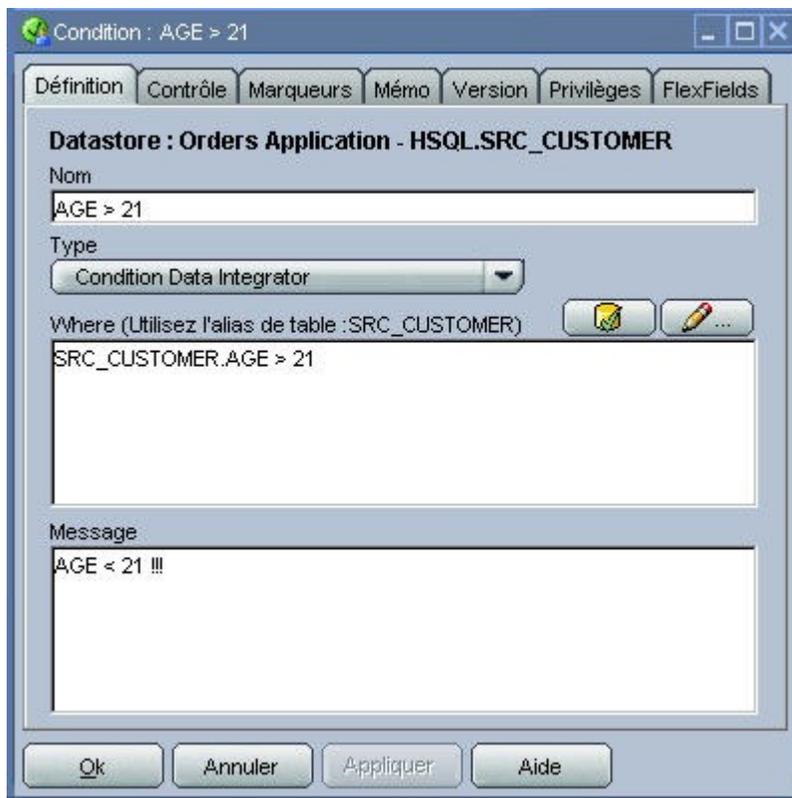
4. Dans l'onglet **Définition**,

- Renseignez le **nom** de votre condition ("AGE > 21" par exemple)
- Choisissez comme **type** de condition **Condition Oracle Data Integrator**
- Dans la clause **Where**, entrez le texte SQL: `SRC_CUSTOMER.AGE > 21`. Vous pouvez saisir ce texte directement ou faire appel à l'éditeur d'expression, accessible par le bouton .

Note : Pour que Oracle Data Integrator génère le bon code SQL, il faut préfixer le nom de colonne AGE par l'**alias** du datastore (SRC_CUSTOMER dans notre cas). Cet alias est défini dans l'onglet **Définition** de l'écran **Datastore** (clic-droit -> **Editer** sur un datastore).

Note: Les contraintes créées par Oracle Data Integrator ne sont pas stockées dans la base de données. Les contraintes sont stockées dans le Référentiel.

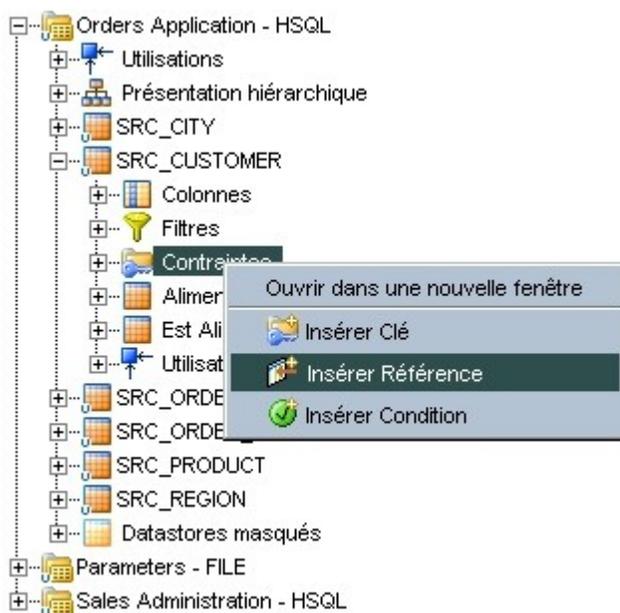
- Dans la zone **Message**, indiquez le message d'erreur tel qu'il apparaîtra dans votre table des erreurs.



5. Cliquez sur **OK** pour enregistrer la contrainte.

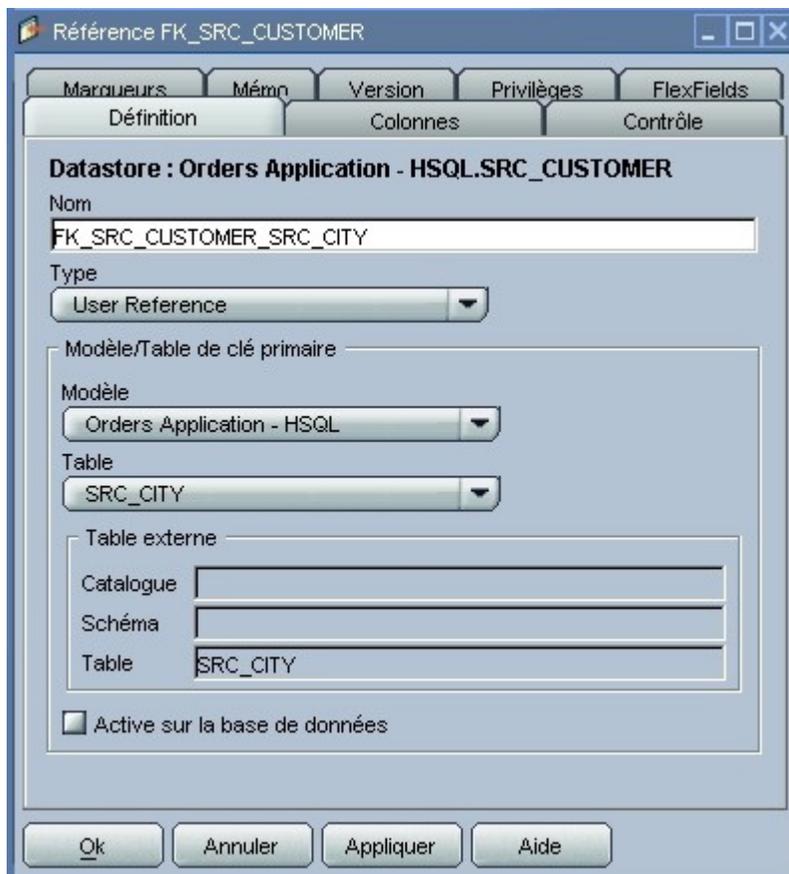
Contrainte Référentielle

1. Dépliez le modèle "Orders Application - HSQL"
2. Dépliez la table "SRC_CUSTOMER"
3. Cliquez avec le bouton droit sur le noeud **Contraintes** et choisissez **Insérer Référence**



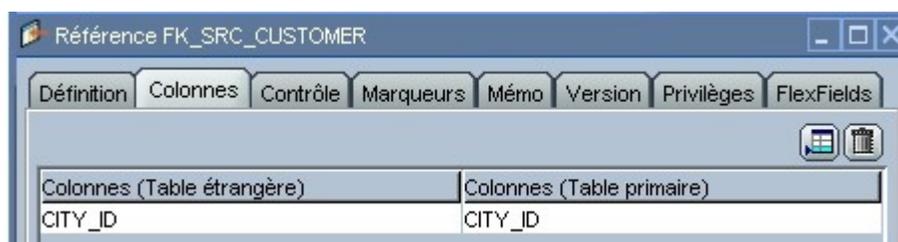
4. Dans l'onglet **Définition**,

- Choisissez comme **type** de référence **User Reference**.
- Dans le menu déroulant correspondant au champ "Table", choisissez "SRC_CITY"
- Votre écran doit ressembler au suivant :



5. Dans l'onglet **Colonnes** :

- Cliquez sur le bouton  afin de définir les colonnes de la référence.
- Choisissez la colonne "CITY_ID" dans la liste des colonnes provenant de la **table étrangère** ("SRC_CUSTOMER") et dans la liste des colonnes provenant de la **table primaire** :



6. Cliquez sur "OK" pour valider la création de cette référence.

Exécution du contrôle statique

1. Faites un clic droit sur le datastore "SRC_CUSTOMER".
2. Choisissez **Contrôle > Contrôler**.
3. Cliquez sur **OK** dans les fenêtres **Exécution** et **Information**.

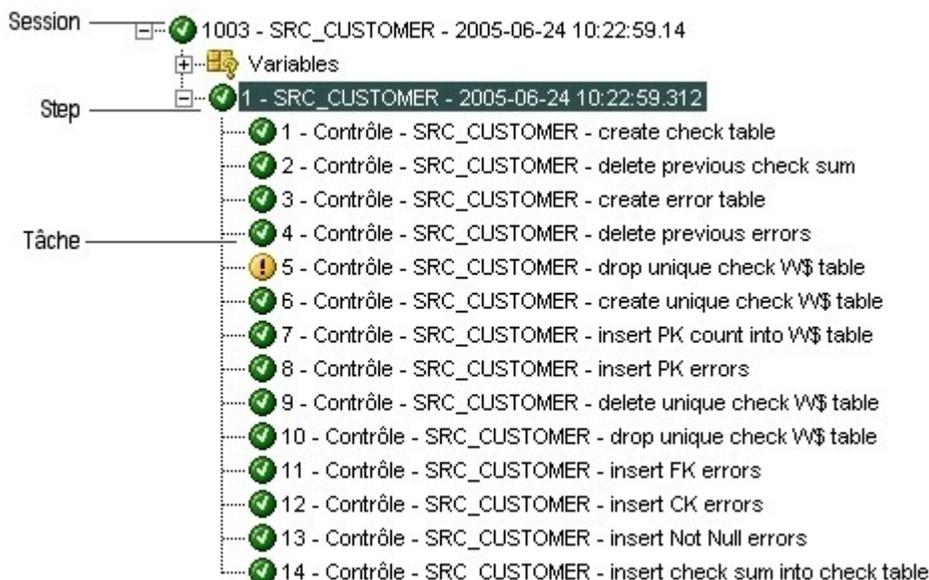


Oracle Data Integrator génère automatiquement le code nécessaires au contrôle de vos données et démarre une session d'exécution.

Suivre l'exécution du contrôle dans Operator

1. Vous pouvez suivre le résultat de votre exécution dans **Operator** en cliquant sur l'icône **Operator**  de votre barre de menu.
2. Dans la fenêtre **Operator** qui s'ouvre vous avez le détail des tâches de la session d'exécution, dans la vue **Liste des Sessions** de l'Operator, sous le noeud **Toutes les Exécutions**.
3. Vous pouvez rafraîchir la fenêtre avec le bouton  si elle était déjà ouverte.

Une session d'exécution apparaît comme suit :



Elle est composée de 3 niveaux :

- La **session**.
- Le **step** (correspond à un objet contrôlé, une interface ou un traitement exécuté)
- La **tâche** (correspond à une tâche élémentaire de l'interface, du contrôle ou du traitement)

Interpréter les résultats

Le nombre d'enregistrements en erreur

1. Dans le **Operator**, ouvrez la fenêtre du **step** (niveau 2) "SRC_CUSTOMER" en double cliquant sur l'icône correspondante dans l'arbre d'exécution. Une nouvelle fenêtre s'ouvre à droite de l'écran.
2. Sélectionnez l'onglet **Exécution** de cette fenêtre. Vous pouvez voir que le contrôle de la table "SRC_CUSTOMER" a donc donné lieu à **9** erreurs. Ces erreurs ont été isolées dans une table d'erreurs :

Nb Insertions	0	Nb Update	0
Nb Delete	0	Nb Erreurs	9

Les données erronées

Vous pouvez avoir accès aux enregistrements erronés en sélectionnant votre table depuis votre modèle et en choisissant **Contrôle > Erreurs** depuis le menu contextuel.

1. Retournez dans le module Designer en cliquant sur le bouton **Designer**  de la barre d'outils
2. Dépliez le modèle "Orders Application - HSQL"
3. Cliquez avec le bouton droit sur le Datastore "SRC_CUSTOMER"
4. Choisissez **Contrôle > Erreurs** dans le menu contextuel
5. La fenêtre listant les erreurs sur les données apparaît :

ERR_TYPE	ERR_MESS	CHECK_DATE	CUSTID	DEAR	LAST_NAME
S	Erreur sur jointure	24/06/05 10:40:27	203	0	Robert
S	AGE > 21 !!!	24/06/05 10:40:27	306	1	Jones
S	AGE > 21 !!!	24/06/05 10:40:27	506	0	Oneda
S	AGE > 21 !!!	24/06/05 10:40:27	206	1	Gentil
S	AGE > 21 !!!	24/06/05 10:40:27	307	2	Hopkins
S	AGE > 21 !!!	24/06/05 10:40:27	407	2	Reinman
S	AGE > 21 !!!	24/06/05 10:40:27	507	0	Okumura
S	AGE > 21 !!!	24/06/05 10:40:27	107	0	Swenson
S	AGE > 21 !!!	24/06/05 10:40:27	207	1	Dupont

Enregistrement 1 sur 9

Le processus de contrôle a permis d'isoler :

- **8** enregistrements qui violent la contrainte "AGE > 21" (qui ont donc moins de 21 ans).
- **1** enregistrement qui viole la contrainte FK_CITY_CUSTOMER (son numéro de ville n'existe pas dans la table "SRC_CITY").

Notez que vous pouvez visualiser l'intégralité de l'enregistrement erroné dans cette fenêtre, et ainsi qualifier les valeurs incorrectes (par exemple le CITY_ID invalide dans le premier enregistrement).

Notez également que le message d'erreur affiché est celui spécifié à la création de la contrainte "AGE > 21".

Maintenant que le contrôle statique a été mis en place, nous pouvons commencer l'implémentation des interfaces d'alimentation.

Les interfaces

Interfaces pour l'intégration de données

Interface "Pop. TRG_CUSTOMER"

Objectif

L'objectif de cette interface est d'alimenter la table cible "TRG_CUSTOMER" du modèle "Sales Administration - HSQL" à partir de la table des clients de "SRC_CUSTOMER" du modèle "Orders Application - HSQL".

Malheureusement, la table "SRC_CUSTOMER" ne contient pas toutes les données attendues.

En effet, vous voulez ajouter à la table cible les informations suivantes :

- La tranche d'âge ("AGE_RANGE") telle que définie dans le fichier "SRC_AGE_GROUP" correspondant à l'âge issu de la table source
- Les nom et prénom du commercial responsable du client ("LAST_NAME" et "FIRST_NAME") tels que définis dans le fichier "SRC_SALES_PERSON" correspondant au numéro de représentant ("SALES_PERS_ID") de la table source
- Le titre du client ("DEAR") stocké dans la table source au format numérique (0,1, 2) transformé en titre lisible (Mr, Mrs, Ms)
- Le nom et le prénom du client source concaténés dans une seule chaîne.

Bien entendu, les données sources ne sont pas toujours cohérentes avec les règles d'intégrité dictées par l'environnement cible. Dans notre cas, 2 règles d'intégrité importantes devront être vérifiées :

- Les clients doivent avoir plus de 21 ans (condition "AGE > 21")
- Les clients doivent être reliés à une ville ("CITY_ID") existante dans la table "TRG_CITY" (référence "FK_CUST_CITY")

Le détail fonctionnel de toutes ces règles ainsi que la procédure à suivre sont fournis ci-dessous :

Définition de l'interface

Datastore cible

Modèle : Sales Administration - HSQL

Datastore : TRG_CUSTOMER

Datastores sources

Modèle	Datastore	Description	Type
--------	-----------	-------------	------

Orders Application - HSQL	SRC_CUSTOMER	Table des clients du système source	Table HSQL
Parameters - FILE	SRC_AGE_GROUP	Fichier des tranches d'âges	Fichier délimité par des points-virgules
Parameters - FILE	SRC_SALES_PERSON	Fichier des représentants	Fichier d'enregistrements à taille fixe

Les jointures

Jointure	Description	Règle SQL	Lieu d'exécution
Clients et Représentants	Pour pouvoir récupérer le nom et le prénom du représentant, il faut joindre SRC_SALES_PERSON et SRC_CUSTOMER	<code>SRC_CUSTOMER.SALES_PERS_ID = SRC_SALES_PERSON.SALES_PERS_ID</code>	Espace de travail
Clients et tranches d'âges	Pour avoir le libellé de la tranche d'âge, il faut que l'âge du client soit compris entre l'âge min et l'âge max du fichier	<code>SRC_CUSTOMER.AGE between SRC_AGE_GROUP.AGE_MIN and SRC_AGE_GROUP.AGE_MAX</code>	Espace de travail

Les règles de transformation

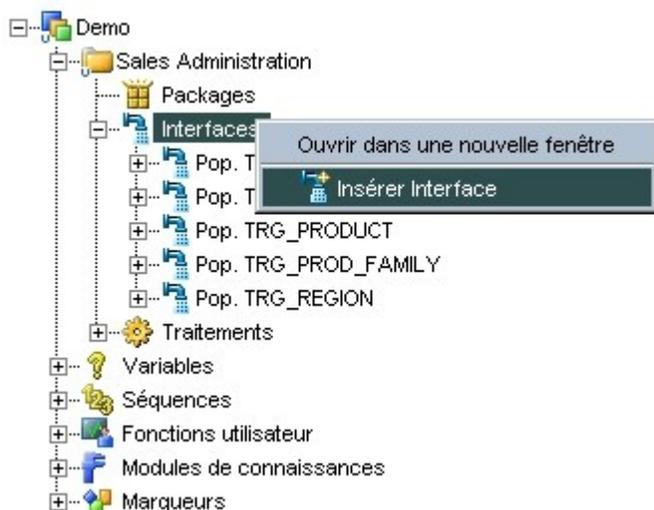
Colonne cible	Provenance	Règle SQL	Lieu d'exécution
CUST_ID	SRC_CUSTOMER.CUSTID	<code>SRC_CUSTOMER.CUSTID</code>	Source
DEAR	Si SRC_CUSTOMER.DEAR = 0 alors 'MR' Si SRC_CUSTOMER.DEAR = 1 alors 'MME' sinon 'MLE'	<code>CASEWHEN (SRC_CUSTOMER.DEAR=0 , 'MR' , CASEWHEN (SRC_CUSTOMER.DEAR=1 , 'MRS' , 'MS'))</code>	Source
CUST_NAME	Concaténation de SRC_CUSTOMER.FIRST_NAME et SRC_CUSTOMER.LAST_NAME en majuscules	<code>SRC_CUSTOMER.FIRST_NAME ' ' UCASE (SRC_CUSTOMER.LAST_NAME)</code>	Source
ADDRESS	SRC_CUSTOMER.ADDRESS	<code>SRC_CUSTOMER.ADDRESS</code>	Source
CITY_ID	SRC_CUSTOMER.CITY_ID	<code>SRC_CUSTOMER.CITY_ID</code>	Source

PHONE	SRC_CUSTOMER.PHONE	SRC_CUSTOMER . PHONE	Source
AGE	SRC_CUSTOMER.AGE	SRC_CUSTOMER . AGE	Source
AGE_RAN GE	SRC_AGE_GROUP.AGE_RA NGE	SRC_AGE_GROUP . AGE_RANGE	Espace de travail
SALES_PE RS	Concaténation de SRC_SALES_PERSON.FIRST _NAME et SRC_SALES_PERSON.LAST _NAME en majuscules	SRC_SALES_PERSON . FIRST_NAME , , UCASE (SRC_SALES_PERSON . LAS T_NAME)	Espace de travail
CRE_DAT E	Date du jour	CURDATE ()	Cible
UPD_DAT E	Date du jour	CURDATE ()	Cible

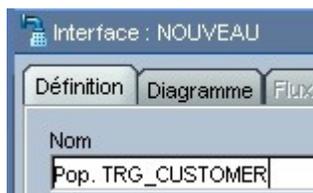
Les étapes à suivre

Insérer une nouvelle interface

1. Dans le dossier "Sales Administration" du projet "Demo", cliquez avec le bouton droit sur Interfaces et choisissez **Insérer Interface**



2. Dans la fenêtre **interface**, tapez le nom de votre interface (Pop. TRG_CUSTOMER) dans le champ **Nom**



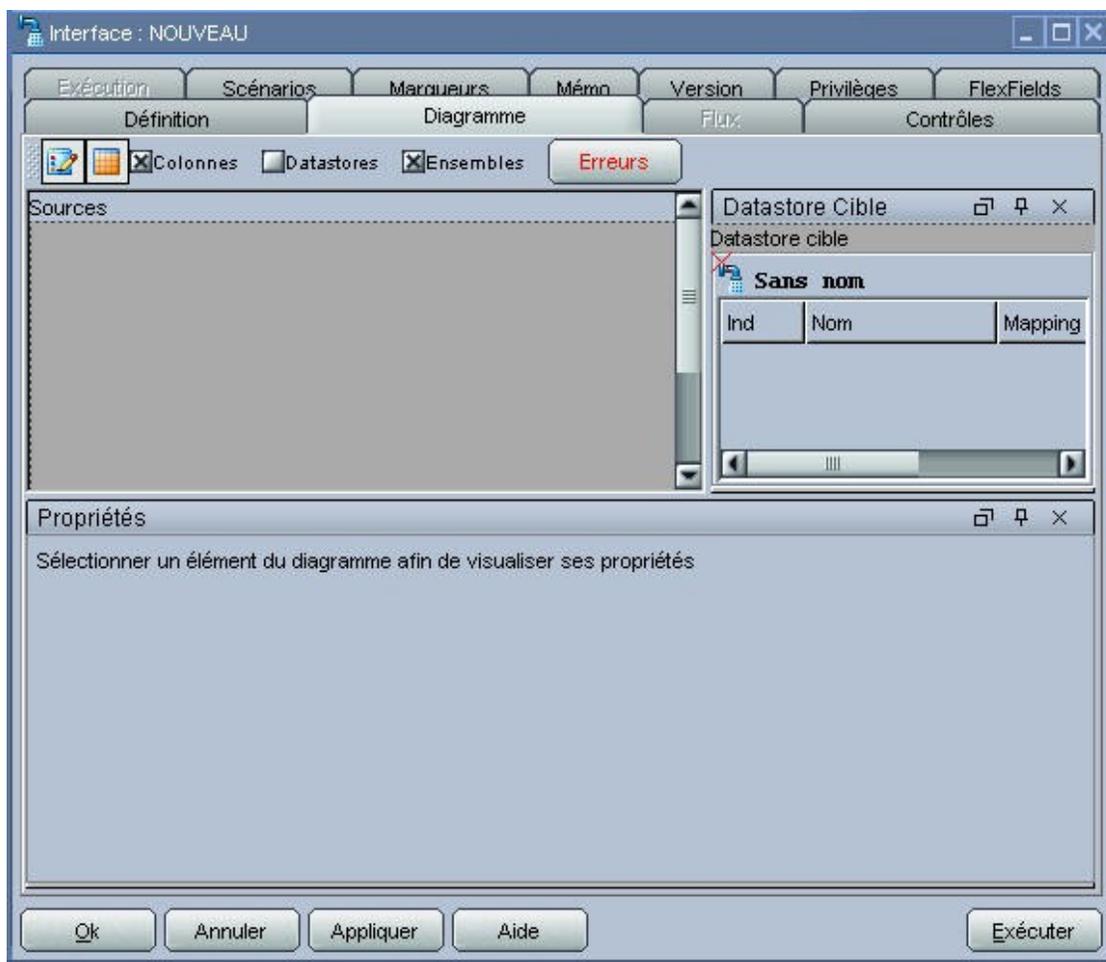
3. Agrandissez votre fenêtre interface.

Définir la table cible

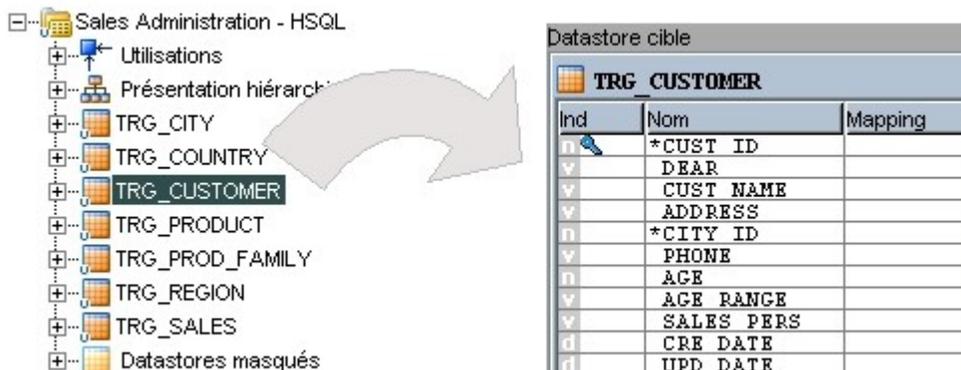
1. Cliquez sur l'onglet **Diagramme**. Une fenêtre d'aide apparaît. Cliquez sur le bouton **OK** pour la fermer.



2. L'onglet **diagramme** de votre interface se présente comme suit :

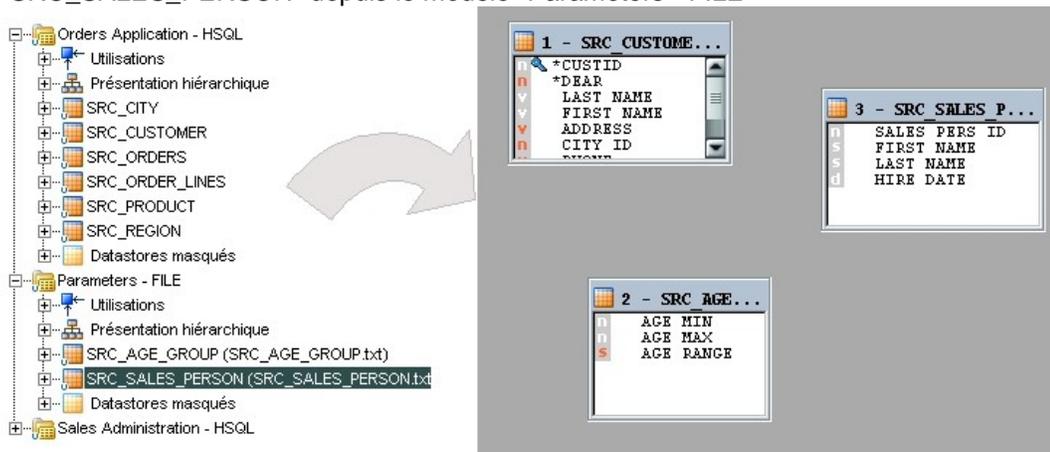


- Sur l'onglet **Diagramme**, glissez la table "TRG_CUSTOMER" depuis le modèle "Sales Administration - HSQL" vers la partie **Datastore cible** à droite de l'écran.



Définir les datastores sources

- Sur l'onglet **Diagramme**, glissez les datastores sources ci-dessous dans le **panneau de composition** (partie gauche de l'onglet diagramme) :
 - "SRC_CUSTOMER" depuis le modèle "Orders Application - HSQL"
 - "SRC_AGE_GROUP" depuis le modèle "Parameters - FILE"
 - "SRC_SALES_PERSON" depuis le modèle "Parameters - FILE"

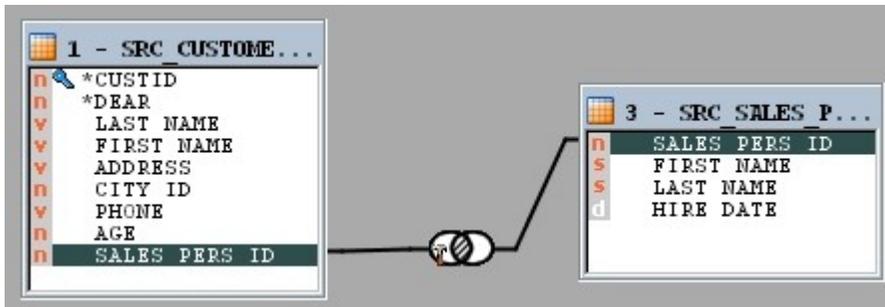


- Une fenêtre apparaît demandant si vous souhaitez que Oracle Data Integrator fasse automatiquement le mapping des colonnes. Cliquez **Oui** pour fermer cette fenêtre.

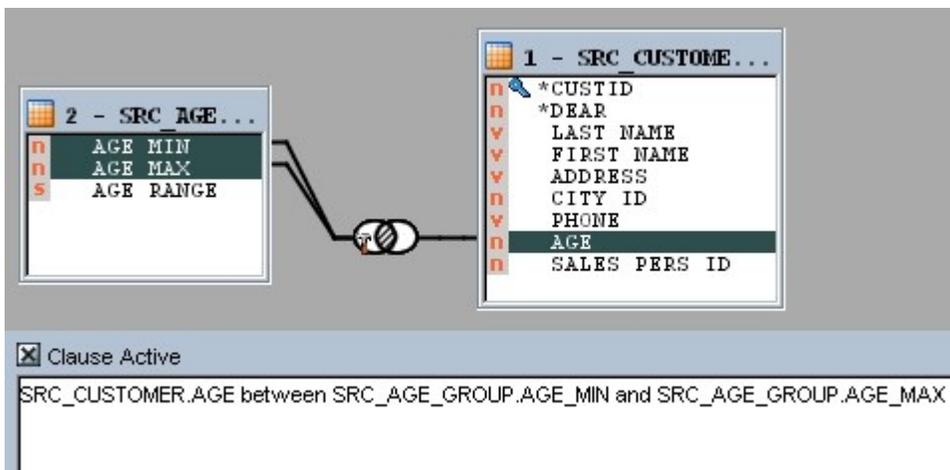


Définir les jointures entre les datastores sources

1. Dans le **panneau de composition**, glissez la colonne "SALES_PERS_ID" de "SRC_CUSTOMER" sur la colonne "SALES_PERS_ID" de "SRC_SALES_PERSON". Ceci permet de définir la première jointure sur le numéro de représentant.



2. Glissez la colonne "AGE" de "SRC_CUSTOMER" sur la colonne "AGE_MIN" de "SRC_AGE_GROUP".
3. Dans le panneau **Propriétés** Modifiez l'**Implémentation** comme suit :
 - Remplacez "=" par " between "
 - Ajoutez " and " à la fin du texte
 - Glissez la colonne "AGE_MAX" de "SRC_AGE_GROUP" dans le champ texte.
4. Vous obtenez le texte suivant : SRC_CUSTOMER.AGE between SRC_AGE_GROUP.AGE_MIN and SRC_AGE_GROUP.AGE_MAX qui correspond à la jointure entre "SRC_CUSTOMER" et "SRC_AGE_GROUP"



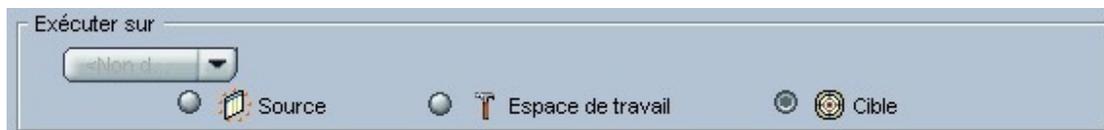
Note : Si des références étaient déjà définies dans les modèles sur les datastore sources, ces références seraient apparues automatiquement comme des jointures dans le diagramme source.

Définir les règles de transformation

Dans la zone **Datastore cible** de votre diagramme, certaines colonnes cibles ont été renseignées par homonymie. Il vous reste à définir les règles pour les zones suivantes : CUSTID, DEAR, CUST_NAME, SALES_PERS, CRE_DATE et UPD_DATE.

La définition des règles de transformation se fait sur chaque colonne cible de la manière suivante :

1. Sélectionnez la colonne cible que vous voulez alimenter en cliquant sur son nom.
2. Dans la zone **Implémentation** qui apparaît en bas à gauche, construisez la règle soit en faisant glisser les colonnes dont vous avez besoin depuis le diagramme des tables sources dans la zone de texte, soit en faisant appel à l'éditeur d'expressions en cliquant sur le bouton .
3. Pensez à spécifier l'environnement dans lequel doivent s'exécuter vos transformations. Pour cela, sélectionnez l'environnement sur lequel vous souhaitez exécuter votre transformation en cochant l'une des cases comme-ci-dessous :

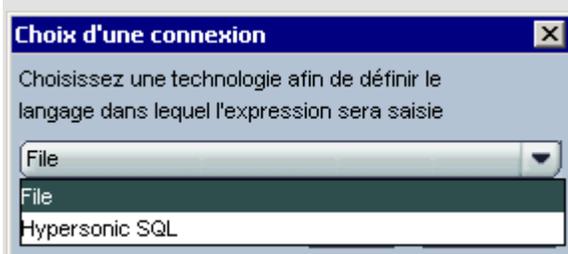


Les règles à implémenter sont les suivantes :

- **CUST_ID** (exécutée en source) : Glissez la colonne "SRC_CUSTOMER.CUSTID" dans le champ **Implémentation**
- **DEAR** (exécutée en source) : Tapez le texte suivant : `CASEWHEN(SRC_CUSTOMER.DEAR=0 , 'MR' , CASEWHEN(SRC_CUSTOMER.DEAR=1 , 'MRS' , 'MS'))`. Bien entendu, vous pouvez glisser au fur et à mesure la colonne "SRC_CUSTOMER.DEAR". Enregistrez votre expression  et validez votre syntaxe .
- **CUST_NAME** (exécutée en source) : Faites appel à l'éditeur d'expression et construisez la règle : `SRC_CUSTOMER.FIRST_NAME || ' ' || UCASE(SRC_CUSTOMER.LAST_NAME)`

Attention : L'éditeur d'expression, qui peut être utilisé pour bâtir les expressions, ne contient pas systématiquement toutes les fonctions d'une technologie. Il ne contient que les fonctions communes à un grand nombre de technologies. Le fait qu'une fonction n'apparaisse pas dans l'éditeur d'expression ne doit pas empêcher de l'utiliser en la tapant.

Note : Si lors de l'appel à l'éditeur d'expression, la fenêtre ci-dessous apparaît,



Choisissez "Hypersonic SQL" et cliquez sur **OK**. Cet écran apparaît afin de spécifier à Oracle Data Integrator le langage SQL que vous allez utiliser via l'éditeur d'expression.

Note : Si vous exécutiez ce mapping sur la cible, l'éditeur d'expressions vous donnerait la syntaxe pour le système de cible (dans ce cas, également Hypersonic SQL)

- **SALES_PERS** (exécutée dans l'espace de travail) : faites appel à l'éditeur d'expression et construisez la règle : `SRC_SALES_PERSON.FIRST_NAME || ' ' || UCASE(SRC_SALES_PERSON.LAST_NAME)`.

Attention : Cette règle doit être exécutée dans l'espace de travail !

- **CRE_DATE** et **UPD_DATE** (exécutée en cible) : saisissez la fonction `CURDATE()` et cochez la case "Mapping Actif". Choisissez ensuite d'exécuter l'expression en cible. Avec **CRE_DATE**, vous voulez que le mapping ne soit exécuté que lors d'un Insert. Pour ceci, scrollez en bas de la fenêtre de mapping et décochez la case Update. Avec **UPD_DATE**, vous ne voulez que la date où la ligne a été mise à jour. Pour ceci, décochez le champ Insert dans le mapping de **UPD_DATE**.



Note : Le bouton  fait appel à votre serveur source pour vérifier la syntaxe du code SQL que vous avez saisi. Cette vérification ne peut se faire que lorsque votre règle est exécutée intégralement sur le serveur source. En cas d'erreur de saisie, la fenêtre qui apparaît vous donne accès au bouton **Détail** qui vous renvoie le message d'erreur remonté par le serveur source.

Au final, vos règles de transformation doivent ressembler à l'écran ci-dessous.

Ind	Nom	Mapping
n	*CUST ID	SRC CUSTOMER.CUSTID
v	DEAR	CASEWHEN(SRC CUSTOMER.DEA...
v	CUST NAME	SRC CUSTOMER.FIRST NAME ...
v	ADDRESS	SRC CUSTOMER.ADDRESS
n	*CITY ID	SRC CUSTOMER.CITY ID
v	PHONE	SRC CUSTOMER.PHONE
n	AGE	SRC CUSTOMER.AGE
v	AGE RANGE	SRC AGE GROUP.AGE RANGE
v	SALES PERS	SRC SALES PERSON.FIRST NA...
d	CRE DATE	CURDATE()
d	UPD DATE	CURDATE()

Les icônes à gauche indiquent la première lettre du type de données de la colonne cible (n : numeric, v : varchar, d: date). Les autres icônes indiquent le lieu d'exécution des expressions:

	source
	espace de travail
	cible

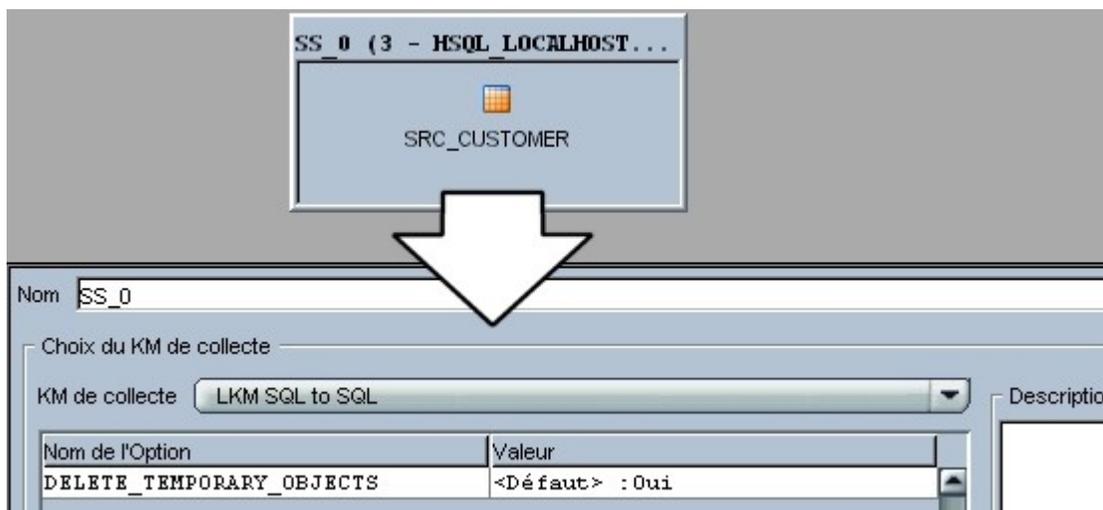
Note : Lorsque l'indicateur **Ind** de votre mapping est à , cela signifie que vous avez commis une erreur. Dans ce cas, sélectionnez le champ cible incriminé et vérifiez ce que vous avez saisi. Alternativement, vous pouvez cliquer sur le bouton **Erreurs** en haut de l'onglet **Diagramme**, puis sur le bouton **Détail** de la fenêtre qui apparaît pour avoir une information plus détaillée du type d'erreur que vous avez commis.

Choisir les stratégies de collecte de données (LKM)

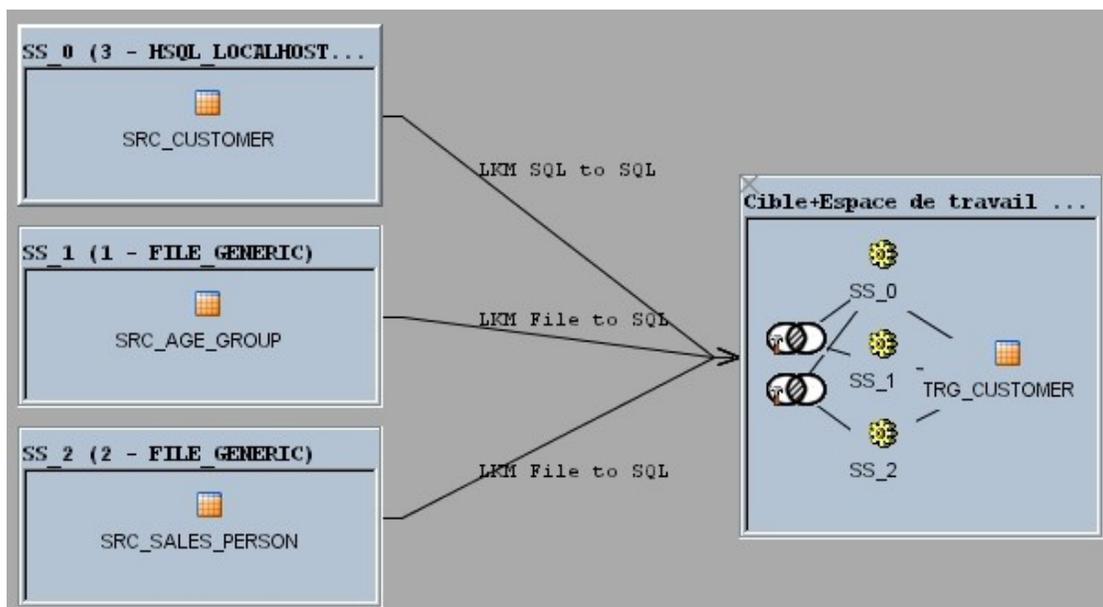
Dans l'onglet **Flux**, Data Integrator vous indique les différentes étapes qu'il doit effectuer pour arriver à exécuter votre interface.

Il vous reste à choisir les manières de faire (**modules de connaissance de collecte - LKM**) pour rapatrier les données des fichiers "SRC_AGE_GROUP", "SRC_SALES_PERSON" et de la table "SRC_CUSTOMER" vers votre environnement cible.

1. Cliquez sur le premier **Source Set** (SS_x) correspondant à la collecte du fichier "SRC_AGE_GROUP" et choisissez dans la liste de modules de connaissance de collecte le module : "LKM File to SQL"



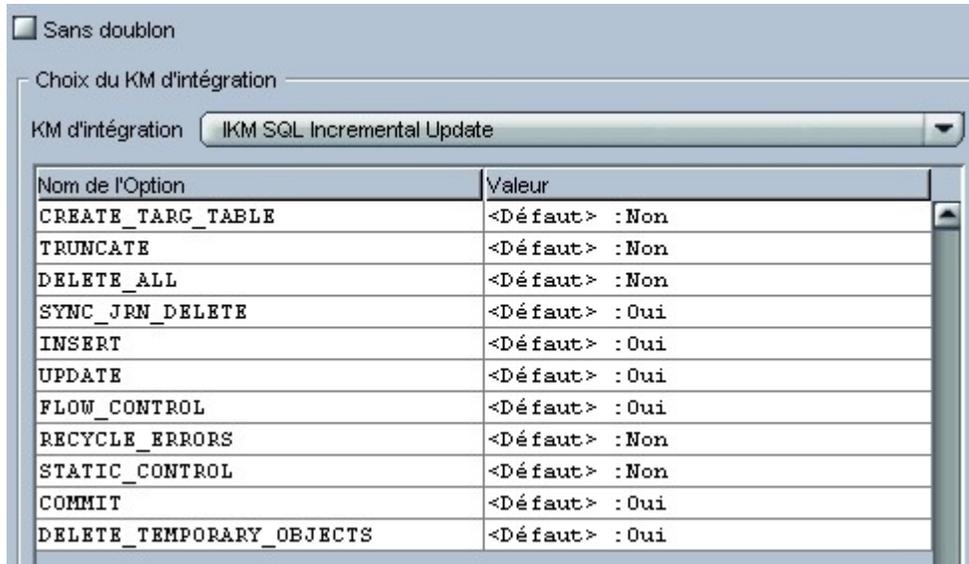
2. Cliquez sur le second **Source Set** (SS_x) correspondant à la collecte du fichier "SRC_SALES_PERSON" et choisissez dans la liste de modules de connaissance de collecte le module : "LKM File to SQL"
3. Cliquez sur le dernier **Source Set** (SS_x) correspondant à la collecte de la table "SRC_CUSTOMER" et choisissez dans la liste de modules de connaissance de collecte le module : "LKM SQL to SQL"
4. Au final votre écran **Flux** devrait ressembler à l'image ci-dessous :



Choisir la stratégie d'intégration des données (IKM)

A l'issue de la phase de collecte, vous devez choisir la stratégie d'alimentation (ou d'intégration) de vos données dans la table cible. Pour cela, vous devez indiquer le module de connaissance d'intégration (IKM) à utiliser.

1. Cliquez sur l'objet **Cible+Espace de Travail** et choisissez le module de connaissance "**IKM SQL Incremental Update**"



2. Dans les options du module de connaissance, laissez les valeurs par défaut, comme ci-dessus.

Ces options permettent de contrôler certains aspects de la stratégie d'intégration, par exemple, avec l'option `FLOW_CONTROL`, vous avez demandé à Oracle Data Integrator de déclencher le contrôle des données de votre flux avant de l'insérer dans votre table cible.

Note : Seul un petit nombre de module de connaissance sont disponibles dans l'environnement de démonstration. Pour utiliser tous les modules disponibles avec Data Integrator, vous devez les importer dans votre projet. Cette étape n'est pas requise pour le projet actuel. Pour plus d'informations sur cette opération, référez-vous au Guide de l'Utilisateur.

Choisir la stratégie de contrôle des données

Jusqu'à présent, vous vous êtes contenté de préciser la manière de constituer le flux de données depuis la source. Il vous reste à définir la manière de contrôler vos données (CKM) ainsi que les contraintes ou conditions à vérifier avant l'intégration.

1. Dans l'onglet **Contrôles**, choisissez le module de connaissance de contrôle : "**CKM HSQL**"



- Définissez les contraintes à vérifier : "AGE > 21", 'PK_TRG_CITY' et "FK_CUST_CITY"

Contraintes	
Nom de la contrainte	Valeur
AGE >21	<Défaut> :Oui
PK_TRG_CUSTOMER	<Défaut> :Oui
FK_CUST_CITY	<Défaut> :Oui

Vous êtes maintenant prêt à exécuter votre interface.

Exécuter l'interface

- Cliquez sur le bouton **Exécuter** dans la fenêtre **Interface** pour exécuter votre interface.
- Une fenêtre vous demande de confirmer la sauvegarde de l'interface. Cliquez sur **Oui**.
- Cliquez sur **OK** dans les fenêtres **Exécution** et **Information**.

Oracle Data Integrator génère automatiquement l'ensemble des traitements et démarre une session d'exécution.

Suivre l'exécution de l'interface dans Operator

- Vous pouvez suivre le résultat de votre exécution dans **Operator** en cliquant sur l'icône  de votre barre de menu.
- Dans la fenêtre **Operator** qui s'ouvre (que vous pouvez rafraîchir avec le bouton  si elle était déjà ouverte), vous avez le détail des tâches de la session d'exécution.

Interpréter les résultats

Les enregistrements traités

- Dans **Operator**, éditez la fenêtre du **Step** "Pop. TRG_CUSTOMER" en double cliquant sur l'icône correspondante dans l'arbre d'exécution. Une nouvelle fenêtre s'ouvre à droite de l'écran.
- Dans l'onglet **Exécution** de cette fenêtre, vous pouvez voir que l'alimentation de la table TRG_CUSTOMER a donc donné lieu à **25** insertions et isolé **9** erreurs dans une table d'erreurs.

Nb Insertions	25	Nb Update	0
Nb Delete	0	Nb Erreurs	9

Les données insérées

Vous pouvez consulter les données insérées par votre interface depuis le module **Designer**

- Dépliez le modèle "Sales Administration - HSQL" et sélectionnez la table "TRG_CUSTOMER".

2. Avec le bouton droit, choisissez **Données**, pour consulter les données de la table cible.

CUST_ID	DEAR	CUST_NAME	ADDRESS	CITY_ID	PHONE	AGE
103	MR	Peter TRAVIS	7835 Hartford Driv	12	(510) 555 4448	34
104	MR	Joe LARSON	87 Carmel Blvd.	13	(213) 555 5095	45
105	MR	Tony GOLDSCHM	91 Torre drive	14	(619) 555 6529	55
106	MR	William BAKER	2890 Grant Avenu	15	(312) 555 7040	64

Les données erronées

L'interface que vous avez exécutée a isolé 9 erreurs dans une table d'erreurs automatiquement créée pour vous. Vous pouvez avoir accès aux enregistrements erronés en sélectionnant votre table cible (TRG_CUSTOMER) depuis votre modèle et en choisissant **Contrôle > Erreurs** depuis le menu contextuel.

ERR_TYPE	ERR_MESS	CHECK_DATE	CUST_ID	DEAR	CUST_NAME
F	Join error (FK_CU	24/06/05 14:11:03	207	MRS	Marie-Chantale
F	Join error (FK_CU	24/06/05 14:11:03	203	MR	Christian ROBE
F	Age < 21 !!!	24/06/05 14:11:03	306	MRS	Mary JONES
F	Age < 21 !!!	24/06/05 14:11:03	506	MR	Kenji ONEDA
F	Age < 21 !!!	24/06/05 14:11:03	206	MRS	Michele GENTIL
F	Age < 21 !!!	24/06/05 14:11:03	407	MS	Heineke REINM,
F	Age < 21 !!!	24/06/05 14:11:03	507	MR	Isao OKUMURA
F	Age < 21 !!!	24/06/05 14:11:03	107	MR	Jack SWENSON
F	Age < 21 !!!	24/06/05 14:11:03	207	MRS	Marie-Chantale

Enregistrement 1 sur 9

L'interface a rejeté :

- 2 enregistrements qui violent la contrainte "FK_CUST_CITY" (qui ont donc un numéro de ville inexistant dans la table des villes).
- 7 enregistrements qui ont moins de 21 ans (contrainte "AGE > 21")

Les données erronées n'ont pas été intégrées dans la table cible.

Modification d'erreurs depuis la source de données

1. Dépliez l'arborescence du modèle "Orders Application - HSQL"
2. Faites un click-droit sur le datastore "SRC_CUSTOMER" et choisissez l'option **Données**
3. Recherchez le client ayant pour identifiant la valeur "203". Vous pouvez trier le tableau en cliquant sur les en-têtes des colonnes. Si le client 203 n'est pas visible, cliquez sur le bouton **Actualiser** pour rafraîchir l'affichage.
4. Double-cliquez sur la valeur de la colonne "CITY_ID" de ce client afin de la modifier. La nouvelle valeur doit être 107 (au lieu de 208)

CUSTID	DEAR	LAST_NAME	FIRST_NAME	ADDRESS	CITY_ID	PHONE	AGE
101	0	Brendt	Paul	10 Jasper Blvd.	107	(212) 555 2146	69
102	0	McCarthy	Robin	27 Pasadena Drive	11	(214) 555 3075	29
103	0	Travis	Peter	7835 Hartford Drive	12	(510) 555 4448	34
104	0	Larson	Joe	87 Carmel Blvd.	13	(213) 555 5095	45
105	0	Goldschmidt	Tony	91 Torre drive	14	(619) 555 6529	55
106	0	Baker	William	2890 Grant Avenue	15	(312) 555 7040	64
107	0	Swenson	Jack	64 Imagination Drive	19	(202) 555 8125	20
201	0	Sartois	Jean	71 rue Rousseau	25	79 23 26 23	69
202	0	Michaud	Philippe	197 impasse Renc	23	78 21 86 20	22
203	0	Robert	Christian	1rue Cezanne	107	42 25 27 29	36
204	1	Martin	Christine	12 allée Victor Hug	24	25 26 46 26	42
205	0	Piaget	Luc	38 allée des Saule	29	53 42 24 28	56
206	1	Gentil	Michele	17montée des Ché	25	65 62 26 13	19
207	1	Dupont	Marie-Chantale	37 rue Murat	20	46 72 23 53	20
301	1	Edwards	Caroline	68 Downing Street	35	243867945	68
302	0	Marlow	Justin	290 Yorkshire Drive	32	653643634	29
303	0	Wilson	John	28 Sutton Row	33	158746231	34
304	0	McCartney	George	45 Glenholme Road	34	323768678	47

5. Cliquez sur **Appliquer** puis **OK**
6. Exécutez à nouveau l'interface "Pop. TRG_CUSTOMER" en choisissant l'option **Exécuter** dans son menu contextuel.

Les enregistrements traités

1. Dans **Operator**, cliquez sur le bouton 
2. Éditez la fenêtre du **Step** "Pop. TRG_CUSTOMER" en double cliquant sur l'icône correspondante dans l'arbre d'exécution. Une nouvelle fenêtre s'ouvre à droite de l'écran.
3. Dans l'onglet **Exécution** de cette fenêtre, vous pouvez que l'alimentation de la table TRG_CUSTOMER a maintenant donné lieu à 1 insertion (l'enregistrement que vous venez de

modifier) et isolé **8** erreurs dans la table d'erreurs.

Nb Insertions	1	Nb Update	0
Nb Delete	0	Nb Erreurs	8

Interfaces pour l'intégration de données

Interface "Pop. TRG_SALES"

Objectif

L'objectif de cette interface est d'alimenter la table cible "TRG_SALES" du modèle "Sales Administration - HSQL" à partir de la table des commandes "SRC_ORDERS" et de la table des lignes de commandes "SRC_ORDER_LINES" du modèle "Orders Application - HSQL". Les données doivent être agrégées avant d'intégrer la table cible. Seules les commandes ayant un status 'CLO' sont à prendre en compte.

Malheureusement, les données sources ne sont pas toujours cohérentes avec les règles d'intégrité dictées par l'environnement cible. Dans notre cas, 2 règles d'intégrité importantes devront être vérifiées :

- Les ventes doivent être reliées à un client ("CUST_ID") existant dans la table "TRG_CUSTOMER" (référence "FK_SALES_CUST")
- Les ventes doivent être reliées à un produit ("PRODUCT_ID") existant dans la table "TRG_PRODUCT" (référence "FK_SALES_PROD")

Le détail fonctionnel des règles et la procédure à suivre sont fournis ci-dessous.

Définition de l'interface

Datastore cible

Modèle : Sales Administration - HSQL

Datastore : TRG_SALES

Datastores sources

Modèle	Datastore	Description	Type
Orders Application - HSQL	SRC_ORDERS	Table des commandes du système source	Table HSQL
Orders Application - HSQL	SRC_ORDER_LINES	Table des lignes de commandes du système source	Table HSQL

Les jointures

Jointure	Description	Règle SQL	Lieu d'exécution
Commandes et Lignes de commandes	Il faut joindre SRC_ORDERS et SRC_ORDER_LINES	SRC_ORDERS.ORDER_ID = SRC_ORDER_LINES.ORDER_ID	Source

Les filtres

Description	Règle SQL	Lieu d'exécution
Ne récupérer que les commandes complètes (CLOSED)	SRC_ORDERS.STATUS = 'CLO'	Source

Les règles de transformation

Colonne cible	Provenance	Règle SQL	Lieu d'exécution
CUST_ID	CUST_ID depuis SRC_ORDERS	SRC_ORDERS.CUST_ID	Source
PRODUCT_ID	PRODUCT_ID depuis SRC_ORDER_LINES	SRC_ORDER_LINES.PRODUCT_ID	Source
FIRST_ORD_ID	Plus petite valeur de ORDER_ID	MIN(SRC_ORDERS.ORDER_ID)	Source
FIRST_ORD_DATE	Plus petite valeur de la date de commande ORDER_DATE depuis SRC_ORDERS	MIN(SRC_ORDERS.ORDER_DATE)	Source
LAST_ORD_ID	Plus grande valeur de ORDER_ID	MAX(SRC_ORDERS.ORDER_ID)	Source
LAST_ORD_DATE	Plus grande valeur de la date de commande ORDER_DATE depuis SRC_ORDERS	MAX(SRC_ORDERS.ORDER_DATE)	Source
QTY	Somme des quantités QTY depuis les lignes de commandes	SUM(SRC_ORDER_LINES.QTY)	Source
AMOUNT	Somme des montants depuis les lignes de commandes	SUM(SRC_ORDER_LINES.AMOUNT)	Source
PROD_AVG_PRICE	Moyenne des montants depuis les lignes de	AVG(SRC_ORDER_LINES.AMOUNT)	Source

Les étapes à suivre

Les étapes suivantes vont vous guider pour la réalisation de cette interface :

Insérer une nouvelle interface

1. Dans le dossier "Sales Administration" du projet "Demo", cliquez avec le bouton droit sur Interfaces et choisissez **Insérer Interface**
2. Dans la fenêtre **interface**, tapez le nom de votre interface (Pop. TRG_SALES) dans le champ **Nom**
3. Agrandissez votre fenêtre interface.

Définir la table cible

1. Sur l'onglet **Diagramme**, glissez la table "TRG_SALES" depuis le modèle "Sales Administration - HSQL" vers la partie **Datastore Cible**.

Définir les datastores sources

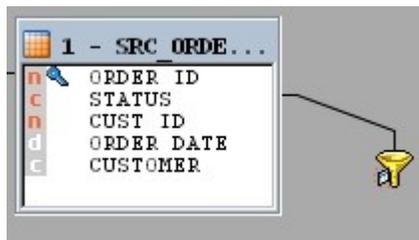
1. Sur l'onglet **Diagramme**, glissez les datastores ci-dessous dans le **panneau de composition** (partie gauche du **diagramme**) :
 - "SRC_ORDERS" depuis le modèle "Orders Application - HSQL"
 - "SRC_ORDER_LINES" depuis le modèle "Orders Application - HSQL"

Définir les jointures entre les datastores sources

1. Dans le **panneau de composition**, glissez la colonne "ORDER_ID" de "SRC_ORDERS" sur la colonne "ORDER_ID" de "SRC_ORDER_LINES". Ceci permet de définir la jointure sur le numéro de commande.

Définir le filtre commandes

1. Dans le **diagramme**, glissez la colonne "STATUS" de "SRC_ORDERS" sur le **panneau de composition**.
2. Modifiez l'**Implémentation** du filtre pour obtenir la règle : `SRC_ORDERS.STATUS = 'CLO'`



Définir les règles de transformation

Dans la zone **Datastore cible** de votre diagramme, certaines colonnes cibles ont été renseignées par homonymie. Il vous reste à définir les règles pour les zones suivantes : FIRST_ORD_ID, FIRST_ORD_DATE, LAST_ORD_ID, LAST_ORD_DATE, QTY, AMOUNT et PROD_AVG_PRICE

Comme dans les exercices précédents, le nom de chaque colonne est représenté sous une forme de tableau. Chaque ligne de notre tableau correspond à un nom de colonne. La définition des règles de transformation se fait sur chaque champ de la table cible de la manière suivante :

1. Sélectionnez le champ de la table cible que vous voulez alimenter
2. Dans la zone **Implémentation** qui apparaît en bas à gauche, construisez la règle soit en faisant glisser les colonnes dont vous avez besoin depuis le diagramme des tables sources dans la zone de texte, soit en faisant appel à l'éditeur d'expressions sur le bouton 

Les règles à implémenter sont les suivantes :

- FIRST_ORD_ID : Glissez la colonne "SRC_ORDERS.ORDER_ID" dans la zone **Implémentation**. Modifiez la règle pour obtenir : `MIN(SRC_ORDERS.ORDER_ID)`
- FIRST_ORD_DATE : Glissez la colonne "SRC_ORDERS.ORDER_DATE" dans la zone **Implémentation**. Modifiez la règle pour obtenir : `MIN(SRC_ORDERS.ORDER_DATE)`
- LAST_ORD_ID : Glissez la colonne "SRC_ORDERS.ORDER_ID" dans la zone **Implémentation**. Modifiez la règle pour obtenir : `MAX(SRC_ORDERS.ORDER_ID)`
- LAST_ORD_DATE : Glissez la colonne "SRC_ORDERS.ORDER_DATE" dans la zone **Implémentation**. Modifiez la règle pour obtenir : `MAX(SRC_ORDERS.ORDER_DATE)`
- QTY : Modifiez la règle pour obtenir : `SUM(SRC_ORDER_LINES.QTY)`
- AMOUNT : Modifiez la règle pour obtenir : `SUM(SRC_ORDER_LINES.AMOUNT)`
- PROD_AVG_PRICE : Glissez la colonne "SRC_ORDERLINES.AMOUNT" dans la zone **Implémentation**. Modifiez la règle pour obtenir : `AVG(SRC_ORDER_LINES.AMOUNT)`

Au final, vos règles de transformation doivent ressembler à l'écran ci-dessous.

Ind	Nom	Mapping
n	*CUST ID	SRC_ORDERS.CUST_ID
n	*PRODUCT ID	SRC_ORDER_LINES.PRODUCT_ID
n	*FIRST ORD ID	MIN(SRC_ORDERS.ORDER_ID)
d	*FIRST ORD ...	MIN(SRC_ORDERS.ORDER_DATE)
n	*LAST ORD ID	MAX(SRC_ORDERS.ORDER_ID)
d	*LAST ORD ...	MAX(SRC_ORDERS.ORDER_DATE)
n	*QTY	SUM(SRC_ORDER_LINES.QTY)
n	*AMOUNT	SUM(SRC_ORDER_LINES.AMOUNT)
n	*PROD AVG ...	AVG(SRC_ORDER_LINES.AMOUNT)

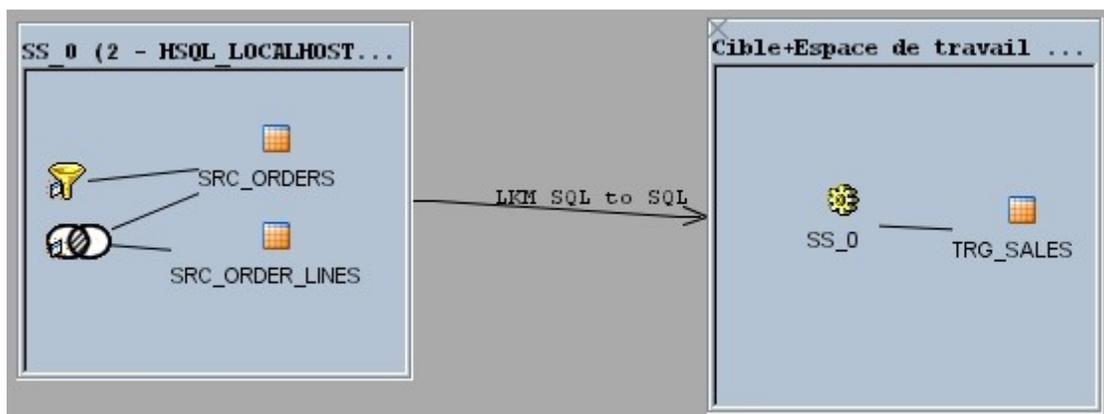
Choisir les stratégies de collecte de données (LKM)

Dans l'onglet **Flux**, Data Integrator vous indique les différentes étapes qu'il doit effectuer pour arriver à exécuter votre interface.

Il vous reste à choisir la manière de faire (**module de connaissance de collecte** - LKM) pour collecter le résultat de l'agrégat de commandes et lignes de commandes vers votre environnement cible.

1. Cliquez sur le **Source Set** (SS_0) correspondant à la collecte du résultat de l'agrégat filtré de vos lignes de commandes et choisissez dans la liste de modules de connaissance de collecte le module : "**LKM SQL to SQL**"

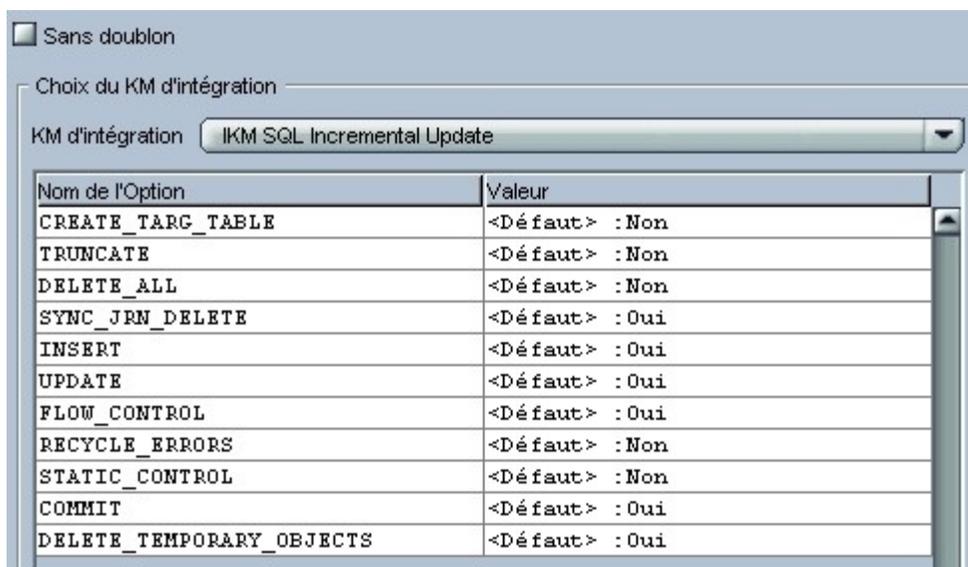
2. Au final votre écran Flux devrait ressembler à l'image ci-dessous :



Choisir la stratégie d'intégration des données (IKM)

A l'issue de la phase de collecte, vous devez choisir la stratégie d'alimentation (ou d'intégration) de vos données dans la table cible. Pour cela, vous devez indiquer le module de connaissance d'intégration (IKM) à utiliser.

1. Cliquez sur l'objet **Cible+Espace de Travail** et choisissez le module de connaissance "IKM SQL Incremental Update"



2. Dans les options du module de connaissance, laissez les valeurs par défaut comme ci-dessus.

Comme précédemment, Oracle Data Integrator déclenchera le contrôle des données de votre flux avant de l'insérer dans votre table cible.

Choisir la stratégie de contrôle des données

Jusqu'à présent, vous vous êtes contenté de préciser la manière de constituer le flux de données depuis la source. Il vous reste à définir la manière de contrôler vos données (CKM) ainsi que les contraintes ou conditions à vérifier avant l'intégration.

1. Dans l'onglet **Contrôles**, choisissez le module de connaissance de contrôle : "CKM HSQL"

Nom de l'Option	Valeur
DROP_ERROR_TABLE	<Défaut> :Non
DROP_CHECK_TABLE	<Défaut> :Non

2. Définissez les contraintes à vérifier : "PK_TRG_SALES", "FK_SALES_CUST" et "FK_SALES_PROD"

Nom de la contrainte	Valeur
PK_TRG_SALES	<Défaut> :Oui
FK_SALES_CUST	<Défaut> :Oui
FK_SALES_PROD	<Défaut> :Oui

Vous êtes maintenant prêt à exécuter votre interface.

Exécuter l'interface

1. Cliquez sur le bouton **Exécuter** dans la fenêtre **Interface** pour exécuter votre interface.
2. Une fenêtre vous demande de confirmez la sauvegarde de l'interface. Cliquez sur **Oui**.
3. Cliquez sur **OK** dans les fenêtres **Exécution** et **Information**.

Oracle Data Integrator génère automatiquement l'ensemble des traitements et démarre une session d'exécution.

Suivre l'exécution de l'interface dans Operator

1. Vous pouvez suivre le résultat de votre exécution dans **Operator** en cliquant sur l'icône  de votre barre de menu.
2. Dans la fenêtre **Operator** qui s'ouvre (que vous pouvez rafraîchir avec le bouton  si elle était déjà ouverte), vous avez le détail des tâches de la session d'exécution.

Interpréter les résultats

Les enregistrements traités

1. Dans Operator, ouvrez la fenêtre du **Step** "Pop. TRG_SALES" en double cliquant sur l'icône correspondante dans l'arbre d'exécution. Une nouvelle fenêtre s'ouvre à droite de l'écran.
2. Dans l'onglet **Exécution** de cette fenêtre, vous pouvez voir le nombre d'enregistrements traités par votre interface :

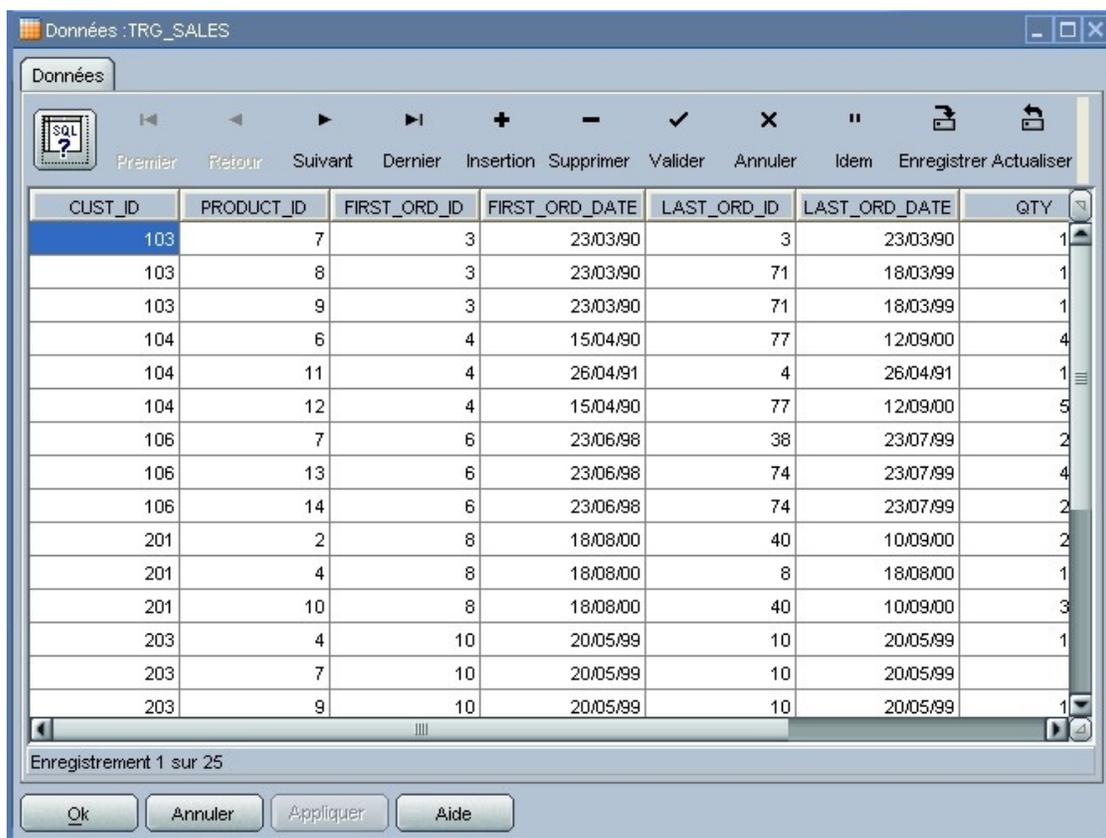
Nb Insertions	62	Nb Update	0
Nb Delete	0	Nb Erreurs	32

L'alimentation de la table TRG_SALES a donc donné lieu à **62** insertions et isolé **32** erreurs dans une table d'erreurs.

Les données insérées

Vous pouvez consulter les données résultat de votre interface depuis le module **Designer**

1. Dépliez le modèle "Sales Administration - HSQL" et sélectionnez la table "TRG_SALES".
2. Faites un clic-droit, choisissez **Données** pour consulter les données de la table cible.



Les données erronées

L'interface que vous avez exécutée a isolé **32** erreurs dans une table d'erreurs automatiquement créée pour vous. Vous pouvez avoir accès aux enregistrements erronés en sélectionnant votre table cible (TRG_SALES) depuis votre modèle et en choisissant **Contrôle > Erreurs** depuis le menu contextuel.

ERR_TYPE	ERR_MESS	CHECK_DATE	CUST_ID	PRODUCT_ID	FIRST_ORD
F	Join error (FK_SA	27/06/05 10:38:56	1 002	4	
F	Join error (FK_SA	27/06/05 10:38:56	1 002	5	
F	Join error (FK_SA	27/06/05 10:38:56	1 002	6	
F	Join error (FK_SA	27/06/05 10:38:56	207	2	
F	Join error (FK_SA	27/06/05 10:38:56	207	6	
F	Join error (FK_SA	27/06/05 10:38:56	207	15	
F	Join error (FK_SA	27/06/05 10:38:56	306	1	
F	Join error (FK_SA	27/06/05 10:38:56	306	6	
F	Join error (FK_SA	27/06/05 10:38:56	306	15	
F	Join error (FK_SA	27/06/05 10:38:56	506	1	

Enregistrement 1 sur 25

Ok Annuler Appliquer Aide

L'interface a rejeté :

- **31** enregistrements qui violent la contrainte "FK_SALES_CUST" (qui ont donc un numéro de client inexistant dans la table des clients).
- **1** enregistrement qui viole la contrainte "FK_SALES_PROD" (qui a donc un numéro de produit inexistant dans la table des produits).

Les données erronées n'ont pas été intégrées dans la table cible.

La mise en Production

Opération

Introduction à la mise en Production

La finalité de Oracle Data Integrator est d'automatiser les échanges d'informations entre applications.

Dans ce but il est nécessaire d'ordonner l'exécution de ces différents développements (interfaces, traitements spécifiques, ...) et de constituer un scénario de production contenant tout le code de chacun des développements réalisés et ainsi prêt à être exécuté.

Packages

Un package est un ensemble de travaux ordonnés, aussi dénommés steps, destinés à être exécutés dans un enchaînement pré-défini.

Il existe plusieurs types de steps :

- Interfaces
- Traitements spécifiques
- Actions/évaluations de variables
- Actions sur les modèles, sous-modèles ou datastores
- Commandes OS
- Outils Oracle Data Integrator

Dans le cadre de cette prise en main, vous avez besoin d'alimenter votre application "Sales Administration" en respectant un certain ordre dans l'enchaînement des interfaces. En effet, du fait de l'existence de contraintes référentielles entre les tables de cette application, vous ne pourrez pas par exemple alimenter la table TRG_CUSTOMER si la table TRG_CITY n'a pas préalablement été alimentée.

Dans ce qui suit, vous allez donc constituer et exécuter un **Package "Load Sales Administration"** à partir des interfaces que vous avez développées ainsi que celles pré-développées et disponibles dans le projet "Demo".

Scénario

Le scénario est destiné à la mise en production du package auquel il est rattaché. Le scénario est le résultat de la génération de code (SQL, shell ...) concernant les objets (interfaces, traitements spécifiques ...) contenus dans un package.

Note : Une fois généré, le scénario est stocké à l'intérieur du référentiel de travail. Le scénario peut être exporté puis importé dans un autre référentiel (distant ou non) pour être exploité dans un ou plusieurs contextes différents.

Note : Le code (langage généré) du scénario est figé, et toutes les modifications postérieures apportées au package et/ou aux modèles de données qui ont contribué à sa création ne le modifieront d'aucune manière.

Les propriétés du scénario sont visibles sur l'onglet **Définition** de la fenêtre **Scénario** du module **Designer** ou via le module **Operator**. L'exploitation d'un scénario s'effectue depuis la même fenêtre ou depuis une fenêtre de commande du système d'exploitation. Dans le module Designer, les scénarios sont regroupés sous leurs packages respectifs.

Dans le cadre de cette prise en main, vous êtes invités à générer un **Scénario** "**LOAD_SALES_ADMINISTRATION**" relatif au package précédemment constitué et à l'exécuter à partir de Oracle Data Integrator ou via une commande OS.

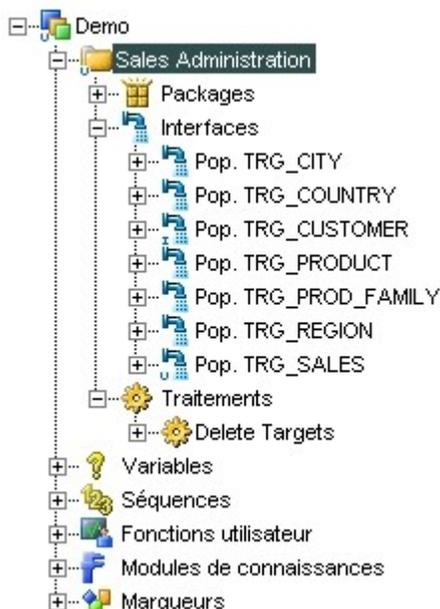
Package "Load Sales Administration"

Objectif

L'objectif de la constitution de ce package est de préparer la mise en production des développements relatifs à l'alimentation de l'application "Sales Administration" et de tester leur bonne exécution selon l'ordonnancement défini.

Les développements livrés avec Oracle Data Integrator

Un certain nombre de développements sont livrés dans le référentiel de démonstration, auxquels s'ajoutent les deux interfaces que vous avez développées :



Interfaces

- **Pop. TRG_COUNTRY** : interface permettant d'alimenter la table TRG_COUNTRY
- **Pop. TRG_REGION** : interface permettant d'alimenter la table TRG_REGION
- **Pop. TRG_CITY** : interface permettant d'alimenter la table TRG_CITY
- **Pop. TRG_PROD_FAMILY** : interface permettant d'alimenter la table TRG_PROD_FAMILY
- **Pop. TRG_PRODUCT** : interface permettant d'alimenter la table TRG_PRODUCT

Traitements

- **Delete Targets** : Traitement spécifique permettant de vider TOUTES les tables de l'application "Sales Administration". Le vidage de ces tables se fait

par simple "Delete" sur chacune des tables.

Traduction du problème

Afin de pouvoir alimenter correctement l'application de Sales Administration (soit en respectant les contraintes d'intégrité référentielles), il faut exécuter les tâches suivantes dans l'ordre suivant :

1. Vider les tables de "Sales Administration" grâce au traitement "Delete Targets"
2. Alimenter la table TRG_COUNTRY grâce à l'interface "Pop. TRG_COUNTRY"

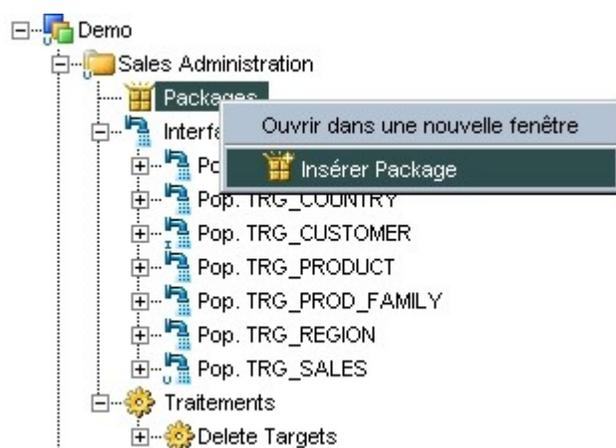
3. Alimenter la table TRG_REGION grâce à l'interface "Pop. TRG_REGION"
4. Alimenter la table TRG_CITY grâce à l'interface "Pop. TRG_CITY"
5. Alimenter la table TRG_PROD_FAMILY grâce à l'interface "Pop. TRG_PROD_FAMILY"
6. Alimenter la table TRG_PRODUCT grâce à l'interface "Pop. TRG_PRODUCT"
7. Alimenter la table TRG_CUSTOMER grâce à l'interface "Pop. TRG_CUSTOMER"
8. Alimenter la table TRG_SALES grâce à l'interface "Pop. TRG_SALES"

Un tel processus d'intégration est construit dans Oracle Data Integrator sous la forme d'un **Package**.

Les étapes à suivre

Création du Package

1. Faites un clic-droit sur l'objet **Package** et choisissez l'option **Insérer package**

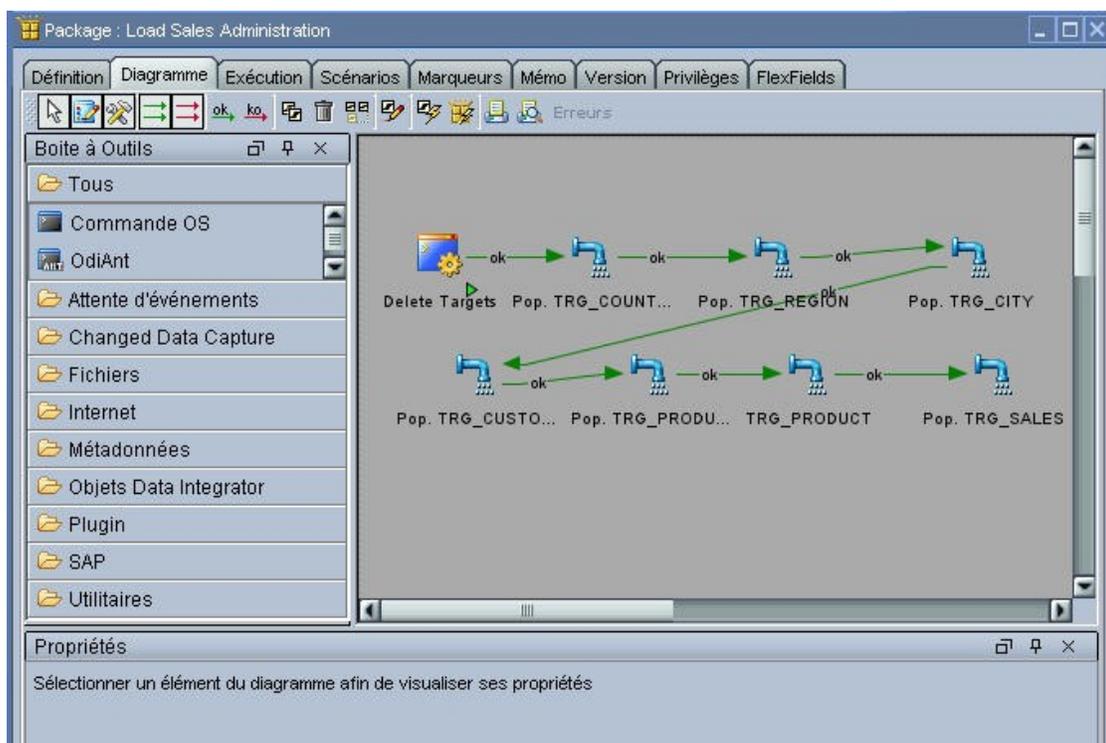


2. Dans la zone **Nom** de la fenêtre apparaissant à l'écran, saisissez "Load Sales Administration".
3. Cliquez sur le bouton **Appliquer** pour valider la création de ce package vide.

Définition des étapes du package

1. Cliquez sur l'onglet **Diagramme** de la fenêtre **Package**
2. Afin de créer les étapes du package, glissez les composants à intégrer au package depuis l'arborescence **Projets** dans le **Diagramme** du package que vous venez de créer. Les composants apparaissent sous forme d'étapes du package. Il ne sont pas ordonnancés ou liés.
3. Pour indiquer la première étape à exécuter (le traitement *Delete Target*), cliquez sur ce step, puis sélectionnez dans son menu contextuel **Étape de Départ**. Une icône apparaît sur ce step.
4. Cliquez sur l'icône **Étape suivante sur réussite**  dans la barre d'outils de l'onglet diagramme
5. Pour ordonnancer les étapes, cliquez une étape, puis sur celle qui la suit. Répétez cette opération pour ordonnancer toutes les étapes comme suit:

1. Delete Targets
 2. Pop. TRG_COUNTRY
 3. Pop. TRG_REGION
 4. Pop. TRG_CITY
 5. Pop. TRG_CUSTOMER
 6. Pop. TRG_PROD_FAMILY
 7. Pop. TRG_PRODUCT
 8. Pop. TRG_SALES
6. Le résultat auquel vous devez aboutir est le suivant :



7. Cliquez sur le bouton **Appliquer** pour valider ce package.

Exécution du package

1. Pour exécuter ce **package**, cliquez sur le bouton **Exécuter**.
2. Cliquez sur **OK** dans les fenêtres **Exécution** et **Information**.

Suivre l'exécution de l'interface dans Operator

En examinant l'exécution dans Operator, vous devriez retrouver les mêmes résultats que ceux obtenus lors de l'exécution de vos interfaces.

- ☰  Toutes les exécutions
 - ☰  12003 - Load Sales Administration - 2005-06-27 11:14:04.39
 - ☰  Variables
 - ☰  0 - Delete Targets - 2005-06-27 11:14:04.5
 - ☰  1 - Pop. TRG_COUNTRY - 2005-06-27 11:14:04.859
 - ☰  2 - Pop. TRG_REGION - 2005-06-27 11:14:05.531
 - ☰  3 - Pop. TRG_CITY - 2005-06-27 11:14:06.109
 - ☰  4 - Pop. TRG_CUSTOMER - 2005-06-27 11:14:06.75
 - ☰  5 - Pop. TRG_PROD_FAMILY - 2005-06-27 11:14:07.953
 - ☰  6 - Pop. TRG_PRODUCT - 2005-06-27 11:14:08.687
 - ☰  7 - Pop. TRG_SALES - 2005-06-27 11:14:09.453

Objectif

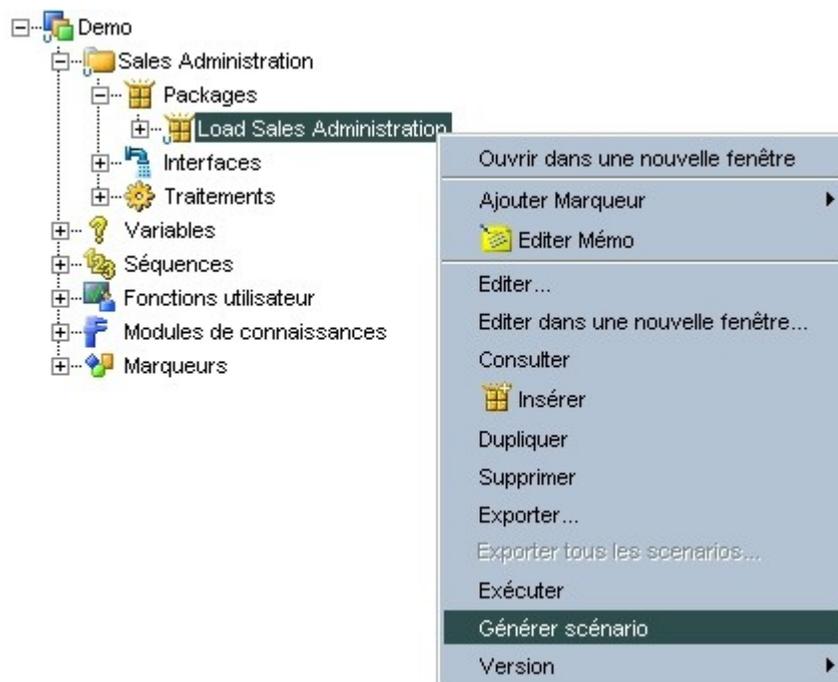
L'objectif de ce chapitre est de constituer un scénario "LOAD_SALES_ADMINISTRATION" destiné à la mise en production du package "Load Sales Administration".

La finalité étant de pouvoir exécuter de manière automatique ce scénario dans une configuration de production.

Les étapes à suivre

Création du Scénario

1. Faites un clic droit sur le nom du **package** "Load Sales Administration"
2. Dans le menu contextuel, choisissez l'option **Générer Scénario**



3. La fenêtre suivante apparaît à l'écran :



4. Cliquez sur le bouton **OK**. Oracle Data Integrator génère le scénario.
5. Le résultat obtenu est le suivant :



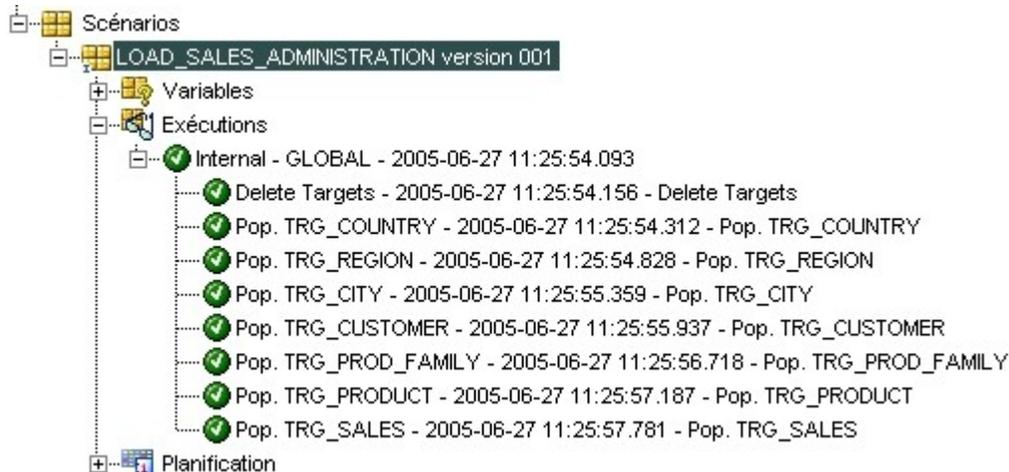
Exécution du Scénario à partir de Oracle Data Integrator

1. Faites un clic droit sur le nom **package** "LOAD_SALES_ADMINISTRATION"
2. Choisissez **Exécuter**
3. Cliquez sur **OK** dans les fenêtres **Exécution** et **Information**.

Suivre l'exécution du scénario

Vous pouvez suivre le résultat de votre exécution dans **Operator** et retrouver les mêmes résultats que ceux obtenus lors de l'exécution de votre package.

Il est également possible de consulter les comptes-rendus d'exécution de votre scénario à partir du module **Designer** en dépliant l'arborescence de votre scénario, comme ci-dessous:



Exécuter le scénario à partir d'une Commande OS

1. Lancez une invite de commande MS-DOS ou Shell unix
2. Positionnez-vous dans le sous-répertoire "bin" du répertoire d'installation de Oracle Data Integrator
3. Saisissez la commande suivante: `startscen LOAD_SALES_ADMINISTRATION 001 GLOBAL "-v=2"`

Note : Les paramètres de la commande `startscen` sont le nom du scénario, sa version, le contexte d'exécution, le niveau de trace. Ces paramètres sont séparés par des espaces.

1. A la fin de l'exécution, votre fenêtre de commande (sous Windows) devrait ressembler à celle-ci:

```
C:\oracledi\bin>startscen LOAD_SALES_ADMINISTRATION 001 GLOBAL "-v=2"

Oracle Data Integrator: Starting scenario
LOAD_SALES_ADMINISTRATION 001 in context GLOBAL ...
06/27/2005 11:33:05 AM(main): Creating session for
scenario:LOAD_SALES_ADMINISTRATION - 001
06/27/2005 11:33:06 AM(main): Session : 14003 is running
06/27/2005 11:33:16 AM(main): Session : 14003 finished with
return code : 0
DwgJv.main: Exit. Return code:0
```

Note: Dans ces exemples, nous utilisons une ligne de commande pour lancer un scénario. Ceci peut être fait de n'importe quelle application tierce. De la même manière, il est possible à Oracle Data Integrator d'interagir avec des application tierces et de les intégrer dans ses processus.

Conclusion

Vous avez maintenant terminé le projet ETL.

Au travers de ces exemples, vous avez pu :

- Mettre en place et réaliser un audit de l'intégrité des données pour l'application "Orders Application"
- Créer deux interfaces d'alimentation de l'application "Sales Administration" à partir des applications "Orders Application" et "Parameters (File)"
- Ordonnancer vos développements et préparer leur mise en production

Ces exemples correspondent à lune des problématiques auxquelles Oracle Data Integrator apporte une solution.

Que peut-on faire d'autre avec Oracle Data Integrator ?

Vous avez pu voir Oracle Data Integrator à l'oeuvre sur un projet data warehouse. Oracle Data Integrator est aussi capable d'adresser d'autres types d'intégration par les données, en batch ou au fil de l'eau. Exemple:

- Migration de données - avec ou sans mécanisme de réplication post-migration entre l'ancien et le nouveau système.
- Intégration de données point-à-point
- Réplication de données

Pour la démonstration, vous avez utilisé Oracle Data Integrator pour accéder aux données de bases de données et de fichiers. Oracle Data Integrator peut également accéder et intégrer toutes les base de données du marché, les systèmes ERP, CRM, mainframes, les fichiers plats, annuaires LDAP, fichiers XML,etc, avec la même facilité.

Oracle Data Integrator est seule plate-forme d'intégration qui unifie l'intégration orientée données, orientée événements et orientée services, au travers d'une interface utilisateur unique, et d'une approche par les règles déclarative communes. Elle permet à l'entreprise de présenter une vue unique de ses Systèmes d'Information, avec un modèle d'accès unifié.

Parmi les raisons d'utiliser Oracle Data Integrator :

- **Support pour une intégration unifiée** : Oracle Data Integrator est la seule solution d'intégration d'applications qui supporte l'intégration par les données, par les événements et par les services avec la même interface utilisateur. Cette fonctionnalité unique permet aux équipes informatiques de couvrir tous les besoins d'intégration (batch et temps réel, asynchrone et synchrone) indépendamment des volumes de données ou des exigences en terme de latence.
- **Productivité améliorée et courbe d'apprentissage raccourcie** : l'approche par les règles déclaratives est partagée dans tout Oracle Data Integrator, indépendamment de l'orientation données, événements ou services de chaque mécanisme d'intégration. Grâce un mode d'utilisation commun et une interface utilisateur partagée, la courbe d'apprentissage est raccourcie et la productivité accrue de manière considérable.
- **Meta-données partagées et réutilisables** : avec un référentiel de méta-données unique et intégré à tous les composants de Oracle Data Integrator, la consistance des processus

d'intégration est garantie. Le référentiel permet par ailleurs la réutilisation entre processus, des règles déclaratives, pour la transformation et la validation des données.

- **Support pour de multiples applications** : Oracle Data Integrator est adaptée pour une large gamme de projets d'intégration : ETL, Migration de données, Master data management, Business Activity Monitoring (BAM), Business Process Management (BPM), Business Process Reengineering (BPR), et intégration par les Services Web .

Aller plus loin avec Oracle Data Integrator

Pour vous familiariser un peu plus avec l'environnement d'Oracle Data Integrator, vous pouvez continuer à utiliser la configuration de démonstration.

Les documents suivants sont disponibles pour vous aider dans l'utilisation de Oracle Data Integrator:

Document	Description
Guide d'installation	Ce guide décrit les pré-requis et explique comment installer Oracle Data Integrator.
Guide de l'Utilisateur	Ce guide décrit les méthodes de travail avec Data Integrator.
Manuel de Référence	Ce manuel décrit en détail l'interface graphique de Oracle Data Integrator.

Contacteur Oracle

Si vous souhaitez de l'aide ou plus d'informations sur Oracle Data Integrator, n'hésitez pas à contacter Oracle.

www.oracle.com