
ETL プロジェクト・スタート・ガイド

このマニュアルについて

このドキュメントの目的は、Oracle Data Integrator の導入によって実現可能な機能の概要を説明することです。

このドキュメントでは、データの品質を検証しつつ異機種間システムのデータをデータ・ウェアハウスにロードするという ETL プロジェクトを、**Oracle Data Integrator** を使用してどのように実現するかを理解できるよう、いくつかの例を示しています。

このプロジェクトは、完了するまでに約 90 分かかります。プロジェクトの途中で作業内容を保存して中断し、再開することもできます。

- 「製品の概要」に進んでください。

製品の概要

Oracle Data Integrator とは

Oracle Data Integrator は、広く使用されているデータ統合ソフトウェア製品です。データの変換および統合プロセスを定義するための新しい宣言的な設計方法を提供しているため、開発およびメンテナンスを短時間で簡単に行えるようになります。Oracle Data Integrator は、固有の E-LT アーキテクチャに基づいて、データの変換および検証プロセスの実行に対して最大限のパフォーマンスを保証するだけでなく、現在入手可能な最も費用効率の高いソリューションでもあります。

Oracle Data Integrator は、データおよびアプリケーションの統合プロジェクトを合理化するための統一されたインフラストラクチャを提供します。

ビジネス上の問題

ますます加速する現代のビジネス環境では、より特化したソフトウェア・アプリケーションを使用することが必要とされます。また、異機種間のハードウェア・プラットフォームやシステムを共存させて、これらのアプリケーションやシステム間で確実にデータを共有できるようにすることが必要です。このような統合要件を実装するプロジェクトは、規格、時間および予算にあわせて作成する必要があります。

固有のソリューション

Oracle Data Integrator では、ETL に対して強力な宣言的設計方法を使用します。この方法では、宣言規則が実装の詳細から分離されます。また、Oracle Data Integrator は、固有の E-LT アーキテクチャに基づいているため、スタンドアロン ETL サーバーや独自のエンジンの必要性が排除され、使用する RDBMS エンジンの能力が利用されます。この組み合わせにより、開発とメンテナンスの両方の生産性が最大限に引き出され、データの変換および検証プロセスを実行する際に最高のパフォーマンスを得られます。

Oracle Data Integrator は、ETL を必要とする 500 以上の企業によって選ばれています。その主な理由は次のとおりです。

- **開発およびメンテナンスの時間短縮と簡略化:** ETL に対する宣言規則主導のアプローチにより、短期間で製品を習得できるため、開発者の生産性が劇的に向上します。また、継続的なメンテナンスが容易になります。この方法では、プロセスの定義が実際の実装から分離され、宣言規則（処理の対象）がデータ・フロー（処理の方法）から分離されます。
- **データ品質のファイアウォール:** Oracle Data Integrator では、ターゲット・アプリケーションに挿入される前に不正なデータが自動的に検出され、リサイクルされます。この処理は、ターゲット・アプリケーションと Oracle Data Integrator の両方で定義された整合性規則および制約に従って実行されます。プログラミングは不要です。
- **より高い実行パフォーマンス:** 従来の ETL ソフトウェアは、行ごとにデータ変換を実行する独自のエンジンに基づいているため、パフォーマンスが制限されます。既存の RDBMS エンジンおよび SQL に基づいて E-LT アーキテクチャを実装することで、セットベース・レベルでターゲット・サーバーでデータ変換を実行できるため、より高いパフォーマンスが得られます。
- **より単純で効率的なアーキテクチャ:** E-LT アーキテクチャでは、ソースとターゲット・サーバーの間に ETL ハブ・サーバーを置く必要がありません。ターゲット・サーバーおよびその RDBMS を活用して複雑な変換を実行します。ほとんどの変換は、エンドユーザーの問合せ処理でサーバーがビジーなとき以外に、バッチ・モードで実行されます。
- **プラットフォームの独立性:** Oracle Data Integrator は、同じソフトウェアを使用するすべてのプラットフォーム、ハードウェアおよび OS をサポートします。
- **データの接続性:** Oracle Data Integrator は、すべての RDBMS をサポートします。これには、Teradata、IBM DB2、Netezza、Oracle、Sybase IQ などのすべての主要なデータ・ウェアハウス・プラットフォーム、およびフラット・ファイル、ERP、LDAP、XML などのその他の多くのテクノロジーが含まれます。
- **経費削減:** ETL ハブ・サーバーおよび ETL エンジン排除することで、ハードウェアおよびソフトウェアの初期購入費用とメンテナンス費用の両方が削減されます。習得時間の短縮および開発者の生産性向上により、プロジェクト全体の人件費、および継続的な拡張の費用が削減されます。

技術アーキテクチャ

Oracle Data Integrator は、モジュール型リポジトリ・アーキテクチャを中心に構築されており、完全に Java で開発されたコンポーネント（グラフィカル・インタフェース・モジュールおよび実行エージェント）から、クライアント/サーバー・モードまたはシン・クライアント・モードでアクセスできます。

リポジトリは包括的でオープンな共有メタデータ・リポジトリです。リポジトリには、変換および統合のプロセスに関与したアクセス済アプリケーションのメタデータ、開発されたプロジェクトのバージョン、およびプロジェクトの使用に必要なすべての情報（計画、スケジュールおよび実行レポート）が格納されます。

トポロジ・マネージャは、グラフィカル・インタフェース・モジュールです。情報システムの物理アーキテクチャおよび論理アーキテクチャを説明するデータの管理に使用されます。サイト、マシンおよびデータ・サーバーの説明は、Oracle Data Integrator を使用して同じインタフェースを異なる環境で実行するために必要です。

デザイナーは、データ整合性のチェックを設計し、変換を構築するためのツールです。具体的には次の機能が含まれます。

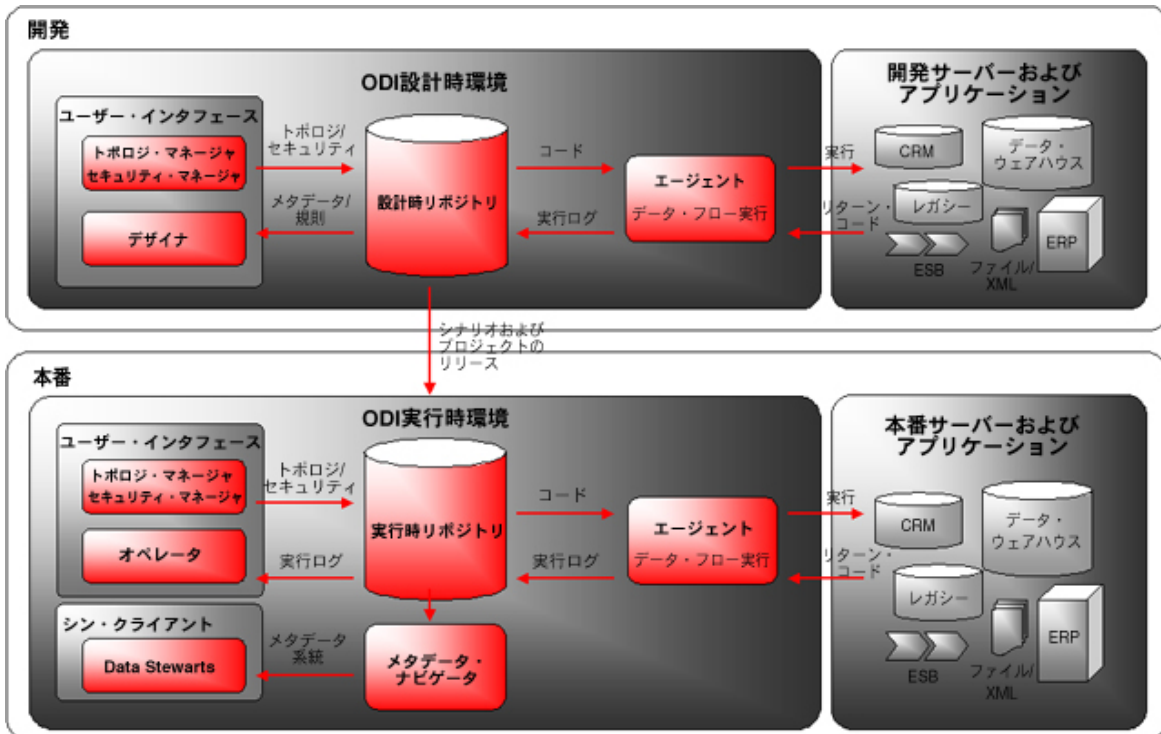
- 既存のアプリケーションまたはデータベースの自動リバース・エンジニアリング
- 変換および統合のインターフェースのグラフィカルな開発およびメンテナンス
- インターフェースでのデータフローの視覚化
- ドキュメントの自動生成
- 生成済コードのカスタマイズ

セキュリティ・マネージャは、Oracle Data Integrator でユーザーおよび権限を管理するためのツールです。

オペレータは、本番の管理および監視モジュールです。IT 本番オペレータ用に設計されています。

エージェントは、デザイナーで開発されたアプリケーション間インターフェースの実行に使用されます。**スケジューラ**を統合し、事前に定義されたスケジュールに従ってインターフェースを実行します。

メタデータ・ナビゲータは、ブラウザに基づくシン・クライアントで、Oracle Data Integrator のリポジトリへのアクセス、リポジトリのすべてのコンテンツの参照、検索および分析、および高度なデータシステムトレース、相互参照、影響分析などの実行に使用します。このツールは、すべての本番プロセスの管理にも使用できます。



Oracle Data Integrator のインストール

このドキュメントで説明する手順を行うには、Oracle Data Integrator がシステムに正しくインストールされていることが必要です。インストールの手順を次に示します。

前提条件

使用するワークステーションが次の要件を満たす必要があります。

- Windows または X Window 機能を持つ OS (Windows、Linux、HP-UX、Solaris など)
- 256MB または 512MB の RAM (推奨)
- 150MB のハード・ディスク
- TCP/IP

インストール

Oracle Data Integrator を CD からインストールする場合

Oracle Data Integrator のセットアップ CD を使用する場合は、次の手順に従います。

1. Oracle Data Integrator の CD-ROM を CD-ROM ドライブに挿入します。
2. CD の `/setup/<system>/` ディレクトリ (`<system>` はプラットフォーム名) に移動します。
3. Windows プラットフォームの場合は `setup.exe` を実行します。UNIX プラットフォームの場合は `sh ./setup.bin` を実行します。
4. セットアップ・プログラムの指示に従います。

Oracle Data Integrator をダウンロード済ファイルからインストールする場合

1. Windows プラットフォームの場合は `setup.exe` を実行します。UNIX プラットフォームの場合は `sh ./setup.bin` を実行します。
2. セットアップ・プログラムの指示に従います。

注意: Java 仮想マシンがセットアップ・プログラムに含まれているため、外部の JVM は不要です。Oracle Data Integrator のインストール・プログラムで提供される JVM を使用してください。

注意: Web サービスを使用するために、Oracle Data Integrator では Java Development Kit が必要になります。これは、インストール・プログラムでは提供されません。Oracle Data Integrator のコンポーネントを起動すると「Oracle Data Integrator で Web サービスを実行するには JDK が必要です。現在は JRE を使用しています。」という警告が表示される場合があります。この警告は無視してください。

注意: このセットアップ・プログラムでは、インストールした Oracle Data Integrator のコンポーネントのショートカットが作成されます。

注意: Microsoft Windows 95 または 98 を使用する場合は、オンライン・ヘルプの「Windows 95/98 での Oracle Data Integrator の実行」を参照してください。

注意: インストール中に、実行するインストールのタイプを選択できます。基本ファイルのみをインストールするには、「評価」インストール・セットを選択します。追加のユーティリテ

ィ (リポジトリ作成用のユーティリティなど) もインストールする場合は、「完全」インストール・セットを選択します。このスタート・ガイドの目的には、どちらのインストールでも使用できます。

プロジェクトの開始

Oracle Data Integrator および評価環境がマシンにインストールされました。これで、プロジェクトを開始できます。

プロジェクト

ETL プロジェクト

この ETL プロジェクトは、データの変換方法および使用する情報システムのデータの整合性をチェックする方法を理解するための例です。

このドキュメントに含まれる例

例の環境

このスタート・ガイドを通して、ある組織の例について学習します。この組織では、本番システムから発行された様々な異機種間のデータソースから取得される販売データを追跡しようとしています。使用されるデータソースは次のとおりです。

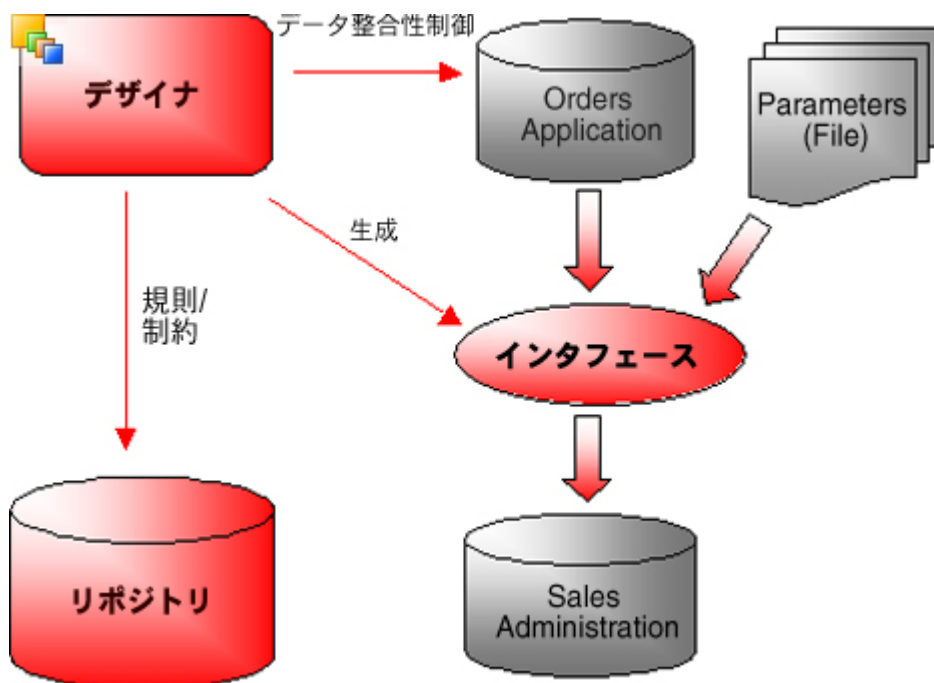
リポジトリ: リポジトリには、それぞれの例での演習に必要なすべてのメタデータが含まれています。リポジトリは、評価の目的上、指定されたデータベースでホストされます。

Orders Application: 顧客の注文を追跡するためのアプリケーション。指定されたデータベース (srcdemo サンプル・データベース) でホストされます。

Parameters (File): 本番システムから発行されたフラット・ファイル (ASCII)。販売員のリストおよび年齢層に区分された年齢が含まれます。

Sales Administration: 販売の管理または追跡。指定された別のデータベース (trgdemo サンプル・データベース) でホストされます。変換機能を使用して、このデータ・ウェアハウスを移入します。

次の図に環境を示します。



データ・モデル

次に示すスキーマ・ダイアグラムは、このスタート・ガイドを通して使用するデータ・モデルを定義しています。

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>ORDER_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

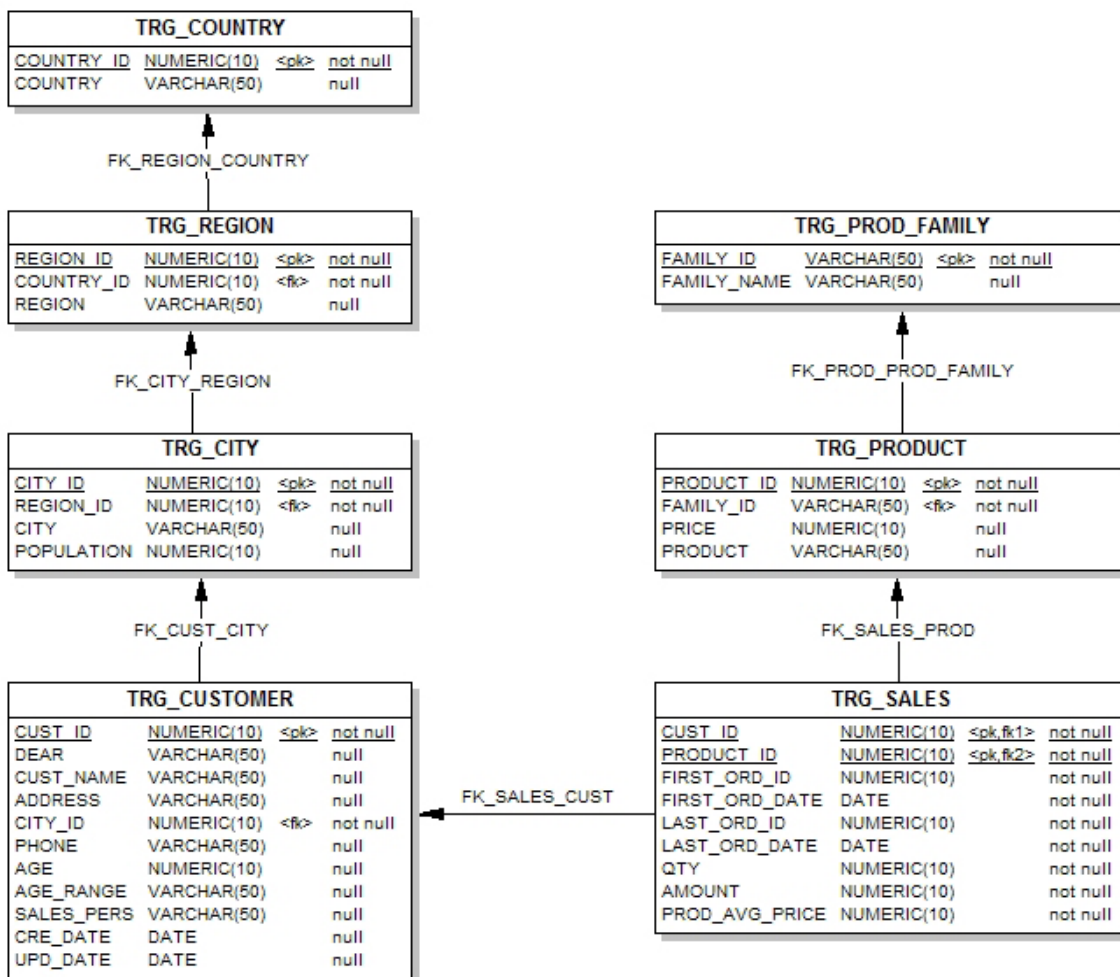
SRC_PRODUCT			
<u>PRODUCT_ID</u>	NUMERIC(10)	<pk>	not null
PRODUCT	VARCHAR(50)		null
PRICE	NUMERIC(10,2)		null
FAMILY_NAME	VARCHAR(50)		null

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



問題

すべてのデータ統合プロジェクトおよび変換プロジェクトに共通する課題は、次のとおりです。

- アプリケーション・データの品質のチェックおよび向上

- 情報システムのビジネス・ルールを考慮した、正確で簡単なアプリケーション間のデータ交換

これらの問題への対処方法を、次の各トピックの中で例を使用して説明します。

- データベースのデータのチェック
- ETL 変換の作成および実行
- 変換結果のクレンジング
- このプロセスの自動化

注意: このスタート・ガイドでは、ETL に関係するプロセスについて説明します。このドキュメントでは触れていませんが、リアルタイム EAI プロセスの作成も同じ方法で行えます。詳細は、このガイドの内容を最後まで学習した後で、Oracle Data Integrator のドキュメントを参照してください。

データのクレンジング

2つの例を実装することで、無効な行を分離しながらアプリケーションのデータの品質を確保するための Oracle Data Integrator の使用方法を学習します。Orders Application 表には、検出されるデータの不整合の数が含まれます。

- SRC_CUSTOMER 制御: 顧客のデータが情報システムで定義された制約に適合することをチェックします。

Sales Administration のロード

2つの単純な例を使用して、Orders Application アプリケーションおよび Parameters (File) アプリケーションのデータを Sales Administration データ・ウェアハウスにロードして生産性を高める方法を説明します。

- Pop. TRG_CUSTOMER インタフェース: ソース顧客データに、年齢層および販売員の情報を追加します。その後、このデータは、情報システムで定義された制約を考慮しながら、Sales Administration アプリケーションにロードされます。
- Pop. TRG_SALES インタフェース: 注文管理アプリケーションによって発行された販売データを要約し、販売追跡アプリケーションにロードします。

Oracle Data Integrator で開発したプロセスの自動化

Oracle Data Integrator のプロセスを自動化する方法を説明します。この演習の目的は、Sales Administration データ・ウェアハウス全体を1回のクリックでロードすることです。

- Load Sales_Administration パッケージ: インタフェースを順序付けしてパッケージ化します。
- LOAD_SALES_ADMINISTRATION シナリオ: このパッケージを自動的に実行するためのコードを生成します。

Oracle Data Integrator の起動

最初の実行

デモンストレーション環境の起動

Oracle Data Integrator で提供されるデータベースには、次の各例で使用するデータおよび初期リポジトリが含まれています。これらの例を参照する前に、**まず、次の手順を行ってください。**

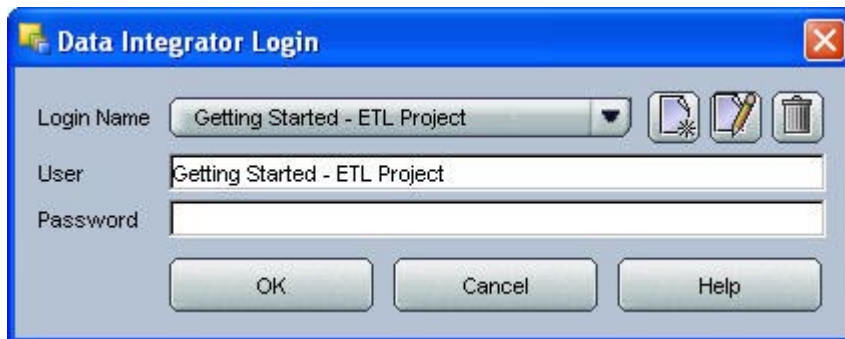
「スタート」→「プログラム」→「Oracle Data Integrator」→「Examples」の順に選択し、「Start Demo Environment」を選択します。リポジトリ・データベース・サーバーおよびソースとターゲットのデータベースが起動します。

これらのデータベース・サーバーを停止するには、同じフォルダ内の **Stop Demo Environment** ショートカットを使用できます。**[Ctrl]を押しながら[C]を押すキーボード・ショートカットを使用したり、実行ウィンドウを閉じてデータベースを停止しないでください。** データベースのデータが破損する可能性があります。

UNIX 環境では、コマンド・プロンプトを使用して Oracle Data Integrator の install/bin ディレクトリにある適切な.sh ファイルを実行できます。その際には、「&」を追加して、サーバーをバックグラウンド・タスクとして起動します（たとえば、リポジトリ・データベースを起動する場合は startdemo.sh&と入力し、停止する場合は stopdemo.sh と入力します）。

デザイナの起動

1. 「スタート」→「プログラム」→「Oracle Data Integrator」→「Designer」の順に選択します。次のウィンドウが表示され、接続先のリポジトリを選択するよう求められます。

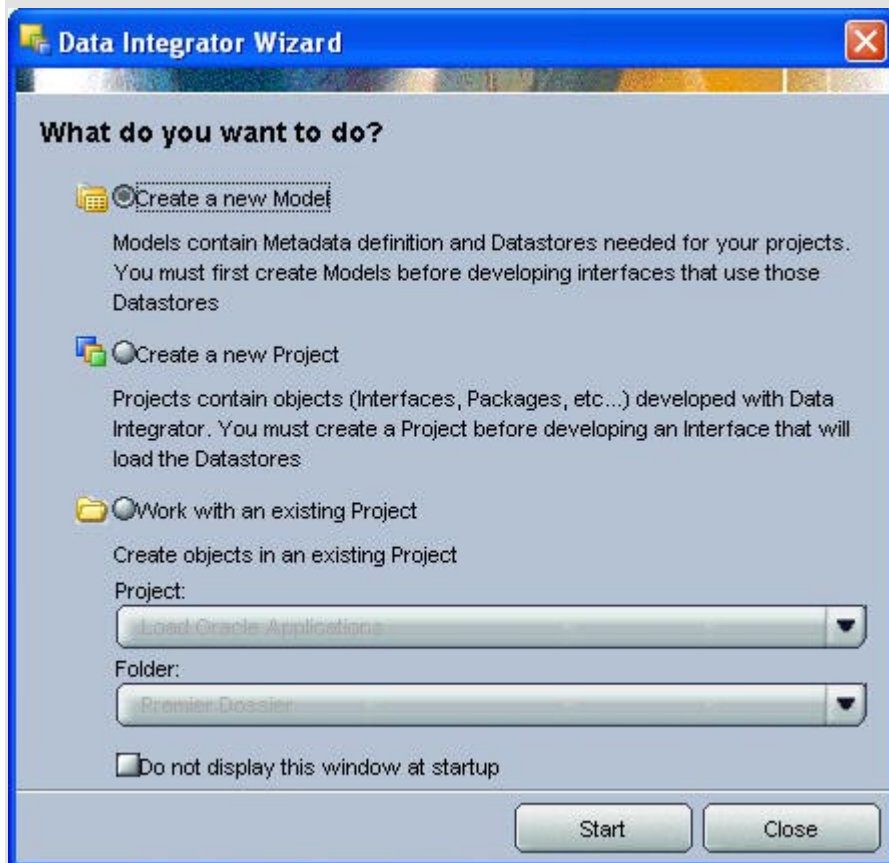


2. 「ログイン名」ドロップダウン・メニューから適切なログインを選択します。「Getting Started - ETL Project」を選択します。
3. 「OK」をクリックしてリポジトリに接続します。

注意: Orders Application アプリケーションおよび Sales Administration アプリケーションのデータベース・サーバーに加えて、前項の説明に従ってリポジトリ・サーバーが起動されたことを確認してください。

4. デザイナが起動します。

注意: デザイナが開くとウィザード・ウィンドウが表示される場合があります。このウィンドウは、「クローズ」をクリックして閉じてかまいません。

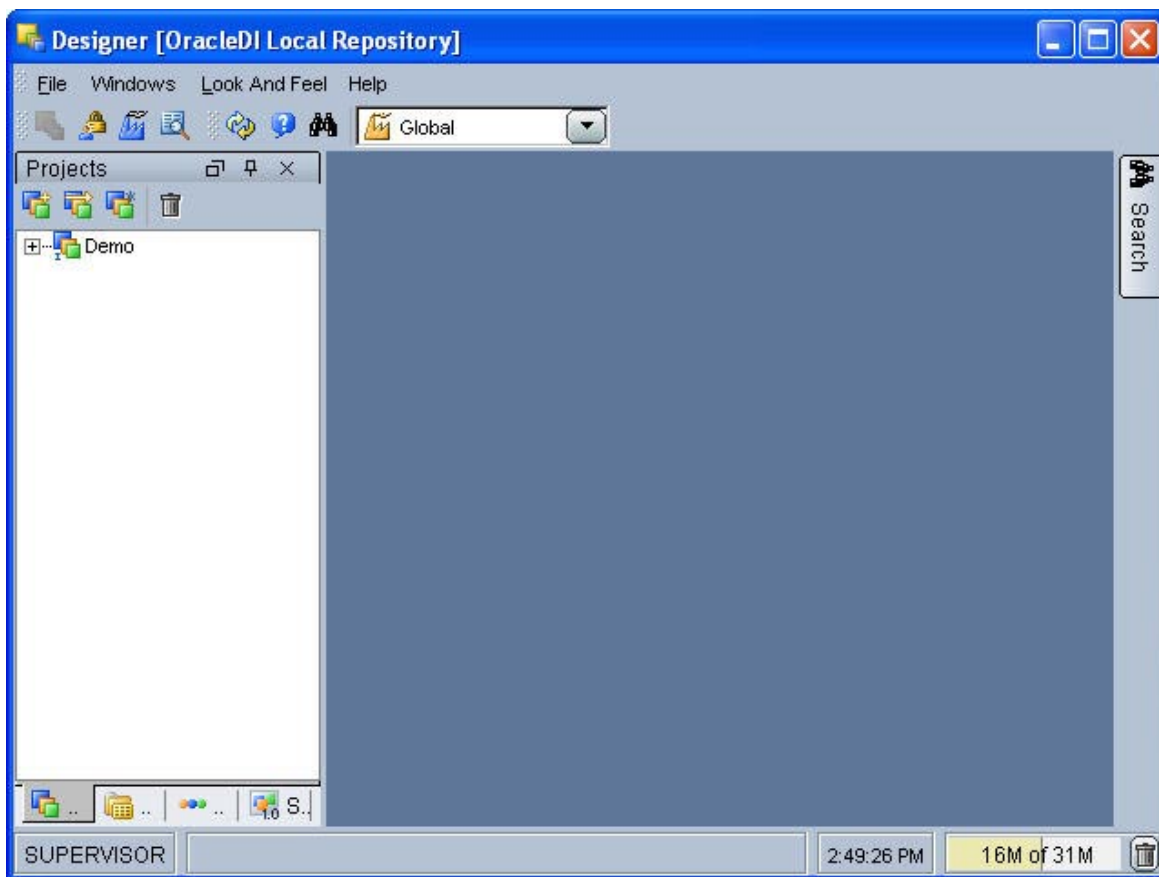


デザイナのインタフェース

デザイナで管理されるものは次のとおりです。

- アプリケーションの**データ・モデル**: データ・サーバーに含まれるすべてのメタデータ（表、列、制約、説明、相互参照など）
- **プロジェクト**: データ・サーバーのロード規則および変換規則（インタフェース、プロシージャ、変数など）

デザイナの GUI は、次のように表示されます。



メニュー

メニューには、次の機能にアクセスするためのプルダウン・メニューが含まれます。

- インポート/エクスポート
- ウィザード
- 表示オプション
- モジュールまたはツリー表示のオープン
- ユーザーのパスワードおよびオプションの変更

ツールバー

ツールバーでは、次の操作が可能です。

- Oracle Data Integrator モジュールのオープン。
- ツリー表示のリフレッシュ。
- オンライン・ヘルプのオープン。
- デフォルト・コンテキストの選択。このプロジェクトで使用するコンテキストは、Global です。

ツリー表示

現行ユーザーが使用できるデザイナー・オブジェクトは、編成されて「プロジェクト」、「モデル」、「ソリューション」および「その他」（ユーザー・ファンクション、グローバル変数および順序）というツリー表示になります。

オブジェクトに対するほとんどのアクションは、（オブジェクトを選択して右クリックすると表示される）ポップアップ・メニューから実行します。

それぞれのツリー表示は、フローティング可能なフレームの中に表示されますが、デザイナーのメイン・ウィンドウの横にドッキングできます。これらのフレームはスタックすることも可能です。複数のフレームをスタックした場合、フレーム・ウィンドウの最下部にタブが表示され、スタックの各フレームにアクセスできるようになります。

ツリー表示フレームの移動、ドッキングおよびスタックは、フレームのタイトルまたはタブを選択し、ドラッグして行います。デザイナーのメイン・ウィンドウにツリー表示フレームが表示されない場合やフレームが閉じた場合は、「ウィンドウ」→「ビューの表示」メニューを使用して開くことができます。

オブジェクトは、すでにデモ・リポジトリに存在します。

- 「モデル」表示では、Orders Application、Parameters および Sales Administration の各アプリケーションに対応するすべてのデータ・モデルが表示されます。
- 「プロジェクト」表示では、Demo プロジェクトおよび Sales Administration フォルダが表示されます。フォルダには、すでに複数のインタフェースが含まれています。このフォルダ内で、新しいインタフェースを開発します。

ワークベンチ

編集または表示するオブジェクトのウィンドウは、ワークベンチに表示されます。

データ整合性制御

データ統合の概要

IT インフラストラクチャで複数のアプリケーションを使用し、異なるデータストア間で様々な形式での読取りや書込みを行う場合、企業内の誰もがデータを簡単に使用できるようにデータを統合するプロセスを実装することが不可欠になります。この目的に使用できるデータ統合の方法は、ETL、データのレプリケーション、データの同期など多数あります。いずれの方法でも、使用する情報システムのデータと組織のアプリケーションを正常に統合するための第一歩は、データの整合性を確保することです。

データ整合性制御の概要

データ整合性の制御は、使用する情報システムのアプリケーションに含まれるすべてのデータの一貫性を確保するために必要です。

アプリケーション・データは、情報システムによって課せられる制約および宣言規則に対して有効でない場合があります。たとえば、顧客が指定されていない注文や商品が指定されていない注文明細行などが検出される可能性があります。

Oracle Data Integrator が提供する作業環境では、制約違反を検出し、リサイクルまたはレポート作成のために格納できます。

制御には、**静的制御**および**フロー制御**の2種類があります。これらの2つの違いを説明します。

静的制御

静的制御は、アプリケーション・データの整合性を検証するための規則が存在することを意味します。これらの規則（制約とも呼ぶ）の一部は、（主キー、参照制約などを使用して）データ・サーバーですでに実装されている可能性があります。

Oracle Data Integrator では、追加の制約を宣言せずに直接サーバーで定義することで、データの検証を調整できます。このプロシージャは**静的制約**と呼ばれます。このプロシージャによって、既存のデータ（つまり静的データ）に対するチェックを直接実行することが可能になるためです。

フロー制御

変換プロセスおよび統合プロセスのターゲットとなる情報システムでは、多くの場合、独自の宣言規則が実装されます。**フロー制御**関数は、アプリケーションが受け取ったデータをこれらのターゲットにロードする前に、制約に従って検証するために使用します。フロー制御プロシージャの詳細は、インタフェースに関する章で説明しています。

利点

データ整合性チェックを実行する主な利点は次のとおりです。

- ターゲット・データベースをライフ・サイクル全体で使用することで、生産性が向上します。データのビジネス・ルール違反によって、ターゲット・データベースのライフ・サイクル全体でアプリケーション・プログラミングの速度は低下します。そのため、転送されたデータをクリーニングすると、アプリケーション・プログラミングの時間を短縮できます。
- ターゲット・データベースのモデルが検証されます。検出される規則違反は、常にソース・データの整合性が不十分であることを意味するわけではありません。ターゲット・モデルの一定の不備を示す可能性もあります。アプリケーションが再書込みされる前にデータを移行すると、現実に即したテスト・データベースを提供しながら、新しいデータ・モデルを検証できます。
- ビジネス・ルール違反をフィルタ処理で排除するために前処理されたデータを使用することで、エンドユーザーへのサービスの質が向上します。

データの整合性の確保は、複雑な作業を伴う場合があります。実際には、宣言規則に違反するすべてのデータを分離してリサイクルする必要があります。つまり、複雑なプログラミングの開発が必要になります（特に、ターゲット・データベースに整合性制約検証のメカニズムが含まれている場合）。操作制約に関しては、（ソース、ターゲットまたはリサイクルされたフローの）誤ったデータを修正する方法を実装し、その方法を企業全体で再利用することが最も効率的です。

例

次に示す例では、データ整合性監査プロセス（静的制御）の流れを説明します。

Orders Application - HSQL アプリケーションには、様々なレベルでビジネス・ルール制約を満たさないデータが含まれています。ここでの目的は、このアプリケーションに含まれるデータのうち、情報システムによって課せられた制約を満たさないデータを検出することです。

SRC_CUSTOMER 制御

目的

ソースのデータには、一貫性のないものが含まれている可能性があります。ターゲット表の一部の制約がソース表内で実装されていない場合、または規則の追加が必要な場合があります。ここでは、次に示す2つの制約があるとします。

- **顧客の年齢は21歳より上であること。**ただし、入力表には21歳以下の顧客に関するレコードが含まれている可能性があります。
 - **CITY_ID 列は、SRC_CITY 表の1つのエントリを参照すること。**ただし、市区町村の表に存在しない値がある可能性があります。

ここでの目的は、これらの2つの制約を満たさない行を検出し、対応する無効なレコードを、分析のためにエラー表に自動でコピーすることです。

問題の解釈

このような規則を強制するには、SRC_CITY 表と SRC_CUSTOMER 表の間の参照制約とともに、チェック制約（条件とも呼ぶ）を使用する必要があります。

Oracle Data Integrator で条件を追加するには、満たす必要がある規則を単純に指定します。参照の追加は、一致する必要がある列を定義するだけで行えます。次に、2つの制約を作成します。

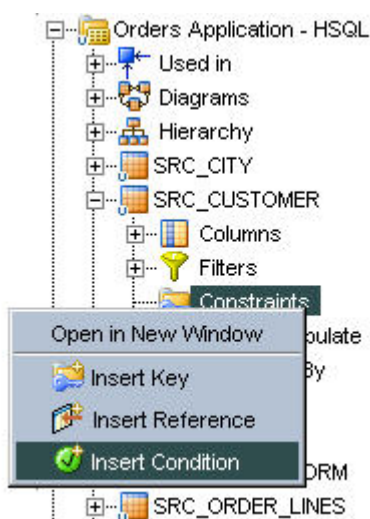
制約の作成

年齢制約

この例では、列にデータ検証条件を追加します。

1. Orders Application - HSQL モデルを開きます。
2. SRC_CUSTOMER 表を開きます。

3. **Constraints** ノードを右クリックして「条件の挿入」を選択します。

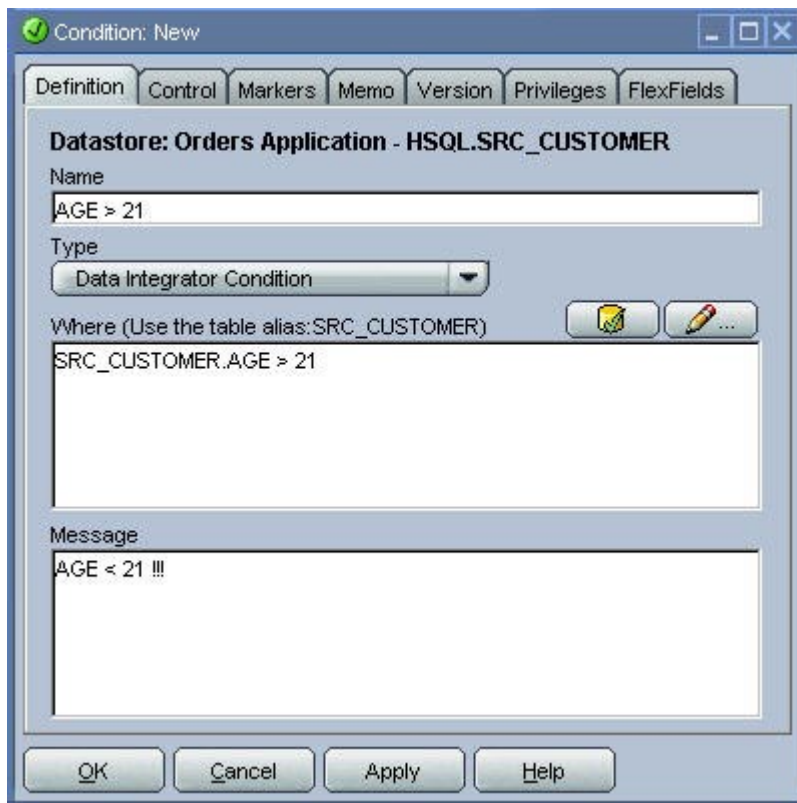


4. 「定義」タブで次の処理を行います。
- 「名前」に条件の名前を入力します ("AGE > 21"など)。
 - 「タイプ」で、条件の種類として「Oracle Data Integrator 条件」を選択します。
 - 「条件」句に、SRC_CUSTOMER.AGE > 21 のSQL コードを入力します。
このテキストは、直接または (✎ ボタンをクリックしてアクセスする) 式エディタを使用して入力できます。

注意: Data Integrator が正しい SQL コードを生成するには、データストア（ここでは SRC_CUSTOMER）の別名を AGE の接頭辞として使用する必要があります。この別名は、「データストア」画面（データストア上で右クリック→「編集」）の「定義」タブで定義されます。

注意: Oracle Data Integrator によって作成された制約は、実際にデータベース上には作成されません。これらの制約はリポジトリに格納されます。

- 「メッセージ」フィールドで、エラー表に表示されるエラー・メッセージを指定します。
- 画面の表示は次のようになります。



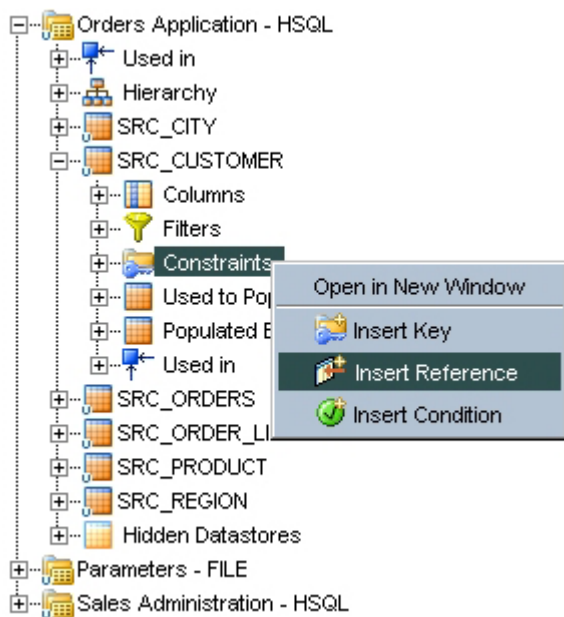
5. 「OK」をクリックして条件を保存します。

参照制約

この例では、SRC_CUSTOMER 表が CITY_ID 列を使用して SRC_CITY を参照するという参照制約を追加します。

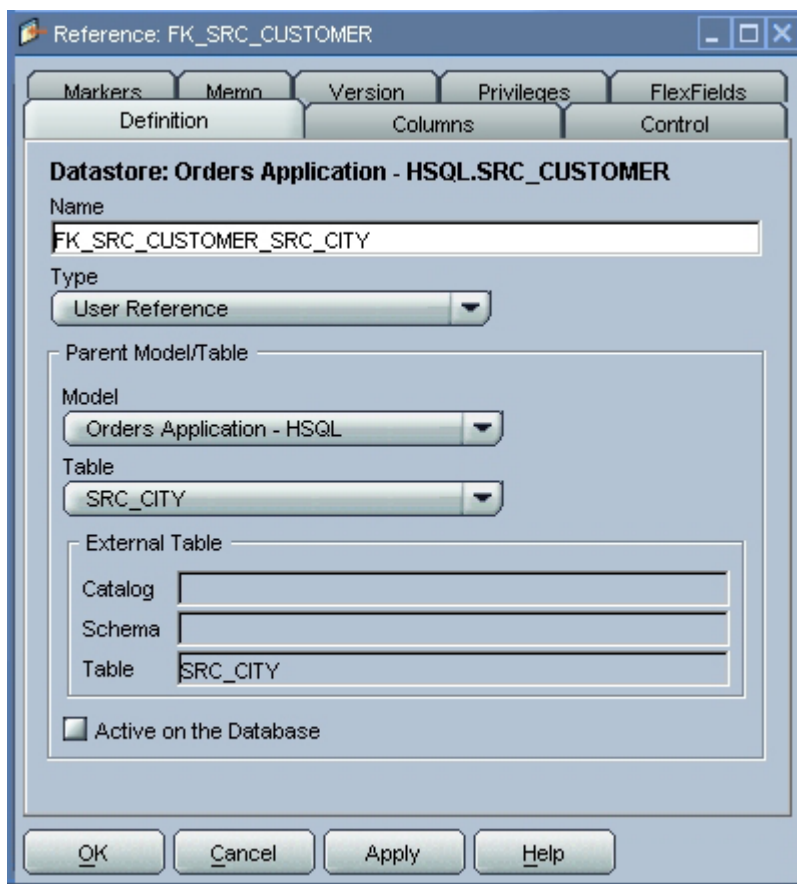
まず、有効な CITY_ID 値をチェックする制約を作成します。作成手順は次のとおりです。


1. Orders Application - HSQL モデルのツリー構造を開きます。
2. SRC_CUSTOMER データストアのツリー構造を開きます。
3. **Constraints** ノードを右クリックして「参照の挿入」を選択します。

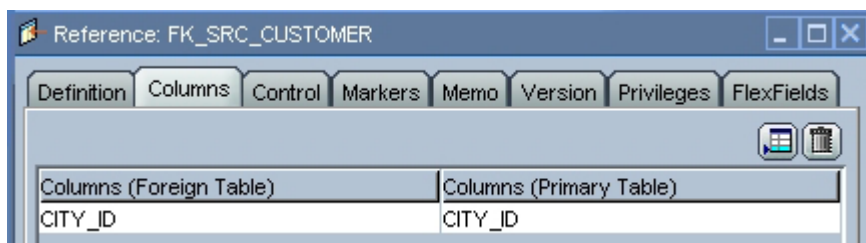


4. 「定義」タブで次の処理を行います。

- 「タイプ」で、この参照の種類として「ユーザー参照」を選択します。
- リンク先となる表に対応するリスト・ボックスで SRC_CITY を選択します。
- 画面の表示は次のようになります。



5. 「列」タブで次の処理を行います。
 -  をクリックして一致する列を定義します。
 - 表示される行をクリックします。適切な表のすべての列が含まれるドロップダウン・リストが表示されます。
 - 「外部表」 (SRC_CUSTOMER) および「プライマリ表」 (SRC_CITY) から CITY_ID を選択します。

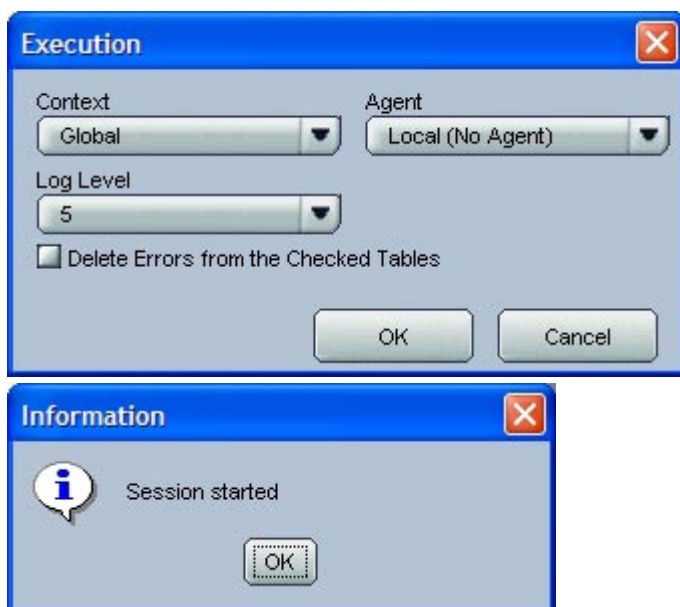


6. 「OK」をクリックしてこの参照を保存します。

静的制御の実行



1. SRC_CUSTOMER データストアを右クリックします。
2. 「制御」 → 「チェック」の順に選択します。

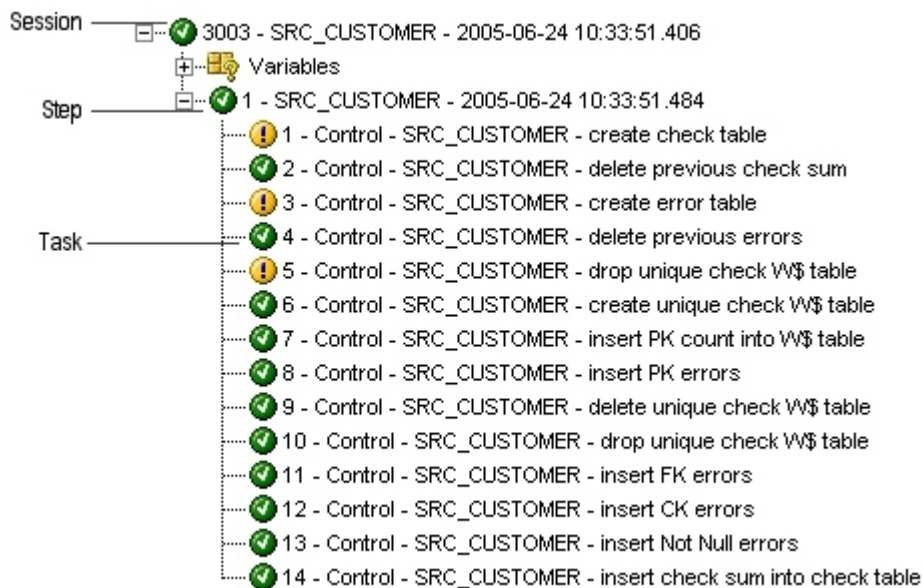
3. 実行ウィンドウおよび情報ウィンドウで「OK」をクリックします。



Oracle Data Integrator により、データ・チェックおよび実行セッションの開始に必要なすべてのコードが自動的に生成されます。

オペレータでの制御の実行の追跡

1. オペレータで実行のステップを確認するには、メニュー・バーの  「オペレータ」 ボタンをクリックします。
 2. オペレータ・ウィンドウが開き、**All Executions** ノードの下にある処理タスクの詳細が、オペレータの「セッション・リスト」表示に表示されます。
 3. このウィンドウがすでに開いている場合は、 ボタンを使用してウィンドウをリフレッシュできます。
- 1 つの実行セッションのログは、次のように表示されます。



ログは、次の3つのレベルで構成されます。

- セッション
- ステップ (チェック対象のデータストア、インタフェースまたはプロシージャに対応)
- タスク (インタフェース、プロセスまたはチェックの要素的タスクに対応)

結果の解釈


無効なレコードの数

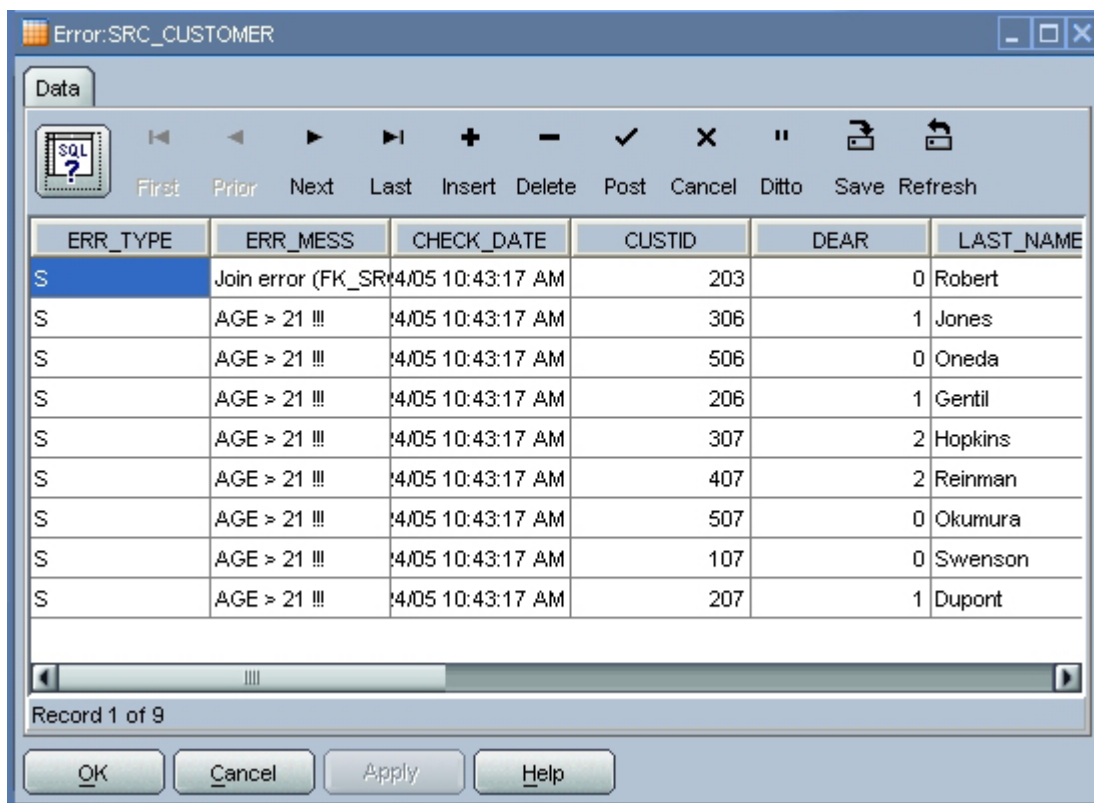
1. **オペレータ**で実行ツリーの対応するアイコンをダブルクリックし、SRC_CUSTOMER のステップ (レベル 2) ウィンドウを編集します。画面の右に新しいウィンドウが開きます。
2. 「**実行**」タブを選択します。SRC_CUSTOMER 表のチェックにより、9 個の無効なレコードが検出されたことが示されています。これらのレコードは、エラー表に分離されています。

Nb of Inserts	0	Nb of Updates	0
Nb of Deletes	0	Nb of Errors	9

無効なレコード

無効なレコードにアクセスするには、モデル内の表を右クリックし、ポップアップ・メニューから「**制御**」→「**エラー**」の順に選択します。

1. デザイナ・モジュールに戻るには、ツールバーの「**デザイナ**」 ボタンをクリックします。
2. Orders Application - HSQL モデルを開きます。
3. SRC_CUSTOMER データストアを右クリックします。
4. ポップアップ・メニューから「**制御**」→「**エラー**」の順に選択します。
5. 次のようなウィンドウが表示され、データ内のエラーが一覧表示されます。



このウィンドウを使用して、チェック処理で無効化されたレコードを確認できます。

- 8 個のレコードが AGE > 21 制約に違反しています（つまり、実際の年齢が 21 歳以下です）。
- 1 個のレコードが FK_CITY_CUSTOMER 制約に違反しています（つまり、CITY_ID 値が SRC_CITY 表に存在しません）。

このウィンドウでは、レコード全体を確認できます。そのため、不正な値（つまり、最上部のレコードの無効な CITY_ID 値）を一目で確認できます。

表示されるエラー・メッセージは、AGE > 21 制約の設定時に定義されたものです。

これで、静的制御の設定が完了しました。次に行うのは、変換インタフェースの実装です。

データ統合用インタフェース

データ統合用インタフェース

Pop. TRG_CUSTOMER インタフェース

目的

最初のインタフェースの目的は、Orders Application - HSQL モデルの SRC_CUSTOMER 表から Sales Administration - HSQL モデルの TRG_CUSTOMER ターゲット表に、データをロードすることです。

ただし、SRC_CUSTOMER 表には、この操作に必要なすべてのデータが含まれていません。ここでは、ターゲット表に次の情報を追加します。

- SRC_AGE_GROUP フラット・ファイルで定義されている、ソース表の AGE 属性に対応する年齢層 (AGE_RANGE)。
- SRC_SALES_PERSON ファイルで定義されている、ソース表の販売員番号 (SALES_PERS_ID) に対応する顧客販売員の氏名 (LAST_NAME および FIRST_NAME)。
- ソース表の DEAR 列の数値データ (0、1、2) からターゲットの標準の敬称テキスト文字列 (Mr、Mrs または Ms) に変換された値。
- ソース顧客の名前および姓を連結したもの。

ソース・データは、ターゲット環境で実装されている整合性規則と一致しない場合があります。この変換では、すべての制約が満たされていることを検証し、無効な行をターゲット・データベースではなくエラー表に格納することで、データをクレンジングします。この場合、次の 2 つの重要な整合性規則を満たす必要があります。

- 顧客の年齢が 21 歳より上であること (条件 AGE > 21)
- 顧客が、TRG_CITY 表に存在する市区町村 (CITY_ID) に関連付けられていること (参照 FK_CUST_CITY)

これらの規則の機能別詳細および手順を次に示します。

インタフェースの定義

この変換に使用するデータの概要は次のとおりです。

ターゲット・データストア

モデル: Sales Administration - HSQL

データストア: TRG_CUSTOMER

ソース・データストア

モデル	データストア	説明	種類
Orders Application - HSQL	SRC_CUSTOMER	ソース・システム の顧客	HSQL 表
Parameters - FILE	SRC_AGE_GROUP	年齢層ファイル	セミコロン区切りのファイル
Parameters - FILE	SRC_SALES_PERSON	販売員ファイル	固定サイズ・レコードのファイル

結合

結合	説明	SQL 規則	実行場所
販売員と顧客	SRC_SALES_PERSON と SRC_CUSTOMER の結合	SRC_CUSTOMER.SALES_PERS_ID = SRC_SALES_PERSON.SALES_PERS_ID	ステージング領域
顧客と年齢層	顧客の年齢がファイル 内の最低年齢から最高 年齢までの間であること が必要	SRC_CUSTOMER.AGE between SRC_AGE_GROUP.AGE_MIN and SRC_AGE_GROUP.AGE_MAX	ステージング領域

変換規則

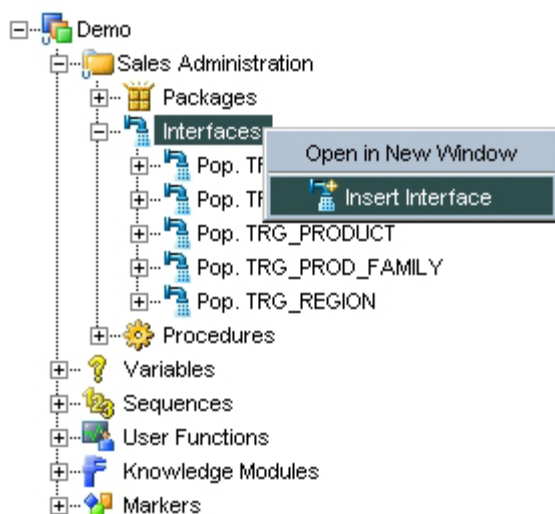
ターゲット列	変換元	SQL 規則	実行場所
CUST_ID	SRC_CUSTOMER.CUSTID	SRC_CUSTOMER.CUSTID	ソース
DEAR	SRC_CUSTOMER.DEAR = 0 の場合は MR SRC_CUSTOMER.DEAR = 1 の場合は MRS、それ以外の場合は MS	CASEWHEN (SRC_CUSTOMER.DEAR = 0, 'MR', CASEWHEN (SRC_CUSTOMER.DEAR = 1, 'MRS', 'MS'))	ソース
CUST_NAME	SRC_CUSTOMER.FIRST_NAME と SRC_CUSTOMER.LAST_NAME (大文字) の連結	SRC_CUSTOMER.FIRST_NAME ' ' UCASE (SRC_CUSTOMER.LAST NAME)	ソース
ADDRESS	SRC_CUSTOMER.ADDRESS	SRC_CUSTOMER.ADDRESS	ソース
CITY_ID	SRC_CUSTOMER.CITY_ID	SRC_CUSTOMER.CITY_ID	ソース
PHONE	SRC_CUSTOMER.PHONE	SRC_CUSTOMER.PHONE	ソース
AGE	SRC_CUSTOMER.AGE	SRC_CUSTOMER.AGE	ソース
AGE_RANGE	SRC_AGE_GROUP.AGE_RANGE	SRC_AGE_GROUP.AGE_RANGE	ステージング領域

	GE_RANGE		域
SALES_PERS	SRC_SALES_PERSON. FIRST_NAME と SRC_SALES_PERSON. LAST_NAME (大文 字) の連結	SRC SALES PERSON.FIRST NAME ' ' UCASE (SRC SALES PERSON.LAST _NAME)	ステージング領 域
CRE_DATE	今日の日付	CURDATE ()	ターゲット
UPD_DATE	今日の日付	CURDATE ()	ターゲット

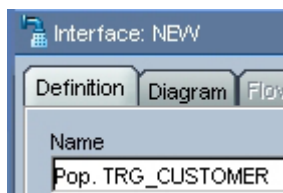
手順

新しいインタフェースの挿入

1. Demo プロジェクトの Sales Administration フォルダで Interfaces を右クリックし、「インタフェースの挿入」を選択します。



2. インタフェース・ウィンドウの「名前」フィールドに、インタフェースの名前 (Pop. TRG_CUSTOMER) を入力します。



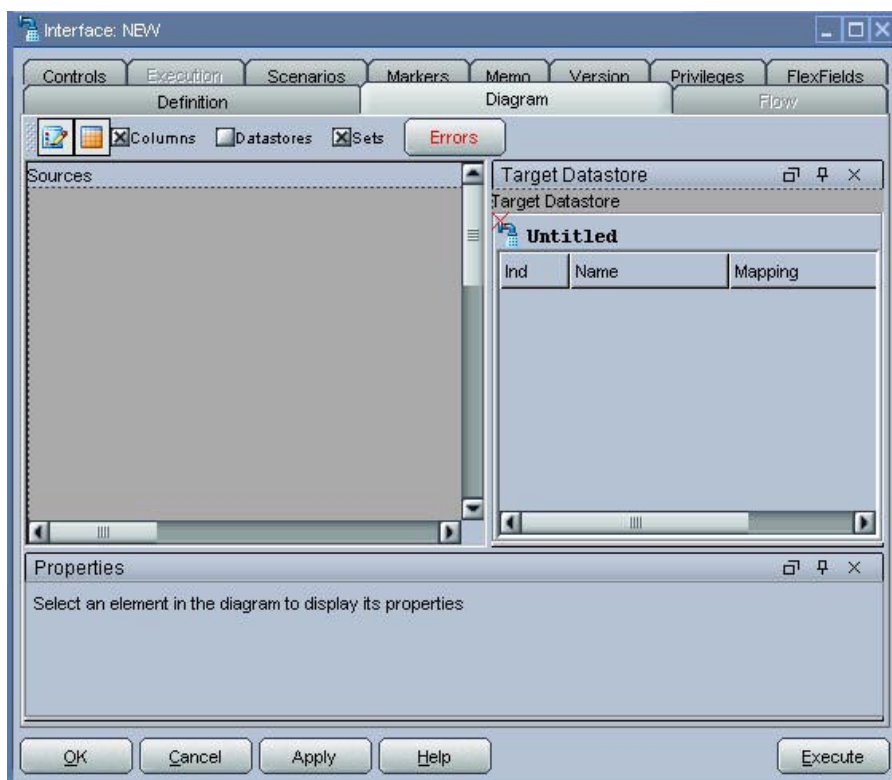
3. ここで、インタフェース・ウィンドウを大きくして見やすくします。

ターゲット表の定義

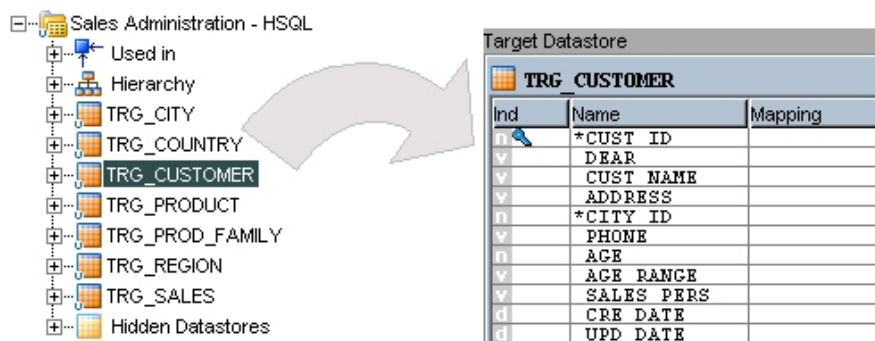
1. 作業環境を入力するために、「**ダイアグラム**」タブをクリックします。情報ボックスが表示されます。単に「**OK**」をクリックしてこのボックスを閉じてかまいません。



2. インタフェースの「**ダイアグラム**」タブの表示は、次のようになります。



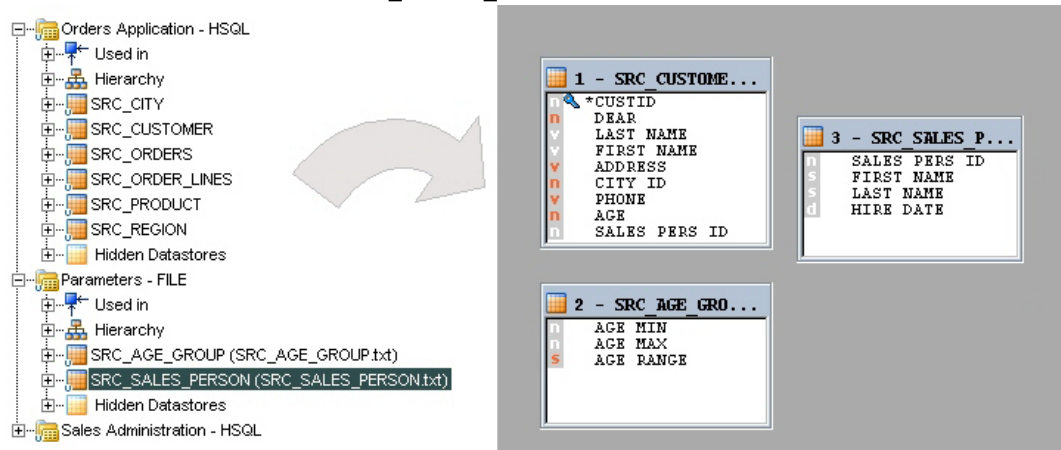
3. Sales Administration - HSQL モデルの TRG_CUSTOMER 表を、「**ダイアグラム**」タブの右にある「**ターゲット・データストア**」領域にドラッグします。



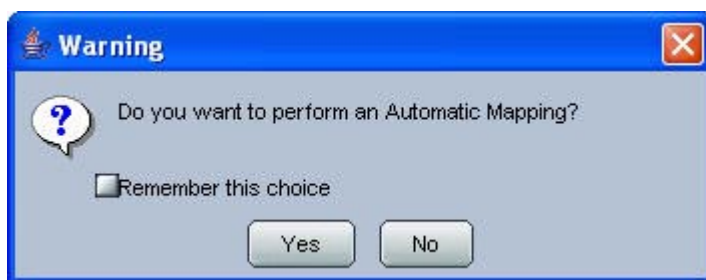
ソース・データストアの定義

1. 「ダイアグラム」タブで、構成パネル（「ダイアグラム」タブの左部分）に次のソース・データストアをドラッグします。

- Orders Application - HSQL モデルの SRC_CUSTOMER
- Parameters - FILE モデルの SRC_AGE_GROUP
- Parameters - FILE モデルの SRC_SALES_PERSON

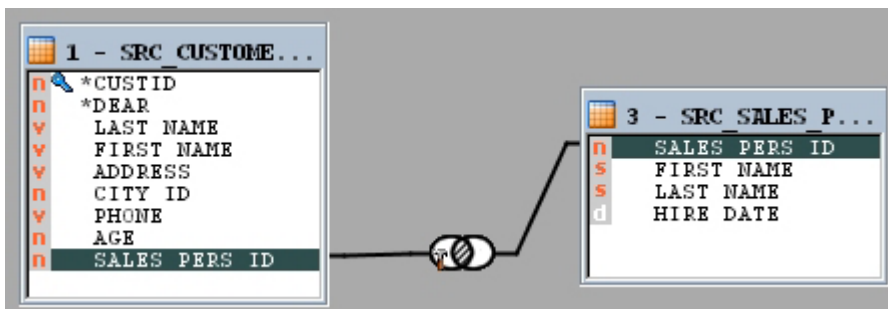


2. 構成パネルにデータストア・ソースをドロップすると、Oracle Data Integrator によって、フィールド対フィールドの自動マッピングの使用を確認するポップアップ・ウィンドウが表示されます。「はい」をクリックしてこのウィンドウを閉じます。

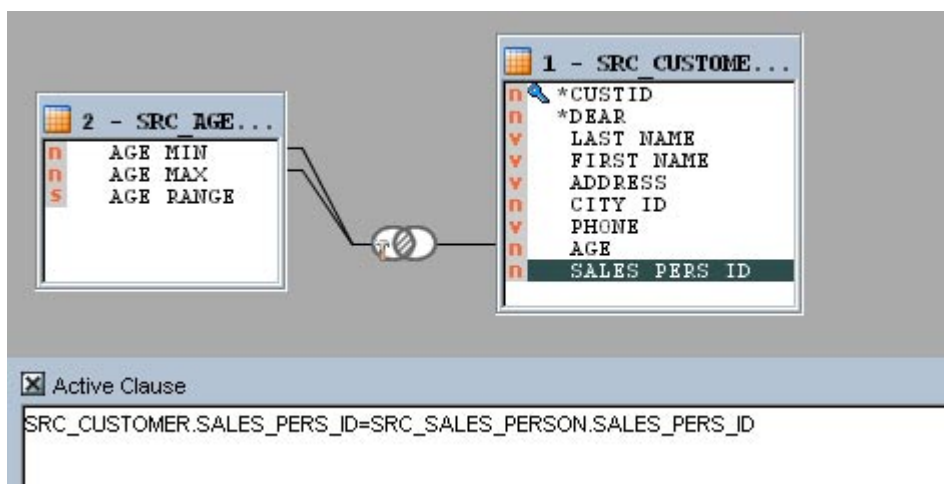


ソース・データストア間の結合の定義

1. 構成パネルで、SRC_CUSTOMER 表の SALES_PERS_ID 列を SRC_SALES_PERSON 表の SALES_PERS_ID 列にドラッグします。販売員識別子に最初の結合が定義されます。



2. SRC_CUSTOMER 表の AGE 列を SRC_AGE_GROUP 表の AGE_MIN 列にドラッグします。
3. 「プロパティ」パネルで**実装**を次のように変更します。
 - =を **between** に置き換えます。
 - テキストの最後に **and** を追加します。
 - SRC_AGE_GROUP 表の AGE_MAX 列をテキスト・フィールドにドラッグします。
4. 変更後のテキストは、SRC_CUSTOMER.AGE between SRC_AGE_GROUP.AGE_MIN and SRC_AGE_GROUP.AGE_MAX になります。これは、SRC_CUSTOMER と SRC_AGE_GROUP の結合に対応します。




注意: ソース・データストアをリンクするために、参照がモデルですでに定義されている場合は、これらの参照が結合としてソース・ダイアグラムに自動的に表示されます。

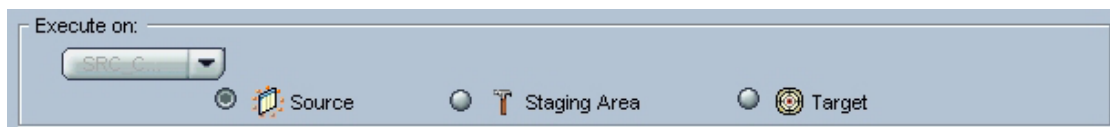
変換規則の定義

「ダイアグラム」の「ターゲット・データストア」領域で、ソースと名前が一致する列が（列名マッチングによって）自動的にマップされます。この場合の必要な処理は、CUSTID、DEAR、



CUST_NAME、SALES_PERS、CRE_DATE および UPD_DATE の各フィールドに対して変換規則を定義することのみです。

変換規則は、各ターゲット列で次のように定義します。

1. ターゲット列の名前をクリックして、ロードするターゲット列を選択します。
2. 左下の「実装」フィールドで規則を作成します。これは、ソース表から必要な列をテキスト・フィールドにドラッグするか、 ボタンをクリックして式エディタを開いて行います。
3. 必ず、変換を実行する環境を選択してください。この環境を選択するには、次に示すラジオ・ボタンの1つを選択します。

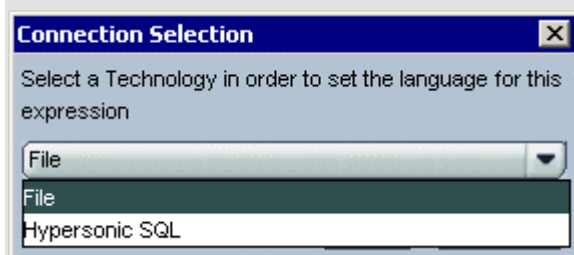


マッピングで次の規則を実装します。

- **CUST_ID** (ソースで実行) : SRC_CUSTOMER.CUSTID 列を「実装」フィールドにドラッグします。これにより、ソース表の CUST_ID 列の名前が、単純にターゲット表にマップされます。
- **DEAR** (ソースで実行) : CASEWHEN (SRC_CUSTOMER.DEAR=0, 'MR', CASEWHEN (SRC_CUSTOMER.DEAR=1, 'MRS', 'MS')) というテキストを入力します。SRC_CUSTOMER.DEAR 列は、必要ときにその都度ドラッグできます。 を使用して式を保存し、 を使用して構文を検証します。これにより、ソース表の DEAR 列が、文字列式 (0>'MR', 1>'MRS', 2>'MS') としてマップされます。
- **CUST_NAME** (ソースで実行) : 式エディタを開いて、SRC_CUSTOMER.FIRST_NAME || ' ' || UCASE (SRC_CUSTOMER.LAST_NAME) という規則を作成します。これにより、各顧客の名前および姓の連結値 (姓は大文字) がマップされます。

注意: 式の作成に使用する式エディタには、1つのテクノロジーに固有のすべての関数が含まれているわけではありません。多くのテクノロジーに共通する関数のみが含まれます。式エディタに表示されない関数でも、手動で入力し、式に使用することが可能です。

注意: 式エディタを開く際に、次のウィンドウが表示される場合があります。



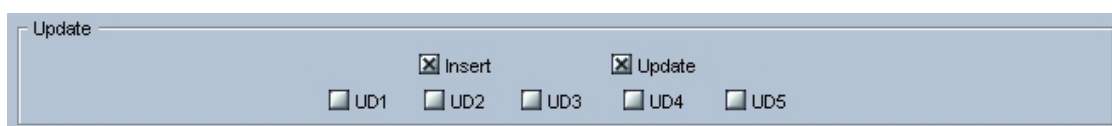
Hypersonic SQL を選択して「OK」をクリックします。このウィンドウでは、式エディタで SQL 言語構文を使用することを選択できます。


注意: このマッピングをターゲットで実行する場合は、式エディタでターゲット・システムの構文 (および、この場合は Hypersonic SQL) が提供されます。

- **SALES_PERS** (ステージング領域で実行) : 式エディタを開いて
`SRC_SALES_PERSON.FIRST_NAME || ' ' ||`
`UCASE(SRC_SALES_PERSON.LAST_NAME)` という規則を作成します。
 これにより、各販売員の名前と姓の連結値 (姓は大文字) がマップされます。

注意: この規則は、必ずステージング領域で実行してください。この例のソースはフラット・ファイルで、それ自体は連結をサポートするエンジンに関連付けられていません。

- **CRE_DATE** および **UPD_DATE** (両方ともターゲットで実行) : `CURDATE()` 関数を入力し、「アクティブ・マッピング」ボックスを選択してターゲットで式を実行することを選択します。**CRE_DATE** については、「挿入」でのみマッピングを実行します。これを行うには、マッピング領域の最下部にスクロールして「更新」を選択解除します。**UPD_DATE** については、レコードが更新された日付のみが必要です。そのため、**UPD_DATE** マッピングの「挿入」チェック・ボックスを選択解除します。




注意:  ボタンをクリックすると、入力済 SQL コードの構文をチェックするために、ソース・サーバーがコールされます。このチェックは、ソース・サーバーで規則全体が実行される時のみ実行可能です。エラーが発生すると、エラー・ウィンドウが表示されます。「詳細」ボタンをクリックすると、ソース・サーバーから送信されたエラー・メッセージが表示されます。

この時点で、変換規則は次の画面のようになります。

Ind	Name	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()

左側のアイコンは、ターゲット列のデータ型の頭文字を示します (n: NUMERIC、v: VARCHAR、d: DATE)。その他のアイコンは、式の実行場所を示します。

	ソース
	ステージング領域
	ターゲット

注意: マッピングの「インジケータ」インジケータによって  アイコンが表示された場合は、マッピングでエラーが発生しています。その場合は、不正なターゲット・フィールドを選択して入力を検証します。もしくは、「ダイアグラム」タブの最上部にある「エラー」ボタンをク

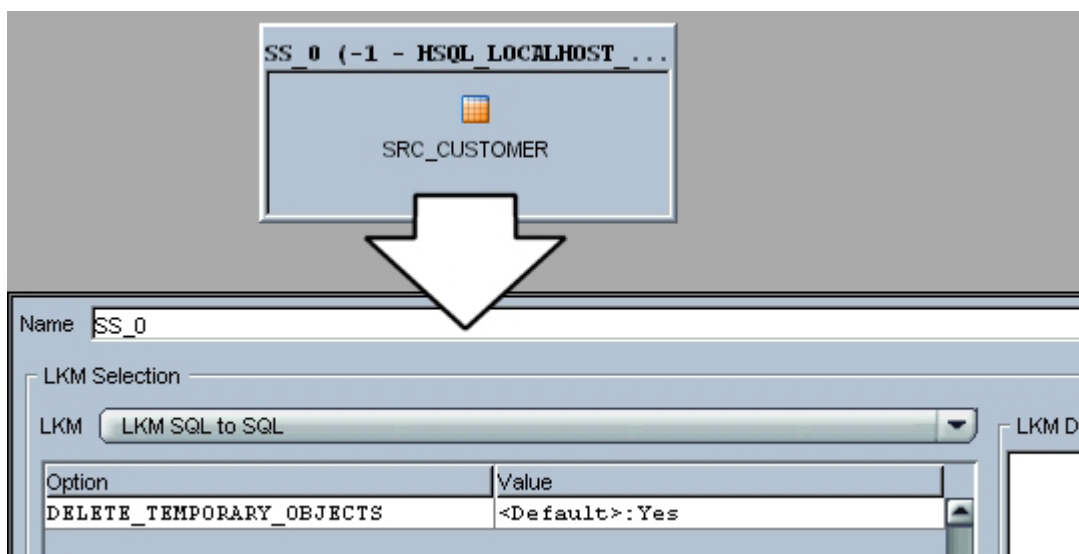
リックし、表示されるウィンドウの「詳細」ボタンをクリックして、エラーの種類の詳細情報を参照します。

データ・ロード戦略 (LKM) の選択

「フロー」タブには、Oracle Data Integrator により、インタフェースの実行に必要な様々な手順が示されます。

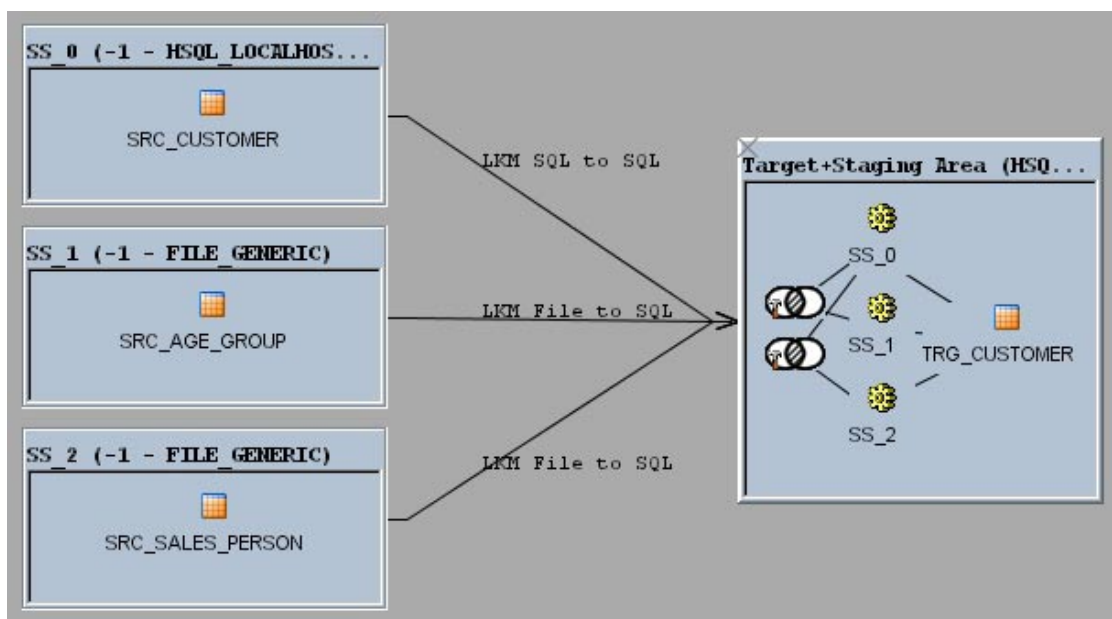
ソース環境の SRC_AGE_GROUP ファイル、SRC_SALES_PERSON ファイルおよび SRC_CUSTOMER 表からデータを取得する方法 (ロード・ナレッジ・モジュール - LKM) を選択する必要があります。

1. SRC_AGE_GROUP ファイルのロードに対応するソース・セット (SS_x) をクリックして、ロード・ナレッジ・モジュールのリストから「LKM File to SQL」モジュールを選択します。



2. SRC_SALES_PERSON ファイルのロードに対応する 2 つ目のソース・セット (SS_x) をクリックして、ロード・ナレッジ・モジュールのリストから「LKM File to SQL」モジュールを選択します。
3. SRC_CUSTOMER 表のロードに対応する最後のソース・セット (SS_x) をクリックして、ロード・ナレッジ・モジュールのリストから「LKM SQL to SQL」モジュールを選択します。

4. この時点で、「フロー」画面の表示は次のようになります。



データ統合戦略 (IKM) の選択

ロード・フェーズが完了したら、データをターゲット表に統合するための戦略を選択する必要があります。これを行うには、使用する統合ナレッジ・モジュール (IKM) を指定する必要があります。

1. 「ターゲット+ステーjing領域」オブジェクトをクリックして「IKM SQL Incremental Update」ナレッジ・モジュールを選択します。

Distinct Rows

IKM Selection

IKM

Option	Value
CREATE_TARG_TABLE	<Default>:No
TRUNCATE	<Default>:No
DELETE_ALL	<Default>:No
SYNC_JRN_DELETE	<Default>:Yes
INSERT	<Default>:Yes
UPDATE	<Default>:Yes
FLOW_CONTROL	<Default>:Yes
RECYCLE_ERRORS	<Default>:No
STATIC_CONTROL	<Default>:No
COMMIT	<Default>:Yes
DELETE_TEMPORARY_OBJECTS	<Default>:Yes

2. ナレッジ・モジュールの「オプション」では、デフォルト値をそのまま使用します。デフォルト値は上に示すとおりです。

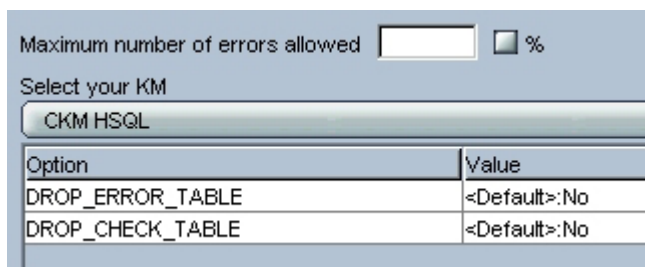
これらのオプションを使用すると、統合戦略の特定の側面を制御できます。たとえば、FLOW_CONTROL オプションを使用することで、ターゲット表に挿入される前のデータのフロー制御操作が、Oracle Data Integrator によってトリガーされます。

注意: ナレッジ・モジュールを選択する際に使用できるモジュールは、少数しかありません。Oracle Data Integrator に含まれるすべてのナレッジ・モジュールを使用するには、それらのモジュールをプロジェクトにインポートする必要があります。ただし、この手順はここでの目的には必要ありません。独自のプロジェクトについてこの手順を行う場合の詳細は、製品ドキュメントのオンライン・ヘルプに含まれる Oracle Data Integrator ユーザーズ・ガイドを参照してください。

データ制御戦略の選択

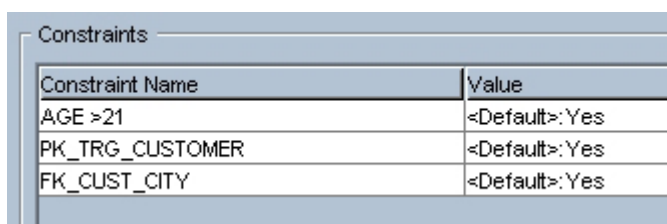
ここまでの処理では、ソースからターゲットへのデータ・フローのみが指定されました。ここでは、データのチェック方法 (CKM) を定義して、データを統合する前に満たす必要がある制約および規則を定義する必要があります。

1. 「制御」タブで、「CKM HSQL」チェック・ナレッジ・モジュールを選択します。



Option	Value
DROP_ERROR_TABLE	<Default>:No
DROP_CHECK_TABLE	<Default>:No

2. 次に示すように、検証する制約 (AGE > 21、PK_TRG_CITY および FK_CUST_CITY) を Yes に設定します。



Constraint Name	Value
AGE > 21	<Default>:Yes
PK_TRG_CUSTOMER	<Default>:Yes
FK_CUST_CITY	<Default>:Yes



インタフェースの実行準備が完了しました。

インタフェースの実行

1. インタフェース・ウィンドウの「実行」ボタンをクリックしてインタフェースを実行します。
2. インタフェースの保存を求めるウィンドウが表示されます。「はい」をクリックします。
3. 実行ウィンドウおよび情報ウィンドウで「OK」をクリックします。

Oracle Data Integrator によって自動的に SQL コードが生成され、実行セッションが開始されます。

オペレータでのインタフェースの実行の追跡

1. オペレータで実行の結果を追跡するには、メニュー・バーの  アイコンをクリックします。
2. オペレータ・ウィンドウが開き、実行処理タスクの詳細が表示されます（このウィンドウがすでに開いている場合は、 ボタンをクリックしてリフレッシュできます）。

結果の解釈

処理されたレコード

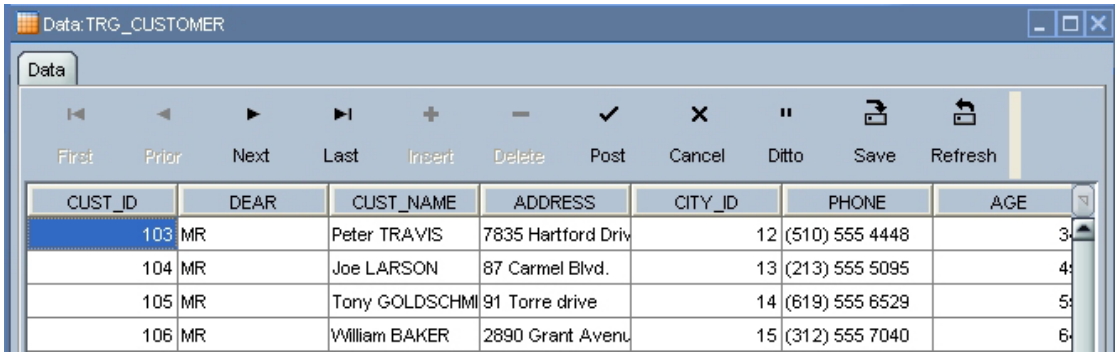
1. 「オペレータ」で実行ツリーの対応するアイコンをダブルクリックし、Pop. TRG_CUSTOMER ステップのウィンドウを開きます。画面の右に新しいウィンドウが開きます。
2. 「実行」タブに、TRG_CUSTOMER 表のロードによって 25 個の挿入が作成され、9 個のエラーがエラー表に分離されたことが示されます。

Nb of Inserts	25	Nb of Updates	0
Nb of Deletes	0	Nb of Errors	9

作成されたデータ

インタフェースによって作成されたデータは、**デザイナ**・モジュールで確認できます。

1. Sales Administration - HSQL モデルを開いて TRG_CUSTOMER 表を選択します。
2. 右クリックして「**データ**」を選択し、ターゲット表のデータを表示します。



CUST_ID	DEAR	CUST_NAME	ADDRESS	CITY_ID	PHONE	AGE
103	MR	Peter TRAVIS	7835 Hartford Drive	12	(510) 555 4448	3
104	MR	Joe LARSON	87 Carmel Blvd.	13	(213) 555 5095	4
105	MR	Tony GOLDSCHMIDT	91 Torre drive	14	(619) 555 6529	5
106	MR	William BAKER	2890 Grant Avenue	15	(312) 555 7040	6

無効なレコードおよび不正なデータ

実行したインタフェースでは、9 個の無効なレコードが識別され、自動作成されたエラー表に分離されました。無効なレコードにアクセスするには、モデルからターゲット表

(TRG_CUSTOMER) を選択し、ポップアップ・メニューから「**制御**」→「**エラー**」の順に選択します。

ERR_TYPE	ERR_MESS	CHECK_DATE	CUST_ID	DEAR	CUST_NAME
F	Join error (FK_CU	/24/05 2:11:03 PM	207	MRS	Marie-Chantale
F	Join error (FK_CU	/24/05 2:11:03 PM	203	MR	Christian ROBE
F	Age < 21 !!!	/24/05 2:11:03 PM	306	MRS	Mary JONES
F	Age < 21 !!!	/24/05 2:11:03 PM	506	MR	Kenji ONEDA
F	Age < 21 !!!	/24/05 2:11:03 PM	206	MRS	Michele GENTIL
F	Age < 21 !!!	/24/05 2:11:03 PM	407	MS	Heineke REINM,
F	Age < 21 !!!	/24/05 2:11:03 PM	507	MR	Isao OKUMURA
F	Age < 21 !!!	/24/05 2:11:03 PM	107	MR	Jack SWENSON
F	Age < 21 !!!	/24/05 2:11:03 PM	207	MRS	Marie-Chantale

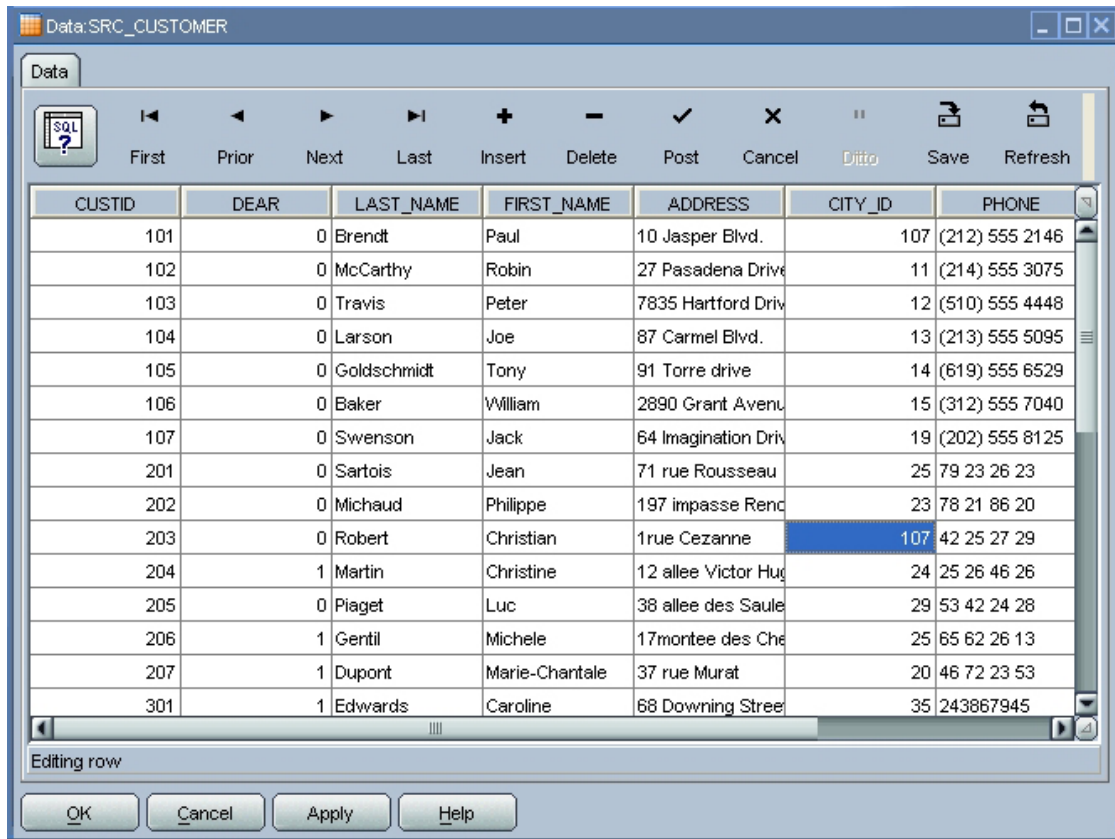
この画面から、次のレコードがインタフェースによって拒否されたことがわかります。

- FK_CUST_CITY 制約を満たしていない（つまり、市区町村の表に存在しない市区町村番号が設定されている）2 個のレコード。
- 21 歳以下の顧客に関するビジネス・ルール（AGE > 21 制約）を満たしていない 7 個のレコード。

無効なレコードはエラー表に保存され、ターゲット表には統合されません。


無効なデータの修正

1. Orders Application - HSQL モデルのツリー構造を開きます。
2. SRC_CUSTOMER を右クリックして「データ」を選択します。
3. CUSTID が 203 であるクライアント行を画面内で探します。列ヘッダーをクリックすると表をソートできます。顧客 203 が表示されない場合は、「リフレッシュ」ボタンをクリックして表示をリフレッシュします。
4. この顧客の「CITY_ID」列の値をダブルクリックして、変更します。新しい値は（208 ではなく）107 にする必要があります。



5. 「適用」をクリックして「OK」をクリックします。
6. コンテキスト・メニューの「実行」オプションを選択して、Pop. TRG_CUSTOMER インタフェースを再実行します。

処理されたレコード

1. 「オペレータ」でをクリックして画面をリフレッシュします。
2. 実行ツリーの対応するアイコンをダブルクリックし、Pop. TRG_CUSTOMER ステップのウィンドウを編集します。画面の右側に新しいウィンドウが開きます。
3. 「実行」タブには、TRG_CUSTOMER 表のロードによって1個の挿入（変更したばかりのレコード）が作成され、8個の無効なレコードがエラー表に分離されたことが示されます。

Nb of Inserts	1	Nb of Updates	0
Nb of Deletes	0	Nb of Errors	8

データ統合用インタフェース（続き）

Pop. TRG_SALES インタフェース

目的

このインタフェースの目的は、Orders Application - HSQL モデルに含まれる注文の SRC_ORDERS 表および注文明細行の SRC_ORDER_LINES 表を、Sales Administration - HSQL モデルの TRG_SALES ターゲット表にロードすることです。データは、ターゲット表に統合する前に集計する必要があります。ステータスが'CLO'の注文のみを使用します。

ただし、ソース・データは、ターゲット環境の整合性規則と一致しない場合があります。この変換では、すべての制約が満たされていることを検証することで、データをクレンジングします。また、すべての無効な行を、ターゲット・データベースではなくエラー表に格納します。この場合、次の2つの重要な整合性規則を満たす必要があります。

- 販売が、TRG_CUSTOMER 表に存在する顧客 (CUST_ID) に関連付けられていること (参照 FK_SALES_CUST)
- 販売が、TRG_PRODUCT 表に存在する商品 (PRODUCT_ID) に関連付けられていること (参照 FK_SALES_PROD)

これらの規則の機能別詳細および手順を次に示します。

インタフェースの定義

ターゲット・データストア

モデル: Sales Administration - HSQL

データストア: TRG_SALES

ソース・データストア

モデル	データストア	説明	種類
Orders Application - HSQL	SRC_ORDERS	ソース・システムの注文表	HSQL 表
Orders Application - HSQL	SRC_ORDER_LINES	ソース・システムの注文明細行表	HSQL 表

結合

結合	説明	SQL 規則	実行場所
コマンドと注文明細行	SRC_ORDERS と SRC_ORDER_LINES の結合が必要	SRC_ORDERS.ORDER_ID = SRC_ORDER_LINES.ORDER_ID	ソース

フィルタ

説明	SQL 規則	実行場所
完了済の注文 (CLOSED) のみを取得	SRC_ORDERS.STATUS = 'CLO'	ソース

変換規則

ターゲット列	変換元	SQL 規則	実行場所
CUST_ID	SRC_ORDERS の CUST_ID	SRC_ORDERS.CUST_ID	ソース
PRODUCT_ID	SRC_ORDER_LINES の PRODUCT_ID	SRC_ORDER_LINES.PRODUCT_ID	ソース
FIRST_ORD_ID	ORDER_ID の最小値	MIN(SRC_ORDERS.ORDER_ID)	ソース
FIRST_ORD_DATE	SRC_ORDERS の ORDER_DATE の最小値	MIN(SRC_ORDERS.ORDER_DATE)	ソース
LAST_ORD_ID	ORDER_ID の最大値	MAX(SRC_ORDERS.ORDER_ID)	ソース
LAST_ORD_DATE	SRC_ORDERS の ORDER_DATE の最大値	MAX(SRC_ORDERS.ORDER_DATE)	ソース
QTY	注文明細行の QTY 数量の合計	SUM(SRC_ORDER_LINES.QTY)	ソース
AMOUNT	注文明細行の金額の合計	SUM(SRC_ORDER_LINES.AMOUNT)	ソース
PROD_AVG_PRICE	注文明細行の平均金額	AVG(SRC_ORDER_LINES.AMOUNT)	ソース

手順

新しいインタフェースの挿入

1. Demo プロジェクトの Sales Administration フォルダで Interfaces を右クリックし、「インタフェースの挿入」を選択します。
2. インタフェース・ウィンドウの「名前」フィールドに、インタフェースの名前 (Pop. TRG_SALES) を入力します。
3. ここで、インタフェース・ウィンドウを大きくして見やすくします。

ターゲット表の定義

1. 「ダイアグラム」タブで、Sales Administration - HSQL モデルの TRG_SALES 表を「ダイアグラム」タブの右にある「ターゲット・データストア」領域にドラッグします。

ソース・データストアの定義

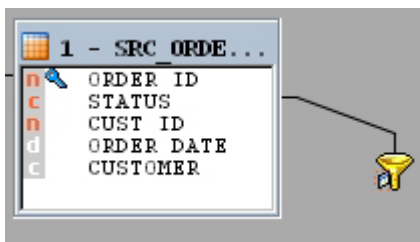
1. 「ダイアグラム」タブで構成パネル（「ダイアグラム」タブの左側）に次のソース・データストアをドラッグします。
 - Orders Application - HSQL モデルの SRC_ORDERS
 - Orders Application - HSQL モデルの SRC_ORDER_LINES

ソース・データストア間の結合の定義

1. **構成パネル**で、SRC_ORDERS の ORDER_ID 列を SRC_ORDER_LINES の ORDER_ID 列にドラッグします。これにより、注文番号の結合句が定義されます。

注文フィルタの定義


1. 「**ダイアグラム**」タブで、"SRC_ORDERS"の STATUS 列を**構成パネル**にドラッグします。
2. SRC_ORDERS.STATUS = 'CLO' と入力して、フィルタ規則の**実装**を変更します。



変換規則の定義

「ダイアグラム」の「**ターゲット・データストア**」領域で、ソースと名前が一致する列が（列名マッチングを使用して）自動的にマップされます。後は、FIRST_ORD_ID、FIRST_ORD_DATE、LAST_ORD_ID、LAST_ORD_DATE、QTY、AMOUNT および PROD_AVG_PRICE の各フィールドのマッピング・ルールを定義するだけです。

前の演習と同様に、各行が 1 つの列名に対応する表に、各列の名前が表示されます。変換規則は、フィールドごとに、ターゲット表で次のように定義します。

1. ロードするターゲット表のフィールドを選択します。
2. ダイアグラムの左下にある「**実装**」フィールドで規則を作成します。これは、ソース表のダイアグラムから必要な列をテキスト・フィールドにドラッグするか、 ボタンをクリックして式エディタをコールして行います。

次の規則を実装します。

- **FIRST_ORD_ID**: SRC_ORDERS.ORDER_ID 列を「**実装**」フィールドにドラッグ・アンド・ドロップします。規則を MIN (SRC_ORDERS.ORDER_ID) に変更します。
これにより、SRC_ORDERS 表の ORDER_ID 列の最小値がターゲット表の FIRST_ORD_ID 列にマップされます。
- **FIRST_ORD_DATE**: SRC_ORDERS.ORDER_DATE 列を「**実装**」フィールドにドラッグ・アンド・ドロップします。規則を MIN (SRC_ORDERS.ORDER_DATE) に変更します。
これにより、SRC_ORDERS 表の ORDER_DATE 列の最小値がターゲット表の FIRST_ORD_DATE 列にマップされます。
- **LAST_ORD_ID**: SRC_ORDERS.ORDER_ID 列を「**実装**」フィールドにドラッグ・アンド・ドロップします。規則を MAX (SRC_ORDERS.ORDER_ID) に変更します。
これにより、SRC_ORDERS 表の ORDER_ID 列の最大値がターゲット表の LAST_ORD_ID 列にマップされます。

- LAST_ORD_DATE: SRC_ORDERS.ORDER_DATE 列を「実装」フィールドにドラッグ・アンド・ドロップします。規則を MAX (SRC_ORDERS.ORDER_DATE) に変更します。これにより、SRC_ORDERS 表の ORDER_DATE 列の最大値がターゲット表の LAST_ORD_DATE 列にマップされます。
- QTY: 規則を SUM (SRC_ORDER_LINES.QTY) に変更します。これにより、商品数量の合計がターゲット表の QTY 列にマップされます。
- AMOUNT: 規則を SUM (SRC_ORDER_LINES.AMOUNT) に変更します。これにより、商品価格の合計がターゲット表の AMOUNT 列にマップされます。
- PROD_AVG_PRICE: SRC_ORDERLINES.AMOUNT 列を「実装」フィールドにドラッグ・アンド・ドロップします。規則を AVG (SRC_ORDER_LINES.AMOUNT) に変更します。これにより、商品価格の平均がターゲット表の PROD_AVG_PRICE 列にマップされます。

この時点で、ターゲット領域の表示は次のようになります。

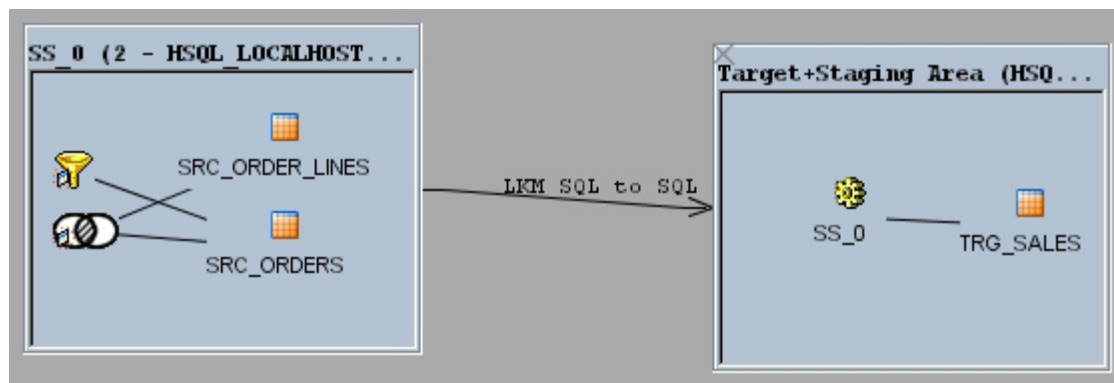
Ind	Name	Mapping
n	*CUST ID	SRC_ORDERS.CUST_ID
n	*PRODUCT ID	SRC_ORDER_LINES.PRODUCT_ID
n	*FIRST ORD ID	MIN(SRC_ORDERS.ORDER_ID)
d	*FIRST ORD ...	MIN(SRC_ORDERS.ORDER_DATE)
n	*LAST ORD ID	MAX(SRC_ORDERS.ORDER_ID)
d	*LAST ORD ...	MAX(SRC_ORDERS.ORDER_DATE)
n	*QTY	SUM(SRC_ORDER_LINES.QTY)
n	*AMOUNT	SUM(SRC_ORDER_LINES.AMOUNT)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

データ・ロード戦略 (LKM) の選択

「フロー」タブには、Oracle Data Integrator により、インタフェースの実行に必要な様々な手順が示されます。

ロード・ナレッジ・モジュール (LKM) を使用して、注文および注文明細行の集計結果をターゲット環境にロードする方法を選択する必要があります。

1. 注文明細行のフィルタ処理済集計結果のロードに対応するソース・セット (SS_0) をクリックします。ロード・ナレッジ・モジュールのリストから「LKM SQL to SQL」モジュールを選択します。
2. この時点で、「フロー」画面の表示は次のようになります。



データ統合戦略 (IKM) の選択

次に、データをターゲット表に統合するための戦略を選択する必要があります。これを行うには、使用する統合ナレッジ・モジュール (IKM) を指定する必要があります。

1. 「ターゲット+ステージング領域」オブジェクトをクリックして「IKM SQL Incremental Update」ナレッジ・モジュールを選択します。

Option	Value
CREATE_TARG_TABLE	<Default>:No
TRUNCATE	<Default>:No
DELETE_ALL	<Default>:No
SYNC_JRN_DELETE	<Default>:Yes
INSERT	<Default>:Yes
UPDATE	<Default>:Yes
FLOW_CONTROL	<Default>:Yes
RECYCLE_ERRORS	<Default>:No
STATIC_CONTROL	<Default>:No
COMMIT	<Default>:Yes
DELETE_TEMPORARY_OBJECTS	<Default>:Yes

2. ナレッジ・モジュール・オプションでは、上のようにデフォルト値をそのまま使用します。

前述のように、Oracle Data Integrator により、ターゲット表に挿入される前に、フロー内のデータがクレンジングされます。

データ制御戦略の選択

ここまでの処理では、ソースからターゲットへのデータ・フローのみが指定されました。ここでは、データのチェック方法 (CKM) を定義して、データを統合する前に満たす必要がある制約または条件を定義する必要があります。

1. 「制御」タブで、「CKM HSQL」チェック・ナレッジ・モジュールを選択します。

Option	Value
DROP_ERROR_TABLE	<Default>:No
DROP_CHECK_TABLE	<Default>:No

2. 検証する制約 (PK_TRG_SALES、FK_SALES_CUST および FK_SALES_PROD) を定義します。

Constraints	
Constraint Name	Value
PK_TRG_SALES	<Default>:Yes
FK_SALES_CUST	<Default>:Yes
FK_SALES_PROD	<Default>:Yes



インタフェースの実行準備が完了しました。

インタフェースの実行

1. インタフェース・ウィンドウの「実行」ボタンをクリックしてインタフェースを実行します。
2. インタフェースの保存を求めるウィンドウが表示されます。「はい」をクリックします。
3. 実行ウィンドウおよび情報ウィンドウで「OK」をクリックします。

Oracle Data Integrator によって自動的に SQL コードが生成され、実行セッションが開始されます。

オペレータでのインタフェースの実行の追跡

1. オペレータで実行の結果を追跡するには、メニュー・バーの  アイコンをクリックします。
2. オペレータ・ウィンドウが開き、実行処理タスクの詳細が表示されます（このウィンドウがすでに開いている場合は、 ボタンをクリックしてリフレッシュできます）。

結果の解釈

処理されたレコード

1. 「オペレータ」で実行ツリーの対応するアイコンをダブルクリックし、Pop. TRG_SALES ステップのウィンドウを開きます。画面の右に新しいウィンドウが開きます。
2. 「実行」タブに、TRG_SALES 表のロードによって 62 個の挿入が作成され、32 個の無効なレコードがエラー表に分離されたことが示されます。

Nb of Inserts	62	Nb of Updates	0
Nb of Deletes	0	Nb of Errors	32

作成されたデータ

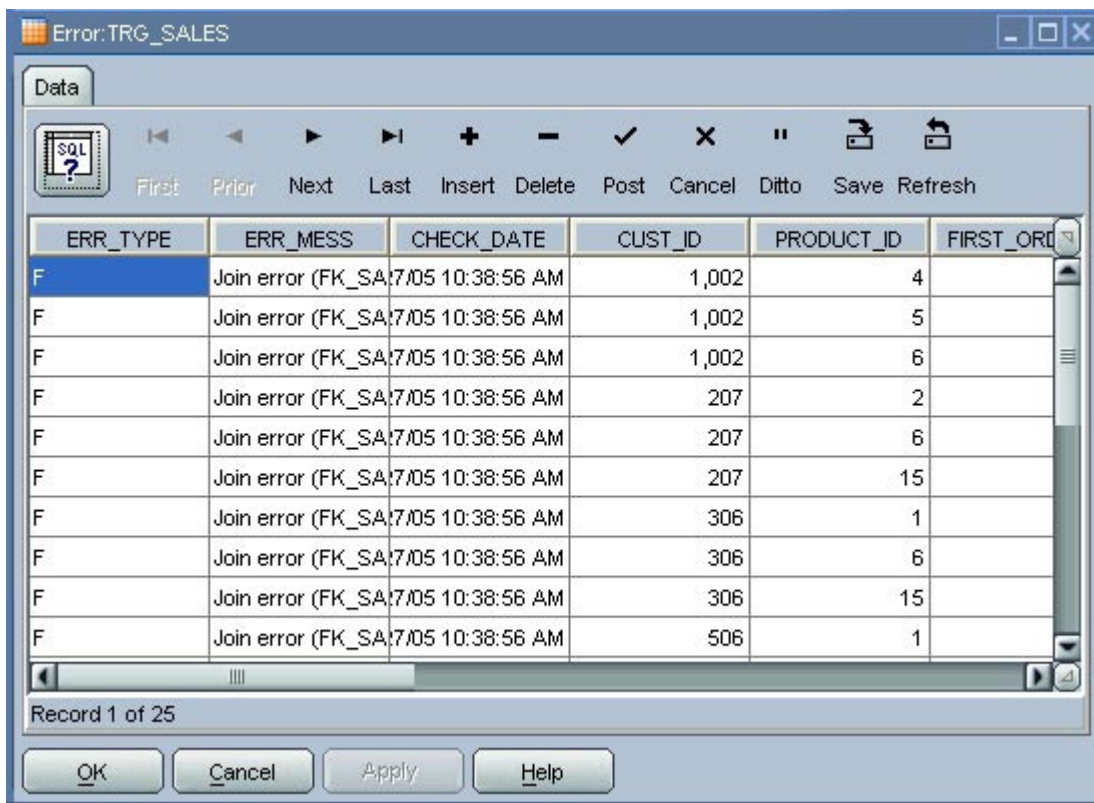
インタフェースによって作成されたデータは、デザイナー・モジュールで確認できます。

1. Sales Administration - HSQL モデルを開いて TRG_SALES 表を選択します。
2. 右クリックして「データ」を選択し、ターゲット表のデータを表示します。

CUST_ID	PRODUCT_ID	FIRST_ORD_ID	FIRST_ORD_DATE	LAST_ORD_ID	LAST_ORD_DATE	QTY
103	7	3	3/23/90	3	3/23/90	1
103	8	3	3/23/90	71	3/18/99	1
103	9	3	3/23/90	71	3/18/99	1
104	6	4	4/15/90	77	9/12/00	4
104	11	4	4/26/91	4	4/26/91	1
104	12	4	4/15/90	77	9/12/00	5
106	7	6	6/23/98	38	7/23/99	2
106	13	6	6/23/98	74	7/23/99	4
106	14	6	6/23/98	74	7/23/99	2
201	2	8	8/18/00	40	9/10/00	2
201	4	8	8/18/00	8	8/18/00	1
201	10	8	8/18/00	40	9/10/00	3
203	4	10	5/20/99	10	5/20/99	1
203	7	10	5/20/99	10	5/20/99	1
203	9	10	5/20/99	10	5/20/99	1

無効なレコード

実行したインタフェースでは、**32** 個の無効なレコードが自動作成されたエラー表に分離されました。これらのレコードにアクセスするには、モデルからターゲット表 (TRG_SALES) を選択し、ポップアップ・メニューから「制御」→「エラー」の順に選択します。



この画面から、次のレコードがインタフェースによって拒否されたことがわかります。

- FK_SALES_CUST 制約に違反している（つまり、顧客番号が顧客の表に存在しない）**31** 個のレコード。
- FK_SALES_PROD 制約に違反している（つまり、商品番号が商品の表に存在しない）**1** 個のレコード。

無効なレコードはエラー表に保存され、ターゲット表には統合されません。

操作

操作

操作の概要

Oracle Data Integrator は、完全なエンタープライズ・アプリケーション統合（EAI）からデータ・ウェアハウスの表への単純な移入まで、アプリケーション間での情報交換を自動化することを目的として設計されています。

そのため、異なる処理（インタフェース、プロシージャなど）の実行を順序付けして、それぞれの処理の即時使用可能なコードを含む本番シナリオを作成する必要があります。

パッケージ

パッケージは、事前定義済の一連のステップで、順番に実行されるように設計されます。次のような様々なステップがあります。

- インタフェース
- プロシージャ
- 変数に対するアクション/評価
- モデル、サブモデルまたはデータストアに対するアクション
- OS コマンド
- Oracle Data Integrator のツール

この演習では、一連のインタフェースを使用して Sales Administration アプリケーションをロードします。このアプリケーションの表の間には参照制約が存在するため、TRG_CUSTOMER 表をロードするには、先に TRG_CITY 表をロードしておく必要があります。

「**Load Sales Administration パッケージ**」の項では、独自に開発したインタフェースおよび Demo プロジェクトに同梱されているインタフェースが含まれるパッケージを作成して実行します。

シナリオ

シナリオは、パッケージに含まれるオブジェクト（インタフェース、プロシージャなど）の部分的に生成されたコード（SQL、シェルなど）です。

注意: 生成されたシナリオは、Work リポジトリに格納されます。格納されたシナリオは、エクスポートして別のリポジトリにインポートすることで、1 つ以上の異なるコンテキストで使用できます。

注意: シナリオのコードが生成されると、そのコードへのアクセスは制限されます。コードの作成に使用されたパッケージまたはデータ・モデル（もしくはその両方）がその後変更された場合でも、コードには影響しません。インタフェースの 1 つが変更されたなどの理由でシナリオを更新する場合は、パッケージから新しいバージョンのシナリオを生成する必要があります。

シナリオのプロパティは、**デザイナー・モジュールのシナリオ・ウィンドウのシナリオ・ウィンドウ**、または**オペレータ・モジュール**から表示できます。シナリオの操作は、同じウィンドウまたは**オペレーティング・システムのコマンド・ウィンドウ**から実行できます。デザイナー・モジュールでは、シナリオが対応するパッケージの下にグループ化されて表示されます。

このスタート・ガイドでは、パッケージを作成し、作成したパッケージから **Load Sales Administration** というシナリオを生成します。また、このシナリオを Oracle Data Integrator から実行し、その後、コマンド・プロンプトから実行します。

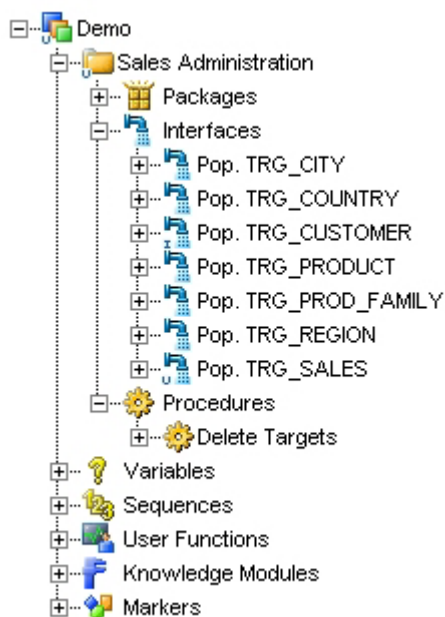
Load Sales Administration パッケージ

目的

このパッケージの目的は、Sales Administration アプリケーションのロードに関する完全なワークフローを定義し、実行順序を設定することです。

Oracle Data Integrator に同梱されている開発済オブジェクト

これまでに開発した2つのインタフェースに加えて、デモ・リポジトリには多数の開発済オブジェクトが同梱されています。



インタフェース

- **Pop. TRG_COUNTRY:** TRG_COUNTRY 表を移入するインタフェース
- **Pop. TRG_REGION:** TRG_REGION 表を移入するインタフェース
- **Pop. TRG_CITY:** TRG_CITY 表を移入するインタフェース
- **Pop. TRG_PROD_FAMILY:** TRG_PROD_FAMILY 表を移入するインタフェース
- **Pop. TRG_PRODUCT:** TRG_PRODUCT 表を移入するインタフェース

プロシージャ

- **Delete Targets:** Sales Administration アプリケーション内のすべての表を空にするプロシージャ。この操作は、各表で DELETE 文を使用して実行されます。

問題の分析

(参照整合性制約に従って) Sales Administration アプリケーションを正しくロードするには、次の順序でタスクを実行する必要があります。

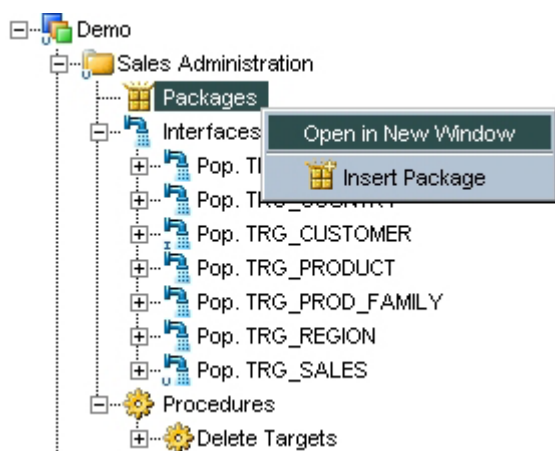
1. Delete Targets プロシージャを使用して Sales Administration 表を空にします。
2. Pop. TRG_COUNTRY インタフェースを使用して TRG_COUNTRY 表をロードします。
3. Pop. TRG_REGION インタフェースを使用して TRG_REGION 表をロードします。
4. Pop. TRG_CITY インタフェースを使用して TRG_CITY 表をロードします。

5. Pop. TRG_PROD_FAMILY インタフェースを使用して TRG_PROD_FAMILY 表をロードします。
 6. Pop. TRG_PRODUCT インタフェースを使用して TRG_PRODUCT 表をロードします。
 7. Pop. TRG_CUSTOMER インタフェースを使用して TRG_CUSTOMER 表をロードします。
 8. Pop. TRG_SALES インタフェースを使用して TRG_SALES 表をロードします。
- このような統合プロセスは、Oracle Data Integrator で **Package** 形式で作成されます。

ステップ


パッケージの作成

1. **Packages** オブジェクトを右クリックして「パッケージの挿入」を選択します。

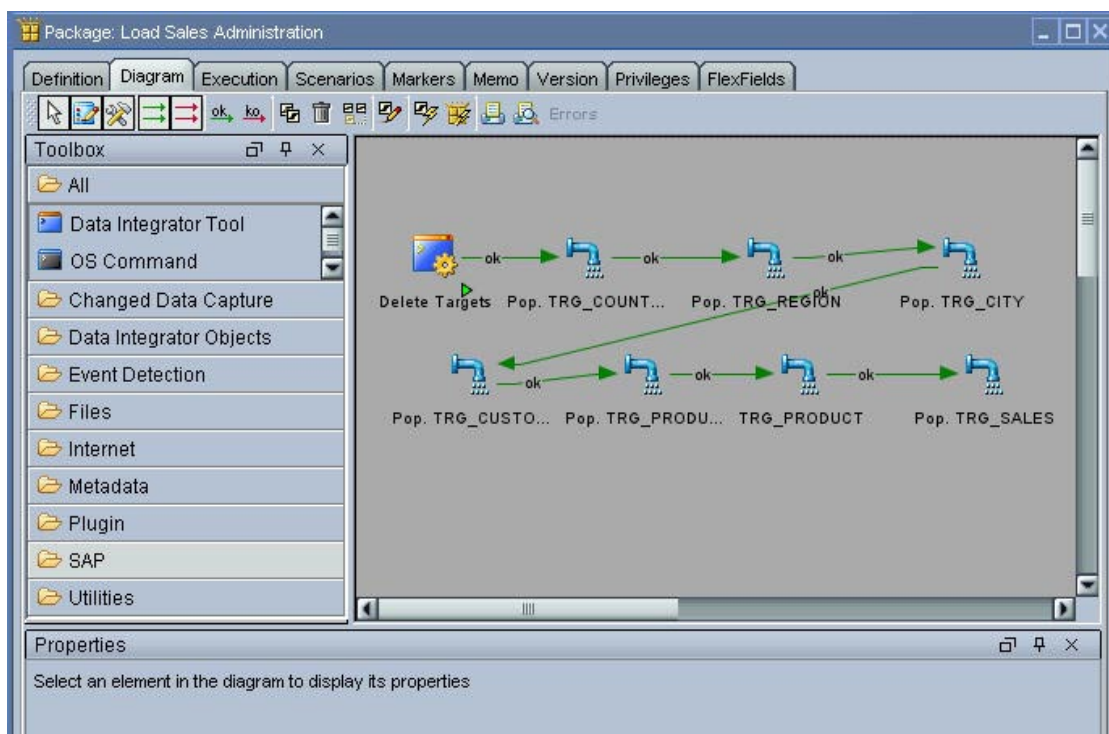


2. 「名前」フィールドに Load Sales Administration と入力します。
3. 「適用」をクリックして、この空のパッケージを保存します。

パッケージのステップの定義

1. パッケージ・ウィンドウで「ダイアグラム」タブをクリックします。
2. パッケージのステップを定義するには、「プロジェクト」ツリーから、パッケージに統合する各コンポーネントを選択し、「ダイアグラム」にドラッグ・アンド・ドロップします。これらのコンポーネントがパッケージのステップとして表示されます。ただし、まだ順序付けされていません。
3. パッケージで最初に実行するステップ (Delete Target プロシージャ) を指定するには、このステップを選択して右クリックし、コンテキスト・メニューから「最初のステップ」を選択します。このステップに、小さい緑色の矢印が表示されます。
4. ダイアグラム・ツールバーの「成功時の次のステップ」 アイコンを選択します。
5. このツールを使用してステップを順序付けするには、1つのステップをクリックしてから、その次に実行するステップをクリックします。この操作を繰り返してすべてのステップの順序を指定し、開発したオブジェクトを次のように順序付けします。

1. Delete Targets (最初のステップ)
 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. 作成したパッケージの表示は次のようになります。



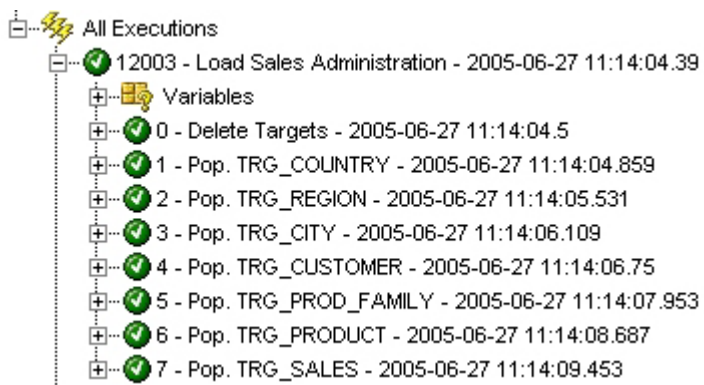
7. 「適用」をクリックしてパッケージへの変更を保存します。

パッケージの実行

1. このパッケージを実行するには、「実行」ボタンをクリックします。
2. 実行ウィンドウおよび情報ウィンドウで「OK」をクリックします。

オペレータでのパッケージの実行の追跡

オペレータで実行を確認すると、個々のインタフェースの実行時に取得されたものと同じ結果が表示されます。



デプロイ

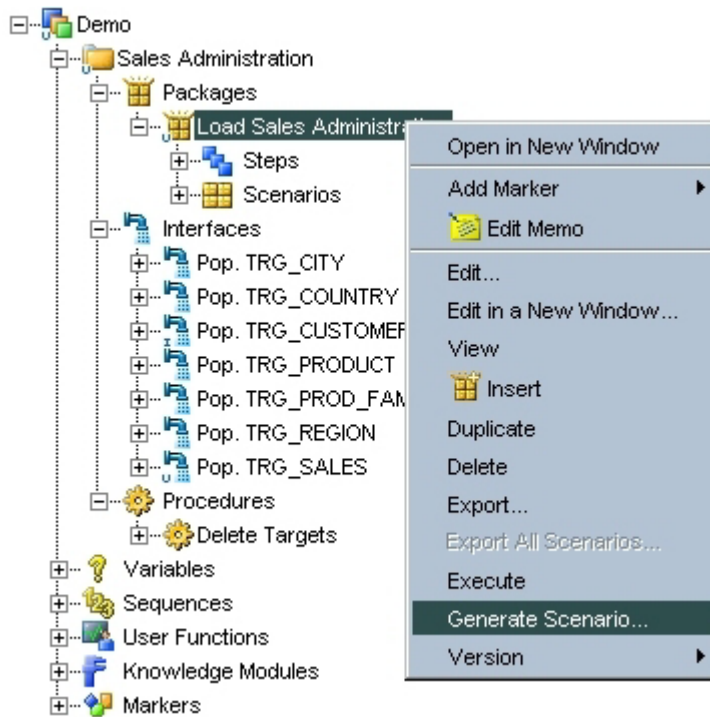
目的

この項の目的は、本番環境でこのパッケージを自動的に実行できるようにすることです。そのためには、Load Sales Administration パッケージの実行に使用するシナリオ `LOAD_SALES_ADMINISTRATION` を作成する必要があります。

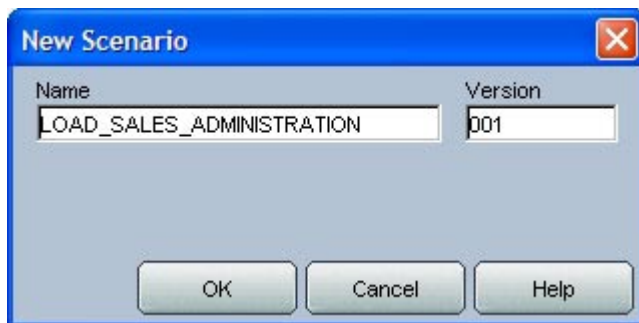
ステップ

シナリオの作成

1. Load Sales Administration パッケージを選択します。
2. 右クリックして「シナリオの生成」を選択します。



3. 次のウィンドウが表示されます。



4. 「OK」 をクリックします。Oracle Data Integrator によってシナリオが処理され、生成されます。
5. 結果は次のようになります。



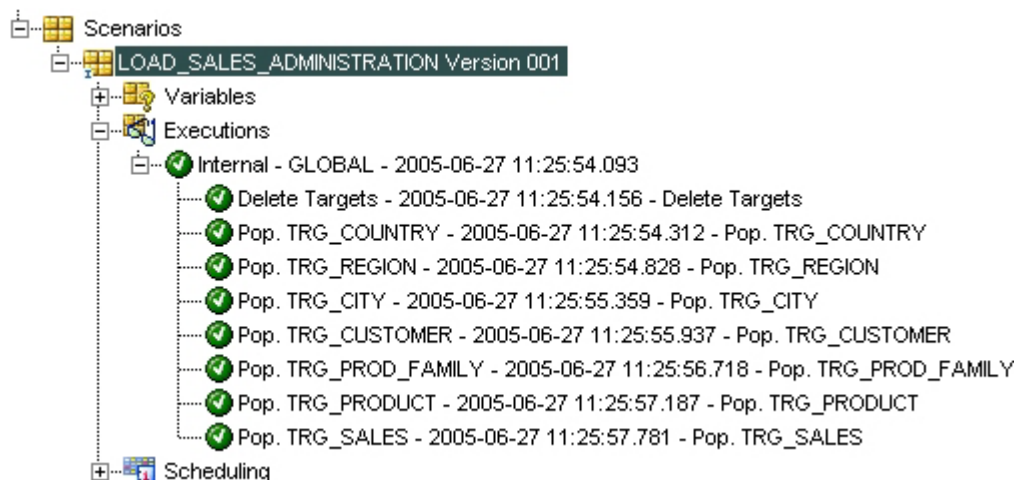
Oracle Data Integrator からのシナリオの実行

1. シナリオを右クリックします。
2. 「実行」を選択します。
3. 実行ウィンドウおよび情報ウィンドウで「OK」をクリックします。

シナリオの実行の追跡

シナリオの実行は、「オペレータ」で確認できます。結果は、パッケージの実行時に取得されるものと同じになります。

また、次に示すように**デザイナー**・モジュールからシナリオのツリー表示を開いて、シナリオの実行レポートを確認することもできます。



OS コマンドからのシナリオの実行

1. MS-DOS コマンド・プロンプト・ウィンドウまたは UNIX のコンソールを開きます。
2. Oracle Data Integrator のインストール・フォルダで bin ディレクトリを開きます。
3. 次のコマンドを入力します。
`startscen LOAD_SALES_ADMINISTRATION 001 GLOBAL "-v=2"`

注意: startscen コマンドのパラメータは次のとおりです。

- Scenario name
- Scenario version
- Scenario context
- Logging level

これらのパラメータはスペースで区切られます。

4. 実行終了時のコマンド・プロンプトの表示は、次のようになります。

```
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
DwqJv.main: Exit.Return code:0
```

注意: この例では、シナリオをコールするためにコマンド・プロンプトを使用します。シナリオのコールは、ほとんどの外部アプリケーションから実行可能です。逆に、すでに使用している外部アプリケーションを Oracle Data Integrator からコールすることもできます。そのため、別のアプリケーションとの相互作用をプロセスの一部として統合できます。

まとめ

短時間で簡単に ETL プロジェクトを完成させることができました。

それぞれの例を通して、次の処理を実行しました。

- Orders Application アプリケーションでデータ整合性規則を定義および実装しました。
- Orders Application アプリケーションおよび Parameters (File) アプリケーションのデータを Sales Administration データ・ウェアハウスにロードするために、2つのインタフェースを作成しました。
- 開発したオブジェクトを順序付けしてプロセスの実行を自動化しました。

これらの例は、Oracle Data Integrator によって1つの回答が提供される、1種類の問題のみに対応しています。

Oracle Data Integrator で実行できるその他の処理

これまでは、一般的なデータ・ウェアハウス・プロジェクトに対する Oracle Data Integrator の使用方法を見てきました。実際には、Oracle Data Integrator は、バッチからニアリアルタイムまであらゆる種類のデータ駆動の統合に対処できます。他にも、次のような例があげられます。

- データの移行（後で旧システムと新規システムの間でレプリケーションを行う場合または行わない場合）
- Point-to-Point データ統合
- データ・レプリケーション

また、この例では、Oracle Data Integrator がリレーショナル・データベースおよびファイルとのみ接続されていました。実際には、同じツールセット内で同じ方法を使用して、ERP および CRM、メインフレーム、フラット・ファイル、LDAP ディレクトリ、XML データソースなどのすべてのデータベース・システムにアクセスし、統合できます。

Oracle Data Integrator は、共通の宣言規則主導アプローチにより、データ、イベントおよびサービスに基づく統合を一体化する、唯一の統合プラットフォームです。これを使用することで、企

業は、統一された単一のアクセス・モデルを使用して、情報システムの単一の表示を提示できません。

Oracle Data Integrator を使用することには、次のような利点があります。

- **統一された統合のサポート:** Oracle Data Integrator は、同じインタフェースを使用した、データ指向、イベント指向およびサービス指向の統合をサポートする唯一の統合アプリケーション・ソフトウェアです。この固有の機能により、IT チームは、データ量やレイテンシ要件に関係なく統合のあらゆるニーズ（バッチおよびリアルタイム、非同期および同期）に対応できます。
- **生産性の強化および短期間での習得:** データ指向、イベント指向およびサービス指向の各統合メカニズムに関係なく、Oracle Data Integrator 全体で、宣言規則主導のアプローチが共有されます。プラットフォーム全体で共通のモデルを使用してユーザー・インタフェースを共有するため、習得時間が短縮され、生産性が劇的に向上します。
- **再利用可能な共有メタデータ:** Oracle Data Integrator のすべてのコンポーネントと完全に統合された単一のメタデータ・リポジトリを使用するため、統合プロセスの一貫性が保証されます。また、このリポジトリは、プロセス全体でのデータ変換およびデータ検証の宣言規則の再利用性を高めます。
- **複数のアプリケーションのサポート:** Oracle Data Integrator は、データ指向、イベント指向およびサービス指向のメカニズムを組み合わせる実装される、広範囲の統合プロジェクト（ETL、データの移行、マスター・データ管理、ビジネス・アクティビティ管理（BAM）、ビジネス・プロセス管理（BPM）、ビジネス・プロセス・リエンジニアリング（BPR）および Web サービス統合）に適しています。

Oracle Data Integrator の詳細情報

Oracle Data Integrator について把握するためには、デモンストレーション環境を使用してください。

次のドキュメントを参照すると、Oracle Data Integrator の詳細を学習できます。

ドキュメント	説明
インストレーション・ガイド	システム要件および Oracle Data Integrator のインストール方法について説明します。
リファレンス・マニュアル	Oracle Data Integrator のグラフィカル・ユーザー・インタフェースの詳細を説明します。
ユーザーズ・ガイド	Oracle Data Integrator での作業方法について説明します。

オラクル社の連絡先

サポートまたは詳細情報が必要な場合は、オラクル社に連絡してください。

www.oracle.com