

OpenStaging ユーザーマニュアル

R 1.0.0

2006 年 2 月



商標

本書に記載されている社名および商品名は、それぞれ各社の商標または登録商標です。

なお、本文中には TM、® マークは明記しておりません。

目次

1.1 OpenStaging の特徴	2
1.1.1 従来のステージング方法	2
1.1.2 OpenStaging の特徴	2
1.2 OpenStaging の画面構成	4
1.2.1 OpenStaging の画面構成	4
1.2.2 使用できない文字列	4
2.1 OpenStaging へのログイン	6
2.2 OpenStaging からのログアウト	7
3.1 データソースを登録する	10
3.2 データソースを編集する	11
3.3 データソースを削除する	13
4.1 ETL ジョブを登録する	16
4.2 ETL オブジェクトを設定する	18
4.2.1 ステップ 1 : ソースとターゲットを登録する	19
4.2.2 ステップ 2 : DB ブリッジを登録する	25
4.2.3 ステップ 3 : Extract を登録する	27
4.2.4 ステップ 4 : Transform を登録する	33
4.2.5 ステップ 5 : Load を登録する	35
4.3 カスタムスクリプトを登録する	37
4.4 XML を編集する	39
4.5 テスト実行をする	40
4.6 モニタリング画面を表示する	42
4.7 オプションメニューを利用する	44
4.7.1 ルックアップを登録する	44
4.7.2 データウェアハウスの履歴管理	47
4.8 ETL ジョブを編集する	55
4.9 ETL ジョブを削除する	55
4.10 プログラムコードを編集する	56
5.1 テーブル移行ジョブを登録する	60
5.2 テーブル移行ジョブを編集する	63
5.3 テーブル移行ジョブを削除する	63
6.1 スケジュールを登録する	66
6.2 スケジュールを編集する	72
6.3 スケジュールを削除する	73
7.1 ユーザー・パスワード設定	76
7.2 コマンドラインからのジョブの実行	76
7.3 サンプルデータ	77

7.3.1 サンプル 1 : 業務用テーブル ER 図	78
7.3.2 サンプル 2 : DWH 用テーブル ER 図	80
7.4 使い方の応用例	81
7.4.1 応用例 1 : xml ファイル	81
7.4.2 応用例 2 : SQL 構文	81
7.4.3 応用例 3 : データ型変換ルール	82
7.5 未サポート RDB の OpenStaging 実装方法	82

OpenStaging の概要

OpenStaging は ETL 処理を GUI で操作できるステージングツールです。オープンソースデータベースを対象とし、膨大で複雑なデータをビジュアルかつ迅速に、役立つデータへと変換します。この章では、OpenStaging の機能と画面構成をご紹介します。

この章は以下のとおり構成されています。

- [OpenStaging の特徴](#)
- [OpenStaging の画面構成](#)

1.1 OpenStaging の特徴

エンドユーザーはインターネットブラウザを用意だけで、ステージング作業を行なうことができます。わかりやすい GUI を提供し、直感的な操作を可能にします。

最大の特徴は、RDBMS の SQL エンジンで変換処理を行うことで、このため RDBMS の処理能力に応じて大規模なデータウェアハウスにも対応できる、実用的なアーキテクチャとなっています。また、ストアドプロシージャの使用によってパフォーマンスが向上し、簡単、短時間なデータ統合を実現します。

1.1.1 従来のステージング方法

業務システムのデータを分析システムのデータに変換するステージングのために、従来は以下の処理をプログラム開発する必要がありました。

- ①「Extract(抽出)」、「Transform(変換)」、「Load(ロード)」の ETL 処理
- ②データのクレンジング
- ③データソースの移行
- ④スケジューリング等の運用管理

そのためデータウェアハウス構築にかかる工数の多くを、ステージング作業に割いてきました。

OpenStaging はこれらの処理を GUI によりノンプログラミングに行うことができ、高いパフォーマンスでステージング作業をサポートします。

1.1.2 OpenStaging の特徴

OpenStaging の主な特徴は以下のとおりです。

- ・ OpenStaging はデータベース内のネイティブストアドプロシージャによってデータの処理を行ないます。通常の ETL ツールでは独自のエンジン内でデータの取り込みや変換処理を行うため、パフォーマンスが低下し、また、生成された処理がブラックボックスとなるためユーザーがコードを確認することができませんでした。OpenStaging はコードジェネレート方式によりこれらの問題を解消します。コードジェネレートによる利点は以下のとおりです。
 - ー DB 内で処理が完結するので、パフォーマンスが高い。
 - ーコードレベルで全てのプログラムを確認、編集することができる。

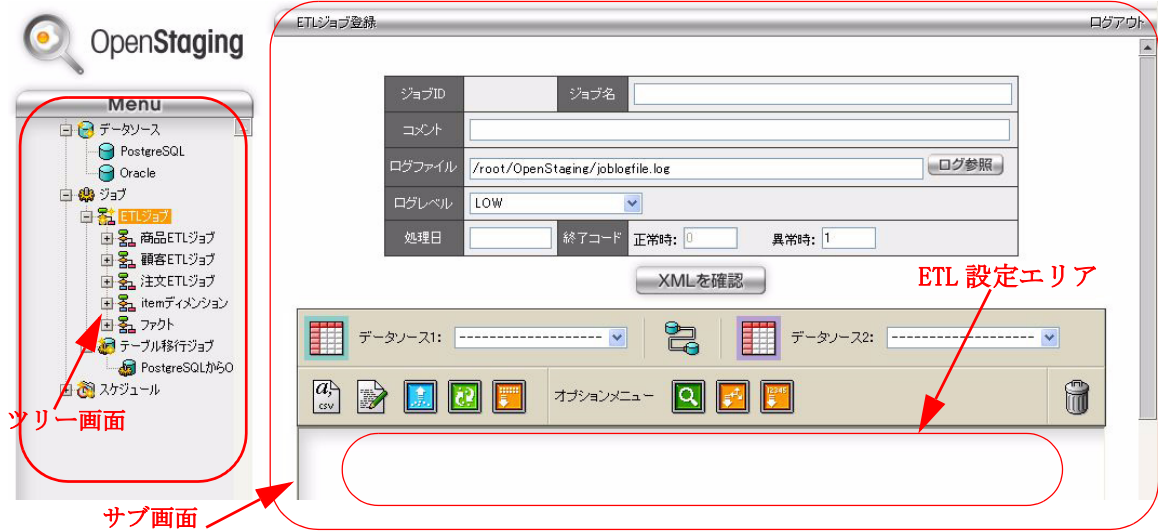
ーデータベースごとの設定ファイルを変更することで、ユーザー環境にあったプログラムを生成することができる。
ー単純なネイティブストアドプロシージャ生成ツールとしても利用できる。

- ・ 柔軟なメタ管理
メタデータ・データベース情報を XML で管理しているため、高いポータビリティ（環境を簡単に移植できる）を備えています。物理的なデータベースの変更も、ETL 設定に反映させたい箇所だけを選んで自動更新することができます。
- ・ DWH 機能
ルックアップ機能、履歴管理機能（サロゲートキー、NBT 処理）などデータウェアハウスならではの多彩な運用を、GUI で簡単に操作できます。
- ・ スケジュール機能
簡易スケジュール機能を搭載し、ジョブをスケジュールすることができます。別のスケジューラ上で運用することも可能です。
- ・ オープンソースデータベース
オープンソースデータベースをサポートしています。
データソースとして、PostgreSQL、MySQL、Oracle、CSV ファイルを指定できます。ターゲットとしては、PostgreSQL、MySQL、Oracle を指定することができます。
- ・ テーブル移行機能
異なるデータソース間でテーブルを簡単に移行することができます。
業務用データをデータウェアハウスのテーブルとして一括して移行するときや、テスト環境を作るときなどに使用できます。

1.2 OpenStaging の画面構成

1.2.1 OpenStaging の画面構成

OpenStaging の各個所の名称は以下のとおりです。



ツリー画面

ツリーオブジェクトからデータソース、ジョブ、スケジュールを選択します。

サブ画面

データソース、ジョブ、スケジュールなどの登録画面を表示します。

ETL 設定エリア

ETL ジョブ登録画面で、ETL 設定オブジェクトを表示します。

1.2.2 使用できない文字列

OpenStaging のデータソース名、ジョブ名、オブジェクト名、スケジュール名には以下の 6 種類の文字列を入力することはできません。入力した場合、記号と丸囲み文字以外はエラーメッセージが表示されます。記号と丸囲み文字は、登録後に再表示すると文字化けします。

記号、(例：～、♪など) 丸囲み文字 (例：①、②など)、「<」、「>」、「&」、「」

起動と終了

OpenStaging の起動と終了の方法を説明します。

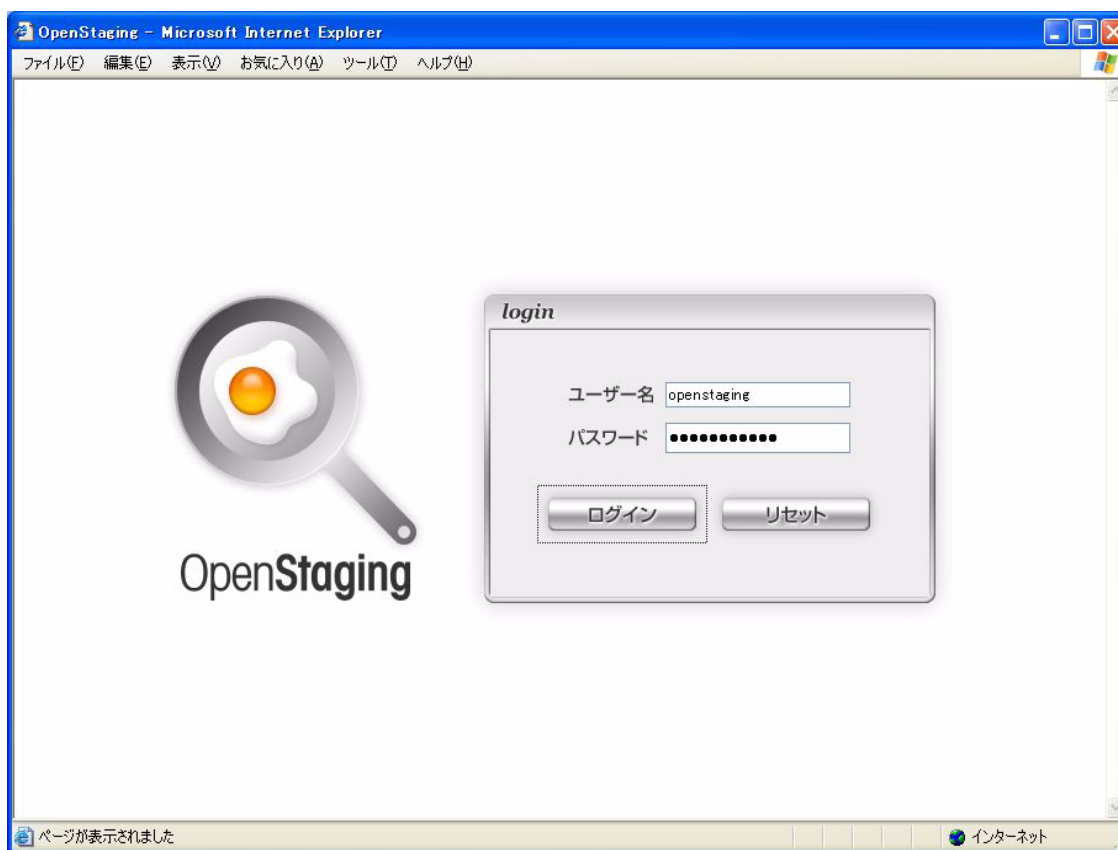
この章は以下の通り構成されています。

- [OpenStaging へのログイン](#)
- [OpenStaging からのログアウト](#)

2.1 OpenStaging へのログイン

OpenStaging にログインするために、事前に次の情報を確認しておきます。

- ・ 管理者のユーザー名とパスワード
 - ・ OpenStaging の URL
1. インターネットブラウザを起動します。
 2. OpenStaging ログイン画面の URL を指定します。



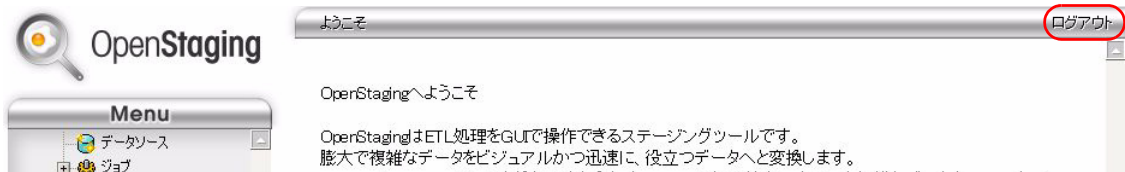
3. ユーザー名とパスワードを入力して、[ログイン] ボタンをクリックします。OpenStaging が起動して、ホーム画面が表示されます。



メモ： [リセット] ボタンをクリックすると、ユーザー名とパスワードがクリアされます。

2.2 OpenStaging からのログアウト

OpenStaging からログアウトするには、画面右上の「ログアウト」をクリックします。



メモ： 何も操作しないまま管理者の設定するアイドル時間を過ぎると、ログイン画面に自動的に遷移します。

データソースの設定

ETL ジョブ、テーブル移行ジョブを作成する前に、処理対象となるテーブルのデータソースを作成します。

この章は以下のとおり構成されています。

- ・ [データソースを登録する](#)
- ・ [データソースを編集する](#)
- ・ [データソースを削除する](#)

3.1 データソースを登録する

- オブジェクトツリーの [データソース] アイコンを選択して、[データソース登録] 画面を表示します。



- 以下の情報を入力します。

データソースID				データソース名	PostgreSQL
コメント					
データベース	PostgreSQL ▼				
JDBC URL	jdbc:postgresql://192.168.1.95:5432/openstaging				
ユーザー名	openstaging		パスワード	●●●●●●●●	
初期実行スクリプト	SET search_path TO my_schema, public;				

項目	説明
データソース ID	データソースのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 新規作成時は非表示となります。
データソース名	データソース名を入力します。
コメント	データソースに対するコメントを入力します。
データベース	MySQL/PostgreSQL/Oracle のうち、使用するデータベースを選択します。
JDBC URL	指定したデータベースに jdbc から接続するための URL を入力します。
ユーザー名	ユーザー名を入力します。
パスワード	パスワードを入力します。
最終更新日	データソースの最終更新日を表示します。変更はできません。 * 新規作成時は非表示となります。

3. [作成] ボタンをクリックして、データソース情報を保存します。

メモ： ツリーオブジェクト [データソース] に、作成したデータソースのアイコンが追加されます。



メモ： 接続テストボタン

[接続テスト] ボタンをクリックして、設定した内容で接続可能かテストすることができます。

3.2 データソースを編集する

データソース情報を編集する場合、またはデータベース構成（テーブル、ビュー、インデックス）の変更を反映させたい場合に使用します。

1. オブジェクトツリーの [データソース] アイコンの内、編集したいデータソースを選択して [データソース情報] 画面を表示します。



2. 項目を編集します。

注記： [データソース ID] は編集できません。

3. [更新] ボタンをクリックして、データソースの情報を保存します。

メモ： データソースの編集について

- ・ データベース内のテーブル（ビュー）、インデックスに変更があった場合、更新時に変更内容が表示されます。
- ・ データソースから削除したテーブルとテーブル構成の等しいテーブルが存在する場合、削除したテーブルのオブジェクト ID を引き継ぐことができます。またテーブルのカラム名を変更した場合も、変更後のカラムに ID を引き継ぐことができます。

差分確認

追加されたテーブル	 SH.PRODUCTS2  SH.PROD					
削除されたテーブル	削除テーブルは、他の同じ構造のテーブルにテーブルIDを引き継ぐことができます。 なし					
カラム構成に変更があったテーブル	削除されたカラムは、追加されたカラムにカラムIDを引き継ぐことができます。  S.EMP <table border="1"> <tr> <td>追加カラム</td> <td>SALES</td> </tr> <tr> <td>削除カラム</td> <td> SAL <input checked="" type="checkbox"/> 引き継ぐ SALES ▼ </td> </tr> </table>		追加カラム	SALES	削除カラム	SAL <input checked="" type="checkbox"/> 引き継ぐ SALES ▼
追加カラム	SALES					
削除カラム	SAL <input checked="" type="checkbox"/> 引き継ぐ SALES ▼					

変更を引き継ぐ場合はチェックオンにします

注記： 差分確認をするときデータソースが MySQL の場合、テーブル名の
大文字小文字の区別は動作対象外です。

例）テーブル名を「PRODUCTS」から「Products」に変更した場合

3.3 データソースを削除する

1. オブジェクトツリーの「データソース」アイコンの内、削除したいデータソースを選択して「データソース情報」画面を表示します。
2. 「削除」ボタンをクリックして、データソースを削除します。削除後、画面は「データソース登録」画面へ切り替わります。

メモ： 削除したデータソースのアイコンが、オブジェクトツリーから削除されます。

ETL ジョブの作成

ETL 処理を行なうためのジョブを作成します。Extract(抽出)、Transform(変換)、Load(ロード)の機能の他に、ルックアップ機能(クレンジング)、履歴管理機能を使用できます。

この章は以下のとおり構成されています。

- [ETL ジョブを登録する](#)
- [ETL オブジェクトを設定する](#)
- [カスタムスクリプトを登録する](#)
- [XML を編集する](#)
- [テスト実行をする](#)
- [モニタリング画面を表示する](#)
- [オプションメニューを利用する](#)
- [ETL ジョブを編集する](#)
- [ETL ジョブを削除する](#)

4.1 ETL ジョブを登録する

- オブジェクトツリーの [ジョブ] - [ETL ジョブ] を選択して、[ETL ジョブ登録] 画面を表示します。



- 以下の情報を入力します。

ジョブID		ジョブ名	商品ETLジョブ	
コメント	商品のETLジョブを作成する			
ログファイル	/root/user01/joblogfile.log			ログ参照
ログレベル	MIDDLE ▼			
処理日	Today	終了コード	正常時: 0	異常時: 1

項目	説明
ジョブ ID	ジョブ ID を OpenStaging が自動的に割り振って表示します。 * 新規作成時は非表示となります。
ジョブ名	ETL ジョブ名を入力します。
コメント	ETL ジョブに対するコメントを入力します。
ログファイル	ログファイルの保存先を絶対パスで入力します。
ログレベル	出力するログの情報レベルを選択します。
処理日	<p>処理日を変数として使用したり、履歴管理の有効期間として使用できます。ETL ジョブの実行日に依存しない日付が必要な場合は、[yyyy/mm/dd] の書式で日付を入力します。日付はアプリケーションサーバーのシステム日付です。</p> <p>* 未入力の場合はジョブ実行時に、自動的にジョブ実行開始日付が設定されます。</p> <p>* ETL ジョブ実行の当日、または前日を設定できます。</p> <p>ー 実行当日の場合「Today」または「0」と入力</p> <p>ー 実行前日の場合「Yesterday」または「-1」と入力</p> <p>* 変数として使用する場合</p> <p>Extract、Transform、カスタムスクリプトで [処理日] を変数「os_pv_process_date」として使用することができます。</p> <p>例) テーブルからデータ抽出 (Extract) するときに time= 処理日で絞り込む場合→</p> <p>[Extract] 登録画面で time 列の [WHERE] に「=os_pv_process_date」と入力</p>

項目	説明
終了コード	ETL ジョブ実行時の終了コードを入力します。 [正常時] 0 [異常時] 異常時の戻り値を入力します。 *[正常時] は変更できません。
最終更新日	ETL ジョブの最終更新日を表示します。変更はできません。 * 新規作成時は非表示となります。

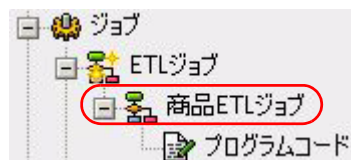
- ETL オブジェクトを設定します。ETL オブジェクトの設定については [『4.2 ETL オブジェクトを設定する』](#) を参照してください。
- [作成] ボタンをクリックして、ETL ジョブの情報を登録します。

メモ： ログ参照ボタン

[ログファイル] で指定したログファイルを表示します。



メモ： ツリーオブジェクト [ジョブ] - [ETL ジョブ] に、作成した ETL ジョブのアイコンが追加されます。また、作成した ETL ジョブの下に、生成されたプログラムコードのアイコンが追加されます。プログラムコードについては [『4.10 プログラムコードを編集する』](#) を参照してください。



4.2 ETL オブジェクトを設定する








ETL オブジェクトの設定方法をステップを追って説明します。

- ・ [ステップ 1 : ソースとターゲットを登録する](#)
- ・ [ステップ 2 : DB ブリッジを登録する](#)
- ・ [ステップ 3 : Extract を登録する](#)
- ・ [ステップ 4 : Transform を登録する](#)

[ステップ 5 : Load を登録する](#)

ETL ジョブ作成画面には ETL 設定のためのアイコンが用意されています。



アイコン	アイコン名	説明
	テーブル・ビュー (データソース 1)	[データソース 1] でデータソースやターゲットとなるテーブルを設定します。
	DB ブリッジ	異なるデータソース間で ETL を設定する場合は、DB ブリッジを介します。
	テーブル・ビュー (データソース 2)	[データソース 2] でデータソースやターゲットとなるテーブルを設定します。
	csv	データソースとなる csv ファイルを設定します。
	カスタム スクリプト	SQL やシェルを設定します。
	Extract	データの抽出を行います。
	Transform	データを変換します。

アイコン	アイコン名	説明
	Load	データソースからターゲットへのロードを設定します。
	ルックアップ	データのクレンジングや置換えを設定します。
	DimensionLoad	ディメンジョン（次元）へのロードを設定します。
	FactLoad	ファクトへのロードを設定します。
	削除	選択したオブジェクトや、マッピング線を削除します。

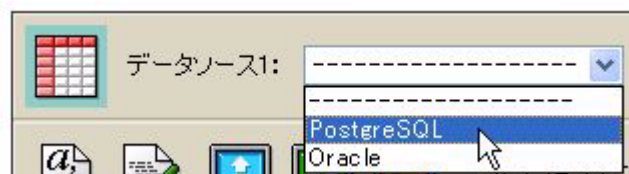
4.2.1 ステップ1：ソースとターゲットを登録する

ソースとしてテーブル、ビュー、csv ファイルを、ターゲットとしてテーブルを設定できます。

4.2.1.1 テーブル・ビューを登録する

1. オブジェクトツリーの [ジョブ] - [ETL ジョブ] を選択して、[ETL ジョブ] 登録画面を表示します。

2. [データソース 1] を選択します。



注記： データソースの変更

データソースの構成が等しい場合に、データソースの変更ができます。変更前データソースと変更後データソースの名前だけが異なる場合、変更が有効です。全く異なるデータソースに変更した場合は、ジョブ設定が正常でなくなることがあります。

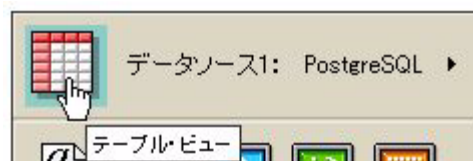
- ①データソース名の横の三角アイコンをクリックすると、[変更] ボタンが表示されます。



- ② [変更] ボタンをクリックすると警告メッセージが表示されます。



- ③変更する場合は [OK] ボタンをクリックして、データソースを選択します。変更しない場合は [キャンセル] ボタンをクリックして、[ETL ジョブ登録] 画面に戻ります。
3. [データソース 1] の [テーブル] アイコンをクリックし、ETL 設定エリアに [テーブル・ビュー] オブジェクトを表示します。



4. [テーブル・ビュー] オブジェクトをダブルクリックして [テーブル・ビュー] 登録画面を表示します。

5. 以下の情報を入力します。

オブジェクトID	1	オブジェクト名	public.item
コメント	ソーステーブルは商品テーブル		

プレビュー

No	カラム	データ型	Primary Key	NOT NULL制約
1	item_id	varchar(8)	Primary Key	NOT NULL
2	short_name	varchar(20)		
3	long_name	varchar(50)		
4	amount	numeric(10,0)		
5	fixed_price	numeric(10,0)		
6	cost_price	numeric(12,2)		
7	selling_price	numeric(10,0)		
8	sub_category_id	varchar(8)		

項目	説明
オブジェクト ID	テーブル・ビューのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。
オブジェクト名	テーブルまたはビューの名前を選択します。 * テーブル名（ビュー名）を選択すると、画面下にテーブル（ビュー）構成が表示されます。
コメント	ソースとなるテーブル・ビューに対するコメントを入力します。

メモ： プレビューボタン

- ① [プレビュー] ボタンをクリックして、選択したテーブルまたはビューのレコードを 100 件表示します。
- ②レコードを 100 件だけ表示する場合は [先頭 100 件表示] ボタンをクリックします。
- ③レコードを全件表示する場合は [全件] ボタンをクリックしま

す。レコードを 100 件表示ずつ表示します。

④ ページアンカーをクリックして、見たいデータを表示します。

⑤ 任意の列を昇順にしたい場合は、列名をクリックします。再度クリックすると、降順になります。

先頭100件表示 全件表示

item_id	short_name	long_name	amount	fixed_price	cost_price	selling_price	sub_category_id
00000023	Turbo Linux	Turbo Linux	1	10000	7000.00	8000	00000006
00000011	Envoy スタンド	Envoy スタンド	1	280000	196000.00	224000	00000004
00000012	Envoy エグゼクティ	Envoy エグゼクティ	1	320000	224000.00	256000	00000004
00000013	Envoy アンバサダー	Envoy アンバサダー	1	480000	336000.00	384000	00000004
00000014	Sentinel スタンド	Sentinel スタンド	1	220000	154000.00	176000	00000005
00000015	Sentinel ファイナ	Sentinel ファイナ	1	260000	182000.00	208000	00000005
00000016	Sentinel マルチメ	Sentinel マルチメ	1	250000	175000.00	200000	00000005
00000017	OSドキュメント	OSドキュメント	1	25000	17500.00	20000	00000006
00000018	ノートPC キー	ノートPC キー	1	42000	29400.00	33600	00000007
00000021	Envoy 内蔵キ	Envoy 内蔵キ	1	59000	41300.00	47200	00000007
00000022	外付け 101キー	外付け 101キー	1	33000	23100.00	26400	00000007
00000023	PCMCIAモデム/F	PCMCIA モデム/F	1	200000	140000.00	160000	00000009
00000024	256MBPCMCIAIL	SIMM-256MBPC	1	20600	14420.00	16480	00000010
00000025	512MBPCMCIAIL	SIMM-512MBPC	1	33300	23310.00	26640	00000010
00000026	マルチメディア	マルチメディア	1	5600	3920.00	4480	00000007
00000027	WindowsXP 1ユ	WindowsXP 1ユ	1	8500	5950.00	6800	00000006
00000028	WindowsXP 5ユ	WindowsXP 5ユ	1	28000	19600.00	22400	00000006
00000029	マウス・パッド	マウス・パッド	1	1000	700.00	800	00000007
00000030	288MB 3.5 FD	288MB 3.5 FD	1	4000	2800.00	3200	00000007

public.item: 23件

6. [更新] ボタンをクリックしてテーブル・ビューの情報を登録します。

メモ: テーブル・ビューオブジェクト名

テーブル・ビューオブジェクトの名称が、選択したテーブル名またはビュー名に変更されます。

public.item



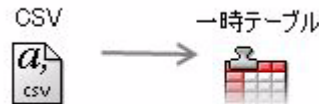
4.2.1.2 CSV を登録する

1. オブジェクトツリーの [ジョブ] - [ETL ジョブ] を選択して、[ETL ジョブ登録] 画面を表示します。

2. [csv] アイコンをクリックし、ETL 設定エリアに [csv] オブジェクトを表示します。



メモ： [csv] オブジェクトと一緒に [一時テーブル] オブジェクトが表示されます。



3. [csv] オブジェクトをダブルクリックして、[csv] 登録画面を表示します。
4. 以下の情報を入力します。

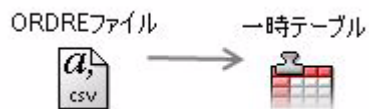
オブジェクトID	2
オブジェクト名	ORDREファイル
コメント	ORDERファイル取り込み
文字コード	Shift_JIS
区切り文字	<input checked="" type="radio"/> カンマ <input type="radio"/> タブ <input type="radio"/> クォート "
タイトル行位置	タイトル行: <input checked="" type="radio"/> あり <input type="radio"/> なし スキップ行数: 3
ファイルパス	/tmp/data1.csv タイトル行を取得 プレビュー 取り込み項目 <input checked="" type="checkbox"/> 1 オーダーID <input checked="" type="checkbox"/> 2 顧客ID <input checked="" type="checkbox"/> 3 商品ID <input checked="" type="checkbox"/> 4 数量 <input checked="" type="checkbox"/> 5 購入金額
最大取得件数	10000

項目	説明
オブジェクト ID	csv のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。
オブジェクト名	csv 名を入力します。
コメント	csv に対するコメントを入力します。
文字コード	ソースとなる csv ファイルの文字コードを選択してください。
区切り文字	csv ファイルで使用している区切り文字を選択してください。
クォート	csv ファイルで使用しているクォートを選択してください。

項目	説明
タイトル行位置	<p>[タイトル行]</p> <p>タイトル行が有る場合は「あり」を、無い場合は「なし」を選択してください。</p> <p>[スキップ行数]</p> <p>スキップする行数を入力してください。</p> <p>例) タイトルの上にコメント行が 3 行ある場合→「3」と入力</p>
ファイルパス	<p>csv ファイルの保存先を絶対パスで入力します。</p> <p>[タイトル行を取得] ボタンをクリックして、[取り込み項目] 欄に項目をセットします。</p> <p>*[取り込み項目] 欄にセットされたチェックボックスの初期表示は、全てチェックオンです。取り込む必要がない項目は、チェックボックスをオフにします。</p>
最大取得件数	<p>取得件数を制限する場合は、取得するデータの最大件数を入力してください。</p>

5. [更新] ボタンをクリックして、csv の情報を保存します。

メモ： csv オブジェクト名
 csv オブジェクトの名称が、入力したオブジェクト名に変更されます。



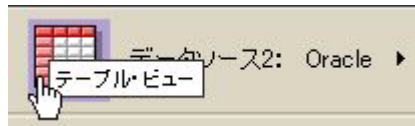
4.2.1.3 ターゲットを登録する

ターゲットとなるテーブルを登録します。ターゲットに設定できるのはテーブルだけです。

1. [データソース 2] を選択します。



2. [データソース 2] の [テーブル] アイコンをクリックし、ETL 設定エリアに [テーブル・ビュー] オブジェクトを表示します。



注記： データソースの指定について

1 つのデータソース内で ETL 処理 (Extract ~ Load) を行なう場合は、DB ブリッジは必要ないので [データソース 2] は選択しません。ソースを登録する時に設定した [データソース 1] が、ターゲットのデータソースとなります。

メモ： テーブル・ビューオブジェクトの色について

[データソース 1] に属する [テーブル・ビュー] オブジェクトと [データソース 2] に属する [テーブル・ビュー] オブジェクトは、オブジェクトの色が異なります。

3. [テーブル・ビュー] オブジェクトをダブルクリックして [テーブル・ビュー] 登録画面を表示します。
4. 以下の情報を入力します。

項目	説明
オブジェクト ID	テーブルのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	テーブル名を選択します。 * テーブル名を選択すると、画面下にテーブル構成が表示されます。
コメント	ターゲットテーブルに対するコメントを入力します。

5. [更新] ボタンをクリックして、テーブルの情報を保存します。

メモ： プレビューボタン

- ① [プレビュー] ボタンをクリックして、選択したテーブルまたはビューのレコードを表示します。
- ② レコードを 100 件だけ表示する場合は [100 件] ボタンをクリックします。
- ③ レコードを全件表示する場合は [全件] ボタンをクリックします。
- ④ 任意の列を昇順にしたい場合は、列名をクリックします。再度クリックすると、降順になります。

6. ターゲットオブジェクト名
テーブルオブジェクトの名称が、選択したテーブル名に変更されます。

4.2.2 ステップ2 : DBブリッジを登録する

異なるデータソース間で ETL を行なう場合は、DB ブリッジを介してデータのロードを行ないます。一つのデータソース内で ETL を行なう場合は、DB ブリッジは必要ありません。

1. [DB ブリッジ] アイコンをクリックし、ETL 設定エリアに [DB ブリッジ] オブジェクトを表示します。



メモ： [DB ブリッジ] オブジェクトと一緒に [一時テーブル] オブジェクトが表示されます。

2. ソースオブジェクトから [DB ブリッジ] オブジェクトへドラッグして、オブジェクトをマッピングします。

注記： ソースが複数の場合

ソースオブジェクト : DB ブリッジオブジェクト = 1:1 になるようにマッピングします。

例) ソースが「EMP」テーブルと「BONUS」テーブルの場合
「EMP」テーブルオブジェクトと EMP 用 [DB ブリッジ] オブジェクトをマッピングし、「BONUS」テーブルオブジェクトと BONUS 用 [DB ブリッジ] オブジェクトをマッピングします。

メモ： [一時テーブル] オブジェクトは [データソース 2] に属することになるため、ソースオブジェクトとオブジェクトの色が異なります。



3. [DB ブリッジ] オブジェクトをダブルクリックして [DB ブリッジ] 登録画面を表示します。
4. 以下の情報を入力します。

オブジェクトID	4	オブジェクト名	PostgreSQLとOracle
コメント	PostgreSQL-Oracle		
フェッチ件数	<input type="text"/>		
コミット件数	100		
最大取得件数	<input type="text"/>		

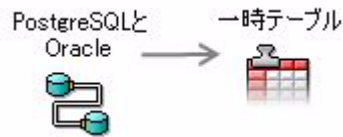
項目	説明
オブジェクト ID	DB ブリッジのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	DB ブリッジ名を入力します。
コメント	DB ブリッジに対するコメントを入力します。

項目	説明
最大取得件数	件数を制限する場合は、取得するデータの最大件数を入力してください。
フェッチ件数	フェッチ単位となる件数を入力します。
コミット件数	コミット単位となる件数を入力します。

5. [更新] ボタンをクリックして、DB ブリッジの情報を保存します。

メモ： DB ブリッジ名

DB ブリッジオブジェクトの名称が、入力したオブジェクト名に変更されます。



4.2.3 ステップ3 : Extract を登録する

4.2.3.1 ソースが一つの場合

1. [Extract] アイコンをクリックし、ETL 設定エリアに [Extract] オブジェクトを表示します。



2. DB ブリッジの [一時テーブル] オブジェクトから [Extract] オブジェクトへドラッグして、オブジェクトをマッピングします。



注記： [Extract] オブジェクトにマッピングできるのは以下のテーブルオブジェクトだけです。



- [テーブル・ビュー] オブジェクト
- csv の [一時テーブル] オブジェクト
- DB ブリッジの [一時テーブル] オブジェクト

メモ： オブジェクトのマッピング

①ソースオブジェクトにカーソルを移動し、マウスポインタが「+」になったところで、[Extract] オブジェクトへドラッグします。

- ② [Extract] オブジェクト上でマウスポインタが [+] になったらカーソルを離します。
- ③ ソースオブジェクトと [Extract] オブジェクトがマッピングされます。

メモ： オブジェクトのドラッグ

- ① オブジェクト名の上にカーソルを移動すると、マウスポインタが  に変わります。
 - ② ポインタが  に変わったところでクリックして、オブジェクトをドラッグできます。
3. [Extract] オブジェクトをダブルクリックし、[Extract] 登録画面を表示します。
 4. 以下の情報を入力します。

オブジェクトID	6	オブジェクト名	商品テーブルを抽出
コメント	itemテーブルをExtract		
最大取得件数	<input type="text"/>		

項目	説明
オブジェクト ID	Extract のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	Extract 名を入力します。
コメント	Extract に対するコメントを入力します。
最大取得件数	抽出するレコード件数を制限したいときに入力します。

5. ソースオブジェクト内の抽出したい項目名をダブルクリックします。

public.item	
item_id	+
short_name	
long_name	

メモ： ソースオブジェクト名をダブルクリックすると、すべての項目が選択項目としてセットされます。

6. 同様に [列の追加] ボタンをクリックして、列を追加することができます。



7. 以下の情報を入力します。

項目名	item_id	sub_category_id	short_name	amount	fixed_price	cost_price
SELECT	public.item.item_id	public.item.sub_category_id	public.item.short_name	public.item.amount	public.item.fixed_price	public.item.cost
並び替え	Asc ▼	Asc ▼	Asc ▼	▼	▼	▼
集計	集計キー ▼	集計キー ▼	集計キー ▼	Sum ▼	Sum ▼	Sum ▼
表示	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
WHERE		= '000000044'				
OR		= '000000077'				

項目	説明
項目名	列の別名を入力してください。 * 列名が初期表示されています。
SELECT	列を選択するか、直接入力してください。
並び替え	Asc、Desc を選択してください。
集計	データを集計する場合、集計キーまたは集計関数を選択してください。 例) GROUP BY ID, NAME としたい場合 → 「ID」列、「NAME」列にそれぞれ「集計キー」をセット 例) SUM(salse) としたい場合→ 「sales」列に「SUM」をセット
表示	列を SELECT 句に加える場合は、チェックをオンにしてください。SELECT 句に加えない場合は、チェックをオフにしてください。
WHERE	WHERE 句を入力してください 例) Where sales > 1000000 としたい場合 → sales 列に「> 1000000」と入力 例) where name like 'A%' としたい場合 → name 列に「like 'A%」と入力
OR	WHERE 句の OR 条件を入力してください (5 個まで)。 例) WHERE ID = 'CC2' OR ID = 'CC4' としたい場合 → ID 列に「'CC4'」と入力。([WHERE] に「= 'CC2'」と入力)

8. 入力した列の情報をコピーする場合は、コピーしたい列を選択してから [列のコピー] ボタンをクリックします。



- メモ： 列の選択
[項目名] の上にあるバーをクリックします。列のセルの色がすべて反転して、列が選択されます。



9. 列を削除する場合は、削除したい列を選択してから [列の削除] ボタンをクリックします。

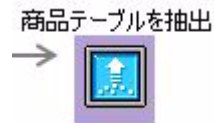


- メモ： 列を全て削除
全ての列を削除する場合は [列を全て削除] ボタンをクリックします。



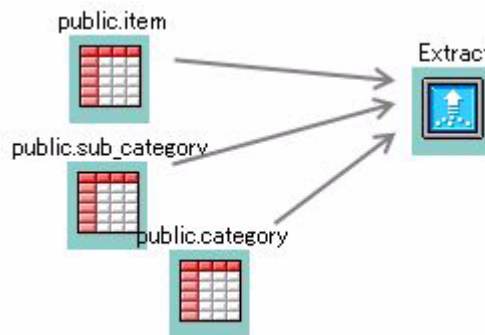
10. [更新] ボタンをクリックして、Extract の情報を保存します。

メモ： Extract オブジェクト名
Extract オブジェクトの名称が、入力したオブジェクト名に変更されます。



4.2.3.2 ソースが複数の場合

1. [Extract] アイコンをクリックし、ETL 設定エリアに [Extract] オブジェクトを表示します。
2. 各ソースオブジェクトから [Extract] オブジェクトへドラッグして、オブジェクトをマッピングします。



3. [Extract] オブジェクトをダブルクリックし、[Extract] 登録画面を表示します。

Extract

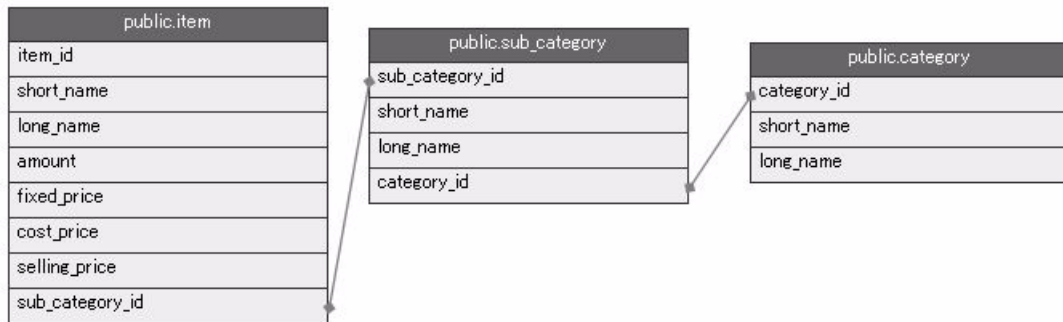
オブジェクトID	7	オブジェクト名	Extract
コメント			
最大取得件数			

リレーションの削除

public.item	public.sub_category	public.category
item_id	sub_category_id	category_id
short_name	short_name	short_name
long_name	long_name	long_name
amount		
fixed_price		
cost_price		
selling_price		
sub_category_id		

4. 「item」の「sub_category_id」列から「sub_category」テーブルの「sub_category_id」列へドラッグしてソースオブジェクトを結合します。同じく「sub_category」テーブルの「category_id」列から「category」

テーブルの「category_id」列へドラッグしてソースオブジェクトを結合します。



5. 外部結合にする場合は、リレーション線を右クリックして、リレーションのメニューを表示します。1:1、1:n、n:1 の内、該当するリレーションをクリックします。

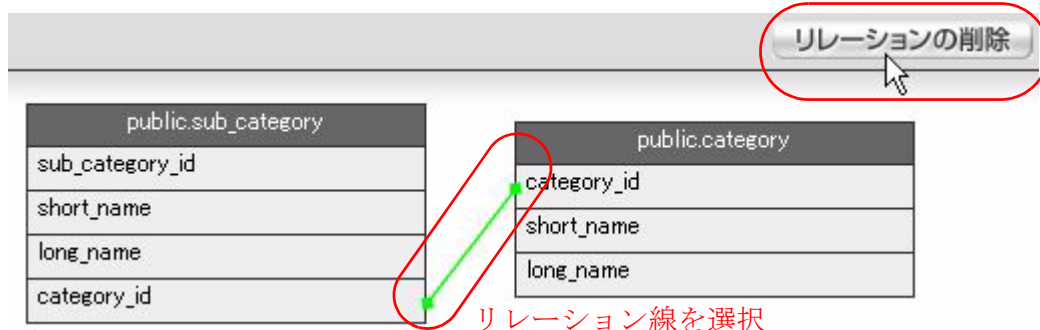


メモ： リレーション変更

リレーションの種類により、線のレイアウトが変更されます。

- 1:1 の場合「—」
- 1:n の場合「→」
- n:1 の場合「←」

6. ソースのリレーションを削除する場合は、削除したい線をクリックしてから [リレーションの削除] ボタンをクリックします。



7. Extract 情報を入力します。
8. [更新] ボタンをクリックして、Extract の情報を保存します。

メモ： Extract オブジェクト名

Extract オブジェクトの名称が、入力したオブジェクト名に変更されます。

4.2.4 ステップ4 : Transform を登録する

1. [Transform] ボタンをクリックし、ETL 設定エリアに [Transform] オブジェクトを表示します。



2. [Extract] オブジェクトから [Transform] オブジェクトへドラッグして、オブジェクトをマッピングします。

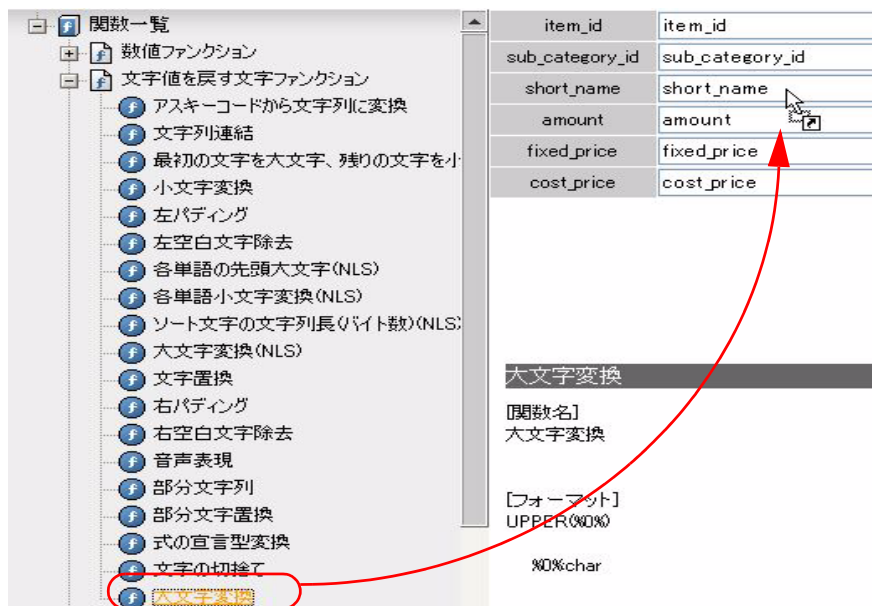


3. [Transform] オブジェクトをダブルクリックし、[Transform] 登録画面を表示します。
4. 以下の情報を入力します。

オブジェクトID	8	オブジェクト名	大文字変換
コメント	short_nameを大文字に変換する		

項目	説明
オブジェクト ID	Transform のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	Transform の名前を入力します。
コメント	Transform に対するコメントを入力します。

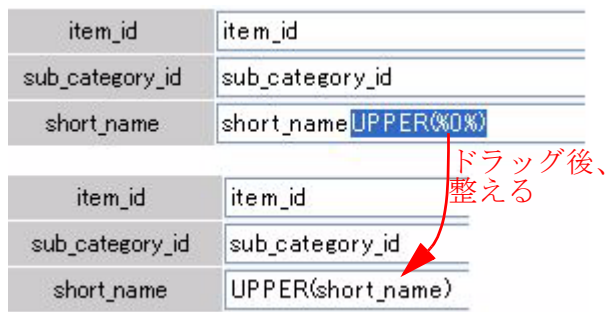
5. 関数ツリーから関数を選択し、関数を設定したい項目名へドラッグします。



メモ： テキストボックスに関数を直接入力することもできます。

メモ： 関数名をクリックすると、右下のエリアに関数の説明が表示されます。

6. テキストボックスにセットした関数を「Upper(short_name)」のように整えます。



7. 値を代入する場合は、テキストボックスに値を入力します。

メモ： sales に 100 を代入したい場合、sales のテキストボックスに「100」と入力します。

8. [更新] ボタンをクリックして Transform の情報を登録します。

メモ： Transform 名
Transform オブジェクトの名称が、入力したオブジェクト名に変更されます。

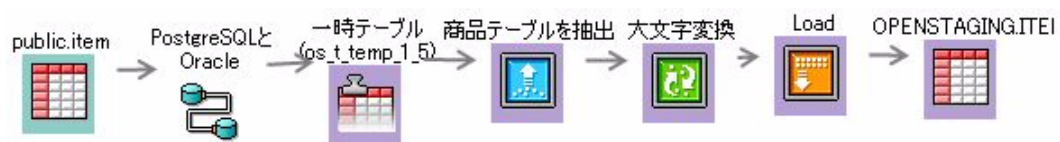


4.2.5 ステップ5 : Load を登録する

1. [Load] アイコンをクリックし、ETL 設定エリアに [Load] オブジェクトを表示します。



2. [Transform] オブジェクトから [Load] オブジェクトへドラッグして、オブジェクトをマッピングします。
3. [Load] オブジェクトからターゲットオブジェクトへドラッグして、オブジェクトをマッピングします。



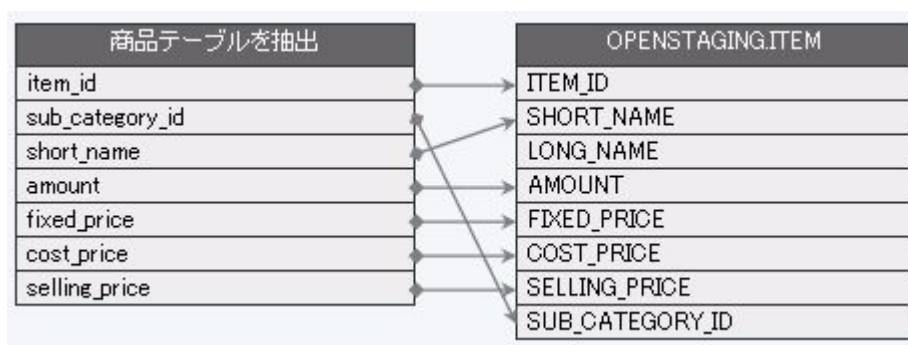
4. [Load] オブジェクトをダブルクリックして [Load] 登録画面を表示します。
5. 以下の情報を入力します。

オブジェクトID	10	オブジェクト名	商品テーブルにロード
コメント	PotgreSQLの商品テーブルからOracleの商品テーブルにロード		
データロード方法	Truncate/Insert <input type="checkbox"/> 差分更新を行う		
自動キャスト	<input checked="" type="checkbox"/> ロード時に自動キャストを行う		
コミット件数	100		
ロードエラー判断件数	1	エラー発生時	<input checked="" type="radio"/> 処理を中断する(異常終了) <input type="radio"/> 後続フロー処理を続行する

項目	説明
オブジェクト ID	Load のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	Load の名前を入力します。
コメント	Load に対するコメントを入力します。
データロード方法	データロード方法を選択します。 [差分更新を行なう] 差分更新を行なう場合はチェックをオンにします。 * 差分更新 ターゲットの既存データの内、Insert も Update もされなかったレコードは、ロード処理の最後に削除します。

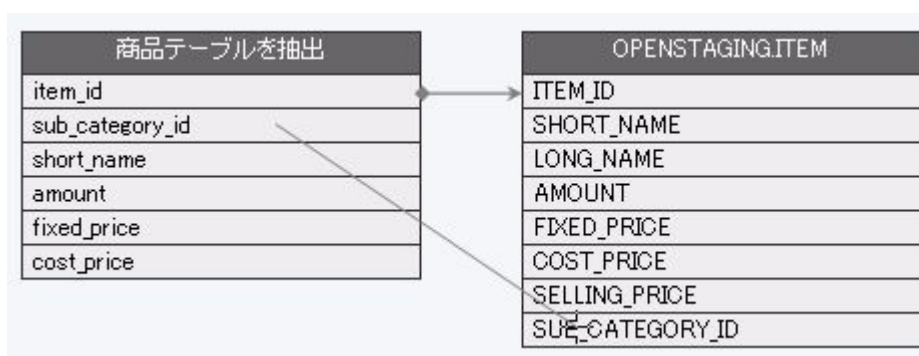
項目	説明
自動キャスト	ターゲットのデータ型に合わせてソースデータをロードする場合は、チェックをオンにします。 *ただし、パフォーマンスが若干低下します。
コミット件数	コミット単位となる件数を入力します。
ロードエラー判断件数	ロードエラーと判断するレコード件数を入力します。 例) エラーレコード 100 件でエラーとする場合 → 「100」と入力
エラー発生時	[処理を中断する (異常終了)] ロードエラーが発生したら処理を終了する場合に選択します。 [後続フロー処理を続行する] ロードエラーが発生しても処理を続ける場合に選択します。

6. ソースの列「item_id」からターゲットの列「ITEM_ID」へドラッグしてマッピングします。同じく、「sub_category_id」「short_name」、「amount」、「fixed_price」、「cost_price」もターゲットとマッピングします。



メモ： マッピング方法

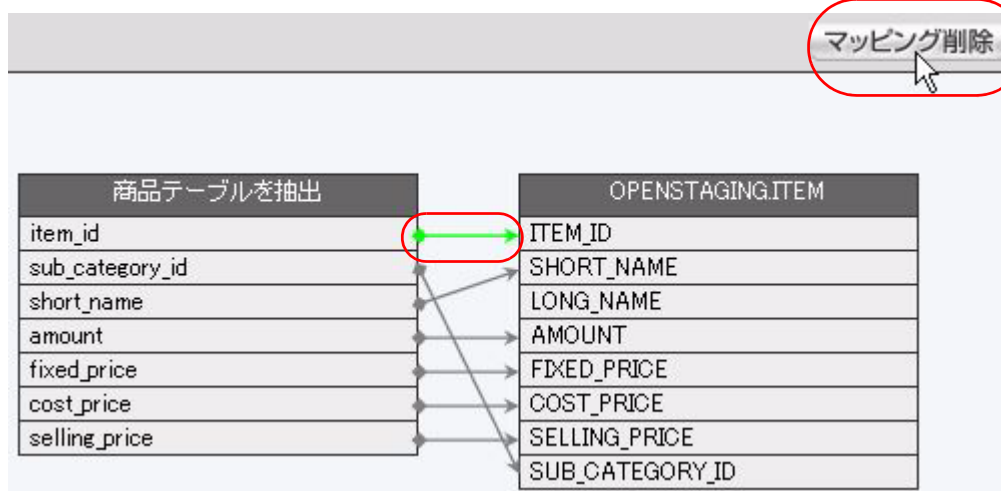
- ①ソースの「sub_category_id」上でマウスポインタが「+」になったら、ターゲットの「SUB_CATEGORY_ID」へドラッグします。



- ②ターゲットの「SUB_CATEGORY_ID」上でマウスポインタが「+」になったらカーソルを離します。

- ③ソース「sub_category_id」とターゲット「SUB_CATEGORY_ID」がマッピングされます。

7. マッピングを削除する場合は、削除したい線を選択してから [マッピングの削除] ボタンをクリックします。



メモ： 1:n のマッピングについて
 ソースからターゲットへのマッピングは、1:1 の他に 1:n にマッピングすることができます。例えばソース [short_name] 列をターゲットの [SHORT_NAME] 列、[LONG_NAME] 列にマッピングすることができます。



8. [更新] ボタンをクリックして Load の情報を保存します。

メモ： Load オブジェクト名
 Load オブジェクトの名称が、入力したオブジェクト名に変更されます。



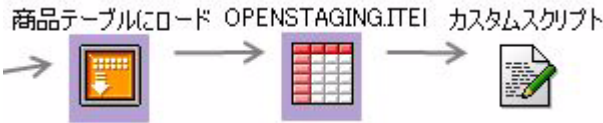
4.3 カスタムスクリプトを登録する

カスタムスクリプトを追加することによって、Analyze 文を実行したり、トリガーファイルを生成したりすることができます。

1. [ETL ジョブ登録] 画面でソース、Load、ターゲットの順にオブジェクトを登録して、マッピングします。
2. [カスタムスクリプト] アイコンをクリックし、ETL 設定エリアに [カスタムスクリプト] オブジェクトを表示します。



3. ターゲット [テーブル・ビュー] オブジェクトから [カスタムスクリプト] オブジェクトへドラッグして、オブジェクトをマッピングします。



4. 以下の情報を入力します。

オブジェクトID	11	オブジェクト名	<input type="text" value="ANALYZE文実行"/>
コメント	<input type="text"/>		
スクリプトタイプ	<input checked="" type="radio"/> SQL <input type="radio"/> シェル		
スクリプト	<input type="radio"/> スクリプトファイルを指定する		
	<input checked="" type="radio"/> スクリプトを直接入力する		

```
analyze table openstaging.item compute statistics;
analyze table openstaging.sub_category compute statistics;
```

項目	説明
オブジェクト ID	カスタムスクリプトのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。 * 変更はできません。
オブジェクト名	カスタムスクリプトの名前を入力します。
コメント	カスタムスクリプトに対するコメントを入力します。

項目	説明
スクリプトタイプ	<p>スクリプトの種類を選択します。</p> <p>[SQL]</p> <p>SQL のラジオボタンをオンにして、スクリプトを実行する際のデータソースを選択します。</p> <p>例) 「データソース 1」 の「product」 テーブルに対してスクリプトを実行する場合→ 「データソース 1」 を選択</p> <p>[シェル]</p> <p>シェルを実行する場合は、シェルのラジオボタンをオンにします。</p>
スクリプト	<p>[スクリプトファイルを指定する]</p> <p>スクリプトをファイルから実行する場合、チェックをオンにします。指定するスクリプトファイルを、絶対パスで入力します。</p> <p>[スクリプトを直接入力する]</p> <p>チェックをオンにして、スクリプトを直接入力します。</p> <p>○ SQL</p> <p>ジョブの最後の処理で Oracle テーブルに対して Analyze 文を実行し、検索時のパフォーマンスを向上させる場合→</p> <p>「analyze table public.sales compute statistics」 と入力</p> <p>*[ETL ジョブ登録] 画面の [処理日] を where 句に指定することができます。</p> <p>例) [処理日] より前のデータを削除する場合→</p> <p>「delete from sales where time < to_date(os_pv_process_date, 'YYYYMMDD')」</p> <p>* スクリプトを連続して入力することもできます。</p> <p>「analyze table public. compute statistics; analyze table public.sales compute statistics;」</p> <p>○ シェル</p> <p>トリガーファイルを生成する場合→ 「mkfile 0.dat」 と入力</p>

5. [更新] ボタンをクリックして、カスタムスクリプトの情報を登録します。

メモ： カスタムスクリプトオブジェクト名
カスタムスクリプトオブジェクトの名称が、入力したオブジェクト名に変更されます。

ANALYZE文実行



4.4 XML を編集する

ETL ジョブの設定 XML ファイルを編集することができます。

- オブジェクトツリーの [ジョブ] - [ETL ジョブ] から XML を編集したいジョブを選択して、[ETL ジョブ登録] 画面を表示します。
メモ： ETL ジョブを新規作成するときも、XML ファイルを編集できます。
- [XML] ボタンをクリックして、[XML] 画面を表示します。

XMLを確認

- XML を編集する場合は [カスタマイズを行なう] チェックボックスをオンにして、テキストエリアを入力可能にします。

チェックオンで入力可能になる

- テキストエリアに表示されている XML を変更します。
- 変更内容を元に戻す場合は、[リセット] ボタンをクリックします。

リセット

- [更新] ボタンをクリックして、XML の情報を保存します。

注記： XML を編集することによってジョブが壊れた場合は、ジョブを作成し直す必要があります。

4.5 テスト実行をする

ETL ジョブを登録後、ETL ジョブをテスト実行することができます。

- オブジェクトツリーの [ジョブ] - [ETL ジョブ] からテスト実行したいジョブを選択して、[ETL ジョブ登録] 画面を表示します。
- [テスト実行] ボタンをクリックして [テスト実行] 画面を表示します。

テスト実行

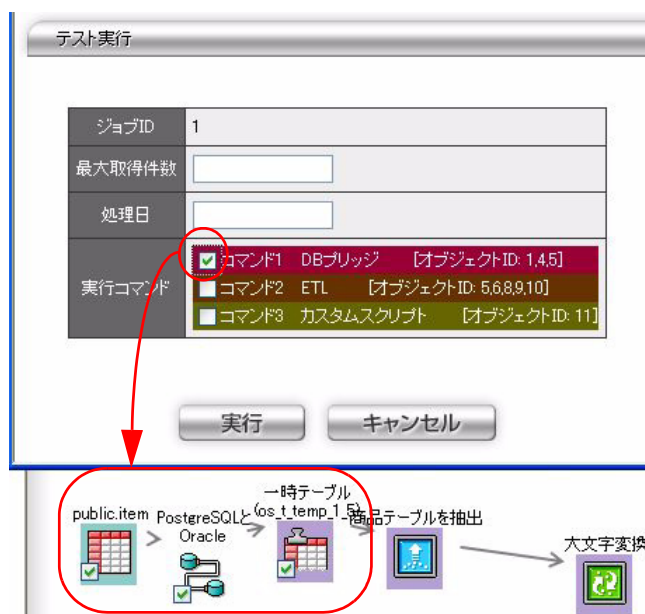
注記： ETL ジョブの新規作成時は、[テスト実行] ボタンは表示されません。

3. 以下の情報を入力します。

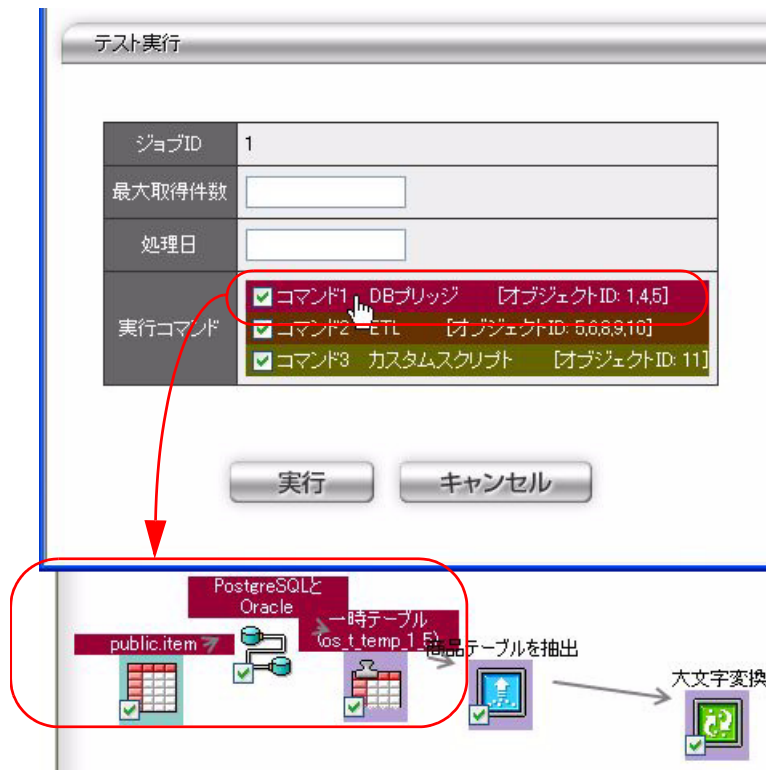
ジョブID	1
最大取得件数	<input type="text"/>
処理日	<input type="text"/>
実行コマンド	<input checked="" type="checkbox"/> コマンド1 DBブリッジ [オブジェクトID: 1,4,5] <input checked="" type="checkbox"/> コマンド2 ETL [オブジェクトID: 5,6,8,9,10] <input checked="" type="checkbox"/> コマンド3 カスタムスクリプト [オブジェクトID: 11]

項目	説明
ジョブ ID	[ETL ジョブ登録] 画面と同じ、ETL ジョブ ID が表示されます * ジョブ ID は変更できません。
最大取得件数	処理件数を制限する場合に入力します。
処理日	ETL ジョブの実行日に依存しない日付が必要な場合、[yyyy/mm/dd] の書式で入力します。日付はアプリケーションサーバーのシステム日付です。詳細は 『4.1 ETL ジョブを登録する』 の [処理日] を参照してください。 * テスト実行で入力した [処理日] は、[ETL ジョブ登録] 画面の [処理日] に反映されません。
実行コマンド	実行したいコマンドを選択します。 * () 内に、各コマンドに属するオブジェクトが表示されています。 * 作成した ETL ジョブによって、コマンド数は異なります。

メモ： 実行コマンドの「コマンド1」チェックボックスをオンにした場合、[ETL ジョブ登録] 画面で「実行コマンド1」に属するオブジェクトのチェックボックスがオンになります。オフにした場合は、「ETL ジョブ登録」画面のオブジェクトのチェックボックスもオフになります。



- 対象オブジェクト ID にカーソルを移動した場合
[ETL ジョブ登録] 画面の ETL オブジェクトの内、コマンドの対象となっているオブジェクトのオブジェクト名が実行コマンドと同じ背景色になります。



- [テスト実行] ボタンをクリックして、ETL ジョブをテスト実行します。
- ETL ジョブのテスト実行が開始されると、[モニタリング] 画面が表示されます。

4.6 モニタリング画面を表示する

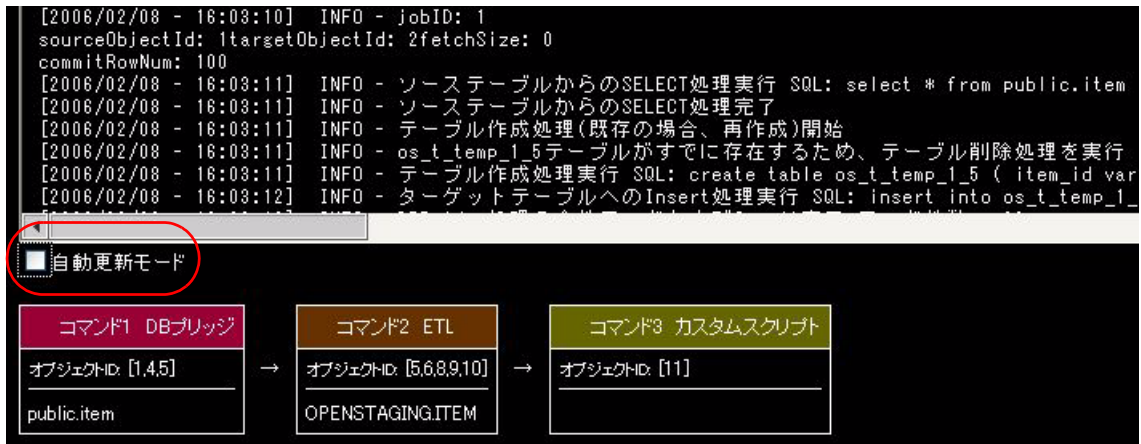
実行されているジョブの進行状況を、モニタリングすることができます。テスト実行中の自動表示以外に、スケジュールなどから実行したジョブについても進行状況を確認することができます。

- オブジェクトツリーの [ジョブ] - [ETL ジョブ] からモニタリングしたいジョブを選択して、[ETL ジョブ登録] 画面を表示します。
- [モニタリング] ボタンをクリックして、[モニタリング] 画面を表示します。

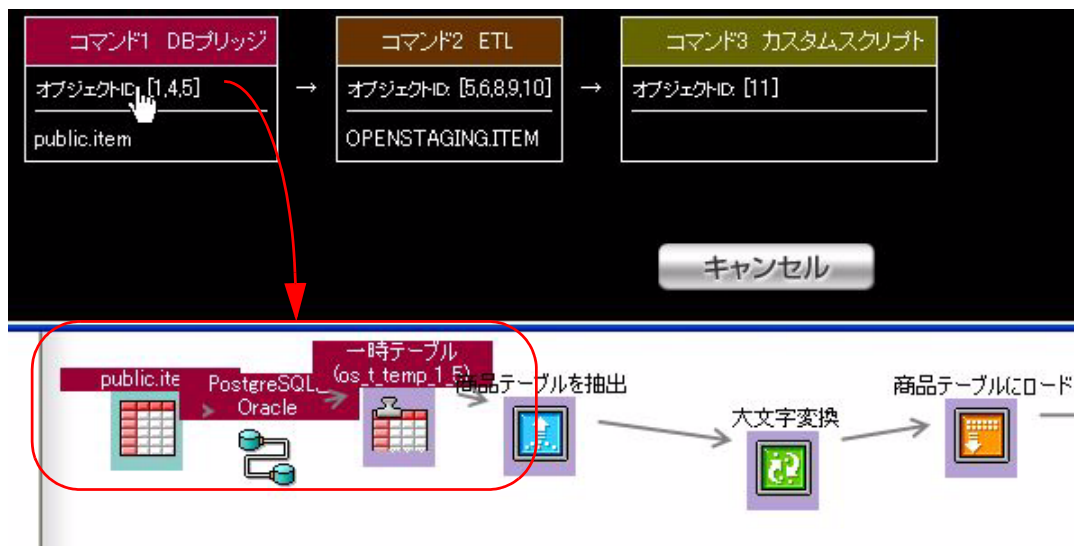


メモ： ETL ジョブの新規作成時は、[モニタリング] ボタンは表示されません。

- ETL ジョブの実行ログを自動更新しない場合は、[自動更新モード] をチェックオフにします。



- モニタリング画面の「コマンド 1」にカーソルを移動すると、コマンド 1 に属する ETL オブジェクト ([ETL ジョブ登録] 画面) のオブジェクト名が実行コマンドと同じ背景色になります。



- 「コマンド 1」のテーブル名「public.item」をクリックすると、item テーブルがプレビューされます。プレビュー画面については [『4.2 ETL オブジェクトを設定する』](#) の「プレビューボタン」を参照してください。



- [キャンセル] ボタンをクリックして、[モニタリング] 画面を閉じます。

4.7 オプションメニューを利用する

マスタテーブルを参照して不一致データを除外 / 置換するルックアップ機能や、データウェアハウスの履歴管理に対応した DimensionLoad、FactLoad を利用することができます。

4.7.1 ルックアップを登録する

マスタテーブルを参照して、不一致データはロードしない（クレンジング）、または他のデータに置換えるよう設定することができます。

1. [ETL ジョブ登録] 画面の ETL 設定エリアにテーブルオブジェクトと Extract オブジェクトをそれぞれ登録の上、マッピングします。

注記： ルックアップのマッピング

- ・ Extract、Transform、Load の ETL 内でマッピングすることができます。
 - ・ テーブルと直接マッピングできます。このときルックアップの次に設定できるオブジェクトは [Load] オブジェクトです。
2. オプションメニューの [ルックアップ] アイコンをクリックし ETL 設定エリアに [ルックアップ] オブジェクトを表示します。



3. [Extract] オブジェクトから [ルックアップ] オブジェクトへドラッグして、オブジェクトをマッピングします。



4. [ルックアップ] オブジェクトをダブルクリックして、[ルックアップ] 登録画面を表示します。

5. 以下の情報を入力してください。

オブジェクトID	3	オブジェクト名	商品クレンジング
コメント			
実行する/しない	<input checked="" type="checkbox"/> このルックアップ処理を実行する		
ルックアップ対象	ルックアップテーブル public.item_lookup_target ▼		
	項目名	ルックアップカラム	置換データ
	<input type="checkbox"/> item_key	----- ▼	
	<input checked="" type="checkbox"/> item_id	item_id ▼	
	<input type="checkbox"/> item_short_name	----- ▼	
	<input type="checkbox"/> item_long_name	----- ▼	
	<input checked="" type="checkbox"/> sub_category_id	sub_category_id ▼	
	<input type="checkbox"/> sub_category_short_name	----- ▼	
	<input type="checkbox"/> sub_category_long_name	----- ▼	
	<input type="checkbox"/> category_id	----- ▼	
マスタテーブルに存在しないレコード	<input checked="" type="radio"/> 除外する <input checked="" type="checkbox"/> 除外されたレコードをテーブルに格納する public.item_lookup_error ▼ <input checked="" type="checkbox"/> 除外されたレコードをファイルに出力する /user01/item_lookup_error.csv <input checked="" type="radio"/> 追記 <input type="radio"/> 置換 <input type="radio"/> 指定のデータに置き換える		

項目	説明
オブジェクト ID	ルックアップのオブジェクト ID を、OpenStaging が自動的に割り振って表示します。
オブジェクト名	ルックアップ名を入力します。
コメント	ルックアップに対するコメントを入力します。
実行する / しない	ETL ジョブ実行時にルックアップを行なう場合は、チェックをオンにします。

項目	説明
ルックアップ対象	<p>[ルックアップテーブル]</p> <p>参照先のマスタテーブルを選択します。</p> <p>[項目名]</p> <p>[ルックアップテーブル] と比較する項目を、チェックオンにします。</p> <p>[ルックアップカラム]</p> <p>[項目名] の参照元となるカラムを選択します。</p> <p>[置換データ]</p> <p>不一致データを置換する場合、置換データを入力します。</p> <p>*[ルックアップテーブルに存在しないレコード] の [指定のデータに置き換える] がチェックオンのときだけ、入力できます。</p>
ルックアップテーブルに存在しないレコード	<p>参照先マスタテーブルとの不一致レコードがあった場合に、除外するか、指定のデータに置き換えるか選択します。</p> <p>○ [除外する] チェックオンの場合</p> <ul style="list-style-type: none"> ・ [除外されたレコードをテーブルに格納する] <p>除外レコードを格納するためのテーブルを選択します。</p> <p>* 除外レコードを格納するテーブルは、ソーステーブルと (Extract 設定時は Extract と) 同じテーブル構成である必要があります。</p> <ul style="list-style-type: none"> ・ [除外されたレコードをファイルに出力する] <p>除外レコードを出力するためのファイルを、絶対パスで入力します。</p> <p>- [追記] : ファイルの内容に追記して出力する場合に選択します。</p> <p>- [置換] : ファイルの内容を置き換えて出力する場合に選択します。</p> <p>* 未選択の場合、除外データはどこにも出力されません。</p> <p>○ [指定のデータに置き換える] チェックオンの場合</p> <p>[ルックアップ対象] の [置換データ] を入力します。</p>

注記： [除外されたレコードをテーブルに格納する] にテーブルを設定する場合、ソーステーブルと (Extract 設定時は Extract の設定と) 同じテーブル構成である必要があります。

6. [更新] ボタンをクリックして、ルックアップの情報を保存します。

メモ： ルックアップ名
ルックアップオブジェクトの名称が、入力したオブジェクト名に変更されます。



4.7.2 データウェアハウスの履歴管理

データウェアハウスの履歴管理をする場合に、DimensionLoad と FactLoad を利用します。DimensionLoad 登録により、ディメンション用テーブルへのロードを行い、また、NBT (ナビゲーションブリッジテーブル) を自動生成します。このディメンションテーブルと NBT をもとに FactLoad 登録を行い、データウェアハウスで使用するファクトテーブルを作成します。履歴管理のためには、DimensionLoad と FactLoad 両方の登録が必要です。

- ・ サロゲートキー
サロゲートキー (代理キー) とは、データウェアハウス内部で使用する専用キーのことです。ディメンションテーブルとファクトテーブルをサロゲートキーで連結して、キーを代用します。また、履歴管理をするときに NBT でサロゲートキーを使用します。
サロゲートキーを使用すると次のような利点があります。
①マスタコードが変更されても時系列に沿ったデータを参照できます。
②数値型のキーで UNION するため、早く集計できます。
- ・ NBT (ナビゲーションブリッジテーブル)
ディメンションテーブルとファクトテーブルの間で、橋渡しとして利用するテーブルです。新キー、旧キー、有効期間を持ち、ディメンションテーブルとファクトテーブルだけではできない、ディメンションメンバの世代管理を行います。NBT があることによって、最新データの集計 (新キー) だけでなく、過去のある時点の状態データを集計 (旧キー) することができます。

＜サロゲートキー・NBT を使わない場合＞

「東京」サポートセンターを 2006 年から「東京第 1」と「東京第 2」に分割した場合、名称変更以前のデータを参照しても「東京」の実績を集計できなくなります。名称変更により、2003 年のデータも「東京第 1」として集計されてしまうためです。また、マスタからデータが削除されると旧データの実績を集計できなくなります。

2003年 サポートセンタマスタ 2006年 サポートセンタマスタ

コード	名称		コード	名称
1	東京	分割	1	東京第1
2	奈良		3	大阪
3	大阪	削除	4	福岡
4	福岡		5	札幌
5	札幌		6	東京第2

マスタとファクトを結合すると、マスタの変更により 2003 年の「東京」が「東京第 1」になってしまう。「奈良」はマスタから削除されたため明細に載らない

マスタ変更前 売上明細

年	コード	名称	製品コード	金額
2003	1	東京	12	100,000
2003	1	東京	18	98,000
2003	2	奈良	25	75,500
2003	4	福岡	48	75,500

マスタ変更後 売上明細イメージ

年	コード	名称	製品コード	金額
2003	1	東京第1	12	100,000
2003	1	東京第1	18	98,000
2003	4	福岡	48	75,500
}				
2006	1	東京第1	13	99,500
2006	6	東京第2	11	65,080
2006	5	札幌	54	86,200

＜サロゲートキー・NBT を使う場合＞

「東京」サポートセンターを 2006 年から「東京第 1」と「東京第 2」に分割した場合。NBT を使用すると、マスタの世代管理をすることができます。

- 旧キーを使って集計する場合
NBT(ナビゲーションブリッジテーブル)の旧キーを使用して集計すると、マスタ変更前のデータも変更後のデータも当時の時点にさかのぼって集計することができます。

2003年 サポートセンタマスタ 2006年 サポートセンタマスタ

コード	名称		コード	名称
1	東京	分割	1	東京第1
2	奈良	削除	3	大阪
3	大阪		4	福岡
4	福岡		5	札幌
5	札幌		6	東京第2

2003 年のデータの場合は「東京」で、
2006 年のデータの場合は「東京第 1」
で集計する。

削除した「奈良」のデータも集計可能

2006年 Dimension

サグレート キー	コード	名称
1	1	東京
2	2	奈良
2	3	大阪
3	4	福岡
4	5	札幌
6	6	東京第2
7	1	東京第1

2006年 NBT

旧キー	新キー	FROM	TO	最新 フラグ
1	7	2000-04-01	2006-03-31	0
2	2	2000-04-01	2009-12-31	1
3	3	2000-04-01	2009-12-31	1
4	4	2000-04-01	2009-12-31	1
5	5	2000-04-01	2009-12-31	1
6	6	2006-04-01	2009-12-31	1
7	7	2006-04-01	2009-12-31	1

2006年 Fact

サグレート キー	コード	年	製品コード	金額
1	1	2003	12	100,000
1	1	2003	18	98,000
2	2	2003	25	75,500
4	4	2003	48	75,500
6	6	2006	11	65,080
7	1	2006	13	99,500
5	5	2006	54	86,200

データウェアハウス上で参照する場合、NBT の旧キーを使うと時をさかのぼってデータウェアハウスのデータを集計することができます。「東京」から「東京第 1」に名称変更しても、名称変更前のデータは「東京」として参照できます。

2003年

		2月	3月
センター	製品	売上金額	売上金額
東京	Envoy エクセクティブ*	600000	...
	ノートPC キャリングケース	35000	...
	モニター 15 Super VGA	2000000	...
奈良	512MBPCMCIAカード*	120000	...
	マルチメディア・スピーカー	300000	...
	WindowsXP 1ユーザーパック	670000	...
	マウス・パッド	30000	...
福岡	USBメモリ 256MB	500000	...
	Envoy スタンダード	1000000	...
	モニター 19 Super VGA	450000	...
	OSドキュメント一式	10000	...

- 新キーを使って集計する場合
NBT(ナビゲーションブリッジテーブル)の新キーを使用して集計すると、マスタ変更前のデータも現在の状態で集計することができます。

す。また、マスタから削除されたデータの集計も可能です。

2003年 サポートセンタマスタ 2006年 サポートセンタマスタ

コード	名称	分割	コード	名称
1	東京	→	1	東京第1
2	奈良	→	3	大阪
3	大阪	→	4	福岡
4	福岡	→	5	札幌
5	札幌	→	6	東京第2

2003年のデータも、2006年のデータも「東京第1」で集計する。
削除した「奈良」のデータも集計可能

2005年 Dimension

サブキー	コード	名称
1	1	東京
2	2	奈良
3	3	大阪
4	4	福岡
5	5	札幌
6	6	東京第2
7	1	東京第1

2005年 NBT

旧キー	新キー	FROM	TO	最新フラグ
1	7	2000-04-01	2006-03-31	0
2	2	2000-04-01	2009-12-31	1
3	3	2000-04-01	2009-12-31	1
4	4	2000-04-01	2009-12-31	1
5	5	2000-04-01	2009-12-31	1
6	6	2006-04-01	2009-12-31	1
7	7	2006-04-01	2009-12-31	1

2006年 Fact

サブキー	コード	年	製品コード	金額
7	1	2003	12	100,000
7	1	2003	18	98,000
2	2	2003	25	75,500
4	4	2003	48	75,500
6	6	2006	11	65,080
7	1	2006	13	99,500
5	5	2006	54	86,200

データウェアハウス上で参照する場合、NBTの新キーを使うと現在のマスタの状態ですべてのデータを参照することができます。「東京」から「東京第1」に名称変更しても、名称変更する前のデータを「東京第1」として参照できます。

2003年

センター	製品	2月	3月
		売上金額	売上金額
東京第1	Envoy イクセクティブ	600000	...
	ノートPC キャリングケース	35000	...
	モニター 15 Super VGA	2000000	...
奈良	512MBPCMCIAカード	120000	...
	マルチメディアスピーカー	300000	...
	WindowsXP 1ユーザパック	670000	...
	マウス・パッド	30000	...
福岡	USBメモリ 256MB	500000	...
	Envoy スタンダード	1000000	...
	モニター 19 Super VGA	450000	...
	OSドキュメント一式	10000	...

4.7.2.1 DimensionLoad を登録する

DimensionLoad を用いたジョブを登録することにより、ディメンションテーブルを生成します。

1. [ETL ジョブ登録] 画面の ETL 設定エリアにソースオブジェクト「public.support_center」、ターゲットオブジェクト「public.dim_support_center」を登録します。
2. [DimensionLoad] ボタンをクリックし、ETL 設定エリアに [DimensionLoad] オブジェクトを表示します。



3. ソースオブジェクト「public.support_center」と [DimensionLoad] オブジェクト、ターゲットオブジェクト「public.dim_support_center」をマッピングします。



4. [DimensionLoad] オブジェクトをダブルクリックして、[DimensionLoad] 登録画面を表示します。
5. 以下の情報を入力します。

オブジェクトID	2	オブジェクト名	サポートセンターディメンション
コメント			
自動キャスト	<input checked="" type="checkbox"/> ロード時に自動キャストを行う		
ディメンション設定	サロゲートキー挿入カラム support_center_key		
コミット件数			
ロードエラー判断件数	1	エラー発生時	<input checked="" type="radio"/> 処理を中断する(異常終了) <input type="radio"/> 後続フロー処理を続行する

項目	説明
オブジェクト ID	DimensionLoad のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。
オブジェクト名	ディメンションロード名を入力します。
コメント	ディメンションロードに対するコメントを入力します。
自動キャスト	ソースデータをターゲットの型に合わせてロードする場合は、チェックをオンにします。
ディメンション設定	サロゲートキーを生成するカラムを選択します。
コミット件数	コミット単位となる件数を入力します。
ロードエラー判断件数	ロードエラーと判断するレコード件数を入力します。 例) エラーレコード 100 件でエラーとする場合 → 「100」と入力
エラー発生時	[処理を中断する (異常終了)] ロードエラーが発生したら処理を終了する場合に選択します。 [後続フロー処理を続行する] ロードエラーが発生しても処理を続ける場合に選択します。

6. ソースの列「support_center_id」からターゲットの列「supoort_center_id」へドラッグしてマッピングします。マッピングした列の [ディメンションコード] と [履歴管理項目] が、それぞれ使用可能になります。
7. ディメンションコードを設定する場合は、「support_center_id」の [ディメンションコード] をチェックオンにします。また

「support_center_name」の[履歴管理項目]をチェックオンにして、ディメンションの履歴管理項目とします。

ディメンションコード

public.support_center	public.dim_support_center	ディメンションコード	履歴管理項目
support_center_id	support_center_key		<input type="checkbox"/>
name	support_center_id	<input checked="" type="radio"/>	<input checked="" type="checkbox"/>
prefecture_id	support_center_name	<input type="radio"/>	<input checked="" type="checkbox"/>
address1	prefecture_id	<input type="radio"/>	<input type="checkbox"/>
address2	address1	<input type="radio"/>	<input type="checkbox"/>
address3	address2	<input type="radio"/>	<input type="checkbox"/>
mail	address3	<input type="radio"/>	<input type="checkbox"/>
tel	mail	<input type="radio"/>	<input type="checkbox"/>
fax	tel	<input type="radio"/>	<input type="checkbox"/>
	fax	<input type="radio"/>	<input type="checkbox"/>

注記： [ディメンションコード]をチェックオンにすると、[履歴管理項目]が自動でチェックオンとなり、使用不可になります。ディメンションコードは必ず履歴管理項目になります。

メモ： ディメンションコードと履歴管理項目

・ディメンションコード

FactLoad を用いたジョブを実行するときに、ディメンションコードが使用されます。ジョブ実行時に、ディメンションコードと有効期限で検索したサロゲートキーを、ファクトテーブルにセットします。

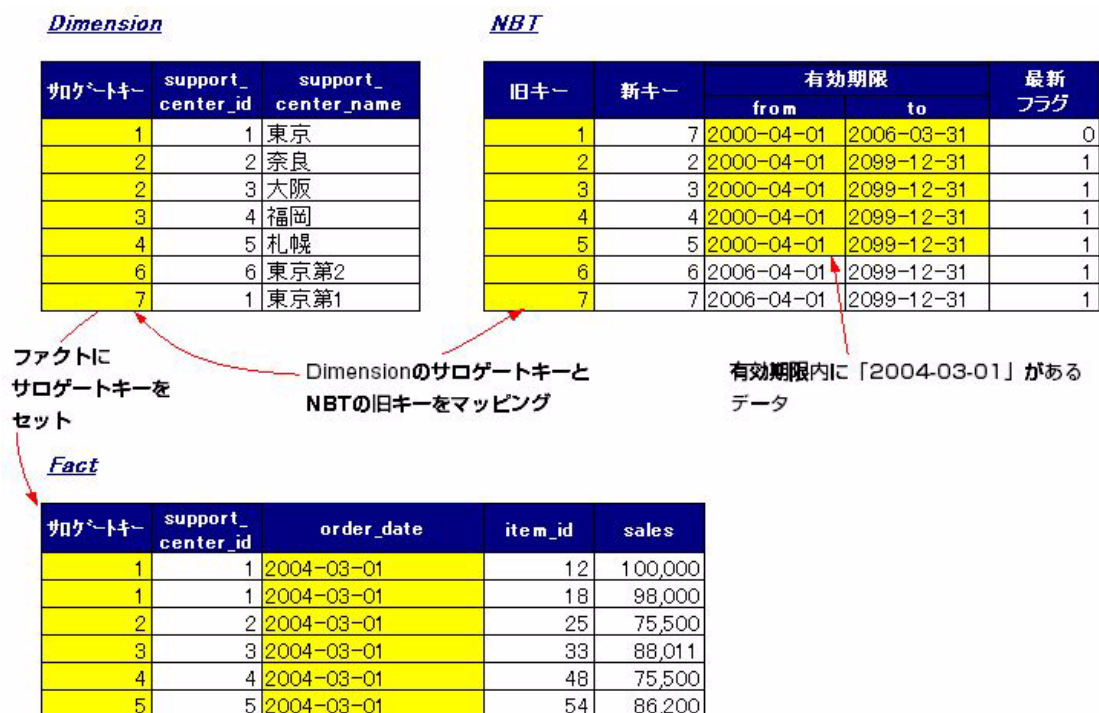
例) FactLoad でディメンションコード「support_center_id」列のジョブを指定し、ファクトデータの日付が「2004-03-01」の場合

①ディメンションテーブルと NBT をサロゲートキー (NBT は旧キー)

で結合します。

②結合した①のデータの中から「support_center_id」、有効期限「2004-03-01」、とが合致するサロゲートキーを取得します。

③取得したサロゲートキーをファクトテーブルにセットします。



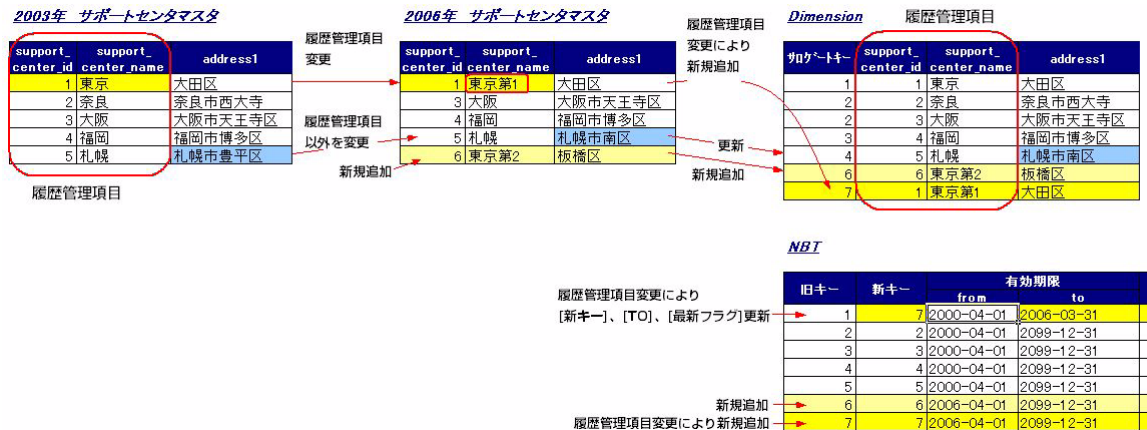
・履歴管理項目

履歴管理項目はNBTの更新を判断するためのカラムで。

DimensionLoad 実行時のNBT更新は以下のとおりです。

パターン	NBT 更新処理
履歴管理項目が異なる	ディメンションメンバが異なる新たなメンバとみなし、データを Insert します。
履歴管理項目は等しいが属性に変更がある	ディメンションメンバは等しいので、属性項目だけ Update します。

例) 履歴管理項目が「support_center_id」、「support_center_name」の場合



1. [更新] ボタンをクリックして、DimensionLoad の情報を保存します。

メモ: DimensionLoad オブジェクト名
DimensionLoad オブジェクトの名称が、入力したオブジェクト名に変更されます。



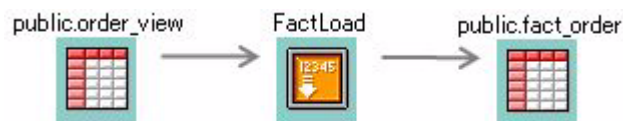
4.7.2.2 FactLoad を登録する

FactLoad を用いたジョブの登録により、ファクトテーブルを生成します。

1. [ETL ジョブ登録] 画面の ETL 設定エリアにソースオブジェクト「public.order_view」、ターゲットオブジェクト「public.fact_order」を登録します。
2. [FactLoad] ボタンをクリックし、ETL 設定エリアに [FactLoad] オブジェクトを表示します。



3. ソースオブジェクト「public.order_view」と [FactLoad] オブジェクト、ターゲットオブジェクト「public.fact_order」をマッピングします。



4. [FactLoad] オブジェクトをダブルクリックして、[FactLoad] 登録画面を表示します。

5. 以下の情報を入力します。

オブジェクトID	2	オブジェクト名	注文ファクト
コメント			
データロード方法	Truncate/Insert ▼		
自動キャスト	<input checked="" type="checkbox"/> ロード時に自動キャストを行う		
コミット件数			
ロードエラー 判断件数	1	エラー発生時	<input checked="" type="radio"/> 処理を中断する(異常終了) <input type="radio"/> 後続フロー処理を続行する

項目	説明
オブジェクト ID	FactLoad のオブジェクト ID を、OpenStaging が自動的に割り振って表示します。
オブジェクト名	ファクトロード名を入力します。
コメント	ファクトロードに対するコメントを入力します。
データロード方法	データロード方法を選択します。 [差分更新を行なう] 差分更新を行なう場合はチェックをオンにします。ターゲットの既存データの内、Insert も Update もされなかったレコードは、ロード処理の最後に削除します。
自動キャスト	ソースデータをターゲットの型に合わせてロードする場合は、チェックをオンにします。
ディメンション設定	サロゲートキーを生成するカラムを選択します。
コミット件数	コミット単位となる件数を入力します。
ロードエラー判断件数	ロードエラーと判断するレコード件数を入力します。 例) エラーレコード 100 件でエラーとする場合 → 「100」と入力
エラー発生時	[処理を中断する (異常終了)] ロードエラーが発生したら処理を終了する場合に選択します。 [後続フロー処理を続行する] ロードエラーが発生しても処理を続ける場合に選択します。

- ソースの列「support_center_id」からターゲットの列「support_center_key」、「support_center_id」へドラッグしてマッピングします。マッピングした列の「日付」と「サロゲートキー変換」が、それぞれ使用可能になります。
- 「order_date」の「日付」をチェックオンにして、有効期限を判断するときの日付カラムとします。
メモ： チェックオンのカラムを日付型とみなします。サロゲートキーを取得するときに、NBT の有効期限と比較するための「日付」カラムとなります。
 ファクトテーブルの中で1列だけ「日付」に設定できます。
- ファクトのカラムの内、どのカラムをディメンションのサロゲートキーとマッピングするか「サロゲートキー変換」で設定します。
 「support_center_key」の「サロゲートキー変換」ー「ディメンション設

定ジョブ」] をチェックオンにして、ETL ジョブを選択可能にします。「サポートセンターディメンション」ジョブを選択します。

サロゲートキーに変換するカラム

public.order_view	public.fact_order	日付	サロゲートキー変換
item_id	item_key	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
campaign_id	item_id	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
support_center_id	campaign_key	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
order_date	campaign_id	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
sales	support_center_key	<input type="radio"/>	<input checked="" type="checkbox"/> デイメンション設定ジョブ 8.サポートセンターディメンション
cost	support_center_id	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ 8.サポートセンターディメンション
	order_date	<input checked="" type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
	amount	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
	sales	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ
	cost	<input type="radio"/>	<input type="checkbox"/> デイメンション設定ジョブ

メモ： デイメンション設定ジョブ

- ・ [デイメンション設定ジョブ] をセットするカラムには、ディメンションコードをマッピングしてください。
- ・ デイメンションテーブルと NBT を検索してサロゲートキーを取得するとき、チェックオンの列をキーとします。ファクトテーブルの複数の列に、[デイメンション設定ジョブ] を選択することができます。

9. [更新] ボタンをクリックして、FactLoad の情報を保存します。

メモ： FactLoad 名

FactLoad オブジェクトの名称が、入力したオブジェクト名に変更されます。



4.8 ETL ジョブを編集する

1. オブジェクトツリーの [ETL ジョブ] アイコンの内、編集したい ETL ジョブを選択して [ETL ジョブ情報] 画面を表示します。
2. 項目を編集します。

注記： Extract 情報を変更すると、次の処理オブジェクトが壊れることがあります。

注記： ジョブ ID は編集できません。

3. [更新] ボタンをクリックして、ETL ジョブの情報を保存します。

4.9 ETL ジョブを削除する

1. オブジェクトツリーの [ETL ジョブ] アイコンの内、削除したい ETL ジョブを選択して [ETL ジョブ情報] 画面を表示します。

2. [削除] ボタンをクリックして、ETL ジョブを削除します。削除後、画面は [ETL ジョブ登録] 画面へ切り替わります。

メモ： 削除した ETL ジョブのアイコンが、オブジェクトツリーから削除されます。

4.10 プログラムコードを編集する

ETL ジョブ登録時に生成されたプログラムを確認、または編集することができます。

1. オブジェクトツリーの [ETL ジョブ] - [(ETL ジョブ名)] の「+」アイコンをクリックして、[プログラムコード] アイコンを表示します。
2. [プログラムコード] アイコンをクリックして [プログラムコード情報] 画面を表示します。

プログラムコード情報

コマンド1 DBブリッジ オブジェクトID:[1, 4, 5]

/usr/local/tomcat5.5/webapps/OpenStagingTest/WEB-INF/conf/sql/1/createTempTable_1_5.sql

```
create table os_t_temp_1_5 ( item_id varchar2(16),short_name varchar2(40),long_name varchar2(100),amount number
(10,0),fixed_price number(10,0),cost_price number(12,2),selling_price number(10,0),sub_category_id varchar2
(16) )
```

☐ カスタマイズを行う

/usr/local/tomcat5.5/webapps/OpenStagingTest/WEB-INF/conf/sql/1/extract_1_4.sql

```
select * from public.item
```

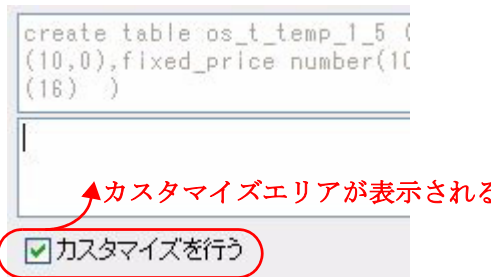
☐ カスタマイズを行う

/usr/local/tomcat5.5/webapps/OpenStagingTest/WEB-INF/conf/sql/1/insertTable_1_5.sql

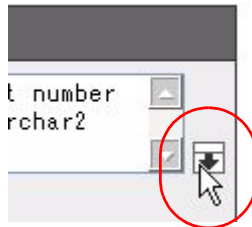
```
insert into os_t_temp_1_5 values(?,?,?,?,?,?,?,?)
```

☐ カスタマイズを行う

3. プログラムコードを編集する場合は「カスタマイズを行う」チェックボックスをオンにして、カスタマイズ用のフォームを表示します。



- メモ： プログラムコードの表示エリアを広げる場合は、カスタマイズフォーム右端のエリア拡張用アイコンをクリックします。



4. 編集したプログラムコードの情報を登録する場合は、「更新」ボタンをクリックします。



テーブル移行ジョブの作成

OpenStaging では異なったデータソース間で、任意のテーブルをエクスポートすることができます。業務用データをデータウェアハウス側に移行して使用する場合や、テスト環境を作る場合などに利用できます。

この章は以下のとおり構成されています。

- ・ [テーブル移行ジョブを登録する](#)
- ・ [テーブル移行ジョブを編集する](#)
- ・ [テーブル移行ジョブを削除する](#)

5.1 テーブル移行ジョブを登録する

異なるデータベースへ、テーブルをエクスポートすることができます。インデックスが作成されている場合は、インデックスもエクスポートします。

1. オブジェクトツリーの [ジョブ] - [テーブル移行ジョブ] を選択して、[テーブル移行ジョブ登録] 画面を表示します。

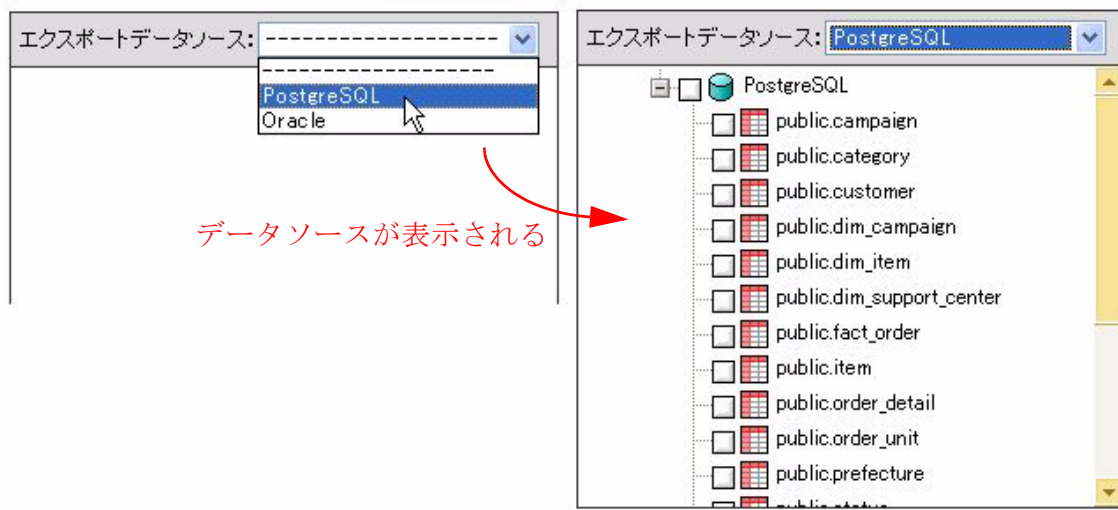


2. 以下の情報を入力します。

ジョブID		ジョブ名	PostgreSQLからOracleへ 移行
コメント			
ログファイル	/tmp/export/user01/joblogfile.log		ログ参照
ログレベル	MIDDLE		
オブジェクト作成方法	Drop/Create		
データコピー方法	Truncate/Insert		<input checked="" type="checkbox"/> データインサート時にインデックスを再構築する
コミット件数	100	終了コード	正常時: 0 異常時: 1

項目	説明
ジョブ ID	テーブル移行ジョブ ID を OpenStaging が自動的に割り振って表示します。 * 新規作成時は非表示となります。
ジョブ名	テーブル移行ジョブ名を入力します。
コメント	テーブル移行ジョブに対するコメントを入力します。
ログファイル	ログファイルの保存先を絶対パスで入力します
ログレベル	ログレベルを選択します。
オブジェクト作成方法	Export、Import 時のオブジェクト作成方法を選択します。
データコピー方法	Export、Import 時のデータコピー方法を選択します。 [データインサート時にインデックスを再構築する] Insert 時にいったんインデックスを削除して、パフォーマンスを上げる場合にチェックをオンにします。
コミット件数	コミット単位となる件数を入力します。
終了コード	テーブル移行ジョブ実行時の終了コードを入力します。 [正常時] 0 [異常時] 異常時の戻り値を入力します。 *[正常時] は変更できません。
最終更新日	テーブル移行ジョブの最終更新日を表示します。変更はできません。 * 新規作成時は非表示となります。

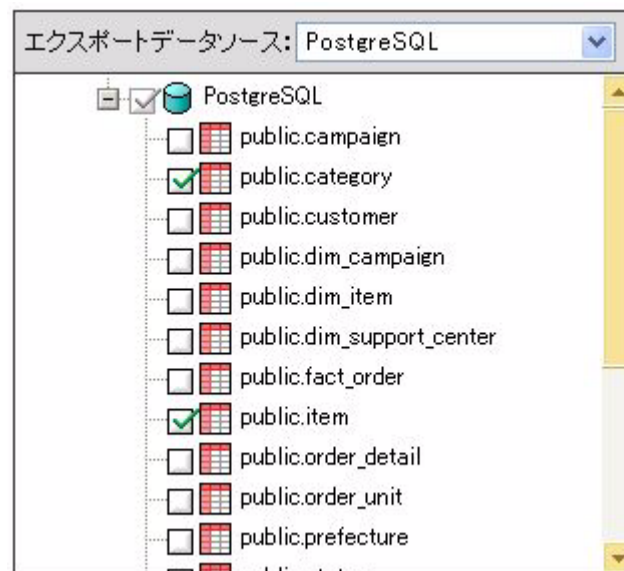
3. [エクスポートデータソース] のリストからエクスポートしたいデータソースを選択すると、データソースのツリーリストが表示されます。



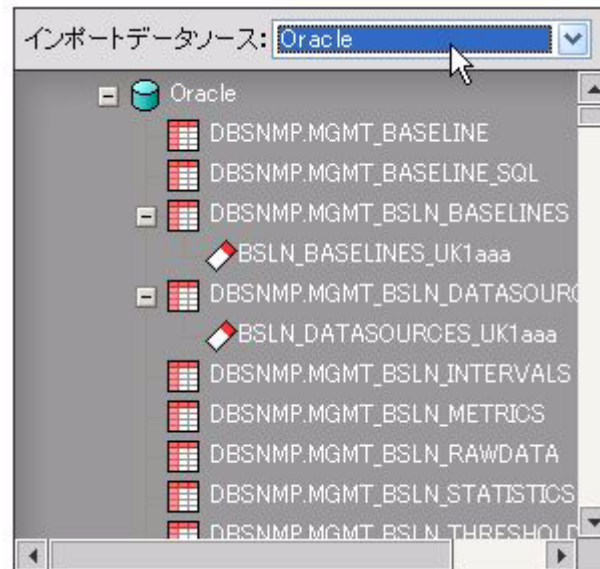
注記： [エクスポートデータソース] のリストにビューは表示されません。ビューをエクスポートすることはできません。

メモ： エクスポートデータソースのツリー
ツリーには、テーブルと B ツリーインデックスが表示されます。
表示順は、OpenStaging が自動的に割り振った ID の昇順です。

4. データソース「PostgreSQL」のツリーから、「public.item」テーブル、「public.sub_category」テーブル、「public.category」テーブルを選択してチェックをオンにします。



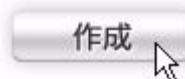
5. [インポートデータソース] のリストからインポート先となるデータソースを選択します。選択したデータソースのツリーが表示され、データソースの構成を確認することができます。



注記： インポートデータソースのツリーは編集できません。

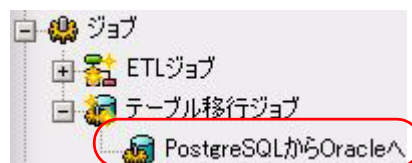
メモ： インポートのデータソースツリー
ツリーには、テーブルと B ツリーインデックスが表示されます。
表示順は、OpenStaging が自動的に割り振った ID の昇順です。

6. [作成] ボタンをクリックして、テーブル移行ジョブの情報を保存します。



メモ： ログ参照ボタン
[ログファイル] で指定したパスに保存されているログファイル
を表示します。

メモ： ツリーオブジェクト [ジョブ] - [テーブル移行ジョブ] に、作成したテーブル移行ジョブのアイコンが追加されます。



7. テーブル移行ジョブを実行する場合は [実行] ボタンをクリックします。設定した内容でテーブル移行ジョブが実行されます。



注記： テーブル移行ジョブの情報を登録したい場合は、実行処理後に [作成] ボタンをクリックしてください。

- XML ファイルを編集する場合は、[XML] ボタンをクリックします。XML の編集については『[4.4XML を編集する](#)』を参照してください。

5.2 テーブル移行ジョブを編集する

- オブジェクトツリーの [テーブル移行ジョブ] アイコンの内、編集したいテーブル移行ジョブを選択して [テーブル移行ジョブ情報] 画面を表示します。
- 項目を編集します。

注記： ジョブ ID は編集できません。

- [更新] ボタンをクリックして、テーブル移行ジョブの情報を保存します。
- テーブル移行ジョブを実行する場合は、[実行] ボタンをクリックします。設定した内容でテーブル移行ジョブが実行されます。

注記： 更新したテーブル移行ジョブの情報を登録したい場合は、実行処理後に [更新] ボタンをクリックしてください。

5.3 テーブル移行ジョブを削除する

- オブジェクトツリーの [テーブル移行ジョブ] アイコンの内、削除したいテーブル移行ジョブを選択して [テーブル移行ジョブ情報] 画面を表示します。
- [削除] ボタンをクリックして、テーブル移行ジョブを削除します。削除後、画面は [テーブル移行ジョブ登録] 画面へ切り替わります。



メモ： 削除したテーブル移行ジョブのアイコンが、オブジェクトツリーから削除されます。

スケジュールの作成

ETL ジョブやテーブル移行ジョブの実行スケジュールを作成することができます。

この章は、以下のとおり構成されています。

- [スケジュールを登録する](#)
- [スケジュールを編集する](#)
- [スケジュールを削除する](#)

6.1 スケジュールを登録する

ETL ジョブ、テーブル移行ジョブのスケジュールを組んで実行します。日時を指定して、スケジュール実行後にメールで通知することができます。

1. オブジェクトツリーの「スケジュール」を選択して、「スケジュール登録」画面を表示します。



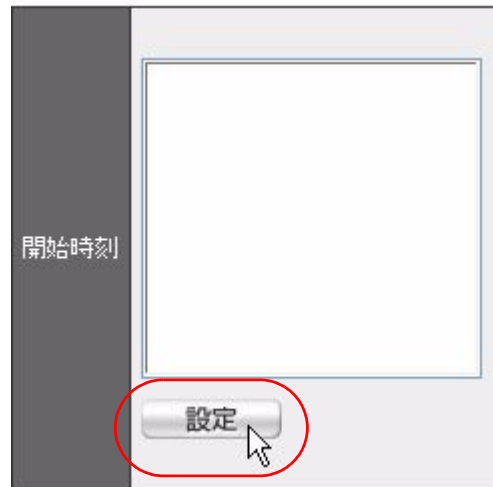
2. 以下の情報を入力します。

スケジュールID	スケジュール名	商品ETLジョブのスケジュール
コメント		
実行する/しない	<input checked="" type="checkbox"/> このスケジュールを実行する	
ログファイル	<input type="text" value="/schedule/item_schedule.log"/> <input type="button" value="ログ参照"/>	
	<input checked="" type="radio"/> 追記 <input type="radio"/> 置換	
メール送信	<input checked="" type="checkbox"/> スケジュール実行終了時にメールを送信する To: <input type="text" value="user01@openstaging.co.jp"/> <input type="button" value="詳細設定"/>	
重複実行禁止	<input checked="" type="radio"/> このスケジュールが実行中のときは実行しない <input type="radio"/> 実行中のスケジュールに、このスケジュールのジョブが含まれているときは実行しない	

項目	説明
スケジュール ID	スケジュール ID を OpenStaging が自動的に割り振って表示します。 * 新規作成時は非表示となります。
スケジュール名	スケジュール名を入力します。
コメント	スケジュールに対するコメントを入力します。
実行する / しない	スケジュールの実行を有効にする場合は、チェックオンにします。
ログファイル	ログファイルの保存場所を絶対パスで入力します。 -[追記]: ファイルの内容に追記して出力する場合に選択します。 -[置換]: ファイルの内容を置き換えて出力する場合に選択します。



項目	説明
メール送信	<p>[スケジュール実行終了時にメールを送信する]</p> <p>スケジュール実行後に通知メールを送信する場合は、チェックをオンにします。</p> <p>[To:]</p> <p>送信先のアドレスを入力します。</p> <p>[詳細設定]</p> <p>クリックすると、[CC:]、[BCC:]、[Reply-To:] 入力のテキストボックスが表示されます。必要に応じて入力します。</p> <p>*Reply-To:</p> <p>通知メールに返信する場合の、返信先アドレスを指定することができます。</p> <p>例) ジョブ実行通知メールの返信先を「***@***.co.jp」に設定する場合→ [Reply-To:] に「***@***.co.jp」と入力</p>
重複実行禁止	<p>このスケジュールの重複実行を禁止する判断基準を選択します。</p> <p>・[このスケジュールが実行中のときは実行しない]</p> <p>スケジュール単位で重複実行を禁止する場合に選択します。[スケジュール] を選択した場合、このスケジュールを重複して実行することはできません。</p> <p>例) スケジュール ID:6 の場合</p> <p>スケジュール ID:6 を即時実行後、すぐに続けてスケジュール ID:6 を即時実行すると実行時エラーが発生します。</p> <p>・[実行中のスケジュールに、このスケジュールのジョブが含まれているときは実行しない]</p> <p>ジョブ単位で重複実行を禁止する場合に選択します。他のスケジュールで実行されているジョブが、このスケジュールでセットしているジョブと等しい場合、このスケジュールの実行を禁止します。</p> <p>例) スケジュール ID:6 にジョブ ID:2、スケジュール ID:10 にジョブ ID:2 をセットしている場合</p> <p>スケジュール ID:10 を実行中に、スケジュール ID:6 を実行しても、実行時エラーが発生します。</p>
最終更新日時	<p>スケジュールの最終更新日を表示します。変更はできません。</p> <p>* 新規作成時は非表示となります。</p>


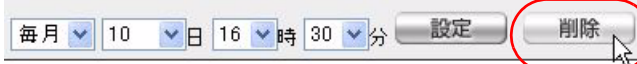
3. スケジュールの開始時刻を設定する場合は、[設定] ボタンをクリックして [開始時刻] 設定画面を表示します。



4. 以下の情報を入力します。

日時指定実行	2006 年 2 月 8 日 18 時 20 分 <input type="button" value="追加"/>
繰り返し実行	毎月 10 日 03 時 00 分 <input type="button" value="追加"/>
トリガー実行	トリガーファイルパス: /root/x <input checked="" type="checkbox"/> 実行後ファイル処理 <input checked="" type="radio"/> 削除 <input type="radio"/> 変更 変更後ファイルパス: <input type="text"/> <input type="button" value="追加"/>
開始時刻	毎月 10 日 03:00 トリガー実行 /root/x

項目	説明
日時指定実行	<p>日時を指定したスケジュールを作成する場合、項目を選択して[追加]ボタンをクリックします。</p>  <p>－[年]、[月]、[日]、[時]、[分]を選択します。</p>
繰り返し実行	<p>ジョブを繰り返し実行する場合は、項目を選択して[追加]ボタンをクリックします。</p> <p>－[毎年/毎月/毎週/毎日]、[日]、[時]、[分]を選択します。</p> <p>*「毎週」を選択すると、「曜日」のリストが表示されます。</p> 

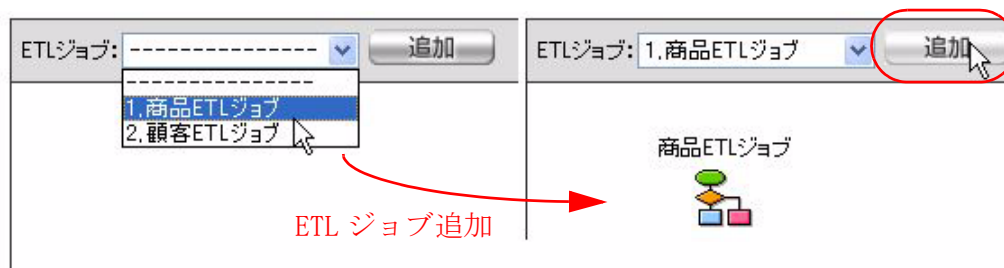
項目	説明
トリガー実行	<p>トリガーファイルを使用する場合は、項目を入力して[追加]ボタンをクリックします。</p> <p>[トリガーファイルパス]</p> <p>トリガーファイルの保存先を絶対パスで入力します。</p> <p>[実行後ファイル処理]</p> <p>スケジュール実行後にトリガーファイルを[削除]または[変更]する場合、チェックをオンにします。</p> <p>実行後ファイルを[削除]するか[変更]するかラジオボタンを選択します。</p> <p>[変更後ファイルパス]</p> <p>[実行後ファイル処理]で[変更]を選択した場合、変更後のファイルパスを絶対パスで入力します。</p>
開始時刻	<p>[日時指定実行]、[繰り返し実行]、[トリガー実行]で追加した値が表示されます。</p> <p>○ [開始時刻] を編集する場合</p> <p>変更したい開始時刻をクリックすると、指定した開始時刻が下欄に表示されます。項目を変更して[設定]ボタンをクリックします。</p>  <p>○ [開始時刻] を削除する場合</p> <p>削除したい開始時刻をクリックすると、指定した開始時刻が下欄に表示されます。下欄の[削除]ボタンをクリックします。</p> 

メモ： 開始時刻を設定せずに、スケジュールをすぐに実行することもできます。[スケジュール登録]画面の[作成して即時実行]ボタンをクリックします。

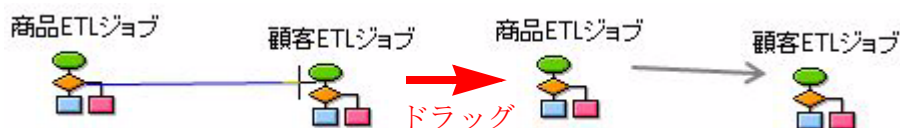
作成して即時実行

5. [更新]ボタンをクリックして開始時刻の情報を登録します。

6. ETL ジョブをスケジュールする場合は、[ETL ジョブ] 選択リストからジョブを選択して [追加] ボタンをクリックします。ジョブ設定エリアに ETL ジョブのアイコンが追加されます。



7. 連続してジョブを設定する場合は、続けて [ETL ジョブ] リストからジョブを選択し [追加] ボタンをクリックします。「顧客 ETL ジョブ」を選択して、[追加] ボタンをクリックします。
8. 複数のジョブを実行する場合、ジョブの実行順序を設定します。ジョブ「商品 ETL ジョブ」からジョブ「顧客 ETL ジョブ」へドラッグしてジョブをマッピングします。

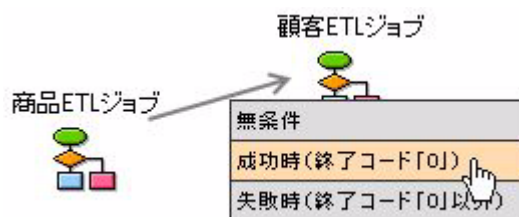


9. ジョブの実行順序を分岐して設定する場合は、実行条件を選択します。マッピングの線上で右クリックして、条件メニューを表示します。

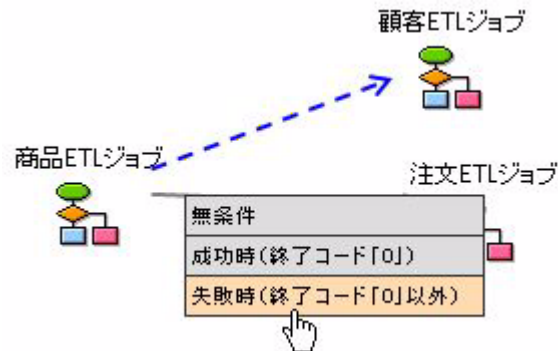
メモ： スケジュールの分岐

例えば商品 ETL ジョブが成功したら顧客 ETL ジョブへ、商品 ETL ジョブが失敗したら注文 ETL ジョブへというように、分岐したスケジュールを実行することができます。

- ①商品 ETL ジョブ→顧客 ETL ジョブのマッピング線を右クリックします。
- ②右クリックメニュー「無条件 / 成功時 / 失敗時」の中から、「成功時」を選択します。

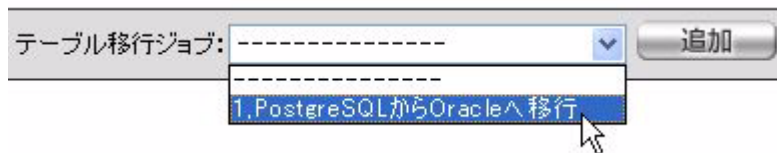


- ③商品 ETL ジョブ→注文 ETL ジョブ間のマッピング線を右クリックします。
- ④右クリックメニュー「無条件 / 成功時 / 失敗時」の中から、「失敗時」を選択します。



注記： マッピング線の初期設定は「無条件」です。分岐条件をすべて無条件にすることはできません。スケジュールを分岐する場合は、分岐条件を変更してください。

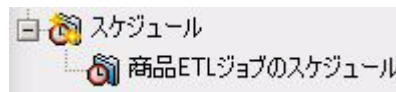
10. テーブル移行ジョブをスケジュールする場合は、[テーブル移行ジョブ] 選択リストからジョブを選択して [追加] ボタンをクリックします。ジョブ設定エリアにテーブル移行ジョブのアイコンが追加されます。



メモ： ETL ジョブとテーブル移行ジョブを組み合わせるスケジュールすることもできます。

11. テーブル移行ジョブの実行順を分岐する場合も、ETL ジョブと同じように設定できます。
12. [作成] ボタンをクリックして、スケジュールの情報を登録します。

メモ： ツリーオブジェクト [スケジュール] に、作成したスケジュールのアイコンが追加されます。



13. スケジュールをすぐに実行したい場合は、[作成して即時実行] ボタンをクリックします。スケジュール内容を登録してから、スケジュールが実行されます。

メモ： [開始時刻] を設定している場合は、即時実行した後、[開始時刻] で指定した日時にスケジュールを実行します。

6.2 スケジュールを編集する

1. オブジェクトツリーの [スケジュール] アイコンの内、編集したいスケジュールを選択して [スケジュール情報] 画面を表示します。

2. 項目を編集します。

注記： スケジュール ID は編集できません。

3. [更新] ボタンをクリックして、スケジュールの情報を登録します。
4. スケジュールをすぐに実行したい場合は、[更新して即時実行] ボタンをクリックします。スケジュール内容を更新してから、スケジュールが実行されます。

メモ： [開始時刻] を設定している場合は、即時実行した後、[開始時刻] で指定した日時にスケジュールを実行します。

6.3 スケジュールを削除する

1. オブジェクトツリーの [スケジュール] アイコンの内、削除したいスケジュールを選択して [スケジュール情報] 画面を表示します。
2. [削除] ボタンをクリックして、スケジュールを削除します。削除後、画面は [スケジュール登録] 画面へ切り替わります。

メモ： 削除したスケジュールのアイコンが、オブジェクトツリーから削除されます。

OpenStaging 応用

この章では OpenStaging 操作の応用について説明します。

この章は、以下のとおり構成されています。

- [ユーザー・パスワード設定](#)
- [コマンドラインからのジョブの実行](#)
- [使い方の応用例](#)
- [サンプルデータ](#)
- [未サポート RDB の OpenStaging 実装方法](#)

7.1 ユーザー・パスワード設定

OpenStaging では、ログイン認証時の使用するユーザーを定義することができます。デフォルトで、admin ユーザーのみが登録されています。

- openstaging ユーザー
ユーザー名：openstaging
パスワード：openstaging
- ユーザー名やパスワードの変更、またはユーザー追加をする場合は次の管理ファイルを編集してください。
OpenStaging ホームディレクトリ /WEB-INF/conf/user/user.conf

注記： 記述方法

- ユーザー名、パスワードはカンマ区切りで記述してください。
- ユーザーを複数定義する場合は、ユーザーごとに改行してください。

7.2 コマンドラインからのジョブの実行

OpenStaging では、付属のスケジュール機能を使わずにコマンドラインからのジョブ実行もサポートしています。そのため「設定は OpenStaging で、運用は別途運用管理ツールで」という使い方もできます。

例) ETL ジョブ ID : 37 を実行する場合のコマンド

```
java -classpath %クラスパス%  
jp.co.iafc.openstaging.jobmanager.JobExecuter ETL 37
```


パラメータ

引数	コマンド	機能
第 1 引数 : ジョブの種類 (必須)	ETL ExpImp	ETL ジョブの場合は「ETL」を、テーブル移行ジョブの場合「ExpImp」を指定します。
第 2 引数 : ジョブ番号 (必須)	999 (数値)	実行するジョブの番号を指定します。
第 3 引数 : 処理日 (任意)	Today Yesterday yyyy/mm/dd	ETL ジョブ実行時の処理日を指定します。 *ETL ジョブ登録画面で設定した [処理日] よりも優先したい場合に、処理日を指定します。
第 4 引数 : 最大取得件数 (任意)	999 (数値)	ETL ジョブ実行時の最大取得件数を指定します。 *csv、Extract、DBブリッジなど各オブジェクトで設定した値よりも優先したい場合に指定します。
第 5 引数 : 実行コマンド番号 (任意)	999 (数値)	ETL ジョブを部分的に実行する場合、実行するコマンド番号をカンマ区切りで指定します。 例) ETL ジョブの実行コマンド 1 ～ 3 を実行する場合 → 「1, 2, 3」と入力

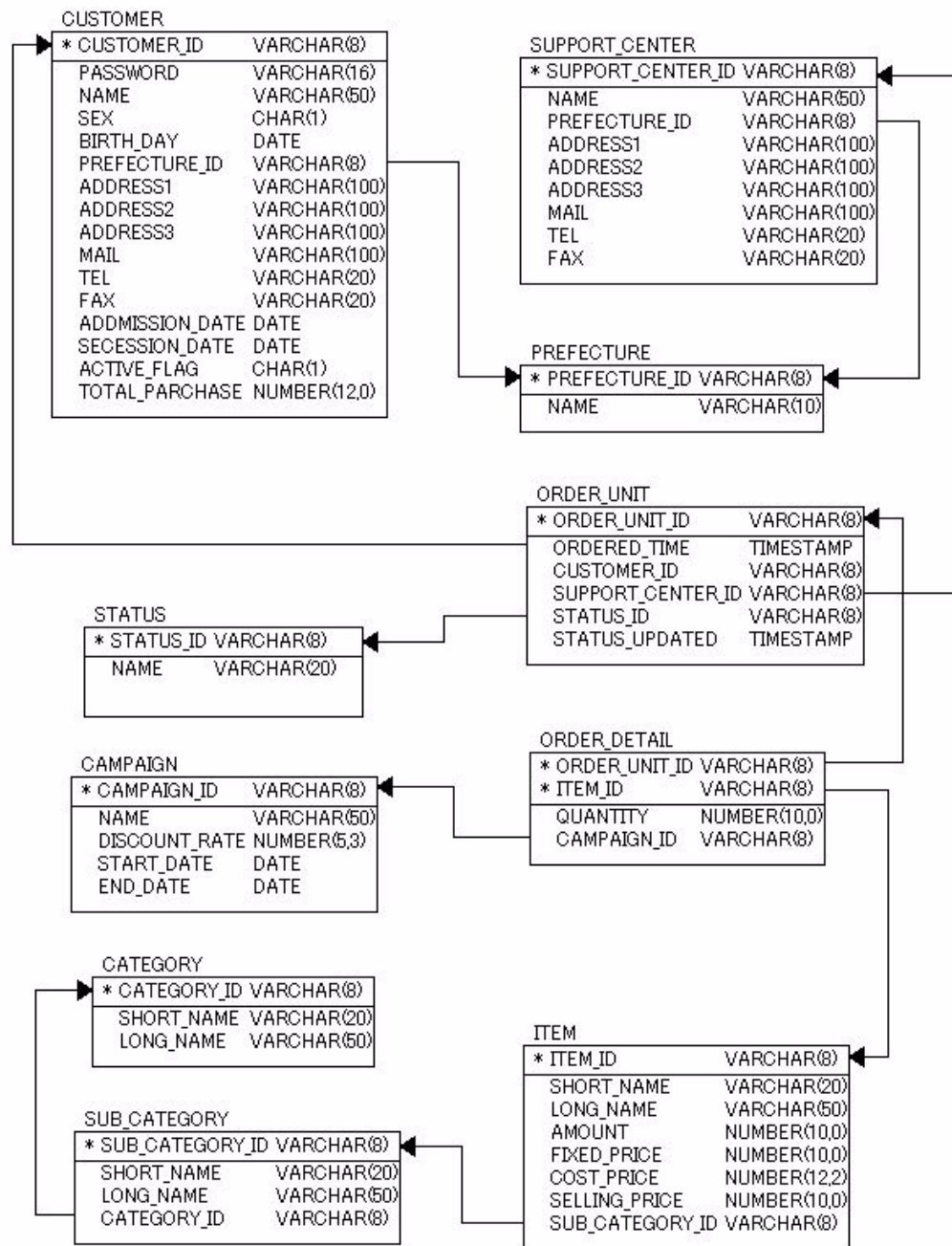
注記： [処理日]、[最大取得件数]、[実行コマンド番号] は、ETL ジョブ実行の場合だけ利用できます。

7.3 サンプルデータ

OpenStaging では、動作確認やパフォーマンス検証のために PostgreSQL、MySQL、Oracle のそれぞれにサンプルデータを用意しています。

7.3.1 サンプル1：業務用テーブルER図

サンプルとして以下のテーブル構成のデータを用意しています。

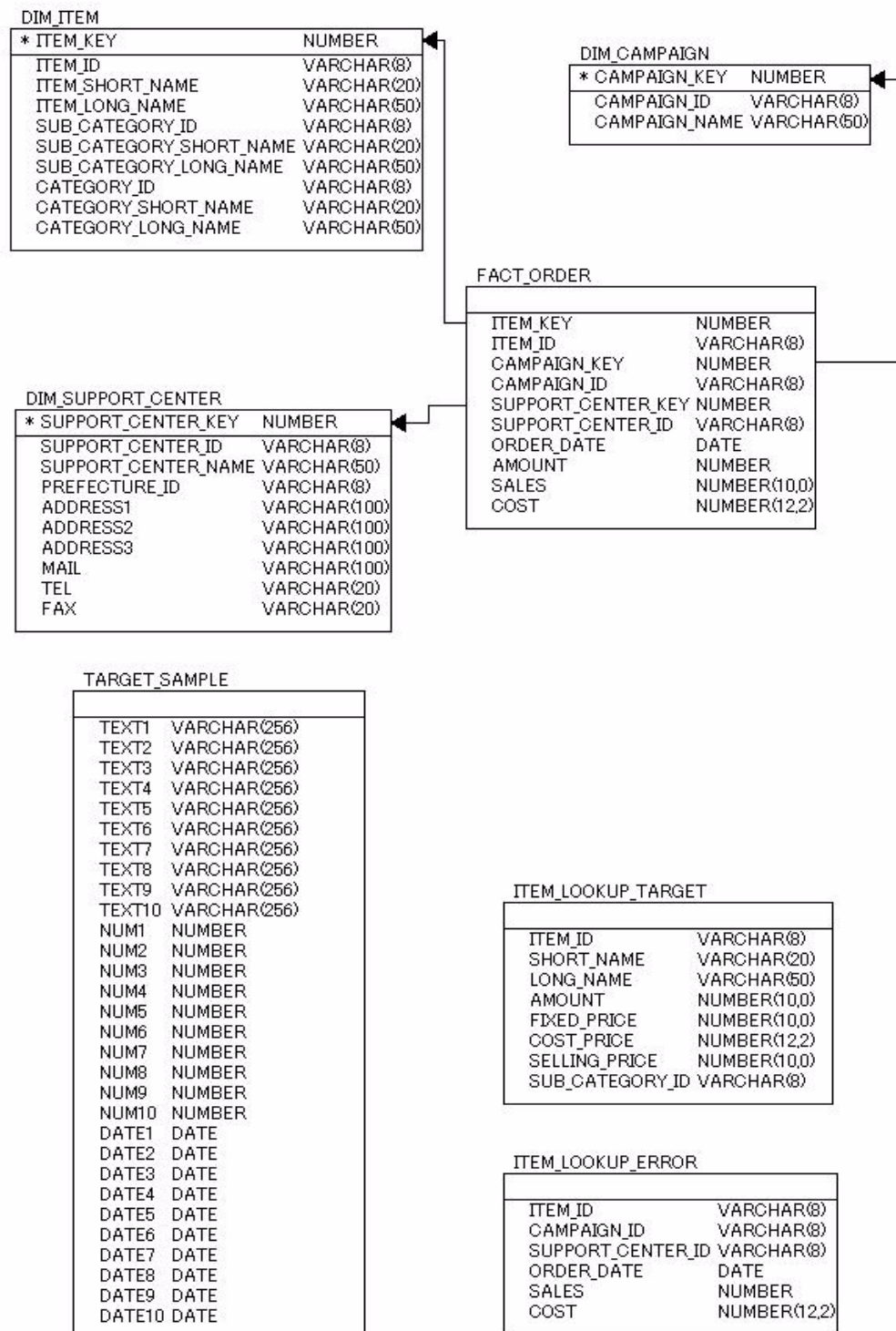


これらのテーブル一覧は以下のとおりです。

テーブル名	説明
CAMPAIGN	キャンペーンマスタ
CATEGORY	商品大分類マスタ
SUB_CATEGORY	商品小分類マスタ
ITEM	商品マスタ
CUSTOMER	顧客マスタ
PREFECTURE	都道府県マスタ
SUPPORT_CENTER	サポートセンターマスタ
ORDER_DETAIL	注文明細データ
ORDER_UNIT	注文データ
STATUS	注文状況マスタ
ORDER_VIEW	注文明細ビュー : 「CAMPAIGN」 テーブル、「ITEM」 テーブル、「ORDER_DETAIL 」 テーブル 「ORDER_UNIT」 テーブル

7.3.2 サンプル2 : DWH 用テーブル ER 図

サンプルとして以下のスタースキーマと検証用テーブルを用意しています。



これらのテーブル一覧は以下のとおりです。

物理名	論理名
DIM_CAMPAIN	キャンペーンディメンションテーブル
DIM_ITEM	商品ディメンションテーブル
DIM_SUPPORT_CENTER	サポートセンターディメンションテーブル
FACT_ORDER	注文ファクトテーブル
TARGET_SAMPLE	汎用ターゲットテーブル
ITEM_LOOKUP_ERROR	ルックアップ処理エラーレコード格納用テーブル
ITEM_LOOKUP_TARGET	ルックアップターゲットテーブル

7.4 使い方の応用例

OpenStaging では、生成されるプログラムだけでなく、OpenStaging の画面の初期表示項目、ストアドプロシージャの雛形、SQL の構文などを自由にカスタマイズできます。

次の箇所をカスタマイズできます。

- ・ ジョブやスケジュールなど各種設定 XML ファイル
- ・ ETL ジョブにより生成されたプログラム（GUI より可能）
- ・ ストアドプロシージャのテンプレート
- ・ SQL の構文
- ・ データ型変換ルール

7.4.1 応用例 1 : xml ファイル

スケジュール画面で初期表示されるログファイルパスを指定します。

1. 「OpenStaging ホームディレクトリ /WEB-INF/conf/scheduler/template.xml」を開きます。
2. LogFile 設定 (Schedule/Log) を編集します。〈Log〉タグに、ログファイルパスを絶対パスで入力します。
編集前：

```
<Log execute="true" append="true"></Log>
```

 編集後：

```
<Log execute="true" append="true">/var/openstaginglog/xxx.log</Log>
```

7.4.2 応用例 2 : SQL 構文

テーブル移行ジョブでテーブルを作成するときに、表領域を指定します。

1. OpenStaging ホームディレクトリ /WEB-INF/conf/sql/orcl.sql」を開きます。
2. CreateTable シンタックス (SQL/Syntax/CreateTable) に「tablespace 表領域名」を追記します。
編集前：

```
<CreateTable>create table %table_name% ( %columns% %constraints% )</CreateTable>
```

編集後：

```
<CreateTable>create table %table_name% ( %columns% %constraints% )


```

7.4.3 応用例 3：データ型変換ルール

ETL ジョブの DB ブリッジ（異なるデータソース間のデータ移行）について、内部的に作成される一時テーブルのデータ型変換ルールを変更します。

例えば、Oracle から MySQL へ変換するときに Oracle の VARCHAR2 型を MySQL の VARCHAR 型ではなく TEXT 型に変換するようにしたい場合

1. 「OpenStaging ホームディレクトリ /WEB-INF/conf/sql/sqltypes.xml」を開きます。
2. SQLTypes 設定（SQLTypesDef/Groups/Group/SQLTypes@db="MySQL"）を編集します。上位にあるデータ型が優先されるため、varchar 行の上に text 行を指定します。

編集前：

```
<SQLType name="varchar" maxLength="255" sizeRatio="2" />
<SQLType name="text" length="false" destonly="true" />
<SQLType name="varbinary" maxLength="255" />
```

編集後：

```
<SQLType name="text" length="false" destonly="true" />
<SQLType name="varchar" maxLength="255" sizeRatio="2" />
<SQLType name="varbinary" maxLength="255" />
```

7.5 未サポート RDB の OpenStaging 実装方法

OpenStaging では、標準で以下のデータベースをサポートしています。

- Oracle
- PostgreSQL
- MySQL

しかし次の設定ファイルなどを追加することにより、これ以外のデータベースにも比較的容易に対応できます。

- ストアドプロシージャのテンプレートファイル
- SQL 定義ファイル
- データ型変換ルール定義ファイル
- DB 対応 Java プログラム

メモ： 詳しくは、『OpenStaging 技術解説書』をご覧ください。